MARCOS AURÉLIO BASSO

UM MÉTODO ROBUSTO PARA MODELAGEM 3D DE AMBIENTES INTERNOS USANDO DADOS RGB-D

CURITIBA

2015

MARCOS AURÉLIO BASSO

UM MÉTODO ROBUSTO PARA MODELAGEM 3D DE AMBIENTES INTERNOS USANDO DADOS RGB-D

Tese apresentada ao Programa de Pós-Graduação em Ciências Geodésicas, setor de Ciências da Terra, Universidade Federal do Paraná, como requisito parcial para obtenção do título de Doutor em Ciências Geodésicas.

Orientador: Daniel Rodrigues dos Santos

CURITIBA

2015

Basso, Marco Aurélio
Um método robusto para modelagem 3D de ambientes internos usando dados RGB-D / Marco Aurélio Basso. – Curitiba, 2015.
120 f. : il.; graf., tab.

Tese (doutorado) – Universidade Federal do Paraná, Setor de Ciências da Terra, Programa de Pós-Graduação em Ciências Geodésicas.
Orientador: Daniel Rodrigues dos Santos
Bibliografia: p.106-112

1. Modelagem de dados. 2. Detectores. 3. Imagem tridimensional I. Santos, Daniel Rodrigues dos. II. Título.

CDD 526.982

TERMO DE APROVAÇÃO

MARCOS AURÉLIO BASSO

"UM MÉTODO ROBUSTO PARA MODELAGEM 3D DE AMBIENTES INTERNOS USANDO DADOS RGB-D"

Tese nº 99 aprovada como requisito parcial do grau de Doutor no Programa de Pós-Graduação em Ciências Geodésicas, Setor de Ciências da Terra da Universidade Federal do Paraná, pela seguinte banca examinadora:

Orientador: Prof. Dr. Daniel Rodrigues dos Santos
Departamento de Geomática, UFPR

Prof. Dr. Antonio Maria Garcia Tommaselli
Universidade Estadual Paulista, UNESP

Prof. Dr. Mário Luiz Lopes Reiss
Universidade Federal do Rio Grande do Sul, UFRGS

Prof. Dr. Quintino Dalmolin
Universidade Federal do Paraná, UFPR

Prof. Dr. Jorge Antonio Silva Centeno
Departamento de Geomática, UFPR

Curitiba, 12 de novembro de 2015.

Dedicatória

Dedico a realização deste trabalho à minha família, a meus pais Rozimbo e Enedina, e a meus irmãos Roseslane, Gilmara, Jean e Dean.

In Memorian ao Professor Camil Gemael.

Agradecimentos

Pela concretização deste trabalho, gostaria de fazer os seguintes agradecimentos:

- A CNPq pela bolsa concedida;
- Ao Prof. Dr Daniel pela orientação do trabalho;
- Aos meus colegas de laboratório Nadissom, Jaqueline, Eliseu, Cauê;
- Aos professores avaliadores da banca o Prof. Dr.Tommaselli, Prof. Dr Mario Reiss, Prof. Dr. Quintino, Prof. Dr. Centeno que através de sugestões e criticas colaboraram em engrandecer meu trabalho;
- Em especial também gostaria de agradecer a Prof. Dr. Edison Mitshita, Prof. Dr. Silvio de Freitas, Prof. Dr. Henrique Firkowski, Hideo Araki, a coordenadora do PPGCG Profa. Dra Luciene Delazari, e a nossa querida secretária Monica;
- Aos meus amigos André, Prof. Dr Wander e Gloria , Henry Montecino, Felipe, Andrey, Prof. Erika e a todos os outros que ajudaram de forma direta e indireta a realização deste trabalho!;
- Também não poderia deixar de agradecer aos desenvolvedores das bibliotecas OpenCV, MRPT, PCL e também do LaTeX, pela ajuda no meu trabalho!

Epígrafe

"Procure ser um homem de valor,
em vez de ser um homem de sucesso".

Albert Einstein

Resumo

O objetivo deste trabalho é propor um método robusto para modelagem 3D de ambientes internos usando dados RGB-D. Basicamente, a modelagem 3D de ambientes está dividida em quatro tarefas, a saber: a escolha do sensor de imageamento; o problema do registro de nuvem de pontos 3D adquiridos pelo sensor de imageamento em diferentes pontos de vista; o problema da detecção de lugares anteriormente visitados (loop closure); e o problema da análise de consistência global. Atualmente, o Kinect é o sensor RGB-D mais empregado na aquisição de dados para modelagem de ambientes internos, uma vez que é leve, flexível e de fácil manuseio. A etapa de registro consiste em determinar os parâmetros de transformação relativa entre pares de nuvens de pontos e, neste trabalho, é dividida em duas partes: a primeira parte consiste em executar o registro inicial dos dados 3D usando pontos visuais e o modelo de corpo rígido 3D; na segunda parte, os parâmetros iniciais são refinados empregando um modelo matemático baseado numa abordagem paralaxe-a-plano, o que torna o método robusto. Para minimizar os efeitos da propagação de erros provocados na etapa de registro dos pares de nuvens de pontos 3D, o método proposto detecta lugares anteriormente visitados usando uma imagem de (frame-chave). Basicamente, é feita uma busca por imagens com grau de similaridade com a imagem de referência e, por fim, é obtida uma nova restrição espacial. A etapa de consistência global cria um grafo dirigido e ponderado, sendo cada vértice do grafo representado pelos parâmetros de transformação obtidos na etapa de registro dos dados, enquanto suas arestas representam as restrições espaciais definidas pelos parâmetros de transformação obtidos entre os lugares revisitados. A otimização deste grafo é feito através do método GraphSLAM. Experimentos foram realizados em cinco ambientes internos e o método proposto propiciou uma acurácia relativa em torno de 6,85 cm. .

Palavras-chave:sensor RGB-D; modelagem 3D; Otimização da trajetória baseado em grafos; registro de pares de nuvens de pontos; análise de consistência global.

Abstract

The objective of this paper is to propose a robust method for 3D modeling indoors using RGB-D data. Basically, the 3D modeling environment is divided into four problems, namely: the choice of the imaging sensor; the cloud Registration problem of 3D points acquired by the imaging sensor in different views; the problem of detection places previously visited (loop closure); and the problem of global consistency analysis. Currently, Kinect is the RGB-D sensor more employed in data acquisition for modeling indoor environments, since they are lightweight, flexible and easy to use. The registration step is to determine the transformation parameters relating between pairs of point cloud and in this paper is divided into two parts: the first part is to run the initial registration of 3D data using visual points and rigid body model 3D; in the second part, the initial parameters are refined using a mathematical model based on a parallax-the-plan approach, which makes the robust method. To minimize the effects of propagation of errors caused in the 3D point cloud pairs registration step, the proposed method detects previously visited places using a reference image (key-frame). Basically, a search for images with degree of correlation is made with the reference image, and finally, a new spatial constraint is obtained. The overall consistency of step creates a directed and weighted graph, each nodes in the graph represented by the transformation parameters obtained in the data registration step, whereas its edges represent the spatial constraints defined by the transformation parameters obtained between Revisited places. The optimization of the graph is made by GraphSLAM method. Experiments were carried out in five indoor and the proposed method provided a relative accuracy around 6,85 cm..

Keywords: RGB-D sensor; mapping 3D; GraphSLAM; pairs registration of point clouds; consistency global analysis.

Lista de Figuras

FIGURA 2.1 DISPOSITIVO KINECT A SER EMPREGADO NESTE TRABALHO E SEUS PRINCIPAIS COMPONENTES 29

FIGURA 2.2 GEOMETRIA DO MODELO PARA ESTIMAÇÃO DA PROFUNDIDADE 30

FIGURA 2.3 A) DERIVADA DE SEGUNDA ORDEM DA CONVOLUÇÃO GAUSSIANA L_{yy} E L_{xy}, B) CORRESPONDENTES APROXIMAÇÕES USANDO FILTROS CAIXA D_{yy} E D_{xy} 37

FIGURA 2.4 A) IMAGEM ORIGINAL, B) IMAGEM INTEGRAL CALCULADA DO PONTO $(0,0)$ ATÉ $(200,200)$ 38

FIGURA 2.5 A) ESPAÇO ESCALA TRADICIONAL, B) ESPAÇO ESCALA ADAPTADO PELO SURF 39

FIGURA 2.6 SUPRESSÃO NÃO MÁXIMA 40

FIGURA 2.7 ORIENTAÇÃO PREDOMINANTE 41

FIGURA 2.8 COMPONENTES DO DESCRITOR 42

FIGURA 2.9 GEOMETRIA EPIPOLAR 43

FIGURA 2.10 A) DETECÇÃO DE *INLIERS*, B)DETECÇÃO DE *OUTLIERS* ... 46

FIGURA 2.11 PONTOS DE INTERESSE OBTIDOS PELO SURF, A) PONTOS HOMÓLOGOS E OUTLIERS B) OUTLIERS REMOVIDAS COM O RANSAC E A MATRIZ FUNDAMENTAL 46

FIGURA 2.12 A) IMAGEM ORIGINAL, B) IMAGEM SEGMENTADA PELO SLIC 48

FIGURA 2.13 A) SLIC COM PARÂMETROS $K = 40$ E $m_k = 20$, B) SLIC COM PARÂMETROS $K = 100$ E $m_k = 10$, C) SLIC COM PARÂMETROS $K = 40$ E $m_k = 20$ 50

FIGURA 2.14 TRANSFORMAÇÃO DO CORPO-RÍGIDO, A)PONT0S CORRESPONDENTES COM DESALINHAMENTO ANGULAR E LINEAR ,B) EFEITO MINIMIZADO APÓS A TRANSFORMAÇÃO 51

FIGURA 2.15 TRAJETÓRIA ESTIMADA DO SENSOR 54

FIGURA 2.16 GRAFO CRIADO COM AS RESPECTIVAS RESTRIÇÕES 55

FIGURA 2.17 GRAFO DE POSE COM OS FECHAMENTOS DE LOOP 56

FIGURA 2.18 ASPÉCTO DA ARESTA QUE SE CONECTA AO VÉRTICE x_i AO x_j 57

FIGURA 2.19 TRAJETÓRIA DE UM ROBO MAPENDO O LABORATÓRIO INTEL, A) TRAJETÓRIA ESTIMADA, B) TRAJETÓRIA OTIMIZADA 63

FIGURA 3.1 PIPELINE DA METODOLOGIA PROPOSTA 66

FIGURA 3.2 ETAPAS DE IMPLEMENTAÇÃO DO SURF 68

FIGURA 3.3 A)CORRESPONDÊNCIA ENTRE OS SEGMENTOS NA IMAGEM DE REFERÊNCIA E NA NUVEM DE PONTOS 3D, B) CORRESPONDÊNCIA ENTRE OS CENTROIDES DA IMAGEM DE PESQUISA E O MAPA DE PARALAXE 75

FIGURA 3.4 REFINAMENTO DOS PARÂMETROS DE TRANSFORMAÇÃO A PARTIR DO MODELO PROPOSTO 78

FIGURA 3.5 JANELA DESLIZANTE PARA VERIFICAR ÁREAS REVISITADAS 79

FIGURA 3.6 JANELA DESLIZANTE PERCORRENDO O GRAFO 80

FIGURA 3.7 ESTRUTURA DO ARQUIVO DO FORMATO `graph` 81

FIGURA 3.8 FLUXOGRAMA DAS ETAPAS DE OTIMIZAÇÃO DO GRAFO .. 82

FIGURA 4.1 A) DISTORÇÃO RADIAL SIMÉTRICA E DESCENTRADA DA CÂMERA IR; B) DISTORÇÃO RADIAL SIMÉTRICA E DESCENTRADA DA CÂMERA RGB 86

FIGURA 4.2 EXPERIMENTO REALIZADO COM O INTERFERÔMETRO E COM O KINECT PARA A NORMALIZAÇÃO DOS VALORES DE PROFUNDIDADE. A) IMAGEM RGB; B) IMAGEM DE PROFUNDIDADE. . 87

FIGURA 4.3 RELAÇÃO LINEAR DA PARALAXE NORMALIZADA COM O INVERSO DA PROFUNDIDADE 88

FIGURA 4.4 RIGISTRO DE NUVENS, (A) DETECTANDO PONTOS HOMÓLOGOS ATRAVÉS DO SURF E RANSAC, (B)SEGMENTAÇÃO DA IMAGEM ATRAVÉS DO SLIC, (C) REGISTRO DO PAR DE NUVEM ATRAVÉS DO MODELO PROPOSTO 89

FIGURA 4.5 SUPERFÍCIE COM SUPERFÍCIE HOMOGÊNEA, (A) DETECTOR SURF E RANSAC, (B) SEGMENTÇÃO APLICANDO SLIC 90

FIGURA 4.6 TRAJETÓRIA DO DISPOSITIVO KINECT REPRESENTADO NA ESTRUTURA DE UM GRAFO, A) TRAJETÓRIA CENÁRIO A, B) TRAJETÓRIA CENÁRIO B, TRAJETÓRIA CENÁRIO C, TRAJETÓRIA CENÁRIO D 91

FIGURA 4.7 FUNÇÃO DE ERRO; A) CENÁRIO A, B) CENÁRIO B 92

FIGURA 4.8 VARIAÇÃO DE ESCALA ENTRE FRAMES-CHAVE E FRAMES-CANDIDATOS EM SITUAÇÃO DE FECHAMENTO DE LOOP (A) DETECÇÃO DOS INLIERS PELO SURF,(B) SEGMENTAÇÃO DA IMAGEM PELO SLIC 93

FIGURA 4.9 DETECÇÃO DE LOOP DETECTADO ENTRE O FRAME 15 E 65 DO CENÁRIO B 94

FIGURA 4.10 FUNÇÃO DE ERRO; A) CENÁRIO C, B) CENÁRIO D 95

FIGURA 4.11 CENÁRIO A RECONSTRUÍDO, A)NUVEM ORIGINAL, B) NUVEM OTIMIZADA 97

FIGURA 4.12 CENÁRIO a VISTA DE TOPO 97

FIGURA 4.13 GRAFO DE POSE CENÁRIO B, A) CENÁRIO NÃO OTIMIZADO, B) CENÁRIO OTIMIZADO 98

FIGURA 4.14 GRAFO DE POSE CENÁRIO C, A) CENÁRIO NÃO OTIMIZADO, B) CENÁRIO OTIMIZADO 99

FIGURA 4.15 RECONSTRUÇÃO TRIDIMENSIONAL DO CENÁRIO D 100

FIGURA 4.16 TRAJETÓRIA DO DISPOSITIVO KINECT REPRESENTADO NA ESTRUTURA DE UM GRAFO; A) TRAJETÓRIA CENÁRIO A; B) TRAJETÓRIA CENÁRIO B; C) TRAJETÓRIA CENÁRIO C; D) TRAJETÓRIA CENÁRIO D 101

FIGURA A.1 INTERFACE DO SISTEMA - MÓDULO I 116

FIGURA A.2 INTERFACE DO SISTEMA - MÓDULO II 117

FIGURA A.3 INTERFACE DO SISTEMA - MÓDULO III 117

FIGURA B.1 CENÁRIO CONSTRUÇÃO, A) RECONSTRUÇÃO 3D, B)GRAFO DE TRAJETÓRIA 118

FIGURA B.2 CENÁRIO QUARTO, A) RECONSTRUÇÃO 3D, B)GRAFO DE TRAJETÓRIA 119

FIGURA B.3 CENÁRIO LAPE, A) RECONSTRUÇÃO 3D, B)CENÁRIO 3D OTIMIZADO 120

Lista de Tabelas

TABELA 4.1 PARÂMETROS DE ORIENTAÇÃO INTERIOR DA CÂMERA IR E RGB DO KINECT 85

TABELA 4.2 PARÂMETROS DE DESLOCAMENTO E ROTAÇÃO DA CÂMERA IR EM RELAÇÃO A CÂMERA RGB DO KINECT 86

TABELA 4.3 ATRIBUTOS DO GRAFO NOS QUATRO CENÁRIOS 92

TABELA 4.4 FUNÇÃO ERRO $e(x_i, x_j)$ 94

TABELA 4.5 ERRO DA OTIMIZAÇÃO 96

Lista de Siglas

RGB-D	*Red,Green,Blue,Depth*
CMOS	*Complementary metal-oxide-semicondutor*
RGB	*Red,Green,Blue*
IR	*Infrared*
LASER	*Light Amplification by Stimulated Emission of Radiation*
fps	*Frame per seconds*
ISPRS	*International Society for Photogrammetry and Remote Sensing*
MVC	matriz variância-covariância
SIFT	*Scale Feature Transform*
SURF	*Speeded-Up Robust Feature*
ICP	Iterative Closest Point
SLAM	*Simultaneous localization and mapping*
TORO	*Tree-based netwORk Optimizer*
HOG-MAN	*Hierarchical Optimizations on Manifolds for Online 2D and 3D mapping*
G^2O	*General Framework Graph Optimization*
POI	Parâmetros de Orientação Interior
POE	Parâmetros de Orientação Exterior
MCD	matriz dos cossenos diretores
STP	*Speed-Up Turbo Pixels*
SLIC	*Simple linear iterative clustering*
CIELAB	*Commision Internationale L'Eclairage Lab*
OpenCV	*Open-Soure Computer Vision*
BSD	*Berkeley Software Distribution*
PCL	*Point Cloud Library*

MRTP	*Mobile Robot Programming Toolkit*
GPL	*General Public License*
LGPL	*Lesser General Public License*
VTK	*Visualização Tookit*
OpenGL	*Open graph Library*

Lista de Símbolos

p	ponto p na superfície
Z_0	plano de referencia
d	paralaxe (disparidade)
b	linha de base
f	distância focal
x',y'	foto-coordenadas do ponto imagem
x_0,y_0	coordenadas do ponto principal no referencial fotogramétrico
δx ,δx	coeficientes de distorção das lentes
X_k, Y_k, Z_k	Coordenadas 3D do ponto na superfície
f	distância focal
S_x e S_y	fatores de escala
$\Delta_\omega, \Delta_\varphi, \Delta_\kappa$	orientação relativa entre os sensores IR e RGB
$\Delta_X, \Delta_Y, \Delta_Z$	translação relativa entre os sensores IR e RGB
$H(x, \sigma)$	*Fast Hassian Detector*
L_{xx}	derivada segunda da convolução Gaussiana
L_{xy}	derivada segunda da convolução Gaussiana
L_{yy}	derivada segunda da convolução Gaussiana
I_Σ	imagem integral
$D_{xx}(x, \sigma)$	filtros da caixa correspondente $L_{xx}(x, \sigma)$
Γ_{subreg}	vetor dos descritores SURF
F	matriz fundamental
$f_{i,j}$	coeficientes da matriz fundamental
CP_1	centro perspectivo da imagem 1
CP_2	centro perspectivo da imagem 2

K	tamanho desejado de *superpixels*
$G(x,y)$	Gradiente da imagem
$I(x,y)$	vector de *lab* correspondente ao pixel na posição (x,y)
D_s	soma da distância Lab
xy	distância plana normalizada S
$F(Xa)$	funcional do método paramétrico
n	número de observações
u	número de parâmetros
X_{k_1}	coordenada X_k referente ao espaço S_1
Y_{k_1}	coordenada Y_k referente ao espaço S_1
Z_{k_1}	coordenada Z_k referente ao espaço S_1
X_{k_2}	coordenada X_k referente ao espaço S_2
Y_{k_2}	coordenada Y_k referente ao espaço S_2
Z_{k_2}	coordenada Z_k referente ao espaço S_2
ω	ângulo de rotação em relação ao eixo X
φ	ângulo de rotação em relação ao eixo Y
κ	ângulo de rotação em relação ao eixo Z
R	matriz de rotação 3D
L_b	vetor das observações
L_0	vetor das observações em função de $F(X_0)$
L	diferença entre as observações $L_0 - L_b$
X	vetor da correção dos parâmetros
X_0	vetor dos parâmetros aproximados
X_a	vetor dos parâmetros ajustados
A	matriz das derivadas parciais em relação aos parâmetros
P	matriz dos pesos
$\hat{\sigma}_0^2$	variância de unidade de peso à posteriori

$\mathbf{\Sigma}_{X_a}$	matriz variância-covariância dos parâmetros
$Pose_i$	matriz homogênea da pose do sensor
T_j^i	matriz homogênea de transformação entre i e j
$\prod$	produtório
x_i	vetor de estados i e j
$z_{i,j}$	vetor das observações
$\mathbf{\Omega}_{i,j}$	matriz informação
$\tilde{z}_{i,j}$	vetor de predição
$e_{i,j}$	vetor do erro em relação a aresta i e j
$F(x)$	modelo funcional do grafo
x^*	vetor de estados ajustado
$\breve{x}$	vetor da estimativa inicial
$\propto$	relação de equivalência
$\mathcal{C}$	conjunto de pares de índices
$\mathbf{A}_{ij}$	matriz das derivadas parciais de e_{ij} em relação ao vértice i
$\mathbf{B}_{ij}$	matriz das derivadas parciais de e_{ij} em relação ao vértice j
b_{ij}	vetor dos coeficientes
M_{ij}	auxiliar do jacobiano
H_{ij}	matriz Hessiana
J_{ij}	matriz Jacobiana
Δx	pequeno incremento
$\boxplus$	operador *manifold*
q	quatérnion
NL	numero de linhas
NC	numero de colunas
tp_x	tamanho do pixel em relação a x
tp_y	tamanho do pixel em relação a y

u	coordenada do pixel no referencial imagem em colunas
v	coordenada do pixel no referencial imagem em linhas
n_x, n_y, n_z	vetor normal do plano
ρ	distância do plano até o CP sensor
$F(X_a, La)$	modelo matemático do método combinado
W	vetor do erro de fechamento
B	matriz das derivadas parciais em relação as observações
tx_a	parâmetro de translação em relação a X ajustado
ty_a	parâmetro de translação em relação a Y ajustado
tz_a	parâmetro de translação em relação a Z ajustado
ω_a	parâmetro de rotação em relação a X ajustado
φ_a	parâmetro de rotação em relação a Y ajustado
κ_a	parâmetro de rotação em relação a Z ajustado
Xk_a	coordenada X da nuvem ajustada
Yk_a	coordenada Y da nuvem ajustada
Zk_a	coordenada Z da nuvem ajustada

Sumário

1 INTRODUÇÃO 18

1.1 CONSIDERAÇÕES INICIAIS 18

1.2 OBJETIVOS GERAIS 20

1.3 OBJETIVOS ESPECÍFICOS 20

1.4 JUSTIFICATIVA E CONTRIBUIÇÃO DO TRABALHO 20

1.5 ESTRUTURA DA TESE 21

2 REVISÃO DE LITERATURA 23

2.1 ESTADO DA ARTE 23

2.1.1 Extração das características visuais da imagem 23

2.1.2 Problema de registro de pares de nuvens 3D 24

2.1.3 Problema da análise da consistência global 26

2.2 SENSORES RGB-D 28

2.2.1 Geração da nuvem de pontos 3D 29

2.2.2 Calibração das câmeras IR e RGB do sensor RGB-D 32

2.2.3 Parâmetros de montagem do sistema 35

2.3 EXTRAÇÃO E DETECÇÃO DE PONTOS EM IMAGEM RGB 36

2.3.1 Detector de pontos SURF 37

2.3.2 Descritor de pontos SURF 40

2.4 ELIMINAÇÃO DE FALSOS POSITIVOS 42

2.4.1 Matriz Fundamental 42

2.4.2 O algoritmo RANSAC 45

2.5 SEGMENTADOR SLIC 47

2.6 REGISTRO DO PARES DE NUVENS 3D 51
2.7 CONSTRUÇÃO DO GRAFO 55
2.7.1 Detecção de lugares anteriormente visitados 55
2.7.2 Otimização do grafo 57

3 MATERIAIS E MÉTODOS 64
3.1 MATERIAIS 64
3.1.1 Recursos de hardware 65
3.1.2 Recursos de software 65
3.2 MÉTODO 66
3.2.1 Etapa 1: Calibração do sensor Kinect 67
3.2.2 Etapa 2: Extração e detecção dos pontos de interesse utilizando o SURF . . . 68
3.2.3 Etapa 3: Eliminação dos outliers 69
3.2.4 Etapa 4: Associação de pontos visuais na nuvem de pontos 3D 70
3.2.5 Etapa 5: Determinação dos parâmetros de transformação iniciais 72
3.2.6 Etapa 6: Refinamento dos parâmetros de transformação iniciais 73
3.2.7 Etapa 7: Construção do grafo 78
3.2.8 Etapa 8: Otimização do grafo 81
3.2.9 Etapa 9: Reconstrução do ambiente 3D 84

4 EXPERIMENTOS E ANÁLISE DOS RESULTADOS 85
4.1 Calibração da Câmera RGB e IR 85
4.2 Reconstrução dos Ambientes Internos 88

5 CONCLUSÕES E RECOMENDAÇÕES 102
5.1 CONCLUSÕES 102
5.2 RECOMENDAÇÕES PARA TRABALHOS FUTUROS 104

Referências Bibliográficas **106**

Apêndice A – Derivadas Parciais do Modelo Proposto **113**

Anexo A – Interface do Sistema **116**

Anexo B – Mais cenários **118**

1 INTRODUÇÃO

1.1 CONSIDERAÇÕES INICIAIS

Atualmente, há uma alta demanda por modelos tridimensionais (3D) de ambientes urbanos, principalmente, devido ao crescimento explosivo da população urbana. Algumas aplicações que exigem a modelagem 3D de interiores são: o estabelecimento de rotas de fugas em edifícios para emergências ou outros incidentes, planejamento de interiores, reconstrução de objetos 3D, navegação e posicionamento, técnicas de mapeamento e localização simultânea de veículos móveis, vigilância e aplicações forenses, mapeamento de corredores, salas, tuneis, cavernas e oficinas, medição de fachada interna de edificações, detecção de rachaduras em túneis metroviários e também em minas subterrâneas para extração de minerais em geral. Em outros casos é possível usar a modelagem 3D de um ambiente interno para *tours* virtuais que podem ser empregados por agências estaduais e federais de diferentes maneiras, bem como por profissionais da Arquitetura e de Engenharia, entre outros. Ainda, a *Google* lançou em 2011 um serviço que permite usuários potenciais fazerem o upload de suas plantas baixas para serem incluídas na ferramenta do *Google maps* tendo como objetivo mapear o interior de edifícios públicos no mundo todo. Este serviço tem sido amplamente utilizado como apoio a grandes empresas interessadas pela modelagem 3D de ambientes internos.

De modo geral, o meio mais adequado para visualizar e representar um ambiente interno é tridimensionalmente, uma vez que, para a maioria das pessoas e usuários de Cartografia, plantas baixas são difíceis de ser interpretadas. Para isto, é necessário investigar métodos para modelagem 3D de ambientes internos de forma rápida, segura e precisa. Basicamente, um método para modelagem 3D de ambientes internos é composto por quatro partes principais: a escolha e a calibração do dispositivo de imageamento; o registro de pares de nuvens de pontos 3D; a detecção de lugares revisitados; e a análise de consistência global. Neste trabalho é empregado o dispositivo *Kinect* para aquisição dos dados RGB-D. Este dispositivo foi desenvolvido pela *Microsoft*, juntamente com a PrimeSense, para fins de entretenimento e jogos computacionais da *Microsoft Xbox* 360. O

Kinect é composto por três sensores: dois sensores CMOS (*Complementary metal-oxide-semicondutor*) que registram energia eletromagnética na faixa do espectro correspondente ao visível (câmera RGB) e infravermelho (câmera IR); e um emissor LASER (*Light Amplification by Stimulated Emission of Radiation*) infravermelho. Os sensores dispostos no *Kinect* podem capturar cenas com $640x480$ *pixels* em uma taxa de 30 *frames* por segundo *(fps)* fps, podendo cada *frame* conter até 300.000 pontos. O *Kinect* é mais barato que as câmeras de distância 3D e os sistemas LASER *scanner*, cujo valor de venda gira em torno de R $350,00, além de ser mais leve, mais flexível e de fácil manuseio.

A calibração dos sensores IR e RGB, e a determinação dos parâmetros de montagem do sistema (orientação relativa e o deslocamento entre as origens dos sistemas) é fator preponderante para a correção dos efeitos sistemáticos causados pelas distorções das lentes e para o emprego da sinergia entre as informações RGB e de profundidade. Khoshelham e Elberink (2012) e Menna et al. (2011) mostraram o potencial da resolução geométrica e da precisão dos pontos tridimensionais propiciados pelo dispositivo, enquanto Henry et al. (2012) demonstraram o seu potencial para modelagem 3D de ambientes internos. A etapa de registro dos pares de nuvens de pontos, neste trabalho, é dividida em duas partes: a primeira parte consiste em determinar os parâmetros de transformação (3 rotações e 3 translações) iniciais usando pontos visuais 2D associados com pontos 3D nas nuvens de pontos e o modelo de corpo rígido 3D; na segunda parte, os parâmetros de transformação são refinados empregando um modelo matemático baseado numa abordagem *paralaxe-a-plano*, tornando o método robusto.

Para minimizar os efeitos da propagação de erros provocados na etapa de registro dos pares de nuvens de pontos 3D, a trajetória do sensor é abstraída sob a estrutura de um grafo dirigido, onde as arestas representam a posição e orientação do dispositivo Kinect, e suas arestas representam as restrições espaciais entre pares de vértices. Estas restrições são determinadas pela revisita de lugares já imageados, denominado fechamento de *loop*. Estes fechamentos são identificados a partir da características pontuais corretas (*inliers*) detectadas pelo descritor e detector SURF. Para encontrar a configuração de vértices que minimizem o erro introduzido pelas arestas, é realizado a otimização do grafo aplicando o MMQ. A partir dos vértices do grafo otimizados é realizado a reconstrução do ambiente imageado.

1.2 OBJETIVOS GERAIS

O objetivo geral deste trabalho é investigar e desenvolver um método robusto para modelagem 3D de ambientes internos usando dados RGB-D.

1.3 OBJETIVOS ESPECÍFICOS

Para alcançar o objetivo geral deste trabalho é necessário executar as seguintes tarefas, a saber:

- Realizar a calibração do sensor RGB e IR do Kinect, com o intuito de determinar os parâmetros de orientação interior, bem como determinar os parâmetros de montagem do sistema (orientação relativa e o deslocamento entre as origens dos sistemas);
- Desenvolver um método robusto para o registro de pares de nuvens de pontos baseado numa abordagem de registro em duas etapas;
- Estudar e desenvolver um modelo matemático baseado numa abordagem paralaxe-a-plano para refinar os parâmetros de transformação entre pares de nuvens;
- Resolver o problema de consistência global abstraindo a trajetória do sensor na estrutura de um grafo dirigido, onde as arestas do grafo representam aposição e orientação (pose) do sensor e as arestas restrições espacias;
- Analisar e discutir os resultados obtidos.

1.4 JUSTIFICATIVA E CONTRIBUIÇÃO DO TRABALHO

Como descrito anteriormente, a modelagem 3D de ambientes internos é uma poderosa ferramenta para diversas aplicações em Engenharia e áreas afins, tais como, para tomada de decisões inteligentes e econômicas, para construção de ambientes virtuais, modelagem de túneis metroviários e de minas subterrâneas. A modelagem 3D de interiores contribui significativamente para uma análise rápida, econômica e sem o contato direto com os objetos de interesse.

O emprego do dispositivo *Kinect* para aquisição das informações geoespaciais se justifica por: apesar de fornecer informações com precisão em torno de $3cm$ a cada $3m$ é um dispositivo flexível e de baixo custo, além de propiciar uma sinergia natural entre os

valores RGB e de profundidade (D), sendo bastante útil em aplicações dinâmicas. Outro fator que justifica o tema de investigação neste trabalho é que a maioria dos métodos existentes para modelagem 3D de ambientes internos não são robustos, uma vez que são baseados em abordagens *ponto-a-ponto* entre a imagem de referência e a imagem de pesquisa, e não empregam uma etapa de refinamento do registro dos dados. A etapa de refinamento é essencial, tendo em vista que: feições pontuais não são facilmente encontradas em ambientes internos com pouca textura; a existência de ruídos e *outliers* é muito comum em grandes volumes de dados; a natureza discreta e aleatória na aquisição dos pontos pelo dispositivo torna o processo de detecção e coleta de pontos de baixa confiabilidade e precisão; e como a densidade dos pontos e o espaçamento são afetados por objetos especulares e de alta absorção de energia eletromagnética no IR, o processo de estabelecimento de correspondências torna-se bastante complexo. Outras razões que justificam o desenvolvimento deste projeto são: a demanda por métodos de modelagem 3D em ambientes internos automáticos e precisos; o desenvolvimento e aquisição do conhecimento da nova tecnologia; e a relevância científica comprovada, devido à importância dispensada pela ISPRS *Commision V Close-Range Sensing: Analysis and Applications*, *WG V-3 Terrestrial laser scanning and 3D imaging* aos objetos de estudo deste trabalho.

Este trabalho tem como principal contribuição o desenvolvimento de uma abordagem para o refinamento dos parâmetros de transformação relativa entre pares de nuvens de pontos baseada em um modelo paralaxe-a-plano. O modelo matemático proposto com esta característica ainda não encontrado na literatura, assim como o emprego da MVC dos parâmetros de transformação na ponderação dos grafos dirigidos, para a análise de consistência global.

1.5 ESTRUTURA DA TESE

Prezando pelo entendimento do leitor no desenvolvimento desta tese, nesta seção será apresentado uma sequência resumida e lógica dos estudos realizados. Sendo assim, o trabalho esta estruturado da seguinte forma:

- **Capitulo 1**: foi apresentado a introdução, o objetivo geral e os objetivos específicos, a justificativa e a sua contribuição;
- **Capitulo 2**: este capítulo esta dividido em duas partes, o estado da arte e a fundamentação teórica.
  - estado da arte: abordará os estudos realizados na extração de características

de feições para o registro de imagens em um contexto geral, o registro entre nuvens de pontos, e o problema de consistência global;

  - fundamentação teórica: abordará o funcionamento e a calibração dos dispositivos RGB-D em especial do Kinect, o detector e descritor SURF, a detecção e eliminação de pontos ruidosos aplicando a matriz fundamenta e o algorítimo RANSAC, o algoritmo SLIC para segmentação da imagem RGB, a determinação dos parâmetros grosseiros por meio da transformação do corpo rígido aplicando o MMQ, e a construção e otimização da trajetória do sensor aplicando grafos;

- **capitulo 3**: irá apresentar os recursos de *hardware* e *software* para o desenvolvimento do trabalho ea metodologia proposta;
- **capítulo 4**: os experimentos e os resultados alcançados;
- **capítulo 5**: a conclusão e recomendação para trabalhos futuros;

2 REVISÃO DE LITERATURA

2.1 ESTADO DA ARTE

2.1.1 Extração das características visuais da imagem

A extração das características visuais entre pares de imagens, também conhecido como correspondência ou *matching* é uma tarefa fundamental para o registro de imagens, e também para detecção de lugares revisitados (fechamento de loop). Conforme se adquire uma sequencia de imagens, a correspondência entre os pares de imagens vai se tornando uma tarefa complexa, devido a variações de escala, rotação e iluminação, aumentado a dificuldade no desenvolvimento de uma solução robusta. Devido a sua complexidade há na literatura de Fotogrametria, Sensoriamento Remoto e de Visão Computacional uma vasta quantidade de métodos que tratam do problema de correspondência em imagens. Primeiramente, foi criado um grupo de métodos de correspondência baseado em área (CIDECIYAN et al., 1992); (BROWN, 1992); (KHER; MITRA, 1993); (ZHENG; CHELLAPPA, 1993); (BELL; DEVARAJAN; APOLLO, 1999) e (BUNTING; LABROSSE; LUCAS, 2010).Apesar de apresentar relativa precisão, os algoritmos são bastante lentos e os pares de imagens devem ser geométrica e radiometricamente similares. Esses algoritmos também não tratavam o problema de variação de escala e rotação com eficiência, obtendo-se muitas ambiguidades, uma vez que múltiplas correspondências podem ser estabelecidas em uma mesma região da imagem, principalmente em áreas homogêneas. Para contornar estas deficiências foram desenvolvidos métodos baseados em correspondência estrutural ou por feições para o registro de pares de imagens. Basicamente, os algoritmos baseados nesses métodos, buscam encontrar feições correspondentes nas imagens que sejam visualmente de interesse, tais como, bordas, regiões, linhas, cantos e curvaturas (GOSHTASBY, 1986), (LI; MANJUNATH; MITRA, 1995), (HABIB; ALRUZOUQ, 2004), (HABIB; KIM; KIM, 2005), (YASEIN; AGATHOKLIS, 2008). De modo geral, este tipo de abordagem apresenta algumas deficiências, principalmente em regiões com pouca textura. (ACHANTA et al., 2012)

Nos anos de 1980 surgiram grupos de algoritmos para detectar pontos de interesses em uma imagem, em que a posição deste ponto pode ser definida de maneira robusta. Isto é realizado através da detecção da intersecção entre duas bordas, ou seja, um ponto é definido para qual existem duas direções diferentes de bordas, e que sejam dominantes em diferentes locais de uma vizinhança do ponto. Para este grupo de algoritmos foram denominados de detectores de cantos (FÖRSTNER; GÜLCH, 1987), (HARRIS; STEPHENS, 1988), (SMITH; BRADY, 1997). A determinação da qualidade dos detectores de cantos é a sua capacidade de detectar o mesmo canto em várias imagens semelhantes sob condições diferentes de iluminação, rotação e escalas o que torna este grupos de algoritmos pouco robusto. Lowe (2004) propôs um algoritmo chamado SIFT (*Scale Feature Transform*) para detecção e extração de pontos de interesse chaves em imagens digitais. Basicamente, o algoritmo é dividido em um detector e um descritor. Ele é invariante a escala, a rotação, a iluminação e a mudança de ponto de vista. Na mesma direção, Bay, Tuytelaars e Gool (2006) desenvolveram o algoritmo SURF (*Speeded-Up Robust Feature*), cujo objetivo é produzir um descritor de pontos visuais que permitam avaliações rápidas e altamente discriminatórias com outros pontos de interesse.

2.1.2 Problema de registro de pares de nuvens 3D

Como os sensores RGB-D fornecem informação de profundidade para cada pixel da imagem, permitindo assim, de certa forma o cálculo das suas coordenadas tridimensionais. Para cada par de nuvens de pontos é necessário determinar os parâmetros de transformação (rotação e translação) entre elas, porém é essencial que o erro de transformação entre os pares de nuvens de pontos seja minimizado. A tarefa que consiste em minimizar esses erros é conhecida como registro de pares de nuvens de pontos tridimensionais (3D). O registro de pares de nuvem de pontos 3D é objeto de intensa investigação científica e pode ser dividido, basicamente, no problema de correspondência e de otimização. O problema de correspondência tem por finalidade encontrar as primitivas na nuvem de pontos de referência com maior similaridade existente na nuvem de pontos de pesquisa. Já o problema de otimização consiste em definir uma transformação que minimiza a soma dos quadrados das distâncias entre dois conjuntos de dados do sensor RGB-D. A hipótese conhecida é que se deve alinhar perfeitamente a nuvem de pontos de pesquisa em relação à nuvem de pontos de referência. Desta forma, a transformação escolhida deve permitir a determinação de parâmetros de transformação (parâmetros de rotações e translações) entre os pares de nuvens de pontos 3D. Vale ressaltar que diversas abordagens propostas foram baseadas no emprego da transformação de corpo rígido 3D. O método mais difun-

dido na literatura é conhecido como ICP (*Iterative Closest Point*) e foi desenvolvido por Besl e McKay (1992). Basicamente, o ICP encontra pares de pontos pseudo conjugados e determina, de forma iterativa, os parâmetros de transformação relativa entre pares de nuvens de pontos. O ICP é de alto custo computacional e requer valores iniciais com alto grau de aproximação, além de alta taxa de sobreposição entre as nuvens de pontos 3D. Vale ressaltar que ao longo dos anos surgiram diversas variações do ICP, cujas abordagens são baseadas em modelos ponto-a-plano e plano-a-plano (CHEN; MEDIONI, 1992), (ZHANG, 1994), (EGGERT; FITZGIBBON; FISHER, 1998), (OKATANI; DEGUCHI, 2000), (PARK; SUBBARAO, 2003), (SEGAL; HAEHNEL; THRUN, 2009). Chen e Medioni (1992) investigaram um algoritmo de registro de pares de nuvem de pontos 3D com base na minimização da soma dos quadrados das distâncias entre pontos e suas superfícies correspondentes. As correspondências ponto-a-plano podem levar a um modelo de estimativa mais robusta que converge rapidamente. Este algoritmo procura o ponto de intersecção entre a superfície de referência e o vetor normal do ponto de origem projetado ortogonalmente. Para encontrar o ponto de origem, os autores usaram uma técnica de pesquisa baseada no método de Newton-Raphson em um sistema de coordenadas ortogonal. Vale ressaltar que este método também não converge sem valores iniciais aproximados.

A partir do século XXI, a sinergia entre imagens ópticas e nuvens de pontos 3D passou a ser explorada, sendo os trabalhos pioneiros desenvolvidos por Bendels et al. (2004), Dold e Brenner (2006), Al-Manasir e Fraser (2006), Barnea e Filin (2007), Ellekilde et al. (2007), Stamos et al. (2008), Prusak et al. (2008), Huhle, Jenke e Straßer (2008) e Kang et al. (2009). Na época, os autores exploraram a integração de sensores ópticos e LASER *scanner*. No entanto, câmeras digitais e equipamentos LASER *scanner* são de alto custo, pesados, de difícil manuseio e de diferentes resoluções geométricas.

A partir do desenvolvimento dos sensores RGB-D Henry et al. (2010), desenvolveram um método para modelagem 3D de ambientes internos empregando a sinergia natural dos dados RGB e de profundidade propiciados pelo dispositivo Kinect. Neste método os autores propuseram nova variação do método ICP, conhecida como RGB-D/ICP. A abordagem tira proveito das informações derivadas das imagens RGB e de profundidade. Um aspecto não mencionado no trabalho dos autores supracitados é a dificuldade em implementar um algoritmo capaz de solucionar o problema de registro dos dados em regiões com superfícies homogêneas e com padrões repetidos e/ou superfície planas. A alta taxa de aquisição de dados, o baixo custo e fácil portabilidade dos sensores RGB-D motivou pesquisas para mapeamento e localização simultâneo de robôs e drones (SLAM) em ambientes internos e em tempo real, dependendo apenas das informações do próprio sensor

(STEINBRÜCKER; STURM; CREMERS, 2011), (DU et al., 2011), (ENGELHARD et al., 2011), (BACHRACH et al., 2012) e (ENDRES et al., 2012). No entanto, o curto alcance, limitado campo de visão e erros inerentes do próprio sensor provocam a construção de modelos inconsistentes. A otimização da trajetória do sensor recai no problema de consistência global, cujo estado da arte será discutido a seguir.

2.1.3 Problema da análise da consistência global

Erros inerentes ao dispositivo RGB-D, como também os procedimentos de registro por pares de nuvens de pontos conduzem a erros que vão se acumulando ao longo da trajetória do sensor resultando em modelagem 3D inconsistentes. Para evitar este problema se faz necessário a aplicação de algoritmos de registro global ou de consistência global, no qual leva em conta toda a área já varrida. Os métodos mais comuns para mesclar todas as varreduras são baseados em informações probabilísticas. Na literatura do SLAM (*Simultaneous localization and mapping*) existe uma grande variedade de abordagens para solução do problema de consistência global. As técnicas mais conhecidas são baseadas em filtragem Kalman, que recursivamente estima uma densidade gaussiana da postura do sensor (SMITH; SELF; CHEESEMAN, 1990), (CASTELLANOS et al., 1999) e os filtros de partículas (MONTEMERLO et al., 2002), (HAHNEL et al., 2003), (GRISETTI et al., 2007),), (EUSTICE; SINGH; LEONARD, 2005), (THRUN et al., 2004). A principal desvantagem destas técnicas é o alto custo computacional, uma vez que mantém várias hipóteses de variáveis de estado, que podem se propagar mais rápido que o deslocamento do sensor, outra desvantagem é que não existe a possibilidade de recuperar as informações já processadas. Também são encontradas na literatura abordagens por suavização. Este método estima a trajetória completa do sensor a partir de todo o conjunto de medidas (LU; MILIOS, 1997), (DELLAERT; KAESS, 2006) e (OLSON; LEONARD; TELLER, 2006), na Fotogrametria o método por suavização é conhecido como *Bundle Adjustment*.

Com o avanço da álgebra computacional e do desenvolvimento de algoritmos para a solução de matrizes esparsas o problema de consistência global passou a ser estudado baseado na teoria dos grafos. Neste caso, o processo estocástico é abordado a partir de um grafo dirigido e ponderado. Esta questão envolve a construção de um grafo, cujos vértices representam as posições do sensor (dado pelos parâmetros de transformação) que descrevem sua trajetória, enquanto as arestas, que ligam dois vértices, representam uma medida do sensor que restringe as posições conectadas (posição relativa de um vértice a outro), as arestas são inerente as incertezas, correspondendo a limitação espacial entre

eles. Obviamente, tais restrições podem ser inconsistentes, já que as observações são sempre afetadas por ruídos gaussianos. Uma vez que um grafo é construído, o gargalo do problema é encontrar uma configuração para os vértices que seja consistente com as medidas realizadas, ou seja, encontrar uma configuração de vértices que minimize o erro induzido pelas arestas. Basicamente, a construção de um grafo de posições relativas é dividida em dois blocos o *front-end* e o *back-end*.

O bloco*front-end* é fortemente dependente dos dados do sensor pois, consiste em calcular as primeiras aproximações da sua trajetória, bem como a detecção de lugares anteriormente visitados (fechamento de *loop*), que através de um histórico de imagens verifica os lugares já varridos. Este bloco representa, portanto, a estimação da sequência de parâmetros de transformação do sensor e estabelece as relações entre eles. Esta tarefa é realizada mediante o registro consecutivo de pares de nuvens de pontos. O fornt end aplicado a sensores RGB-D realize as seguintes funções:

- Extração e detecção dos pontos de interesse na imagem RGB;
- Associação dos pontos de interesse 2D com seus correspondentes na nuvem de pontos 3D e o cálculo aproximado dos parâmetros de transformação através das transformação rígida entre pares de nuvens;
- Refinamento dos parâmetros iniciais através de métodos iterativos;
- Detecção de lugares anteriormente visitados.

O bloco *back-end* consiste em ajustar os vértices do grafo a partir das restrições impostas pelas arestas. Portanto, a a precisão e a eficiência do back-end é crucial para o projeto de um bom sistema. Mais recentemente diversos otimizadores baseados em grafos vêm sendo apresentados para esta tarefa, tais como o TORO (*Tree-based netwORk Optimizer*) aplicando uma parametrização em árvore que decompõe o problema em vários sub-problemas por meio de arvore geradora minima, cada sub-problema é resolvido aplicando o gradiente descendeste forma independente, as soluções são integradas através da taxa de aprendizagem (GRISETTI et al., 2009); o HOG-MAN (*Hierarchical Optimizations on Manifolds for Online 2D and 3D mapping*) que é um otimizador hierárquico que divide o problema em diferentes níveis de resolução, ou seja, durante o funcionamento on-line, a abordagem corrige apenas a estrutura grosseira da cena e não o modelo global, atualizando as partes do modelo que precisam ser considerados para estabelecer a associações entre os dados (GRISETTI et al., 2010a), e o G^2O(*General Framework Graph*

Optimization) cujo método propicia a redefinição precisa da função de erro (KÜMMERLE et al., 2011), baseado no método GraphSLAM (THRUN; MONTEMERLO, 2006).

2.2 SENSORES RGB-D

Alguns sensores de imageamento são capazes de propiciar imagens 3D medindo a distância do sensor aos objetos, ao invés da intensidade refletida ou difusa da luz sobre a superfície. A saída fornecida por estes sensores é conhecida como imagens de distância ou de profundidade. Neste caso, o valor de cada pixel representa a distância entre um ponto de referência e um ponto da superfície imageada. Alguns destes sensores fornecem dados RGB-D, uma vez que as componentes RGB associadas a cada pixel na imagem são registradas, juntamente com a distância. Existem dois tipos de sensores para aquisição de dados 3D: os sensores passivos; e os sensores ativos. Os sensores passivos dependem de fonte de luz externa para registrar as informações eletromagnéticas refletidas pela superfície física. Tais sistemas são baseados em câmeras monoculares (NEIRA et al., 1997) e estéreo-câmeras (HARTLEY; ZISSERMAN, 2003). Outra forma de aquisição de dados 3D é através de sistemas de padrão de luz estruturada (MAAS, 1993), (REISS; TOMMASELLI, 2011) entre outros. Basicamente, o sistema projeta um padrão de luz estruturada na forma de grade ou linhas sobre a superfície, baseado na projeção de um feixe LASER ou um quadro padrão, sendo refletido pela superfície e registrado por uma câmera digital. Usualmente, um algoritmo de correlação de imagens é utilizado para encontrar os padrões correspondentes e, por fim, calculadas as informações desejadas.

No caso dos sensores ativos a energia eletromagnética é gerada e emitida para interagir com a superfície. Desta forma, é possível determinar o tempo de emissão e recepção do pulso de energia emitido e, juntamente com a velocidade da luz (c), calcular a distância entre o sensor e a superfície imageada. Esta categoria de sensores ativos é conhecida como sensores LiDAR (*Light Detection and Ranging*).

Como descrito anteriormente, a Microsoft, juntamente com a PrimeSense, desenvolveu o dispositivo Kinect para fins de entretenimento e jogos computacionais da Microsoft Xbox 360. Este dispositivo é composto por: dois sensores CMOS que registram energia eletromagnética na faixa do espectro correspondente ao visível e ao infravermelho, e um emissor LASER infravermelho. A Figura 2.1 mostra o Kinect e seus principais componentes.

A câmera RGB é um sensor de quadro com resolução de 640x480 pixels que re-

FIGURA 2.1 – DISPOSITIVO KINECT A SER EMPREGADO NESTE TRABALHO E SEUS PRINCIPAIS COMPONENTES
FONTE: Burrus (2011)

gistra três canais de 8 bits, vermelho, verde e azul (RGB) e também é composta por um Filtro de *Bayer* que coloca os valores RGB em um padrão de grade de cores alternadas. O sistema de sensores infravermelho é composto por um emissor LASER de diodo 830 nm (infravermelho próximo) que emite um pulso que se espalha como padrão de luz estruturada. A câmara de IR contém um filtro óptico com uma sensibilidade mínima para comprimentos de onda que diferem do diodo LASER, resultando num padrão nítido da projeção sobre a imagem IR. Sua distância operacional varia entre $0,7$e $6,0$ metros (detalhes sobre acurácia dos valores de profundidade ver Khoshelham e Elberink (2012)). O sensor gera valores de disparidade de 11 *bits* para cada pixel, que finalmente são mapeados na imagem de profundidade.

2.2.1 Geração da nuvem de pontos 3D

De acordo com Freedman et al. (2012) o princípio de funcionamento do dispositivo Kinect para determinação dos valores de profundidade é conhecido como triangulação 3D. Basicamente, o sensor LASER emite um pulso na região do espectro correspondente ao infravermelho e um padrão de luz estruturada é criado a partir de um processo de difração da luz. O padrão de luz estruturada é usado para marcar a distância entre o sensor e a superfície física (Z_0) que é, previamente, armazenada na memória do dispositivo. Em outras palavras, é criado um plano de referência. Um p qualquer do padrão de luz estruturada, contido no plano de referência, é registrado na imagem IR com sua posição deslocada (ver o ponto k na Figura 2.2). Para resolver o problema de correspondência, cada pixel na imagem IR é comparado com os pixels do padrão de luz estruturada (armazenado anteriormente) através de uma convolução em torno da sua vizinhança, cujo tamanho da janela de correlação pode variar de $7x7$, $9x9$ ou $9x7$. O pixel na imagem IR apresentando a maior correspondência dentro da janela de correlação é selecionado como o pixel correspondente. O deslocamento entre o pixel de referência do padrão de

luz estruturada e seu correspondente na imagem IR indica o valor de paralaxe (d). Desta forma, o valor de profundidade de cada pixel na imagem IR pode ser obtido em função de d, da distância focal do sensor IR, da linha de base (b) e de Z_0. A Figura 2.2 mostra a geometria do modelo matemático proposto por Khoshelham e Elberink (2012) para calcular as coordenadas tridimensionais de cada ponto na imagem IR.

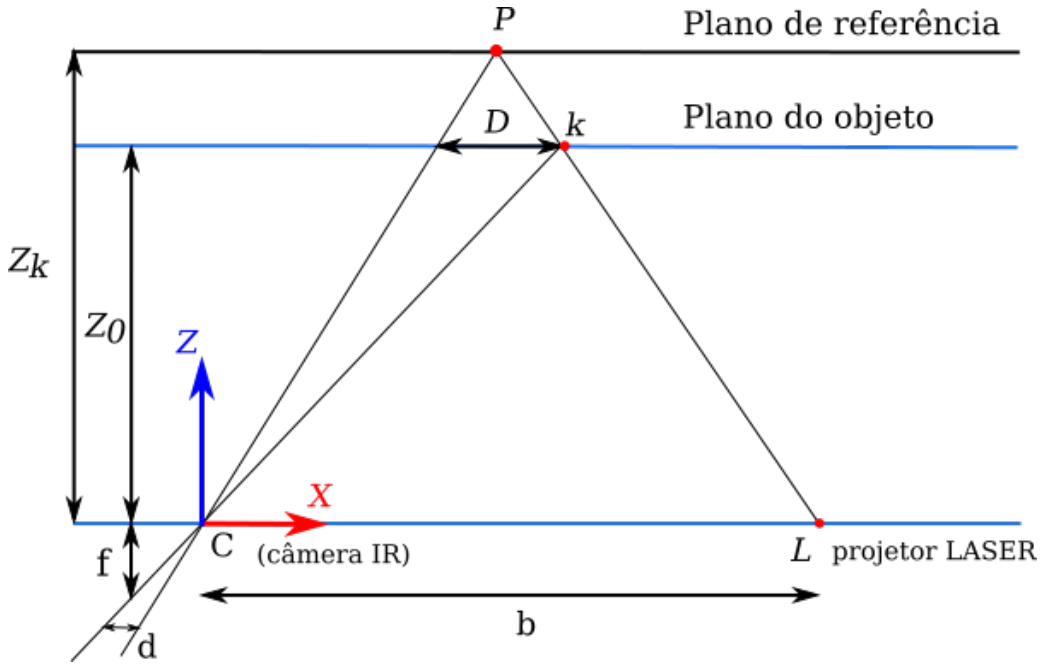


FIGURA 2.2 – GEOMETRIA DO MODELO PARA ESTIMAÇÃO DA PROFUNDIDADE
FONTE: Khoshelham e Elberink (2012)

A Figura 2.2 apresenta a geometria do modelo matemático, proposto por Khoshelham e Elberink (2012) para calcular os valores de profundidade Z_k e as coordenadas X_k e Y_k de cada pixel na imagem IR. Na Figura 2.2 percebe-se a relação entre a distância de um ponto P ao sensor, relativo ao plano de referência do padrão de luz estruturada e a medida de paralaxe. O sistema referencial do sensor IR é um sistema tridimensional com origem no centro perspectivo CP da câmera IR. O eixo Z é ortogonal ao plano da imagem apontando para a superfície física, enquanto o eixo X é perpendicular ao eixo Z na direção da linha de base determinada entre o sensor IR e o emissor LASER. O eixo Y é ortogonal aos eixos X e Z tornando o sistema de referencial dextrogiro. Por semelhança de triângulos, tem-se:

$$\frac{d}{f} = \frac{Z_0 - Z_k}{Z_0} \tag{2.1}$$

$$\frac{d}{f} = \frac{D}{Z} \tag{2.2}$$

Nas Equação 2.1 e 2.2, Z denota a distância (profundidade) do ponto no espaço objeto, f a distância focal da câmera *IR*, D é o deslocamento de no espaço objeto, e d é a paralaxe ou disparidade observada no espaço imagem. Substituindo D da Equação 2.2 na 2.1, tem-se:

$$Z_0 = \frac{Z_0}{1 + \frac{Z_0}{fb}d} \tag{2.3}$$

Vale ressaltar que o sensor não retorna a imagem de paralaxe bruta. Em vez disso, ele fornece os valores normalizados, entre 0 e 2047 ($11 - bits$). A Equação 2.4 define a relação entre a paralaxe bruta e seu valor normalizado (d').

$$d = md' + n \tag{2.4}$$

Sendo, m e n os fatores de normatização. Substituindo a Equação 2.4 na 2.3 e fazendo as devidas manipulações matemáticas, tem-se:

$$Z_k^{-1} = \left(\frac{m}{fb}\right)d' + \left(Z_0^{-1} + \frac{n}{fb}\right) \tag{2.5}$$

Desde que os parâmetros Z_0, b e f são devidamente obtidos por um processo de auto-calibração, a Equação 2.5 pode ser usada para calcular os valores de profundidade. A coordenada do ponto, juntamente, com a distância focal determina a escala para o determinado ponto. Sendo assim, as coordenadas planimétricas podem ser obtidas por meio das equações:

$$X_k = \frac{Z_k}{f}(x' - x_0 + \delta x) \tag{2.6}$$

$$Y_k = \frac{Z_k}{f}(y' - y_0 + \delta y) \tag{2.7}$$

Onde:

x',y': foto-coordenadas do ponto imagem (observadas no plano focal da câmera IR);

x_0,y_0: coordenadas do ponto principal no referencial fotogramétrico;
δx ,δx: coeficientes de distorção das lentes;

Finalmente, tem-se como resultado uma nuvem de pontos tridimensional, cujos valores X_k, Y_k, Z_k são atribuídos a cada ponto na superfície física imageada.

2.2.2 Calibração das câmeras IR e RGB do sensor RGB-D

A calibração de câmeras é um procedimento fundamental nos levantamentos fotogramétricos, segundo Eisenhart (1963) calibrar é uma maneira refinada de realizar medidas. Pois é neste procedimento que permite estimar os parâmetros que definem a projeção de um ponto tridimensional no espaço-objeto com o seu correspondente homólogo no espaço-imagem. Os referidos parâmetros incluem a geometria interna e óptica da câmara denominados de POI composto pela distância focal, deslocamento do ponto principal e distorções de lentes. E através da calibração também são determinados os parâmetros de posicionamento e orientação da câmera no espaço tridimensional denominados POE. A utilização de câmeras devidamente calibradas possibilita a redução de erros sistemáticos relacionados com a obtenção da posição e orientação tridimensional ou com a reconstrução 3D de objetos a partir de imagens. Portanto, para uma maior sinergia entre as informações obtidas pelo sensor IR e RGB do Kinect é necessário a determinação dos POI de ambos os sensores, e os parâmetros de montagem entre os dois sensores através dos POE.

Em conjunto, os POI e POE definem as condições de formação de uma imagem; ou seja, a transformação associada à projeção de pontos 3D do espaço imagem, de coordenadas (X,Y,Z), em pontos 2D da imagem, de coordenadas (x',y'), conforme mostra a Equação 2.8 (ZHANG, 1999):

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} fS_x & \tau & x_0 \\ 0 & fS_y & y_0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \end{bmatrix} \cdot \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \tag{2.8}$$

Onde:
x' e y': as coordenadas do ponto p' no espaço-imagem;
X, Y e Z: as coordenadas do ponto correspondente no espaço-objeto;
x_0 e y_0:as coordenadas do ponto principal;
τ: o fator de não-ortogonalidade entre os eixos da câmera;
f: a distância focal da câmera;

S_x e S_y: os fatores de escala em X e Y;
r_{ij}: corresponde aos elementos da matriz de rotação em função dos ângulos de Euler(ω, φ, κ) com $i = j = 1, ..., 3$;
t_x, t_y e t_z são os parâmetros de translação e (.) é o operador de multiplicação.

Ainda é necessário considerar os coeficientes de distorção do sistema de lentes da câmera (distorção radial e descentrada), como segue (BROWN, 1992):

$$\delta_x = \bar{x}(k_1 r^2 + k_2 r^4 + k_3 r^6) + P_1(r^2 + 2\bar{x}^2) + 2P_2 \bar{x}\bar{y}^2 \tag{2.9}$$

$$\delta_y = \bar{y}(k_1 r^2 + k_2 r^4 + k_3 r^6) + P_2(r^2 + 2\bar{y}^2) + 2P_1 \bar{x}\bar{y}^2 \tag{2.10}$$

Sendo $\bar{x}$ e $\bar{y}$ as coordenadas do ponto no espaço-imagem corrigidas das distorções radiais simétricas e descentradas, dado por $\bar{x} = x^{'} - x_0$ e $\bar{y} = y^{'} - y_0$; $r = \sqrt{(\bar{x}^2 + \bar{y}^2}$; os coeficientes de distorção radial das lentes k_1,k_2...k_n e os termos da distorção descentrada P_1 e P_2.

A relação entre os pontos no espaço-imagem e seus correspondentes no espaço-objeto é obtida usando um plano padrão. Usualmente, um tabuleiro de xadrez é empregado como campo de calibração e, por este motivo, a coordenada Z é inserida como injunção com valor igual a *zero*. A injunção imposta define uma transformação projetiva entre o campo de calibração e o plano da imagem. Desta forma, o ponto $\bar{p} = (\bar{x}, \bar{y})$ no espaço-imagem e seu correspondente $P = (X, Y, 1)$ no espaço-objeto são representados por coordenadas homogêneas, como segue (ZHANG, 1999):

$$s.p = H.P \tag{2.11}$$

Conforme Hartley e Zisserman (2003) a Equação 2.11 é dado por s um fator de escala, e a matriz homográfica H definida como $H = A\begin{bmatrix} r_1 & r_2 & t \end{bmatrix}$. O elemento A, é o primeiro termo da Equação 2.8, r_1 e r_2 elementos da primeira e segunda coluna da matriz de rotação e t o vetor de translação.

Como uma das finalidades da calibração é determinar simultaneamente os POE de cada imagem, isto é realizado a através da Equação 2.11, segundo (ZHANG, 1999) através da matriz H, fazendo:

$$H = \begin{bmatrix} h_1 & h_2 & h_3 \end{bmatrix} = sA\begin{bmatrix} r_1 & r_2 & t \end{bmatrix} \tag{2.12}$$

Segundo (ZHANG, 1999) r_1 e r_2 são vetores ortogonais, obtém-se duas restrições básicas nos POI, dada a condição:

$$h_1^T A \frac{1}{A^T} A^{-1} h_2 = 0 \tag{2.13}$$

$$h_1^T A \frac{1}{A^T} A^{-1} h_1 = h_2^T A \frac{1}{A^T} h_2 \tag{2.14}$$

Sendo $B = A^T * A^{-1}$ uma matriz simétrica, é possível definir os elementos de B pelo vetor $bz = \begin{bmatrix} B_{11} & B_{12} & B_{22} & B_{13} & B_{23} & B_{33}. \end{bmatrix}^T$ Seja o $i - ésimo$ vetor coluna da matriz homográfica $h_i = \begin{bmatrix} h_{i1} & h_{i2} & h_{i3} \end{bmatrix}^T$, chega-se então:

$$h_i^T B h_j = v_{ij}^T \tag{2.15}$$

onde $v_{ij} = \begin{bmatrix} h_{i1}h_{j1} & h_{i1}h_{j2} + h_{i2}h_{j1} & h_{i2}h_{j2} & h_{i3}h_{j1} + h_{i1}h_{j3} & h_{i3}h_{j2} + h_{i2}h_{j3} & h_{i3}h_{j3} \end{bmatrix}^T$.

Assim, a partir das Equação 2.13 e Equação 2.14, para cada imagem capturada obtêm-se duas equações homogêneas, como segue (ZHANG, 1999):

$$\begin{bmatrix} v_{12}^T \\ (v_{11} - v_{22}) \end{bmatrix} bz = 0 \tag{2.16}$$

E finalmente, para determinar os POI é necessário encontrar os valores do vetor bz através da Equação 2.16, no qual significa resolver um sistema da forma $V \cdot bz = 0$, sendo V uma matriz de dimensões $2n \times 6$, sendo n o número de imagens capturadas. Para a determinação das coordenadas do ponto principal u_0 e v_0 (sistema imagem) é dado a seguinte fórmula:

$$v_0 = \frac{(B_{12}B_{13} - B_{11}B_{23})}{B_{11}B_{22} - B_{12}^2)} \tag{2.17}$$

$$u_0 = \frac{\tau v_0}{fS_y} - \frac{B_{13} fS_x^2}{\tau} \tag{2.18}$$

Para o cálculo do fator de escala s é empregado a seguinte equação:

$$s = \frac{B_{33} - (B_{13}^2 + v_0(B_{12}B_{13} - B_{11}B_{23}))}{B_{11}} \tag{2.19}$$

Para calcular os fatores de escala fS_x e fS_y em relação ao eixo X e Y respecti-

vamente, é utilizado as seguintes equações:

$$fS_x = \sqrt{\frac{s}{B_{11}}} \tag{2.20}$$

$$fS_y = \sqrt{\frac{sB_{11}}{B_{11}B_{12} - B_{12}^2}} \tag{2.21}$$

O fator de não similaridade τ entre os eixos é calculado pela equação:

$$\tau = -B_{12}fS_x^2\frac{fS_y}{s} \tag{2.22}$$

Segundo Zhang (1999) os POE da câmera é determinado através das seguintes equações:

$$r_1 = \lambda A^{-1}h_1 \tag{2.23}$$

$$r_2 = \lambda A^{-1}h_2 \tag{2.24}$$

$$r_3 = r_1 \otimes r_2 \tag{2.25}$$

$$t = \lambda A^{-1}h_3 \tag{2.26}$$

Sendo, $\lambda = \frac{1}{\|A^{-1}h_1\|} = \frac{1}{\|A^{-1}h_2\|}$ e o operador do produto vetorial. Vale ressaltar que outra solução para a determinação da matriz de rotação pode ser encontrado em Golub e Loan (2012).

2.2.3 Parâmetros de montagem do sistema

Quando determinados os parâmetros de orientação exterior de cada imagem no processo de auto calibração é necessário calcular os parâmetros de montagem do sistema, que representam a orientação relativa $(\Delta_\omega, \Delta_\varphi, \Delta_\kappa)$ e o deslocamento $(\Delta_X, \Delta_Y, \Delta_Z)$ entre as origens dos sistemas das câmeras RGB e IR, de modo a ser possível corrigi-los, por meio da multiplicação do vetor tridimensional por uma matriz de cossenos diretores (MCD). Esta matriz é gerada por vetores unitários em um sistema com bases ligeiramente diferentes. As

matrizes de orientação relativa são obtidas por meio de uma característica da multiplicação de matrizes, generalizada:

$$R_C^A = R_B^A \cdot R_C^B \tag{2.27}$$

Onde:
R: matriz de rotação em função dos ângulos de Euler;
A, B e C: são sistemas com diferentes orientações;
R_Y^X: matriz que rotaciona (muda de base) um vetor do sistema X para o sistema Y;
O Deslocamento entre as origens é dado pela expressão:

$$T_{IR}^{RGB} = T_{RGB} - RT_{IR}^T \tag{2.28}$$

Assim, a matriz que rotaciona um vetor de um sistema A, para um sistema C pode ser obtido pelo produto matricial à direita da matriz de A para B pela matriz de B para C. Para aplicar ao dispositivo, basta substituir os sistemas A, B ou C, pelo sistema adequado.

2.3 EXTRAÇÃO E DETECÇÃO DE PONTOS EM IMAGEM RGB

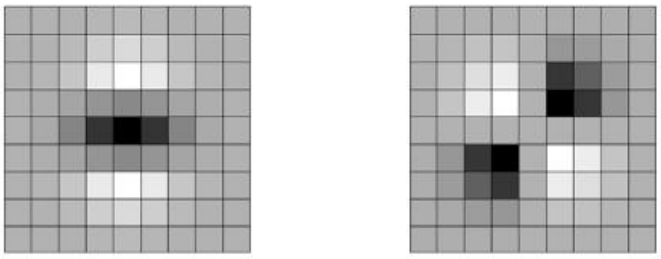
Nesta subseção será descrito o método empregado para extração, detecção e estabelecimento automático de pontos homólogos a partir de pares de imagens RGB. Neste trabalho foi usado o detector e descritor SURF (*Speeded Up Robust Features*). O SURF, proposto por Bay, Tuytelaars e Gool (2006), é um algoritmo rápido e robusto para extração e detecção de pontos em uma imagem RGB. Basicamente, este algoritmo é dividido em duas partes: a extração; e a descrição dos pontos. No algoritmo SURF, os pontos são localizados utilizando um detector Hessiano. As orientações dos pixels são obtidas através da transformada de Haar (*Haar wavelets*) e os descritores são formados a partir da comparação com a vizinhança dos pontos de interesse.

2.3.1 Detector de pontos SURF

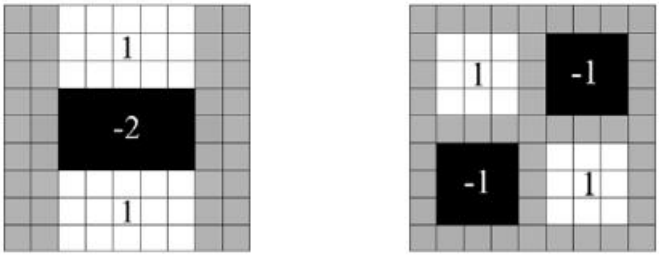
O detector de pontos de interesse do algoritmo SURF é baseado na matriz Hessiana e é conhecido como *Fast-Hessian Detector*. Dado um ponto $x = (x, y)^T$ na imagem , a matriz Hessiana $H(x, \sigma)$ em x na escala σ é definido, como segue (BAY; TUYTELAARS; GOOL, 2006):

$$H(x, \sigma) = \begin{bmatrix} L_{xx}(x, \sigma) & L_{xy}(x, \sigma) \\ L_{xy}(x, \sigma) & L_{yy}(x, \sigma) \end{bmatrix} \tag{2.29}$$

Sendo, $L_{xx}(x, \sigma)$ a derivada de segunda ordem da convolução gaussiana de parâmetro $\sigma \left(\frac{\partial^2}{\partial^2 x} g(\sigma) \right)$ no ponto x, por similaridade tem $L_{yy}(x, \sigma)$ e $L_{xy}(x, \sigma)$. A convolução gaussiana de $9x9$ é reduzida a filtros caixa, como mostra a Figura 2.3, podendo ser rapidamente calculados com o emprego de imagem integral.



(a)



(b)

FIGURA 2.3 – A) DERIVADA DE SEGUNDA ORDEM DA CONVOLUÇÃO GAUSSIANA L_{yy} E L_{xy}, B) CORRESPONDENTES APROXIMAÇÕES USANDO FILTROS CAIXA D_{yy} E D_{xy}
FONTE: Bay, Tuytelaars e Gool (2006)

A imagem integral é uma representação intermediária da imagem original que permite a soma dos valores dos pixeis em áreas retangulares na imagem de forma muito eficiente. Sendo a imagem I de um ponto x, a imagem integral I é a soma dos pixeis contidos em um retângulo que delimitam a origem da imagem $(0,0)$ e do ponto x. A Figura 2.4 (b) mostra a imagem integral calculada até o ponto (200,200).

(a) (b)

FIGURA 2.4 – A) IMAGEM ORIGINAL, B) IMAGEM INTEGRAL CALCULADA DO PONTO $(0,0)$ ATÉ $(200,200)$
FONTE: Algaba, Blanco e González (2012)

A imagem integral pode ser definida com a seguinte expressão:

$$I_{\Sigma} = \sum_{i=0}^{i \leq x} \sum_{j=0}^{j \leq y} I(i,j) \tag{2.30}$$

Após calcular a hessiana para toda imagem em diferentes escalas é obtido o determinante da hessiana rápida em cada ponto, como segue (BAY; TUYTELAARS; GOOL, 2006):

$$det(H_{rapida})(x,\sigma) = D_{xx}(x,\sigma) - (0.9D_{xy}(x,\sigma))^2 \tag{2.31}$$

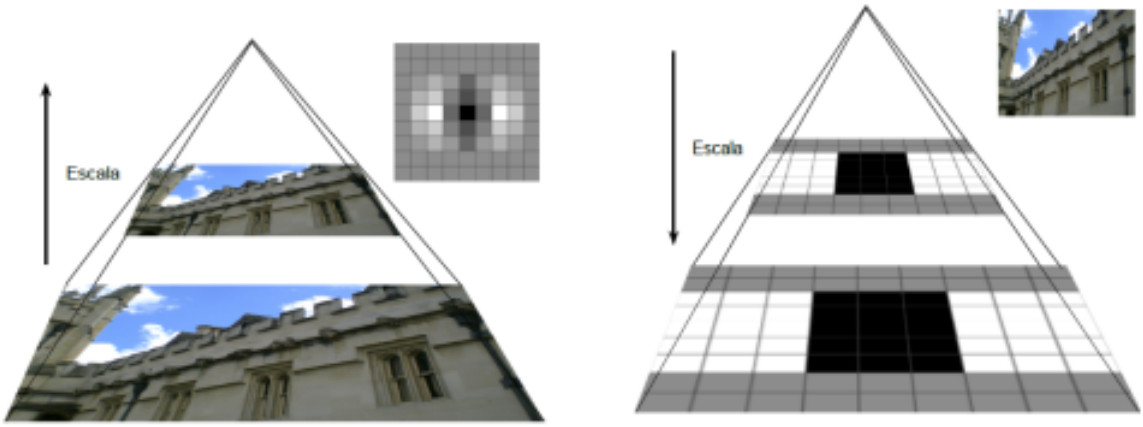
Onde:
$D_{xx}(x,\sigma)$: corresponde ao filtro caixa correspondente a $L_{xx}(x,\sigma)$;
$D_{yy}(x,\sigma)$ e $D_{x,y}(x,\sigma)$ são obtidos por similaridade;

O valor do determinante é utilizado para classificar o máximo e o mínimo da função mediante o teste da segunda derivada, ou seja, se a matriz *Hessiana* é definida positiva em x, então $f(x)$ tem o mínimo local em x. Se a matriz *Hessiana* é definida negativa, então $f(x)$ tem o máximo local em x. Se a matriz *Hessiana* tem autovalores positivos e negativos, então $f(x)$ é um ponto de sela [1]. Como o determinante é um produto dos autovalores da *Hessiana*, os pontos podem ser classificados pelo sinal do resultado. Se o determinante for negativo, então os valores próprios têm sinais diferentes e assim o ponto não é um extremo; se positivo, então os valores próprios são positivos ou negativos

[1]Segundo Hale (1971) ponto de sela é o ponto sobre uma superfície no qual a declividade é nula, mas não se trata de um extremo local (máximo ou mínimo). É o ponto sobre uma superfície na qual a elevação é máxima numa direção e mínima na direção perpendicular. O nome vem da semelhança com uma sela de montaria das superfícies em torno de um ponto de sela

e o ponto é um extremo.

No entanto, para detectar os pontos de interesse mediante o determinante da matriz Hessiana é necessário aplicar o conceito de *espaço-escala*. Um *espaço-escala* é uma função contínua que pode ser usada para encontrar extremos através de diversas escalas. Um *espaço-escala* é implementado como uma pirâmide de imagens em que, iterativamente, são aplicados filtros de suavização por convolução e, desta forma, o tamanho da imagem é reduzido. No detector SURF, a aplicação do *espaço-escala* é feita baseada na construção dos filtros caixas permitindo calcular vários níveis de resposta no *espaço-escala* de forma simultânea. Na Figura 2.5 (a) nota-se que a imagem muda de escala de forma iterativa, enquanto é suavizada por um filtro Gaussiano. Já na Figura 2.5 (b) é empregada uma técnica onde a imagem não muda de escala, uma vez que a convolução é efetuada com filtros de tamanho crescente.



(a) (b)

FIGURA 2.5 – A) ESPAÇO ESCALA TRADICIONAL, B) ESPAÇO ESCALA ADAPTADO PELO SURF
FONTE:Algaba, Blanco e González (2012)

Finalmente, é aplicada uma supressão não máxima nesses determinantes em uma vizinhança $3 \times 3 \times 3$ para encontrar os melhores pontos de interesse. Neste caso, cada ponto candidato é comparado com seus 26 vizinhos mais próximos (8 na escala do candidato e 9 nas escalas superior e inferior), ver Figura 2.6.

Para extrair um ponto com precisão subpixel é aplicada uma expansão de Taylor de segunda ordem na matriz *Hessiana* $H(x,\sigma)$, como segue (BAY; TUYTELAARS; GOOL, 2006):

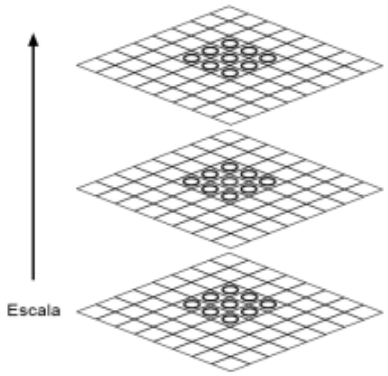


FIGURA 2.6 – SUPRESSÃO NÃO MÁXIMA
FONTE: Adaptado de Bay, Tuytelaars e Gool (2006)

$$H(x,\sigma) = H + \frac{\partial H^T}{\partial x}x + \frac{1}{2}^T \frac{\partial^2 H}{\partial x^2} \tag{2.32}$$

Para extrair um ponto com precisão subpixel é aplicada uma expansão de Taylor de segunda ordem na matriz *Hessiana* $H(x,\sigma)$, como segue:

$$\widetilde{x} = -\left(\frac{\partial^2 H^{-1}}{\partial x^2}\frac{\partial H}{\partial x}\right) \tag{2.33}$$

O ponto interpolado é obtido com o cálculo da matriz 3×3 $\frac{\partial^2 H}{\partial x^2}$ e o vetor 3×1 $\frac{\partial H}{\partial x}$. Os elementos resultantes determinam os níveis de intensidade dos pontos de interesse. A invariância a mudança de iluminação também é obtida graças ao emprego da transformada *Haar*, que mede a diferença de iluminação entre os vizinhos ao invés do valor propriamente dito.

2.3.2 Descritor de pontos SURF

O calculo do descritor SURF pode ser dividido em duas etapas. Na primeira consiste em extrair a informação de orientação dos pontos de interesse. A orientação dos pontos é determinada através da resposta da transformada de *Haar* nas direções x e y, tornando o descritor invariante a rotações. Isto é feito calculando as respostas da transformada*Haar* de tamanho 4σ dentro de um raio 6σ em volta do ponto de interesse detectado. As respostas são ponderadas com uma gaussiana centrada no ponto de interesse com desvio padrão de $2,5\sigma$. As respostas são representadas como um vetor de pontos correspondentes com as direções do vetor gradiente avaliado em cada um dos pontos da região circular. Isso é feito selecionando a direção predominante das respostas. Desta

forma deve ser gerada uma seção com circunferência de $\frac{\pi}{3}$em torno da origem do ponto de interesse e adicionada as componentes das respostas que estão dentro da seção. O vetor resultante de maior módulo representa a orientação resultante do ponto de interesse. O processo de seleção da orientação predominante é ilustrado na Figura 2.7.

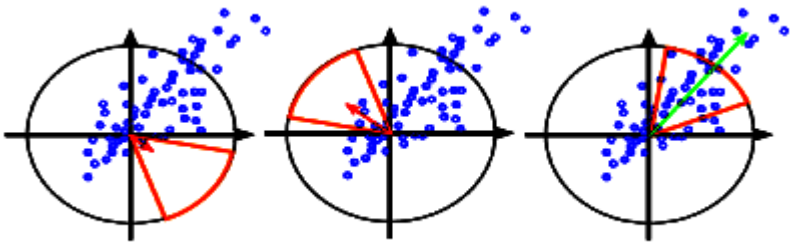


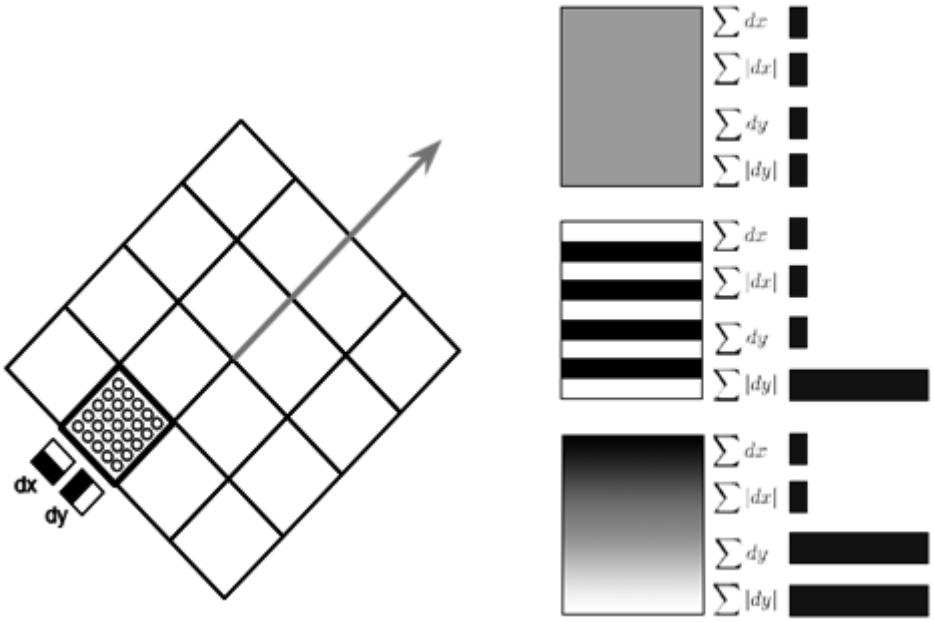
FIGURA 2.7 – ORIENTAÇÃO PREDOMINANTE
FONTE: Algaba, Blanco e González (2012)

A segunda etapa consiste em construir as componentes do descritor. A construção dos componentes do descritor SURF pode ser dividida em dois processos:

- Construir uma janela retangular de tamanho 20σ em torno do ponto de interesse detectado e orientado, conforme a orientação predominante da resposta da transformada de *Haar*;
- Dividir cada janela em sub-regiões retangulares $4x4$ para obter a transformada de *Haar* de cada sub-região. Dentro de cada sub-região calcula-se a resposta da transformada supracitada de tamanho 2σ sobre 25 pontos distribuídos de forma regular sobre a sub-região. Considerando as resposta da transformada *Haar* nas direções x e y , que agora serão denominadas de dx e dy, respectivamente, devido a correção aplicada pelo vetor orientação, cada sub-região apresenta um vetor de 4 elementos, como segue:

$$\Gamma_{subregião} = \begin{bmatrix} \Sigma dx \\ \Sigma |dx| \\ \Sigma dy \\ \Sigma |dy| \end{bmatrix} \tag{2.34}$$

O retângulo marcado na Figura 2.8, ilustra uma das 16 sub-regiões, enquanto os 25 pontos representam os pixels com as respostas da transformada *Haar*. Como é mostrado na Figura 2.8(a), as respostas se comportam em função das orientações predominantes dos



(a) (b)

FIGURA 2.8 – COMPONENTES DO DESCRITOR
FONTE: Algaba, Blanco e González (2012)

pontos de interesse, enquanto na Figura 2.8(b) é mostrado as componentes do descritor SURF calculados para três amostras diferentes.

Como cada sub-região apresenta um vetor com 4 elementos, o descritor resultante é do tamanho $4 \times 4 \times 4 = 64$. Este descritor é invariante a mudança de escala, de rotação e de iluminação.

2.4 ELIMINAÇÃO DE FALSOS POSITIVOS

Esta etapa intermediária consiste em eliminar falsos positivos (*outliers*) detectados pelo SURF. Estes falsos positivos comprometem a qualidade na determinação dos parâmetros de transformação inicial. Para este fim, pode ser empregada a matriz fundamental. Esta matriz faz uma relação geométrica entre os pontos homólogos com a linha epipolar e, posteriormente, pode ser usado o algoritmo RANSAC para selecionar os pares de pontos correspondentes mais prováveis.

2.4.1 Matriz Fundamental

Uma cena que se observa de posições diferentes apresenta uma série de relações geométricas entre os pontos no espaço-objeto e suas respectivas projeções no espaço-

imagem. Estas relações geométricas são obtidas sob a hipótese de que a câmera pode ser aproximada à uma projeção perspectiva, formando assim as bases da geometria epipolar entre duas imagens. A geometria epipolar estabelece as correspondências com os pontos $p_1 \leftrightarrow p_2$ espaço-imagem, com o correspondente ponto P no espaço-objeto e o centro perspectivo do sensor (CP_1 e CP_2), conforme ilustra a Figura 2.9

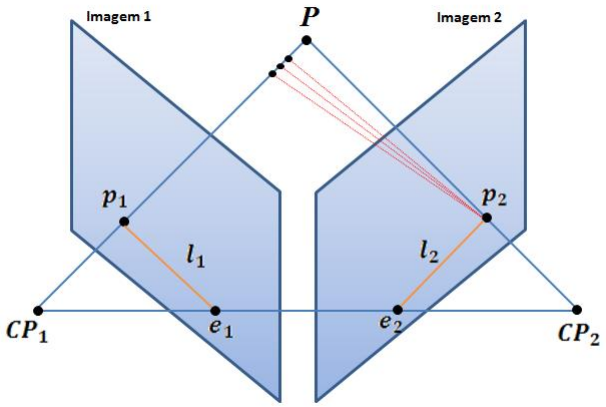


FIGURA 2.9 – GEOMETRIA EPIPOLAR
FONTE: Adaptado de Hartley e Zisserman (2003)

Na geometria epipolar, as coordenadas homogenias de p_1 e p_2 se relacionam através da matriz fundamental F que é mediante a Equação. $p_2 F p_1 = 0$, que na forma matricial é expressa como segue (HARTLEY; ZISSERMAN, 2003):

$$p_2 F p_1 = \begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix} \begin{bmatrix} f_{11} & f_{21} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{23} & f_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix} \tag{2.35}$$

onde,f_{ij} são os coeficientes da matriz fundamental a ser determinados;

A matriz fundamental F é uma matriz 3×3 que representa algebricamente a geometria epipolar, transforma os pontos p_1 de uma $Imagem_1$ em uma linha epipolar $l_2 = F p_1$ sobre a $Imagem_2$. A Equação 2.35 restringe a posição da correspondência do p_2 ponto p_1 sobre a linha epipolar l_2. A relação que exprime a Equação. 2.35 que é extraído, a partir unicamente das correspondência dos pontos $p_1 \leftrightarrow p_2$ em um par de imagens estéreo, pode-se calcular F caso tiver um número suficiente de correspondências $p_1 \leftrightarrow p_2$ a Equação 2.36 pode ser usada para calcular a matriz fundamental. Mais especificamente, escrevendo: $p_1 = \begin{bmatrix} x_1 & y_1 & 1 \end{bmatrix}^T$ e , cada par de correspondências dá lugar a uma equação linear nas incógnitas da f_{ij}. Os coeficientes da equação podem ser facilmente escritos em termos das coordenadas da imagem p_1 e p_2 , como segue (HARTLEY; ZISSERMAN,

2003):

$$x_2x_1 + f_{11} + x_2f_{12} + x_2f_{13} + y_1f_{22} + y_2f_{13} + x_1f_{31}y_1f_{32} + f_{33} = 0 \tag{2.36}$$

Considerando $f = \begin{bmatrix} f_{11} & f_{12} & f_{13} & f_{21} & f_{22} & f_{23} & f_{31} & f_{32} & f_{33} \end{bmatrix}$ um vetor de 9 elementos, por tanto é necessário ter ao menos 9 pares de correspondência para ter solução única, o elemento f_{33} que é o fator de escala, pode ser injucionado com o valor $f_{33} = 1$, desta forma a Equação. 2.37 pode representado na como um produto escalar, da forma que segue:

$$f \begin{bmatrix} x_2x_1 & x_2y_1 & x_2 & y_2x_1 & y_2y_1 & x_2 & x_1 & y_1 & 1 \end{bmatrix} = 0 \tag{2.37}$$

A partir de um conjunto com 8 números de correspondência e de um conjunto de equações se obtém um sistema na forma linear (HARTLEY; ZISSERMAN, 2003):

$$Af = \begin{bmatrix} x_{2i}x_{1i} & x_{1i}y_{1i} & y_{2i}x_{1i} & x_{2i} & y_{2i}x_{2i} & y_{2i} & x_{1i} & y_{1i} & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{2i}x_{1i} & x_{1i}y_{1i} & y_{2i}x_{1i} & x_{2i} & y_{2i}x_{2i} & y_{2i} & x_{1i} & y_{1i} & 1 \end{bmatrix} \tag{2.38}$$

Segundo Hartley e Zisserman (2004), se o posto de é A exatamente 8, a solução do sistema linear é gerada pelo espaço nulo de A . A correspondência dos pontos não é exata, pois ocorrem erros na determinação de correspondências. Uma solução aproximada para o sistema linear homogêneo pode ser obtida pela decomposição em valores singulares (SVD), em que f corresponde ao autovetor da matriz, A^TA relativo ao menor autovalor de A^TA.

A matriz Fundamental é uma matriz singular de posto 2 e os epipolos e_2 e e_1 são calculados pelos espaços nulos à esquerda e a direita de, F respectivamente. Sabe-se que as linhas epipolares são coincidentes aos epipolos, conforme ilustrado na Figura 2.9. A matriz fundamental calculada pela Equação 2.36 nem sempre é singular e pode fazer com que as linhas epipolares não sejam coincidentes (HARTLEY; ZISSERMAN, 2003) .

Para resolver esse problema, deve ser imposta uma condição restritiva de singularidade, substituindo F pela matriz F' sendo que F minimiza $\|F - F'\|$ e possui a restrição que o determinante de F' é igual a zero. A matriz F' pode ser calculada utilizando o SVD. A matriz F é decomposta em $F = UDV^T$ sendo $D = diag(\sigma_1, \sigma_2, \sigma_3)$ que satisfaz $\sigma_1 < \sigma_2 < \sigma_3$ e F' é calculado por $F' = Udiag(\sigma_1, \sigma_2, 0)V^T$ que minimiza $\|F - F'\|$.

A matriz F pode ser obtida diretamente pelas coordenadas no sistema imagem

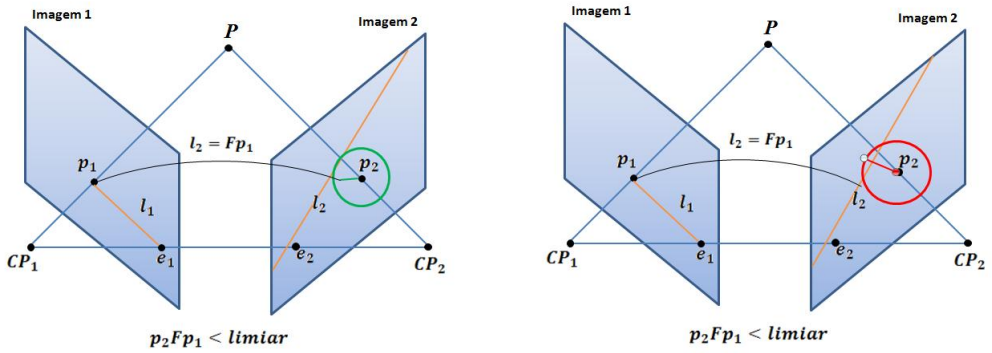
não corrigida dos do deslocamento principal e da distorção de lentes (GALO, 2003), sendo assim pode ser utilizada para recuperar os parâmetros POI para o caso de câmaras não calibradas (BARAKAT; DOUCETTE; MIKHAIL, 1994), (ZHANG, 1994) e (HARTLEY, 1997)

2.4.2 O algoritmo RANSAC

O algoritmo RANSAC foi desenvolvido por Fischler e Bolles (1981), é iterativo e empregado para estimar parâmetros de um modelo matemático a partir de um conjunto de dados observados que contenham valores espúrios e ruídos. No RANSAC é assumido que o conjunto de dados está composto por dois tipos de dados: aqueles que podem ser aplicados ao modelo (*inliers*); e aqueles que não satisfazem o modelo (*outliers*). O algoritmo padrão do RANSAC obtém como valores de entrada, um conjunto M de observações, um modelo que pode estimar mediante N observações e um valor de tolerância para determinar quando uma observação se ajusta ao modelo. O RANSAC trata de estimar de modo iterativo o modelo que melhor se ajusta a os dados da seguinte maneira:

1. Selecionando N observações aleatórias do conjunto de dados como possíveis *inliers*;
2. Estimar o vetor de parâmetros X do modelo fazendo o ajuste nas N observações consideradas como inliers,
3. Calcular o numero de k de observações de M que satisfazem o modelo com o vetor de parâmetros X com determinada tolerância;
4. Se K for alto o suficiente, o modelo é aceito porque se enquadra bem a um número elevado de *inliers* e a função é retornada com êxito;
5. Repetir os passos (1) a (4) em um número limitado de iterações;
6. Caso não foi atingido para estimar um modelo que se ajusta aos dados com tolerância especificadas, a função retorna falso;

O algoritmo RANSAC poderá ser empregado para melhorar a robustez do SURF eliminando os *outiliers* e conseguir um conjunto robusto de correspondências. Figura 2.10 ilustra a detecção de *inliers* e *outliers* a partir da matriz fundamental e do algoritmo RANSAC. Na Figura 2.10 (a), o RANSAC estima a matriz fundamental F e detecta os *outliers* da seguinte forma: caso a distância da linha epipolar$l_2 = Fp_2$ for menor que



(a) (b)

FIGURA 2.10 – A) DETECÇÃO DE *INLIERS*, B)DETECÇÃO DE *OUTLIERS*
FONTE: Adaptado Hartley e Zisserman (2003)

um certo limiar pré-definido, os pares correspondentes serão considerados *inliers*, caso contrário são considerados *outliers* e eliminados do processo (ver Figura 2.10 (b)).

Na Figura 2.11 (a) nota-se, visualmente, a presença de falsos positivos (*outliers*) no estabelecimento de pontos correspondentes. Caso estes pontos não sejam removidos do processamento, podem ser obtidos parâmetros de transformação com valores espúrios comprometendo, de forma significativa, a modelagem 3D do ambiente e também a etapa de detecção de áreas revisitadas, na construção do grafo.

(a) (b)

FIGURA 2.11 – PONTOS DE INTERESSE OBTIDOS PELO SURF, A) PONTOS HOMÓLOGOS E OUTLIERS B) OUTLIERS REMOVIDAS COM O RANSAC E A MATRIZ FUNDAMENTAL
FONTE: O Autor (2015)

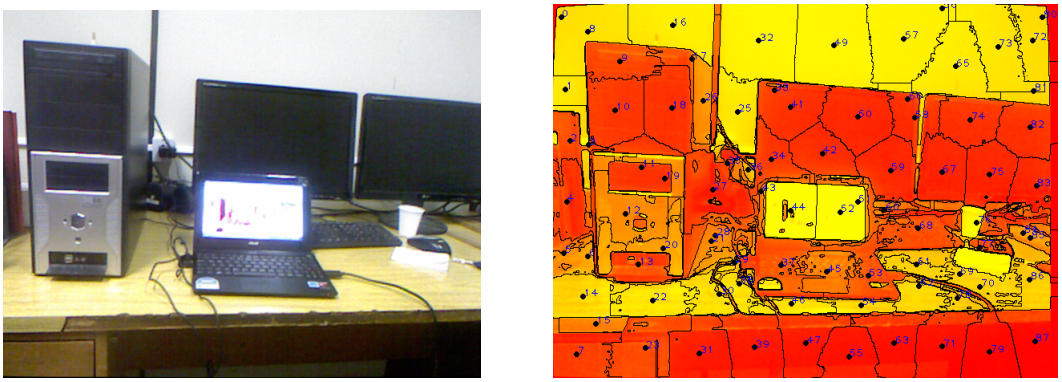
Apesar da correspondência correta, vale ressaltar que os pixels associados com valores de paralaxe nula também são eliminados do processamento, restando apenas os pares de pontos válidos. Estes pontos são diretamente associados com o seus respectivos pontos no espaço-objeto como será visto mais adiante.

2.5 SEGMENTADOR SLIC

Em processamento de imagens e visão computacional, segmentação de imagens é o processo de particionamento de uma imagem digital em vários segmentos (conjuntos de pixels). A finalidade da segmentação é simplificar ou alterar a representação de uma imagem em algo que é mais significativo e mais fácil de extrair informações. De modo geral, a segmentação de imagens é o processo de atribuição de um rótulo para cada pixel em uma imagem de tal modo que pixels com o mesmo rótulo compartilham certas características. Os métodos de segmentação mais conhecidos na comunidade fotogramétrica são os algoritmos baseado no agrupamento por *Thresholding* como crescimento de regiões (FU; MUI, 1981) e o Watershed (RUSS, 2015). Outro grupo de algoritmos para segmentação de imagens que será aplicado dentro do contexto desse trabalho, para a extração de planos na imagem são os algoritmos baseados em *data-clustering*. Dentro desta classe de algoritmos existem os algoritmos denominados superpixels.

Nos últimos anos diversos algoritmos foram desenvolvidos visando obter *superpixels* em uma imagem como Turbo Pixels apresentado por Levinshtein et al. (2009) que consiste agrupar *superpixels* a partir de um fluxo geométrico, a ideia básica é criar e dilatar pixels sementes cuja a borda contenham superpixels; o STP (*Speed-Up Turbo Pixels*) apresentado por Çiğla e Alatan (2010) o método consiste em modelar as imagens como um grafo ponderado cujos vértices representam o *superpixels* e cortes normalizados são utilizados para obter segmentação final.

Para extração dos planos da imagem RGB-D fornecido pelo dispositivo Kinect será empregado o segmentador SLIC (*(Simple linear iterative clustering)*) proposto por Achanta et al. (2010). Este algoritmo gera *superpixels* mediante ao agrupamento de pixels baseado em sua cor, similaridade e proximidade dos pixels no plano da imagem. Isso é feito nas cinco dimensões do espaço de cores Lab_{xy}, sendo L_{ab} o vetor cor do pixel no espaço de cor CIELAB *(Commision Internationale L'Eclairage Lab)* e xy é posição dos pixels. O espaço de cor CIELAB é utilizado por ser perceptivelmente uniforme para pequenas distâncias radiométricas de cor. Não é possível simplesmente utilizar a distância euclidiana neste espaço $5D$ sem normalizar as distâncias espaciais. De forma a agrupar pixels neste espaço $5D$, que, por conseguinte, apresentar uma nova distância medida que considera o tamanho *superpixel*. Segundo (ACHANTA et al., 2012) é imposto semelhanças de cores bem como a proximidade do pixel neste espaço 5D tal que os tamanhos esperados dos fragmentos em sua extenção espacial sejam aproximadamente iguais (ver Figura 2.12).



(a) (b)

FIGURA 2.12 – A) IMAGEM ORIGINAL, B) IMAGEM SEGMENTADA PELO SLIC
FONTE: O Autor (2015)

O algoritmo tom como entrada um número desejado tamanho *superpixels* K. Para uma imagem com N pixels, o tamanho aproximado de cada super pixel é portanto $\frac{N}{K}$ pixels. Para *superpixels* aproximadamente de igual tamanho, haveria um centroide para cada *superpixel* em cada intervalo da grade $S = \sqrt{\frac{N}{K}}$. Inicialmente são selecionados os K centroides dos *superpixels* $C_k = \left[l_k, a, b_k, x_k, y_k\right]$ com $k = \left[1, k\right]$ dos intervalos regulares da grade S.

Dado que a extensão espacial de qualquer *superpixel* é de aproximadamente S^2 (área aproximada de um super pixel), pode-se assumir que os pixels que estão associados a um centroide tem uma área de $2S \times 2S$ em torno dele no plano xy do *superpixel*. Tornando a área de pesquisa para os pixels mais próxima a cada centro de agrupamento. A distância Euclidiana no espaço de cor CIELAB é perceptivamente significativa para distâncias pequenas (m na Equação 2.41). Se as distâncias espaciais dos pixels excedem o limite de percepção de cor, então eles começam a superar as semelhanças de cor (resultando em superpixels que ignoram os limites da região, apenas a proximidade no plano da imagem). Portanto, em vez de usar uma norma simples Euclidiana no espaço $5D$, é usado a função Ds para definir a distância, definida a seguir:

$$d_{Lab} = ((l_k - l_i)^2 + (a_k - a_i)^2 + (b_k - b_i)^2)^{\frac{1}{2}} \tag{2.39}$$

$$d_{xy} = ((x_k - x_i)^2 + (y_k - y_i)^2)^{\frac{1}{2}} \tag{2.40}$$

$$D_s = d_{Lab} + \frac{mk}{s} d_{xy} \tag{2.41}$$

Onde:
D_s : é a soma da distância Lab ;
xy : distância plana normalizada S;

Conforme Achanta et al. (2012) o parâmetro m inserido em D_s permite o controle da convexidade dos *superpixels*. Quanto maior o valor de m, mais será enfatizada a proximidade espacial. Este valor pode estar no intervalo com valores de 1 a 20. O algoritmo começa testando os K centroides regularmente espaçados e, movê-los para localizações de sementes correspondentes para a posição mais inferior do gradiente numa vizinhança 3×3. Isto é feito para evitar colocá-los numa borda (área fronteiriça dos objetos na imagem) e para reduzir as possibilidades do algoritmo escolher um pixel ruidoso. Os gradientes da imagem são calculados como mostra a equação a seguir (ACHANTA et al., 2012):

$$G(x,y) = \|I(x+1,y) - I(x-1,y) - I(x-1,y)\|^2 + \|I(x,y+1) - I(x-1,y) - I(x,y-1)\|^2 \tag{2.42}$$

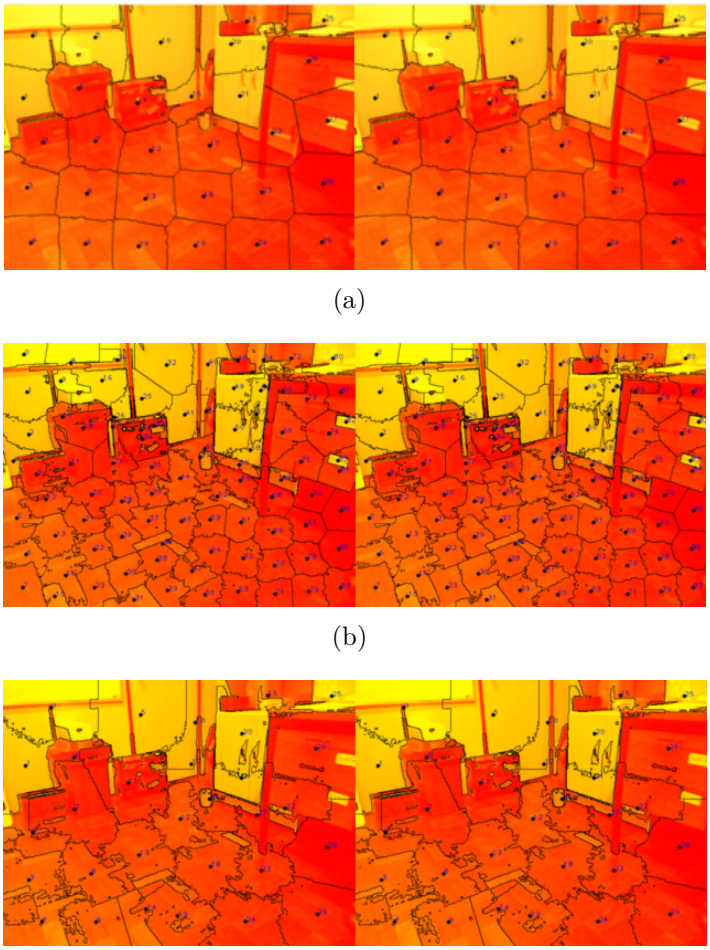
Onde:
$G(x,y)$: gradiente da imagem;
$I(x,y)$:é o vector de lab correspondente ao pixel na posição (x,y);
$\|.\|$: referente a norm ℓ^2 [1].

Isso leva em conta a cor de intensidade. Cada pixel na imagem é associado ao centroide mais próximo, cuja área de pesquisa alcance o pixel. Depois que todos os pixels estiverem sido associados ao centroide mais próximo, um novo centroide é calculado como a média do vetor de todos os pixels pertencentes aos superpixels. O algoritmo repete, de forma iterativa, o processo de associar pixels com o centroide mais próximo até a convergir. Ao finalizar o processo, alguns rótulos podem estar dispersos, ou seja, poucos pixels na vizinhança de um segmento têm o mesmo rótulo que ele, mas não estão conectados ao segmento. Embora seja raro, isso pode ocorrer, porque o algoritmo não reforça a conectividade explicitamente (ACHANTA et al.,2010). No entanto, o método reforça a conectividade no ultimo passo do algoritmo por meio da re-rotulação de segmentos disjuntos com o rotulo do maior segmento+ vizinho.

A Figura 2.13 mostra os segmentos obtidos com diferentes valores de K e m_k, na Figura 2.13 (a) foi utilizado um valor de $K = 40$ e e $m_k = 20$foram segmentados 28 superpixels (S). Nota-se que os segmentos obtidos não são fidedignamente representados, pois não definem bem a borda que separam os objetos. Já na Figura 2.13 (b) foram

[1] ℓ^2 ou $|x|$é uma norma do vetor definido para um vetor complexo (HORN; JOHNSON, 2012)

usados valores de 100 e 10 para K e m_k, respectivamente. Nota-se, portanto, os segmentos definem melhor as bordas dos objetos. Na Figura 2.13 (c) foram empregados valores de $K = 10$ e $m_k = 10$.



(a)

(b)

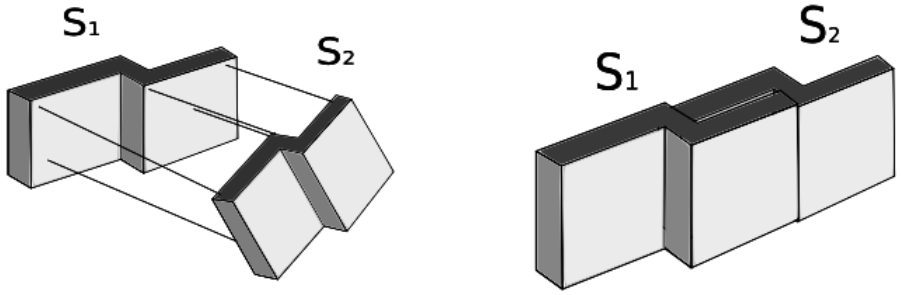
(c)

FIGURA 2.13 – A) SLIC COM PARÂMETROS $K = 40$ E $m_k = 20$, B) SLIC COM PARÂMETROS $K = 100$ E $m_k = 10$, C) SLIC COM PARÂMETROS $K = 40$ E $m_k = 20$
FONTE: O Autor (2015)

Conforme apresentado na Figura 2.13 (a) Nota-se que os segmentos obtidos não são fidedignamente representados, pois não definem bem a borda que separam os objetos. Já Figura 2.13 (b) foram usados valores de $K = 100$ e $m = 10$ para e , respectivamente. Nota-se, portanto, os segmentos são melhores definidos. Mas existe um porém, quanto maior o número de superpixels menor o número de pixels em cada segmento, o que dificulta o algoritmo em determinar os parâmetros do plano. Já na Figura 2.13 (c) foram empregados valores de $K = 40$ e $m = 10$, sendo obtidos os melhores resultados.

2.6 REGISTRO DO PARES DE NUVENS 3D

Dado um conjunto de pontos o registro dos pares de nuvens de pontos 3D consiste em determinar os parâmetros de transformação relativa entre os dados. O modelo matemático mais empregado em abordagens *ponto-a-ponto* é a modelo de corpo-rígido 3D. A transformação do corpo-rígido 3D tem como característica modificar apenas as rotações e translações do objeto transformado, preservando portanto sua forma e sua escala. Em hipótese, se um conjunto de pontos 3D correspondentes está desalinhado e transladado um em relação ao outro, como mostra a Figura2.14(a) basta aplicar uma transformação de corpo rígido e minimizar os efeitos supracitados, como apresentado na Figura2.14(b).



(a) (b)

FIGURA 2.14 – TRANSFORMAÇÃO DO CORPO-RÍGIDO, A)PONT0S CORRESPONDENTES COM DESALINHAMENTO ANGULAR E LINEAR ,B) EFEITO MINIMIZADO APÓS A TRANSFORMAÇÃO
FONTE: O Autor (2015)

Na prática para cada ponto no espaço $S_1 = (X_{k1}, Y_{k1}, Z_{k1})$ pode ser definido um mapeamento isogonal com escala (λ) fixa par as coordenadas no espaço $S_2 = (X_{k_2}, Y_{k2}, Z_{k2})$ através da seguinte relação funcional:

$$F(X_a) : \begin{bmatrix} X_{k2} \\ Y_{k2} \\ Z_{k2} \end{bmatrix} = R(\omega, \varphi, \kappa) \begin{bmatrix} X_{k1} \\ Y_{k1} \\ Z_{k1} \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} \tag{2.43}$$

Onde:

$X_{k_1}, Y_{k_1}, Z_{k_1}$: coordenadas referente ao espaço S_1;

$X_{k_2}, Y_{k_2}, Z_{k_2}$: coordenadas referente ao espaço S_2;

t_x, t_y, t_z: vetor de translação;

ω, φ, κ:ângulos eulerianos de rotação;

A matriz de rotação R que contêm os ângulos eulerianos é definido por:

$$R = \begin{bmatrix} r_{11} & r_{21} & r_{31} \\ r_{21} & r_{22} & r_{32} \\ r_{31} & r_{23} & r_{33} \end{bmatrix} \tag{2.44}$$

Onde:
$r_{11} = cos(\varphi)cos(\kappa)$;
$r_{12} = cos(\omega)sen(\kappa) + sen(\omega)sen(\varphi)cos(\kappa)$;
$r_{13} = sin(\omega)sen(\kappa) - cos(\omega)sen(\varphi)cos(\kappa)$;
$r_{21} = cos(\varphi)sen(\kappa)$;
$r_{22} = cos(\omega)cos(\kappa) - sen(\omega)sen(\varphi)sen(\kappa)$;
$r_{23} = sen(\omega)cos(\varphi) + cos(\omega)sen(\varphi)sen(\kappa)$;
$r_{31} = sen(\varphi)$;
$r_{32} = -sen(\varphi)cos(\phi)$;
$r_{33} = cos(\omega)cos(\varphi)$

A Equação 2.43 envolve 6 parâmetros $X = \begin{bmatrix} t_x & t_y & t_z & \omega & \varphi & \kappa \end{bmatrix}^T$ a serem determinados pelo *MMQ*. Neste caso o modelo paramétrico é não linear, deve ser aplicados valores iniciais $X^0 = \begin{bmatrix} t_{x0} & t_{y0} & t_{z0} & \omega_0 & \varphi_0 & \kappa_0 \end{bmatrix}^T$. A partir desse vetor é calculado a primeira aproximação dos valores do vetor X através da equação (GEMAEL, 1994):

$$_uX_1 = -(_nA_u^T{}_nP_nA)^{-1}{}_nA_u^T{}_nP_{nn}L_1 \tag{2.45}$$

Onde:

n: número de observações, ou seja, das coordenadas 3D das correspondência corretas;
u: número de parâmetros; A: matriz das derivadas parciais em relação aos parâmetros definido por $A = \left.\frac{\partial F(X)}{\partial X}\right|_{X_a = X_0}$
P: matriz dos pesos;
L:é o vetor da diferença entre os valores observados Lb com as observações estimadas a partir dos parâmetros iniciais $F(X_0) = L_0$, resultando em:

$$L = L_b - L_0 \tag{2.46}$$

Determinados os valores do vetor X é realizado um teste, enquanto os valores destes parâmetros forem maior que um determinado limiar, os valores de X serão corrigidos

ao vetorX_0 resultado no vetor dos parâmetros ajustados X_a, dado por:

$$X^a = X^0 + X \tag{2.47}$$

O vetor dos resíduos ${}_nV_1$, é o vetor das correções das observações L_b, expresso pela seguinte equação:

$$V = AX + L \tag{2.48}$$

A estimativa de precisão dos parâmetros, é definido a partir da variância de unidade de peso a *posteriori* $\hat{\sigma}_0^2$, dado pela equação:

$$\hat{\sigma}_0^2 = \frac{V^T PV}{n-u} \tag{2.49}$$

Frequentemente tem-se interesse no conhecimento da precisão do parâmetros calculados através da matriz variância-covariância Σ_{X_a}, através da seguinte equação:

$$\Sigma_{X_a} = \hat{\sigma}_0^2 (A^T PA)^{-1} \tag{2.50}$$

Considerando uma sequencia finita de quadros $(frame_0, frame_1, \ldots frame_n)$ e um conjunto de parâmetros de transformação (pose do sensor), denominado de $Pose_k$, a matriz e transformação no $frame_k \in [0, n]$ é representada por:

$$Pose_k = \begin{bmatrix} r_{11} & r_{21} & r_{31} & t_x \\ r_{21} & r_{22} & r_{32} & t_y \\ r_{31} & r_{23} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.51}$$

Para $i > 0$ tem-se a transformação de T_{i-1}^i que liga a posição $Pose_i$. Cada transformação T pode ser representado pela matriz 4×4 (Equação 2.51). Se $Pose_0$ determina a posição e orientação inicial, pode-se então calcular a $Pose_i$ através do produto de todas as transformações, como segue:

$$Pose_i = \prod_{i=n}^{1} T_{i-1}^i Pose_0 \tag{2.52}$$

Como o produto de matrizes não é comutativa é essencial seguir a ordem correta ao multiplica-las . Por exemplo, caso queira determinar a trajetória do sensor com uma

sequência de 12 *frames* ($Pose_{12}$), realiza-se a seguinte operação:

$$Pose_{12} = T_{11}^{12} T_{10}^{11} T_{10}^{9} \dots T_{4}^{3} T_{3}^{2} T_{2}^{1} T_{1}^{0} Pose_0 \tag{2.53}$$

Onde:

Arbitrando portanto a posição e orientação inicial como zero a $Pose_0$ resulta na matriz identidade ${}_4I_4$, como segue:

$$Pose_0 = \begin{bmatrix} & & t_{x0} \\ & R_0 & t_{y0} \\ & & t_{z0} \\ & & 1 \end{bmatrix} = {}_4I_4 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.54}$$

Aplicando as transformações consecutivas da Equação 2.53 do sensor no espaço 3D, descrevendo portanto a seguinte trajetória conforme ilustra a Figura 2.15:

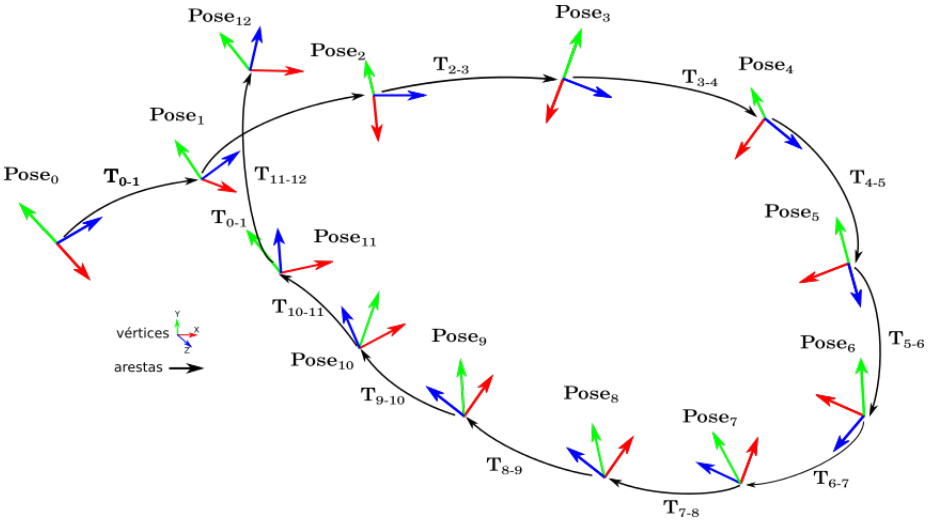


FIGURA 2.15 – TRAJETÓRIA ESTIMADA DO SENSOR
FONTE: O Autor (2015)

Apesar de serem empregados recursos na eliminação de *outliers* isto não garante consistência do modelo 3D a ser criado devido a erros sistemáticos e aleatórios presentes no sensor. Para a minimização destes erros é necessário realizar uma análise de consistência global.

2.7 CONSTRUÇÃO DO GRAFO

2.7.1 Detecção de lugares anteriormente visitados

Sem qualquer informação sobre o percurso realizado pelo sensor e simplesmente, mantendo um histórico dos *frames* anteriores é possível verificar se o *frame* atual *(frame-chave)* corresponde a algum das anteriormente visitados. Se a observação atual contém semelhanças suficientes com as observações anteriormente visitadas, a transformação pode ser calculada entre as nuvens de pontos. Desta forma, uma nova restrição (arestas) pode ser inserida a partir dele, podendo ser várias vezes repetidas. Para fazer esta verificação no histórico de *frames* anteriormente visitados é utilizado os pontos detectados pelo SURF e filtrados pelo RANSAC (*inliers*). Existem diversos outros métodos para esta finalidade, tais como histogramas de imagens, transformada de *Fourier-Mellin* (CHECCHIN et al., 2010), redes neurais (AL-MANASIR; FRASER, 2006), entre outros. Mas aplicando o numero de *inliers*, alem de possibilitar a identificação das semelhanças entre os lugares revisitados, possibilita também realizar a transformação rígida entre as nuvens correspondentes. Neste caso, se o número de *inliers* for maior que um limiar pré-determinado, os *frames* anteriormente visitados serão considerados como áreas revisitadas e o *frame* é detectado. A Figura 2.16 ilustra as restrições do grafo, as arestas de cor preta representam as ligações de *frames* consecutivos, as arestas na cor magenta são denominadas de lugares revisitados.

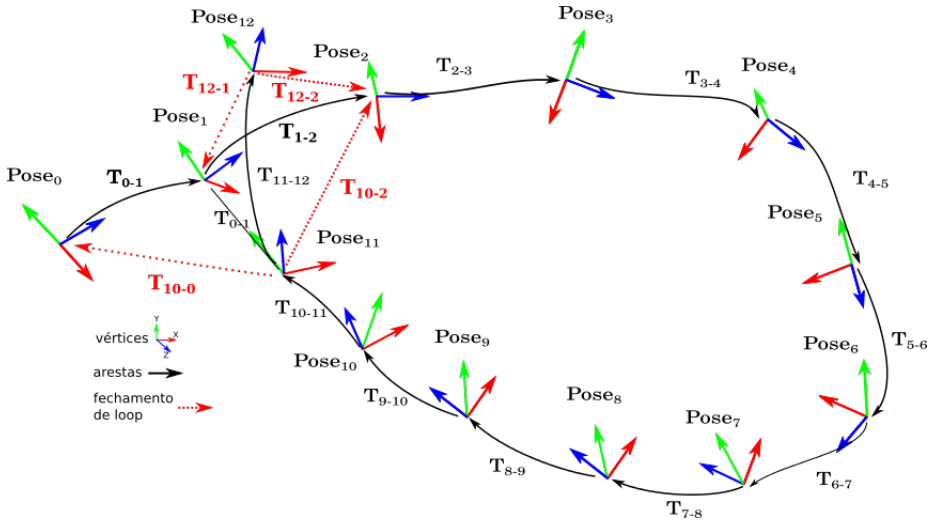


FIGURA 2.16 – GRAFO CRIADO COM AS RESPECTIVAS RESTRIÇÕES
FONTE: O Autor (2015)

O ideal para visitar o histórico de *frames* seria permutar todos os vértices do grafo, formando assim um grafo completo, além de desnecessário é computacionalmente

inviável para grandes cenários imageados, uma vez que novas observações são constantemente adicionadas. Várias alternativas podem ser empregadas utilizando um subconjunto de vértices e, a partir disto, comparar com o histórico de imagens correspondentes ao subconjunto. Kerl, Sturm e Cremers (2013) elaborou uma esfera de tamanho arbitrário que percorre o grafo e através do frame-chave , os vértices contidos dentro da esfera, são considerados *frames* candidatos, caso o numero de *inliers* seja acima do valor estabelecido o local é considerado revisitado; Henry et al. (2012) e Algaba, Blanco e González (2012) agruparam os vértices mais próximos ao *frame-chave* para detectar os lugares anteriormente visitados.Högman (2012) empregou uma janela deslizante que agrupa um histórico de 5 frames candidatos. Neste trabalho, o frame que tiver maior número de inliers é considerado de maior potencial e uma nova aresta é criada entre o vértice potencial e vértice correspondente ao *frame-chave*. A Figura2.17 apresenta o imageamento de uma mesa de escritório , observa-se o grafo representando a trajetória do dispositivo RGB-D, com as respectiva restrições criadas na cor ciano, em vermelho a trajetória estimada, na cor verde a trajetória (ideal) e na cor magenta o grafo otimizado.

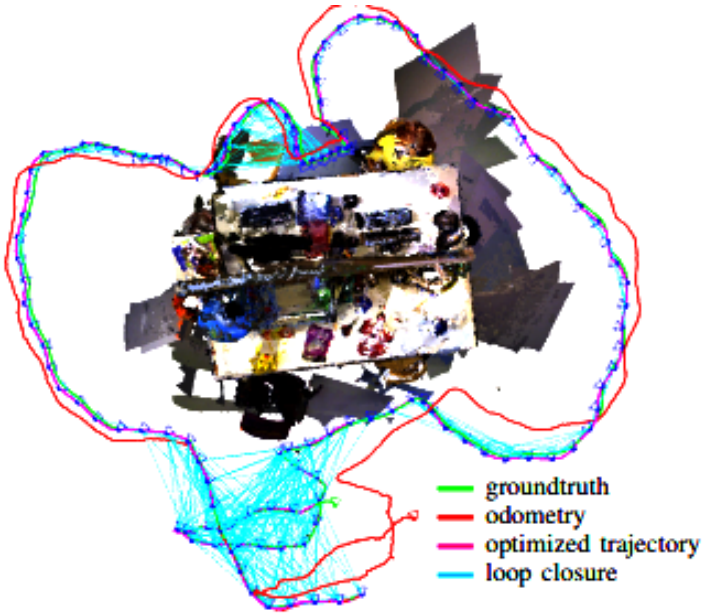


FIGURA 2.17 – GRAFO DE POSE COM OS FECHAMENTOS DE LOOP
FONTE: Kerl, Sturm e Cremers (2013)

Usualmente, na estrutura da aresta do grafo são inseridos os seguintes dados: os parâmetros de transformação relativa entre os pares de nuvens de pontos correspondente aos dois vértices; e a matriz informação Ω_{ij} . Essa matriz é um dado importante na ponderação do grafo, pois representa a qualidade dos parâmetros de transformação.

2.7.2 Otimização do grafo

Considerando $x = [x_1, \ldots, x_T]^T$ um vetor de parâmetros de estado, sendo cada x_i contendo a posição e orientação relativa do vértice i e as arestas do grafo contendo z_{ij} e Ω_{ij} e que são, respectivamente, as medidas observada e a matriz de transformação entre os 2 vértices i e j. A predição da medida nada mais é que a transformação relativa entre os dois vértices. O logaritmo da função de máxima verossimilhança é dado por (GRISETTI et al., 2010b):

$$l_{ij} \propto [z_{ij} - \hat{z}_{ij}(x_i, x_j)]^T \Omega ij [z_{ij} - \hat{z}_{ij}(x_i, x_j)] \tag{2.55}$$

A função de erro $e(x_i, x_j, z_{ij})$ calcula a diferença entre a medida estimada z_{ij} e a medida realizada pelo sensor $\hat{z}_{ij}$. Por simplificação de notação, tem-se (GRISETTI et al., 2010b)

$$e(x_i, x_j) = z_{ij} - \hat{z}_{ij}(x_i, x_j) \tag{2.56}$$

Na Figura 2.18 são encontradas as funções e os parâmetros utilizados para definir as arestas do grafo que conectam o vértice x_i ao vértice x_j. Esta aresta é originada a partir das observações z_i,j ou seja, fornecida a posição dos vértices é possível calcular a medida esperada $widehatz_{ij}$ que representa x_j viso a partir de x_i . O erro $e(x_i, x_j)$ depende do deslocamento entre a medida real e a esperada. Uma aresta é caracterizada por sua função de erro $e(x_i, x_j)$ pela matriz de informação Ω_{ij} da medida que contém uma incerteza.

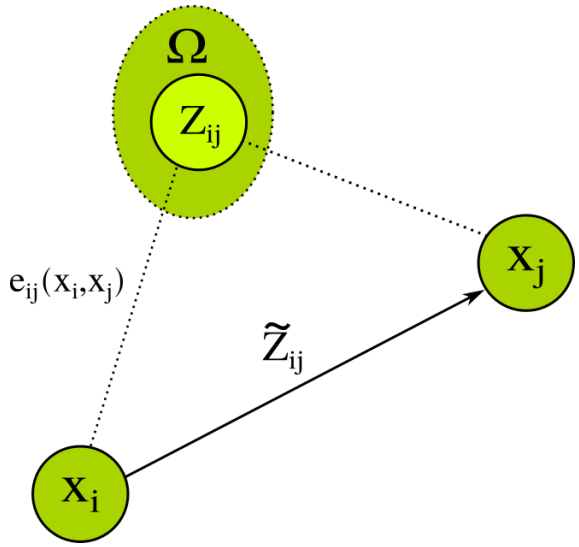


FIGURA 2.18 – ASPÉCTO DA ARESTA QUE SE CONECTA AO VÉRTICE x_i AO x_j
FONTE: adaptado de Grisetti et al. (2010b)

Seja $\mathcal{C}$ um conjunto de pares de índices, para o qual existe uma restrição de

observação z . O critério de otimização é baseado no critério de máxima verossimilhança que consiste em encontrar as configurações dos vértices $x*$ que minimiza o funcional $F(x)$ de todas as observações, como segue (GRISETTI et al., 2010b):

$$F(x) = \sum_{ij \in \mathcal{C}} \underbrace{e_{ij}^T \Omega_{ij} e_{ij}}_{F_{ij}} \tag{2.57}$$

Portanto, para resolver a Equação 2.57, faz-se (GRISETTI et al., 2010b):

$$x* = \underbrace{argmin(F(x))}_{x} \tag{2.58}$$

Através de um parâmetro inicial que corresponde à posição do sensor pode-se obter a solução numérica da Equação 2.58 utilizando os algoritmos de $Gauss - Newton$ ou $Levenberg - Marquardt$. O objetivo é aproximar a função erro por sua expansão da primeira ordem de Taylor em torno da estimativa inicial $\breve{x}$, como segue (GRISETTI et al., 2010b):

$$e_{ij}(\breve{x} + \Delta x_j, \Delta x_j) = e_{ij}(\breve{x} + \Delta_x) \tag{2.59}$$

$$\cong e_{ij} + J_{ij} \Delta x \tag{2.60}$$

A matriz J é a matriz jacobiana de $e_{ij}(x)$ avaliada em $\breve{x}$ e $e_{ij}(\breve{x})$, definida como:

$$J_{ij} = \begin{bmatrix} 0 \dots 0 & \underbrace{\mathbf{A}_{ij}}_{vertice_i} & 0 \dots 0 & \underbrace{\mathbf{B}_{ij}}_{vertice_j} & 0 \dots 0 \end{bmatrix} \tag{2.61}$$

Onde:
$\mathbf{A}_{ij} = \frac{\partial e_{ij}(x)}{\partial (x_i)}$ é a matriz das derivadas parciais de e_{ij} em relação ao vértice i;
$\mathbf{B}_{ij} = \frac{\partial e_{ij}(x)}{\partial (x_j)}$ é a matriz das derivadas parciais de e_{ij} em relação ao vértice j;

Substituindo a Equação 2.60 em termos do erro da Equação 2.57, tem-se (GRISETTI et al., 2010b):

$$F_{ij} = e_{ij}(\breve{x} + \Delta x)^T \Omega ij e_{ij}(\breve{x} + \Delta x) \tag{2.62}$$

$$\cong (e_{ij} + J_{ij} \Delta x)^T \Omega_{ij} (e_{ij} + J_{ij} \Delta x) \tag{2.63}$$

$$= \underbrace{e_{ij}^T \Omega_{ij} e_{ij}}_{c_{ij}} + 2\underbrace{e_{ij}^T \Omega ij J_{ij}}_{b_{ij}} + \Delta_x + \Delta x^T \underbrace{J_{ij}^T \Omega_{ij} J_{ij}}_{H_{ij}} \Delta x \tag{2.64}$$

Onde b_{ij} é dado pelo seguinte vetor:

$$b_{ij} = \begin{bmatrix} \vdots \\ A_{ij}\Omega_{ij}e_{ij} \\ \vdots \\ B_{ij}\Omega_{ij}e_{ij} \\ \vdots \end{bmatrix} \tag{2.65}$$

E a matriz Hessiana H_{ij}, é dado como:

$$H_{ij} = \begin{bmatrix} \ddots & & & \\ & \mathbf{A}_{ij}^T\Omega_{ij}\mathbf{A}_{ij} & \dots & \mathbf{A}_{ij}^T\Omega_{ij}\mathbf{B}_{ij} \\ & \vdots & \ddots & \vdots \\ & \mathbf{B}_{ij}^T\Omega_{ij}\mathbf{A}_{ij} & \dots & B_{ij}^T\Omega_{ij}\mathbf{B}_{ij} \\ & & & \ddots \end{bmatrix} \tag{2.66}$$

Com esta aproximação "local"pode-se escrever o funcional $F(x)$ da Equação 2.57, como segue (GRISETTI et al., 2010b):

$$F(\breve{x} + \Delta x) = \sum_{ij \in \mathcal{C}} F_{ij}(\breve{x} + \Delta x) \tag{2.67}$$

$$\cong \sum_{ij \in \mathcal{C}} c_{ij} + b_{ij}\Delta x + \Delta x^T + H_{ij}\Delta x \tag{2.68}$$

$$= c + 2b^T \Delta x \Delta x + \Delta x^T + H\Delta x \tag{2.69}$$

Finalmente derivando o funcional $F(\breve{x} + \Delta x)$ em relação a Δx, e igualando a zero, é obtido o sistema linear, que se deseja resolver (GRISETTI et al., 2010b)

$$H\Delta x* = -b \tag{2.70}$$

A solução linearizada se obtém atualizando a estimativa inicial com os incrementos calculados, como segue:

$$x^* = \breve{x}^* + \Delta x \tag{2.71}$$

O procedimento descrito acima é baseado em uma abordagem clássica para a minimização da função erro com diversas variáveis. Nesta abordagem é assumido que os parâmetros se encontram em um espaço Euclidiano. Porém, ocorre um problema nas componentes de rotação, na determinação dos parâmetros de transformação rígida, uma vez que não representam nenhuma forma no espaço Euclidiano e, portanto, pode levar a soluções *sub-ótimas*. Por isso, o conceito de *Manifold* [2] será abordado aqui.

O *Manifolds* é um espaço matemático não necessariamente euclidiano, mas que pode ser tratado como tal. Apesar da determinação da translação formar um espaço euclidiano, isto não ocorre na determinação dos parâmetros de rotação, podendo apresentar algumas desvantagens. Uma dessas desvantagens seria um efeito chamado de "*bloqueio de gimbal*"que na prática seria o alinhamento de dois eixos de rotação (ALGABA; BLANCO; GONZÁLEZ, 2012). Caso φ sofra uma rotação de $\pm 90^o$ irá se alinhar ao eixo ω. Desta forma, haverá perda de um grau de liberdade nas componentes de rotação e por mais que ocorra rotação em κ não será compensado.

Uma estratégia para representar os ângulos eulerianos consiste em utilizar um *Manifold* especificado pelo operador $\boxplus$, fazendo a ligação entre uma variação do vetor Δ_X no espaço euclidiano e uma variação no *Manifold*, como segue (GRISETTI et al., 2010b):

$$\Delta_x \rightarrow x \boxplus \Delta x \tag{2.72}$$

Com este operador uma nova função erro é definida:

$$\breve{e}_{ij}(\Delta\tilde{x}_i, \Delta\tilde{x}_j) \triangleq e_{ij}(\breve{x}_i \boxplus \Delta\tilde{x}_i, \breve{x}_j \boxplus \Delta\tilde{x}_j) \tag{2.73}$$

$$= \breve{e}_{ij}(\breve{x} \boxplus \Delta\tilde{x}) \cong \breve{e}_{ij} + \tilde{J}_{ij} + \Delta\tilde{x} \tag{2.74}$$

Onde $\breve{x}$ se estende ao longo do espaço original "sobre-parametrizado", por exemplo quaternions, que é uma alternativa para a parametrização das rotações no espaço. O termo $\Delta\tilde{x}$ é um pequeno incremento em torno $\breve{x}$ da posição original e é expressa em uma representação minimalista (GRISETTI et al., 2010a). O vetor Δx^T é um vetor de 6 posições, $\Delta x^T = (\Delta\tilde{t}^T, \tilde{q}^T)$, onde $\Delta\tilde{t}^T$ denota as translações e $\tilde{q}^T$ as rotações $\begin{bmatrix}\Delta q_x & \Delta q_y & \Delta q_z\end{bmatrix}^T$. Re-

[2] Variedade (*manifold*) é um espaço topológico que é localmente Euclidiano (isto é, em torno de cada ponto, existe uma vizinhança que é topologicamente a mesma que a unidade de esfera aberta em $\Re^N$). Para ilustrar essa ideia, considere a antiga crença de que a Terra era plana, em contraste com a evidência moderna que ela é redonda. A discrepância surge essencialmente do fato de que nas pequenas escalas que vemos, a Terra, de fato, parece plana. Em geral, qualquer objeto que é quase "plano"em pequenas escalas é uma *variedade*(ROWLAND, 2010).

ciprocamente, $\breve{x}^T = (\breve{t}^T, \breve{q}^T)$ utiliza o quartenion $\breve{q}$ para codificar a parte racional. Assim, o operador $\boxplus$ poderá expressar a conversão $\Delta\tilde{q}$ em um quartenion completo Δq seguindo a transformação $\Delta x^T = (\Delta t^T \Delta q^T)$ para $\breve{x}$. Com a nova função de erro e_{ij} a matriz Jacobiana $\tilde{J}_{ij}$ é expressa da seguinte forma (GRISETTI et al., 2010b):

$$\tilde{J}_{ij} = \left.\frac{\partial e_{ij}(\tilde{x} \boxplus \Delta\tilde{x})}{\partial \Delta\tilde{x}}\right|_{\Delta\tilde{x}=0} \tag{2.75}$$

Na Equação 2.75 e_{ij} depende apenas de $\Delta\tilde{x}_i$ e $\Delta\tilde{x}_j$ e pode ser expandida conforme apresentado na Equação 2.61:

$$\tilde{J}_{ij} = \begin{bmatrix} 0 & \dots & 0 & \tilde{\mathbf{A}}_{i,j} & 0 & \dots & 0 & \tilde{\mathbf{B}}_{ij} & 0 \dots & 0 \end{bmatrix} \tag{2.76}$$

Onde a matriz $\tilde{\mathbf{A}}_{i,j}$ das derivadas parciais de e_{ij} em relação ao vértice i possui a seguinte característica no *manifold*:

$$\tilde{\mathbf{A}}_{i,j} = \left.\frac{\partial e_{ij}(\breve{x} \boxplus \Delta\tilde{x})}{\partial \Delta\tilde{x}_i}\right|_{\Delta\tilde{x}=0} \tag{2.77}$$

Analogamente a matriz $\tilde{\mathbf{A}}_{i,j}$, a matriz $\tilde{\mathbf{B}}_{i,j}$ das derivadas parciais de e_{ij} em relação ao vértice j possui a seguinte característica no *manifold*:

$$\tilde{\mathbf{B}}_{i,j} = \left.\frac{\partial e_{ij}(\breve{x} \boxplus \Delta\tilde{x})}{\partial \Delta\tilde{x}_j}\right|_{\Delta\tilde{x}=0} \tag{2.78}$$

Aplicando a regra da cadeia para as derivadas parciais e aproveitando a avaliação da matriz Jacobiana em $\Delta\tilde{x} = 0$, os blocos "não-nulos"são dados por:

$$\frac{\partial e_{ij}(\breve{x} \boxplus \Delta\tilde{x}_i)}{\partial \Delta\tilde{x}} = A_i M_i \tag{2.79}$$

$$\frac{\partial e_{ij}(\breve{x} \boxplus \Delta\tilde{x}_j)}{\partial \Delta\tilde{x}} = A_i M_j \tag{2.80}$$

Onde a componentes M_i e M_j são determinados pelas seguintes derivadas:

$$M_i = \left.\frac{\breve{x}_i \boxplus \Delta\tilde{x}_i)}{\partial \Delta\tilde{x}_i}\right|_{\Delta\tilde{x}=0} \tag{2.81}$$

$$M_j = \left. \frac{\breve{x}_j \boxplus \Delta\tilde{x_j})}{\partial \Delta\tilde{x}_j} \right|_{\Delta\tilde{x}=0} \tag{2.82}$$

Por conseguinte, pode-se derivar a partir da Jacobiana não definida do *Manifold* da Equação2.61. Um Jacobiano em um *manifold* apenas multiplicando seus blocos *non-zero* com a derivada $\boxplus$ operador calculado em $\breve{x}_i$ e $\breve{x}_j$.

Com uma extensão direta da notação, pode-se inserir a Equação 2.73 na 2.63, isto conduz à seguinte sistema linear:

$$\tilde{H}\Delta\tilde{x}^* = -\tilde{b} \tag{2.83}$$

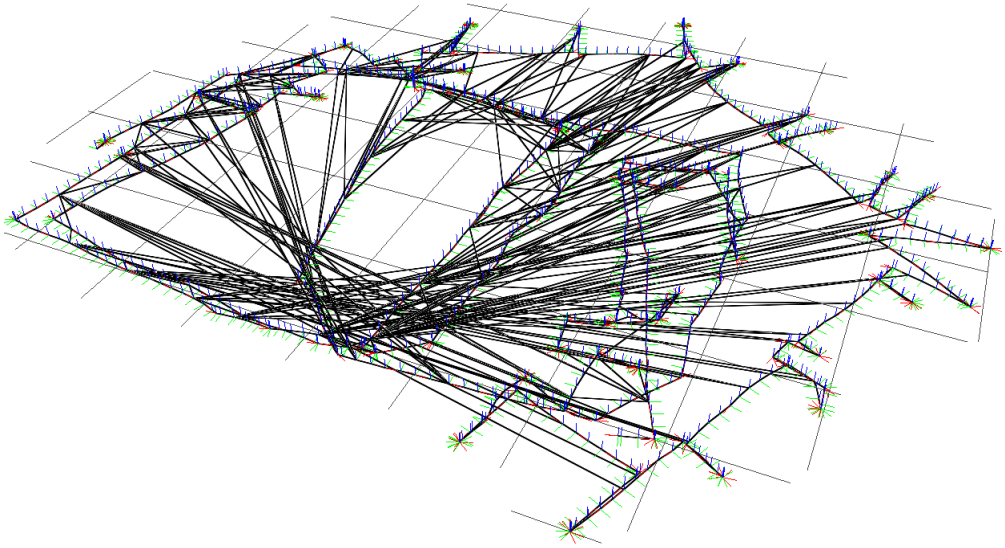
Como os incrementos $\Delta\tilde{x}$ são calculados em um espaço Euclidiano em torno da estimativa inicial $\breve{x}$, é necessário convertê-los, novamente, em um espaço sub-parametrizado mediante o operador $\boxplus$. Sendo assim, cada atualização da Equação2.71 é dada como (GRISETTI et al., 2010b):

$$x^* = \breve{x} \boxplus \Delta\tilde{x} \tag{2.84}$$

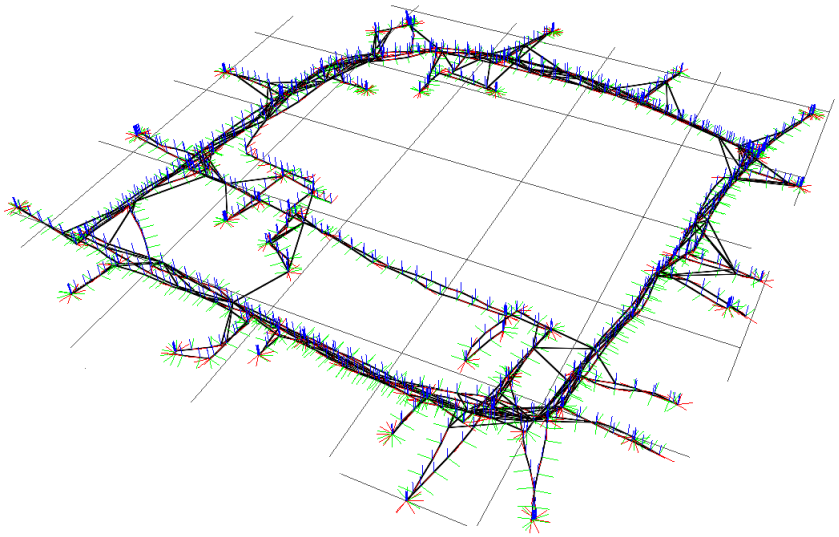
Portanto, a formalização do problema de minimização aplicado ao conceito de *manifold* consiste em determinar, primeiramente, uma aproximação Euclidiana em torno de uma estimativa inicial através da Equação 2.83 e acumular os incrementos em um espaço global não Euclidiano mediante a Equação2.84.

Para ilustrar o exemplo da aplicação do algoritmo de otimização baseado em do grafo, e também para testar o módulo "otimiza grafo"do sistema desenvolvido neste trabalho (ver anexo **??**), foi utilizado um arquivo formato `.graph` disponível em (CARLONE, 2015). Este arquivo contém as informações da trajetória 2D de um robô equipado com laser, mapeando salas e corredores do laboratório da Intel, estes dados estão armazenados na estrutura de um grafo, onde os vértices armazenam as transformações consecutivas (x) e as arestas as restrições espaciais (z) e a matriz informação (Ω). Figura 2.19 (a) mostra 1228 vértices representando o deslocamento do sensor no espaço 2D, as 1483 arestas realizando as restrições espaciais entre os vértices,nota-se que o erro se propaga em relação a todas as posições anteriores, originando uma trajetória inconsistente.

No entanto, na 2.19 (b) apresenta o grafo otimizado, as arestas contém apenas o erro de medição entre duas posições, observa-se os vértices se deslocando para que as



(a)



(b)

FIGURA 2.19 – TRAJETÓRIA DE UM ROBO MAPENDO O LABORATÓRIO INTEL, A) TRAJETÓRIA ESTIMADA, B) TRAJETÓRIA OTIMIZADA
FONTE: disponível em <http://www.lucacarlone.com/index.php/resources/datasets>

restrições impostas pelas arestas sejam atendidas e o problema de acúmulo de erro seja atenuado.

A seguir serão apresentados os materiais e o método proposto para execução deste trabalho de pesquisa.

3 MATERIAIS E MÉTODOS

No capítulo anterior foram apresentados as fundamentações teórica das etapas necessárias para a modelagem 3D de ambientes internos utilizando sensores RGB-D. No qual consistiu em apresentar: as principais características dos sensores RGB-D, em especial o Kinect; a formulação matemática do cálculo das coordenadas 3D de cada pixel da imagem fornecido pelo dispositivo a partir dos seus dados de paralaxe; a calibração da câmera IR e RGB do sensor dispositivo, como também a determinação dos parâmetros de montagem entre estes sensores (boresigth e level-arm); a extração das características visuais da imagem como pontos homólogos e remoção dos outliers, o registro entre os pares de nuvens consecutivas, e finalmente a etapa de fechamento de loop e otimização do grafo de poses.

Neste capítulo portanto, será apresentado a metodologia para o desenvolvimento do sistema de mapeamento 3D. A inovação proposta por esta metodologia, em relação ao estado da arte é o desenvolvimento de um modelo matemático que realiza o refinamento dos parâmetros obtidos ponto a ponto. Este modelo baseado em paralaxe-plano consiste primeiramente em extrair os planos da imagem RGB, utilizando o segmentador SLIC, a partir das relações 3D de cada segmento com os respectivos pontos é determinado os vetores normais de cada plano, e a partir da relação entre o as normais do plano, como o centroide de cada segmento no espaço imagem é realizado o refinamento destes parâmetros de transformação rígida.

3.1 MATERIAIS

Os recursos de software e hardware empregados para a realização deste trabalho serão descritos a seguir.

3.1.1 Recursos de hardware

- Laptop Samsung ; Processador Intel Core i7; Placa de Video Nvidia Optmus; 8 mega de memória RAM; Sistema Operacional LINUX / Ubuntu 14.04 e ambiente de programação QT Creator com linguagem C++;
- Sensor RGB-D Kinect adquirido com recursos de bolsa de bancada processo n^o 400400/2013-8 linha 2, Bolsa Pesquisador Visitante Especial (PVE);
- um interferômetro a LASER da marca Hewlett Packard modelo 5508A.

3.1.2 Recursos de software

Para o desenvolvimento deste projeto foram empregadas diversas bibliotecas computacionais *open-source*, cada uma com sua funcionalidade específica. Esta seção descreve brevemente a funcionalidade de cada uma delas.

- Software RGB-D Demo(BURRUS, 2014) fará a captura das sequencia de imagens RGB do Kinect, como também os dados de paralaxe;
- OpenCV (*Open-Soure Computer Vision*): biblioteca escrita em C/C++ (BRADSKI,), multiplataforma e open-source, distribuída sob licença BSD (*Berkeley Software Distribution*) . Esta biblioteca inclui algoritmos de processamento de imagens e visão computacional;
- PCL (*Point Cloud Library*): biblioteca desenvolvida em C++ (RUSU; COUSINS, 2011), *open-source*, distribuído sob licença BSD. Esta biblioteca inclui algoritmos para processamento de nuvens de pontos n- dimensionais;
- MRTP (*Mobile Robot Programming Toolkit*): é uma biblioteca open-source multiplataforma desenvolvida em C++ (BLANCO, 2009), distribuído como licença GPL (*General Public License*). Esta biblioteca para aplicações de SLAM, com recursos de visão computacional, calibração de câmeras, e otimização de grafos;
- Eigen: biblioteca escrita em C++, open-source (GUENNEBAUD; JACOB et al., 2010), distribuída sob licença LGPL (*Lesser General Public License*), inclui ferramentas para álgebra linear como operações matriciais e vetoriais, bem como resolução de sistemas lineares;

- Boost: Conjunto de bibliotecas desenvolvida em C++ (SIEK; LEE; LUMSDAINE, 2000), multiplataforma e *open-source*, distribuída sob-licença BSD. Esta biblioteca possui diversas rotinas de álgebra linear, cálculo matricial, ponteiros inteligentes, processamento de imagens, operação com grafos, etc; e

- VTK(*Visualização Tookit*): biblioteca desenvolvida em C++ (SCHROEDER; MARTIN; LORENSEN, 2003), baseado em OpenGL, multiplataforma e open-source, distribuída sob licença BSD. Esta biblioteca possui diversos algoritmos para geração e visualização de dados tridimensionais, etc.

- aplicativo *Mashlab* desenvolvido por Cignoni, Corsini e Ranzuglia (2008) que contém ferramentas para medidas de distância na nuvem de pontos;

3.2 MÉTODO

Nesta subseção do trabalho é apresentado a metodologia proposta para modelagem 3D de ambientes internos usando dados RGB-D. O sistema computacional desenvolvido esta dividido em 3 Módulos principais (Figura 3.1).

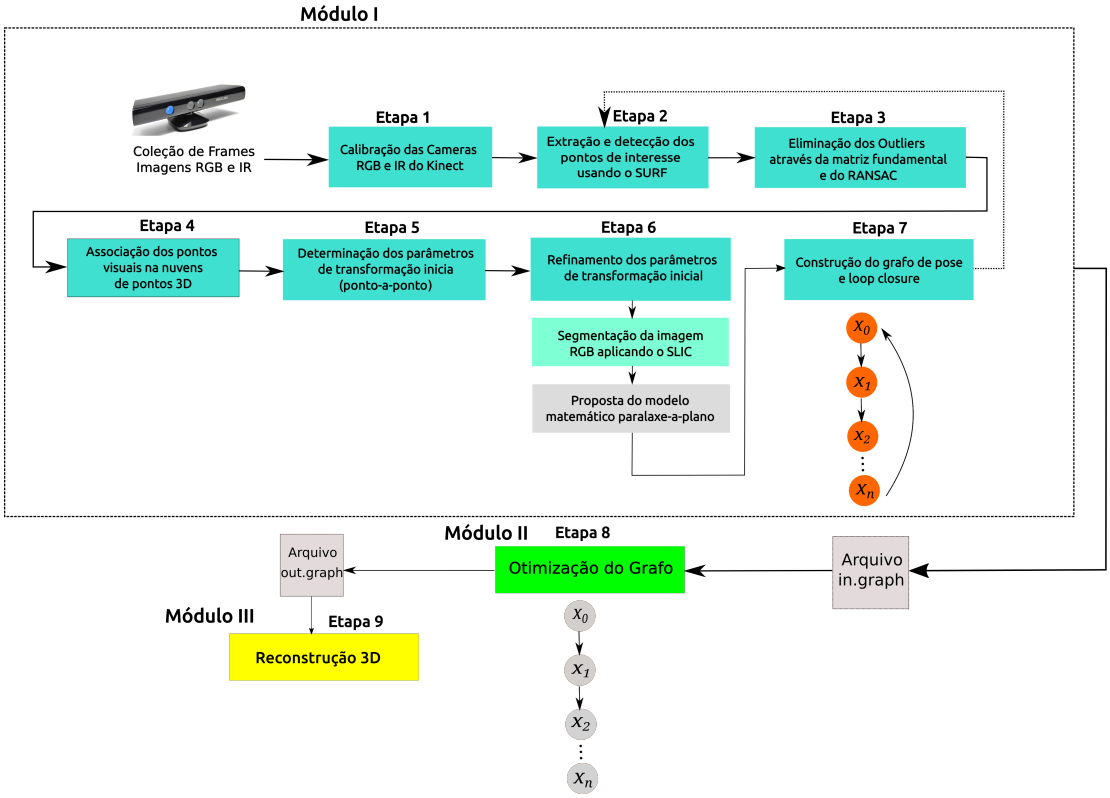


FIGURA 3.1 – PIPELINE DA METODOLOGIA PROPOSTA
FONTE: O Autor (2015)

Conforme a figura supracitada , os Módulos desenvolvidos realizam as seguintes tarefas:

- Módulo I;
  - realizar a leitura dos dados capturados pelo aplicativo *RGB-Demo*, armazenados em disco, imagem RGB, IR e os dados de paralaxe, como também os parâmetros POI do Kinect;
  - detecção dos pontos de interesse pelo SURF e a eliminação dos *outliers* por meio do RANSAC e da matriz fundamental;
  - Associação dos pontos visuais 2D com a nuvem de pontos 3D;
  - determinação dos parâmetros de transformação ponto-a-ponto (corpo rígido);
  - aplicação do modelo matemático proposto baseado em paralaxe plano;
  - construção do grafo de pose;
- Módulo II
  - otimização do grafo de poses;
- Módulo III
  - reconstrução do ambiente imageado, a partir dos parâmetros ajustados da etapa de otimização do grafo;

Ao todo foram realizadas 9 etapas que serão descrita a seguir.

3.2.1 Etapa 1: Calibração do sensor Kinect

A primeira etapa do desenvolvimento do projeto consiste em determinar os POI dos sensores RGB e IR do Kinect, bem como os parâmetros de montagem do sistema a partir dos POE, conforme apresentado na seção 2.2.2. Foi utilizado o aplicativo *Kinect Calibratio*n da biblioteca MRPT, sua utilização é composta pelos seguintes passo:

1. obtenção de um conjunto de imagens RGB e IR do padrão de calibração coplanar com diferentes posições e orientações do dispositivo Kinect;
2. extração dos vértices dos quadrados que constituem o padrão de calibração em todas as imagens adquiridas;

3. cálculo dos POI, POE, como também dos parâmetros de montagem entre os sensores IR e RGB;

3.2.2 Etapa 2: Extração e detecção dos pontos de interesse utilizando o SURF

Nas seções 2.3.1 e 2.3.2 foram apresentadas a fundamentação teórica do detector e descritor SURF. Para a sua implementação foi utilizado a classe *FeatureDetector* da biblioteca OpenCV. Esta classe têm invólucros com uma interface comum que permite alternar facilmente entre diferentes algoritmos para resolver o mesmo problema. O SURF é dividido em duas partes a primeira implementação do detector, a segunda é parte é o descritor (ver Figura 3.2).

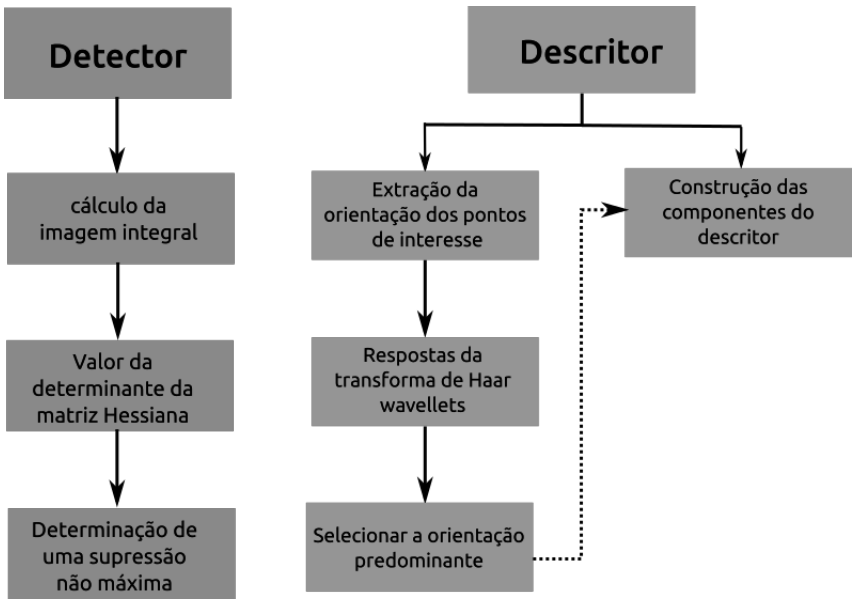


FIGURA 3.2 – ETAPAS DE IMPLEMENTAÇÃO DO SURF
FONTE: O Autor (2015)

Conforme a figura supracitada, na etapa de detecção o primeiro passo é criar a imagem integral e isso é feito usando uma rotina do OpenCV embutido. O próximo passo determinar o valor da Hessiana, para isso é necessário calcular a Hessiana para toda imagem integral em diferentes escalas. O valor do determinante é utilizado para classificar o máximo e o mínimo da função mediante o teste da segunda derivada. Para detecção dos pontos de interesse mediante o determinante Hessiana é necessário aplicar o conceito de espaço-escala. Finalmente o ultimo passo é aplicar uma supressão não máxima nesses determinantes em uma vizinhança $3x3x3$ para encontrar os melhores pontos de interesse.

Na etapa do descritor SURF, o algoritmo é dividido em duas tarefas na primeira

é a extração da informação de orientação dos pontos de interesse; a outra tarefa é a construção das componentes do descritor. Para a tarefa da informação da orientação dos pontos de interesse é necessário determinar a resposta da Haar *wavelets*, isso consiste através da resposta da transformada de Haar nas direções x e y. No passo da seleção da resposta predominante, um vetor de pontos correspondentes com as direções do vetor gradiente é avaliado, o vetor resultante de maior módulo representa a orientação resultante do ponto de interesse. Para a tarefa de construção das componentes do descritor é necessário construir uma janela retangular em torno do ponto de interesse já detectado e orientado; e finalmente, dividir cada janela em sub- regiões retangulares 4x4 para obter a transformada de Haar de cada sub-região.

Um parâmetro importante no SURF é o *minHessian*, este parâmetro define um limiar para o valor da determinante Hessiana, caso este valor seja muito alto , o detector SURF irá detectar poucos pontos de interesse mas robustos, caso contrário o *minHessian* baixo irá detectar muitos pontos de interesse, mas pouco robustos. Para este trabalho foi usado um valor de *minHessian* de valor 100.

3.2.3 Etapa 3: Eliminação dos outliers

Após serem encontradas as correspondências visuais nas imagens, para cada par de imagens RGB é efetuado a detecção e remoção dos *outliers*. Esta etapa é realizado em duas fases. A primeira fase consiste na geração da geometria epipolar com o cálculo da Matriz Fundamental F através da associação entre os pontos detectados (ver Equação 2.35) com o RANSAC e o MMQ. Para solucionar $f_{i,j}$ é aplicado o MMQ. Entretanto, como há muitas correspondências entre pontos visuais, é necessário verificar quais devem ser usadas para gerar a melhor Matriz Fundamental. Assim, são geradas diversas Matrizes Fundamentais, a partir de diferentes permutações entre pares de correspondências até que se alcance uma solução ótima . Como esse processo tem um alto custo computacional, o RANSAC foi empregado, conforme explicado na seção 2.4 o RANSAC gera modelos a partir de um conjunto mínimo de dados. A partir destes modelos e utilizando o critério de parada com 30% de pontos remanescentes (*inliers*), ou seja, na iteração que se alcança 30 ou menos do número total de correspondências como inliers, é obtido o modelo mais representativo entre os avaliados.

A segunda fase consiste na remoção de falsas correspondências usando um limiar de distância perpendicular da reta epipolar e o ponto correspondente na imagem de pesquisa.

3.2.4 Etapa 4: Associação de pontos visuais na nuvem de pontos 3D

Como os *inliers* detectados estão no sistema referencial digital, cuja origem é o canto superior esquerdo da imagem, deve-se realizar uma transformação para o sistema referencial com origem no centro da fotografia e, posteriormente, corrigir os efeitos sistemáticos provocados pelas distorções das lentes e transformar as coordenadas para o sistema referencial fotogramétrico. Essas tarefas são realizadas da seguinte forma, a saber:

$$\begin{bmatrix} x_f \\ y_f \end{bmatrix} = \begin{bmatrix} tp_l & 0 \\ 0 & tp_c \end{bmatrix} \begin{bmatrix} u - \frac{NC-1}{2} \\ v - \frac{NL-1}{2} \end{bmatrix} - \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} + \begin{bmatrix} \delta_{rx} \\ \delta_{ry} \end{bmatrix} + \begin{bmatrix} \delta_{dx} \\ \delta_{dy} \end{bmatrix} \tag{3.1}$$

Onde:
x_f, y_f : coordenadas no sistema fotogramétrico;
tp_l, tp_c : tamanho do pixel em coluna e linha;
u, v : coordenadas do pixel em coluna e linha no referencial imagem;
x_0, y_0 : deslocamento do ponto principal no referencial fotogramétrico
δ_{rx}, δ_{ry} : coeficientes de distorção radial simétrica ;
δ_{dx}, δ_{ry} : coeficientes de distorção descentrada do sistema de lentes das câmera;

Cada par de pontos de *inliers* correspondentes na imagem RGB deve ser associado ao seu ponto correspondente 3D na nuvem. Esta tarefa pode ser realizada a partir dos valores dos POI calibrados dos sensores RGB e IR, bem como dos parâmetros de montagem do dispositivo. Como ocorre um desalinhamento angular entre as imagens RGB e IR, por isso, é proposto uma normalização da imagem RGB. Tendo em vista que, a falta de correção dos desalinhamentos existentes entre os sensores RGB e IR provoca incertezas na associação direta dos pontos correspondentes da imagem RGB com a imagem IR e, consequentemente, na determinação dos parâmetros de transformação. A normalização das imagens RGB consiste em 5 tarefas:

1. A primeira tarefa consiste em retificar a imagem RGB, através da Equação 3.2 e 3.3, conforme Mikhail, Bethel e McGlone (2001):

$$x_n = -f_{IR} \frac{r_{11}x_f + r_{12}y_f - r_{13}f_{RGB}}{r_{13}x_f + r_{32}y_f - r_{33}f_{RGB}} \tag{3.2}$$

$$y_n = -f_{IR}\frac{r_{21}x_f + r_{22}y_f - r_{23}f_{RGB}}{r_{13}x_f + r_{32}y_f - r_{33}f_{RGB}} \tag{3.3}$$

Onde:

x_n e x_n: as coordenadas normalizadas no sistema referencial com origem no centro da imagem IR;

x_f e x_f: as coordenadas no espaço-imagem corrigidas da distorções radial simétrica e descentrada do sistema de lentes;

f_{RGB}: distância focal calibrada da câmera RGB;

f_{IR}: distância focal calibrada da câmera IR;

r_{ij}: os elementos da matriz de rotação composta pelos parâmetros angulares da montagem do dispositivo $\Delta\omega$,$\Delta\varphi$ e $\Delta\kappa$;

2. Apesar de a imagem RGB estar alinhada com a imagem IR a diferença entre as origens dos sistemas referenciais não está compensada. Para tratar este problema uma linha epipolar é projetada na imagem IR. A projeção da linha é feita em função do valor de profundidade máximo e mínimo de alcance do dispositivo que corresponde $Z_{min} = 0,40m$ e $Z_{min} = 7,0m$;

3. Para todos os pixels da imagem IR que estejam dentro do intervalo definido pela linha epipolar serão calculadas suas coordenadas 3D e reprojetados para o sistema referencial digital da imagem RGB através da Equações 3.4 e 3.5, como segue:

$$u_n = round\left(f_{IR}\left(\frac{X_k}{Z_K}\right) - \left(\frac{x_p}{tp_c}\right)\right) \tag{3.4}$$

$$v_n = round\left(f_{IR}\left(\frac{Y_k}{Z_K}\right) - \left(\frac{y_p}{tp_l}\right)\right) \tag{3.5}$$

Onde:

u_n e v_n: as coordenadas dos pixels reprojetados para a imagem IR normalizada;

x_p e y_p : as coordenadas no sistema referencial do centro da imagem IR adicionadas dos erros sistemáticos (ponto principal e distorções das lentes da câmera IR);

X_k,Y_k e Z_k: Coordenadas 3D do pixel associado na nuvem de ponto, conforme apresentado nas Equações 2.6, 2.7 e 2.5;

4. Verifica-se portanto, se o pixel transformado na tarefa 3 corresponde ao pixel extraído pelo detector/descritor SURF na imagem RGB;

5. Para que suas coordenadas do ponto 3D correspondente sejam associadas ao ponto detectado na imagem RGB é necessário que a diferença entre eles sejam abaixo de um limiar;

6. Repetem-se todas as tarefas para todos os pontos detectados na imagem RGB pelo SURF;

Finalmente, é obtido um conjunto de pontos correspondentes associados à nuvem de pontos 3D. Desta forma, dado um conjunto de pontos 3D correspondentes, será aplicado um modelo matemático para a determinação dos parâmetros de transformação rígida para cada par de nuvens de pontos.

3.2.5 Etapa 5: Determinação dos parâmetros de transformação iniciais

Considerando que foram detectados *n inliers* na imagem RGB de referência e de pesquisa, fazendo a associação com os pontos 3D na nuvem de pontos, de referência e de pesquisa, respectivamente, pode-se aplicar o modelo funcional de corpo rígido 3D (ver Equação 2.43) e o MMQ para determinar os parâmetros de transformação inicial, como segue:

$$\begin{aligned}
Xk_1^{pesq} &= Xk_1^{ref} r_{11} + Yk_1^{ref} r_{12} + Zk_1^{ref} r_{13} + t_x \\
Yk_1^{pesq} &= Xk_1^{ref} r_{21} + Yk_1^{ref} r_{22} + Zk_1^{ref} r_{23} + t_y \\
Zk_1^{pesq} &= Xk_1^{ref} r_{31} + Yk_1^{ref} r_{32} + Zk_1^{ref} r_{33} + t_z \\
&\vdots \\
Xk_k^{pesq} &= Xk_k^{ref} r_{11} + Yk_k^{ref} r_{12} + Zk_k^{ref} r_{13} + t_x \\
Yk_k^{pesq} &= Xk_k^{ref} r_{21} + Yk_k^{ref} r_{22} + Zk_k^{ref} r_{23} + t_y \\
Zk_k^{pesq} &= Xk_k^{ref} r_{31} + Yk_k^{ref} r_{32} + Zk_k^{ref} r_{33} + t_z
\end{aligned} \tag{3.6}$$

Note que a Equação 3.6 envolve 6 parâmetros $X = \begin{bmatrix} t_x & t_y & t_z & \omega & \varphi & \kappa \end{bmatrix}^T$. Portanto cada ponto fornece 3 equações e sendo a solução do sistema não linear é requerido no mínimo 2 pontos (6 observações) para obter solução única . A solução desta equação é feita através do modelo paramétrico não linear do MMQ. O vetor é dado por:

$$Lb = \begin{bmatrix} Xk_1^{pesq} & Yk_1^{pesq} & Zk_1^{pesq} & \dots & Xk_k^{pesq} & Yk_k^{pesq} & Zk_k^{pesq} \end{bmatrix}^T \tag{3.7}$$

O vetor X^0 dos parâmetros iniciais é inicializado como zero para todos os pa-

râmetros, a matriz A das derivadas parciais em relação a os parâmetros, e dodo por:

$$A = \left.\frac{\partial F(X)}{\partial X}\right|_{X=X_0} = \begin{bmatrix} \frac{\partial(X_1^{ref})}{\partial t_x} & \frac{\partial(X_1^{ref})}{\partial t_x} & \frac{\partial(X_1^{ref})}{\partial t_x} & \frac{\partial(X_1^{ref})}{\partial \omega} & \frac{\partial(X_1^{ref})}{\partial \varphi} & \frac{\partial(X_1^{ref})}{\partial \kappa} \\ \frac{\partial(Y_1^{ref})}{\partial t_x} & \frac{\partial(Y_1^{ref})}{\partial t_x} & \frac{\partial(Y_1^{ref})}{\partial t_x} & \frac{\partial(Y_1^{ref})}{\partial \omega} & \frac{\partial(Y_1^{ref})}{\partial \varphi} & \frac{\partial(Y_1^{ref})}{\partial \kappa} \\ \frac{\partial(Z_1^{ref})}{\partial t_x} & \frac{\partial(Z_1^{ref})}{\partial t_x} & \frac{\partial(Z_1^{ref})}{\partial t_x} & \frac{\partial(Z_1^{ref})}{\partial \omega} & \frac{\partial(Z_1^{ref})}{\partial \varphi} & \frac{\partial(Z_1^{ref})}{\partial \kappa} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{\partial(X_k^{ref})}{\partial t_x} & \frac{\partial(X_k^{ref})}{\partial t_x} & \frac{\partial(X_k^{ref})}{\partial t_x} & \frac{\partial(X_k^{ref})}{\partial \omega} & \frac{\partial(X_k^{ref})}{\partial \varphi} & \frac{\partial(X_k^{ref})}{\partial \kappa} \\ \frac{\partial(Y_k^{ref})}{\partial t_x} & \frac{\partial(Y_k^{ref})}{\partial t_x} & \frac{\partial(X_Y^{ref})}{\partial t_x} & \frac{\partial(Y_k^{ref})}{\partial \omega} & \frac{\partial(Y_k^{ref})}{\partial \varphi} & \frac{\partial(Y_k^{ref})}{\partial \kappa} \\ \frac{\partial(Z_k^{ref})}{\partial t_x} & \frac{\partial(Z_k^{ref})}{\partial t_x} & \frac{\partial(Z_Y^{ref})}{\partial t_x} & \frac{\partial(Z_k^{ref})}{\partial \omega} & \frac{\partial(Z_k^{ref})}{\partial \varphi} & \frac{\partial(Z_k^{ref})}{\partial \kappa} \end{bmatrix} \tag{3.8}$$

A solução dos parâmetros através do método paramétrico é aplicando a Equação 2.45 até a 2.47, conforme apresentado na seção **??**

3.2.6 Etapa 6: Refinamento dos parâmetros de transformação iniciais

Para refinar os parâmetros de transformação inicial é proposto um modelo matemático baseado em uma abordagem *paralaxe-a-plano*. Basicamente esta etapa do método é dividida em quatro partes:

1. extrair um conjunto de segmentos nos pares de imagens RGB, calcular os centroides e encontrar suas correspondências;

2. fazer a associação do conjunto de centroides da imagem de pesquisa com os valores de paralaxe propiciados pelo dispositivo Kinect e associar o conjunto de pontos de cada segmento, extraído na imagem de referência, com seus correspondentes na nuvem de pontos 3D (de referência);

3. ajustar, pelo MMQ, os segmentos planos da nuvem de pontos de referência e determinar seus parâmetros

4. aplicar o modelo proposto e refinar os parâmetros de transformação inicial.

A *primeira parte* desta etapa do método é feita empregando o segmentador SLIC conforme apresentado no subitem 2.5. Este segmentador irá decompor a imagem em um número limitado de regiões, sendo definido o número de segmentos através do parâmetro K. Os pixeis sementes são distribuídos de maneira regular na imagem, a partir da distancia radiométrica entre o pixel semente e seus pixeis vizinhos. Caso a distancia entre

o pixel semente e o pixel candidato for menor que um limiar pré-definido, este pixel será agregado a outros de mesma similaridade. Então, é criada uma matriz com as mesmas dimensões da imagem ($640x480$), no qual são armazenados os rótulos de cada pixel. A correspondência entre os segmentos planos de cada par de imagens RGB consecutivos é realizado a partir do respectivo centroide de cada segmento. A correspondência entre os planos é realizada a partir da menor distância entre os centroides.

A *segunda parte* consiste em fazer a associação dos pontos de cada segmento com a nuvem de pontos 3D correspondente (conforme mostrado na etapa 4). Na imagem de referência, os pixeis rotulados de cada segmento são associados aos seus respectivos pontos na nuvem de pontos 3D, de tal forma que, cada região segmentada na imagem seja associada ao seu respectivo plano na nuvem de pontos.

A *terceira parte* consiste em determinar os coeficientes do plano (n_x, n_y, n_z e ρ) possam ser determinados pelo MMQ. Já na imagem de pesquisa, os centroides de cada segmento são associados aos seus valores de paralaxe no mapa de paralaxe, de tal forma que, cada centroide terá como informação as coordenadas no sistema referencial fotogramétrico e um valor de paralaxe .

Na Figura 3.3 para cada ponto de um segmento na imagem RGB de referência é assumido que existe um ponto na imagem IR de referência $\mathbf{p}_{RGB}(x_f, y_f)^{Ref} \rightarrow \mathbf{p}_{IR}(x_f, y_f)^{Ref}$. Usando as Eq 2.4 e Eq 2.5 são calculadas as coordenadas de cada pontos. Consequentemente, é ajustado o segmento plano pelo MMQ e obtidos seus coeficientes. Na Figura 3.3 (b) para cada centroide de um segmento na imagem RGB de pesquisa é assumido que existe um ponto na imagem IR de pesquisa $\mathbf{p'}_{IR}(x_f, y_f)^{Pesq} \rightarrow \mathbf{p'}_{IR}(x_f, y_f)^{Pesq}$, cujos valores de paralaxe (d) são diretamente associados.

Neste trabalho é proposto um modelo matemático baseado em uma abordagem *paralaxe-a-plano* para refinar os parâmetros de transformação inicial determinados no processo anterior, descrito na etapa 4 (subitem 2.6). Sabe-se que a equação paramétrica do plano no espaço é dado por:

$$n_x X k^{pesq} + n_y Y k^{pesq} n_z Z k^{pesq} - \rho = 0 \tag{3.9}$$

E considerando que as coordenadas de um ponto na nuvem de pontos 3D é determinada pela Equação 3.6 e combinando com a 3.9, tem-se:

$$n_x \frac{Z_k}{f} x_f + n_y \frac{Z_k}{f} y_f + n_z \frac{1}{md' + n} - \rho = 0 \tag{3.10}$$

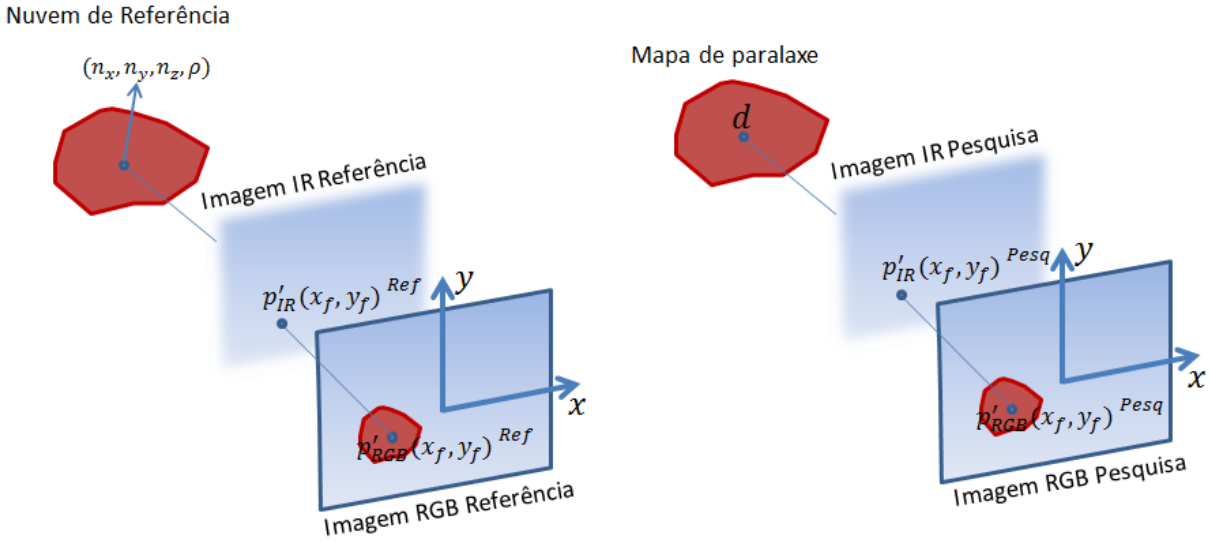


FIGURA 3.3 – A)CORRESPONDÊNCIA ENTRE OS SEGMENTOS NA IMAGEM DE REFERÊNCIA E NA NUVEM DE PONTOS 3D, B) CORRESPONDÊNCIA ENTRE OS CENTROIDES DA IMAGEM DE PESQUISA E O MAPA DE PARALAXE
FONTE: O Autor (2015)

Sabendo-se que as coordenadas homogêneas do ponto $\mathbf{P}$ é dado por:

$$\mathbf{P} = \begin{bmatrix} X_k \\ Y_k \\ Z_k \\ 1 \end{bmatrix} \therefore \mathbf{P} = \begin{bmatrix} \frac{Z_k}{f} x_f \\ \frac{Z_k}{f} y_f \\ \frac{1}{md'+n} \\ 1 \end{bmatrix} \tag{3.11}$$

Considerando que o ponto $\mathbf{P'}$, é o ponto resultante de uma translação e rotação:

$$\mathbf{P'} = R\mathbf{P} + t \therefore \begin{bmatrix} \mathbf{P'} \\ 0 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{23} & r_{23} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.12}$$

Combinando a Equação 3.10 e a 3.11 com o ponto $\mathbf{P}$', obtêm-se:

$$\begin{bmatrix} n_x & n_y & n_z & -\rho \end{bmatrix} \mathbf{P'} = 0 \tag{3.13}$$

Algebricamente obtêm-se a seguinte expressão:

$$\begin{gathered} \frac{Z_k}{f} x_f (n_x r_{11} + n_y r_{21} + n_z r_{31}) + \frac{Z_k}{f} y_f (n_x r_{21} + n_y r_{22} + n_z r_{23}) + \\ f(n_x r_{21} + n_y r_{22} + n_z r_{33} + f(m + nd')(n_x t_x - \rho + n_y t_y + n_z t_z) = 0 \end{gathered} \tag{3.14}$$

Matricialmente o modelo matemático proposto pode ser escrito da seguinte forma:

$$\begin{bmatrix} n_x & n_y & n_z & -\rho \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{23} & r_{23} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_f \\ y_f \\ f \\ f(n+md') \end{bmatrix} = 0 \tag{3.15}$$

Conforme resultou a Equação 3.15, caracterizando portanto, o modelo combinado ou implícito para a solução do MMQ. Segundo Gemael (1994) o modelo funcional do método combinado é dado por:

$$F(X_a, L_a) = 0 \tag{3.16}$$

Como a Equação 3.15 é não linear, é necessário realizar a expansão da série de Taylor através método iterativo para a solução dos parâmetros. A expansão da série de Taylor, resulta em (DALMOLIN, 2002):

$$F(X_a^i, L_a^i) = \left.\frac{\partial F}{X_a^i}\right|_{X_a^i = X_a^{i-1}} (X_a^i - X_a^{i-1}) + \left.\frac{\partial F}{L_a^i}\right|_{l_a^i = l_a^{i-1}} (L_a^i - L_a^{i-1}) + F(X_a^{i-1}, L_a^{i-1}) \tag{3.17}$$

como

$$L_a^i - L_a^{i-1} = L_a^i - L_b + L_b - L_A^{i-1} \tag{3.18}$$

podendo ser escrita na forma

$$A_i X_i + B_i V_i + B_i (L_b - La^{i-1}) + F(X_a^{i-1}, L_a i - 1) \tag{3.19}$$

resultando em:

$$A_i X_i + B_i V_i + W_i = 0 \tag{3.20}$$

Onde:
O vetor Lb:, é o vetor das observações, formado pelos seguintes elementos:

$$Lb = \begin{bmatrix} n_x & n_y & n_z & \rho & x_f & y_f & d' \end{bmatrix}^T \tag{3.21}$$

A matriz A, é a matriz das derivadas parciais em relação aos parâmetros:

$$\left.\frac{\partial F(X)}{\partial X}\right|_{X_a = X_0} \tag{3.22}$$

A matriz B matriz das derivadas parciais em relação as observações, sendo K é o número de *superpixels* extraídos da imagem.

$$\left.\frac{\partial F(X)}{\partial L_a}\right|_{L_a=L_b} = \begin{bmatrix} \frac{\partial F(X)}{\partial n_{x1}} & \frac{\partial F(X)}{\partial n_{y1}} & \frac{\partial F(X)}{\partial n_{z1}} & \frac{\partial F(X)}{\partial \rho_1} & \frac{\partial F(X)}{\partial x_{f1}} & \frac{\partial F(X)}{\partial y_{f1}} & \frac{\partial F(X)}{\partial d'_1} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{\partial F(X)}{\partial n_{xK}} & \frac{\partial F(X)}{\partial n_{yK}} & \frac{\partial F(X)}{\partial n_{zK}} & \frac{\partial F(X)}{\partial \rho_K} & \frac{\partial F(X)}{\partial x_{fK}} & \frac{\partial F(X)}{\partial y_{fK}} & \frac{\partial F(X)}{\partial d'_K} \end{bmatrix} \tag{3.23}$$

O vetor W, é o vetor do erro de fechamento, definido por:

$$W_i = B_i(L_b - L_a^i) + F(X_a^{i-1}, L_a^{i-1}) \tag{3.24}$$

O vetor V, é o vetor dos resíduos e calculado pela seguinte equação:

$$V - P^{-1}B^T K B_i P^{-1} B_i^T)^{-1} AX + W \tag{3.25}$$

O vetor X, é o vetor dos parâmetros aproximados é calculado por:

$$X = -(A_i^T B_i P^{-1} B_i^T)^{-1} A_i^T) A_i^T (B_i P^{-1} B_i^T) W_i \tag{3.26}$$

Na primeira iteração ($i = 1$) o ponto de expansão X_a^0e L_a^0 devem ser dados com uma boa aproximação dentro dos limites da natureza da não linearidade do problema considerado (DALMOLIN, 2002). Para isso foi empregado os parâmetros calculados no caso ponto-a-ponto como parâmetros aproximados $X_0 = \begin{bmatrix} t_x & t_y & t_z & \omega & \varphi & \kappa \end{bmatrix}^T$ O calculo do vetor dos parâmetros ajustados é realizado pela correção:

$$X_a = X + X_0 \tag{3.27}$$

O processamento vai se repetindo até que todos os elementos do vetor dos parâmetros aproximados X sejam menor que uma tolerância pré-definida. Como o vetor X_0 já fornecem uma boa aproximação, a convergência tende ser bastante rápida.

A estimativa de precisão dos parâmetros, é similar a Equação 2.6 do método paramétrico, dado pela equação:

$$\hat{\sigma}_0^2 = \frac{V^T P V}{n - u} \tag{3.28}$$

Finalmente, a *quarta parte* consiste no refinamento dos parâmetros grosseiros definido pela Equação 2.43, onde seus valores ajustados X_a, assume como parâmetros iniciais X_0 no modelo proposto *paralaxe-a-plano*.

A Figura 3.4 ilustra o processamento da estratégia adotada, a segmentação da imagem de referência e de pesquisa RGB, rotulação dos planos, o calculo dos parâmetros do plano n_x, n_y, n_z e ρ na imagem de referência,

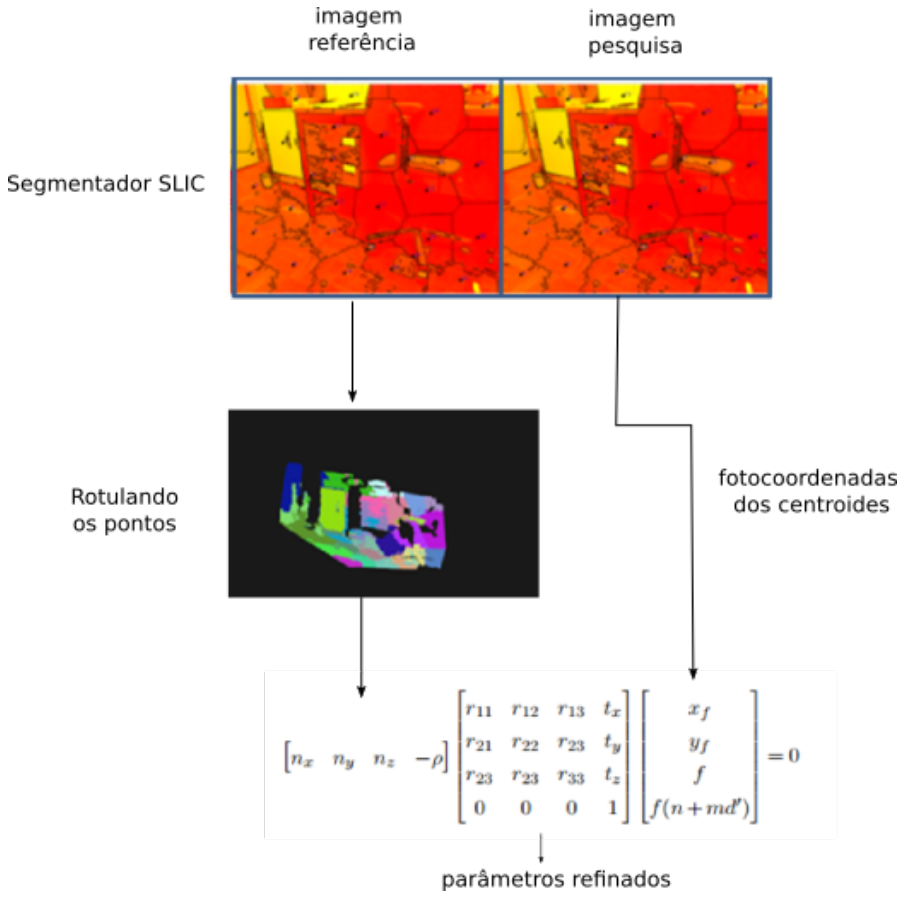


FIGURA 3.4 – REFINAMENTO DOS PARÂMETROS DE TRANSFORMAÇÃO A PARTIR DO MODELO PROPOSTO
FONTE: O Autor (2015)

3.2.7 Etapa 7: Construção do grafo

A detecção de lugares anteriormente visitados é feito, basicamente, pelo número de *inliers* entre os pares de imagens detectados pelo SURF e filtrados através do RANSAC. Caso o par de imagens contenham um numero de *inliers* acima de um limiar pré-determinado indica que esta área imageada está sendo revisitada e, portanto, uma nova aresta é adicionada ao grafo. A estratégia adotada para verificar se uma área já foi revisitada é empregar uma janela deslizante que percorrerá todos os vértices do grafo. Com isto, pretende-se diminui o esforço computacional, uma vez que a cada passo do algoritmo é verificado apenas um grupo de vértices. A Figura 3.5 ilustra o funcionamento da janela deslizante para verificar as áreas revisitadas. Basicamente, é selecionado um grupo de

7 vértices, onde o frame-chave P_5 deverá comparar a sua imagem em relação a quatro vértices candidatos P_0, P_1, P_2 e P_3. Caso o número de comparação for maior que um limiar pré-definido (l_b) uma nova aresta é adicionada ao grafo. Por exemplo, caso a imagem P_5, quando comparada com a imagem P_2, apresentar um número de *inliers* maior que um limiar, o vértice P_5 será ligado ao P_2, e através de um *buffer* que armazena as nuvens de pontos referentes a cada vértice, é realizado a transformação de corpo rígido entre a nuvem P_2 e P_5. Para aumentar a eficiência da buscas, um raio de busca com raio arbitrário acompanha o *frame-chave*, para verificar os vértices que estão inseridos dentro deste raio como *frames* candidatos.

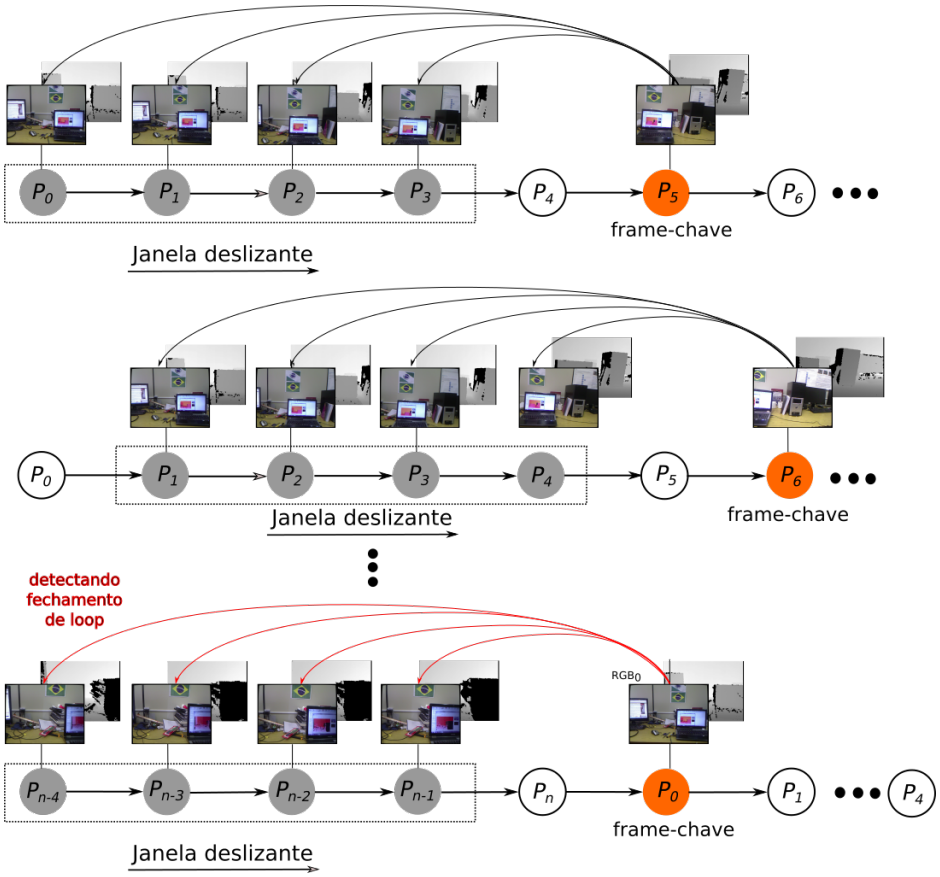


FIGURA 3.5 – JANELA DESLIZANTE PARA VERIFICAR ÁREAS REVISITADAS
FONTE: O Autor (2015)

A adição de novas arestas é dado por quatro parâmetros, são eles:

- O indexador do vértice j, referente ao *frame candidato*;
- O indexador do vértice i, referente ao *frame-chave*;
- A matriz T_i^j contendo os parâmetros da transformação relativa entre a nuvem referente ao *frame-candidato* com o *frame-chave*;

- A matriz informação, sendo neste trabalho adotada a inversa da matriz variância-covariância dos parâmetros de transformação;

Visitados este grupo de vértices, o próximo vértice e o novo *frame-chave* e a janela deslizante é deslocada para frente. Isto é feito de forma consecutiva até todos os vértices forem visitados. A Figura 3.6 mostra as arestas criadas e o comportamento da janela deslizante.

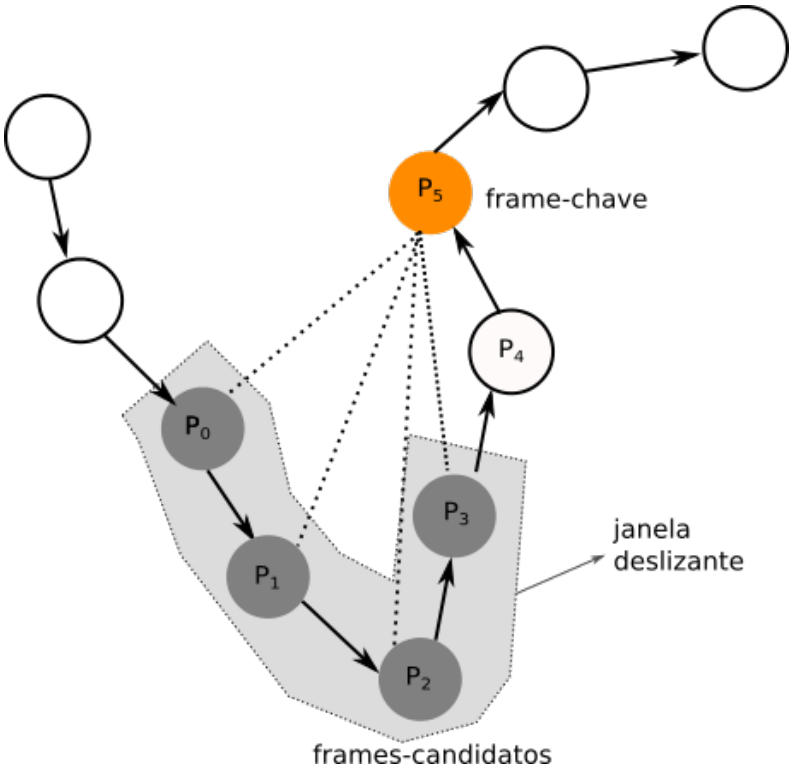


FIGURA 3.6 – JANELA DESLIZANTE PERCORRENDO O GRAFO
FONTE: O Autor (2015)

O arquivo será armazenado em um arquivo no formato `.graph`. Nesta estrutura de arquivo devem ser armazenados a informação dos vértices, contendo os dados dos parâmetros de transformações consecutivas do sensor, assim como as informações das arestas (parâmetros de transformação relativa e da matriz informação).

Conforme ilustra a Figura 3.7, o arquivo `graph` tem na primeira parte do arquivo referente aos dados dos vértices do grafo, denominado `VERTEX3` no campo id é armazenado o índice do vértice, no campo `tx`, `ty`, `tz` são os campos referentes as translações, o campo `omega`, `phi` e `kappa` são as informações de rotação. No segundo bloco do arquivo devem ser armazenadas as informações das arestas do grafo denominado de `EDGE3`, o campo `fron_id` é a informação do vértice i, `to_id` é referente o vértice j, os campos `tx_r`, `ty_r`, `tz_r`

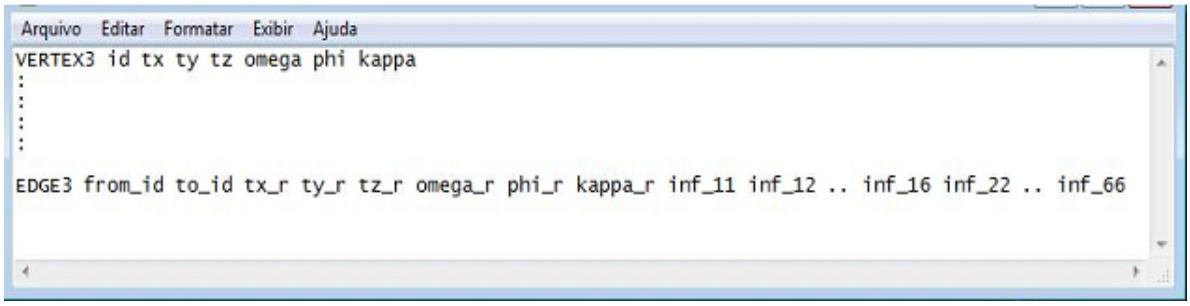


FIGURA 3.7 – ESTRUTURA DO ARQUIVO DO FORMATO `graph`
FONTE: O Autor (2015)

, `omega_r`,`phi_r`, `kappa_r` são os campos referentes as transformações relativas entre os vértices i e j, os campos `inf_11` até `inf_66` são os elementos da matriz informação Ω.

3.2.8 Etapa 8: Otimização do grafo

Nesta etapa do processo é realizado a otimização do grafo de pose, através da leitura do arquivo `in.graph`, aplicando a formulação matemática conforme apresentado na seção 2.7.2, é realizado a reconfiguração dos vértices do grafo, os dados de entrada são as seguintes:

- Os vértices do grafo,são as transformações consecutivas a partir de um vértice inicia, formado pelo vetor x:

$$x_{ij} = [tx, ty, tz, \omega, \varphi, \kappa] : Pose_j^i = \begin{bmatrix} r_{11} & r_{21} & r_{31} & t_x \\ r_{21} & r_{22} & r_{32} & t_y \\ r_{31} & r_{23} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.29}$$

- As arestas do que atuam como injunções é dado pelas transformações relativas entre os vértices, formado pelo vetor z_{ij}:

$$z_{ij} = [tx, ty, tz, \omega, \varphi, \kappa] : T_j^i = \begin{bmatrix} r_{11} & r_{21} & r_{31} & t_x \\ r_{21} & r_{22} & r_{32} & t_y \\ r_{31} & r_{23} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.30}$$

- A matriz informação que pondera a aresta do grafo é dado pela inversa da matriz variância-covariância, das transformações:

$$\Omega_{ij} = \Sigma_{Xa}^{-1} \tag{3.31}$$

A Figura 3.8 ilustra o fluxograma do processo, que retorna um arquivo `out.graph` contendo os novos valores para os vértices e da matriz informação, os valores das arestas se mantêm fixos.

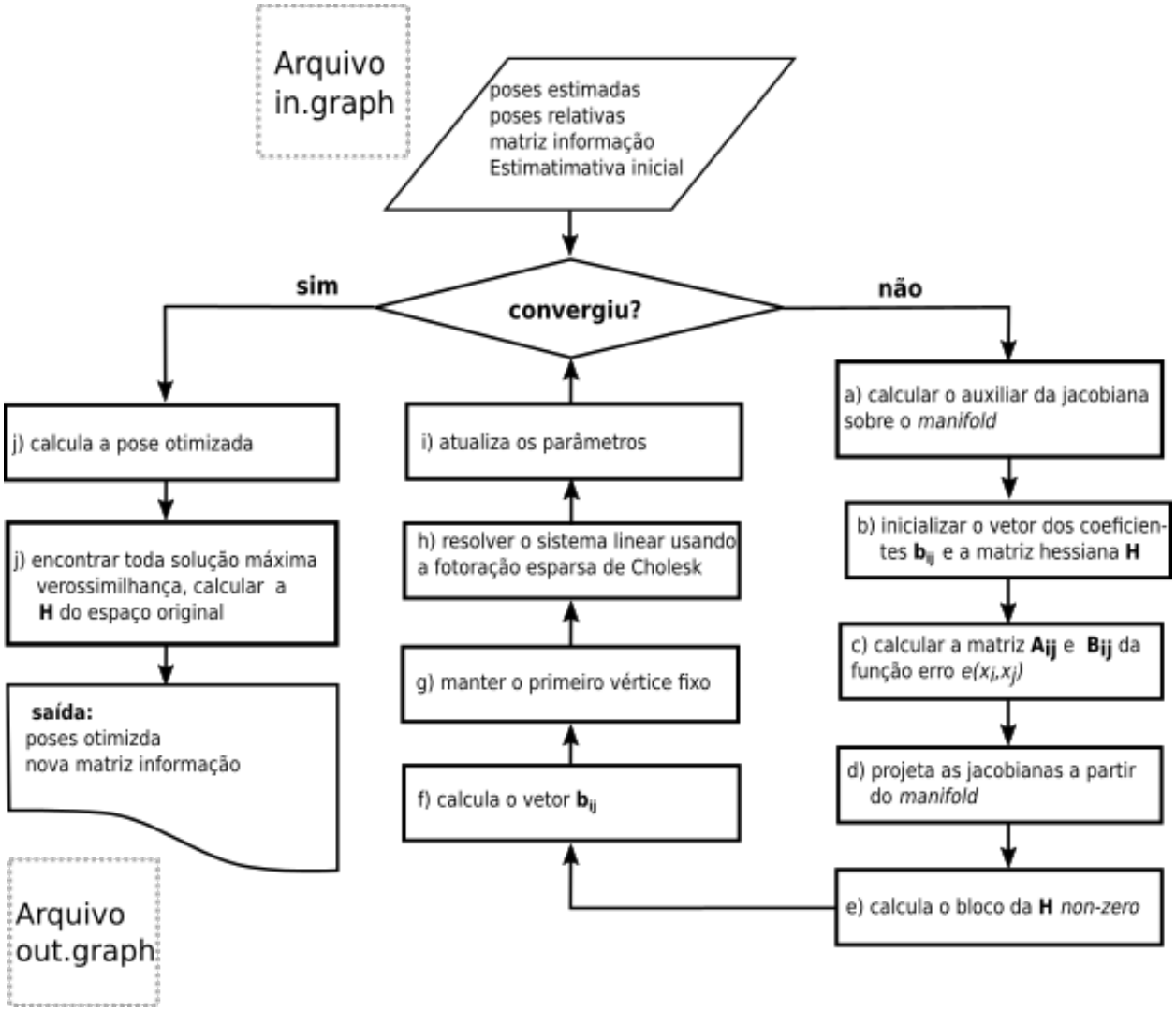


FIGURA 3.8 – FLUXOGRAMA DAS ETAPAS DE OTIMIZAÇÃO DO GRAFO
FONTE: O Autor (2015)

Conforme apresentado na figura supracitada, a otimização realiza as seguintes tarefas:

(a) calcula o auxiliar jacobiano sobre o *manifold*:

$$M_i = \frac{\breve{x}_i \boxplus \Delta\tilde{x}_i)}{\partial \Delta\tilde{x}_i}\bigg|_{\Delta\tilde{x}=0}$$

(b) Inicializa o vetor dos coeficientes b_{ij} e a matriz Hessiana:

$$\breve{b}_{ij} \leftarrow 0$$

$$\breve{H}_{ij} \leftarrow 0$$

(c) calcula as matrizes jacobianas $\mathbf{A}_{ij}$ e $\mathbf{B}_{ij}$ da função erro $e_{i,j}$:

$$\mathbf{A}_{ij} \leftarrow \frac{\partial e_{ij}(x)}{\partial x_i}$$

$$\mathbf{B}_{ij} \leftarrow \frac{\partial e_{ij}(x)}{\partial x_j}$$

(d) projeta as jacobianas a partir dos *manifolds*:

$$\breve{A}_{ij} \leftarrow \mathbf{A}_{ij} M_i$$

$$\breve{B}_{ij} \leftarrow \mathbf{B}_{ij} M_j$$

(e) calcula o bloco da Hessiana não nulo:

$$\breve{H}_{ii} \leftarrow \breve{H}ii + \breve{A}^T \mathbf{\Omega}_{ij} \breve{A}_{i,j}$$

$$\breve{H}_{ji} \leftarrow \breve{H}ji + \breve{B}^T \mathbf{\Omega}_{ij} \breve{A}_{i,j}$$

$$\breve{H}_{ij} \leftarrow \breve{H}ij + \breve{A}^T \mathbf{\Omega}_{ij} \breve{B}_{i,j}$$

$$\breve{H}_{jj} \leftarrow \breve{H}jj + \breve{B}^T \mathbf{\Omega}_{ij} \breve{B}_{i,j}$$

(f) calcula o vetor dos coeficientes:

$$\breve{b}_i \leftarrow \breve{b}_i + \breve{A}_{ij}^T \mathbf{\Omega}_{ij} e_{ij}$$

$$\breve{b}_j \leftarrow \breve{b}_j + \breve{B}_{ij}^T \mathbf{\Omega}_{ij} e_{ij}$$

(g) manter o primeiro vértice fixo:

$$H_{11} \leftarrow \breve{H}_{11} + I$$

(h) resolver o sistema linear usando a fatoração esparsa de *Cholesk*

$$\Delta\tilde{x} \leftarrow solve\left(\breve{H}\Delta\tilde{x} = -\breve{b}\right)$$

(i) atualiza os parâmetros de transformação:

$$\breve{x}_i \leftarrow \breve{x}_i \boxplus \Delta\tilde{x}_i$$

(j) calcula a solução otimizada

$$x^* \leftarrow \breve{x}$$

(k) encontra a solução da máxima verossimilhança:

$$H^* \leftarrow 0$$

Retorna a hessiana para o espaço original

$$H_{ii} \leftarrow Hii + \mathbf{A}^T \mathbf{\Omega}_{ij} \mathbf{A}_{i,j}$$

$$H_{ji} \leftarrow Hji + \mathbf{B}^T \mathbf{\Omega}_{ij} \mathbf{A}_{i,j}$$

$$H_{ij} \leftarrow Hij + \mathbf{A}^T \mathbf{\Omega}_{ij} \mathbf{B}_{i,j}$$

$$H_{jj} \leftarrow Hjj + \mathbf{B}^T \mathbf{\Omega}_{ij} \mathbf{B}_{i,j}$$

3.2.9 Etapa 9: Reconstrução do ambiente 3D

A ultima etapa, consiste simplesmente em ler as poses otimizadas do arquivo `out.graph` contendo os parâmetros de transformação ajustados. Aplicando a transformação inversa do corpo-rígido (Equação 3.32), obtêm-se portanto as coordenadas ajustadas.

$$\begin{bmatrix} X_{ka} \\ Y_{ka} \\ Z_{ka} \end{bmatrix} = R(\omega_a, \varphi_a, \kappa_a) \begin{bmatrix} X_k - tx_a \\ Y_k - ty_a \\ Z_k - tz_a \end{bmatrix} \tag{3.32}$$

A partir destas coordenadas X_{ka}, Y_{ka} e Z_{ka}, é realizado a reconstrução do ambiente imageado por meio das coordenadas X_k, Y_k e Z_k de cada nuvem de entrada.

4 EXPERIMENTOS E ANÁLISE DOS RESULTADOS

4.1 Calibração da Câmera RGB e IR

A calibração do sensor Kinect foi realizado utilizando as imagens da câmera IR e RGB. A determinação dos parâmetros de orientação interior foi feita usando o aplicativo Kinect *Stereo Calibration Wizard* da biblioteca MRPT. Para calibrar os sensores supracitados foram capturados um total de 35 pares de imagens IR e RGB, totalizando 70 imagens. Para modelar as distorções das lentes foi utilizado o método de Zhang (1999) conforme descrito no na seção 2.2.2. A Tabela 4.1 mostra os resultados obtidos no processo de calibração (considerando o tamanho do pixel $9,3\mu$).

TABELA 4.1 – PARÂMETROS DE ORIENTAÇÃO INTERIOR DA CÂMERA IR E RGB DO KINECT

POI	Câmera IR	Câmera RGB
f_x	5,3989mm $\pm 0,0003mm$	4,8831mm $\pm 0,0003mm$
f_y	5,3172mm $\pm 0,0004mm$	4,706mm $\pm 0,0004mm$
x_0	0,0552mm $\pm 7,4400\ 10^{-5}mm$	-0,0336mm $\pm 7,4400\ 10^{-5}mm$
y_0	0,1155mm $\pm 7,4400\ 10^{-5}mm$	-0,1586mm $\pm 7,4400\ 10^{-5}mm$
k_1	-3,6793 $10^{-2}mm^{-2}$	9,5719 $10^{-2}mm^{-2}$
k_2	4,2651 $10^{-2}mm^{-4}$	-6,1111 $10^{-2}mm^{-4}$
k_3	9,2290 $10^{-2}mm^{-6}$	-1,7587 $10^{-1}mm^{-6}$
p_1	-1,1228 $10^{-3}mm^{-1}$	-5,9230 $10^{-4}mm^{-1}$
p_2	1,1552 $10^{-3}mm^{-1}$	5,1245 $10^{-3}mm^{-1}$

FONTE: O Autor (2015)

Logo, na Tabela 4.2 são apresentados os parâmetros de montagem do sistema, composto pelo deslocamento e rotação dos eixos entre as câmeras IR e RGB do Kinect.

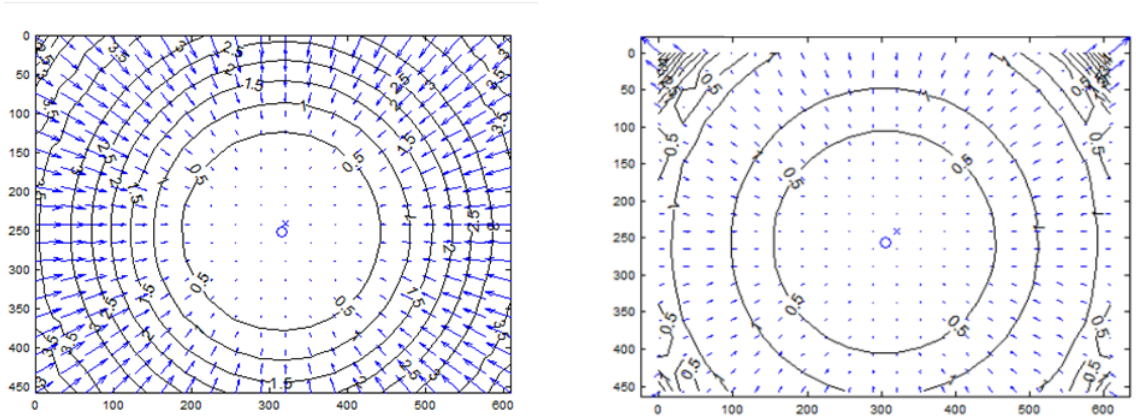
Conforme os valores apresentados na Tabela 4.1 há um grande deslocamento do ponto principal em relação ao eixo y tanto na imagem RGB quanto na imagem IR. Considerando o tamanho do pixel em $9,3\mu$ deslocamento apresentado na imagem RGB é de $0,1586mm$, o que equivale a um deslocamento de 17 pixels. Enquanto na imagem

TABELA 4.2 – PARÂMETROS DE DESLOCAMENTO E ROTAÇÃO DA CÂMERA IR EM RELAÇÃO A CÂMERA RGB DO KINECT

Δ_X	Δ_Y	Δ_Z	Δ_ω	Δ_φ	Δ_κ
$26,1448mm$	$0,3433mm$	$-2,1776mm$	$0,167131^o$	$0,307700^o$	$0,096684^o$

FONTE: O Autor (2015)

IR este deslocamento de $0,1155mm$ equivale a 12 pixels. Para o eixo x foram obtidos $0,0552mm$ na câmera IR o que equivale a 6 pixels, enquanto a câmera RGB $-0,0366mm$ que equivale a 4 pixels. Esse deslocamento provocado no eixo y é devido a redução da imagem IR, uma vez que a dimensão do seu sensor é de 1280×1024. No entanto, devido ao limite de banda da porta USB a imagem é redimensionada pela metade, ou seja, 640×480 compatível com a RGB. No eixo y isso se torna mais proeminente em razão de ocorrer a perda de 32 pixels ($\left(\frac{1024}{2}\right) - 480 = 32$). No caso da câmera RGB esse deslocamento esta dentro de padrões de câmeras classificadas como de baixo custo. A Figura 4.1 mostra o efeito combinado de distorção, tanto para o sensor IR, quanto para o sensor RGB.



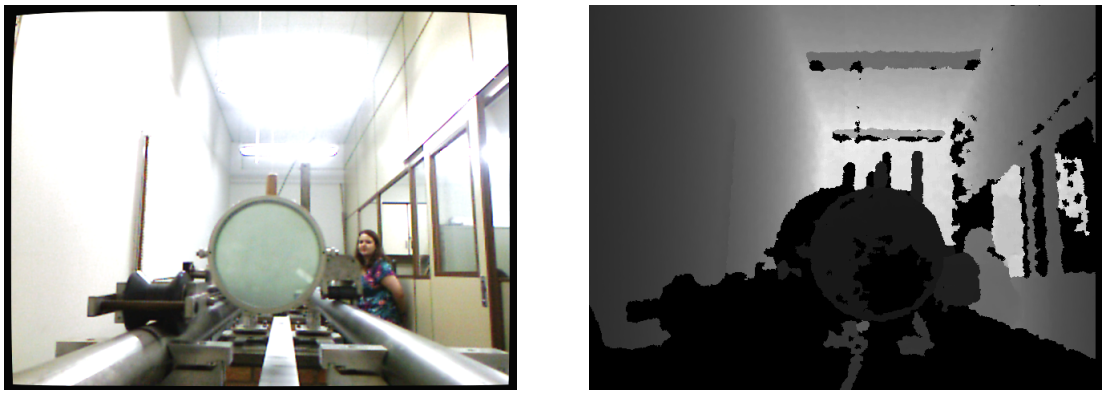
(a) (b)

FIGURA 4.1 – A) DISTORÇÃO RADIAL SIMÉTRICA E DESCENTRADA DA CÂMERA IR; B) DISTORÇÃO RADIAL SIMÉTRICA E DESCENTRADA DA CÂMERA RGB
FONTE: O Autor (2015)

É possível observar na Figura 4.1 o efeito de maior distorção descentrada na imagem IR em comparação com a imagem RGB. A magnitude de distorções radiais, no entanto, é maior na imagem RGB, particularmente no canto superior esquerdo, onde distorções radiais chegam a 4 pixels ($0,037mm$). Isto pode ser verificado por meio de um exame das curvas de distorção radiais. Segundo Khoshelham e Elberink (2012) uma distorção de $0,08mm$ no espaço de imagem corresponde a um desalinhamento de 8 cm na faixa máxima do sensor (5m). Já o cálculo da orientação relativa entre os sensores

IR e RGB do Kinect foi realizado a partir das Equações 2.2.3 e 2.2.3, através das médias aritméticas das observações. Como descrito anteriormente, Khoshelham e Elberink (2012) propuseram um método que expressa o valor d em função de uma normalização linear. Neste caso, d é reescrito na forma $md' + n$, sendo m e n os parâmetros de uma normalização linear (na verdade uma desnormalização) e d' a paralaxe normalizada.

Neste trabalho, foi empregado um interferômetro a LASER da marca Hewlett Packard modelo 5508A para medir os deslocamentos entre o sensor e uma placa de invar com o objetivo de poder calcular as grandezas de m e n. O Kinect foi posicionado de frente a uma placa de invar (ver Figura 4.2) e foram medidos com o interferômetro laser 33 deslocamentos com 10 cm cada da plataforma contendo essa placa e o espelho reprojetor do laser e, em cada intervalo, foi gerado com o Kinect um arquivo com as paralaxes. Assim, através das paralaxes normalizadas de um ponto na placa e do inverso das distâncias medidas com o interferômetro foi realizado o ajuste por mínimos quadrados dessas medidas resultando nos parâmetros m e n. A partir da obtenção desses valores é possível calcular os valores de profundidade de acordo com as paralaxes normalizadas observadas.



(a) (b)

FIGURA 4.2 – EXPERIMENTO REALIZADO COM O INTERFERÔMETRO E COM O KINECT PARA A NORMALIZAÇÃO DOS VALORES DE PROFUNDIDADE. A) IMAGEM RGB; B) IMAGEM DE PROFUNDIDADE.
FONTE: O Autor (2015)

Vale ressaltar que o interferômetro a LASER propicia medidas com precisão em torno 0,01 mm. A Figura 4.3 ilustra um gráfico que mostra a relação profundidade-paralaxe, onde é possível perceber a relação linear expressada pela Equação $md' + n$ resultando nos valores para m de $-3,07x10-5$ e para n de $0,0334$.

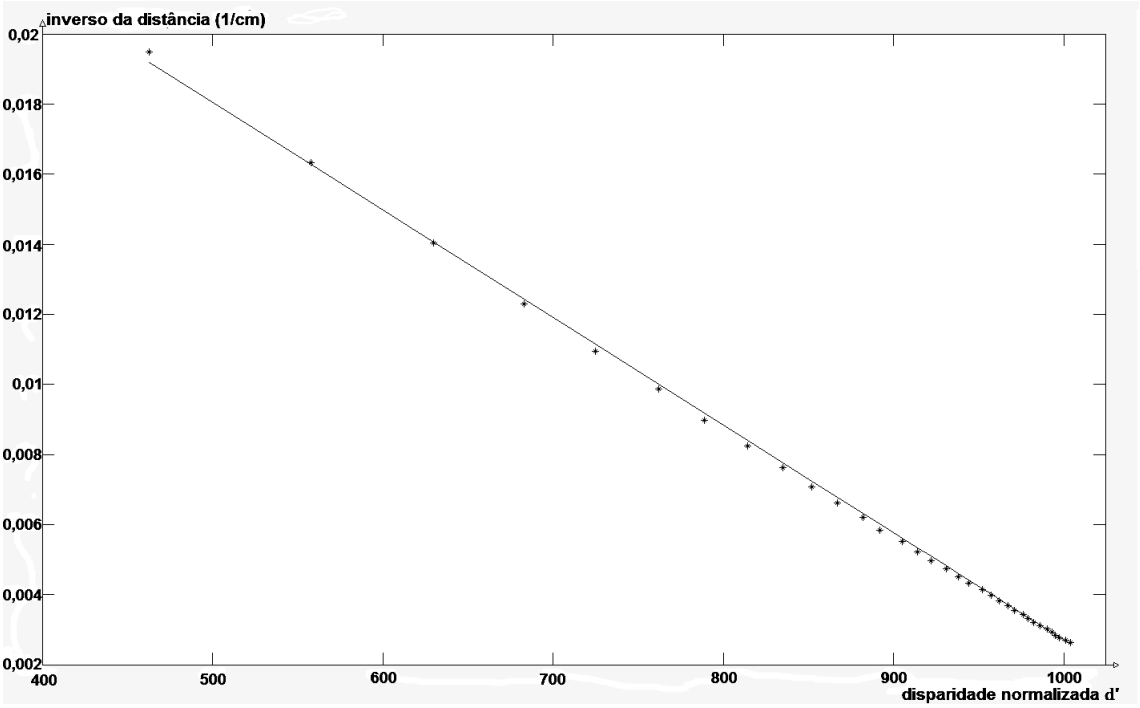


FIGURA 4.3 – RELAÇÃO LINEAR DA PARALAXE NORMALIZADA COM O INVERSO DA PROFUNDIDADE
FONTE: O Autor (2015)

4.2 Reconstrução dos Ambientes Internos

A partir da obtenção dos parâmetros da calibração foi possível conduzir os experimentos. Para avaliar o método proposto neste trabalho foram realizados 4 ensaios em cenários (A-D) diferentes que representam ambientes internos, tais como, salas, cozinha, etc. O dispositivo Kinect usado neste trabalho foi configurado para obter dados com uma frequência de 25 fps através do aplicativo RGB-Demo. Este aplicativo realiza as coletas dos dados de cada *frame* e armazena-os em disco, estes dados são, a imagem RGB, a imagem IR e o arquivo contendo os valores de paralaxe discretizados em 11 bits. Para este trabalho não serão empregados medidas verdadeiras das cenas, ou seja, pontos fiduciais com coordenadas absolutamente definidas, a orientação da nuvem de pontos representando o cenário reconstruído está com orientação relativa.

A Figura 4.4 (a) exibe a aplicação do detector SURF e do RANSAC detecção de pontos homólogos entre os pares de imagem de referência e pesquisa, nota-se uma superfície com variadas formas, texturas e valores de profundidade, com estas características, o algoritmo SURF não apresentou dificuldades na detecção de pontos, e consequentemente na determinação dos parâmetros de transformação rígida por meio da Equação2.43. Obtendo portanto, uma ótima estimativa para o vetor X_a. Na Figura 4.4 (b) é empregado o

segmentador SLIC para segmentar a imagem em um numero limitado de *superpixels* com $K = 100$ e parâmetro $m = 5$. Realizando a correspondência entre os planos a partir da menor distância entre os centroides da imagem de referência com a imagem de pesquisa. Na imagem de referência é realizado a associação entre os pixeis rotulados contidos em cada *superpixels*, com o conjunto de pontos correspondentes do espaço objeto. Para cada conjunto de pontos, foi calculado seus parâmetros do plano (n_x, n_y, n_z, ρ) pelo MMQ. Conhecido os valores dos parâmetros do plano, foi aplicado o modelo proposta através da Equação 3.15, onde o vetor X_0 assume os valores do vetor X_a determinado no processo anterior. Como o vetor X_0 teve uma ótima aproximação, o modelo proposto convergiu rapidamente, com menos de 4 a 5 iterações e os valores calculados dos parâmetros ajustados obtiveram valores bastante baixos, bem próximos a *zero*, significa que os parâmetros calculados anteriormente já obtiveram uma estimativa ótima. Na Figura 4.4 (c) é apresentado o resultado do registro entre as duas nuvens aplicando o modelo proposto.

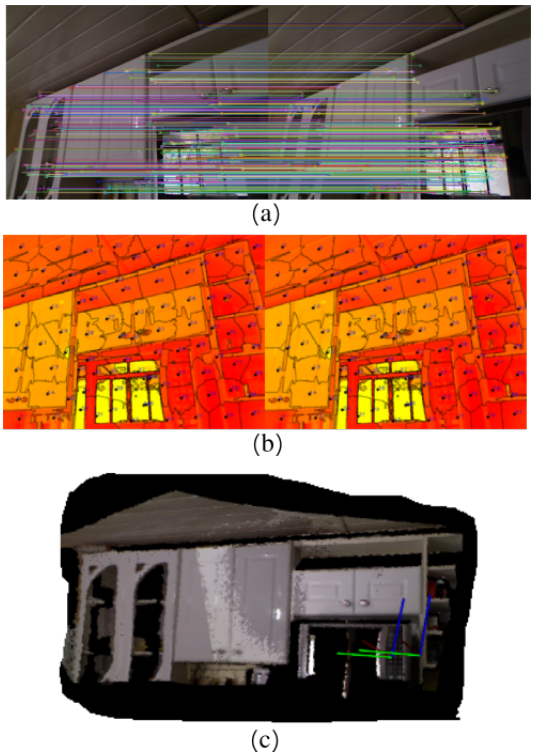
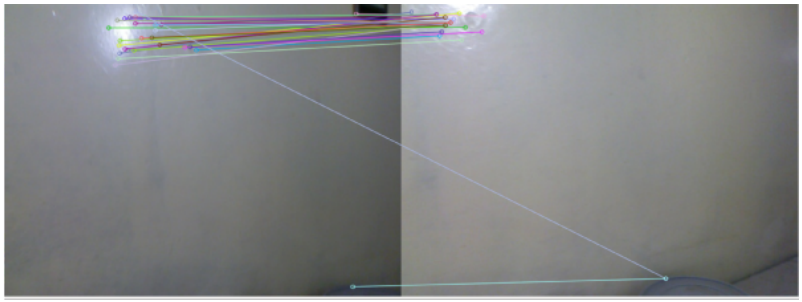


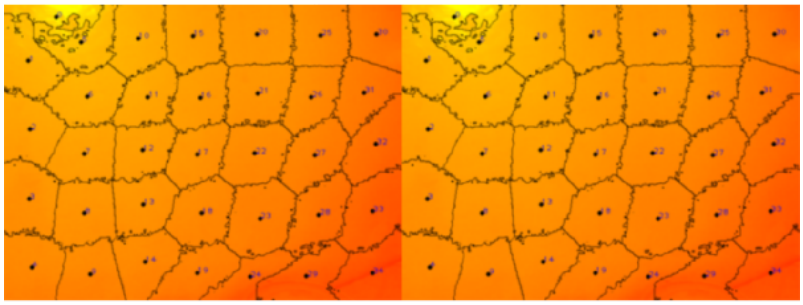
FIGURA 4.4 – RIGISTRO DE NUVENS, (A) DETECTANDO PONTOS HOMÓLOGOS ATRAVÉS DO SURF E RANSAC, (B)SEGMENTAÇÃO DA IMAGEM ATRAVÉS DO SLIC, (C) REGISTRO DO PAR DE NUVEM ATRAVÉS DO MODELO PROPOSTO
FONTE: O Autor (2015)

As superfícies homogêneas, ou seja, com pouca variação de textura como paredes lisas, comprometem a eficiência do detector SURF. A Figura 4.5 (a) exibe esta situação, apesar dos *inliers* detectados estarem em número suficiente para a determinação dos

parâmetros, estão concentrados em apenas uma região da imagem, isto ocorre devido a baixa respostas das componentes Σdx, $\Sigma|dx|$, Σdy e $\Sigma|dy|$ do descritor SURF nas demais regiões, acarretando uma ineficiência na determinação dos parâmetros de transformação. Na 4.5 (b) apresenta a segmentação da imagem pelo SLIC, para posterior refinamento dos parâmetros grosseiros aplicando o modelo *paralaxe-a-plano*.



(a)



(b)

FIGURA 4.5 – SUPERFÍCIE COM SUPERFÍCIE HOMOGÊNEA, (A) DETECTOR SURF E RANSAC, (B) SEGMENTÇÃO APLICANDO SLIC
FONTE: O Autor (2015)

A partir da equação 2.54, são realizadas as transformações consecutivas, e estes parâmetros são armazenados na estrutura do vértice deste grafo (x_i). O grafo vai crescendo de acordo com o deslocamento do dispositivo Kinect pelo cenário, o *frame-chave* (frame atual i), vai verificando as imagens já varridas, através da janela deslisante e do raio de busca de $40cm$. Sendo assim, é formado um conjunto de vértices contendo informações das áreas já imageadas denominados *frames-candidatos (j)*. O detector SURF irá verificar o nível de semelhança entre o *frame candidato* e o *frame chave* através do numero de *inliers* detectados, caso este valor for maior que 50, a área é considerada como revisitada e um novo fechamento de *loop* é detectado, e consequentemente uma nova aresta é criada. Esta aresta irá armazenar os parâmetros de transformação relativa (z_{ij}) entre o *frame-candidato* e o *frame-chave* e a matriz $\Sigma_{X_a}^{-1}$ (matriz informação). A Figura 4.6 mostra os respectivos grafos dos cenários A, B, C e D.

Nota-se no grafo do cenário C (Figura 4.6) (c), que as adição de arestas tendem a

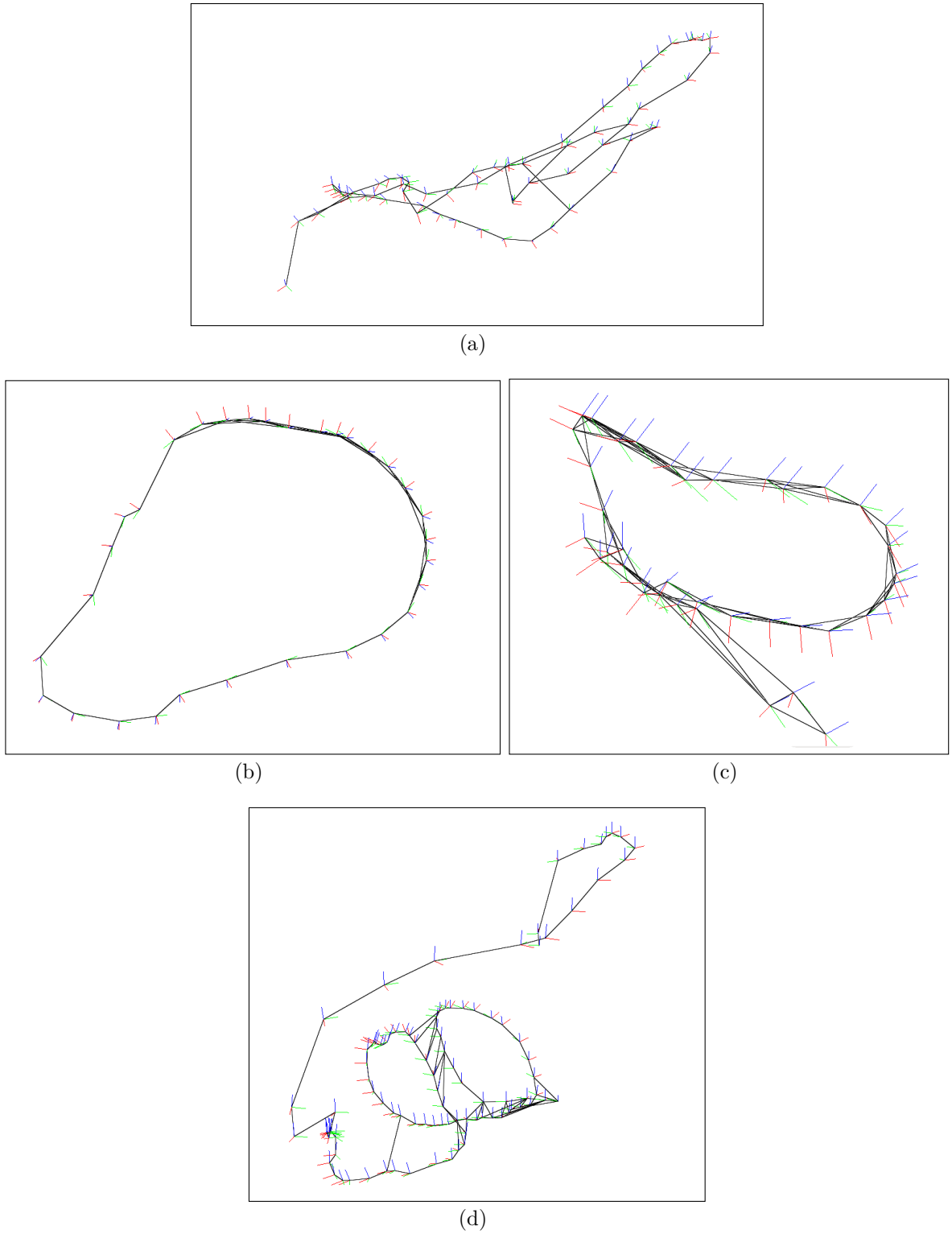


FIGURA 4.6 – TRAJETÓRIA DO DISPOSITIVO KINECT REPRESENTADO NA ESTRUTURA DE UM GRAFO, A) TRAJETÓRIA CENÁRIO A, B) TRAJETÓRIA CENÁRIO B, TRAJETÓRIA CENÁRIO C, TRAJETÓRIA CENÁRIO D
FONTE: O Autor (2015)

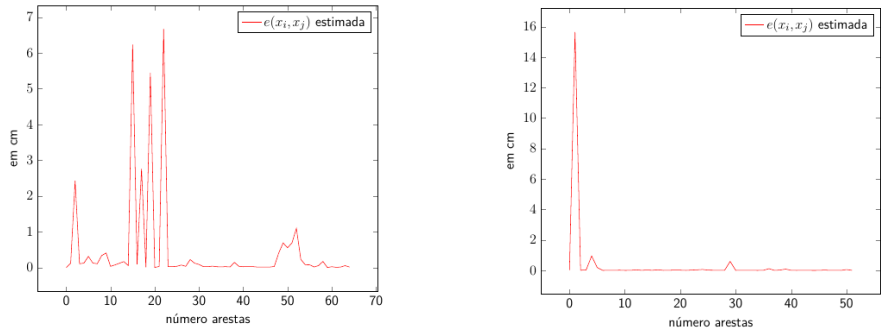
ser criadas em vértices próximos a seus vizinhos, isso indica que os *frames* contidos dentro

do raio de busca, e da janela deslizante tem pouca mudanças de ponto de vista, portanto variação de escala e rotação mínimas comparadas ao *frame-chave*, e por conseguinte um número abundante de *inliers*, em razão destes conjuntos de *frames* compartilharem características semelhantes da cena. Mas conforme esta distancia aumenta, verifica-se um menor número das arestas de fechamento de *loop*, em razão dos *frames-candidatos* estarem se afastando mais da cena atual (*frame-chave*) e consequentemente um número mais escasso de *inliers* detectados. Esta tendência ocorre com os demais grafos. A Tabela 4.3 apresenta os atributos dos grafos construídos representando a trajetória do dispositivo Kinect nos quatro cenários.

TABELA 4.3 – ATRIBUTOS DO GRAFO NOS QUATRO CENÁRIOS

Cenário	num. vértices	num. arestas	trajetória (m)
A	62	65	9,73
B	33	52	3,99
C	50	91	2,75
D	101	131	9,77

Conforme discutido na seção 2.7.2, a otimização do grafo tem por finalidade encontrar uma configuração vértices que minimizam o erro introduzido pelas arestas aplicando o funcional (ver Equação 2.58). O sistema é *sobre-determinado* ou seja, com mais observações do que equações para minimizar a soma do quadrado dos erros. Para a otimização do grafo foi empregado o *framework GraphSlam*, disponível na biblioteca MRPT. Para uma analise quantitativa da otimização é verificado os valores computados da função erro $e(x_i, x_j)$ das arestas. Esta função definida pela Equação 2.56, tem por finalidade determinar a diferença entre o valor calculado e observado, Figura 4.7 apresenta o gráfico da função erro para os cenários A e B.

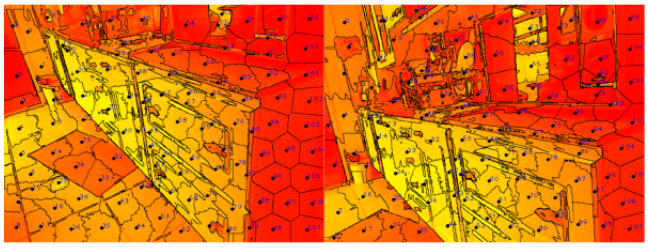


(a) (b)

FIGURA 4.7 – FUNÇÃO DE ERRO; A) CENÁRIO A, B) CENÁRIO B
FONTE: O Autor (2015)

Conforme os gráficos apresentados na Figura 4.7 (a) e (b), os valores próximos a zero, significa que x_i e x_j, correspondem perfeitamente a restrição impostas pelas arestas (z_{ij}), esta característica ocorre em principalmente em arestas de ligação consecutivas onde os *frames* tem pouca varição de escala e rotação devido a pouca mudança de ponto de vista para estas cenas, facilitando o funcionamento do detector SURF, além disso, os gráficos também indicaram a eficiência do modelo proposto em superfície com pouca variação de textura. Por outra lado, os gráficos informam também uma maior propagação erros nas arestas de fechamento de *loop*, se comparadas com arestas de ligação consecutivas, esta maior incertezas ocorrem por fatores como, variação de rotação e de escala encontradas em *frames* de fechamento de *loop*. A Figura 4.8 (a) apresenta este tipo de situação, onde o detector SURF detecta com eficiência os *frames* de *fechamento de loop* entre $i = 65$ e $j = 15$ do cenário A, porém, a Figura 4.8 (b) mostra a segmentação da mesma cena pelo SLIC, neste tipo de situação a correspondência entre os planos foi deficiente, na aplicação do modelo *paralaxe-a-plano* determinando parâmetros com valores espúrios, e desta forma sendo descartados. Outro problema apresentado na segmentação da imagem é a detecção de planos áreas oclusas, ou seja, sem informação de profundidade dos pixeis, devido a superfícies muitos próximas ou distantes da faixa de funcionamento do sensor dificultando a convergência do MMQ na determinação do parâmetros do plano (n_x, n_y, n_z e ρ).

(a)



(b)

FIGURA 4.8 – VARIAÇÃO DE ESCALA ENTRE FRAMES-CHAVE E FRAMES-CANDIDATOS EM SITUAÇÃO DE FECHAMENTO DE LOOP (A) DETECÇÃO DOS INLIERS PELO SURF,(B) SEGMENTAÇÃO DA IMAGEM PELO SLIC
FONTE: O Autor (2015)

A Figura 4.9 mostra a detecção de *fechamento de loop* entre os frames $i = 32$

e $j = 1$, (primeiro e último) do cenário B, apesar do SURF com um número de *inliers* acima do limiar, detectando a área como revisitada, a imagem tem baixa taxa de sobreposição, grande variação em escala e também iluminação. Nota-se a deficiência na detecção de características pontuais e eliminação do *outliers* pelo RANSAC, isso se deve também a definição do *limiar minHessian* do detector SURF que foi fixado com parâmetro 100. É necessário em trabalhos futuros tornar este *limiar* variável, que se adapte automaticamente com com as características do ambiente.

FIGURA 4.9 – DETECÇÃO DE LOOP DETECTADO ENTRE O FRAME 15 E 65 DO CENÁRIO B
FONTE: O Autor (2015)

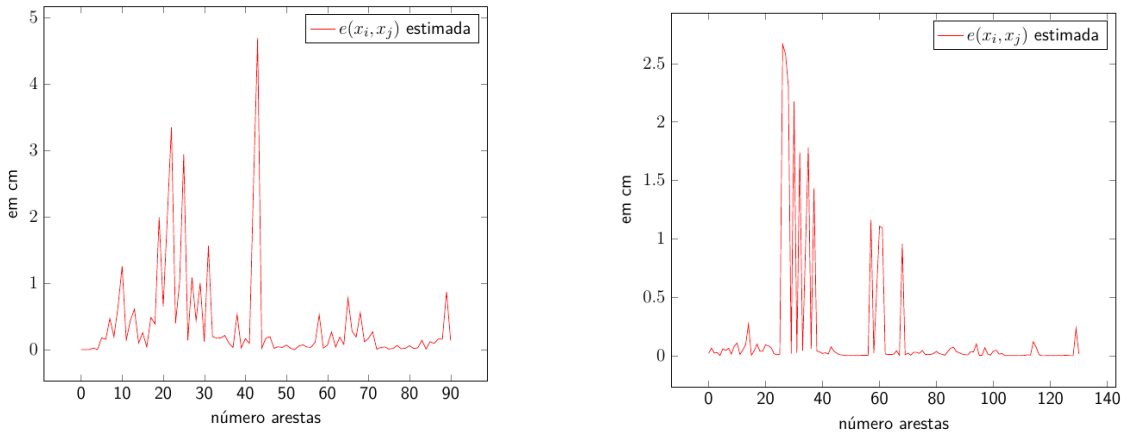
A Figura 4.10 mostra a função erro $e(x_i, x_j)$ para os cenários C e D. Observa-se na figura, a mesma tendência apresentado no cenário A e B, com a propagação de erros sendo maior nas arestas de fechamento de *loop*, essa disposição também indica a dificuldade de aplicar um limiar de *inliers* ideal para indicar o nível semelhança entre as cenas. Apesar da propagação dos erros nas arestas de ligação, estes apresentam em menor magnitude. Isto se deve também as características do cenário, sendo que os cenários C e D apresentavam uma maior variação na textura da superfície imageada.

A Tabela apresenta o dados estatístico da função erro dos grafos representando a trajetória do sensor nos 4 cenários.

TABELA 4.4 – FUNÇÃO ERRO $e(x_i, x_j)$

Cenário	minimo (cm)	máximo(cm)	média (cm)	desv. padrão (cm)
A	0.000	6,669	0,477	1,341
B	0.000	15,654	0,350	2,169
C	0.000	4,685	0,412	0,771
D	0.000	4,662	0,182	0,510

Analisando os dados fornecidos pela Tabela 4.3 nota-se um maior erro no cenário A e B, também vale salientar, que os cenários C e D tiveram um maior recobrimento em números de *frame* em relação ao cenário A e B, isso possibilitou um maior número de



(a) (b)

FIGURA 4.10 – FUNÇÃO DE ERRO; A) CENÁRIO C, B) CENÁRIO D
FONTE: O Autor (2015)

áreas revisitada e consequentemente um maior número de arestas, fornecendo de certa forma uma maior rigidez ao grafo.

Após a construção do grafo, é gerado o arquivo denominado `in.graph`, contendo as informações do grafo como o vetor (x_i) representando os vértices, z_{ij} as arestas e Σ_{ij} a matriz informação. A partir desse arquivo é realizado a otimização global do grafo utilizando o Módulo II do sistema desenvolvido através do GraphSLAM, e retornando um arquivo de saída denominado `out.graph` contendo os novos valores para os parâmetros de posição e orientação dos vértice e da matriz informação, vale lembrar que os parâmeros de transformação entre os vértices, que representam as arestas são mantidos fixos, portanto as arestas no arquivo `out.graph`, contem os mesmo valores do arquivo `in.graph`.

Para verificar a acuracidade da otimização do grafo foi calculado o RMS, este índice é definido a partir do valores do erro global, fornecido pelo algorítimo *levenberg-marquardt*. Através da equação:

$$RMS = \sqrt{\frac{erro_{global}}{num_{arestas}}} \tag{4.1}$$

A Tabela 4.5 apresenta o RMS da otimização do grafo, considerando o limiar de 50 *inliers*, para detectar o lugar como revisitado, e o numero de iterações necessárias para a convergência do algoritmo *levenberg-marquardt*, como também o seu erro final.

Nota-se que foi empregado um limiar de 50 *inliers* para os quatro cenários, o GraphSlam implementado no Módulo II do sistema, convergiu com 7 iterações, este numero baixo é devido as otimizações parciais realizadas em cada *fechamento de loop* de-

TABELA 4.5 – ERRO DA OTIMIZAÇÃO

Cenário	*inliers*	num. iterações	RMS (cm)
A	50	7	11,970
B	50	7	10,245
C	50	7	9,083
D	50	7	6,026

tectado, deste modo o grafo de entrada já tima uma ótima aproximação.

A reconstrução dos cenários é realizado por meio da leitura do arquivo `out.graph` do Módulo III do sistema desenvolvido. Este módulo extrai as informações do vértices contendo os parâmetros de posição ajustados (X_{ka}, Y_{ka}, Z_{ka}) e os de orientação $(\omega_a, \varphi_a, \kappa_a)$. Aplicando a Equação 3.32 é realizado a transformação das coordenadas 3D para todos os pontos. Para um processamento computacional mais ágil as nuvens foram remostradas a uma resolução de 1cm. Teoricamente, cada nuvem de pontos pode ser composto de 307200 $(640x480)$ pontos. Porém, esse número não é efetivo, uma vez que a informação de profundidade não está disponível para todos os pontos devido ao limite de alcance do sensor e as áreas de oclusões causadas pela diferença de ponto de vista. Por exemplo, considerando uma nuvem padrão de 200000 pontos, para cada pixel, são utilizados 24 bits para as posições (X, Y, Z) e 24 bits para as cores RGB, preenchidos de 8 bytes para o alinhamento na memória. Portanto, uma única nuvem de pontos colorida ocupa cerca de $6MB(200000 \times 8)$ na memória. Porém, esta reamostragem interfere na qualidade da visualização. A Figura 4.11 (a) mostra as 255554 pontos que representam a nuvem resultante do cenário A, esta nuvem de pontos constitui em modelar tridimensionalmente um ambiente interno de uma residencia usual do tipo cozinha, nota-se os elementos que compõe o cenário, como mesa, cadeiras e armários. Logo na Figura 4.11 (b) mostra o mesmo cenário, mas com os parâmetros de posição e orientação ajustados, observa-se uma melhora geométrica e visual do cenário reconstruído.

Para uma melhor percepção da diferença da modelagem tridimensional do cenário A, a Figura 4.12 exibe o mesmo cenário mas com vista de topo. Neste ponto de vista é possível perceber a diferença do condicionamento da nuvem de pontos após a otimização global do grafo. Na área em destaque, nota-se um alinhamento mais coerente da nuvens de pontos em relação a parede.

Para verificar a precisão da modelagem tridimensional, foram realizadas medidas em algumas feições da nuvem de pontos, e através de um equipamento de medição do tipo trena foi mensurada à distância da feição homóloga do cenário real. Para o cenário A,

(a)

(b)

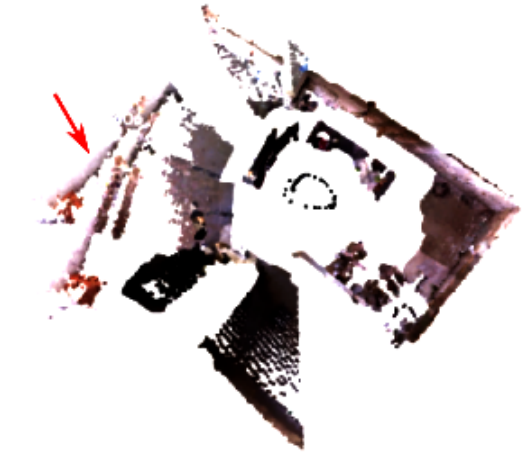
FIGURA 4.11 – CENÁRIO A RECONSTRUÍDO, A)NUVEM ORIGINAL, B) NUVEM OTIMIZADA
FONTE: O Autor (2015)

FIGURA 4.12 – CENÁRIO a VISTA DE TOPO
FONTE: O Autor (2015)

foi realizado a medida da parede de uma extremidade a outra na nuvem de pontos, esta distância foi de 6,08 m, enquanto na trena esta medida foi de 6,20 m resultando portanto, em uma diferença de -12 cm. Também foi realizado uma medição do teto ao chão,na nuvem esta medida foi de $2,14$ m enquanto no cenário real foi de $2,17$ m , portanto uma diferença de -3 cm. Com essas medidas os resultados alcançados tiveram um resultado pior do que a precisão nominal do Kinect que é de $3cm$ para cada $3m$. Esta precisão

bastante baixa reflete com os apresentados na Tabela 4.5.

Na Figura 4.13 (a) mostra os 115354 pontos que modelam tridimensionalmente o cenário B, que consistiu em reconstruir um ambiente interno de uma casa em construção, para isto foi tomado sequencia de 33 *frames* com o Kinect, e o rotacionando aproximadamente no mesmo plano, ou seja, com poucas variações ao ângulos ω e φ, e encerrando o imageamento próximo a área imageada de partida , a Figura 4.13 (b) apresenta o resultado da reconstrução do cenário B, com os parâmetros ajustados, verifica-se a minimização do erro na área fechamento de *loop* (área indicada da figura).



(a)

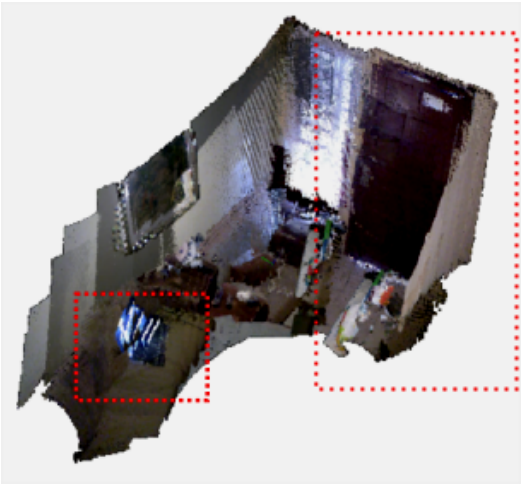
(b)

FIGURA 4.13 – GRAFO DE POSE CENÁRIO B, A) CENÁRIO NÃO OTIMIZADO, B) CENÁRIO OTIMIZADO
FONTE: O Autor

Neste cenário também foram realizadas medidas de distância para verificara precisão da modelagem, a medida de comprimento de uma extremidade a outra do cenário na

nuvem de pontos foi de $5,57$ m,e a medida do comprimento utilizando a trena no cenário real foi de $5,60$ m, portanto uma diferença de $-3cm$, precisão esta tolerável dentro das limitações do sensor Kinect. A medida de largura na nuvem de pontos foi de $3,68cm$ e a medida da largura usando a trena foi de $3,62$ m, portanto um erro de 6 cm, esse erro foi mais acentuado em razão da baixa sobreposição dos *frames* de fechamento *loop*.

O Cenário C constitui em reconstruir tridimensionalmente um ambiente interno de uma residência usual do tipo sala, a Figura 4.14 (a) ilustra os 141604 pontos que formam a reconstrução do ambiente, nota-se os elementos comum neste tipo de ambiente. Na Figura 4.14 (b) é apresentado o resultado da nuvem de pontos após a otimização da trajetória do sensor, observe-se os elementos em destaque, uma melhora geométrica e visual.



(a)

(b)

FIGURA 4.14 – GRAFO DE POSE CENÁRIO C, A) CENÁRIO NÃO OTIMIZADO, B) CENÁRIO OTIMIZADO
FONTE: O Autor

Para verificação da precisão do imageamento 3D, foi realizado medição da altura e largura do ambiente. Para altura foi obtidos os seguintes valores $2,06$ m no ambiente imageado, e $2,05$ no cenário físico, portanto um erro de apenas $1cm$, e para a largura foram registrado a mesma distância $2,08$ m tanto na nuvem, quanto no cenário real, esta precisão foi maior que a nominal do Kinect, isso ocorre devido a um ambiente rico em detalhes e textura variadas. Os *frames* capturados promovem uma maior área recoberta, melhorando a geometria do grafo.

O Cenário D constitui em reconstruir tridimensionalmente um ambiente interno do tipo despensa, a Figura 4.15 mostra as 293794 pontos que formam o modelo 3D, nota-se a correção da nuvem de pontos depois da otimização da trajetória. Para verificação da precisão, foram realizadas medidas de comprimento, altura e largura do cenário. Para o comprimento foi alcançado o valor de 4,09m e 4,07m na nuvem de pontos e na trena respectivamente, ocasionando um erro de $2cm$, na largura $3,53$ m e $3,52$ m, resultando um erro de $1cm$.

FIGURA 4.15 – RECONSTRUÇÃO TRIDIMENSIONAL DO CENÁRIO D
FONTE: O Autor

A Figura 4.16 apresenta a trajetória, antes e depois da otimização do grafo.

Nota-se como a orientação do mapeamento era relativo, portanto, sem pontos fiduciais com coordenadas absolutas para orientar o modelo, deste modo, nota-se que a trajetória estimada, em relação a otimizada sofre uma translação e rotação.

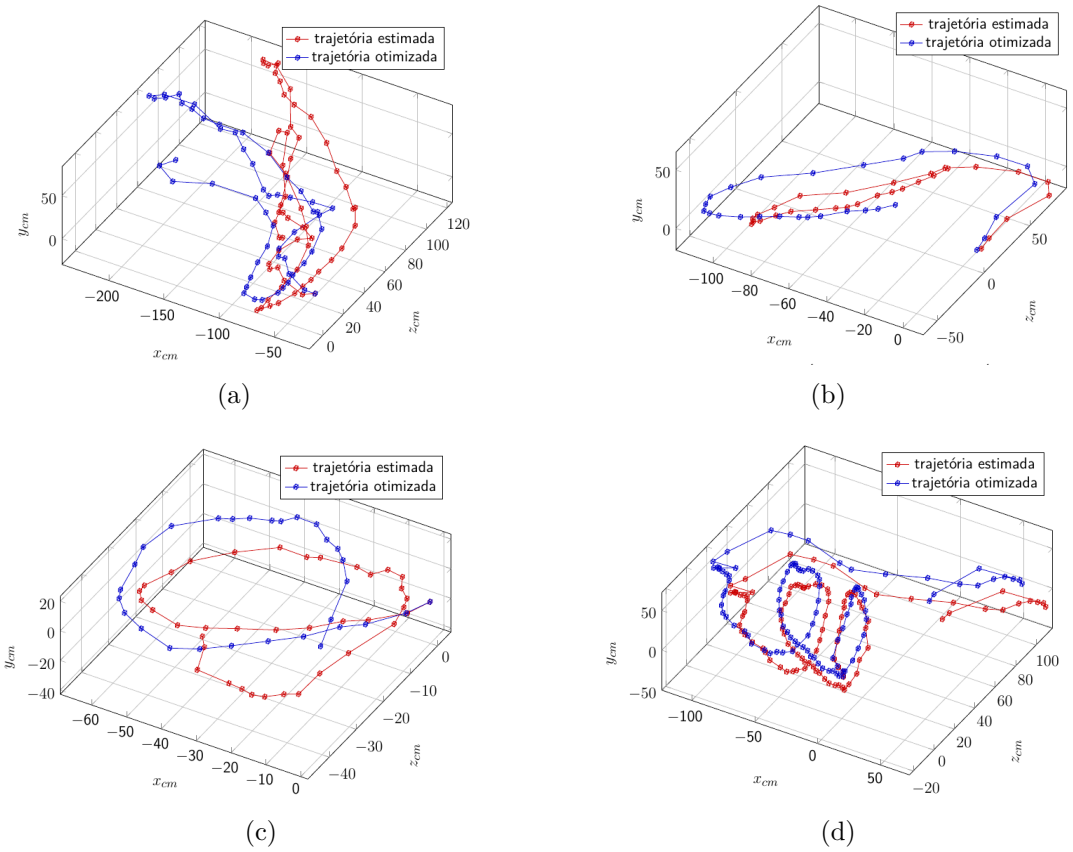


FIGURA 4.16 – TRAJETÓRIA DO DISPOSITIVO KINECT REPRESENTADO NA ESTRUTURA DE UM GRAFO; A) TRAJETÓRIA CENÁRIO A; B) TRAJETÓRIA CENÁRIO B; C) TRAJETÓRIA CENÁRIO C; D) TRAJETÓRIA CENÁRIO D
FONTE: O Autor

5 CONCLUSÕES E RECOMENDAÇÕES

5.1 CONCLUSÕES

Neste trabalho constituiu em desenvolver um método robusto para modelagem 3D de ambientes internos usando dados RGB-D. Para obter as vantagens sinérgicas das informações visuais e de profundidade do sensor *Kinect* foi realizado a calibração dos sensores RGB e IR. Através desse procedimento foi possível determinar os parâmetros de orientação interior de ambas as câmeras e os parâmetros de montagem do sistema, este procedimento permitiu uma maior aderência entre os dados do RGB com o IR e consequentemente uma melhora na determinação dos pontos 3D.

A extração e detecção dos pontos de interesse 2D entre os pares de imagens foi realizada com o detector/descritor SURF e a detecção de *outiliers* pela matriz fundamenta e o RANSAC. A detecção e a correspondência automáticas nos pares de imagens RGB apresentaram-se extremamente eficientes em ambientes que apresentavam uma variada textura na imagens RGB capturadas, além de ser extremamente rápido. Porém, em ambientes com textura homogêneas como paredes lisas, este detector apresentou certa deficiência, o que comprometia a qualidade dos parâmetros de transformação rígida *ponto-a-ponto*. Para contornar este problema foi desenvolvido um modelo matemático, baseado em paralaxe-plano. Este modelo permitiu o refinamento dos parâmetros já determinados, contornando a deficiência do SURF em áreas de textura homogenia. A segmentação da imagem pelo SLIC, para posterior rotulação dos pontos na nuvem, tornou o processamento bastante rápido, evitando a criação de arvores de busca *(k-dtree)* para esta finalidade, o que tornaria o processamento mais oneroso.

Para o problema de consistência global a trajetória do sensor foi representado na estrutura de um grafo dirigido e ponderado. Onde seus vértices representam a posição e orientação do sensor, enquanto as arestas representam as transformações relativas entre os vértices. Esta abstração da trajetória possibilitou recuperação da informação de lugares que já foram imageados, e comparar as semelhanças destas cenas do passado com a cena

atual (*frame-chave*) para detecção de fechamento de loop. Por meio da janela deslizante e de um raio de busca percorrendo o grafo, formavam um novo conjuntos de vértices (frames-candidatos) para serem comparados com o *frame-chave*, aplicando o detector SURF e o RANSAC era verificado a semelhança entre as duas cenas, caso o numero de *inliers* fosse maior 50, o local era reconhecido como revisitado e uma nova aresta era criada. Esta estratégia possibilitou um processamento mais rápido, evitando assim a permutação de todos os vértices do grafo. A ponderação do grafo foi realizado pela matriz informação, definida neste trabalho como a Σ_{Xa}^{-1} dos parâmetros. A otimização do grafo constituiu em determinar a melhor configuração de vértices impostas pelas arestas, o sistema de equações não linear e sobre-determinado, possibilitou a minimizar o erros das arestas pelo MMQ, deste modo reorganizando os vértices do grafo até o algoritmo *levemberg-masquart* convergir. A otimização foi realizado em duas etapas, era realizado uma otimização parcial a cada fechamento de loop e posteriormente era realizado a otimização global de todos os vértices, desta forma reduzindo o numero de iterações nesta etapa, logo que o grafo inicial já teria uma configuração ótima. Para este trabalho foi decidido manter as notações matemáticas originais, porém, pouco usuais no campo da Geodésia Mas visto que toda a documentação e os demais *framework's* como TORO, G^2O se baseiam nessa notação.

Foram realizados experimentos em quatro cenários internos diferentes, os resultados apresentaram que as arestas de fechamento de loop tiveram maior tendência para a propagação de erros. Estas incertezas se deve a fatores como, variação de escala, rotação até mesmo iluminação entre o *frame-chave* e o *frame-candidato*. Tornando o problema da detecção e remoção de outliers pelo SURF e RANSAC e da correspondência de planos um problema complexo. A estratégia de decidir se determinada área ja foi ou não imageada pelo numero de *inliers* é bastante simplória, pois o limiar minHessiam foi fixado em 100, portanto não adaptável em diferentes ambientes, tornando o desempenho do SURF limitado em imagens com estas características. A segmentação pelo SLIC, e a rotulação dos pontos no espaço-imagem formando planos,tem problemas em áreas de oclusão e onde a superfície imageada ultrapassa o limite funcionamento do sensor, causando planos sem informação de profundidade.

De modo geral, o modelo proposto se portou de maneira eficiente nas transformações entre *frames* consecutivos mesmo com textura homogênea, por outro lado se portou ineficiente em locais revisitados, devido a variação de rotação e escala. Mas estes fatores estão relacionados com a correspondência entre os planos, não com o modelo em si. Diante disso os estudos futuros irão focar na eficiência da correspondência entre os planos com alta variação de escala e rotação. Vale também ressaltar que o processamento dos

dados foi *off-line*, sendo assim, a variável tempo e o custo de processamento não foram discutidos e avaliados aqui. Apesar da variável tempo ser uma informação importante na domada de decisão de robôs ou veículos autômatos no mapeamento móvel. Mas dentro do objetivo deste trabalho a principal análise é geométrica, verificando a qualidade final do registro entre nuvens e a trajetória do sensor.

5.2 RECOMENDAÇÕES PARA TRABALHOS FUTUROS

Para uma melhora do sistema, alguns pontos foram destacados, que futuramente serão implementados ou corrigidos, são eles:

- Detector e descritor SURF: apesar de ser bastante otimizado e robusta na detecção de pontos homólogos, apresenta algumas deficiências em imagens com variação de escala, rotação, situações que geralmente ocorrem quando é detectados lugares revisitados no caso de fechamento de *loop*. Portanto, se faz necessário a adição de mais detectores, funcionando em conjunto com o SURF aumentando a robustez. Outra questão relacionado com o SURF, seria desenvolver um limiar *minHessian* ajustável, ou seja, que se ajuste mantendo o número de pontos detectados fixos, indiferente da textura do ambiente imageado;

- Apesar do modelo paralaxe-plano apresentar resultados satisfatório em *frames* com alta taxa de sobreposição, como acontece com os frames consecutivos, mesmo com textura homogênea, esse seu desempenho depende da correspondência dos planos entre as imagens de referencia e pesquisa, para isso seria apresentado futuramente uma abordagem que relacionasse o SURF, com um segmentador de imagens, por exemplo *Watershed* ou de crescimento de regiões, para cada *inlier* detectado, tornando-o pixel semente, desta forma ja rotulando os planos correspondentes referente a cada pixel semente.

- Pequenas adaptações do modelo matemático permitirá sua aplicações a outros dispositivos de imageamento 3D, como as cameras ToF;

- Futuramente desenvolver um sistema on-line, dessa forma possibilitaria novas aplicações ao sistema, como mapeamento móvel simultâneo e realidade aumentada;

- O paradigma de ajustamento baseado em grafo vem sendo aplicado não apenas na área da robótica na resolução de problemas do SLAM, trabalhos recentes de (AGARWAL; BURGARD; STACHNISS, 2014) vem desenvolvendo estudos sobre a

sinergia entre a otimização do grafo no SLAM e a sua aplicação no ajustamento de redes geodésicas. Frente a isso, futuramente sera desenvolvido um aplicativo baseado em grafo aplicando o ajustamento combinado, com as notações matemáticas clássicas da Geodésia.

- Testar outros métodos de otimização por grafo, tais como o G2o, TORO e iSAM.

Referências Bibliográficas

ACHANTA, R. et al. Slic superpixels compared to state-of-the-art superpixel methods. **Pattern Analysis and Machine Intelligence, IEEE Transactions on**, IEEE, v. 34, n. 11, p. 2274–2282, 2012.

AGARWAL, P.; BURGARD, W.; STACHNISS, C. Survey of geodetic mapping methods: Geodetic approaches to mapping and the relationship to graph-based slam. **IEEE Robotics & Automation Magazine**, IEEE, v. 21, n. 3, p. 63–80, 2014.

AL-MANASIR, K.; FRASER, C. S. Registration of terrestrial laser scanner data using imagery. **The Photogrammetric Record**, Wiley Online Library, v. 21, n. 115, p. 255–268, 2006.

ALGABA, M.; BLANCO, J.; GONZÁLEZ, J. **Desarrollo e implementación de un método de generación de mapas 3d usando el sensor kinect**. Tese (Doutorado) — Masters thesis, Higher Technical School of Computer Science Engineering, MAPIR Group, System Engineering and Automation Department, University of Málaga, 2012.

BACHRACH, A. et al. Estimation, planning, and mapping for autonomous flight using an rgb-d camera in gps-denied environments. **The International Journal of Robotics Research**, Sage Publications, v. 31, n. 11, p. 1320–1343, 2012.

BARAKAT, H. F.; DOUCETTE, P.; MIKHAIL, E. M. Photogrammetric analysis of image invariance. In: INTERNATIONAL SOCIETY FOR OPTICS AND PHOTONICS. **Spatial Information from Digital Photogrammetry and Computer Vision: ISPRS Commission III Symposium**. [S.l.], 1994. p. 25–34.

BARNEA, S.; FILIN, S. Registration of terrestrial laser scans via image based features. **International Archives of Photogrammetry and Remote Sensing**, v. 36, n. 3/W52, p. 26–31, 2007.

BAY, H.; TUYTELAARS, T.; GOOL, L. V. Surf: Speeded up robust features. In: **Computer vision–ECCV 2006**. [S.l.]: Springer, 2006. p. 404–417.

BELL, W. B.; DEVARAJAN, V.; APOLLO, S. J. Analysis of area-based image matching under perspective distortion for a planar object model. **Journal of Electronic Imaging**, International Society for Optics and Photonics, v. 8, n. 1, p. 112–125, 1999.

BENDELS, G. H. et al. Image-based registration of 3d-range data using feature surface elements. In: CITESEER. **VAST**. [S.l.], 2004. p. 115–124.

BESL, P. J.; MCKAY, N. D. Method for registration of 3-d shapes. In: INTERNATIONAL SOCIETY FOR OPTICS AND PHOTONICS. **Robotics-DL tentative**. [S.l.], 1992. p. 586–606.

BLANCO, J.-L. **Contributions to Localization, Mapping and Navigation in Mobile Robotics**. Tese (Doutorado) — PhD. in Electrical Engineering, University of Malaga, nov 2009. Disponível em: <http://www.mrpt.org/Paper:J.L-._Blanco_Phd_Thesis>.

BRADSKI, G. **Dr. Dobb's Journal of Software Tools**.

BROWN, L. G. A survey of image registration techniques. **ACM computing surveys (CSUR)**, ACM, v. 24, n. 4, p. 325–376, 1992.

BUNTING, P.; LABROSSE, F.; LUCAS, R. A multi-resolution area-based technique for automatic multi-modal image registration. **Image and Vision Computing**, Elsevier, v. 28, n. 8, p. 1203–1219, 2010.

BURRUS, N. **Kinect RGB Demo v0. 4.0**. 2011.

BURRUS, N. Kinect rgb demo. **Manctl Labs**, 2014.

CARLONE. **Pose graph optimization datasets**. 2015. Access date: 1 jun. 2015. Disponível em: <http://www.lucacarlone.com/index.php/resources/datasets>.

CASTELLANOS, J. A. et al. The spmap: A probabilistic framework for simultaneous localization and map building. **Robotics and Automation, IEEE Transactions on**, IEEE, v. 15, n. 5, p. 948–952, 1999.

CHECCHIN, P. et al. Radar scan matching slam using the fourier-mellin transform. In: SPRINGER. **Field and Service Robotics**. [S.l.], 2010. p. 151–161.

CHEN, Y.; MEDIONI, G. Object modelling by registration of multiple range images. **Image and vision computing**, Elsevier, v. 10, n. 3, p. 145–155, 1992.

CIDECIYAN, A. V. et al. Registration of high-resolution images of the retina. In: INTERNATIONAL SOCIETY FOR OPTICS AND PHOTONICS. **Medical Imaging VI**. [S.l.], 1992. p. 310–322.

ÇIĞLA, C.; ALATAN, A. A. Efficient graph-based image segmentation via speeded-up turbo pixels. In: IEEE. **Image Processing (ICIP), 2010 17th IEEE International Conference on**. [S.l.], 2010. p. 3013–3016.

CIGNONI, P.; CORSINI, M.; RANZUGLIA, G. Meshlab: an open-source 3d mesh processing system. **Ercim news**, v. 73, n. 45-46, p. 6, 2008.

DALMOLIN, Q. **Ajustamento por mínimos quadrados**. [S.l.]: Imprensa Universitária-UFPR, 2002.

DELLAERT, F.; KAESS, M. Square root sam: Simultaneous localization and mapping via square root information smoothing. **The International Journal of Robotics Research**, SAGE Publications, v. 25, n. 12, p. 1181–1203, 2006.

DOLD, C.; BRENNER, C. Registration of terrestrial laser scanning data using planar patches and image data. **International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences**, Citeseer, v. 36, n. 5, p. 78–83, 2006.

DU, H. et al. Interactive 3d modeling of indoor environments with a consumer depth camera. In: ACM. **Proceedings of the 13th international conference on Ubiquitous computing**. [S.l.], 2011. p. 75–84.

EGGERT, D. W.; FITZGIBBON, A. W.; FISHER, R. B. Simultaneous registration of multiple range views for use in reverse engineering of cad models. **Computer Vision and Image Understanding**, Elsevier, v. 69, n. 3, p. 253–272, 1998.

EISENHART, C. Realistic evaluation of the precision and accuracy of instrument calibration systems. **J. Res. Natl. Bur. Stand.(US) C**, v. 67, p. 161–187, 1963.

ELLEKILDE, L.-P. et al. Dense 3d map construction for indoor search and rescue. **Journal of Field Robotics**, Wiley Online Library, v. 24, n. 1-2, p. 71–89, 2007.

ENDRES, F. et al. An evaluation of the rgb-d slam system. In: IEEE. **Robotics and Automation (ICRA), 2012 IEEE International Conference on**. [S.l.], 2012. p. 1691–1696.

ENGELHARD, N. et al. Real-time 3d visual slam with a hand-held rgb-d camera. In: **Proc. of the RGB-D Workshop on 3D Perception in Robotics at the European Robotics Forum, Vasteras, Sweden**. [S.l.: s.n.], 2011. v. 180.

EUSTICE, R. M.; SINGH, H.; LEONARD, J. J. Exactly sparse delayed-state filters. In: IEEE. **Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on**. [S.l.], 2005. p. 2417–2424.

FISCHLER, M. A.; BOLLES, R. C. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. **Communications of the ACM**, ACM, v. 24, n. 6, p. 381–395, 1981.

FÖRSTNER, W.; GÜLCH, E. A fast operator for detection and precise location of distinct points, corners and centres of circular features. p. 281–305, 1987.

FREEDMAN, B. et al. **Depth mapping using projected patterns**. [S.l.]: Google Patents, abr. 3 2012. US Patent 8,150,142.

FU, K.-S.; MUI, J. A survey on image segmentation. **Pattern recognition**, Elsevier, v. 13, n. 1, p. 3–16, 1981.

GALO, M. **Automação dos processos de correspondência e orientação relativa em visão estéreo**. Tese (Doutorado) — Universidade Estadual de Campinas, 2003.

GEMAEL, C. **Introdução ao ajustamento de observações: aplicações geodésicas**. [S.l.]: Editora UFPr, 1994.

GOLUB, G. H.; LOAN, C. F. V. **Matrix computations**. [S.l.]: JHU Press, 2012.

GOSHTASBY, A. Piecewise linear mapping functions for image registration. **Pattern Recognition**, Elsevier, v. 19, n. 6, p. 459–466, 1986.

GRISETTI, G. et al. Efficient estimation of accurate maximum likelihood maps in 3d. In: IEEE. **Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on**. [S.l.], 2007. p. 3472–3478.

GRISETTI, G. et al. Hierarchical optimization on manifolds for online 2d and 3d mapping. In: IEEE. **Robotics and Automation (ICRA), 2010 IEEE International Conference on**. [S.l.], 2010. p. 273–278.

GRISETTI, G. et al. A tutorial on graph-based slam. **Intelligent Transportation Systems Magazine, IEEE**, IEEE, v. 2, n. 4, p. 31–43, 2010.

GRISETTI, G. et al. **Toro-tree-based network optimizer**. 2009.

GUENNEBAUD, G.; JACOB, B. et al. **Eigen v3**. 2010. Http://eigen.tuxfamily.org.

HABIB, A. F.; ALRUZOUQ, R. I. Line-based modified iterated hough transform for automatic registration of multi-source imagery. **The Photogrammetric Record**, Wiley Online Library, v. 19, n. 105, p. 5–21, 2004.

HABIB, A. F.; KIM, C.-J.; KIM, E.-M. Linear features for semi-automatic registration and change detection of multi-source imagery. In: IEEE. **Geoscience and Remote Sensing Symposium, 2005. IGARSS'05. Proceedings. 2005 IEEE International**. [S.l.], 2005. v. 3, p. 2117–2120.

HAHNEL, D. et al. An efficient fastslam algorithm for generating maps of large-scale cyclic environments from raw laser range measurements. In: IEEE. **Intelligent Robots and Systems, 2003.(IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on**. [S.l.], 2003. v. 1, p. 206–211.

HALE, J. K. **Functional differential equations**. [S.l.]: Springer, 1971.

HARRIS, C.; STEPHENS, M. A combined corner and edge detector. In: CITESEER. **Alvey vision conference**. [S.l.], 1988. v. 15, p. 50.

HARTLEY, R.; ZISSERMAN, A. **Multiple view geometry in computer vision**. [S.l.]: Cambridge university press, 2003.

HARTLEY, R. I. In defense of the eight-point algorithm. **Pattern Analysis and Machine Intelligence, IEEE Transactions on**, IEEE, v. 19, n. 6, p. 580–593, 1997.

HENRY, P. et al. Rgb-d mapping: Using depth cameras for dense 3d modeling of indoor environments. In: CITESEER. **In the 12th International Symposium on Experimental Robotics (ISER**. [S.l.], 2010.

HENRY, P. et al. Rgb-d mapping: Using kinect-style depth cameras for dense 3d modeling of indoor environments. **The International Journal of Robotics Research**, SAGE Publications, v. 31, n. 5, p. 647–663, 2012.

HÖGMAN, V. Building a 3d map from rgb-d sensors. 2012.

HORN, R. A.; JOHNSON, C. R. **Matrix analysis**. [S.l.]: Cambridge university press, 2012.

HUHLE, B.; JENKE, P.; STRASSER, W. On-the-fly scene acquisition with a handy multi-sensor system. **International Journal of Intelligent Systems Technologies and Applications**, Inderscience Publishers, v. 5, n. 3-4, p. 255–263, 2008.

KANG, Z. et al. Automatic registration of terrestrial laser scanning point clouds using panoramic reflectance images. **Sensors**, Molecular Diversity Preservation International, v. 9, n. 4, p. 2621–2646, 2009.

KERL, C.; STURM, J.; CREMERS, D. Dense visual slam for rgb-d cameras. In: IEEE. **Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on**. [S.l.], 2013. p. 2100–2106.

KHER, A. R.; MITRA, S. Registration of noisy sar imagery using morphological feature extractor and 2d cepstrum. In: INTERNATIONAL SOCIETY FOR OPTICS AND PHOTONICS. **San Diego'92**. [S.l.], 1993. p. 281–291.

KHOSHELHAM, K.; ELBERINK, S. O. Accuracy and resolution of kinect depth data for indoor mapping applications. **Sensors**, Molecular Diversity Preservation International, v. 12, n. 2, p. 1437–1454, 2012.

KÜMMERLE, R. et al. g 2 o: A general framework for graph optimization. In: IEEE. **Robotics and Automation (ICRA), 2011 IEEE International Conference on**. [S.l.], 2011. p. 3607–3613.

LEVINSHTEIN, A. et al. Turbopixels: Fast superpixels using geometric flows. **Pattern Analysis and Machine Intelligence, IEEE Transactions on**, IEEE, v. 31, n. 12, p. 2290–2297, 2009.

LI, H.; MANJUNATH, B.; MITRA, S. K. A contour-based approach to multisensor image registration. **Image Processing, IEEE Transactions on**, IEEE, v. 4, n. 3, p. 320–334, 1995.

LOWE, D. G. Distinctive image features from scale-invariant keypoints. **International journal of computer vision**, Springer, v. 60, n. 2, p. 91–110, 2004.

LU, F.; MILIOS, E. Globally consistent range scan alignment for environment mapping. **Autonomous robots**, Springer, v. 4, n. 4, p. 333–349, 1997.

MAAS, H.-G. Robust automatic surface reconstruction with structured light. **International Archives of Photogrammetry and Remote Sensing**, INTERNATIONAL SOCIETY FOR PHOTOGRAMMETRY & REMOTE SENSING, v. 29, p. 709–709, 1993.

MENNA, F. et al. Geometric investigation of a gaming active device. In: INTERNATIONAL SOCIETY FOR OPTICS AND PHOTONICS. **SPIE Optical Metrology**. [S.l.], 2011. p. 80850G–80850G.

MIKHAIL, E. M.; BETHEL, J. S.; MCGLONE, J. C. **Introduction to modern photogrammetry**. [S.l.]: John Wiley & Sons Inc, 2001.

MONTEMERLO, M. et al. Fastslam: A factored solution to the simultaneous localization and mapping problem. In: **Aaai/iaai**. [S.l.: s.n.], 2002. p. 593–598.

NEIRA, J. et al. Mobile robot localization and map building using monocular vision. In: CITESEER. **In The 5th Symposium for Intelligent Robotics Systems**. [S.l.], 1997.

OKATANI, I. S.; DEGUCHI, K. A method for fine registration of multiple view range images considering the measurement error properties. In: IEEE. **Pattern Recognition, 2000. Proceedings. 15th International Conference on**. [S.l.], 2000. v. 1, p. 280–283.

OLSON, E.; LEONARD, J.; TELLER, S. Fast iterative alignment of pose graphs with poor initial estimates. In: IEEE. **Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on**. [S.l.], 2006. p. 2262–2269.

PARK, S.-Y.; SUBBARAO, M. A fast point-to-tangent plane technique for multi-view registration. In: IEEE. **3-D Digital Imaging and Modeling, 2003. 3DIM 2003. Proceedings. Fourth International Conference on**. [S.l.], 2003. p. 276–283.

PRUSAK, A. et al. Pose estimation and map building with a time-of-flight-camera for robot navigation. **International Journal of Intelligent Systems Technologies and Applications**, Inderscience Publishers, v. 5, n. 3-4, p. 355–364, 2008.

REISS, M. L.; TOMMASELLI, A. M. A low-cost 3d reconstruction system using a single-shot projection of a pattern matrix. **The Photogrammetric Record**, Wiley Online Library, v. 26, n. 133, p. 91–110, 2011.

ROWLAND, T. Manifold. mathworld-a wolfram web resource. **available at mathworld. wolfram. com/Manifold. html**, 2010.

RUSS, J. C. **The image processing handbook**. [S.l.]: CRC press, 2015.

RUSU, R. B.; COUSINS, S. 3D is here: Point Cloud Library (PCL). In: **IEEE International Conference on Robotics and Automation (ICRA)**. Shanghai, China: [s.n.], 2011.

SCHROEDER, W. M.; MARTIN, K.; LORENSEN, B. The visualization toolkit: An object oriented approach to 3d graphics 3rd edition, kitware. **Inc. Publisher**, 2003.

SEGAL, A.; HAEHNEL, D.; THRUN, S. Generalized-icp. In: **Robotics: Science and Systems**. [S.l.: s.n.], 2009. v. 2, n. 4.

SIEK, J.; LEE, L.-Q.; LUMSDAINE, A. **Boost Random Number Library**. June 2000. Http://www.boost.org/libs/graph/.

SMITH, R.; SELF, M.; CHEESEMAN, P. Estimating uncertain spatial relationships in robotics. In: **Autonomous robot vehicles**. [S.l.]: Springer, 1990. p. 167–193.

SMITH, S. M.; BRADY, J. M. Susan- a new approach to low level image processing. **International journal of computer vision**, Springer, v. 23, n. 1, p. 45–78, 1997.

STAMOS, I. et al. Integrating automated range registration with multiview geometry for the photorealistic modeling of large-scale scenes. **International Journal of Computer Vision**, Springer, v. 78, n. 2-3, p. 237–260, 2008.

STEINBRÜCKER, F.; STURM, J.; CREMERS, D. Real-time visual odometry from dense rgb-d images. In: IEEE. **Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on**. [S.l.], 2011. p. 719–722.

THRUN, S. et al. Simultaneous localization and mapping with sparse extended information filters. **The International Journal of Robotics Research**, SAGE Publications, v. 23, n. 7-8, p. 693–716, 2004.

THRUN, S.; MONTEMERLO, M. The graph slam algorithm with applications to large-scale mapping of urban structures. **The International Journal of Robotics Research**, SAGE Publications, v. 25, n. 5-6, p. 403–429, 2006.

YASEIN, M. S.; AGATHOKLIS, P. A feature-based image registration technique for images of different scale. In: IEEE. **Circuits and Systems, 2008. ISCAS 2008. IEEE International Symposium on**. [S.l.], 2008. p. 3558–3561.

ZHANG, Z. Iterative point matching for registration of free-form curves and surfaces. **International journal of computer vision**, Springer, v. 13, n. 2, p. 119–152, 1994.

ZHANG, Z. Flexible camera calibration by viewing a plane from unknown orientations. In: IEEE. **Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on**. [S.l.], 1999. v. 1, p. 666–673.

ZHENG, Q.; CHELLAPPA, R. A computational vision approach to image registration. **Image Processing, IEEE Transactions on**, IEEE, v. 2, n. 3, p. 311–326, 1993.

APÊNDICE A – Derivadas Parciais do Modelo Proposto

Considerando o modelo proposto:

$$F(X_a, La) : \begin{bmatrix} n_x & n_y & n_z & -\rho \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{23} & r_{23} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_f \\ y_f \\ f \\ f(n + md') \end{bmatrix} = 0 \tag{A.1}$$

Onde:

$r_{11} = cos(\varphi)cos(\kappa)$; $r_{12} = cos(\omega)sen(\kappa) + sen(\omega)sen(\varphi)cos(\kappa)$;

$r_{13} = sin(\omega)sen(\kappa) - cos(\omega)sen(\varphi)cos(\kappa)$;

$r_{21} = cos(\varphi)sen(\kappa)$;

$r_{22} = cos(\omega)cos(\kappa) - sen(\omega)sen(\varphi)sen(\kappa)$;

$r_{23} = sen(\omega)cos(\varphi) + cos(\omega)sen(\varphi)sen(\kappa)$;

$r_{31} = sen(\varphi)$;

$r_{32} = -sen(\varphi)cos(\phi)$;

$r_{33} = cos(\omega)cos(\varphi)$

Derivadas parciais do modelo em relação aos parâmetros:

$$\frac{\partial F}{\partial t_x} = n_x f\,(m + nd')$$

$$\frac{\partial F}{\partial t_y} = n_y f\,(m + nd')$$

$$\frac{\partial F}{\partial t_z} = n_z f\,(m + nd')$$

$$\frac{\partial F}{\partial \kappa} = n_x(r_{11}x_f + r_{12})y_f + r_{13}f + t_x f(m + n)) + n_y(r_{21}x_f + r_{22}y_f + r_{23}f + t_y f(m + n))$$

$$\frac{\partial F}{\partial \varphi} = n_x\left(\frac{\partial r_{11}}{\partial \varphi}x_f + \frac{\partial r_{21}}{\partial \varphi}y_f + \frac{\partial r_{13}}{\partial \varphi}f\right) + n_y\left(\frac{\partial r_{21}}{\partial \varphi}x_f + \frac{\partial r_{22}}{\partial \varphi}y_f + \frac{\partial r_{23}}{\partial \varphi}f\right) + n_z\left(\frac{\partial r_{31}}{\partial \varphi}x_f + \frac{\partial r_{32}}{\partial \varphi}y_f + \frac{\partial r_{33}}{\partial \varphi}f\right)$$

Onde:

$$\frac{\partial r_{11}}{\partial \varphi} = -sen\varphi cos\kappa$$

$$\frac{\partial r_{12}}{\partial \varphi} = -sen\omega cos\varphi cos\kappa$$

$$\frac{\partial r_{13}}{\partial \varphi} = cos\omega cos\varphi cos\kappa$$

$$\frac{\partial r_{21}}{\partial \varphi} = sen\varphi cos\kappa$$

$$\frac{\partial r_{22}}{\partial \varphi} = -sen\omega cos\varphi sen\kappa$$

$$\frac{\partial r_{23}}{\partial \varphi} = cos\omega cos\varphi sen\kappa$$

$$\frac{\partial r_{31}}{\partial \varphi} = cos\varphi$$

$$\frac{\partial r_{32}}{\partial \varphi} = sen\omega sen\varphi$$

$$\frac{\partial r_{33}}{\partial \varphi} = -cos\omega sen\varphi$$

$$\frac{\partial F}{\partial \omega} = n_x\left(-r_{13}y_f + r_{12}f\right) + n_y\left(-r_{23}y_f + r_{22}f\right) + n_z\left(-r_{33}y_f + r_{32}f\right)$$

Derivadas parciais em relação as observações:

$$\frac{\partial F}{\partial x_f} = n_x r_{11} + n_y r_{21} + n_z r_{31}$$

$$\frac{\partial F}{\partial y_f} = n_x r_{12} + n_y r_{22} + n_z r_{32}$$

$$\frac{\partial F}{\partial d'} = n_x t_x + n_y t_y + n_z t_z + \rho$$

$$\frac{\partial F}{\partial n_x} = r_{11}xf + r_{12}y_f + t_x f\left(m + nd'\right)$$

$$\frac{\partial F}{\partial n_y} = r_{21} x f + r_{22} y_f + t_y f \left(m + n d' \right)$$

$$\frac{\partial F}{\partial n_z} = r_{31} x f + r_{32} y_f + t_x f \left(m + n d' \right)$$

$$\frac{\partial F}{\partial \rho} = f \left(m + n d' \right)$$

ANEXO A – Interface do Sistema

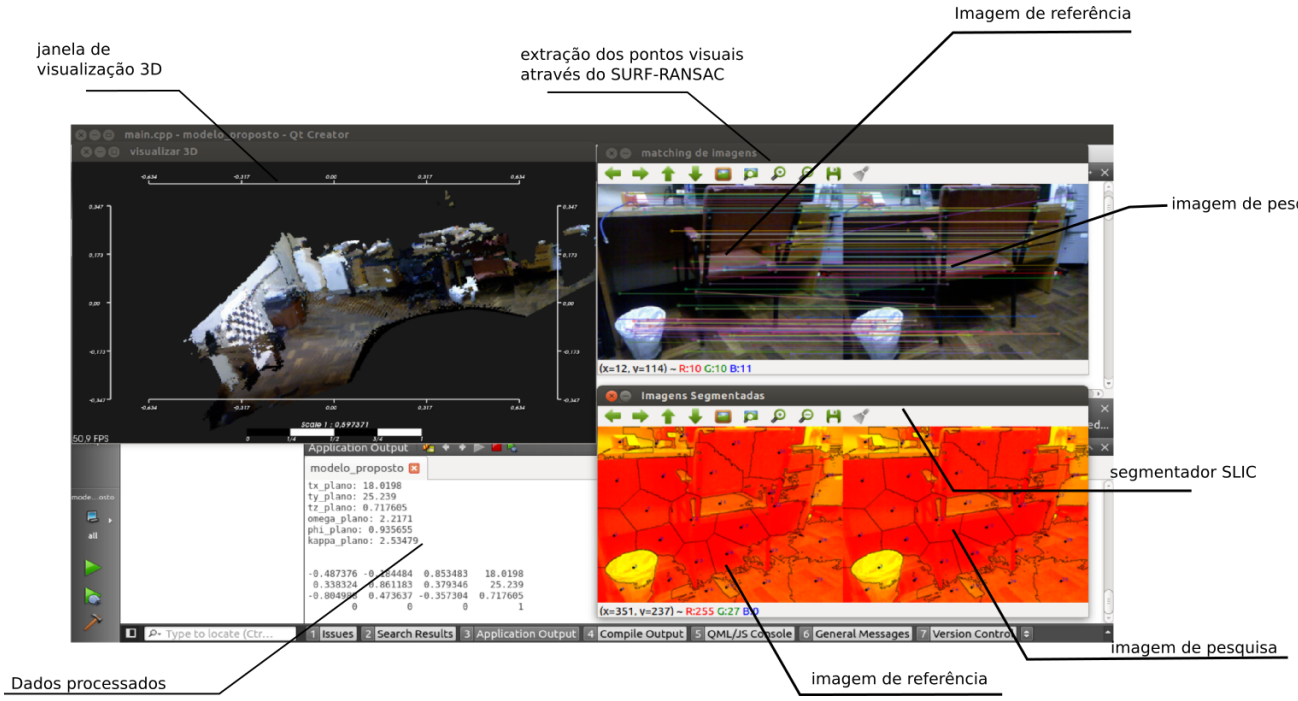


FIGURA A.1 – INTERFACE DO SISTEMA - MÓDULO I
FONTE: O Autor (2015)

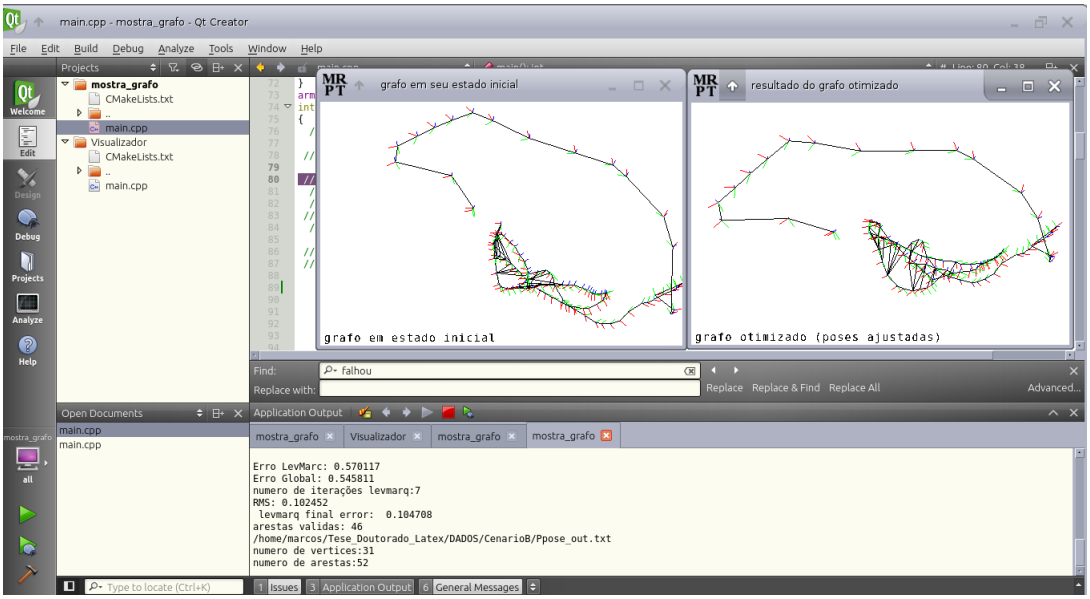


FIGURA A.2 – INTERFACE DO SISTEMA - MÓDULO II
FONTE: O Autor (2015)

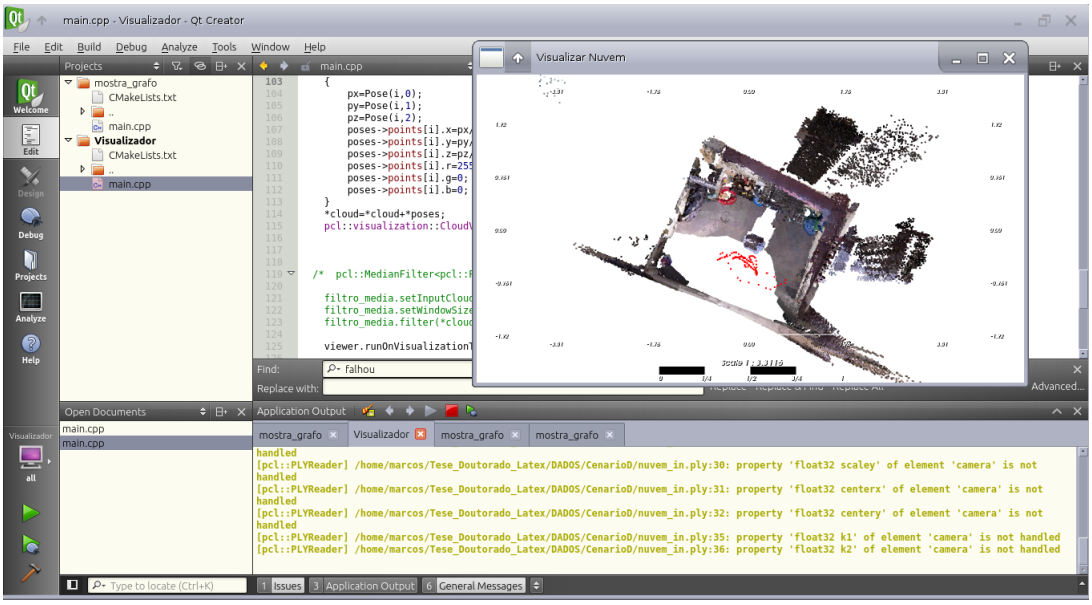
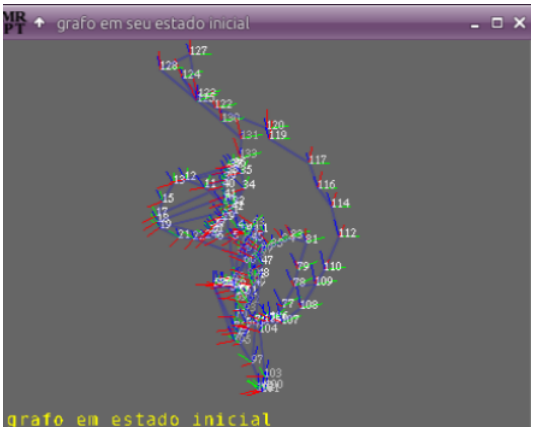


FIGURA A.3 – INTERFACE DO SISTEMA - MÓDULO III
FONTE: O Autor (2015)

ANEXO B – Mais cenários

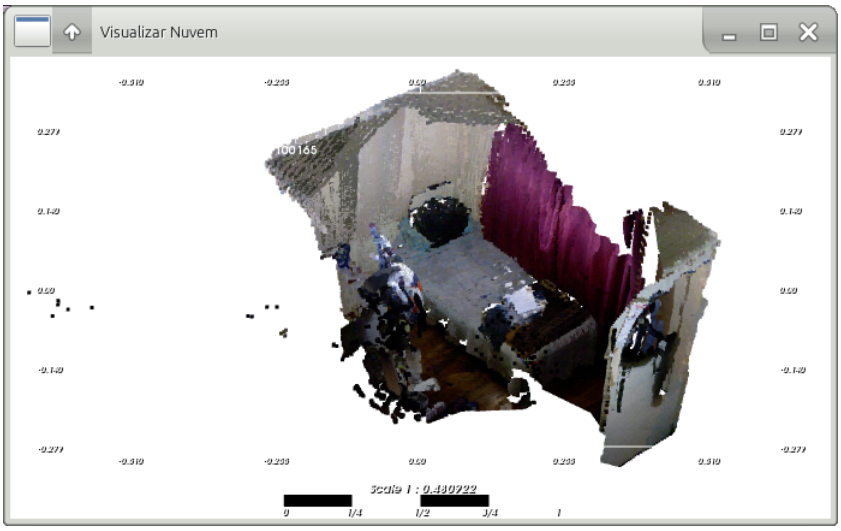
Este anexo apresenta alguns cenários reconstruídos aplicando a metodologia proposta

(a)

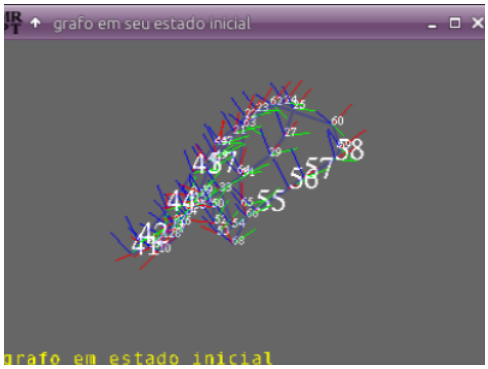


(b)

FIGURA B.1 – CENÁRIO CONSTRUÇÃO, A) RECONSTRUÇÃO 3D, B)GRAFO DE TRAJETÓRIA
FONTE: O Autor



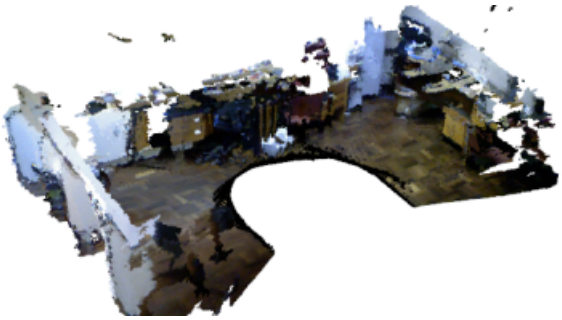
(a)



(b)

FIGURA B.2 – CENÁRIO QUARTO, A) RECONSTRUÇÃO 3D, B)GRAFO DE TRAJETÓRIA
FONTE: O Autor

(a)



(b)

FIGURA B.3 – CENÁRIO LAPE, A) RECONSTRUÇÃO 3D, B)CENÁRIO 3D OTIMIZADO
FONTE: O Autor