

UNIVERSIDADE FEDERAL DO PARANÁ

RUBENS ANTONIO LEITE BENEVIDES

OTIMIZAÇÃO GLOBAL DO REGISTRO DE NUVENS DE PONTOS 3D POR
INTERPOLAÇÃO LINEAR ESFÉRICA (SLERP) DE QUATÉRNIOS EM CIRCUITOS
FECHADOS

CURITIBA

2021

RUBENS ANTONIO LEITE BENEVIDES

OTIMIZAÇÃO GLOBAL DO REGISTRO DE NUVENS DE PONTOS 3D POR
INTERPOLAÇÃO LINEAR ESFÉRICA (*SLERP*) DE QUATÉRNIOS EM CIRCUITOS
FECHADOS

Dissertação apresentada ao Programa de Pós-Graduação em Ciências Geodésicas, Setor de Ciências da Terra, da Universidade Federal do Paraná, como requisito parcial para a obtenção do título de mestre em Ciências Geodésicas.

Orientador(a): Prof. Dr. Daniel Rodrigues dos Santos

CURITIBA

2021

Catálogo na Fonte: Sistema de Bibliotecas, UFPR
Biblioteca de Ciência e Tecnologia

B465o

Benevides, Rubens Antonio Leite

Otimização global do registro de nuvens de pontos 3D por interpolação linear esférica (SLERP) de quatérnios em circuitos fechados [recurso eletrônico] / Rubens Antonio Leite Benevides. – Curitiba, 2021.

Dissertação - Universidade Federal do Paraná, Setor de Ciências da Terra, Programa de Pós-Graduação em Ciências Geodésicas, 2021.

Orientador: Daniel Rodrigues dos Santos

1. Programação (Matemática). 2. Otimização global. 3. Interpolação. 4. Cartografia – Uso do laser. 5. Laser – Sistema de varredura. 6. Quatérnios. I. Universidade Federal do Paraná. II. Santos, Daniel Rodrigues dos. III. Título.

CDD: 519.7

Bibliotecário: Elias Barbosa da Silva CRB-9/1894



MINISTÉRIO DA EDUCAÇÃO
SETOR DE CIÊNCIAS DA TERRA
UNIVERSIDADE FEDERAL DO PARANÁ
PRÓ-REITORIA DE PESQUISA E PÓS-GRADUAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO CIÊNCIAS
GEODÉSICAS - 40001016002P6

TERMO DE APROVAÇÃO

Os membros da Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação em CIÊNCIAS GEODÉSICAS da Universidade Federal do Paraná foram convocados para realizar a arguição da Dissertação de Mestrado de **RUBENS ANTONIO LEITE BENEVIDES** intitulada: **OTIMIZAÇÃO GLOBAL DO REGISTRO DE NUVENS DE PONTOS 3D POR INTERPOLAÇÃO LINEAR ESFÉRICA (SLERP) DE QUATÉRNIOS EM CIRCUITOS FECHADOS**, sob orientação do Prof. Dr. DANIEL RODRIGUES DOS SANTOS, que após terem inquirido o aluno e realizada a avaliação do trabalho, são de parecer pela sua APROVAÇÃO no rito de defesa.

A outorga do título de mestre está sujeita à homologação pelo colegiado, ao atendimento de todas as indicações e correções solicitadas pela banca e ao pleno atendimento das demandas regimentais do Programa de Pós-Graduação.

CURITIBA, 09 de Março de 2021.

Assinatura Eletrônica

09/03/2021 20:36:03.0

DANIEL RODRIGUES DOS SANTOS

Presidente da Banca Examinadora (UNIVERSIDADE FEDERAL DO PARANÁ)

Assinatura Eletrônica

09/03/2021 21:08:23.0

LUÍS AUGUSTO KOENIG VEIGA

Avaliador Interno (UNIVERSIDADE FEDERAL DO PARANÁ)

Assinatura Eletrônica

09/03/2021 21:00:19.0

MAURICIO GALO

Avaliador Externo (UNIVERSIDADE EST. PAULISTA JÚLIO DE MESQUITA FILHO/PR. PRUDENT)

Centro Politécnico - Caixa Postal 19001 - CURITIBA - Paraná - Brasil

CEP 81531-980 - Tel: (41) 3361-3153 - E-mail: cpgcg@ufpr.br

Documento assinado eletronicamente de acordo com o disposto na legislação federal Decreto 8539 de 08 de outubro de 2015.

Gerado e autenticado pelo SIGA-UFPR, com a seguinte identificação única: 81630

Para autenticar este documento/assinatura, acesse <https://www.prppg.ufpr.br/siga/visitante/autenticacaoassinaturas.jsp> e insira o código 81630

AGRADECIMENTOS

Agradeço, primeiramente, ao professor Daniel Rodrigues dos Santos, que durante minha orientação, fez mais do que me ensinar operações matemáticas. Foi uma boa companhia, um exemplo de humildade, reconhecimento e perseverança na solução de problemas. Agradeço à Universidade Federal do Paraná, que com sua estrutura e qualidade de serviços deu suporte a essa jornada de dois anos do mestrado. Agradeço ao Programa de Pós-graduação em Ciências Geodésicas (PPGCG) e seus docentes, por sempre estarem disponíveis para sanar minhas dúvidas. E agradeço à CAPES pela bolsa de estudos que tornou possível minha estadia em Curitiba.

Não menos importante, agradeço imensamente amigos e colegas que fiz neste programa. Em maior ou menor grau todos vocês contribuíram para a realização deste trabalho. Em especial, gostaria de citar os amigos Kauê Vestana, Elias Nain, Luiz Bossola, Daniel Andrade (Fluniel :), Kalima Pitombeira, Raimundo Sales, Samoel Giehl, Vinícius Martins e Josevando de Sousa. Todos vocês me inspiraram e deram significado à minha existência. Obrigado por tudo isso.

“Uma nova verdade científica não triunfa com a convicção dos seus opositores ou através do esforço em fazê-los ver a luz; triunfa, geralmente, porque esses opositores finalmente morrem e cresce uma nova geração mais familiarizada com ela.”

“A ciência avança de funeral em funeral”

- Max Planck

Ao prof. José Lincoln de Sousa Menezes,
em memória de sua doce docência.

RESUMO

Reconstruir digitalmente o mundo 3D a nossa volta é um tema que interessa a diversas áreas do conhecimento e tem vastas aplicações. Na última década, a quantidade de sensores LIDAR (*Light Detection and Ranging*) cresceu continuamente em função deste fato, o que desencadeou alta demanda pelo aprimoramento do produto bruto desta tecnologia, as nuvens de pontos 3D. Neste contexto, o Laser Scanner Terrestre (LST) se estabeleceu como padrão de qualidade na digitalização 3D de ambientes complexos. Este instrumento utiliza o sensoriamento remoto ativo do tipo LIDAR para gerar nuvens de pontos. Todavia, devido ao modo de operação por varredura, estas nuvens sofrem com obstruções na linha de visão do laser. A consequência disto é que, para obter a digitalização completa de um objeto, diferentes locais são ocupados pelo sensor, e cada posição nova estabelece um sistema de coordenadas relativo àquela posição. Estimar os parâmetros da transformação relativa entre estes sistemas é o problema conhecido como registro de nuvens de pontos 3D. Habitualmente, na reconstrução de um ambiente, múltiplas nuvens de pontos são registradas ao longo de um circuito. Mas, transformar todas as coordenadas destas nuvens para uma única origem global, acumula os erros de cada transformação envolvida. Distribuir este erro ao longo dos parâmetros de orientação de todas as nuvens é o problema abordado neste trabalho. Para isto, um circuito fechado é definido entre as nuvens de um ambiente. Em seguida, as transformações entre pares são estimadas na etapa de registro local. Neste trabalho, três estratégias de registro local foram adotadas, uma automática, uma manual e outra de refinamento por ICP (*Iterative Closest Point*). Obtido o circuito, os parâmetros de orientação global de cada nuvem são calculados por composição das transformações em pares. O ajuste dos parâmetros das rotações globais ocorre convertendo-os para quatérnios, onde cada quatérnio que rotaciona uma nuvem para a origem é interpolado com outro. Utiliza-se a técnica *SLERP* (*Spherical Linear Interpolation*), com intervalos de interpolação deduzidos pelo Método dos Mínimos Quadrados (MMQ). Translações são ajustadas independentemente com o modelo de (LU e MILIOS, 1997). Cinco conjuntos de dados foram utilizados nos testes. Fez-se uma análise qualitativa, em cada, por observação direta, e uma análise quantitativa, em que a distância entre os parâmetros otimizados e não-otimizados foi medida para a verdade de campo (*groundtruth*). Os resultados foram positivos em 4 dos 5 conjuntos. Na análise qualitativa, melhorias gerais foram observadas na organização global das nuvens de cada conjunto. Na análise quantitativa, os parâmetros de cada transformação se aproximaram do *groundtruth*, tanto os parâmetros de rotação quanto os de translação.

Palavras-chave: Nuvem de pontos 3D. Otimização Global. SLERP. Laser Scanner Terrestre. Quatérnios.

ABSTRACT

Digitally reconstructing the 3D world around us is a topic that interests several areas of knowledge and has wide applications. In the last decade, the number of LIDAR (Light Detection and Ranging) sensors has grown continuously due to this fact, which has triggered high demand for the improvement of the crude product of this technology, the 3D point clouds. In this context, the Terrestrial Laser Scanner (TLS) has established itself as a quality standard in 3D scanning of complex environments. This instrument uses active remote sensing of the LIDAR type to generate point clouds. However, due to the sweeping mode of operation, these clouds suffer from obstructions in the laser line of sight. The consequence of this is that, in order to obtain a complete digitization of an object, different locations are occupied by the sensor, and each new position establishes a coordinate system relative to that position. Estimating the parameters of the relative transformation between these systems is the problem known as 3D point cloud registration. Usually, in the reconstruction of an environment, multiple point clouds are recorded along a circuit. But, transforming all the coordinates of these clouds to a single global source, accumulates the errors of each transformation involved. Distributing this error over the orientation parameters of all clouds is the problem addressed in this work. For this, a closed circuit is defined between the clouds of an environment. Then, the transformations between pairs are estimated at the local registration stage. In this work, three local registration strategies were adopted, an automatic, a manual and another for refinement with ICP (Iterative Closest Point). Once the circuit is obtained, the global orientation parameters of each cloud are calculated by composing the transformations in pairs. The adjustment of global rotation parameters occurs by converting them to quaternions, where each quaternion that rotates a cloud to the origin is interpolated with another. The SLERP (Spherical Linear Interpolation) technique is used, with interpolation intervals deduced by the Least Squares Method (LSQ). Translations are adjusted independently with the model of (LU and MILIOS, 1997). Five sets of data were used in the tests. There was a qualitative analysis, in each, by direct observation, and a quantitative analysis, in which the distance between the optimized and non-optimized parameters was measured for the groundtruth. The results were positive in 4 of the 5 datasets. In the qualitative analysis, general improvements were observed in the global organization of the clouds in each set. In the quantitative analysis, the parameters of each transformation approached the groundtruth, both the rotation and translation parameters.

Keywords: 3D point cloud. Global optimization. SLERP. Terrestrial Laser Scanner. Quaternions.

LISTA DE FIGURAS

Figura 1 - Quantidade de artigos publicados com as palavras "LIDAR/ <i>Terrestrial Laser Scanner</i> ".	3
Figura 2 - N° de Patentes que reivindicam sistemas/métodos LIDAR por ano.	17
Figura 3 - Oclusão causada na linha de visão do LST devido obstrução por uma pessoa.	18
Figura 4 - Pontos correspondentes encontrados entre um par de nuvens 3D levantadas por LST com uso de técnicas da visão computacional.	19
Figura 5 - (a) Múltiplos caminhos possíveis de serem utilizados no problema da reconstrução 3D. (b) Categorização dos problemas envolvidos.	24
Figura 6 - Diagrama de como é registrado o tempo entre emissão e recepção de um pulso laser.	28
Figura 7 - Esquema genérico do princípio de varredura do LST.	29
Figura 8 - Matriz de multiplicação entre os elementos da base quaterniônica.	31
Figura 9 - Composição de rotações entre sistemas de coordenadas por quatérnios.	35
Figura 10 - Par de nuvens de pontos referência e pesquisa antes (a) e depois (b) do registro local.	36
Figura 11 - Modelo 3D globalmente consistente obtido após a etapa de refinamento global de nuvens de pontos 3D.	37
Figura 12 - Eixos do sistema Darboux: vetores u , v , w com origem no ponto p_s .	40
Figura 13 - Diagrama da região de influência do descritor PFH.	41
Figura 14 - Diagrama da região de influência do descritor FPFH.	41
Figura 15 - Distância que é minimizada no algoritmo ICP ponto-a-plano.	43
Figura 16 - Esquema simplificado do método proposto.	48
Figura 17 - Circuito fechado com $n = 5$ estações ao redor de uma casa.	55
Figura 18 - Grafo representativo de 5 estações ocupadas por um sensor LST.	58
Figura 19 - Construção da matriz jacobiana utilizada no modelo LUM para o grafo da Figura 18 e sua simplificação para a topologia do grafo na Figura 17.	60
Figura 20 - Datasets utilizados na avaliação do método.	64
Figura 21 - Reconstrução do <i>dataset Courtyard</i> e geometria das estações.	66
Figura 22 - Reconstrução do <i>dataset Facade</i> e geometria das estações.	67
Figura 23 - Reconstrução do <i>dataset Arch</i> e geometria das estações.	68
Figura 24 - Reconstrução do <i>dataset Bremen</i> e geometria das estações.	69
Figura 25 - Reconstrução do <i>dataset UFPR</i> e geometria das estações.	70
Figura 26 - Etapas da otimização nas nuvens de <i>Courtyard</i> .	71
Figura 27 - Etapas da otimização nas nuvens de <i>Facade</i> .	73
Figura 28 - Etapas da otimização nas nuvens de <i>Arch</i> .	74
Figura 29 - Etapas da otimização nas nuvens de Bremen.	75
Figura 30 - Etapas de otimização nas nuvens de Bremen+ICP.	77
Figura 31 - Etapas da otimização nas nuvens de UFPR.	78
Figura 32 - RMSE entre translações otimizadas e <i>groundtruth</i> , <i>Courtyard</i> .	80
Figura 33 - RMSE entre rotações otimizadas e <i>groundtruth</i> , <i>Courtyard</i> .	80
Figura 34 - RMSE entre translações das poses otimizadas e <i>groundtruth</i> , <i>Facade</i> .	82

Figura 35 - RMSE entre rotações das poses otimizadas e <i>groundtruth</i> , <i>Facade</i>	82
Figura 36 - RMSE entre translações das poses otimizadas e <i>groundtruth</i> , <i>Arch</i>	83
Figura 37 - RMSE entre rotações das poses otimizadas e <i>groundtruth</i> , <i>Arch</i>	83
Figura 38 - RMSE entre translações das poses otimizadas e <i>groundtruth</i> , <i>Bremen</i>	85
Figura 39 - RMSE entre rotações das poses otimizadas e <i>groundtruth</i> , <i>Bremen</i>	85
Figura 40 - RMSE entre translações das poses otimizadas e <i>groundtruth</i> , <i>Bremen+ICP</i>	86
Figura 41 - RMSE entre rotações das poses otimizadas e <i>groundtruth</i> <i>Bremen+ICP</i>	86
Figura 42 - RMSE entre translações das poses otimizadas e <i>groundtruth</i> , <i>UFPR</i>	87
Figura 43 - RMSE entre rotações das poses otimizadas e <i>groundtruth</i> <i>UFPR</i>	87
Figura 44 - Erro total na translação das poses cada <i>dataset</i>	88
Figura 45 - Erro total na rotação das poses de cada <i>dataset</i>	89
Figura 46 - Melhor modelo a ser aplicado, por <i>dataset</i> , segundo a maior melhora obtida na translação das poses.	90
Figura 47 - Estimativa ótima de poses em múltiplos circuitos de estações LST	93

LISTA DE TABELAS

Tabela 1 - Exemplos de aplicações do LST.	2
Tabela 2 - Resumo dos trabalhos investigados no estado da arte.	27
Tabela 3 - Parâmetros utilizados no pré-processamento das nuvens e no FGR.	51
Tabela 4 - Descrição dos <i>datasets</i>	64

LISTA DE SIGLAS OU ABREVIATURAS

4PCS - 4 Point Congruent Set
BIM - Building Information Modelling
BSP - Binary Shape Context
CPD - Coherent Point Drift
DFS - Deep First Search
DoG - Difference of Gaussians
FGR - Fast Global Registration
FPFH - Fast Point Feature Histogram
HMMR - Hierarchical Merging based Multiview Registration
ICP - Iterative Closest Point
LAPE - Laboratório de Pesquisa em Dados Espaciais
LASER - Light Amplification by Stimulated Emission of Radiation
LIDAR - Light Detection and Ranging
LST - Laser Scanner Terrestre
LUM - Modelo de ajuste global das translações conforme (LU e MILIOS, 1997)
MMQ - Método dos Mínimos Quadrados
MST - Minimum Spanning Tree
MVC - Matriz Variância-Covariância
NDT - Normal Distribution Transform
PFH - Point Feature Histogram
PCA - Principal Component Analysis
PCL - Point Cloud Library
RANSAC - RANdom SAmple Consensus
SLERP - Spherical Linear Interpolation
UFPR - Universidade Federal do Paraná
VLAD - Vector of Locally Aggregated Descriptor

Sumário

1	INTRODUÇÃO	1
1.1	Relevância para a ciência e economia	1
1.2	Questão Norteadora	17
1.3	Objetivos	21
1.4	Estrutura do Trabalho	21
2	REVISÃO DE LITERATURA	22
2.1	Tecnologia LIDAR	28
2.1.1	Laser Scanner Terrestre - LST	29
2.2	Quatérnios	30
2.2.1	Propriedades axiomáticas dos quatérnios	30
2.2.2	Quatérnios e rotações no espaço 3D	33
2.3	Registro de nuvens de pontos 3D	36
2.3.1	Registro local de nuvens de pontos 3D	37
2.3.2	<i>Fast Point Feature Histogram</i> (FPFH)	39
2.3.3	<i>Iterative Closest Point</i> (ICP)	42
2.4	Interpolação Linear Esférica (SLERP)	43
3	METODOLOGIA	47
3.1	Materiais	47
3.1.1	Hardware	47
3.1.2	Software	47
3.2	Método	48
3.2.1	Aquisição de dados	49
3.2.2	Pré-processamento	49
3.2.3	Registro local	50
3.2.4	Método de refinamento global proposto	51
4	AValiação Experimental	62
4.1	Datasets	62
4.2	Registro local e circuitos	65
4.2.1	<i>Courtyard</i>	65
4.2.2	<i>Facade</i>	66
4.2.3	<i>Arch</i>	67
4.2.4	Bremen	68
4.2.5	UFPR	69

4.3 Avaliação qualitativa das etapas de otimização global	70
4.3.1 <i>Courtyard</i>	71
4.3.2 <i>Facade</i>	72
4.3.3 <i>Arch</i>	74
4.3.4 Bremen	75
4.3.5 UFPR	78
4.4 Avaliação quantitativa das etapas de otimização global	79
4.4.1 <i>Courtyard</i>	80
4.4.2 <i>Facade</i>	81
4.4.3 <i>Arch</i>	83
4.4.4 Bremen	84
4.4.5 UFPR	87
4.4.6 Resumo Quantitativo.....	88
5 CONCLUSÕES	91
5.1 Recomendações.....	92
6 REFERÊNCIAS	95
APENDICE A – TRANSFORMACOES DO CIRCUITO, LOOPCLOSURE E POSES DE COURTYARD.....	102
APENDICE B – TRANSFORMACOES DO CIRCUITO, LOOPCLOSURE E POSES DE FACADE.....	108
APENDICE C – TRANSFORMACOES DO CIRCUITO, LOOPCLOSURE E POSES DE ARCH.....	113
APENDICE D – TRANSFORMACOES DO CIRCUITO, LOOPCLOSURE E POSES DE BREMEN.....	117
APENDICE E – TRANSFORMACOES DO CIRCUITO, LOOPCLOSURE E POSES DE BREMEN (ICP)	126
APENDICE F – TRANSFORMACOES DO CIRCUITO, LOOPCLOSURE E POSES DE UFPR.....	134

1 INTRODUÇÃO

Nas últimas décadas o impulso exponencial da tecnologia abalou paradigmas em diversas ciências, a Cartografia – e áreas afins – se transformaram radicalmente neste salto para o futuro. A própria existência do engenheiro cartógrafo e agrimensor passou a ser ameaçada conforme o maquinário se tornou mais preciso e os algoritmos mais rápidos e inteligentes. Na pesquisa de Frey e Osborne (2017, p.1, tradução nossa) “O futuro do emprego: o quão suscetíveis são os empregos à informatização?”, a chance de automação num futuro próximo, calculada para o bacharel em cartografia e fotogrametria, foi de 88%. Para técnicos de levantamento e mapeamento foi de 96%, e para o motorista de caminhões pesados foi de 79%.

Este estudo, exposto de forma pessimista, pode ser visto de outra perspectiva. Se a fotogrametria foi automatizada, é porque os pesquisadores desta área estudaram e implementaram sistemas inteligentes, capazes de reproduzir o seu trabalho com qualidade igual ou superior ao que faziam. Por sua vez, isto é mais provável de acontecer quando diferentes áreas do conhecimento dialogam. Um exemplo desse diálogo no escopo das Ciências Geodésicas são as contribuições mútuas entre Fotogrametria e Visão Computacional, que ajudaram a dissipar o conhecimento extremamente específico produzido por ambas (HARTLEY E MUNDY, 1993).

É no contexto destas duas áreas que se insere este trabalho. Em resumo, faz-se aqui a investigação laboratorial de um modelo matemático que aprimore a orientação de nuvens de pontos 3D geradas por Laser Scanners Terrestres (LST). Como nuvens de pontos 3D são o produto bruto de sensores que utilizam o Sensoriamento Remoto ativo do tipo LIDAR (*Ligth Detection And Ranging*), acredita-se que os modelos aqui propostos serão igualmente benéficos para quaisquer outros sistemas baseados nesta tecnologia.

1.1 Relevância para a ciência e economia

Não é arriscado supor que o LST contribuiu significativamente para os números apresentados na pesquisa de Frey e Osborne (2017). Este equipamento se padronizou entre os métodos de levantamento da agrimensura, e como uma evolução natural das estações totais, ele permite a aquisição de coordenadas 3D na superfície dos objetos de forma extremamente rápida e com precisão milimétrica (SHAN e TOTH,

2018, p. 30). Comparado aos métodos tradicionais de levantamento, e.g. teodolito, estação total e GNSS, o LST opera em ambientes mais complexos e inacessíveis, dispensa a escolha de pontos específicos e digitaliza quase todo o objeto em poucos minutos, levantando pontos a uma frequência de 10^4 a 10^6 Hz (idem, 2018, p. 32).

É claro que há particularidades nas aplicações de cada instrumento, que não devem ser pensados como concorrentes, mas sim como auxiliares em suas funções. Todavia, as limitações que os cercam, como a necessidade de ambientes externos para o GNSS, e de prismas para as estações totais, não aparentam balancear as vantagens do LST, que é onerado apenas pelo seu alto custo. Tamanha flexibilidade também faz com que a quantidade de aplicações deste equipamento seja vasta, como mostra a Tabela 1.

Tabela 1 - Exemplos de aplicações do LST.

Engenharia Civil	Infraestrutura (pontes, estradas, edificações internas/externas, dutos ...) Renovações (modificação, adição, remoção ...) Modelos 3D de cidades Análise de deformação (túneis, pontes, diques, edifícios ...)
Engenharia Médica	Análise da deformação de partes do corpo antes/depois de cirurgias. Ortodontia, reconstrução de próteses e implantes.
Ciências Forenses	Documentação da cena de crimes
Engenharia Mecânica	Controle robótico (escaneamento da cena para determinação da próxima posição, navegação interna/externa ...) Controle da qualidade de pequenas partes mecânicas
Sensoriamento Remoto	Análise de deformação (erosão, depósito de sedimento, biomassa de florestas ...) Detecção de mudanças Classificação
Cinematografia e jogos eletrônicos	Modelos 3D realistas de objetos, cenários e pessoas para animação
Arqueologia	Digitalização de arquivos (monumentos históricos e culturais, preservação do patrimônio/documentação) Modelos de visualização 3D de sítios arqueológicos e museus

Fonte: adaptada de (SOUDARISSANANE, 2016)

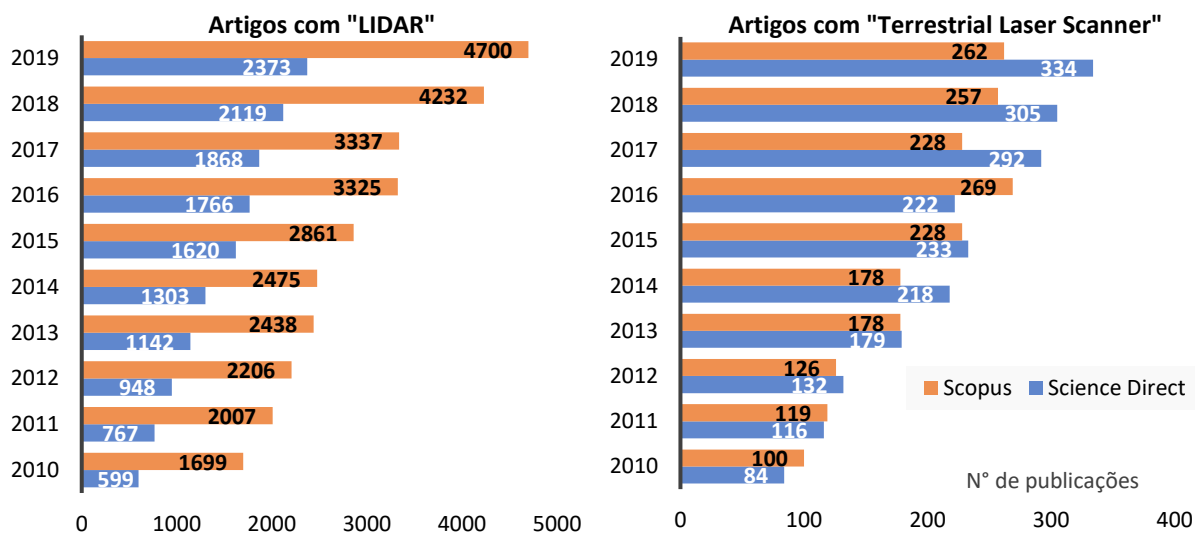
Embora ainda sejam relativamente caros, o crescimento acelerado do mercado LIDAR indica que o barateamento do LST é uma questão de tempo. Dois levantamentos internacionais, *Allied Market Research* (2020) e *Markets and Markets* (2020), preveem crescimento de 20 % ao ano entre 2020 e 2025 no mercado de tecnologias LIDAR, essa mesma taxa era de 9% quando foi prevista cinco anos antes entre 2015 e 2020 (*Markets and Markets*, 2015 apud THEILER et al., 2015, p. 1).

Mais recursos promove mais pesquisa, o que gera mais patentes, por sua vez, estas se traduzem em retorno direto para a sociedade, realimentando o ciclo com

recursos. Três fatos suportam que este ciclo está acelerando em cada uma de suas etapas:

Fato 1: pesquisa. A quantidade de publicações na última década com o termo LIDAR e *Terrestrial Laser Scanner* cresceu ininterruptamente. A Figura 1 mostra dois gráficos com a quantidade de artigos publicados que possuem as palavras “LIDAR” ou “*Terrestrial Laser Scanner*” no título, resumo ou palavras-chave, por ano, na última década.

Figura 1 - Quantidade de artigos publicados com as palavras “LIDAR/*Terrestrial Laser Scanner*”.



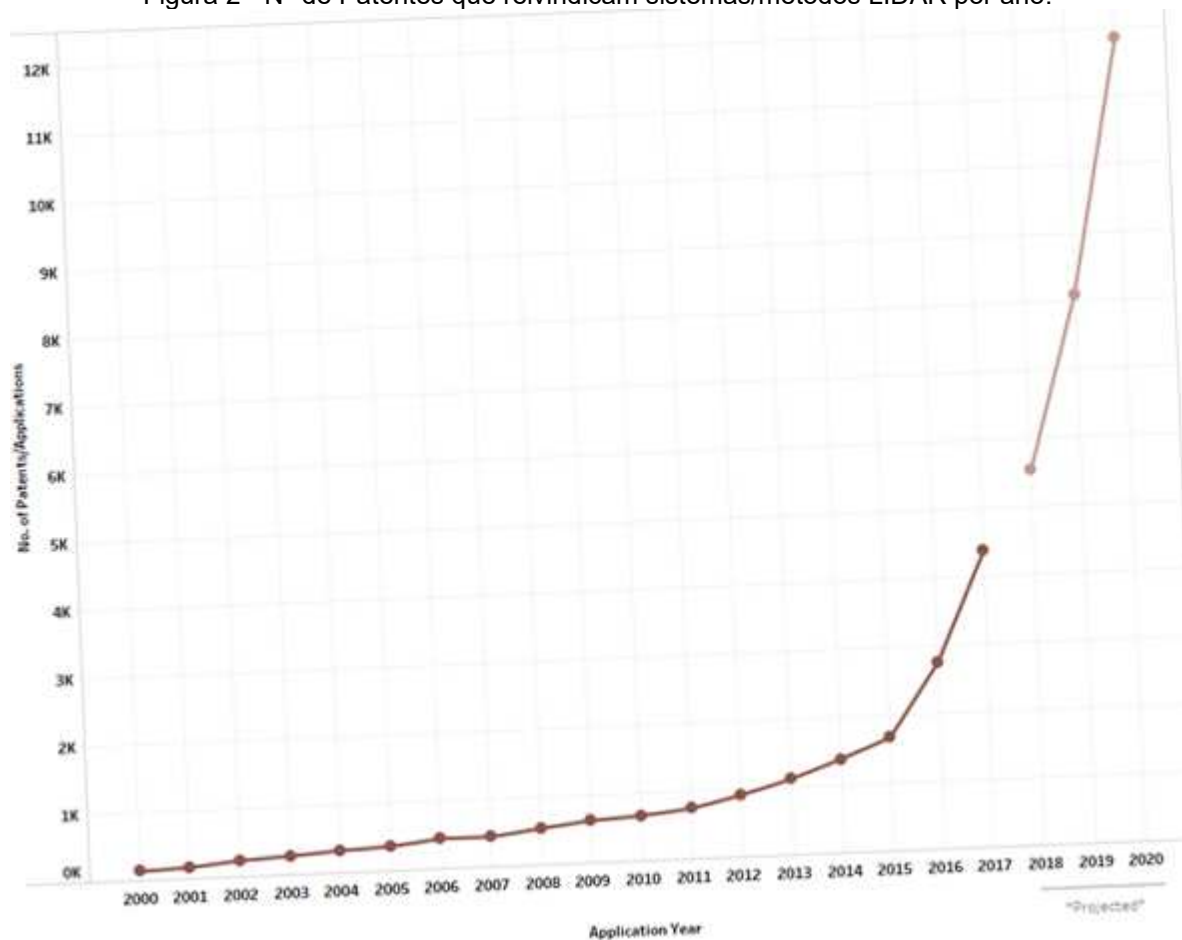
Fonte: elaborada pelo autor (2020)

Dados provenientes das bases Scopus e Science Direct após pesquisa direta pelos termos.

Fato 2: patentes. A quantidade de patentes relacionadas à tecnologia LIDAR começou crescimento exponencial a partir de 2015 (ver Fig. 2 na página seguinte).

Fato 3: retorno direto para a sociedade. Em outubro de 2020 a empresa Apple apresentou o seu smartphone *iPhone 12 Pro*, um dos primeiros do mundo com sensor LIDAR totalmente integrado, isto indica que muito provavelmente a presença deste sensor passará a ser um padrão no projeto de smartphones, mercado que ultrapassará 1,3 trilhão de dólares em 2025 (*Research and Markets, 2020b*). Esta quantia é quase 20 % do PIB brasileiro de 2019. Não seria exagero supor que a tecnologia LIDAR terá um impacto no mundo semelhante ao que as câmeras digitais tiveram na primeira década de 2000, dada a tendência que outras empresas têm de seguir os padrões da empresa Apple. Esta foi uma das primeiras empresas a estabelecer câmeras como padrões em smartphones.

Figura 2 - N° de Patentes que reivindicam sistemas/métodos LIDAR por ano.



Fonte: (MAXVAL, 2020)

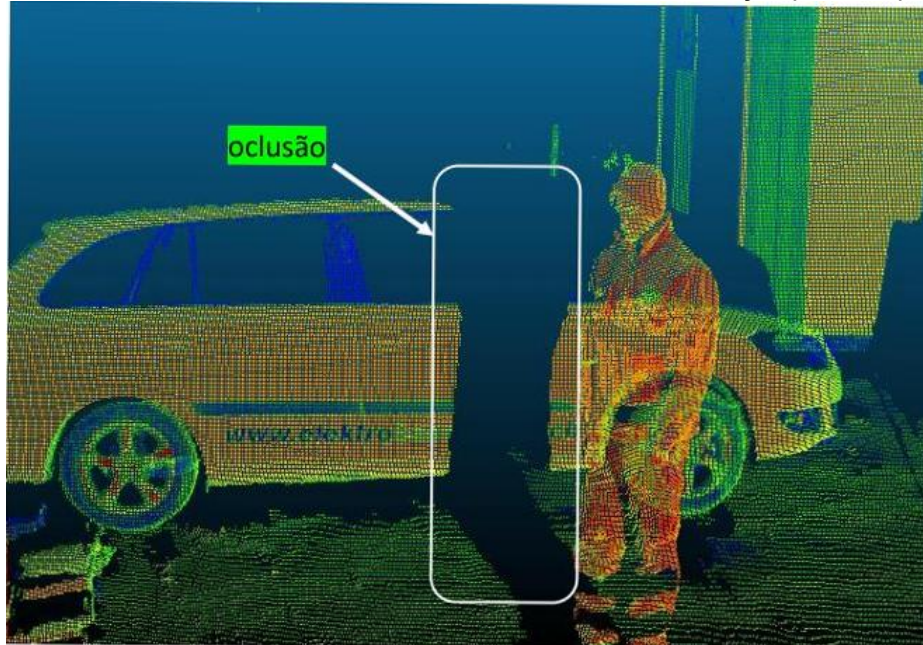
Outra vantagem de investigar algoritmos que aprimorem o produto bruto da tecnologia LIDAR é a presença desta em uma vasta gama de dispositivos, que vão muito além do mapeamento com nuvens de pontos 3D, principal produto bruto do LST. Isto movimenta a engenharia reversa, e.g. no Sensoriamento Remoto, onde sensores ativos que não foram pensados originalmente para o mapeamento, como o Kinect, são usados como alternativa mais barata que os aparelhos totalmente dedicados (KHOSHELHAM, 2011). No exemplo do Kinect, tem-se um concorrente do LST para mapeamento de ambientes *indoor* (DOS SANTOS, et al. 2016), embora custando apenas 1 % do preço médio de um LST tradicional.

1.2 Questão Norteadora

Sabe-se que qualquer varredura feita com um LST sofrerá com oclusões na linha de visão do laser, devido ao modo de operação por emissão de pulsos (Figura 3). A única maneira de contornar essa limitação é posicionar o instrumento em locais

diferentes, de forma que os objetos da cena sejam completamente escaneados em meio às várias posições ocupadas.

Figura 3 - Oclusão causada na linha de visão do LST devido obstrução por uma pessoa.



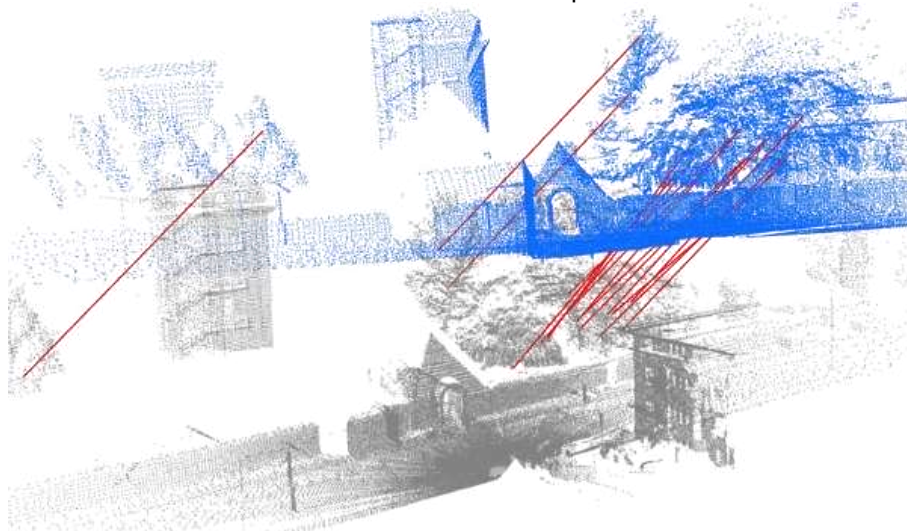
Fonte: adaptada do dataset *Courtyard* de Theiler et al. (2015).

O ato de mudar o LST de lugar define uma nova origem para as coordenadas lidas pelo sensor, portanto, um objeto que esteja presente em duas nuvens de pontos diferentes será representado por coordenadas distintas em cada uma delas. Tal inconsistência é corrigida ao referenciar uma das nuvens de pontos na origem da outra, definindo arbitrariamente uma como referência e outra como móvel. Estimar essa transformação de coordenadas – transformação de corpo rígido 3D – é o problema conhecido como registro de nuvens de pontos 3D (*3D point cloud registration*), e precisa ser solucionado sempre que o sensor se mover, independentemente do tipo de sensor LIDAR ou da aplicação a que se destinem as nuvens de pontos 3D.

A princípio, o registro pode ser solucionado de forma manual ou automática. Na forma manual, um operador indica quais são os pontos correspondentes entre um par de nuvens de pontos, normalmente com a ajuda de alvos reflexivos, e adota uma solução do tipo mínimos quadrados. O problema desta abordagem é o tempo gasto pelo operador. A taxa tempo-em-campo por tempo-em-escritório com nuvens de pontos chega a ser 1/10, significa que para cada dia gasto em campo colhendo-se dados passa-se 10 dias em escritório processando-os (THEILER et al. 2015).

No caso automático essa taxa é significativamente nivelada, pois as correspondências entre nuvens são encontradas por técnicas da visão computacional (Figura 4). Estes métodos automáticos são denominados grosseiros (*coarse registration*) por alguns autores, pois estimam apenas uma transformação aproximada (DÍEZ et al, 2015). Este também é o caso das transformações calculadas com seleção manual, dado que não há correspondências exatas entre pares de nuvens de pontos. Neste sentido, existem métodos de registro para refinamento (*fine registration*) que operam em uma transformação inicial conhecida, seja ela de origem automática ou manual, como é o caso do algoritmo *Iterative Closest Point* (ICP) e suas diversas variações (RUSINKIEWICZ e LEVOY, 2001).

Figura 4 - Pontos correspondentes encontrados entre um par de nuvens 3D levantadas por LST com uso de técnicas da visão computacional.



Fonte: (RUSU et al. 2008).

Em todos os casos citados – registro manual, automático/*coarse* ou refinamento/*fine* – a transformação é calculada par-a-par entre as nuvens de pontos (*pairwise registration* ou registro local). No entanto, habitualmente tem-se múltiplas nuvens em um projeto, se não houver sobreposição entre todas elas – e normalmente não há – será necessário compor as transformações de coordenadas, dando origem às chamadas poses, que acumulam várias transformações relativas para uma origem global. Portanto, poses acumulam os erros de cada registro local ao longo do circuito. Logo, o problema do registro local naturalmente se expande para outro mais complexo, a otimização global das poses. Como tratar o erro que se acumula nas

transformações relativas entre sistemas e distribuí-lo uniformemente pela rede de pares de nuvens? Esta é a questão norteadora deste trabalho.

A solução deste problema é complexa, semelhante à atribuição de pesos no ajustamento de observações. Não é necessário investigar a literatura a fundo para notar a ausência de solução matemática ótima neste caso, basta lembrar que, apenas na etapa de registro local, dezenas de métodos de ponderação podem ser utilizados entre correspondências (BABIN, GIGUERE e POMERLEAU, 2019) e qualquer uma destas atribuições de pesos é um problema de otimização/ajustamento.

Como a otimização global das poses também é um problema que se baseia em ponderação, inevitavelmente se tem um processo de atribuição de pesos em cima de pesos, e isto, obviamente, pode continuar *ad infinitum*. Portanto, é imprescindível a investigação de modelos matemáticos que ofereçam soluções, mesmo que aproximadas, ao problema. Se estas soluções não necessitarem de aproximações iniciais ou métodos iterativos, isto é, forem lineares, mais importante ainda é que sejam objeto de análise, dado que serão mais rápidas e confiáveis.

Dentre os modelos matemáticos que oferecem soluções à otimização global de poses, muitos exploram o fato de que entre múltiplas sobreposições de nuvens há circuitos fechados (THEILER et al., 2015). Isto, naturalmente, leva a uma restrição de fechamento. O levantamento de poligonais topográficas, por exemplo, explora uma restrição deste tipo para ajustamento das observações. Esta restrição também pode ser explorada na otimização de poses, dado que poses calculadas por caminhos diferentes deveriam retornar à mesma origem global em um circuito fechado.

Sabe-se que devido à inevitabilidade de erros aleatórios, composições por caminhos diferentes resultarão em parâmetros diferentes para poses que deveriam ser iguais. Isto, claro, estabelece uma superabundância de informações. Nasce daí a hipótese de que deve existir uma função capaz de ajustar, simultaneamente, todos os parâmetros das poses em um circuito fechado. Esta função, claro, terá que ser deduzida por um modelo de minimização de erros, que normalmente é o Método dos Mínimos Quadrados (MMQ).

Encontrar esta função, principalmente uma que seja linear, não é simples. Todavia, acredita-se que parametrizando as rotações das poses por quatérnios (DAN et al. 1998) e aplicando a técnica de Interpolação Linear Esférica (*Spherical Linear interpolation - SLERP*) (SHOEMAKE, 1985) uma função para refinamento global das rotações pode ser estabelecida com o uso do MMQ. Esta função, como explicado, irá

aproveitar-se das restrições de fechamento num circuito (*loopclosure constraint*) e deverá distribuir o erro que se acumula na rotação, produzindo, portanto, um modelo de nuvens de pontos 3D mais consistente geometricamente. Adicionalmente, pretende-se explorar a integração deste modelo com o modelo global e linear de Lu e Milios (1997) para refinamento das translações das poses.

1.3 Objetivos

O objetivo geral deste trabalho é investigar um modelo linear de refinamento global de nuvens de pontos 3D com base na técnica SLERP. Esta técnica será aplicada nos quatérnios que representem rotações de poses em circuitos fechados. Espera-se que o modelo seja capaz de ajustar, simultaneamente, todos os parâmetros de rotação e translação obtidos na etapa de registro local. Para alcançar este objetivo geral, definem-se os seguintes objetivos específicos:

- I. Desenvolver uma função de refinamento global, que seja linear, dos parâmetros de rotação por MMQ aplicado na Interpolação Linear Esférica (SLERP) de quatérnios.
- II. Integrar a função desenvolvida no objetivo anterior ao modelo global de Lu e Milios (1997) de ajustamento das translações.
- III. Analisar a qualidade do modelo proposto por ensaios experimentais, garantindo heterogeneidade nos testes, com diferentes *datasets*, com circuitos de geometria diversa e com métodos de registro local distintos.

1.4 Estrutura do Trabalho

Este trabalho está estruturado em cinco capítulos. No Capítulo 1 tem-se a introdução, a motivação da pesquisa, a questão norteadora e os objetivos a serem atingidos. No Capítulo 2 apresenta-se a revisão de literatura, onde se discute, o que se acredita ser, os principais trabalhos relacionados à otimização global de poses de nuvens de pontos 3D. No Capítulo 3 são descritos os recursos utilizados e o método proposto para refinamento global das poses. No Capítulo 4 apresentam-se os *datasets*, os ensaios experimentais e as discussões sobre os resultados. Por fim, no Capítulo 5, tem-se as conclusões e recomendações de investigação na mesma linha de pesquisa.

2 REVISÃO DE LITERATURA

É inevitável revisar o registro local de nuvens de pontos 3D, pois, embora não seja o foco principal deste trabalho, sem esta etapa não há como obter-se poses que possam ser otimizadas. Os algoritmos para registro local podem ser divididos em duas categorias principais, de registro grosseiro, que entregam uma sobreposição aproximada entre pares de nuvens (*coarse registration*), e de registro fino, que trabalham em cima de uma transformação conhecida para melhorá-la (*fine registration*) (HOLZ et al., 2015). A solução mais comum do registro local é juntar as duas soluções em sequência, utilizando a saída do método grosseiro na entrada do método fino, esta abordagem se chama *coarse-to-fine registration* (idem, 2015).

O maior desafio dos métodos de registro grosseiro consiste no estabelecimento automático de correspondências (DÍEZ et al, 2015), esse problema é parcialmente resolvido pelo uso de descritores, que geram assinaturas para uma primitiva geométrica – pontos, retas ou planos – com base na sua vizinhança. Devido a grande redundância dos dados em nuvens de pontos 3D, principalmente em nuvens de LST, que são massivas, falsas correspondências sempre estão presentes em grande quantidade. Portanto, é comum a presença de vários métodos de rejeição no espaço dos descritores para que estes sejam eficientes (ZHOU et al. 2016).

De acordo com Weimann (2016, p. 39, tradução nossa) um descritor “consiste em uma representação compacta de propriedades globais ou locais dos dados fornecidos”. Descritores globais operam abreviando a informação contida na nuvem inteira e podem ser interpretados como uma assinatura desta, são úteis quando busca-se saber quais nuvens se sobrepõem. A ideia é trabalhar com nuvens menos densas e selecionar primitivas que carregam mais informação, como os pontos e retas que definem vértices e bordas de objetos. Por outro lado, diz-se que o descritor é local, quando a primitiva geométrica recebe uma assinatura em função da sua vizinhança, e.g. normais dos pontos vizinhos, curvaturas, histogramas etc. Estes são úteis para estabelecer pontos correspondentes entre nuvens (pontos homólogos).

Portanto, um descritor é uma função matemática que opera sobre a vizinhança de um ponto/reta/plano ou sobre a nuvem toda. Composições entre descritores globais e locais são possíveis e comuns, podendo um operar sobre o outro para garantir maior eficácia. A busca por correspondências entre nuvens exige que os

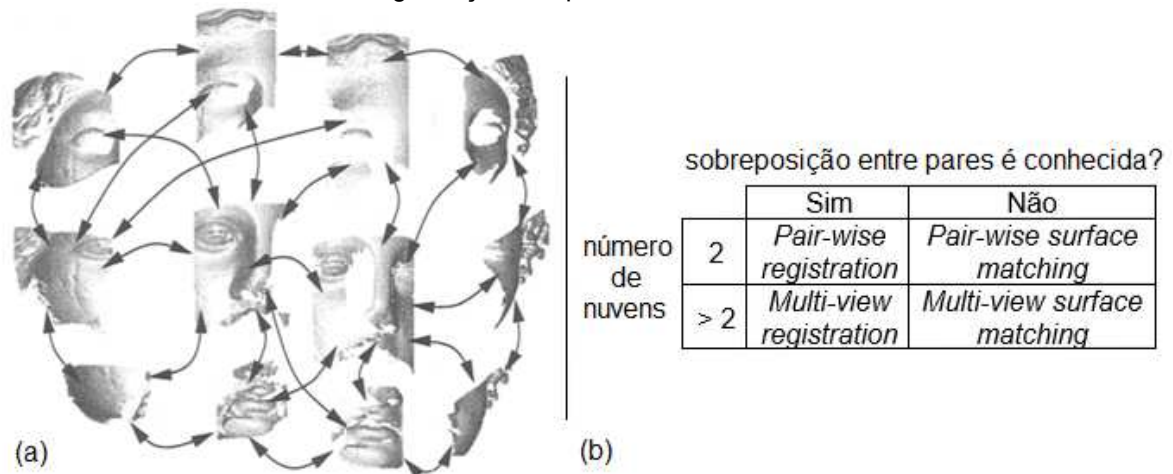
descritores sejam invariantes a rotações e translações, o que garante sucesso no registro local, mesmo quando as nuvens estão arbitrariamente orientadas no espaço.

Uma vez determinadas as transformações entre os pares de nuvens de pontos, parte-se para a otimização global, isto é, as transformações em pares são compostas para uma única origem, chamadas poses. A otimização das poses é a otimização global. Para dados de LST, a literatura costuma apresentar soluções separadas em duas etapas, primeiro, uma abordagem de registro local (aos pares) é definida, e então uma estratégia de otimização global é aplicada (DONG et al. 2018), (PAVAN, DOS SANTOS e KHOSHELHAM, 2020) e (THEILER et al. 2015).

O uso de grafos $G = (Vértices, Arestas)$ é comum na etapa de otimização global das transformações, pois esse modelo representa cada posição do sensor por um vértice e cada transformação por uma aresta. Um dos primeiros a adotar essa estratégia em otimização global de nuvens de pontos foi (LU e MILIOS, 1997). Os autores aplicaram o MMQ para propagar variâncias na matriz adjacente, que representa o grafo e suas relações de vizinhança. Borrmann et al. (2008) adaptaram este método para o caso 3D de ajuste das translações.

É importante destacar que estes trabalhos focam apenas no modelo matemático que distribui o erro ao longo do circuito. Considera-se conhecido o caminho entre as nuvens que apresentam sobreposição, isto é, quais pares foram bem-sucedidos no registro local. Trabalhos que investigam quais nuvens se sobrepõem e qual o melhor circuito a ser utilizado, como o de (HUBER e HEBERT, 2003), enquadram-se em outra categoria, chamada *multi-view surface matching* e de complexidade mais elevada. A Figura 5 (a) apresenta uma visão do problema de otimização com múltiplos caminhos entre nuvens de pontos, em (b) tem-se um resumo adaptado de (idem, 2003) que categoriza os problemas discutidos até aqui em quatro nomenclaturas.

Figura 5 - (a) Múltiplos caminhos possíveis de serem utilizados no problema da reconstrução 3D. (b) categorização dos problemas envolvidos.



Fonte: (a) (BENJEMAA e SCHMITT, 1998); (b) (HUBER e HEBERT, 2003).

As soluções discutidas a seguir são todas do tipo *multi-view surface matching*, isto é, resolvem os três problemas principais da reconstrução 3D, são eles: (1) detectar quais pares se sobrepõem, (2) registrar o par e (3) otimizar todos os pares registrados. Neste nosso trabalho, a etapa (3) é o que interessa.

Theiler et al. (2015) criou o descritor K-4PCS (*keypoint 4 Point Congruent Set*) para solucionar o registro local de nuvens do LST. Para isto ele combinou o descritor global 4PCS (*4 Point Congruent Set*) de Aiger et al. (2008) com o SIFT-3D (RUSU & COUSINS, 2011). O SIFT-3D reduz o par de nuvens em *keypoints* enquanto o 4PCS busca conjuntos de 4 pontos aproximadamente coplanares em ambas as nuvens e os descreve por relações geométricas invariantes em \mathbb{R}^3 . Uma métrica é estabelecida e os pontos correspondentes são estimados, permitindo calcular a transformação de coordenadas entre o par.

Para detectar se existe sobreposição entre pares, todos os pares do conjunto de nuvens são testados pelo modelo de registro local, tornando-o exaustivo computacionalmente. Na etapa de refinamento global, Theiler et al. (2015), aborda o problema como um grafo, as arestas e pontos representam as transformações e nuvens, respectivamente. Uma função energia é definida para o grafo e circuitos fechados são encontrados utilizando o algoritmo de busca em profundidade (EVEN, 2011). Sob restrições de fechamento de circuitos, todo o conjunto de poses é refinado globalmente. As translações são otimizadas pelo modelo de Lu e Milios (1997). Finalmente, todos os pares são refinados com o algoritmo ICP ponto-a-plano (CHEN

E MEDIONI, 1992). Utiliza-se a árvore de extensão mínima (*Minimum Spanning Tree - MST*) do grafo nesta etapa.

Visando diminuir o registro exaustivo que Theiler et al. (2015) faz entre todos os pares de nuvens, Dong et al., (2018), propuseram o uso de Vetores de Descritores Agregados Localmente (*Vector of Locally Aggregated Descriptor - VLAD*). Estes vetores medem a similaridade entre varreduras do LST e realizam um registro hierárquico, denominado de HMMR (*Hierarchical Merging based Multiview Registration*). Pela semelhança entre os vetores que descrevem cada nuvem, evita-se explorar caminhos inválidos, sem sobreposição. O registro local é feito com base no descritor Contexto Binário de Forma (*Binary Shape Context - BSP*), que é uma proposta anterior dos autores (DONG et al., 2017). Esse descritor opera em cima de *keypoints* obtidos como sugerido em (MIAN et al., 2010). O resultado é refinado com o ICP *multiview* de Williams e Bennamoun (2001).

Choi et al. (2015) apresentam um método para registro de dados RGB-D, de sensores como o *Kinect*, que incorpora a modelagem de *outliers* de Black e Rangarajan (1996) na função objetiva do registro. Por meio da detecção de lugares revisitados (*loopclosure*) entre as nuvens, a função é otimizada aplicando-se o método **g²o**, para grafos, de Kümmerle et al. (2011). O registro local ocorre utilizando o descritor FPFH (*Fast Point Feature Histogram*), de RUSU et al. (2009) com o algoritmo ICP em abordagem *coarse-to-fine*.

Também trabalhando com descritores FPFH, Weber et al. (2015), refina globalmente as nuvens de pontos geradas pelo *Kinect*. Para isto, os autores utilizam um grafo em que cada aresta/transformação é ponderada inversamente pela quantidade de pontos correspondentes encontrados no descritor FPFH. A premissa é de que arestas com mais pontos em comum apresentarão maior similaridade. Em seguida a MST do grafo é extraída e as nuvens são registradas em abordagem *coarse-to-fine* utilizando o ICP. Uma nuvem âncora é escolhida aleatoriamente como origem global da MST.

No contexto de sensores *Velodyne*, Vlaminck et al. (2019), propuseram uma otimização global que distribui o erro angular aplicando a Interpolação Linear Esférica (SLERP) de (SHOEMAKE, 1985). Com os resíduos do registro local de cada par de nuvens, os autores ponderam cada aresta do circuito, permitindo calcular uma métrica para as poses. Cada pose tem sua rotação interpolada com a rotação *loopclosure*,

que é a resultante do produto de todas as rotações entre pares. O intervalo de interpolação é proporcional à distância para a pose inicial, de forma que as transformações mais próximas do final, as que acumulam mais erros, recebem maior ajuste. O registro local entre as nuvens do sensor é feito pelo algoritmo 4PCS, que então é refinado pelo ICP em abordagem *coarse-to-fine*.

De volta ao contexto do LST, Pavan e Dos Santos (2017) propuseram um método de ajuste simultâneo das rotações aplicando a Decomposição em Valores Singulares (*Singular Value Decomposition* - SVD). Para isto, os autores parametrizam as rotações em quatérnios, em seguida, os organizam na sua forma matricial. Uma hipermatriz é montada com estes quatérnios seguindo a matriz adjacência do grafo de sobreposição. Após decompor a hipermatriz por SVD, cada rotação é globalmente otimizada, e se beneficia de múltiplos caminhos no grafo. As translações são então ajustadas pelo modelo de (LU & MILIOS, 1997). O registro local é feito detectando-se correspondências de planos.

Nesta análise da literatura destacam-se alguns paradigmas na otimização global de nuvens de pontos:

- 1) O emprego de grafos para explorar a sobreposição e ordenamento das nuvens é comum, pois estas estruturas permitem representar caminhos, estabelecer circuitos e ordenar transformações;
- 2) Os registros locais são feitos majoritariamente em abordagem *coarse-to-fine*. O refinamento utiliza alguma variação do algoritmo ICP;
- 3) No registro grosseiro os descritores FPFH (RUSU et al., 2009) e 4PCS (AIGER et al., 2008) são os mais comuns. Normalmente adiciona-se o estimador aleatório RANSAC (FISCHLER E BOLLES, 1981), em uma ou mais das etapas.
- 4) Há uma deficiência de trabalhos na literatura que abordam o refinamento global das rotações. As estratégias para mitigação do erro não tratam o erro em si, apenas o evita, seja por melhores caminhos, num grafo de transformações, ou por meio de estimadores robustos a *outliers*, como o RANSAC.
- 5) O emprego do modelo (LU e MILIOS, 1997) é padrão no refinamento das translações.

Em resumo, todo o processo de refinamento global é customizável em diferentes estágios, pois, há uma grande variedade de algoritmos para se obter a transformação entre pares. Diferentes modelos de rejeição podem ser utilizados e diversas funções otimizadoras que podem ser estabelecidas. Portanto, os trabalhos podem seguir

abordagens radicalmente diferentes. Para elucidar esta revisão, a Tabela 2 mostra um resumo das principais características dos trabalhos apresentados.

Tabela 2 - Resumo dos trabalhos investigados no estado da arte.

Trabalho	Dados utilizados	Modo de detecção de sobreposição entre nuvens	Registro local	Estratégia de refinamento global	Principal Vantagem	Principal Desvantagem
Theiler et al. (2015)	LST	Registros K-4PCS entre todos os pares de nuvens	K-PCS (<i>coarse</i>) + ICP (<i>fine</i>)	Otimização da energia do grafo + <i>Minimum Spanning Tree</i> + Algoritmo Lu e Milios (1997)	Foi testado com grande diversidade de ambientes e funcionou bem em quase todos	É lento, faz todas as combinações possíveis de registros entre nuvens
Choi et al. (2015)	RGB-D Kinect	FPFH + Black; Rangarajan (1996)	FPFH de Ruso et al. (2009) (<i>coarse</i>) + ICP (<i>fine</i>)	Otimização de grafos g^2o de Kümmerle et al. (2011)	Altamente robusto a <i>outliers</i>	Algoritmo para uso em tempo real, não foi testado com dados de LST
Weber et al. (2015)	RGB-D Kinect	Similaridade FPFH + RANSAC	Arun et al. (1987) (<i>coarse</i>) + ICP (<i>fine</i>)	<i>Minimum Spanning Tree</i> + RANSAC	Abordagem simples, usa algoritmos prontos (<i>off-the-shelf</i>)	Computa correspondências FPFH entre todos os pares de nuvens
Pavan e Dos Santos (2017)	LST	Similaridade plano-a-plano + RANSAC	Método de Horn et al. (1987) (<i>coarse</i>) + ICP (<i>fine</i>)	Ajuste linear das rotações por SVD + Algoritmo Lu e Milios (1997)	É rápido, usa SVD e realmente trata o erro das rotações.	Limitado a ambientes antrópicos, estabelece correspondências com planos
Dong et al. (2018)	LST	Similaridade entre VLAD	BSP (<i>coarse</i>) + ICP (<i>fine</i>)	HMMR + ICP de Williamse e Bennamoun (2001).	Não necessita fazer todos os registros entre nuvens	Complexo, utiliza conjuntos de descritores e ICP <i>multiview</i>
Vlaminck et al. (2019)	LiDAR móvel Velodyne	4PCS	4PCS (<i>coarse</i>) + ICP (<i>fine</i>)	SLERP nos lugares revisitados	Adaptação do 4PCS para rodar em vários núcleos	Algoritmo para uso em tempo real, não foi testado com dados de LST

Fonte: o autor (2020).

Dentre os trabalhos, o que mais se assemelha ao proposto aqui é o de Vlaminck et al. (2019). É importante lembrar que não estamos propondo uma solução totalmente automática de reconstrução 3D, a contribuição neste trabalho objetiva apenas a etapa de refinamento global.

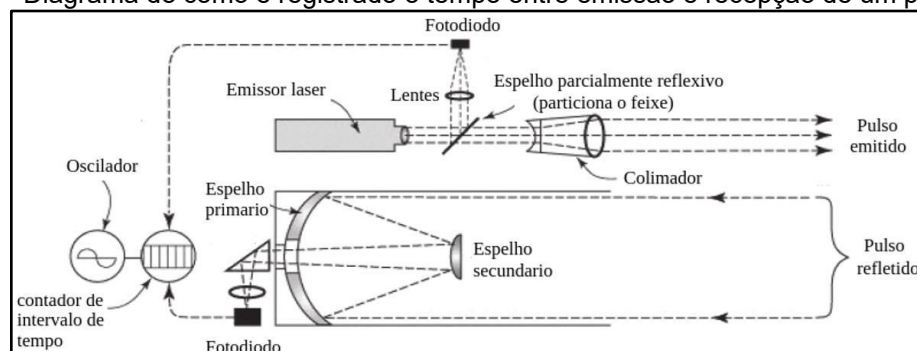
2.1 Tecnologia LIDAR

A tecnologia LIDAR equivale ao RADAR (*Radio Detection And Ranging*), mas em vez de utilizar micro-ondas, trabalha com radiações próximas do espectro visível em forma de LASER, palavra acrônimo de *Light Amplification by Stimulated Emission of Radiation* (Amplificação da Luz por Emissão Estimulada de Radiação). Um laser é um dispositivo que produz radiação eletromagnética com características especiais, tais como ter comprimento de onda bem definido e ondas sincronizadas em fase (FARJAS et al., 2012). Estes aspectos determinam sua propagação no espaço como um feixe reto (ondas colimadas).

Em levantamentos topográficos, onde se busca a representação do terreno por meio de coordenadas tridimensionais, o laser foi utilizado como distanciômetro eletrônico. Posteriormente, a fusão deste dispositivo com os medidores angulares deu origem às estações totais. Com os avanços na eletrônica dos lasers, o processo culminou na tecnologia LIDAR, que opera registrando milhões de pontos de forma rápida e automática (FARJAS et al., 2012).

Mensurar distâncias é, possivelmente, a operação mais básica que um laser pode fazer. De acordo com Shan e Toth (2018), para este fim, eles se subdividem em dois tipos principais, os que medem o tempo entre emissão e recepção do pulso (*Time-of-Flight - ToF*), e calculam a distância pela velocidade da luz; e os que operam contando ciclos de fase entre ondas emitidas e recebidas, os ditos lasers de onda contínua (*Continuos Wave - CW*). Neste trabalho, apenas sistemas LIDAR que utilizam laser *ToF* são apresentados. Para oferecer medidas confiáveis ao nível do metro, sistemas *ToF* medem pico segundos de intervalo utilizando alta frequência (SCHOENKNECHT, 2010). A Figura 6 apresenta um diagrama simplificado sobre como o tempo é registrado em sistemas assim.

Figura 6 - Diagrama de como é registrado o tempo entre emissão e recepção de um pulso laser.



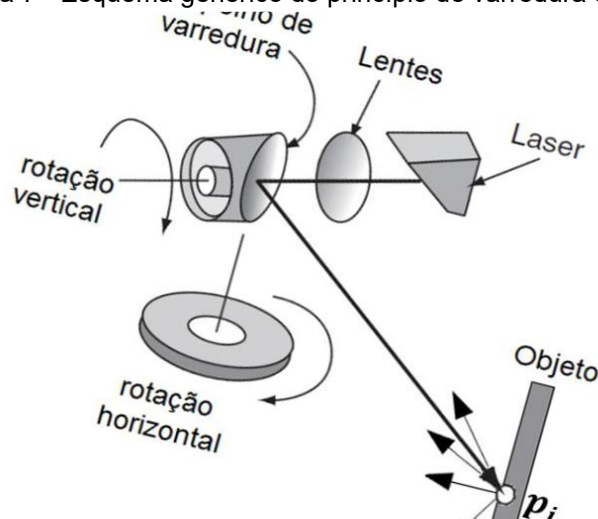
Fonte: adaptada de Shan e Toth (2018, p. 15).

Como pode ser observado na Figura 6, um pulso gerado pelo emissor é dividido em duas partes por um espelho semi-reflexivo. Uma das partes segue em direção ao colimador, é o pulso emitido para fora do sistema. A outra parte do pulso ativa o fotodiodo responsável por transformar a onda em pulso elétrico, que inicia a contagem de tempo pelo oscilador. Quando o pulso emitido encontra uma superfície ele é refletido e retorna ao sistema. Após ser redirecionado pelos espelhos, um primário e outro secundário, o feixe passa por um prisma e é convertido em pulso elétrico por outro fotodiodo, que finaliza a contagem do tempo.

2.1.1 Laser Scanner Terrestre - LST

A princípio, um LST seria qualquer um que não operasse no modo aerotransportado, todavia, não se trata de obedecer apenas a este critério, mas de possuir um funcionamento específico que consiste na rotação de um espelho em cima de uma base estática. O espelho de varredura gira rapidamente na vertical definindo um plano. Com o giro da base, que é mais lento, este plano varre o ambiente ao redor. A Figura 7 mostra o esquema genérico do princípio de varredura em um LST.

Figura 7 - Esquema genérico do princípio de varredura do LST.



Fonte: adaptado de (SHAN e TOTH, 2018).

Na Figura 7, um ponto p_i registrado no sistema de coordenadas local do LST é definido por um ângulo vertical (α), um ângulo horizontal (θ) e uma distância (ρ) calculada pela eq. 1, a saber:

$$\rho = \frac{\Delta t}{2} c \quad (1)$$

sendo Δt o tempo de voo do pulso e c a velocidade da luz no vácuo (299.792.458 m/s). A informação registrada pelo sistema está em coordenadas polares, para obter-se pontos no sistema cartesiano aplicam-se as transformações na eq. 2, como segue:

$$\begin{aligned} x &= \rho \cos \alpha \cos \theta \\ y &= \rho \sin \alpha \cos \theta \\ z &= \rho \sin \theta \end{aligned} \quad (2)$$

Além das coordenadas polares, alguns sistemas registram a intensidade (I) do pulso refletido. No caso de sistemas com câmeras digitais acopladas, também é salva a informação RGB, que é associada a cada ponto do objeto contido na superfície física. Logo, um ponto p_i qualquer, pode ser individualizado por qualquer uma das seguintes tuplas:

$$\begin{aligned} p_i &= \{x, y, z\} \\ p_i &= \{x, y, z, I\} \\ p_i &= \{x, y, z, I, r, g, b\} \end{aligned} \quad (3)$$

2.2 Quatérnios

Quatérnios são definidos na matemática como números hipercomplexos de quatro dimensões. Foi neste contexto que Hamilton os descobriu enquanto procurava por uma álgebra que fizesse com vetores 3D aquilo que os números complexos fazem com os vetores 2D, isto é, os rotacione por meio de multiplicações (CONWAY e SMITH, 2003). Em linguagem formal, Hamilton procurava por uma álgebra de divisão normada (NEVES, 2008).

2.2.1 Propriedades axiomáticas dos quatérnios

Em 1835 Hamilton encontrou seus números, hoje denotados pelo conjunto H em sua homenagem. Para isto, Hamilton definiu uma base de quatro eixos ortogonais entre si, com 3 unidades imaginárias e uma real. Assim, a base canônica dos quatérnios é formada por i , j , k e 1 , sendo os três primeiros elementos números complexos definidos por $i = j = k = \sqrt{-1}$. Uma vez que as bases representam a ideia abstrata de ortogonalidade, os múltiplos da unidade i não podem ser somados com

números reais, nem com outros múltiplos de j ou k , pois variam em uma direção independente. Essa ideia deu origem às bases vetoriais modernas que utilizamos hoje na álgebra linear, e levou à concepção atual que temos dos vetores e suas propriedades de produto interno e externo. As propriedades dos quatérnios são construídas a partir das seguintes multiplicações, a saber (NEVES, 2008):

$$\begin{aligned} ij &= k = -ji \\ jk &= i = -kj \\ ki &= j = -ik \end{aligned} \quad (4)$$

Os produtos não são comutativos, dado que:

$$\begin{aligned} ji &= -k \\ kj &= -i \\ ik &= -j \end{aligned} \quad (5)$$

Vemos que a multiplicação dos elementos i , j e k é idêntica ao produto externo dos versores i , j , k da base \mathbb{R}^3 . A Figura 8 mostra a matriz de multiplicação dos elementos da base quaterniônica.

Figura 8 - Matriz de multiplicação entre os elementos da base quaterniônica

×	1	i	j	k
1	1	i	j	k
i	i	-1	k	-j
j	j	-k	-1	i
k	k	j	-i	-1

Fonte: Da Silva et al. (2012)

Pode-se representar-se um quatérnio q como (SILVA, 2014):

$$q = [q_0, q_x, q_y, q_z] \quad (6)$$

ou

$$q = q_0 + q_x i + q_y j + q_z k$$

Na eq. (6) $q_0, q_x, q_y, q_z \in \mathbb{R}$. Outra forma mais concisa de representar um quatérnio consiste em pensar nele como a soma de um escalar com um vetor. O escalar é dado pela parte real e o vetor pela parte imaginária, como segue:

$$q = s, \vec{v} \quad (7)$$

ou

$$q = s + \vec{v}$$

A adição de dois quatérnios $p, q \in H$ atua somando apenas as componentes de bases iguais, como segue:

$$q + p = q_0 + p_0 + (q_x + p_x)i + (q_y + p_y)j + (q_z + p_z)k \quad (8)$$

O elemento neutro e o oposto da adição, na equação acima, são dados por $q = 0 + 0i + 0j + 0k$ e $q = -q_0 - q_x i - q_y j - q_z k$, respectivamente, o que preserva as propriedades de associação e comutação na adição. A multiplicação de dois quatérnios q e $p \in H$ pode ser feita utilizando a propriedade distributiva. Apresentamos a forma matricial conforme Horn (1987), que escreve um quatérnio em forma de matriz e o outro em forma de vetor coluna. Como a multiplicação não é comutativa, duas multiplicações podem ser feitas entre 2 quatérnios, uma multiplicação pela esquerda (eq. 9) e outra pela direita (eq. 10). A multiplicação pela direita apenas inverte a submatriz em destaque, como segue:

$$[q] * p = \begin{bmatrix} q_0 & -q_x & -q_y & -q_z \\ q_x & q_0 & -q_z & q_y \\ q_y & q_z & q_0 & -q_x \\ q_z & -q_y & q_x & q_0 \end{bmatrix} \begin{bmatrix} p_0 \\ p_x \\ p_y \\ p_z \end{bmatrix} \quad (9)$$

$$q * [p] = \begin{bmatrix} q_0 & -q_x & -q_y & -q_z \\ q_x & q_0 & q_z & -q_y \\ q_y & -q_z & q_0 & q_x \\ q_z & q_y & -q_x & q_0 \end{bmatrix} \begin{bmatrix} p_0 \\ p_x \\ p_y \\ p_z \end{bmatrix} \quad (10)$$

O produto interno de dois quatérnios é a soma do produto de suas componentes:

$$\langle p, q \rangle = p_0 q_0 + p_x q_x + p_y q_y + p_z q_z \quad (11)$$

A magnitude de um quatérnio é a dada pela sua norma-2, como segue:

$$\|q\| = \sqrt{q_0^2 + q_x^2 + q_y^2 + q_z^2} \quad (12)$$

Um quatérnio é dito unitário se a sua norma é igual a 1. Se “.” denota o produto interno de dois quatérnios, então q será unitário se:

$$\sqrt{q \cdot q} = 1 \quad (13)$$

O conjugado de um quatérnio consiste em trocar o sinal da sua parte vetorial:

$$\begin{aligned} \bar{q} &= q_0 - q_x i - q_y j - q_z k \\ &\text{ou} \\ \bar{q} &= (s, -\vec{v}) \end{aligned} \quad (14)$$

A matriz associada a um quatérnio conjugado é simplesmente a transposta do quatérnio original, e ambas são ortogonais (ZHANG, 1997). Se são ortogonais, então

o produto entre elas é a unidade matricial, isto é $QQ^T = q\bar{q} * I$, onde I é a matriz identidade 4x4. Logo, o produto do quatérnio pelo seu conjugado é um número real, similarmente ao que ocorre com números complexos, e disto decorre que um quatérnio não nulo possui seu inverso multiplicativo dado por (HORN, 1987):

$$q^{-1} = \frac{\bar{q}}{\|q\|^2} = \frac{q_w - q_x i - q_y j - q_z k}{q_w^2 + q_x^2 + q_y^2 + q_z^2} \quad (15)$$

A equação (15) define a divisão entre quatérnios, ela mostra que para quatérnios unitários, o inverso do quatérnio será igual ao conjugado, pois $\|q\|^2 = 1$. O produto interno de quatérnios (ou o ângulo entre eles) também é conservado devido à ortogonalidade das matrizes, pois:

$$(qp) \cdot (qr) = (Qp) \cdot (Qr) = (Qp)^T(Qr) \quad (16)$$

e:

$$(Qp)^T(Qr) = p^T Q^T Qr = p^T(q \cdot q)Ir \quad (17)$$

então:

$$(Qp)^T(Qr) = p^T Q^T Qr = p^T(q \cdot q)Ir \quad (18)$$

O lado direito da Equação (18) se resume em $(p \cdot r)$, se q for um quatérnio unitário. Neste caso especial tem-se que:

$$(pq) \cdot (pq) = (p \cdot p)(q \cdot q) \quad (19)$$

Portanto, a magnitude do produto interno é igual ao produto das magnitudes. Além disso:

$$(pq) \cdot r = p \cdot (rq) \quad (20)$$

2.2.2 Quatérnios e rotações no espaço 3D

Euler demonstrou que toda composição de rotações, por mais complicada que seja, pode ser substituída por uma única rotação no espaço (BIASI E GATTASS, 2007). Por exemplo, dado um ponto em \mathbb{R}^3 representado pelo vetor $\vec{d} = (d_x \ d_y \ d_z)^T$ e seja $r_{(\theta, \vec{n})}$ uma rotação de ângulo θ , no sentido anti-horário, em torno do eixo unitário $\vec{n} = (n_x \ n_y \ n_z)^T$ que passa pela origem. O vetor resultante da aplicação da rotação $r_{(\theta, \vec{n})}$ em \vec{d} será dado por (Idem, 2007):

$$r(\vec{d}) = (\cos\theta)\vec{r} + (1 - \cos\theta)(\vec{n} \cdot \vec{r})\vec{n} + (\sin\theta)(\vec{n} \times \vec{r}) \quad (21)$$

Como rotações são operadores lineares, não alteram a magnitude dos vetores, logo:

$$\|r(\vec{d})\| = \|\vec{d}\| \quad (22)$$

Se dois vetores recebem a mesma rotação, então o ângulo entre eles permanece o mesmo, logo, as rotações também preservam o produto interno. Como os quatérnios unitários também preservam o produto interno (eq. 20), é possível representar as rotações por meio de quatérnios unitários se existir alguma forma de mapear quatérnios em vetores (HORN, 1987).

Isto pode ser feito de maneira muito simples, basta zerar a parte real e trabalhar com quatérnios imaginários puros. Da mesma forma, um escalar também pode ser representado por um quatérnio apenas com a parte real. Então, mapear quatérnios imaginários puros neles mesmos (vetores para vetores), pode ser feito pela seguinte composição de multiplicações:

$$\vec{d}_R = q\vec{d}\bar{q} \quad (23)$$

Sendo q o quatérnio que codifica a rotação $r_{(\theta, \vec{n})}$, dado por:

$$q = \cos \frac{\theta}{2} + \text{sen} \frac{\theta}{2} (n_x i + n_y j + n_z k) \quad (24)$$

Horn (1987) mostrou essa propriedade escrevendo o quatérnio q e seu conjugado \bar{q} na forma matricial, a saber:

$$q\vec{d}\bar{q} = (Q\vec{d})\bar{q} = Q^T(Q\vec{d}) = (Q^T Q)\vec{d} \quad (25)$$

Onde:

$$Q^T Q = \begin{bmatrix} q \cdot q & 0 & 0 & 0 \\ 0 & q_0^2 + q_x^2 - q_y^2 - q_z^2 & 2(q_x q_y - q_0 q_z) & 2(q_x q_z + q_0 q_y) \\ 0 & 2(q_y q_x + q_0 q_z) & q_0^2 - q_x^2 + q_y^2 - q_z^2 & 2(q_y q_z - q_0 q_x) \\ 0 & 2(q_z q_x + q_0 q_y) & 2(q_z q_y + q_0 q_x) & q_0^2 - q_x^2 - q_y^2 + q_z^2 \end{bmatrix} \quad (26)$$

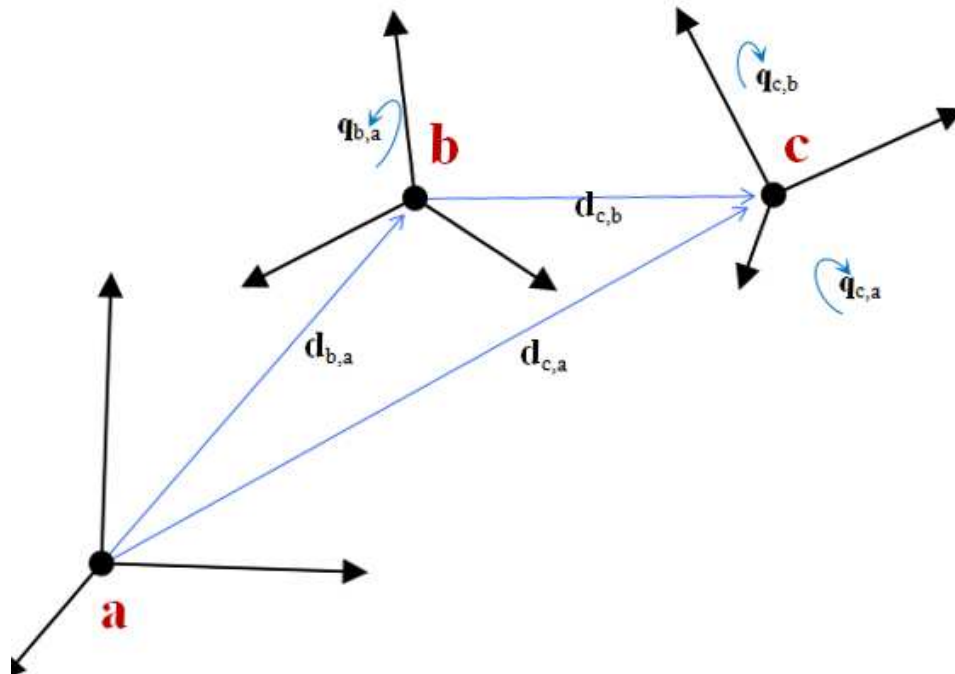
Resumindo, para rotacionar o vetor \vec{d} por um ângulo θ ao redor de um eixo $\vec{n} = n_x i + n_y j + n_z k$ no sentido anti-horário, primeiro escreve-se o ponto como um quatérnio imaginário puro $p_i = 0 + ix + jy + kz$, depois normaliza-se o eixo \vec{n} e calcula-se o quatérnio q dado por:

$$q = \cos \frac{\theta}{2} + \text{sen} \frac{\theta}{2} (\vec{n}) \quad (27)$$

O vetor rotacionado será um quatérnio com a parte real nula, dado pela eq. 23. Esta operação define o operador ortogonal quatérnio $d_q: \mathbb{R}^3 \rightarrow \mathbb{R}^3$ que apresenta um

homomorfismo com o grupo $SO(3)$ ao qual pertencem as matrizes de rotação (SHOEMAKE, 1985), mais detalhes sobre isso serão dados na seção 2.4. A composição de rotações pela multiplicação em cadeia, como ocorre nas matrizes, também é válida para os quatérnios, sendo feita naturalmente pela multiplicação sucessiva destes. Seja $q_{b,a}$ um quatérnio representando a rotação do sistema a para o sistema b e $q_{c,b}$ outro quatérnio representando a rotação do sistema b para o sistema c , o quatérnio dado por $q_{c,a} = q_{b,a}q_{c,b}$ rotaciona o sistema a em torno de um único eixo diretamente para o referencial c , como mostra a Figura 9.

Figura 9 - Composição de rotações entre sistemas de coordenadas por quatérnios.



Fonte: Tasora et al. (2015).

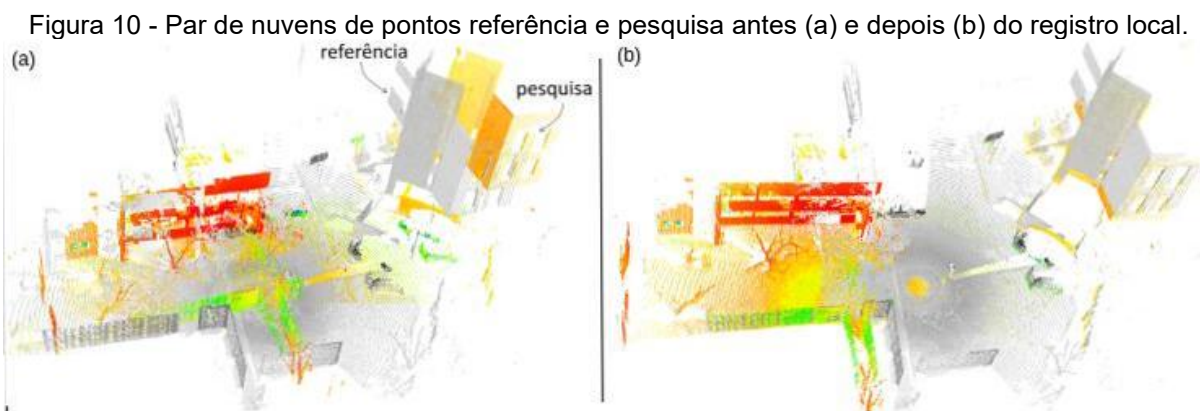
A vantagem direta desta solução é que quatérnios são mais eficientes para representar rotações, pois usam apenas 4 parâmetros em vez dos 9 da matriz de rotação, portanto, operam com menos aritmética e são mais estáveis numericamente. Matrizes necessitam obedecer 6 restrições para representarem rotações, quatérnios só precisam obedecer à de ser unitário. Assim como as matrizes, quatérnios são imunes a singularidades de *Guimbal Lock* (DAM et al., 1998). Esta representação compacta de rotações leva a fórmulas simples e elegantes de interpolação. Mais detalhes sobre esse tópico são apresentados na seção 2.4.

2.3 Registro de nuvens de pontos 3D

O registro de nuvens de pontos pode ser dividido em duas partes, o registro grosseiro (*coarse*) e o registro fino (*fine*). A principal diferença entre eles é que o primeiro usa uma quantidade de informação reduzida para calcular uma transformação inicial aproximada, o segundo tipo usa mais informação – mais pontos da nuvem – para calcular resultados melhores, mas isto torna-o sujeito mínimos locais na sua solução, o que é o equivalente a calcular transformações que sobrepõem as nuvens erroneamente.

Embora seja comum uma abordagem *coarse* para inicializar o registro fino, é importante dizer que alguns modelos não necessitam de uma transformação inicial. É o caso da Transformação por Distribuição de Normais (*Normal Distribution Transform - NDT*) de Biber e Strasser (2003). Da mesma forma há exceções para o registro grosseiro, atualmente existem métodos capazes de alcançar qualidade comparável ao ICP, de forma rápida e sem inicialização, como o caso do algoritmo *Fast Global Registration* (ZHOU et al., 2016), que será utilizado neste trabalho.

Em todos os casos a transformação que se procura é a de corpo rígido 3D, pois dispositivos LIDAR medem distâncias absolutas no espaço, uma distorção de escala não seria bem-vinda. Essa transformação sempre é obtida por algum critério de otimização, tal como o MMQ. A Figura 10 mostra um exemplo do resultado obtido antes e após aplicar o registro local em um par de nuvens de pontos, apenas a nuvem de pesquisa é deslocada.



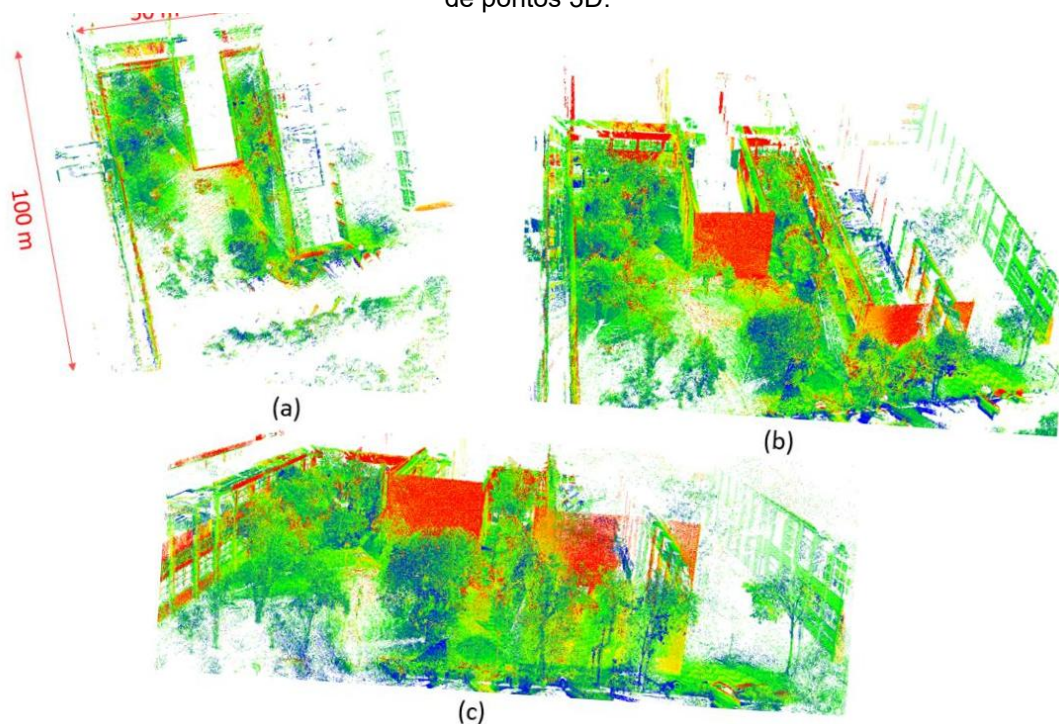
Fonte: o autor (2020).

No caso do LST, cada estação ocupada no terreno define uma origem, mas como esclarecido no início, oclusões impedem a varredura completa dos objetos. Logo, são

necessárias várias estações com diferentes pontos de vista para conseguir mapear uma superfície por completo. Se forem ocupadas n estações, no mínimo $n - 1$ transformações devem existir para conectar todas elas. Mas as transformações podem criar ciclos, caso haja sobreposições múltiplas. Na nomenclatura dos grafos, isto dá origem a cliques, que são conjuntos de 3 vértices ou mais totalmente conectados.

O objetivo último de todo o registro é transformar o conjunto de nuvens para um único sistema de coordenadas, isto significa, quase sempre, compor transformações entre sistemas, o que acumula erros residuais, daí a necessidade de uma etapa que optimize as transformações reduzindo esse erro globalmente. A Figura 11 mostra um modelo 3D globalmente consistente, obtido após a etapa de refinamento global de nuvens de pontos 3D.

Figura 11 - Modelo 3D globalmente consistente obtido após a etapa de refinamento global de nuvens de pontos 3D.



Vista superior (a), e arbitrárias em (b) e (c).

Fonte: o autor (2020).

2.3.1 Registro local de nuvens de pontos 3D

No formalismo matemático, o problema do registro local é definido como: dado um par de nuvens de pontos M (pesquisa) e S (referência) em um espaço dimensional

finito \mathbb{R}^d , encontrar a transformação T que quando aplicada a M minimize a distância entre M e S . Em 3 dimensões tem-se $d = 3$. Escrito como função que mapeia \mathbb{R}^d em \mathbb{R}^d , deseja-se:

$$T^* = \arg \min_{T \in \tau} \text{dist}(T(M), S) \quad (28)$$

sendo τ o conjunto de todas as transformações T existentes entre M e S , e T^* a transformação ótima desejada. A função distância (ou função objetiva) mais utilizada é o somatório dos resíduos ao quadrado, calculados com a norma euclidiana entre pontos. Isto torna o problema do tipo Mínimos Quadrados, a saber:

$$T^* = \arg \min \sum_{i=1}^{num} \|T(m_i) - s_i\| \quad (29)$$

sendo m_i e s_i pontos correspondentes presentes nas nuvens M e S , e num a quantidade de pontos correspondentes estabelecidos. A transformação T^* trabalhada no escopo deste trabalho é especificamente a de corpo rígido 3D. Os parâmetros independentes desta transformação são 6, três de translação e três de rotação, um para cada eixo cartesiano. A forma comum de representação da transformação é por uma matriz quadrada 4x4, chamada de matriz de transformação homogênea, que contém a matriz de rotação e um vetor de translação, organizados como segue:

$$\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \overbrace{\begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix}}^T \begin{bmatrix} X' \\ Y' \\ Z' \\ 1 \end{bmatrix} \quad (30)$$

Esta forma de organização permite aplicar a matriz $R_{3 \times 3}$, composta pelos elementos r_{ij} , e o vetor translação $t = [t_x \ t_y \ t_z]^T$ em um vetor p qualquer pela multiplicação direta de $T(R, t)$ por p , desde que p também esteja definido em coordenadas homogêneas, isto é: $p = [X', Y', Z', 1]$. A matriz de rotação R é, na verdade, a composição de três rotações em torno de cada um dos eixos cartesianos, isto é, sejam $\theta_x, \theta_y, \theta_z$ ângulos de Euler em torno do terno x, y, z e:

$$R_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta_x) & \text{sen}(\theta_x) \\ 0 & -\text{sen}(\theta_x) & \cos(\theta_x) \end{bmatrix} \quad (31)$$

$$R_y = \begin{bmatrix} \cos(\theta_y) & 0 & -\text{sen}(\theta_y) \\ 0 & 1 & 0 \\ \text{sen}(\theta_y) & 0 & \cos(\theta_y) \end{bmatrix} \quad (32)$$

$$R_z = \begin{bmatrix} \cos(\theta_x) & \text{sen}(\theta_x) & 0 \\ -\text{sen}(\theta_x) & \cos(\theta_x) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (33)$$

as respectivas matrizes de rotação para cada eixo. Uma das formas de se obter \mathbf{R} é por:

$$\mathbf{R} = R_z R_y R_x \quad (34)$$

Como o produto de matrizes não é comutativo, o mapeamento entre ângulos de Euler e matrizes de rotação 3D não é único. Para obtê-los a partir de uma matriz de rotação, sem ambiguidade, é necessário conhecer a ordem como foi definida a multiplicação das matrizes, que podem ser de até 3! formas. Como rotações também podem ser definidas de duas formas – intrínsecas ou extrínsecas – aumenta-se para 12 as possibilidades de mapeamento entre ângulos de Euler e matrizes de rotação. (SLABAUGH, 1999).

Ângulos de Euler também não estabelecem uma métrica no espaço das rotações (HUYNH, 2009), portanto interpolações entre eles não são possíveis. Além disso, utilizá-los é estar sujeito ao problema de *Gimbal Lock*. Diante disso tudo, uma vez que se passe ao espaço das matrizes de rotação, o retorno a ângulos de Euler não é recomendado.

Como descrito no início do capítulo, dentro dos métodos de registro grosseiro observados na revisão de literatura, os algoritmos 4PCS (AIGER et al. 2008) e o descritor FPFH (RUSU et al., 2009) são os mais comuns. A seguir se apresentam mais detalhes do descritor FPFH, que por ser adotado no método de registro grosseiro FGR, será utilizado indiretamente neste trabalho.

2.3.2 *Fast Point Feature Histogram (FPFH)*

O descritor local FPFH (RUSU et al., 2009) é a versão mais rápida do *Point Feature Histogram* (Histograma Característico do Ponto - PFH) que foi apresentado primeiramente em (RUSU et al., 2008). Este descritor local individualiza a presença do ponto em uma nuvem de pontos utilizando as normais da sua vizinhança, permitindo estabelecer correspondências (pontos homólogos) entre pares de nuvens. Como o próprio nome diz, o PFH utiliza histogramas para identificar pontos, isto é vantajoso, porque histogramas são menos ambíguos que vetores de escalares calculados por outras características.

O algoritmo opera da seguinte forma, dado um ponto p e seus k vizinhos dentro de um raio r , (1) calculam-se as todas as normais dos pontos interiores utilizando a Análise de Componentes Principais (*Principal Component Analysis - PCA*) (HOPPE et al. 1992); (2) para cada par de ponto p_s, p_t e suas normais n_s, n_t (sendo p_s o ponto com o menor ângulo entre sua normal e a linha que conecta os dois pontos) se define um sistema de eixos u, v, w chamado *Darboux*, da seguinte forma: $u = n_s$, $v = (p_t - p_s) \times u$ e $w = u \times v$. (3) calculam-se as variações angulares de n_s e n_t como segue (RUSU et al. 2008):

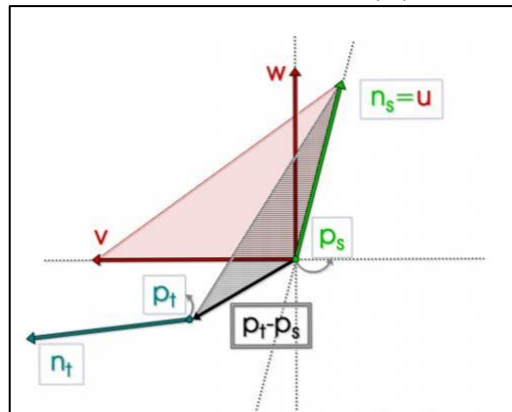
$$\alpha = v \cdot n_t \quad (35)$$

$$\phi = \frac{u \cdot (p_t - p_s)}{\|p_t - p_s\|} \quad (36)$$

$$\theta = \arctan\left(\frac{w \cdot n_t}{u \cdot n_t}\right) \quad (37)$$

A Figura 12 apresenta o esboço geométrico do sistema Darboux definido em um par de pontos p_s, p_t , bem como as normais destes pontos.

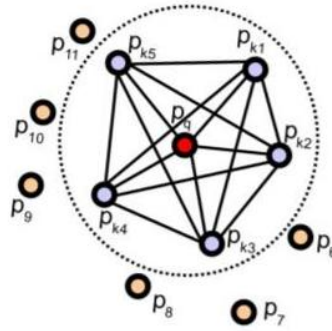
Figura 12 - Eixos do sistema *Darboux*: vetores u, v, w com origem no ponto p_s .



Fonte: Rusu et al. (2008).

Para cada um dos valores nas equações 36, 37 e 38 é construído um histograma subdividido em 5 partes. Como são 3 variáveis, o histograma é tridimensional e pode ser interpretado como um cubo subdividido em $5^3 = 125$ partes, portanto, o ponto é caracterizado como um vetor de 125D. O histograma construído utiliza todos os pares dentro da área de influência, a Figura 13 apresenta essa região para um ponto qualquer p_q , em vermelho. Os pontos vizinhos, p_k , estão em azul.

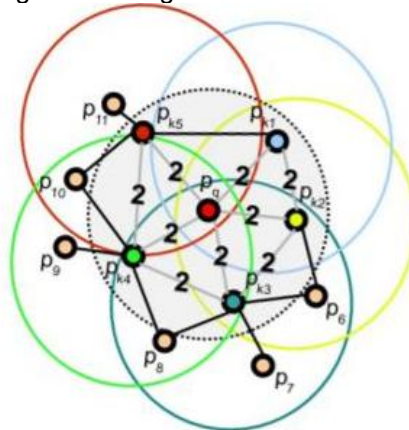
Figura 13 - Diagrama da região de influência do descritor PFH.



Fonte: Rusu et al. (2009).

O problema desta abordagem é a conexão entre todos os pontos da vizinhança, que eleva a complexidade computacional do algoritmo para $O(k^2)$, sendo k a quantidade de pontos. Para contornar essa desvantagem, Rusu et. al (2009), propõem uma variação que tem complexidade linear $O(k)$ chamada FPFH (*Fast Point Feature Histogram*), e para isto os autores simplificaram o cálculo do histograma considerando os vizinhos diretos e indiretos do ponto p_q , mas sem interligar todos (Figura 14). Os valores calculados são ponderados segundo sua conexão com o ponto p_q , conexões diretas recebem o dobro de peso em relação às outras.

Figura 14 - Diagrama da região de influência do descritor FPFH.



Círculo pontilhado: raio do ponto de referência p_q . Círculos coloridos: raio dos vizinhos. Arestas cinzas são ligações diretas, recebem peso 2. Arestas pretas são ligações indiretas, recebem peso 1. A ligação é direta se um ponto estiver na vizinhança do outro e ambos estiverem na vizinhança do ponto principal.

Fonte: (RUSU et. al., 2009).

O descritor opera em todos os pontos da nuvem e retorna aqueles que normalmente guardam mais informação (menor entropia), como pontos de bordas e cantos. Após definir-se uma métrica/filtragem as correspondências são estabelecidas

e então um método de registro é aplicado para encontrar a transformação que registra uma nuvem de pontos na outra (RUSU et al. 2009). Os autores utilizam a amostragem RANSAC na etapa grosseira e em seguida aplicam o ICP, ou seja, também definem uma abordagem *coarse-to-fine* com o descritor.

2.3.3 *Iterative Closest Point (ICP)*

No algoritmo ICP, para cada ponto na nuvem de referência é encontrado o ponto mais próximo na nuvem de pesquisa, esta etapa consiste no estabelecimento de correspondências e assume que os pontos mais próximos entre as nuvens são homólogos. Rusinkiewicz e Levoy (2001) citam várias abordagens para estabelecimento de correspondências que otimizam o algoritmo ICP, as mais eficientes se baseiam na ponderação de pontos pelas suas distâncias e intensidades. O ICP original não considera esta etapa.

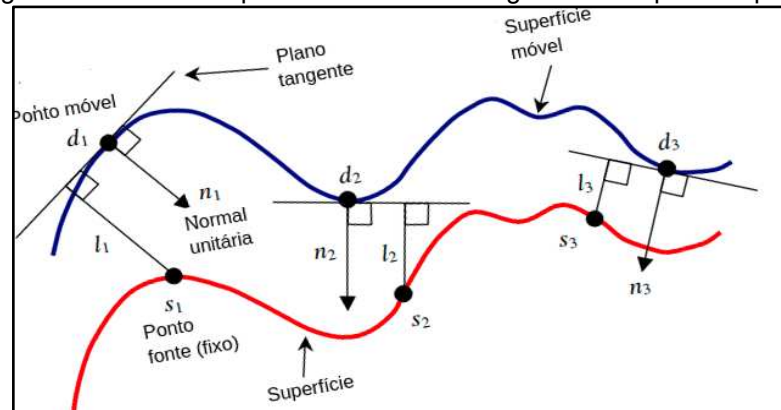
Na segunda etapa do ICP é estimada a melhor combinação de R e t aplicando o método de Horn et al. (1987), que aplica a Decomposição em Valores singulares (SVD) na matriz covariância-cruzada do par de nuvens. O autovetor associado ao maior autovalor da matriz é o quatérnio unitário q que representa matriz de rotação ótima R^* procurada. Com esta rotação a nuvem de pesquisa é orientada na mesma direção da nuvem de referência. A translação t é calculada em seguida pela simples diferença entre os centroides das nuvens.

A transformação de corpo rígido $T(R, t)$, então definida, é aplicada na nuvem de pesquisa, aproximando-a da nuvem de referência. Este processo é repetido até que se atinja o número máximo de iterações ou o RMSE das distâncias entre os pontos das duas nuvens caia abaixo de um limiar pré-definido. A solução por decomposição matricial utiliza todos os pontos da MVC, logo não é robusta a *outliers*, mas é mais vantajosa que o tradicional ajustamento paramétrico (MMQ), pois resolve o sistema em uma iteração.

Como as equações envolvidas na transformação de corpo rígido 3D não são lineares e as correspondências pontuais não são garantidas, muitos autores utilizam outras feições geométricas para aumentar a eficiência do método. Uma destas soluções é a proposta por Chen e Medioni (1992), chamada de ICP ponto-a-plano por minimizar distâncias entre pontos da nuvem fixa e planos da nuvem móvel (Fig. 15).

Os planos são tangentes a pontos móveis, que correspondem a pontos fixos na nuvem de referência.

Figura 15 - Distância que é minimizada no algoritmo ICP ponto-a-plano.



Correspondências ocorrem entre pontos e planos.

Fonte: adaptada de Burgard et. al. (s/ ano).

A abordagem ponto-a-plano também não é fechada – iterações sem garantia de convergência – mas apresenta uma taxa de convergência/iteração maior que a do ICP ponto-a-ponto (BESL E MCKAY, 1992), principalmente com nuvens de ambientes antrópicos, onde a presença de feições planas é considerável. Chen e Medioni (1992) linearizam as equações ao considerar superfícies que diferem por pequenas rotações, isto é, assumem $\sin(\theta) = \theta$ e $\cos(\theta) = 1$. A solução normalmente é implementada utilizando o método de Levenberg-Marquadt, o que acelera a convergência do algoritmo (RUSINKIEWICZ e LEVOY, 2001).

Os planos utilizados nesta modalidade de ICP implicam no cálculo das normais de cada ponto. Este é um passo crítico que depende da quantidade de vizinhos k , definidos para cada ponto, ou da escolha de um raio r , que considere os pontos interiores como vizinhos. O cálculo da normal é feito pela técnica citada PCA, onde a direção de menor dispersão dos pontos – autovetor associado ao menor autovalor – representa a normal daquela vizinhança (HOPPE et al. 1992).

2.4 Interpolação Linear Esférica (SLERP)

A Interpolação Linear Esférica (SLERP) foi originalmente proposta na animação gráfica por Shoemake (1985) como solução do problema de interpolação entre duas rotações mantendo a velocidade angular do objeto constante. Como esclarecido no

seu artigo, a topologia do grupo $SO(3)$ é uma superfície esférica, pois o conjunto de todas as orientações no espaço define uma esfera, chamada esfera gaussiana. Em $SO(3)$, S significa “*Special*” e $O(3)$ vem da definição de grupos de matrizes ortonormais $O(n) = \{matrizes(n \times n) \mid O^T O = I\}$. Significa que o grupo $SO(3)$ é formado por todas as matrizes 3x3 que tem determinante 1 e são ortonormais, isto é, todas as matrizes de rotação possíveis.

O subconjunto dos quatérnios unitários, chamado de S^3 , também define um grupo, e este grupo tem a mesma topologia de uma esfera, ou melhor, uma hiperesfera, pois um quatérnio é composto de 4 números. Não há como visualizar esta esfera, pois está imersa em \mathbb{R}^4 . Ela é chamada de S^3 , na topologia, porque é uma superfície 3D imersa no espaço \mathbb{R}^4 . Para entender, basta fazer uma analogia com a esfera que vemos no mundo tridimensional, que é uma superfície 2D imersa em \mathbb{R}^3 , chamada S^2 . Na terminologia dos espaços topológicos, o círculo também é uma esfera, e recebe o nome de S^1 , pois tem dimensão 1, embora imerso no espaço \mathbb{R}^2 . Todas estas entidades têm área e volume n-dimensionais bem definidos matematicamente para qualquer dimensão (SMITH e VAMANAMURTHY, 1989).

O importante a entender aqui é que existe uma conexão entre os espaços $SO(3)$ e S^3 . Na álgebra abstrata, estes espaços são chamados de grupos. Esta conexão é representada como $SO(3):S^3 \rightarrow 1:2$. Significa que cada ponto da esfera $SO(3)$ mapeia para dois pontos na esfera S^3 . Em termos práticos, isso significa que para cada matriz de rotação R , existem dois quatérnios que a representam: q e $-q$ (DAM et al., 1998). Na linguagem da álgebra de grupos se diz que $SO(3)$ é duplamente coberto por S^3 .

Por causa desta bijeção entre $SO(3)$ e S^3 , a razão entre as áreas das esferas S^2/S^3 é exatamente $\pi/2$. A interpretação geométrica disso é que os ângulos em 4D estão duplicados, uma volta neste espaço corresponde a duas no mundo tridimensional (CONWAY e SMITH, 2003). Se lembrarmos da eq. 27, quando se quer representar uma rotação de ângulo θ por um quatérnio, seu valor é dividido pela metade. Uma outra propriedade interessante desta conexão entre os grupos $SO(3)$ e S^3 é o truque de Dirac¹.

Tudo isto é para mostrar que a interpolação entre duas rotações, dadas por dois quatérnios distintos, é exatamente interpolar entre dois pontos na superfície de uma

¹ *Dirac's string trick*. <<http://ariwatch.com/VS/Algorithms/DiracStringTrick.htm>>. Acesso em: 07/01/2021

esfera 4D, com o cuidado de evitar a ambiguidade $(q, -q)$. Como veremos, a técnica de interpolação SLERP define um arco de circunferência máxima nesta esfera 4D. Segundo De Biasi e Gattass (2007), realizar esta interpolação é simples, primeiramente encontra-se a distância angular entre os dois quatérnios na superfície da hipersfera, como ambos são unitários, o produto interno destes fornece o cosseno do ângulo entre eles:

$$q_1 \cdot q_2 = \cos\Omega \Rightarrow \Omega = \arccos(q_1 \cdot q_2) \quad (38)$$

O quatérnio q_3 a uma posição angular intermediária α entre q_1 e q_2 , isto é, entre 0 e Ω em relação a q_1 , é dado por (idem, 2007):

$$q_3 = q_1 \frac{\text{sen}(\Omega - \alpha)}{\text{sen}(\Omega)} + q_2 \frac{\text{sen}(\alpha)}{\text{sen}(\Omega)} \quad (39)$$

Substituindo α por $t\Omega$, onde $t \in [0,1]$ obtém-se:

$$q_3 = q_1 \frac{\text{sen}((1-t)\Omega)}{\text{sen}(\Omega)} + q_2 \frac{\text{sen}(t\Omega)}{\text{sen}(\Omega)} \quad (40)$$

A notação acima (eq. 40) para a SLERP entre dois quatérnios é útil na implementação, mas a notação exponencial é mais concisa e será usada na demonstração do método, por isso, a apresentamos a seguir. Considere um quatérnio q em notação polar, conforme a eq. (27): $q = \cos(\theta/2) + \vec{n} \text{sen}(\theta/2)$, fazendo $\theta/2 = \beta$ e expandindo em série de Taylor as funções sen e cos tem-se:

$$\vec{n} \text{sen}(\beta) = \vec{n} \left(\beta - \frac{\beta^3}{3!} + \frac{\beta^5}{5!} - \frac{\beta^7}{7!} + \dots \right) \quad (41)$$

$$\cos(\beta) = 1 - \frac{\beta^2}{2!} + \frac{\beta^4}{4!} - \frac{\beta^6}{6!} + \dots \quad (42)$$

Aplicando a identidade de Euler, a notação polar passa a ser representada como uma exponencial complexa, a saber:

$$e^{\vec{n}\beta} = \cos\beta + \vec{n} \text{sen}\beta \quad (43)$$

portanto:

$$q = \cos\beta + \vec{n} \text{sen}\beta \rightarrow q = e^{\beta\vec{n}} \quad (44)$$

logo:

$$\ln(q) = \beta\vec{n} \quad (45)$$

elevando o quatérnio q a um número real t :

$$q^t = e^{t \ln(q)} \rightarrow e^{t\beta\vec{n}} = \cos(t\beta) + \vec{n} \operatorname{sen}(t\beta) \quad (46)$$

A equação (44) é a notação exponencial para quatérnios, ela representa uma parametrização de $SO(3)$ dada por $c: \mathbb{R}^3 \rightarrow S^3$, onde $c(t) = q^t$. Se $t \in [0,1]$ então $c(t)$ é linear e define um arco de circunferência máxima na esfera de quatérnios (SCHULZ, 2010). Portanto, $c(t)$ define uma geodésica em S^3 , fato pelo qual o termo SLERP é também conhecido como interpolação geodésica. Desta forma, tem-se como alternativa à eq. (40) a fórmula (DAM et al. 1998):

$$q_3 = \operatorname{slerp}(q_1, q_2, t) = q_1(q_1^{-1}q_2)^t \quad (47)$$

onde q_1 e q_2 são os quatérnios que entram na função e q_3 é o quatérnio interpolado a uma distância $t \in [0,1]$ entre q_1 e q_2 . Por ser linear, a fórmula também é simétrica (DAM et al. 1998):

$$\operatorname{slerp}(q_1, q_2, t) = \operatorname{slerp}(q_2, q_1, 1 - t)$$

3 METODOLOGIA

Este capítulo apresenta os materiais, as etapas de pré-processamento dos dados, os parâmetros que serão utilizados e o desenvolvimento em detalhes do modelo proposto para refinamento global de nuvens de pontos 3D.

3.1 Materiais

O material mais importante deste trabalho são os cinco *datasets* de nuvens de pontos 3D, todos originários de LST. Mais detalhes sobre os *datasets* estão no capítulo 4. Em segundo plano tem-se o *hardware* e os *softwares* utilizados para a manipulação das nuvens de pontos, em sua maior parte disponíveis no computador do Laboratório de Pesquisa em Dados Espaciais (LAPE) do setor de Geomática na Universidade Federal do Paraná (UFPR).

3.1.1 Hardware

O desktop do LAPE é montado com a seguinte configuração:

- Processador - Intel i7-9700F a 3.0 GHz com 8 núcleos e a 4.5 GHz com um núcleo;
- Memória RAM - 32 GB distribuídos em dois pentes DDR4 a 2400 MHz;
- Memória fixa - 3 TB distribuídos em duas unidades SSD de 480 GB e 1 HDD de 2 TB;
- Processador gráfico - AMD R7-200 com 2GB de V-RAM GDDR6 a 5 GHz.

3.1.2 Software

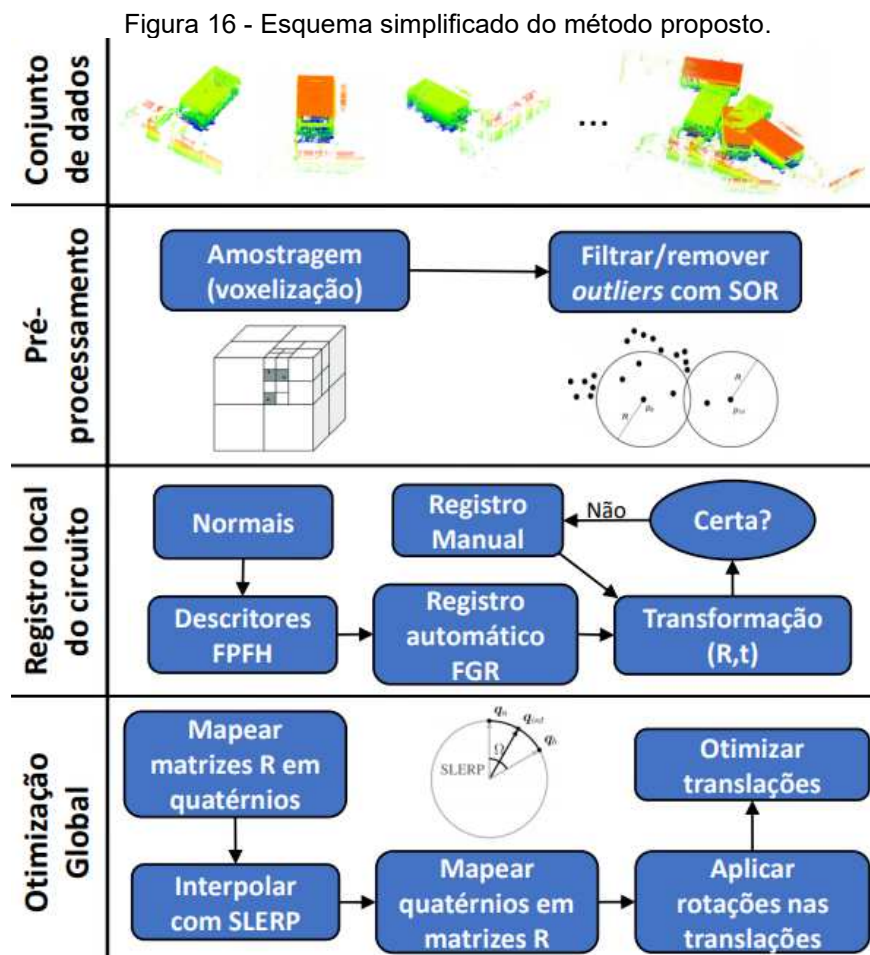
O desktop possui os seguintes programas e linguagens de programação acessíveis:

- Sistemas operacionais:
 - *Linux Ubuntu 18.04 LTS*;
 - *Windows 10*.
- *Softwares*:
 - *MATLAB 2020*;

- *Cloud Compare 2.6.1*;
- *Visual Studio Code* (editor de código);
- Linguagens de programação:
 - *Python 3*;
 - *Matlab*.

3.2 Método

A Figura 16 ilustra o esquema simplificado do método proposto.



fim, na quarta etapa, tem-se a otimização global das poses, que são calculadas através dos circuitos da etapa anterior.

3.2.1 Aquisição de dados

Devido à pandemia causada pelo covid-19, não foi possível o trabalho em campo com o LST disponível no laboratório da UFPR, o BLK-360. Mas isto em nada impediu a aquisição de *datasets*. Quatro conjuntos de nuvens foram obtidos de forma online e um disponibilizado pelo professor Daniel Santos, este último, infelizmente, sem a verdade de campo (*groundtruth*) das poses. No total tem-se 41 nuvens com mais de 1 milhão de pontos cada, mais detalhes sobre os *datasets* estão na introdução do Capítulo 4 (Avaliação Experimental).

3.2.2 Pré-processamento

O pré-processamento consiste na amostragem e filtragem das nuvens para reduzir o tempo de execução e os mínimos locais da função de registro. A literatura é vasta nesta parte, em alguns casos filtragem e amostragem se confundem, de forma que a composição e combinação das duas resultam em dezenas de métodos. Uma revisão sobre filtros e métodos de amostragem para nuvens de pontos pode ser lida em (HAN et al., 2017).

Uma forma eficiente de amostragem adotada para dados LST, utilizada por Theiler et al., (2015), consiste em subdividir o espaço da nuvem de pontos em octantes, sucessivamente, até certo tamanho mínimo. Os octantes formam uma estrutura chamada *octree*, que define cubos tão pequenos quanto o desejado. Estes cubos são conhecidos como *voxels* (*volume-elements*). *Voxels*, assim como pixels, podem conter diversos atributos, nos interessa o centroide dos pontos no seu interior. Uma nuvem de centroides terá densidade aproximadamente constante, o que é bastante desejável, pois densidades diferentes ao longo da nuvem enviesam a seleção de correspondências e inviabilizam métricas de avaliação do registro local. Utilizaremos *voxels* de 0,2 metros para amostragem de todas as nuvens

Após a amostragem tem-se a filtragem com o filtro *Statistical Outlier Removal* (SOR) de (RUSU et al., 2008). Este filtro opera por dois parâmetros: uma quantidade mínima de pontos k , e um desvio padrão σ , calculado no espaço das médias dos k

vizinhos mais próximos (*k-nearest neighbor* - knn) de cada ponto. Os knn de cada ponto são encontrados pelo algoritmo FLANN (*Fast Library for Approximate Nearest Neighbour*) de (MUJA e LOWE, 2009). Um ponto é considerado *outlier* se estiver fora do intervalo $[-\alpha\sigma, \alpha\sigma]$, sendo α um escalar. Adota-se $\alpha = 1.0$ como um valor padrão.

A escolha deste filtro e método de amostragem se devem ao uso comum e bem estabelecido na literatura, uma prova disso é que ambos estão implementados em diversas bibliotecas para a manipulação de nuvens de pontos 3D. Alguns exemplos: *PyntCloud*, biblioteca em python (CASTRO, 2020) *Point Cloud Library* biblioteca em C++ (RUSU e COUSINS, 2011), *Computer Vision Toolbox* (MATLAB, 2020) e *Open3D*, biblioteca em python (ZHOU et al. 2018).

3.2.3 Registro local

Como solução para o registro local emprega-se o algoritmo *Fast Global Registration* (FGR) de (ZHOU et al., 2016). O FGR utiliza o descritor FPFH apresentado na seção 2.3.2, que necessita das normais dos pontos da nuvem, definiremos como knn das normais a quantidade de pontos utilizada no cálculo de cada normal. Além disso, é preciso definir uma vizinhança para o descritor operar em cada ponto, chamaremos essa vizinhança de knn do FPFH. Após uma série de testes, adotou-se 10 vizinhos para o cálculo das normais e 50 vizinhos para o cálculo do descritor. Estes valores foram definidos empiricamente, mas a literatura também sugere valores próximos a este (ZHOU, et al. 2018).

Embora os parâmetros do algoritmo FGR venham previamente definidos em (ZHOU, et al. 2016), a sua configuração se deu com base na observação dos testes de registro entre nuvens dos diferentes *datasets*. Diante disso, tomamos a liberdade de aumentar a quantidade de iterações do RANSAC para 3000 (o padrão é 640). O conjunto consenso de correspondências também foi elevado para 300 (o padrão é 64). Isto foi necessário para aumentar a robustez do algoritmo no registro das nuvens de LST, dado que os parâmetros do FGR foram originalmente definidos com testes em nuvens menores.

Na seleção de correspondências entre os pares de nuvens, uma distância máxima de correspondência também precisa ser definida. Seguindo as recomendações da biblioteca *Open3D*, na qual o algoritmo está documentado, adotou-se 2 vezes o valor do *voxel* utilizado na amostragem: 0,2 metros. Por fim,

mudou-se o parâmetro que controla a convexidade da função de rejeição de outliers para 0,8 (o padrão é 1.4). Mais detalhes sobre esses parâmetros são dados no artigo original do algoritmo (ZHOU et al., 2016).

Caso não seja possível a reconstrução automática de algum *dataset* com o FGR, todo o circuito será registrado manualmente. Isto significa que se o FGR falhar em pelo menos um dos pares do circuito, todas os pares daquele *dataset* serão registrados manualmente. Isto é necessário para garantir que os erros ao longo do circuito sejam os mais uniformes possíveis, pois o erro de um registro bem-sucedido pelo FGR é diferente do erro de um registro manual. A Tabela 3 reúne os parâmetros utilizados no pré-processamento e no registro automáticos das nuvens de pontos.

Tabela 3 - Parâmetros utilizados no pré-processamento das nuvens e no FGR.

Pré-processamento (amostragem, filtragem, cálculo de normais e descritor FPFH)	
Tamanho do voxel	= 0.1 m
Desvio do filtro (α)	= 1.0 d.p.
knn do filtro	= 10 pts.
knn das normais	= 10 pts.
knn do FPFH	= 50 pts.
Registro local por FGR (ZHOU et al. 2016)	
Iterações RANSAC (critério de parada)	= 3000
Conjunto consenso FPFH	= 300
Distância máx. de correspondência.	= 0,2 m
Parâmetro de controle da convexidade da função	= 0,8

Fonte: o autor (2020).

3.2.4 Método de refinamento global proposto

Para aplicar o método global proposto neste trabalho é necessário um circuito fechado de transformações na rede de estações do LST. O erro presente em cada transformação deve ser aceitável, pois se trata de um refinamento. Erros grosseiros nos registros locais ao longo do circuito corromperão os resultados. Se a ordem das varreduras for conhecida, a estratégia automática de registro com o FGR será aplicada nesta ordem, seguindo a numeração das nuvens dentro do *dataset*. Assume-se que há sobreposição entre as nuvens na ordem de coleta.

Outra ordem de registro local poderia ser usada, por exemplo, um caminho com maior sobreposição entre pares que garantisse menor erro total de fechamento

(*loopclosure*), todavia, a investigação de ciclos em grafos é um problema complexo que foge dos objetivos deste trabalho.

Estabelecido um circuito entre as nuvens, tem-se a aplicação do método para otimização global dos parâmetros de todas as transformações. Na primeira etapa as matrizes de rotação são mapeadas para quatérnios, com estes é feita a Interpolação Linear Esférica (SLERP), que é a proposta original deste trabalho. Em seguida os quatérnios são mapeados de volta para matrizes de rotação e estas são aplicadas nas translações do circuito. Na segunda etapa do método os parâmetros de translação são ajustados pelo modelo de Lu e Milios (1997).

3.2.4.1 Distribuição global do erro na rotação por SLERP

Uma vez estabelecida a nuvem de referência global para o conjunto de nuvens, são estimadas as transformações sucessivas do circuito pelo registro local. Cada matriz de rotação $R_{(n-1,n)}$ que conecte nuvens X_{n-1} e X_n será composta com a seguinte, até se obter o conjunto de rotações que coloca cada nuvem na origem global. Isto é feito pela sequência de multiplicações a seguir:

$$\begin{aligned} R_1 &= M_1 \\ R_2 &= M_2 R_1 \\ &\vdots \\ R_n &= M_n R_{n-1} \\ R_{n+1} &= M_{n+1} R_n \end{aligned} \quad (48)$$

sendo R_i a matriz de rotação do vértice X_i referenciada ao vértice X_0 e M_i a matriz de rotação do vértice X_i referenciado ao vértice X_{i-1} , com $i = 1, 2, \dots, n$. Se as estimativas das rotações não apresentassem erros a última matriz de rotação seria igual à identidade: $R_{n+1} = I$. No entanto, erros aleatórios se acumulam e a matriz se distancia da identidade quanto maior for o erro presente no circuito. A multiplicação das matrizes de rotação pode ser escrita como composições de quatérnios. Deste modo, as eq. 48 são rescritas como segue:

$$\begin{aligned} r_1 &= q_1 \\ r_2 &= q_2 r_1 \\ &\vdots \\ r_n &= q_n r_{n-1} \\ r_{n+1} &= q_{n+1} r_n \end{aligned} \quad (49)$$

sendo r_i o quatérnio que representa a rotação do vértice X_i referenciado ao vértice X_0 e q_i o quatérnio representa a rotação do vértice X_i referenciado ao vértice X_{i-1} com

$i = 1, 2, \dots, n$. O quatérnio r_{n+1} será o da rotação *loopclosure*, aquele que acumula o erro das rotações e aproxima-se da identidade. Pela propriedade de multiplicação dos quatérnios define-se um novo quatérnio p_i :

$$\begin{aligned} p_1 &= r_1 r_{n+1}^{-1} \\ p_2 &= r_2 r_{n+1}^{-1} \\ &\vdots \\ p_n &= r_n r_{n+1}^{-1} \end{aligned} \quad (50)$$

p_i também representa uma rotação para a origem, mas acumula rotações pelo caminho inverso do circuito. Portanto, cada vértice do circuito, contém duas rotações que levam à mesma orientação em relação à origem: r_i e p_i . Para obter uma rotação ótima com base na superabundância destas duas rotações disponíveis, sugere-se que os quatérnios r_i e p_i sejam interpolados com SLERP, como segue:

$$\begin{aligned} \hat{q}_1 &= r_1 (r_1^{-1} p_1)^{\tau_1} \\ \hat{q}_2 &= r_2 (r_2^{-1} p_2)^{\tau_2} \\ &\vdots \\ \hat{q}_n &= r_n (r_n^{-1} p_n)^{\tau_n} \end{aligned} \quad (51)$$

realizando a substituição das eq. 50 em 51, tem-se:

$$\begin{aligned} \hat{q}_1 &= r_1 (r_1^{-1} r_1 r_{n+1}^{-1})^{\tau_1} = r_1 r_{n+1}^{-\tau_1} \\ \hat{q}_2 &= r_2 (r_2^{-1} r_2 r_{n+1}^{-1})^{\tau_2} = r_2 r_{n+1}^{-\tau_2} \\ &\vdots \\ \hat{q}_n &= r_n (r_n^{-1} r_n r_{n+1}^{-1})^{\tau_n} = r_n r_{n+1}^{-\tau_n} \end{aligned} \quad (52)$$

sendo \hat{q}_i o quatérnio que representa a rotação ajustada do vértice X_i referenciadas ao vértice X_0 , com $i = 1, 2, \dots, n$. O ajustamento é aplicado sobre os parâmetros τ_i , que são os expoentes de cada interpolação. Com à restrição dada pela eq. 53, cada equação se aproxima o máximo possível do quatérnio identidade, como segue:

$$\begin{aligned} \hat{q}_1^{-1} q_1 &= (r_1 r_{n+1}^{-\tau_1})^{-1} q_1 \approx 1 \\ \hat{q}_2^{-1} q_2 \hat{q}_1 &= (r_2 r_{n+1}^{-\tau_2})^{-1} q_2 r_1 r_{n+1}^{-\tau_1} \approx 1 \\ &\vdots \\ \hat{q}_n^{-1} q_n \hat{q}_{n-1} &= (r_n r_{n+1}^{-\tau_n})^{-1} q_n r_{n-1} r_{n+1}^{-\tau_{n-1}} \approx 1 \\ q_{n+1} \hat{q}_n &= q_{n+1} r_n r_{n+1}^{-\tau_n} \approx 1 \end{aligned} \quad (53)$$

realizando as devidas inversões de quatérnios e a substituição da eq. 49 na eq. 53, as equações acima se reescrevem como:

$$\begin{aligned}
\hat{q}_1^{-1} q_1 &= r_{n+1}^{\tau_1} r_1^{-1} r_1 = r_{n+1}^{\tau_1} \approx 1 \\
\hat{q}_2^{-1} q_2 \hat{q}_1 &= r_{n+1}^{\tau_2} r_2^{-1} r_2 r_{n+1}^{-\tau_1} = r_{n+1}^{\tau_2} r_{n+1}^{-\tau_1} \approx 1 \\
&\vdots \\
\hat{q}_n^{-1} q_n \hat{q}_{n-1} &= r_{n+1}^{\tau_n} r_n^{-1} r_n r_{n+1}^{-\tau_{n-1}} = r_{n+1}^{\tau_n} r_{n+1}^{-\tau_{n-1}} \approx 1 \\
q_{n+1} \hat{q}_n &= r_{n+1} r_{n+1}^{-\tau_n} \approx 1
\end{aligned} \tag{54}$$

utilizando a propriedade da multiplicação de potência de quatérnios: $q^{\tau_1} q^{\tau_2} = q^{\tau_1 + \tau_2}$, válida para a multiplicação de potências de dois quatérnios iguais, segue-se que:

$$\begin{aligned}
\hat{q}_1^{-1} q_1 &= r_{n+1}^{\tau_1} \approx 1 \\
\hat{q}_2^{-1} q_2 \hat{q}_1 &= r_{n+1}^{\tau_2} r_{n+1}^{-\tau_1} = r_{n+1}^{\tau_2 - \tau_1} \approx 1 \\
&\vdots \\
\hat{q}_n^{-1} q_n \hat{q}_{n-1} &= r_{n+1}^{\tau_n} r_{n+1}^{-\tau_{n-1}} = r_{n+1}^{\tau_n - \tau_{n-1}} \approx 1 \\
q_{n+1} \hat{q}_n &= r_{n+1} r_{n+1}^{-\tau_n} = r_{n+1}^{1 - \tau_n} \approx 1
\end{aligned} \tag{55}$$

Assim como os números reais, todo quatérnio elevado a 0 é igual a 1. Com isto é conjecturado que, para obter quatérnios ótimos, é necessário minimizar cada potência das eq. 55, ou seja, todos os expoentes devem se aproximar de 0. Isto pode ser feito minimizando o seguinte sistema de equações lineares:

$$\begin{aligned}
\tau_1 &= \theta_1 \\
\tau_2 - \tau_1 &= \theta_2 \\
&\vdots \\
\tau_n - \tau_{n-1} &= \theta_{n-1} \\
1 - \tau_n &= \theta_n
\end{aligned} \tag{56}$$

Para esta tarefa é aplicado o modelo paramétrico linear do MMQ, isto é, $\hat{t} = (J^T J)^{-1} J^T b$. Onde J é a matriz jacobiana de derivadas parciais, b é o vetor das observações. As variáveis do sistema são os expoentes da interpolação SLERP $(\tau_1, \tau_2, \dots, \tau_n)$, a quantidade de equações é função dos n vértices/estações do circuito, e seguem o padrão em 56. Percebe-se que n vértices/estações, em um circuito fechado, determinam $n - 1$ equações em 56. A solução que minimiza esse sistema é dada matricialmente por:

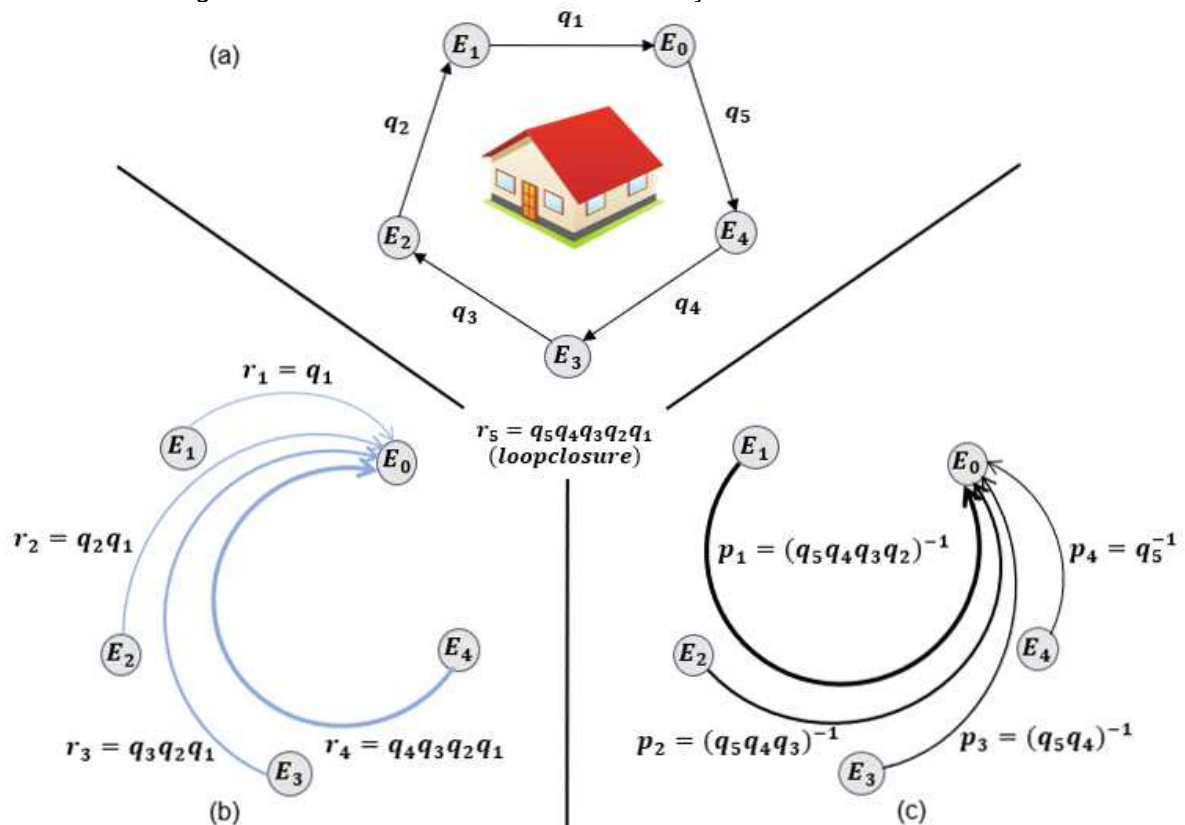
$$\begin{bmatrix} \tau_1 \\ \tau_2 \\ \vdots \\ \tau_{n-1} \\ \tau_n \end{bmatrix} = \left(\begin{bmatrix} 2 & -1 & \dots & 0 & 0 \\ -1 & 2 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & 2 & -1 \\ 0 & 0 & \dots & -1 & 2 \end{bmatrix} \right)^{-1} \left(\begin{bmatrix} 1 & 0 & \dots & 0 & 0 \\ -1 & 1 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & -1 & 1 \\ 0 & 0 & \dots & 0 & -1 \end{bmatrix} \right)^T \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} \tag{57}$$

Este modelo, uma vez solucionado, demonstra quais são os intervalos (expoentes) de interpolação ótimos entre os quatérnios. Como o resultado do sistema é função apenas da quantidade de estações, dada uma quantidade de estações, o resultado é

previamente conhecido para um circuito fechado, pois a matriz J segue um padrão simples, e o vetor b , igualmente. Portanto, a solução também segue um padrão simples. Vejamos um exemplo, no caso de um circuito com 5 estações, o vetor solução será $[\tau_1 \ \tau_2 \ \tau_3 \ \tau_4] = \left[\frac{1}{5} \ \frac{2}{5} \ \frac{3}{5} \ \frac{4}{5} \right]$. Ou seja, a solução segue $\frac{1}{n}, \dots, \frac{n-2}{n}, \frac{(n-1)}{n}$.

Daremos uma intuição geométrica para este resultado com um exemplo de otimização das rotações em um circuito fechado de 5 estações. Neste circuito tem-se cinco rotações entre pares de nuvens, são elas q_1, q_2, q_3, q_4 e q_5 (Figura 17a). Para obter-se a rotação de cada nuvem para a origem global, E_0 , faz-se as composições $r_1 = q_1$, $r_2 = q_2 q_1$, $r_3 = q_3 q_2 q_1$ e $r_4 = q_4 q_3 q_2 q_1$, conforme a Figura 17 (b). O quatérnio *loopclosure* será dado por $r_5 = q_5 q_4 q_3 q_2 q_1$, e deve se aproximar do quatérnio identidade, isto é: $r_5 = 1 + 0i + 0j + 0k$.

Figura 17 - Circuito fechado com $n = 5$ estações ao redor de uma casa.



(a) representação das rotações entre pares de estações. (b) rotações compostas para a origem E_0 no sentido horário. (c) rotações compostas para a origem E_0 no sentido anti-horário.

Fonte: autor (2020).

As espessuras das linhas na Figura 17 (b) e (c) são proporcionais à quantidade de rotações acumuladas nos quatérnios r_i e p_i . Perceba que cada uma das rotações compostas r_i pode ser obtida no sentido contrário, como mostra a Fig. 17 (c). Por

exemplo, a rotação r_4 , que orienta a nuvem em E_4 para E_0 , pode ser obtida simplesmente invertendo-se o quatérnio q_5 , na Figura 17 (c), este é o quatérnio p_4 . A rotação r_3 , que orienta a nuvem em E_3 para E_0 , pode ser obtida invertendo-se o produto q_5q_4 , este é o quatérnio p_3 . A rotação p_2 será o inverso de $q_5q_4q_3$. Por último, a rotação p_1 , que equivale a obter r_1 indo pelo outro lado do circuito, pode ser obtida com o inverso de $q_5q_4q_3q_2$. Estes quatérnios são exatamente os p_i da eq. (50), mas aqui eles estão sendo calculados por:

$$\begin{aligned} p_4 &= (q_5)^{-1} \\ p_3 &= (q_5q_4)^{-1} \\ p_2 &= (q_5q_4q_3)^{-1} \\ p_1 &= (q_5q_4q_3q_2)^{-1} \end{aligned} \tag{58}$$

Não é fácil perceber isto na demonstração, pois na eq. 50 obtém-se p_i com a multiplicação de r_i por r_{n+1}^{-1} . Neste exemplo, r_{n+1} é o quatérnio r_5 , chamado de *loopclosure* por acumular os erros nas rotações no circuito. A eq. 50 nos diz que podemos obter exatamente os mesmos p_i acima por:

$$\begin{aligned} p_1 &= r_1r_5^{-1} \\ p_2 &= r_2r_5^{-1} \\ p_3 &= r_3r_5^{-1} \\ p_4 &= r_4r_5^{-1} \end{aligned} \tag{59}$$

e isto não é nada óbvio.

Sabendo que p_i acumula rotações no sentido contrário do circuito, o vetor solução dos intervalos de interpolação passa a fazer sentido, pois as frações $[\tau_1 \ \tau_2 \ \tau_3 \ \tau_4] = [1/5 \ 2/5 \ 3/5 \ 4/5]$ estão distribuindo o erro acumulado em cada rotação segundo a quantidade de estações no circuito. A interpolação do primeiro par de rotações, de E_1 para E_0 , ocorre em $1/5$, $\hat{q}_1 = \text{slerp}(r_1, p_1, 1/5)$, porque r_1 é simplesmente q_1 , mas p_1 vem pelo outro lado do circuito e acumula o produto de quatro rotações $p_1 = (q_5q_4q_3q_2)^{-1}$. Como p_1 contém mais erro, a interpolação ocorre em $1/5$ da distância entre os dois, dando mais peso para r_1 . A largura das linhas na Figura 17 (b) e (c) busca representar essa assimetria.

Em seguida tem-se a interpolação do par de rotações de E_2 para E_0 , que ocorre em $2/5$: $\hat{q}_2 = \text{slerp}(r_2, p_2, 2/5)$. Isto é razoável, pois $r_2 = q_2q_1$ e $p_2 = (q_5q_4q_3)^{-1}$. Novamente, a interpolação é mais próxima de r_2 do que de p_2 , pois este segundo contém mais erro, dado que é calculado como o produto de 3 rotações. Caso o nº de estações do circuito fosse par, uma das rotações seria interpolada exatamente em

$n/2n$, isto é, em $1/2$, pois a quantidade de rotações acumuladas por um lado seria igual a quantidade pelo outro. Finalizando o nosso exemplo, o último par de rotações interpoladas é calculado como: $\hat{q}_4 = \text{slerp}(r_4, p_4, 4/5)$. Desta vez a interpolação ocorre mais próxima de p_4 pois ele é dado apenas por $(q_5)^{-1}$, enquanto r_4 é o produto de quatro rotações $r_4 = q_4 q_3 q_2 q_1$.

Em suma, o intervalo de interpolação $t \in [0,1]$ cresce conforme o acúmulo de rotações, o que era esperado de um modelo que distribui o erro acumulado entre as poses, que são transformações entre pares compostas. Portanto, t atua como um ponderador que dá mais importância à rotação com menos erros. Isto assume, implicitamente, duas premissas, (1) que quanto mais rotações forem acumuladas, mais erro elas possuem; (2) que o erro nas rotações é distribuído uniformemente e cresce linearmente.

3.2.4.2 Distribuição global do erro na translação por Lu e Milios (1997)

Em otimização combinatória a abordagem por grafos é a essência do método. O trabalho de Lu e Milios (1997) aborda as transformações entre nuvens desta forma. O algoritmo ficou conhecido na literatura como LUM. Os autores formulam o problema considerando um robô que, a cada deslocamento, captura o perfil horizontal 2D do ambiente. Assim, para cada varredura existe uma nuvem de pontos e uma única estação. Calcula-se a pose do robô, em determinada estação, pelo registro da nuvem de pontos seguinte na nuvem de pontos da estação anterior, isto é, pelo registro de E_n em $E_{(n-1)}$, método chamado de odometria por laser.

Para ilustrar a inconsistência de poses obtidas por múltiplos caminhos, considere uma estação inicial E_0 , como na Figura 18. Seja r uma função de registro, onde $T_{ab} = r(E_a, E_b)$ é a transformação que registra E_a em E_b . Conforme a figura 18, tem-se que a pose de E_3 é simplesmente o registro direto de E_3 em E_0 , ou seja: $Pose(E_3) = T_{30} = r(E_3, E_0)$. Mas, como há múltiplas sobreposições, pode-se encontrar esta pose por diferentes composições de transformações relativas, por exemplo:

$$T_{30}' = f(T_{32}, T_{24}, T_{40}) \quad (60)$$

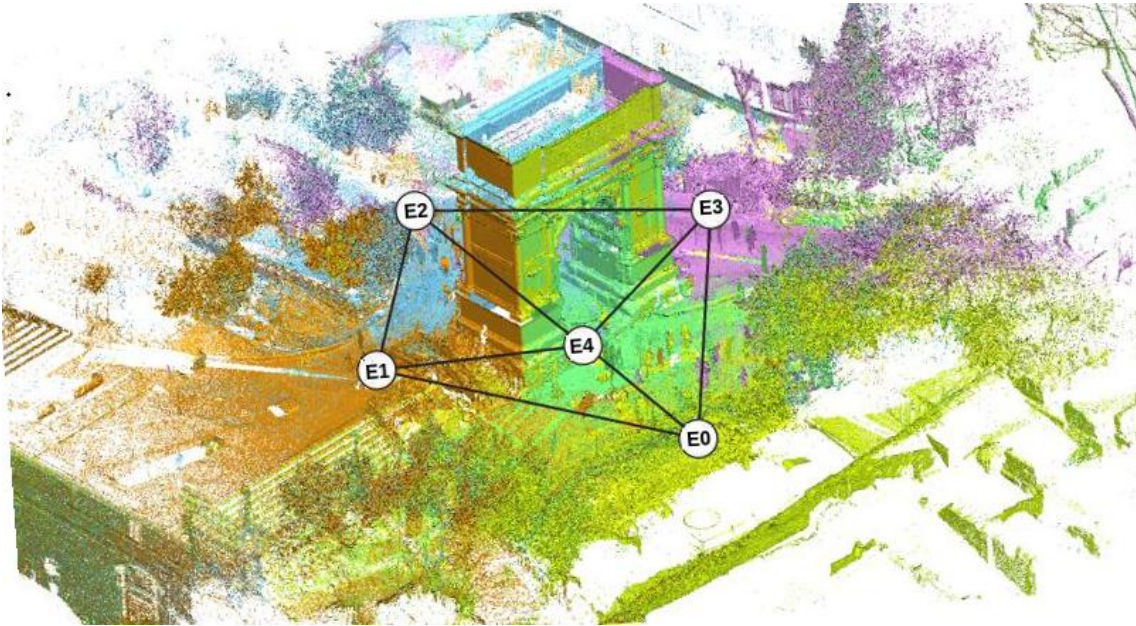
ou ainda:

$$T_{30}'' = f(T_{34}, T_{40}) \quad (61)$$

onde f é a função que compõe várias transformações. Sua operação é acumular sucessivas transformações entre sistemas. Como o efeito de aplicar rotações e translações sobre um objeto não é comutativo, adota-se aqui que a matriz de rotação opera no vetor multiplicando-o pela esquerda, como segue:

$$T_{ab} = f(T_a, T_b) = \begin{bmatrix} R_a & t_a \\ 0 & 1 \end{bmatrix} \begin{bmatrix} R_b & t_b \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} R_a R_b & R_a t_b + t_a \\ 0 & 1 \end{bmatrix} \quad (62)$$

Figura 18 - Grafo representativo de 5 estações ocupadas por um sensor LST.



Fonte: Theiler et al. (2015).

Agora, suponha que um algoritmo de registro forneça as transformações da eq. 61, com aqueles subíndices a eq. 62 torna-se:

$$T_{30} = f(T_{34}, T_{40}) = \begin{bmatrix} R_{34} & t_{34} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} R_{40} & t_{40} \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} R_{34}R_{40} & R_{34}t_{40} + t_{34} \\ 0 & 1 \end{bmatrix} \quad (63)$$

Isto posto, fica claro que as diferentes estimativas de T_{30} , se obtidas por diferentes caminhos, serão discordantes. Mesmo que algum tipo de média fosse utilizada para estimar a pose de E_3 : $Pose(E_3) = média(T_{30}, T'_{30}, T''_{30})$, outro problema surgiria, pois as transformações T_{32} e T_{34} se tornariam inconsistentes. Então, em uma rede com múltiplas nuvens, alterações na pose de uma devem ser propagadas para as outras, portanto, nenhuma transformação encontrada utilizando apenas um par é solução ótima de um sistema com mais de duas nuvens que se sobrepõem mutuamente.

Para solucionar o problema da rede conflitante de transformações, onde vários circuitos podem ser percorridos e diferentes erros obtidos, Lu e Milios (1997) definem uma nuvem fixa e minimizam a variância das transformações entre elas como uma

distância de Mahalanobis. Dado dois conjuntos de valores x_i e $x_j \in \mathbb{R}^3$ cada um com sua respectiva MVC, C_i e C_j , seja C_{ij} a covariância conjunta e $f: \mathbb{R}^3 \rightarrow \mathbb{R}^3$ um modelo que mapeia x_i em x_j , quer-se encontrar os parâmetros do modelo f que minimiza a distância entre os dois conjuntos. Matematicamente, isto equivale a encontrar o mínimo da seguinte função:

$$W = \sum((x_i - x_j) - (\bar{d}_{ij}))C_{ij}^{-1}((x_i - x_j) - (\bar{d}_{ij})) \quad (64)$$

Onde \bar{d}_{ij} é assumido como uma observação aleatória com distribuição normal de média μ e covariância Σ :

$$\bar{d}_{ij} \sim N(\mu, \Sigma) \quad (65)$$

Em outras palavras, a equação 64 é o quadrado dos resíduos do modelo, cada um ponderado pela sua respectiva covariância, minimizá-la é aplicar o MMQ ponderado. Para usar todos os caminhos simultaneamente, Lu e Milios (1997) determinam a matriz adjacência H do dígrafo (grafo direcionado) e organizam as equações de observação da seguinte forma:

$$D = HX \quad (66)$$

onde X é o vetor das translações das poses, concatenadas como $X = (X_0, X_1, \dots, X_n)$. A origem do sistema é estabelecida definindo-se a primeira estação como $X_0 = (0,0,0)$. Reescrevendo a eq. 64 em forma matricial o somatório é removido, a saber:

$$W = (HX - \bar{D})^T C^{-1} (HX - \bar{D}) \quad (67)$$

agora \bar{D} é a concatenação de todas as observações \bar{d}_{ij} para cada equação correspondente D_{ij} . E C^{-1} é a matriz quadrada formada pelas submatrizes C_{ij} concatenadas diagonalmente. A solução mínima dessa função para X é:

$$X_{min} = (H^T C^{-1} H)^{-1} H^T C^{-1} \bar{D} \quad (68)$$

e a MVC de X_{min} será:

$$C_{Xmin} = (H^T C^{-1} H)^{-1} \quad (69)$$

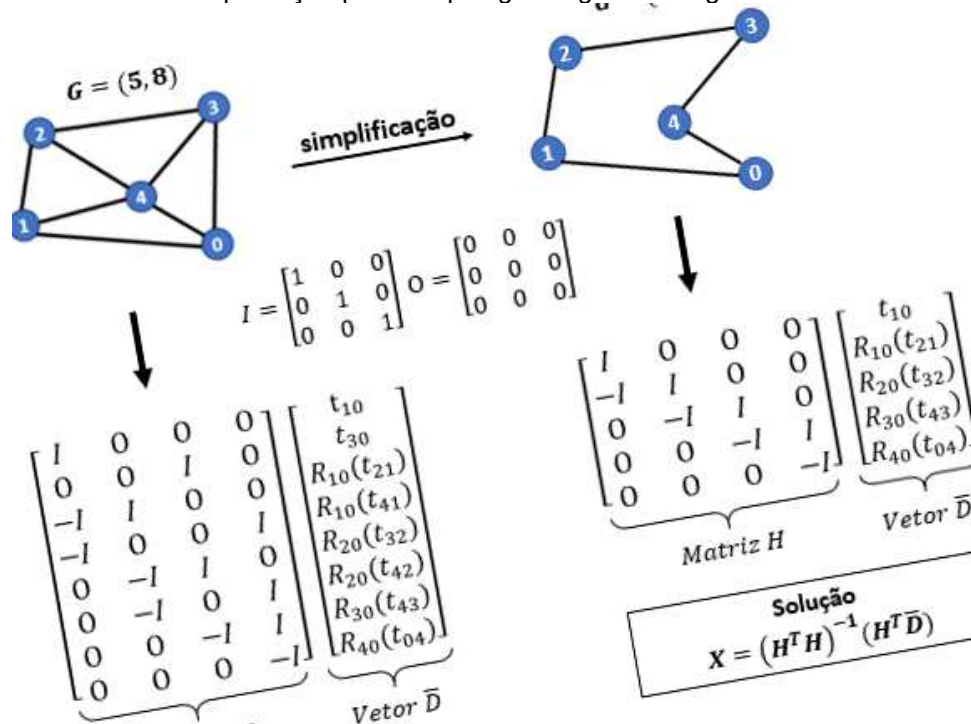
Lu e Milios (1997) aplicaram estas equações em um robô com 3 graus de liberdade, isto é, cada estação determina um vetor de observação com x , y e uma orientação θ . O mesmo modelo foi adaptado para três dimensões por Borrmann et al. (2008) e tornou-se comum na literatura de SLAM (STACHNISS et al., 2007). No caso 3D, o sistema determina x , y , z em conjunto com os ângulos de Euler ω , φ e κ , por isto é chamado de SLAM6D ou SLAM Completo.

O algoritmo foi aplicado com sucesso em diferentes tipos de dados, como os do *Kinect* da *Microsoft* (WEBER et al. 2015) e em dados de LST (THEILER et al., 2015), (PAVAN e DOS SANTOS, 2017). Neste trabalho, o modelo usará o MMQ não ponderado, pois o vetor translação será obtido iterativamente pelo ICP, que não retorna exatidões. Todavia, isto não impede que outras formas de ponderação sejam exploradas.

Para total esclarecimento, demonstra-se um exemplo completo da aplicação deste modelo. Primeiramente, o grafo de estações da Figura 18 será simplificado. A topologia adotada será a mesma do exemplo de otimização das rotações (Figura 17). Na Figura 18 o grafo $g = (5,8)$ tem múltiplos circuitos, o que se deseja é apenas um circuito, isto é, um grafo em forma de anel com 5 vértices e 5 arestas: $g = (5,5)$.

Assume-se que todas as transformações entre pares de estações são conhecidas, a simplificação adotada está na Figura 19. As matrizes jacobianas, de derivadas, são as matrizes H , e estão organizadas de acordo com o grafo. O modelo é solucionado sem pesos/variâncias para as observações \bar{D} , que são as translações de cada par rotacionadas para a origem global.

Figura 19 - Construção da matriz jacobiana utilizada no modelo LUM para o grafo da Figura 18 e sua simplificação para a topologia do grafo na Figura 17.



Fonte: o autor (2021).

Na Figura 19, cada aresta do grafo estabelece uma equação para determinada pose. A matriz H é organizada segundo as derivadas parciais $\frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z}$ das equações das poses, elas não serão mostradas aqui, pois as equações são longas e podem ser obtidas por composição (eq. 63).

Cada observação em \bar{D} é um vetor translação entre um par de nuvens rotacionado para a origem global. Desta forma, a solução em X são as translações das poses otimizadas/ajustadas. Como a otimização das rotações ocorre na forma de quatérnios, estes são mapeados de volta para matrizes de rotação antes de entrarem no modelo, isto é: $\hat{q}_1, \hat{q}_2, \hat{q}_3, \hat{q}_4 \rightarrow R_{10}, R_{20}, R_{30}, R_{40}$. Portanto, otimizam-se as rotações primeiro, depois ocorre a otimização das translações. Como dito anteriormente, ambos os modelos são independentes, pode-se aplicar um ou outro.

Encerra-se aqui a demonstração do modelo de otimização global proposto. A seguir, no capítulo 4, este modelo será testado em diversos *datasets*, com diferentes geometrias e quantidade de estações, mas sempre com a mesma topologia de um circuito fechado (um anel).

4 AVALIAÇÃO EXPERIMENTAL

Neste capítulo avalia-se o resultado do registro local e das etapas de otimização global nos *datasets* de nuvens de pontos 3D. Primeiramente, os *datasets* são apresentados com suas características principais, em seguida, se detalha o circuito utilizado e a forma de registro local adotada em cada um deles. Por fim, as transformações que registram cada nuvem na origem global – as poses – são comparadas diretamente com as poses do *groundtruth* do respectivo *dataset*. Esta etapa se divide na análise do erro de translação e do erro de rotação. A lógica da comparação é simples, espera-se que após a otimização das poses o erro diminua, e assim as poses se aproximem mais do *groundtruth*.

Todas as transformações obtidas pelo registro em pares ao longo do circuito e todas as poses finais, calculadas nas etapas de otimização, estão disponíveis nos apêndices para fins de reprodutibilidade dos resultados. Tomou-se o cuidado de garantir que nenhuma transformação em par fosse estimada com qualidade superior ao *groundtruth*, caso contrário, as poses também poderiam ter qualidade superior, e por motivos óbvios, a comparação do antes e depois estaria comprometida.

4.1 Datasets

Para os experimentos tem-se cinco *datasets*: *Courtyard*, *Facade*, *Arch*, Bremen e UFPR. Os três primeiros pertencem a Theiler et al. (2015) e foram obtidos diretamente no site da universidade *ETH Zurich*². *Courtyard* é composto por 8 nuvens de pontos e representa os arredores de uma tumba no deserto. A característica principal deste *dataset* é a ausência de elementos verticais e formas geométricas definidas. No geral, as nuvens apresentam elevada sobreposição entre si, em média 70 %, e devido à ausência de obstruções na linha de visão laser, todas elas se sobrepõem.

O *dataset Facade* conta com 7 nuvens de pontos e retrata o levantamento de uma fachada, em trabalho de alvenaria. A presença de elementos urbanos é significativa, tem-se casas, carros e pessoas, o que gera artefatos móveis nas nuvens de pontos. Vegetações estão presentes, mas de forma pouco significativa. Neste *dataset* e em *Courtyard*, o desnível entre as estações do LST é elevado, há

² Disponível em: <Automatic registration of partially overlapping terrestrial laser scanner point clouds – Photogrammetry and Remote Sensing | ETH Zurich>. Acesso em: 12/01/2021.

diferenças superiores a 3 metros de altura. A sobreposição entre as nuvens também é elevada, em média 60 %, principalmente devido à proximidade entre as estações, que se situam todas dentro de um raio de 10 m.

O *dataset Arch* é composto por 5 nuvens ao redor do monumento Arco de Tito, em Roma. A quantidade de vegetação neste *dataset* é elevada. Devido ao ambiente urbano e turístico, artefatos móveis também são marcantes. As estações são esparsas, apenas 5 ao longo dos 70 m do circuito. Isto faz com que a sobreposição entre as nuvens de pontos seja baixa, em média 40%. Portanto, este é um dataset desafiador para o registro local FGR. Neste e nos *datasets* anteriores, o *groundtruth* das poses foi obtido por registro manual de alvos reflexivos Riegl com posterior refinamento em ICP *multiview* de software comercial.

O quarto *dataset*, Bremen, foi levantado por (BORRMANN e ELSEBERG, 2008) e está disponível no Repositório de Escaneamentos Robóticos 3D³. São 13 nuvens de pontos obtidas na cidade de Bremen, Alemanha. As estações do LST são esparsas, em média 30 m distantes umas das outras. Elas seguem em volta de uma edificação com arquitetura icônica, cercada de prédios históricos e uma igreja. A sobreposição média entre as nuvens é de aproximadamente 50 %, o *groundtruth* deste dataset foi fornecido, segundo os autores, ele foi calculado utilizando-se SLAM6D com o software de código aberto 3DTK.

O quinto e último *dataset*, disponibilizado pelo professor Daniel Santos, conta com 8 nuvens de pontos obtidas no campus da UFPR. Elas se distribuem em torno de uma edificação simples, com as estações do LST esparsas. Há pouca sobreposição entre as nuvens, em média 35%. Este é o único dataset que não conta com *groundtruth*, portanto, adotou-se como poses *groundtruth* as calculadas com transformações entre pares refinadas por ICP (seção 2.3.3). A Tabela 4 reúne as informações gerais de cada *dataset*. A Figura 20 apresenta cada um deles em perspectiva arbitrária.

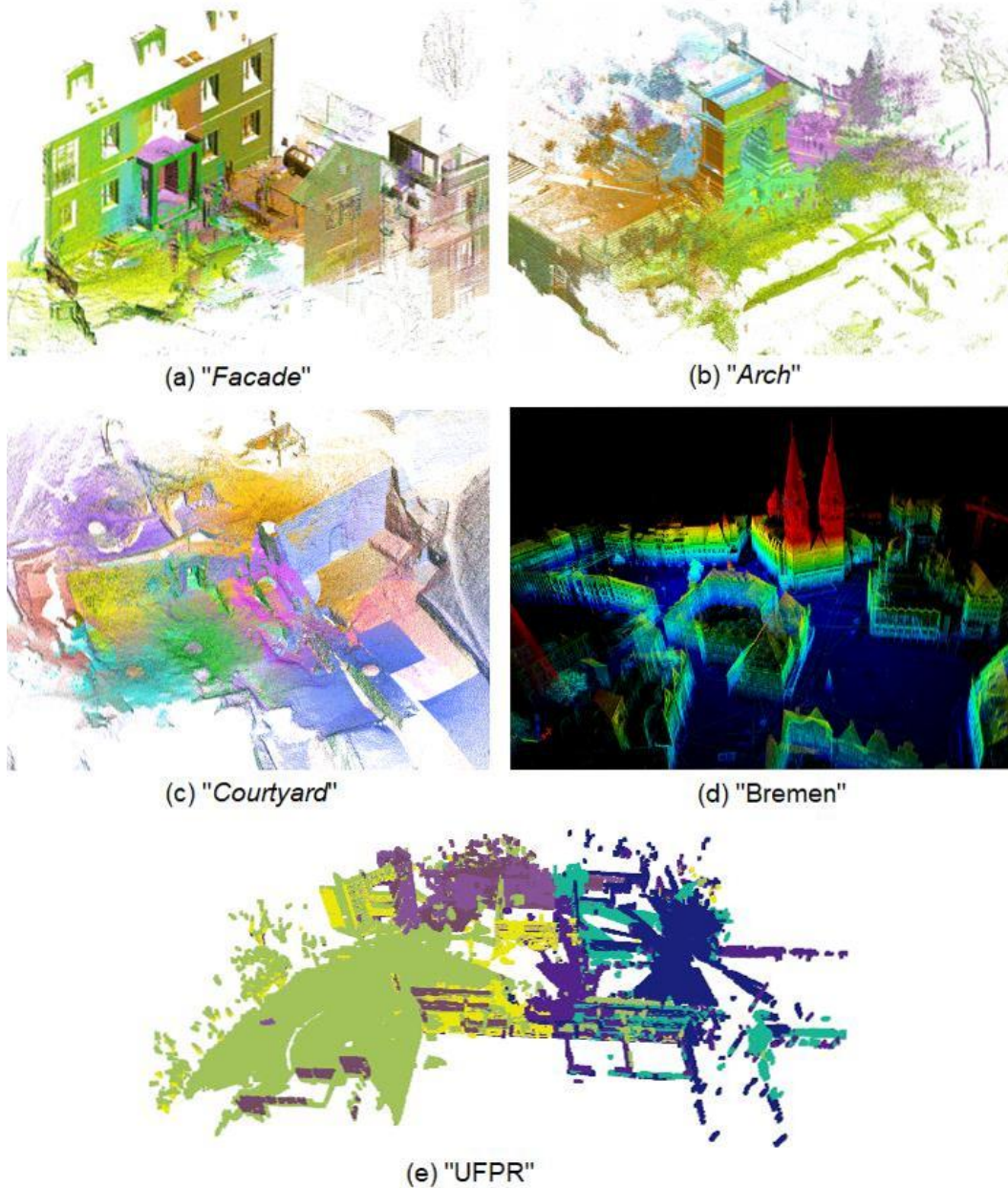
³ Disponível em: <Robotic 3D Scan Repository (uni-osnabrueck.de)>. Acesso em: 12/01/2021.

Tabela 4 - Descrição dos datasets.

Dataset	LST	Nuvens	Pts./ nuvem	Groundtruth	Sobreposição Média (%)
Courtyard	<i>Zoller+Fröhlich Imager 5006i</i>	8	> 20 M	ICP <i>multiview</i>	70
Facade	<i>Faro Focus 3D</i>	7	> 20 M	ICP <i>multiview</i>	60
Arc	<i>Faro Focus 3D</i>	5	> 20 M	ICP <i>multiview</i>	40
Bremen	<i>Riegl VZ400</i>	13	SLAM6D	50	
UFPR	<i>Leica BLK-360</i>	8	> 2 M	ICP	35

Fonte: o autor (2021).

Figura 20 - Datasets utilizados na avaliação do método.



Fonte: elaborada pelo autor (2020).

4.2 Registro local e circuitos

Com exceção do *dataset* Bremen, apenas um circuito fechado por *dataset* será utilizado. A origem global dos sistemas foi definida arbitrariamente na origem da primeira nuvem de pontos dentro do *dataset*. A ordem de registro seguiu-se como no exemplo do método (ver Figura 17), faz-se o registro da segunda nuvem na primeira, da terceira na segunda, e assim por diante, até que o circuito seja fechado. Esta ordem de registro é arbitrária, qualquer outra poderia ser adotada para o estabelecimento do circuito, dado que há sobreposições múltiplas entre as nuvens de todos os *datasets*.

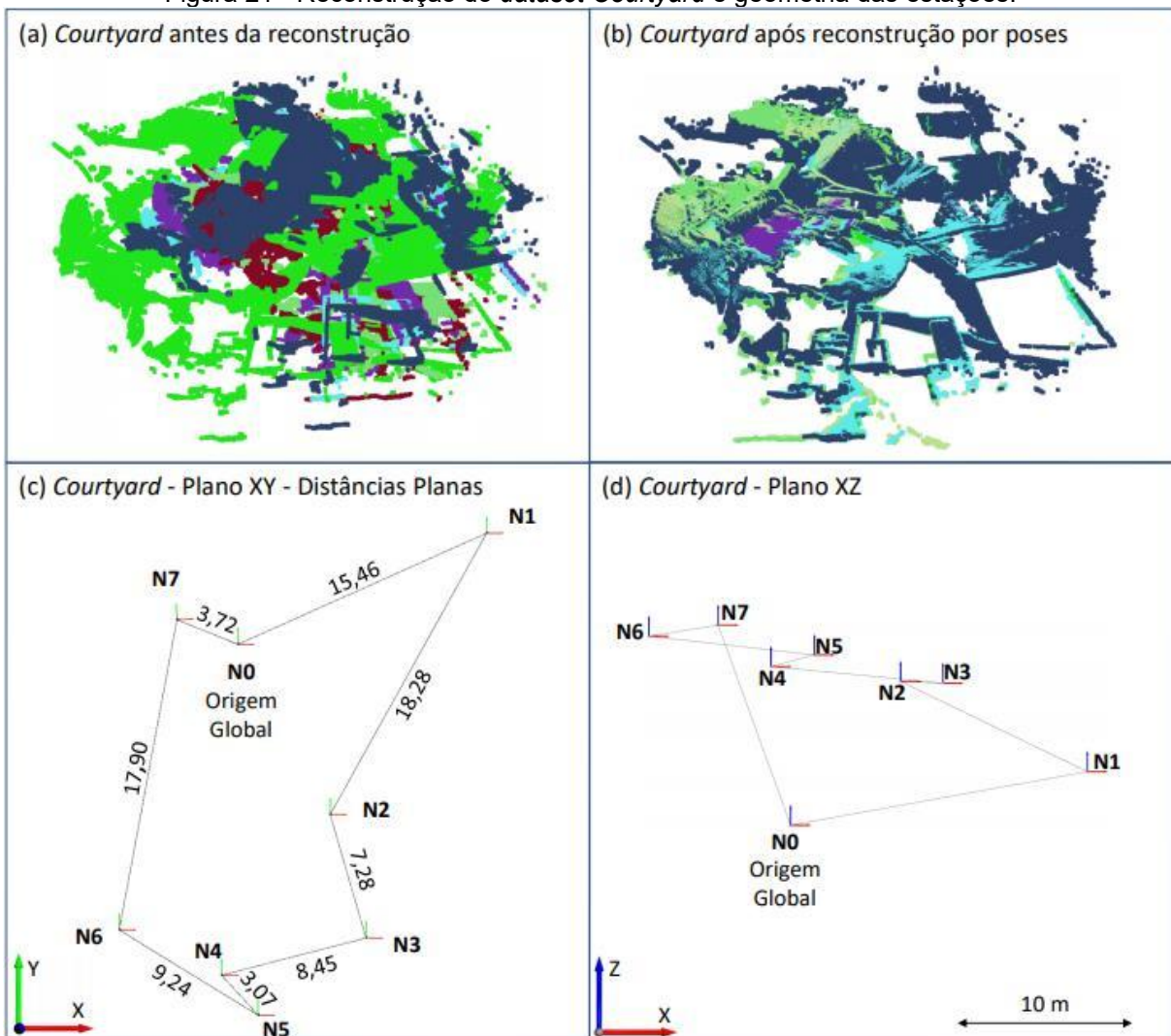
Como detalhado na metodologia, fez-se a tentativa de registro automático em todos os *datasets* com o algoritmo FGR, naqueles em que reconstrução automática falhou, todos os pares do circuito foram registrados manualmente. Independentemente do método de registro grosseiro dos pares, o posterior refinamento com o ICP foi aplicado, mas nesta etapa os pares dos *datasets* de Theiler et al. (2015) apresentaram resultados inesperados.

As transformações refinadas com o ICP deram qualidade superior ao *groundtruth*. Isto é um problema, pois, se as transformações entre pares são melhores que o *groundtruth*, então as poses calculadas através delas também o serão. Isto inviabiliza a comparação entre poses, dado que a referência para comparação terá qualidade inferior ao que se compara. Para evitar este problema, abandonou-se a estratégia de registro *coarse-to-fine* FGR+ICP ou Manual+ICP nos *datasets* de Theiler et al. (2015). Nestes, apenas as transformações da etapa *coarse* foram utilizadas no cálculo das poses.

4.2.1 Courtyard

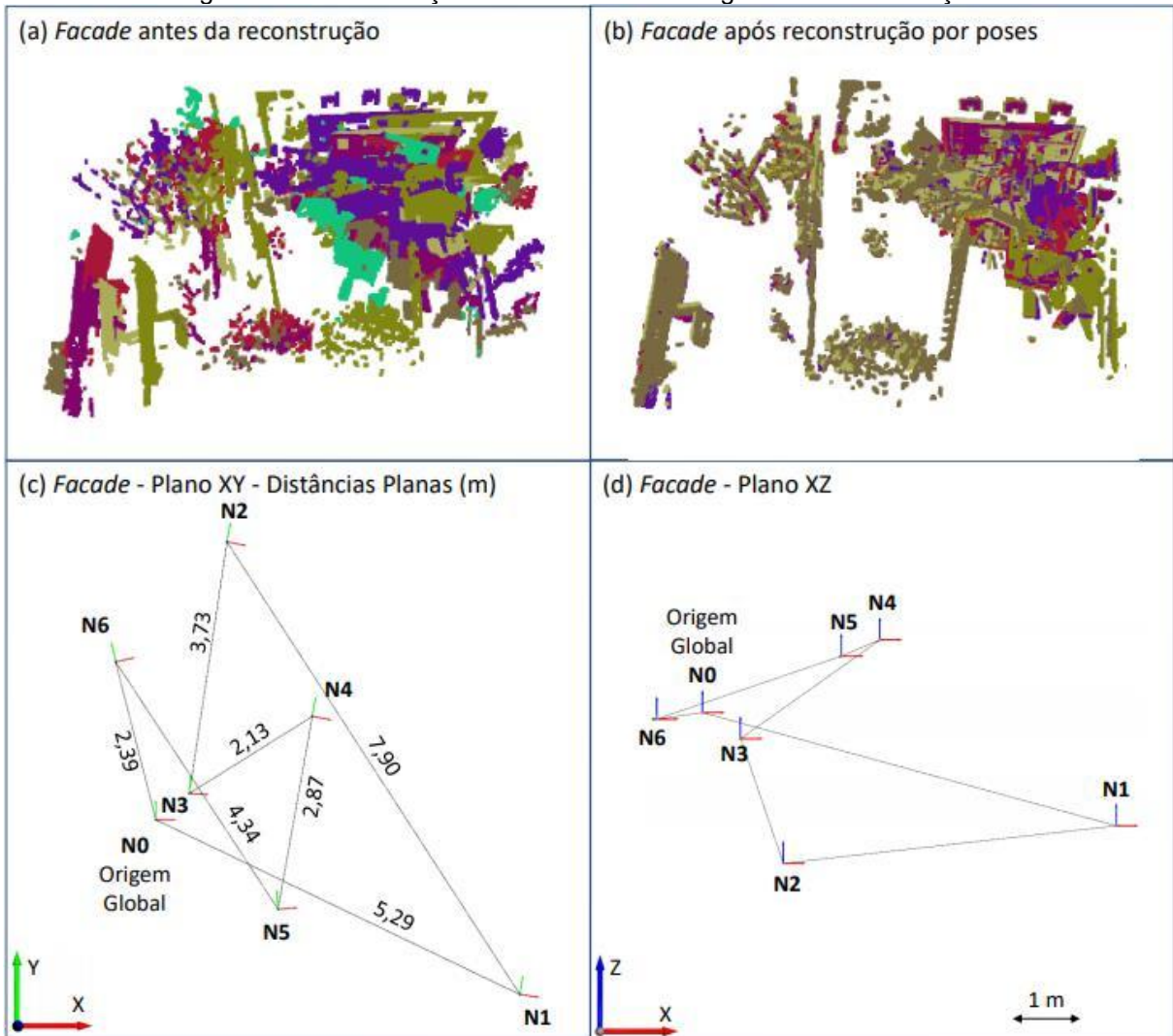
O *dataset Courtyard* foi inteiramente reconstruído pelo FGR. Isto significa que o algoritmo foi capaz de registrar, com qualidade aceitável, todos os pares ao longo do circuito de forma automática. As nuvens de pontos 3D amostradas, antes e após a reconstrução, são apresentadas na Figura 21 (a) e (b). Em (c) e (d) tem-se a visualização em planta (plano XY) e lateral (plano XZ) do circuito em escala.

Figura 21 - Reconstrução do *dataset Courtyard* e geometria das estações.



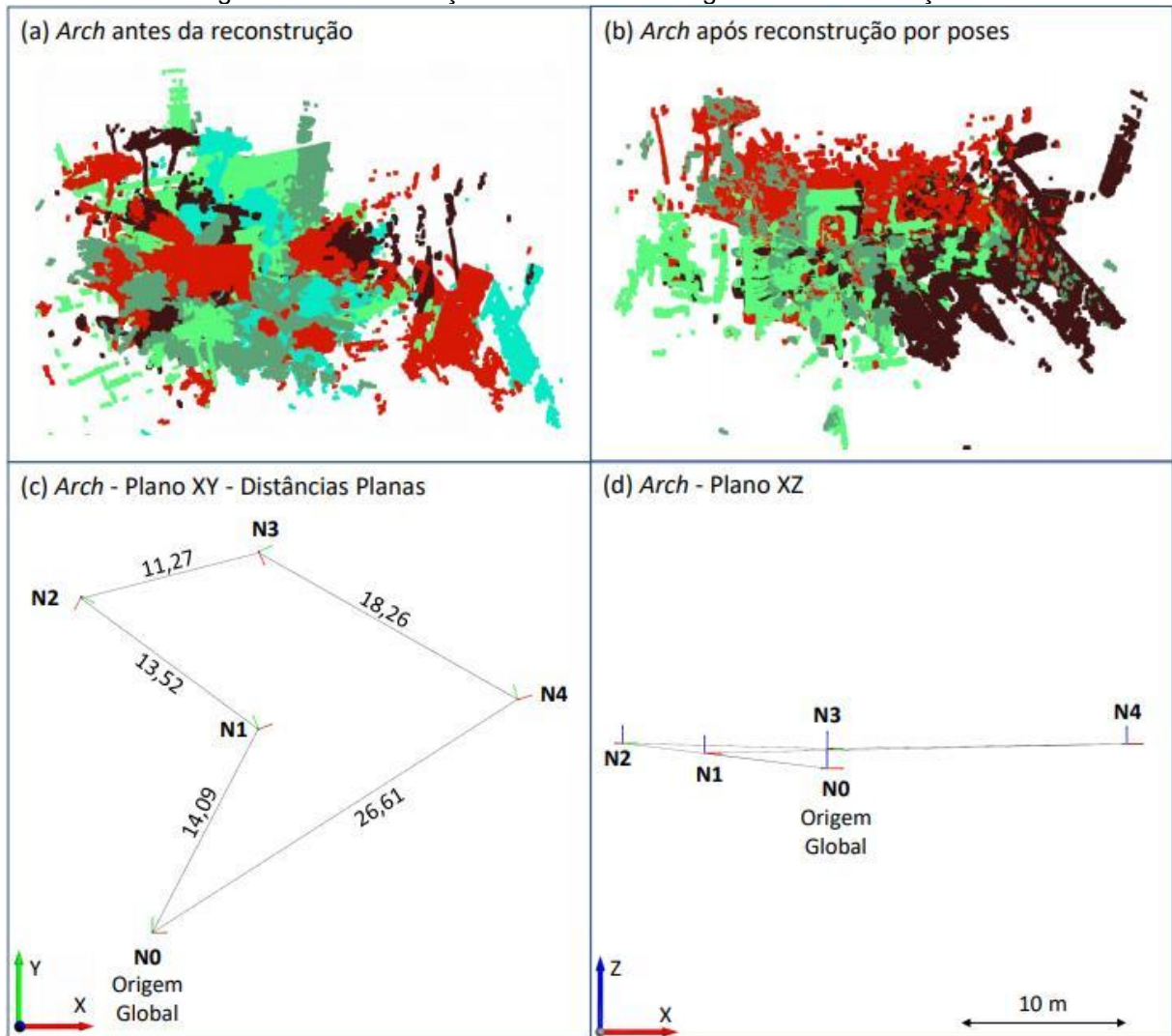
4.2.2 *Facade*

A reconstrução do *dataset Facade* também se deu de forma automática pelo FGR, provavelmente porque este *dataset*, assim como o anterior, apresenta boa sobreposição média entre as nuvens. Os parâmetros de pré-processamento e do algoritmo FGR foram exatamente os mesmos. As nuvens de pontos 3D amostradas, antes e após a reconstrução pelas poses, são apresentadas na Figura 22 (a) e (b). Em (c) e (d) tem-se a visualização da planta (plano XY) e lateral (plano XZ) do circuito em escala.

Figura 22 - Reconstrução do *dataset Facade* e geometria das estações.

4.2.3 Arch

Os pares do circuito no *dataset Arch* foram registrados manualmente. Exaustivos testes foram feitos com os parâmetros do algoritmo FGR e do pré-processamento, mas não foi possível obter transformações aceitáveis em todos os pares do circuito. A pouca sobreposição entre os pares (em média 40 %) é o provável fator responsável pela impossibilidade de automação da reconstrução. No único par com sobreposição superior a 60%, par N3-N2, o registro FGR foi bem-sucedido. As nuvens de pontos 3D amostradas, antes e após a reconstrução pelas poses, são apresentadas na Figura 23 (a) e (b). Em (c) e (d) tem-se a visualização em planta (plano XY) e lateral (plano XZ) do circuito.

Figura 23 - Reconstrução do *dataset Arch* e geometria das estações.

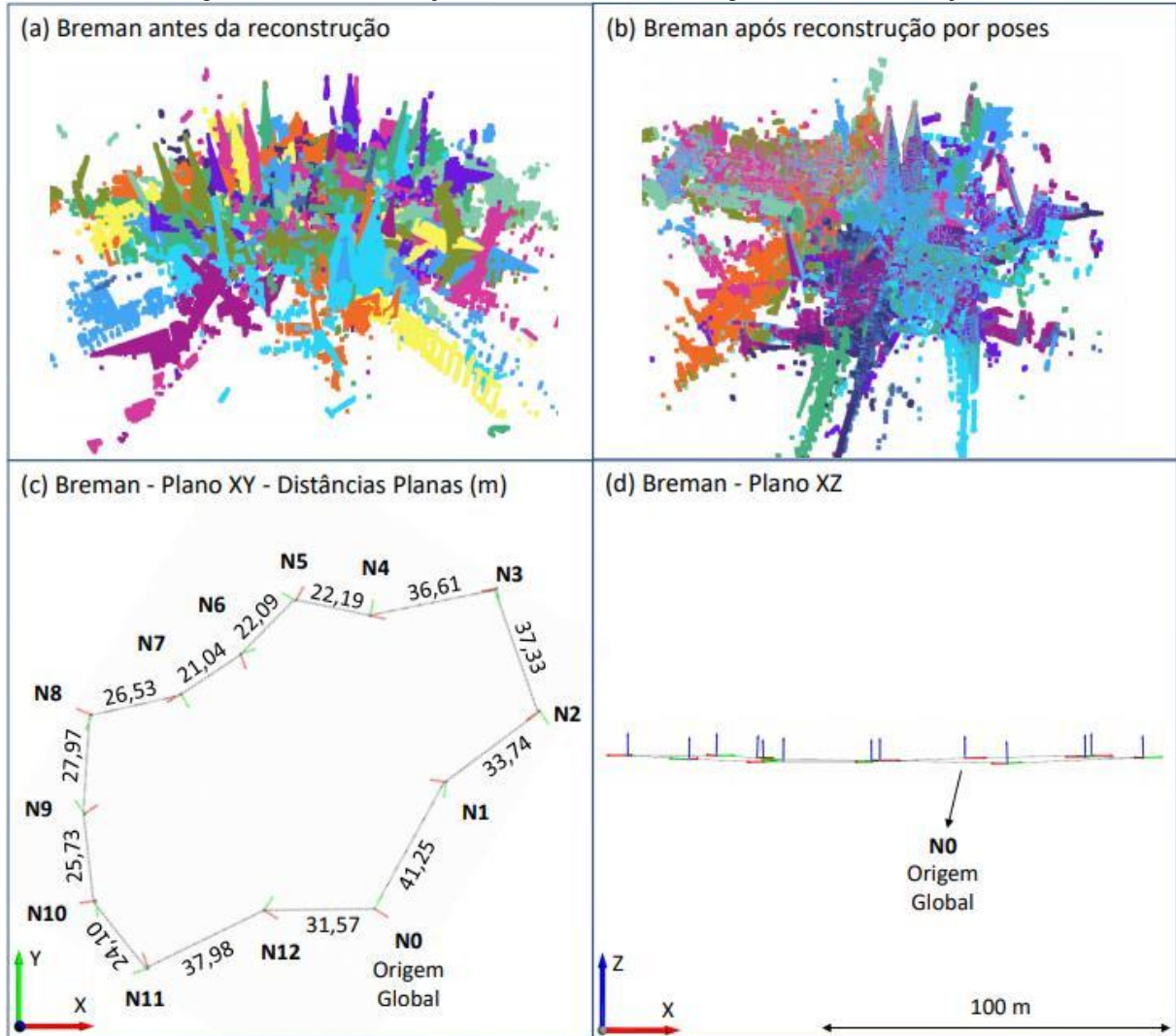
4.2.4 Bremen

Bremen é o maior e mais complexo dos cinco *datasets*, uma vez que apenas dois pares, dos 13 no circuito, foram registrados de forma automática. Sendo assim, o circuito foi inteiramente registrado manualmente. O refinamento dos pares pelo ICP foi feito em seguida, portanto, para o mesmo circuito, tem-se dois conjuntos de poses neste *dataset*, um conjunto de poses calculadas por composição de transformações manuais e outro conjunto de poses calculadas por composição de transformações refinadas (poses ICP).

Diferente dos *dataset* anteriores, de (THEILER et al., 2015), não foi observado o problema de poses superiores ao groundtruth após refinamento com o ICP. As nuvens de pontos 3D amostradas, antes e após a reconstrução pelas poses, são

apresentadas na Figura 24 em (a) e (b). Em (c) e (d) tem-se a visualização da planta (plano XY) e lateral (plano XZ) do circuito em escala.

Figura 24 - Reconstrução do dataset Bremen e geometria das estações.

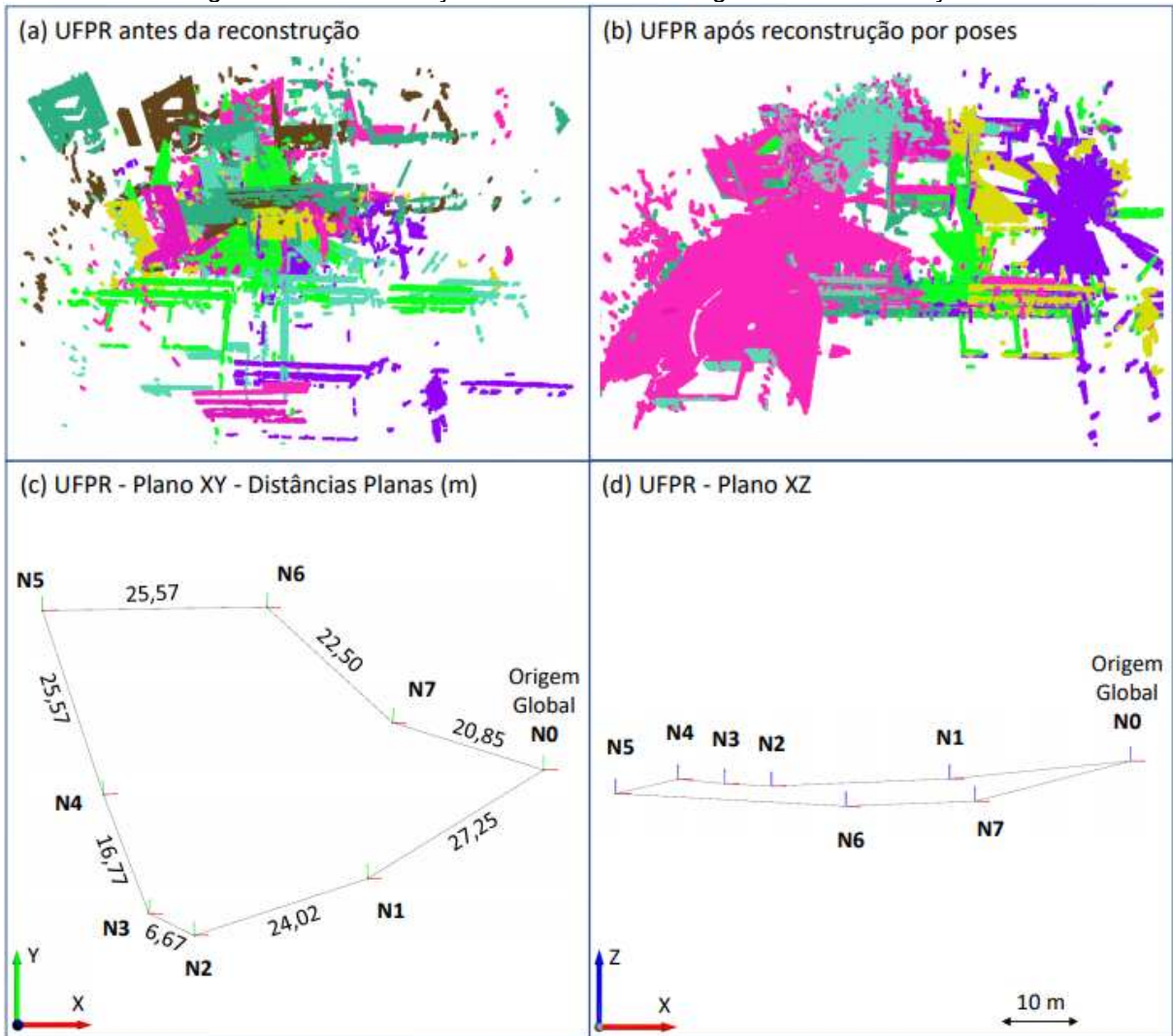


Fonte: o autor (2020)

4.2.5 UFPR

No *dataset* UFPR o algoritmo FGR foi capaz de registrar, de forma automática, os três pares de maior sobreposição, entre os oito do circuito. Portanto, também foi adotado o registro manual com todos os pares. Não foi possível utilizar um circuito adicional, refinado pelo ICP, pois, como explicado antes, o circuito refinado foi adotado como o *groundtruth* deste *dataset*. As nuvens de pontos 3D amostradas, antes e após a reconstrução pelas poses, são apresentadas na Figura 25 (a) e (b). Em (c) e (d) tem-se a visualização em planta (plano XY) e lateral (plano XZ) do circuito em escala.

Figura 25 - Reconstrução do dataset UFPR e geometria das estações.



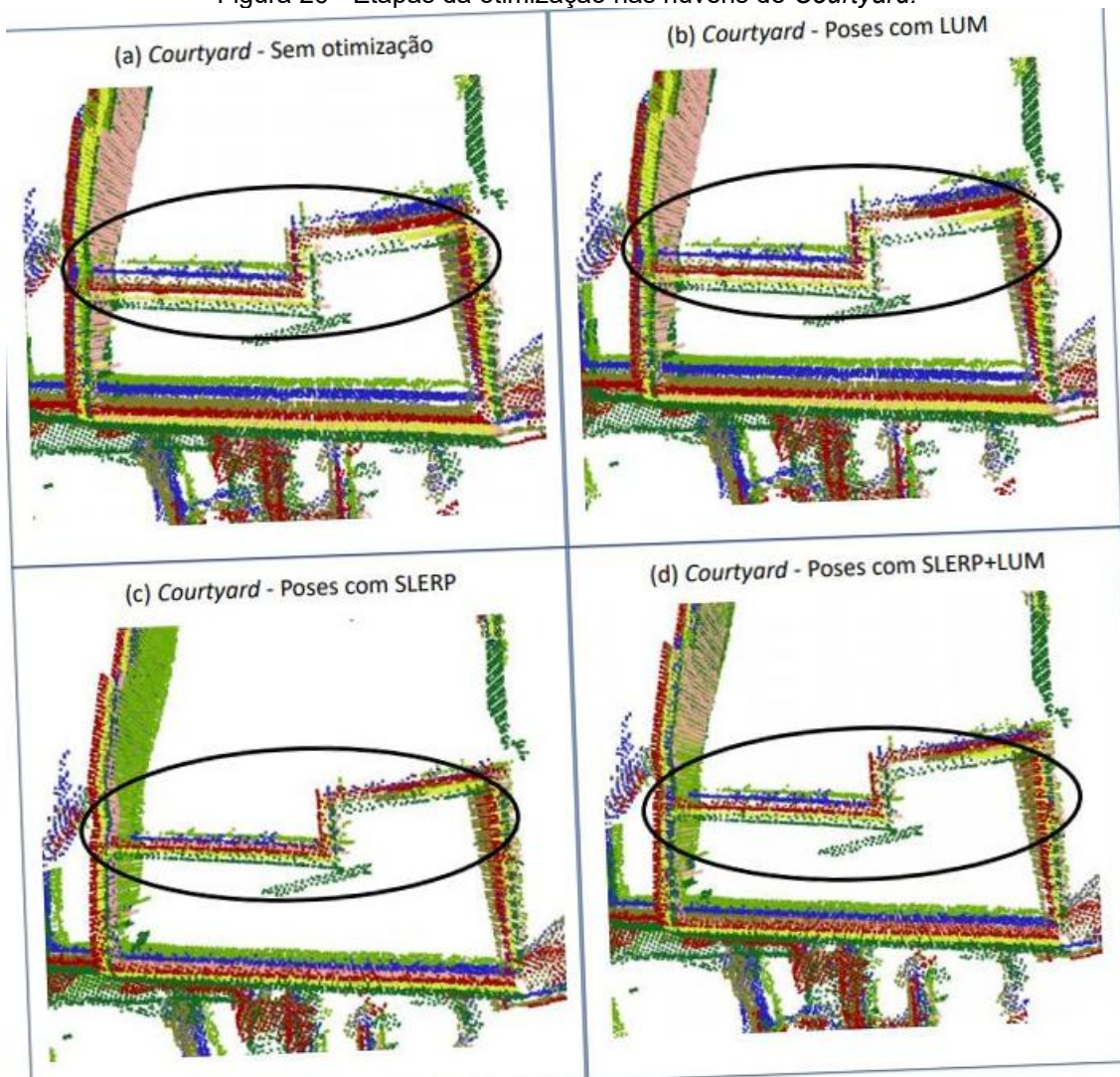
4.3 Avaliação qualitativa das etapas de otimização global

A análise qualitativa das poses se dá pela observação da qualidade da reconstrução do *dataset*. Para isto, observaremos os locais de geometrias facilmente reconhecíveis na cena escaneada. Desta seção em diante, a abreviação LUM será adotada para se referir à otimização obtida com o modelo (LU e MILIOS, 1997), o termo SLERP será adotado para se referir à otimização das rotações; e SLERP+LUM será adotado para se referir à aplicação de ambos os modelos no *dataset* considerado. Como as etapas SLERP e LUM são independentes, apresenta-se a reconstrução dos *datasets* em quatro etapas: (a) poses sem otimização; (b) poses após LUM; (c) poses após SLERP; e (d) poses após SLERP+LUM.

4.3.1 Otimização de Courtyard

A Figura 26 apresenta uma das poucas partes de *Courtyard* com geometria de fácil reconhecimento para uma avaliação qualitativa. Tem-se múltiplas paredes que deveriam estar sobrepostas. Cada nuvem está com uma cor distinta para facilitar a visualização. As mudanças, de acordo com as etapas definidas na subseção anterior, estão nas subfiguras (a), (b), (c) e (d), respectivamente. Manteve-se a posição da câmera inalterada em cada subfigura.

Figura 26 - Etapas da otimização nas nuvens de *Courtyard*.



Fonte: o autor (2020)

Na Figura 26, as nuvens que compõem o *dataset Courtyard* praticamente não se alteram com a otimização das translações, como mostram as subfiguras (a) e (b). Dentre todas as reconstruções do *dataset*, a melhor está na subfigura (c), em que

apenas o modelo SLERP foi aplicado, pois é onde se observa maior coerência geométrica das nuvens de pontos. Esta coerência fica clara na região destacada com uma elipse preta, pois as múltiplas paredes, que são evidentes em (a) e (b), tendem a se organizar como uma única após as otimizações SLERP e SLERP+LUM. Nenhuma das reconstruções é aceitável para um projeto, mas claramente há um impacto positivo do modelo SLERP.

Neste *dataset*, o modelo SLERP teve maior impacto positivo que a combinação SLERP+LUM. Não se sabe o porquê deste resultado, uma hipótese é de que a otimização das translações talvez tenha prejudicado a reconstrução devido a algum erro grosseiro na translação durante o registro dos pares. Para testar essa hipótese, uma análise quantitativa e outra qualitativa foi feita em cada um dos pares registrados. A primeira análise se resumiu em procurar por erros através da observação direta, mas não se viu nenhum erro grosseiro entre pares de nuvens do registro.

Na segunda avaliação, duas métricas foram utilizadas. Ambas estão disponíveis na biblioteca Open3D para avaliação do registro em pares de nuvens de pontos. A primeira métrica se chama *inlier RMSE*, ela mede a raiz do erro quadrático médio (REQM) entre todas as correspondências encontradas pelo FGR. Quanto menor, melhor. A segunda métrica se chama *Fitness*, ela mede a sobreposição que a nuvem de pesquisa faz na nuvem alvo, é calculada como a razão entre os pontos da nuvem de pesquisa que estão mais próximos dos pontos da nuvem alvo. Seu valor é dado em porcentagem, quanto maior, melhor.

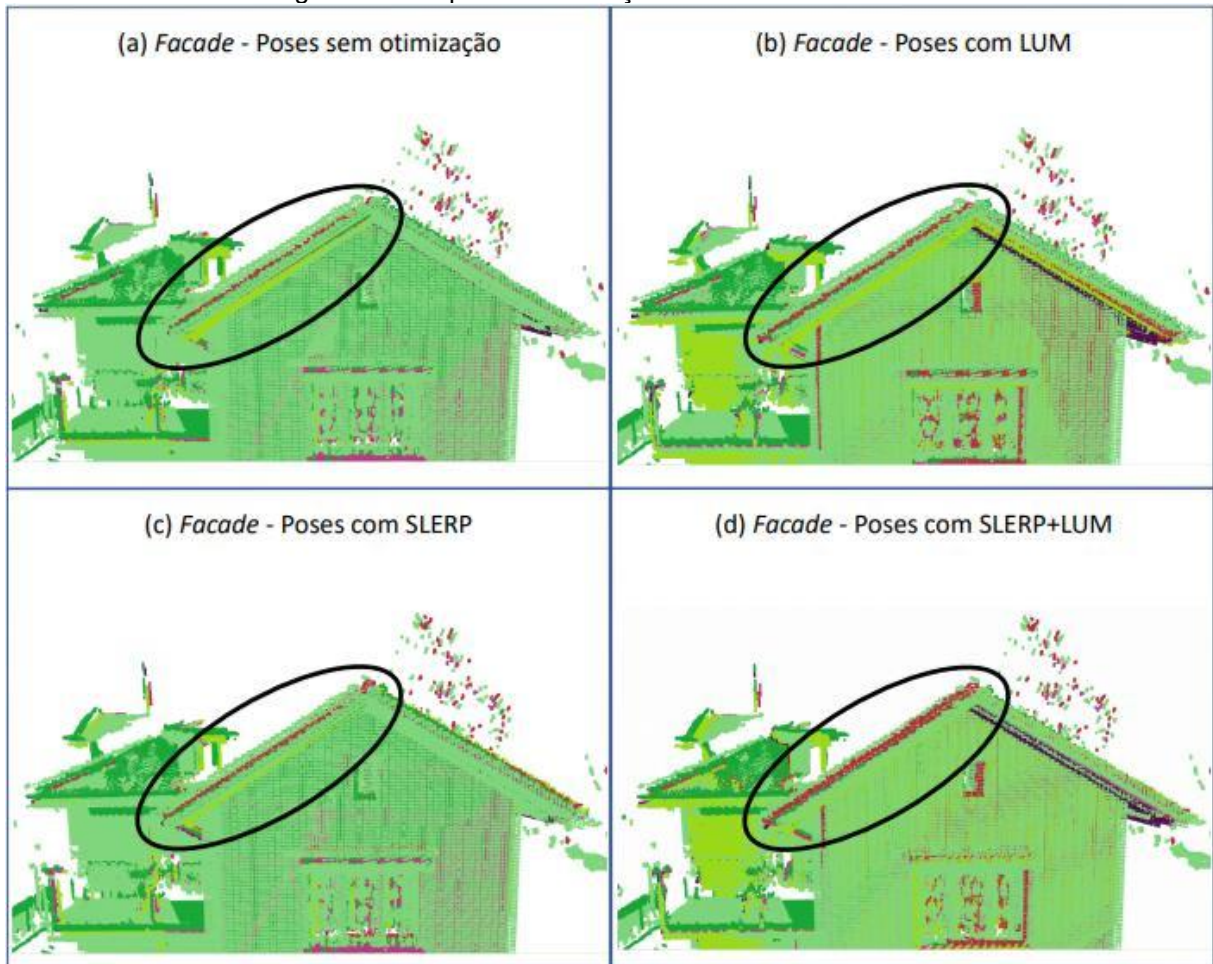
Nenhuma das duas métricas retornaram valores atípicos em nenhum dos pares de *Courtyard*. O RMSE se manteve próximo dos 10 cm em todos, o que faz sentido considerando o *voxel* utilizado na amostragem das nuvens, de 10 cm. A sobreposição entre os pares também não explica o comportamento atípico da otimização. Todos os pares entregaram valores de sobreposição próximos aos que seria encontrado se fossem utilizadas as transformações do *groundtruth*. Como este trabalho não é dedicado à avaliação dos registros em pares, não se fez outras investigações a respeito deste problema.

4.3.2 *Facade*

A Figura 27 segue a mesma lógica anterior, desta vez evidencia-se uma fachada com telhado para facilitar a análise das nuvens nas diferentes etapas. Cada nuvem

está com uma cor distinta para facilitar a visualização. As mudanças, de acordo com as etapas, estão nas subfiguras (a), (b), (c) e (d). Manteve-se a posição da câmera inalterada em cada subfigura.

Figura 27 - Etapas da otimização nas nuvens de *Facade*.



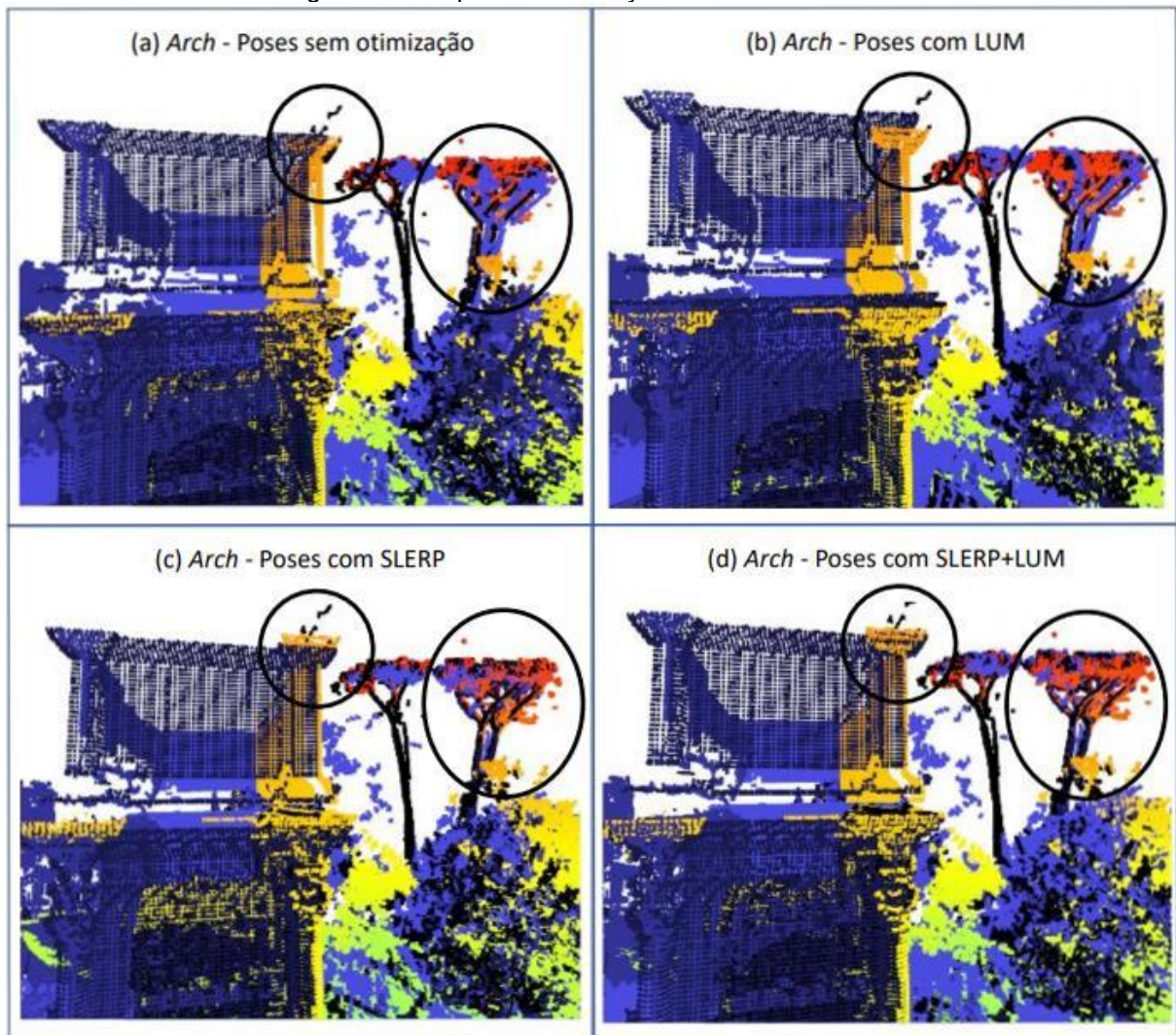
Fonte: o autor (2020)

As mudanças em *Facade* são sutis, mas é possível observar na Figura 27 (d) que, em relação a (a), houve certa melhoria na organização do telhado. Outro indício de aprimoramento é a ponta do telhado mais pontiaguda em (d). Novamente, a etapa (b) não teve efeitos perceptíveis no *dataset*. As diferenças na etapa SLERP, em (c), indicam leve melhora em relação às poses não otimizadas e às poses com LUM. Desta vez não é possível dizer, com base apenas na observação direta, se a otimização SLERP foi melhor que SLERP+LUM, mas é possível afirmar que em ambas houve melhoras sutis em relação a (a) e (b).

4.3.3 Arch

Tem-se, na Figura 28, a visualização do lado esquerdo do Arco de Tito. Ao fundo, duas árvores. O mesmo padrão das análises anteriores é mantido.

Figura 28 - Etapas da otimização nas nuvens de Arch



Fonte: o autor (2020)

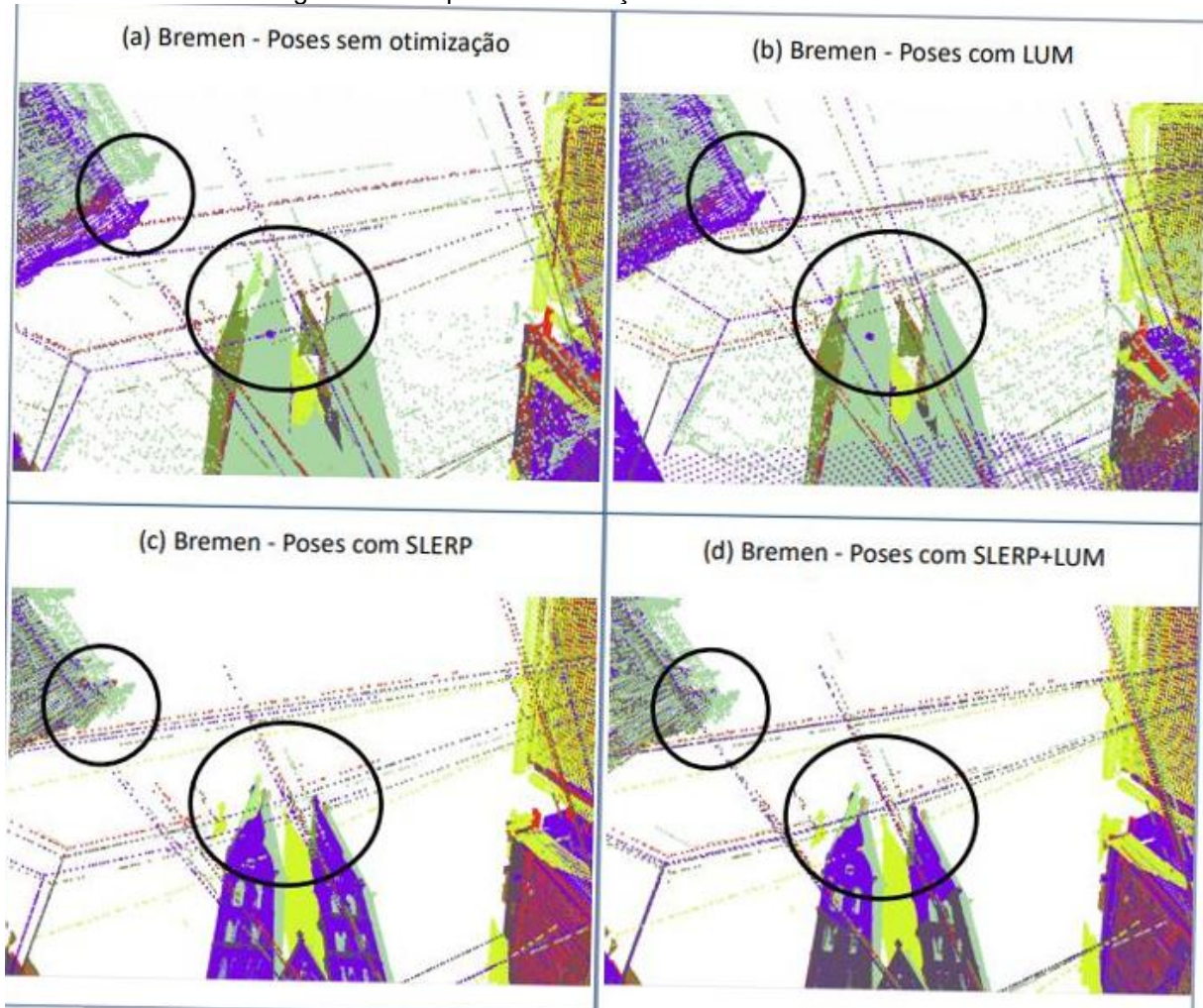
As mudanças em *Arch* são um pouco mais evidentes, na Figura 28 (a), é possível observar que o canto direito do arco está relativamente organizado, mas a árvore no fundo, a direita, está duplicada. Após a otimização LUM (b), o fundo de árvores permanece igual, mas a nuvem amarela é deslocada para baixo em relação à nuvem roxa, desorganizando o canto direito do arco. Após a otimização SLERP (c), a duplicação da árvore ao fundo desaparece, mas a nuvem amarela foi deslocada para cima, novamente desorganizando o canto do arco. Com a otimização SLERP+LUM (d),

a duplicação da árvore some e o canto direito do arco se organiza. Ao menos visualmente, que a reconstrução deste *dataset* é melhor com a otimização completa das poses, SLERP+LUM.

4.3.4 Bremen

Bremen conta com dois circuitos, as mudanças neste dataset são apresentadas na Figura 29, para o circuito manual, e na Figura 30, para o circuito refinado com ICP.

Figura 29 - Etapas da otimização nas nuvens de Bremen



Fonte: o autor (2020)

A perspectiva da Figura 29 simula um observador que, estando na praça principal de Bremen, observa a igreja por entre os fios que se estendem acima da sua vista. O que este observador hipotético veria no mundo real? Ele notaria a intersecção de quatro fios formando um quadrilátero, duas torres pontiagudas da igreja, ao fundo, e

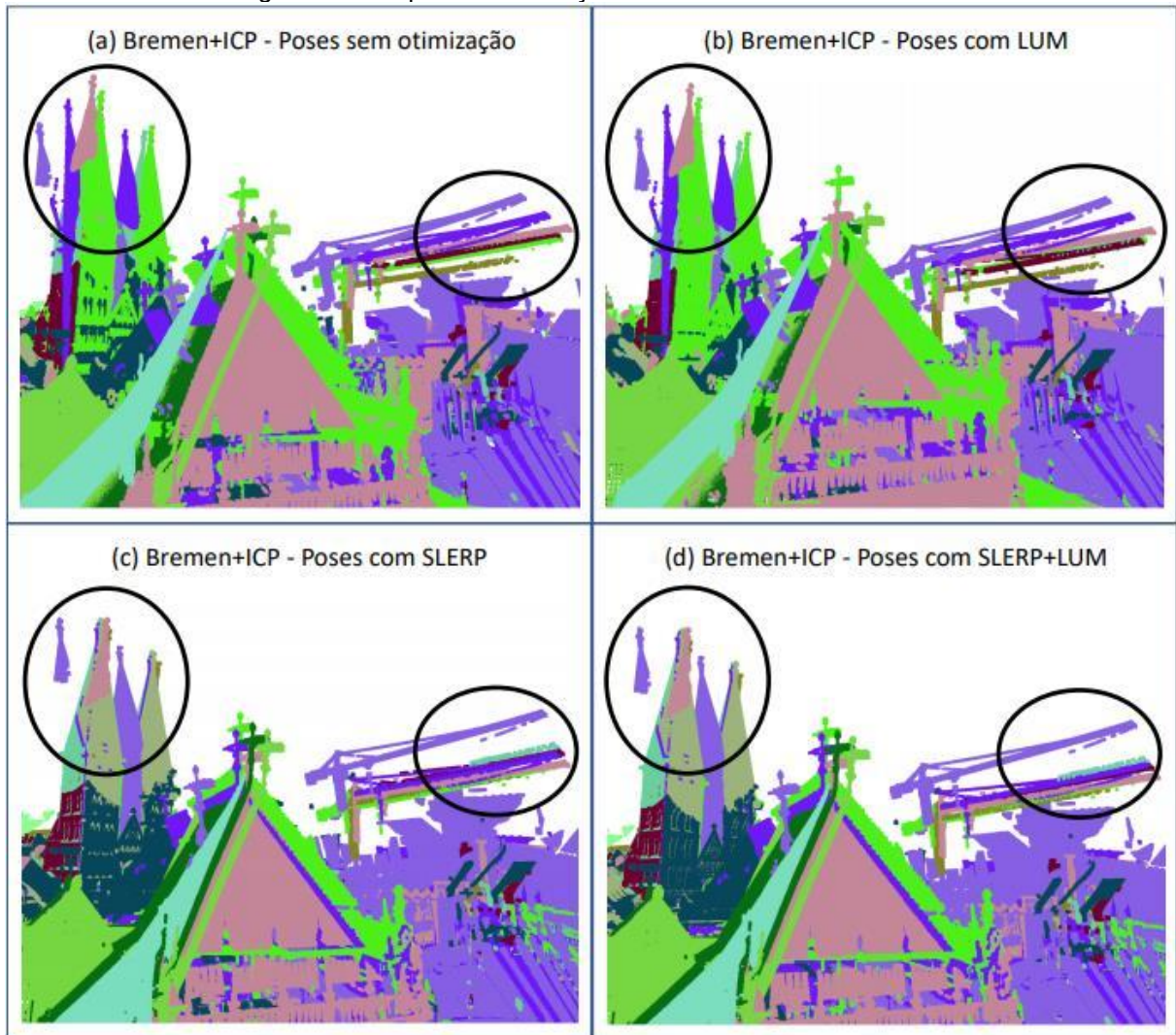
alguns telhados na sua visão periférica. O que se observa na Figura 29 em (a), (b), (c) e (d) é diferente disso, mas se aproxima do que foi descrito, conforme as etapas de otimização ocorrem na ordem citada.

No geral, o *dataset* está mal reconstruído pelas poses, pois o erro presente em cada par é grande. O registro manual foi feito com poucas correspondências. Além disso, o *dataset* Bremen possui o maior de todos os circuitos. Por um lado, isto fez os efeitos de cada etapa da otimização serem evidentes, por outro, o resultado da melhor otimização, SLERP+LUM na subfigura (d), está longe da qualidade aceitável para um projeto com nuvens de pontos 3D.

Analisando cuidadosamente a cena, a nuvem amarela é a de maior desvio em relação às outras. Como ocorreu nos *datasets* anteriores, praticamente não se veem mudanças após o ajuste na etapa LUM, apenas a nuvem azul é levemente deslocada para cima, o que acaba sobrepondo o chão na parte inferior da subfigura (b). Com a etapa SLERP, praticamente tudo muda de lugar na subfigura (c), há melhorias nas torres da igreja, nos fios e telhados que são fáceis de visualizar. Por fim, aplicando SLERP+LUM, uma leve melhora ocorre na subfigura (d), pois a fiação se torna menos dispersa.

Na Figura 30 se analisa o mesmo circuito, a diferença é que cada par foi refinado pelo algoritmo ICP. As nuvens estão em cores diferentes e a perspectiva também mudou, agora a vista é por cima dos telhados. Vê-se o topo das torres da igreja e outros objetos, como o telhado com ponta em forma de cruz. Ao fundo, na direita das subfiguras, há um guindaste, que aparece em 6 das 13 nuvens.

Figura 30 - Etapas de otimização nas nuvens de Bremen+ICP



Fonte: o autor (2020)

Praticamente todas as observações que foram feitas na Figura 29 se mantêm para a Figura 30. Não se observa mudanças da etapa (a) para (b), a maior diferença aconteceu na etapa (c), com o modelo SLERP, e uma das nuvens continua com grande desvio em relação às outras. Na Figura 29, esta nuvem era a amarela, na Figura 30, é a roxa.

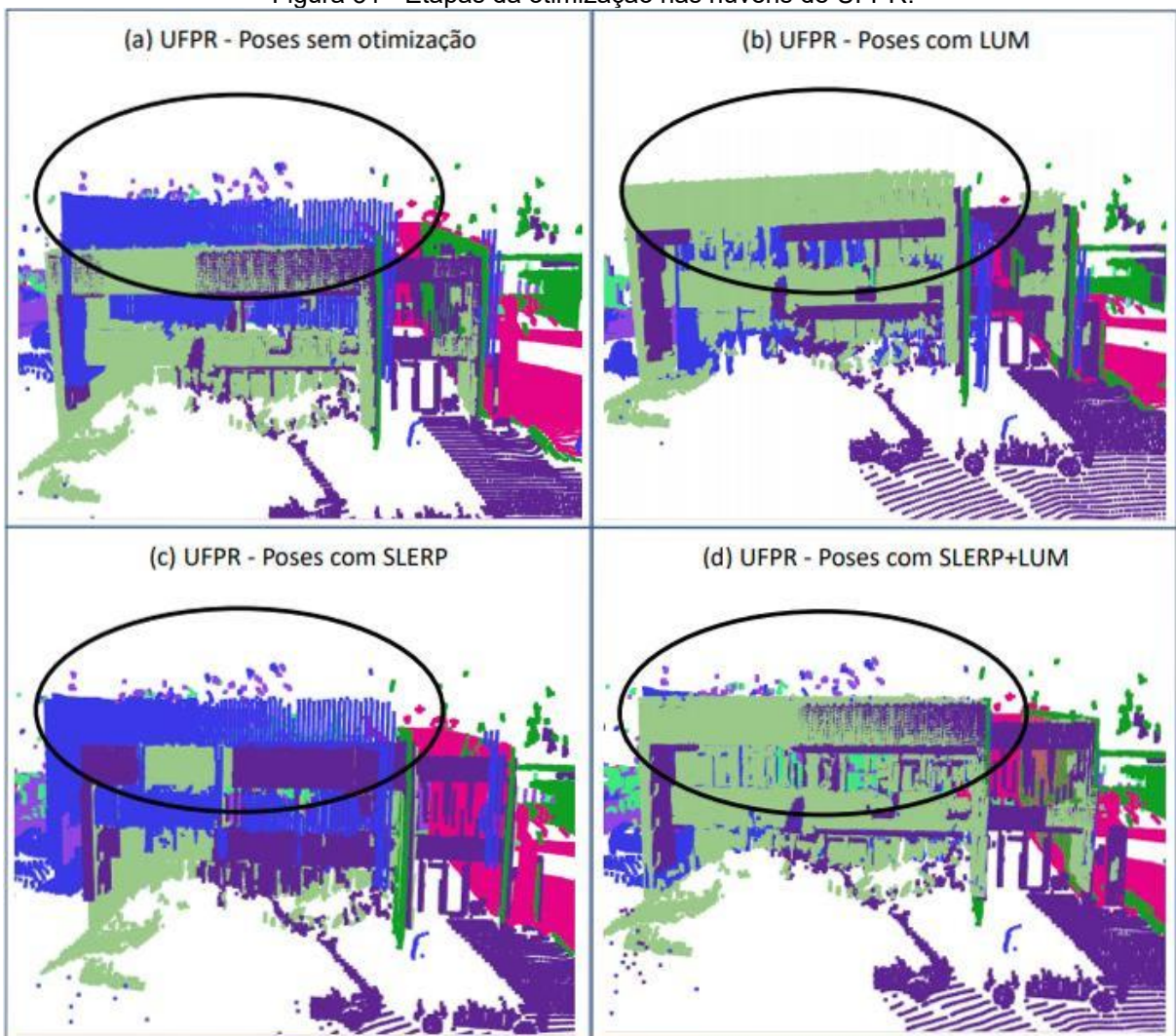
Observando o lado esquerdo das subfiguras (c) e (d), no topo da igreja vê-se que a nuvem roxa não se fundiu às outras, que tendem a formar uma única ponta. O guindaste, no canto direito, mostra o mesmo comportamento, várias nuvens se aproximaram mais, mas a nuvem roxa permaneceu separada das demais. Por fim, no telhado ao centro, as pontas estão menos dispersas, indicando uma melhora na reconstrução do *dataset*, embora ainda com erros grosseiros.

Não se conseguiu amenizar este erro aumentando-se a quantidade de iterações do ICP, o que implica fortes mínimos locais em quase todas as transformações estimadas pela seleção manual de correspondências. Uma alternativa para isto é mover a nuvem de pesquisa para bem próximo da nuvem alvo antes de iniciar o registro.

4.3.5 UFPR

A Figura 31 apresenta o resultado das etapas de otimização no *dataset* UFPR.

Figura 31 - Etapas da otimização nas nuvens de UFPR.



Fonte: o autor (2021)

Começando pela reconstrução sem otimização, na Figura 31 (a), percebe-se que a fachada do prédio está mal organizada, devido ao deslocamento da nuvem azul em

relação à nuvem verde. Após a etapa LUM (b), as nuvens são significativamente niveladas, mostrando, pela primeira vez, uma mudança perceptível na etapa LUM. Na etapa SLERP, a nuvem azul é deslocada para a frente do prédio, e as outras nuvens se organizam ainda mais, como a vermelha. Por fim, com a otimização SLERP+LUM, a reconstrução da fachada do prédio se apresenta melhor do que em todas as etapas anteriores.

4.4 Avaliação quantitativa das etapas de otimização global

Nesta etapa se analisa a diferença que cada otimização faz na pose das nuvens de cada *dataset*. Diferenças entre poses $\|T_i^g - T_i\|$ se subdividem em erro na translação $\|t_i^g - t_i\|$ e erro na rotação $\|R_i^g - R_i\|$, g refere-se às poses do *groundtruth*, e i varia de 1 até a n -ésima pose do *dataset*. O cálculo do erro na translação será dado pelo RMSE (*Root Mean Square Error*), que se reduz a uma distância euclidiana. Para resumir nossa análise, não se observa cada componente Δx , Δy e Δz , apenas a distância total dada por $RMSE_{t_i} = \sum(\Delta x_i^2 + \Delta y_i^2 + \Delta z_i^2)^{1/2}$.

Rotações seguem a mesma lógica, a definição de RMSE se aplica perfeitamente nas subtrações entre matrizes de rotação, pois coincide com a definição da métrica de Frobenius para matrizes do grupo $SO(3)$. Logo, subtraem-se as matrizes de rotação e calcula-se o somatório dos resíduos ao quadrado. A raiz quadrada deste número será um escalar adimensional $d \in [0, 2\sqrt{2}]$, que conforme (HUYNH, 2009), satisfaz a todos os axiomas de uma métrica no espaço das rotações. Não usaremos ângulos de Euler, pois não definem uma métrica e poluem a análise gráfica com três componentes para o erro de rotação em cada pose.

Relembramos que rotações e translações são otimizadas de forma independente. O refinamento das rotações afeta as translações, mas o ajuste das translações não tem impacto nas rotações. Nos trabalhos desta natureza, isto justifica que se observe apenas o erro nas translações, mas este trabalho tem como foco as rotações, portanto, avaliaremos o erro na rotação e seu impacto na translação. Como na análise qualitativa, o erro na translação é avaliado em quatro etapas em cada *dataset*. Nas rotações são apenas duas etapas, antes e após SLERP. As variações no RMSE das rotações foram pequenas, o limite superior do gráfico está sempre em 0,3. No RMSE das translações as variações foram grandes, não foi possível manter a escala fixa.

4.4.1 Courtyard

O dataset *Courtyard* conta com 8 nuvens de pontos 3D, isto significa 7 poses em relação a uma origem global. A Figura 32 apresenta o gráfico do RMSE nas translações das poses, em relação ao *groundtruth*, calculado após cada uma das etapas, (a) sem otimização, (b) LUM, (c) SLERP e (d) SLERP+LUM. A Figura 33 apresenta o gráfico do RMSE das rotações antes e após a otimização SLERP.

Figura 32 - RMSE entre translações otimizadas e *groundtruth*, *Courtyard*.

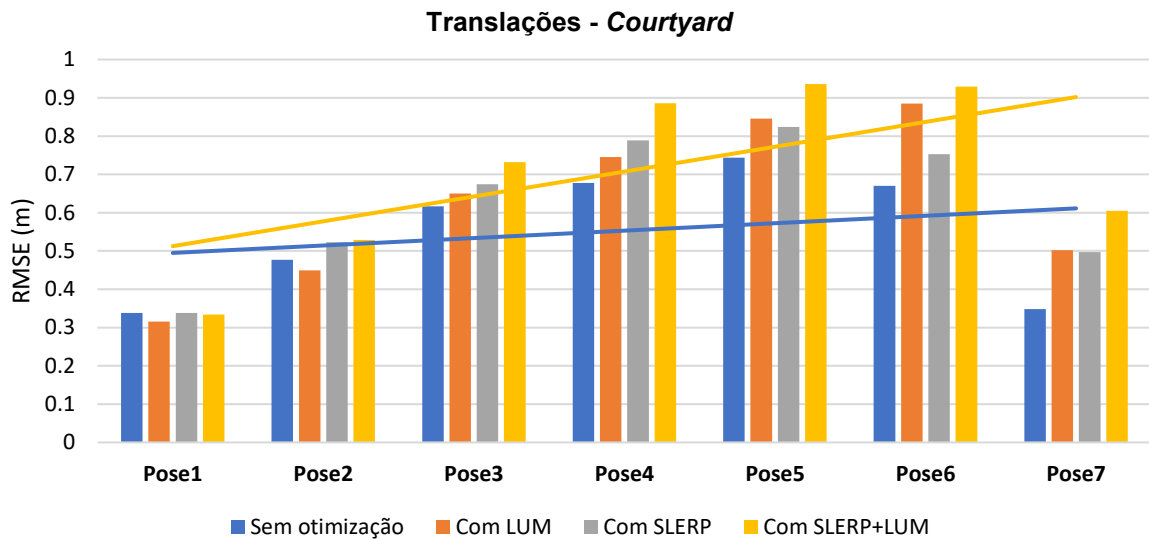
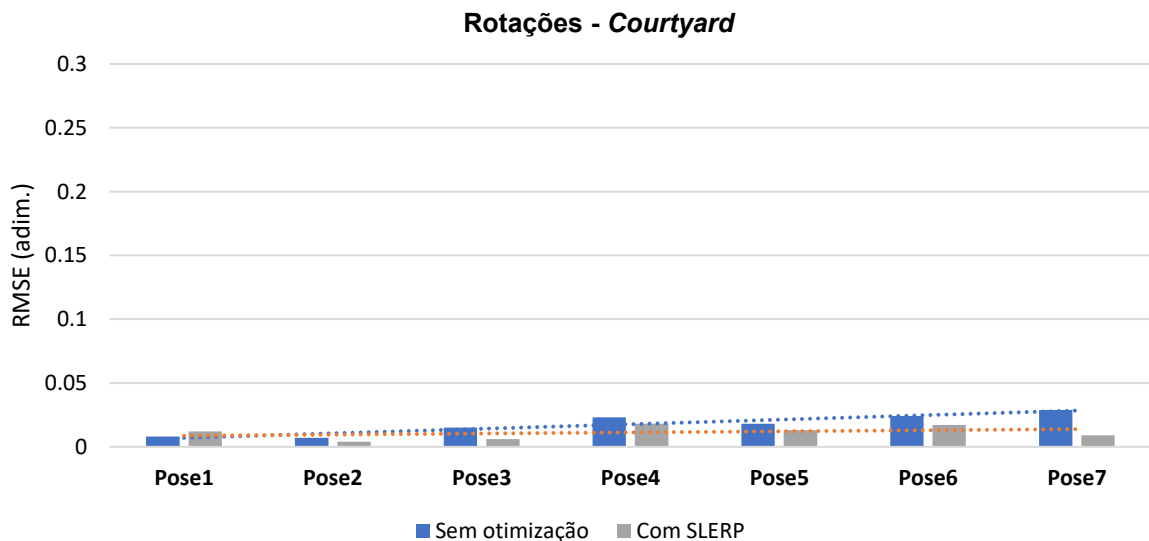


Figura 33 - RMSE entre rotações otimizadas e *groundtruth*, *Courtyard*.



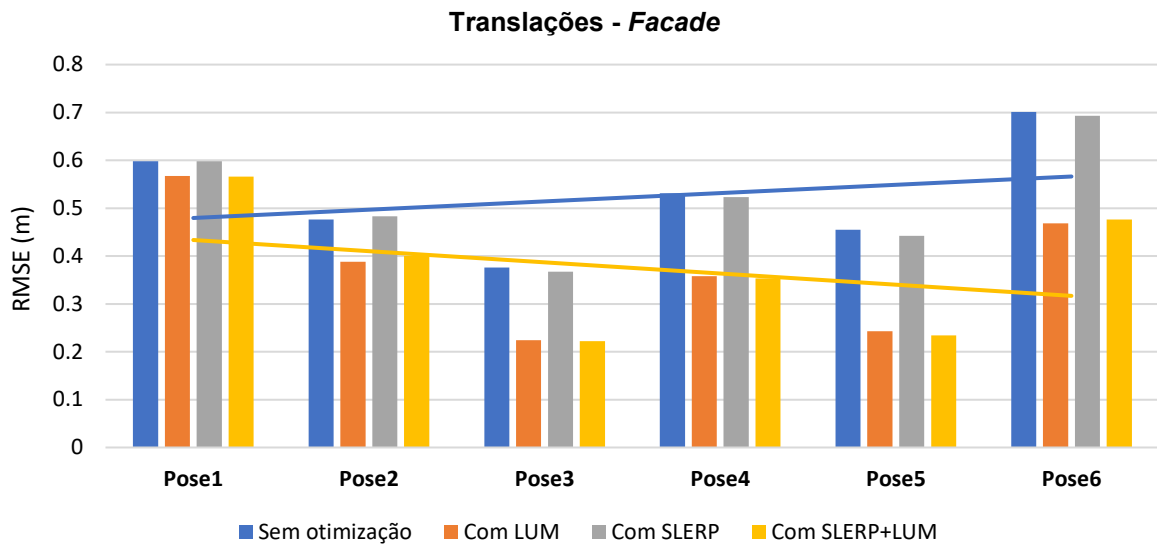
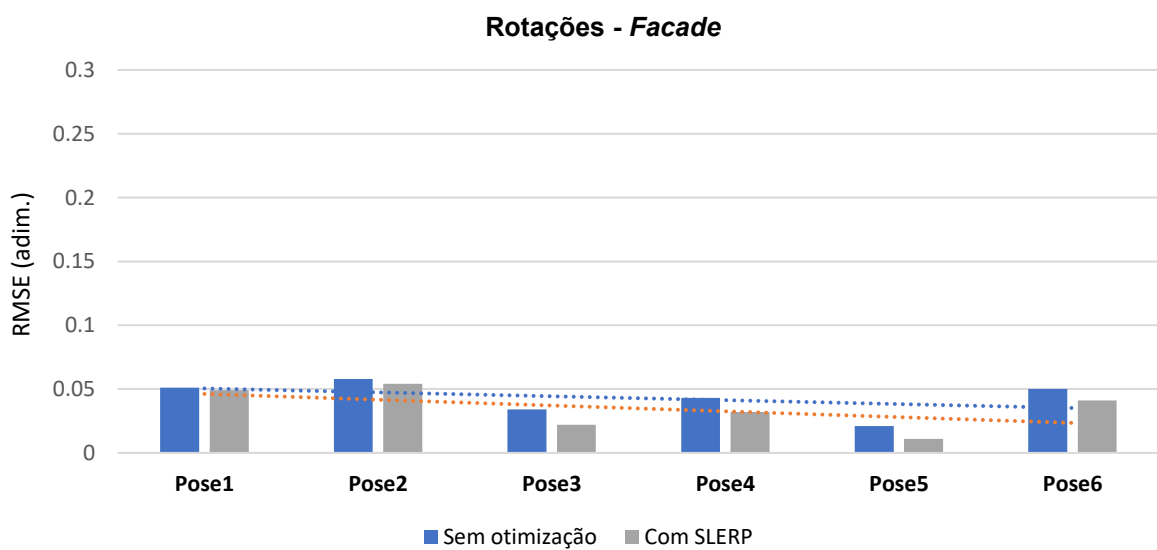
Na Figura 32, o erro entre cresce com as poses, como espero da composição multiplicativa ao longo do circuito. Na última pose, sem otimização, o erro é só 1 cm maior que na primeira, o que mostra um caso fortuito dos erros anulando-se com a composição. Com exceção da primeira pose, todas as outras se distanciaram do *groundtruth*. A aplicação dos modelos LUM, SLERP e SLERP+LUM contrariou o resultado que se espera de uma otimização. A linha de tendência azul mostra que o erro das poses sem otimização é mais uniforme, pois se aproxima mais da horizontal. Também é mais baixa, o que significa menor magnitude total do RMSE.

Por outro lado, na Figura 33, com exceção da rotação na primeira pose, todas as outras se aproximaram mais do *groundtruth*. As duas retas pontilhadas mostram que a técnica SLERP se comportou como o esperado, isto é, diminuiu o erro de forma mais acentuada nas poses finais e aumentou levemente nas iniciais. Pode-se imaginar que a redução de erro na rotação não foi suficiente para compensar um possível aumento de erro introduzido pela otimização LUM. Todavia, se isto fosse verdade, a etapa SLERP, isoladamente, teria otimizado as translações, mas não é isto que se observa.

No geral, embora seja contrário ao esperado, esse caso atípico concorda com a análise qualitativa descrita anteriormente, em *Courtyard*. Uma qualidade inferior havia sido observada com a otimização LUM, mas houve clara melhoria após a etapa SLERP, ainda que o gráfico da Figura 32 diga que após SLERP a reconstrução deveria ser pior. Novamente, não se tem uma hipótese para este comportamento, pois a análise do circuito mostrou que todos os pares de nuvens foram registrados com qualidade aceitável, além disso, a etapa de ajuste das rotações reduziu o erro em cada rotação, embora seu impacto nas translações seja negativo.

4.4.2 *Facade*

O dataset *Facade* conta com 7 nuvens de pontos 3D, são 6 poses em relação a uma origem global. As Figuras 34 e 35 apresentam o RMSE nas translações e rotações de cada pose em relação ao *groundtruth* do *dataset*.

Figura 34 - RMSE entre translações das poses otimizadas e *groundtruth*, *Facade*.Figura 35 - RMSE entre rotações das poses otimizadas e *groundtruth*, *Facade*.

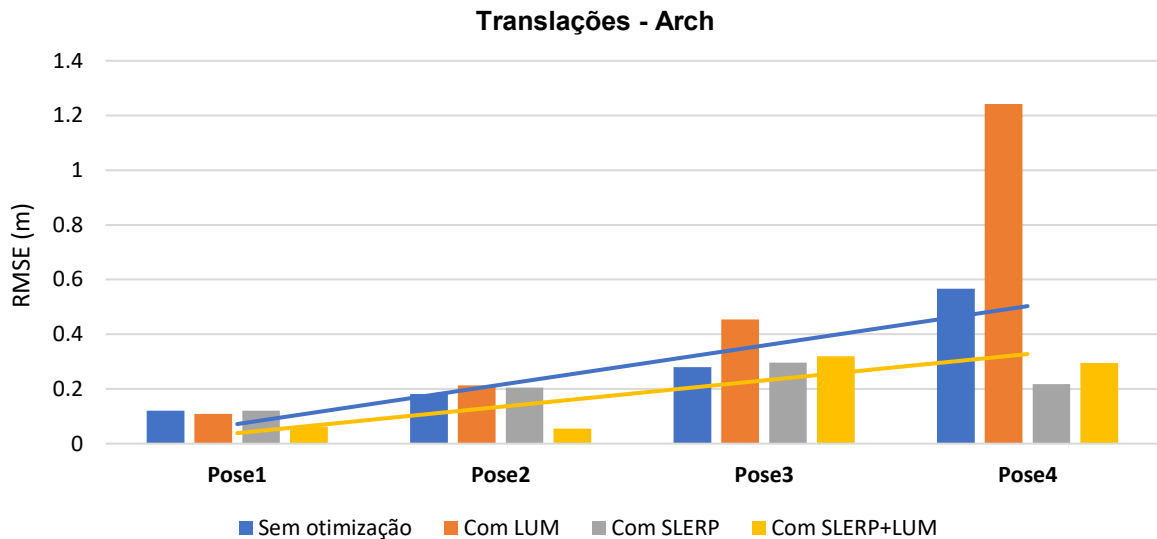
Observa-se, nos gráficos das Figuras 34 e 35, que todas as poses de *Facade* se aproximaram mais do *groundtruth*, tanto nas rotações quanto nas translações. Também fica claro, na Figura 34, que este *dataset* se beneficiou mais da otimização das translações. Nas poses 2 e 6, por exemplo, o resultado teria sido melhor se apenas LUM fosse aplicada. O refinamento das rotações por SLERP as aproximou do *groundtruth*, como mostra a Figura 35, mas isto não alterou significativamente a qualidade das translações. Praticamente toda a melhora adveio da aplicação do modelo LUM. De forma geral, o erro ao longo do circuito diminuiu, como mostram as

linhas de tendência azul e amarela. A inclinação de ambas as retas é quase a mesma, então não podemos afirmar que o erro se tornou mais uniforme, mas se inverteu na direção de crescimento. No gráfico da Figura 35, o mesmo ocorre com as rotações.

4.4.3 Arch

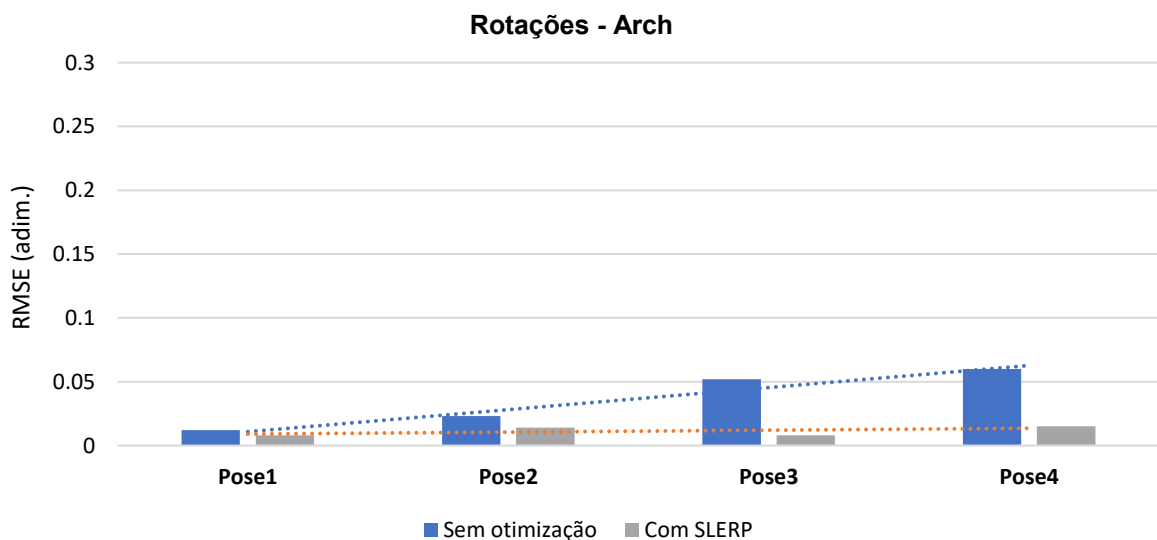
O *dataset Arch* conta com 5 nuvens de pontos 3D, portanto, são 4 poses. As Figuras 36 e 37 apresentam os gráficos do RMSE, em relação ao *groundtruth*, das translações e rotações de cada pose.

Figura 36 - RMSE entre translações das poses otimizadas e *groundtruth*, Arch.



Fonte: o autor (2021).

Figura 37 - RMSE entre rotações das poses otimizadas e *groundtruth*, Arch.

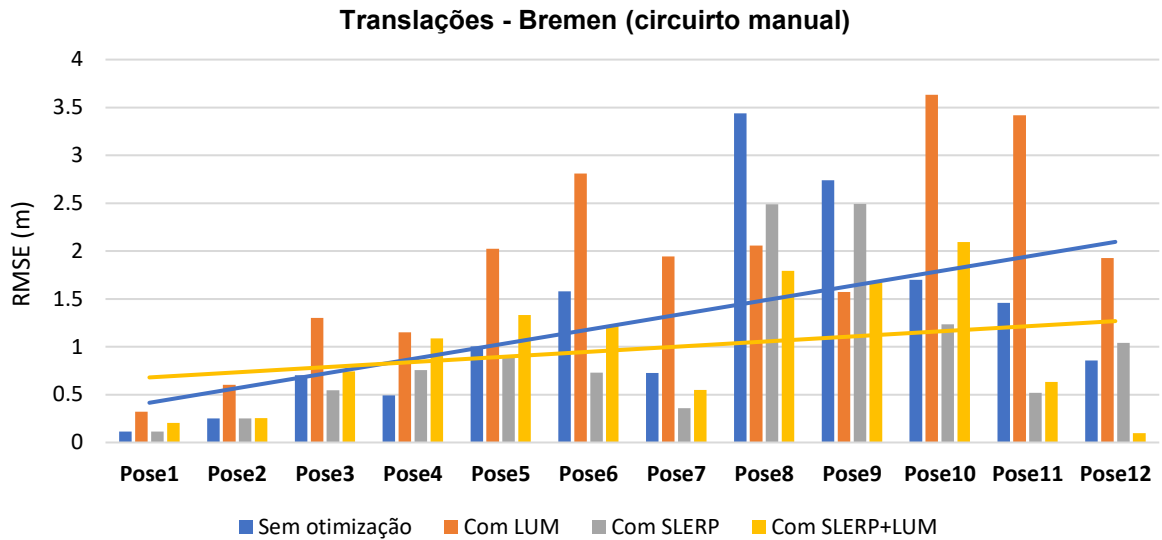
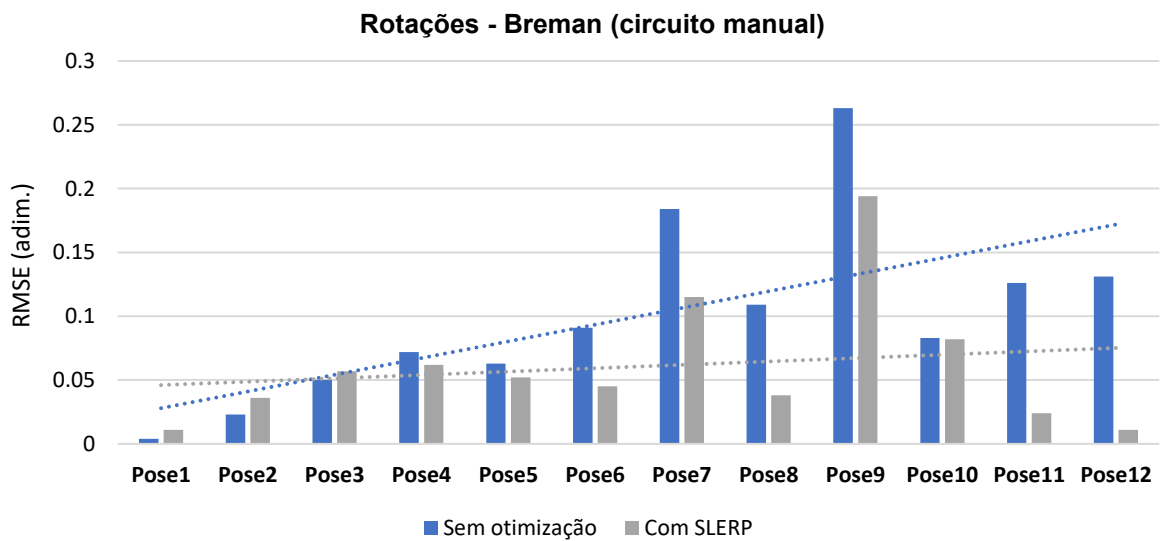


Fonte: o autor (2021).

Nas figuras 36 e 37, o erro ao longo das poses de *Arch* aumenta nas rotações e translações, como esperado da composição. Na Figura 36, o ajuste das translações aumentou o erro nas poses 2, 3 e 4, nesta última, de forma drástica. Na Figura 37, o ajuste das rotações as aproximou muito do *groundtruth*. O impacto da etapa SLERP nas translações foi maior nas poses 3 e 4, por outro lado, nas poses 1 e 2, o benefício apenas ocorre com a aplicação conjunta dos modelos. Neste *dataset*, em ambas as figuras, as linhas de tendência indicam um circuito com erro menor e mais uniforme.

4.4.4 Bremen

O *dataset* Bremen conta com 13 nuvens de pontos 3D, ou seja, 12 poses em relação a uma origem global. As Figuras 38 e 39 apresentam os gráficos do RMSE, em relação ao *groundtruth*, das translações e rotações, respectivamente.

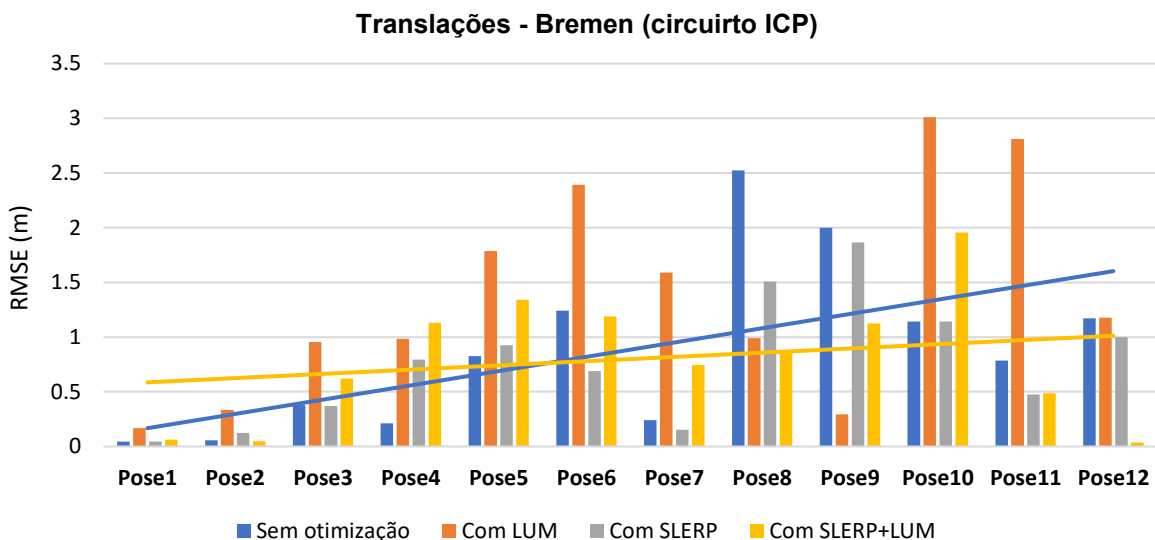
Figura 38 - RMSE entre translações das poses otimizadas e *groundtruth*, Bremen.Figura 39 - RMSE entre rotações das poses otimizadas e *groundtruth*, Bremen.

No gráfico da Figura 38, o erro aumenta com o crescimento das poses, pois mais transformações em pares estão acumulando-se. Após a etapa LUM, em praticamente todas as poses o erro aumentou, as exceções foram as poses 8 e 9. Com a otimização SLERP o erro tende a aumentar nas primeiras poses e diminuir nas últimas, pois o modelo faz uma distribuição do erro acumulado ao longo do circuito. Neste *dataset*, fica claro que o erro não cresce uniformemente, dado que poses mais distantes não necessariamente possuem maior erro.

A tendência de nivelamento do modelo fica mais evidente na Figura 39, onde tem-se o RMSE da rotação. Observa-se que o erro aumentou nas quatro primeiras poses e diminuiu nas outras 8, a única exceção foi a pose 10. A inclinação das retas ajustadas aos dados, que diminui após a otimização SLERP e LUM, indica mais uniformidade e menor erro total ao longo do circuito.

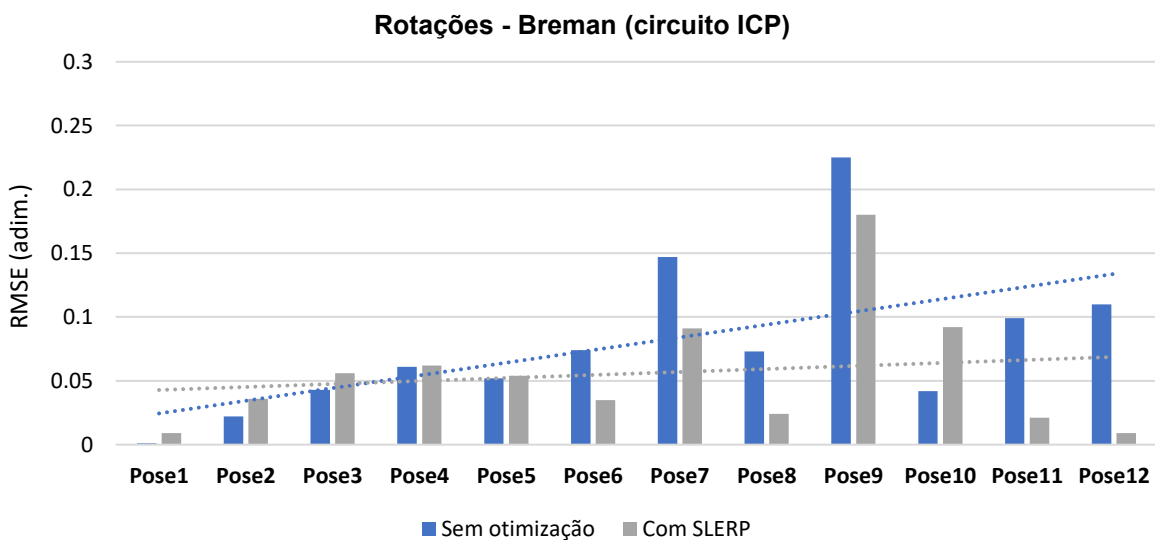
Nas Figura 40 e 41 tem-se os mesmos gráficos, mas desta vez para o circuito Bremen+ICP. A única diferença notável entre estes e os anteriores é a magnitude dos erros, que diminuiu, como esperado para um circuito foi refinado com o ICP. Toda a distribuição do RMSE ao longo das poses continua igual.

Figura 40 - RMSE entre translações das poses otimizadas e *groundtruth*, Bremen+ICP.



Fonte: o autor (2021)

Figura 41 - RMSE entre rotações das poses otimizadas e *groundtruth* Bremen+ICP.



Em resumo, o modelo aumentou o erro de algumas poses em ambos os circuitos, mas diminuiu bastante em outras, como nas poses 8 e 12, que se beneficiaram da aplicação de ambas as otimizações.

4.4.5 UFPR

O *dataset* UFPR conta 8 nuvens de pontos de 3D, isto é, tem-se 7 poses. Nas Figuras 42 e 43 apresentam-se os gráficos do RMSE das poses, em relação ao *groundtruth*, para translações e rotações, respectivamente.

Figura 42 - Distâncias entre translações das poses otimizadas e *groundtruth*, UFPR.

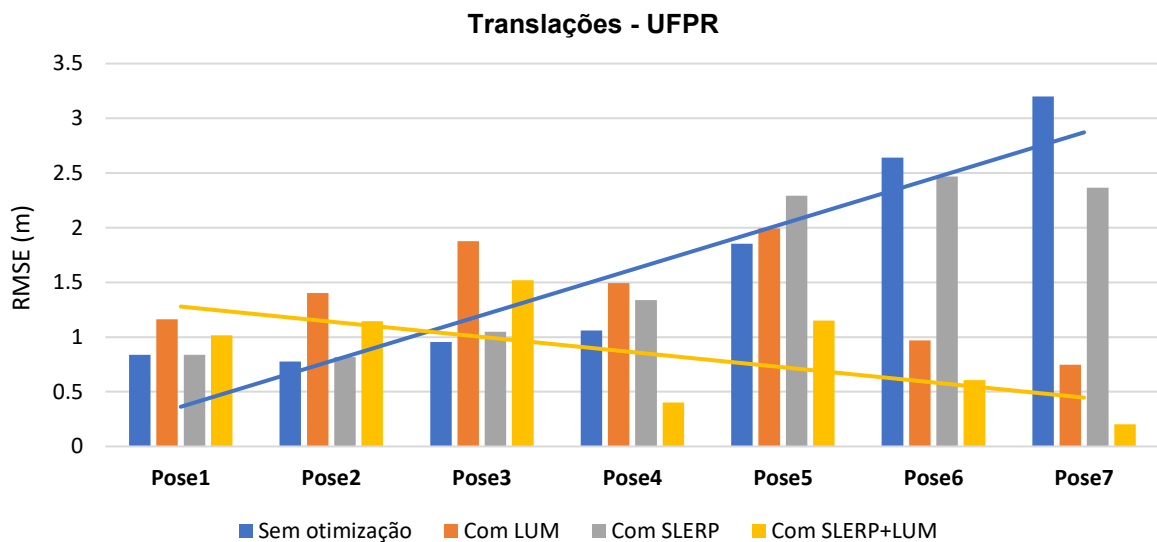
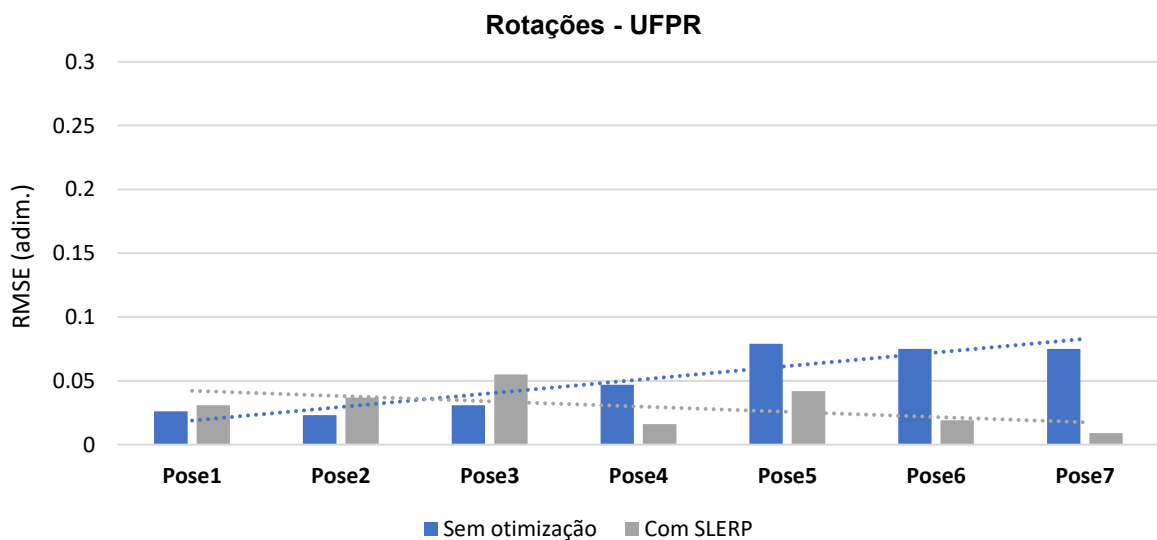


Figura 43 - Distâncias entre rotações das poses otimizadas e *groundtruth* UFPR.

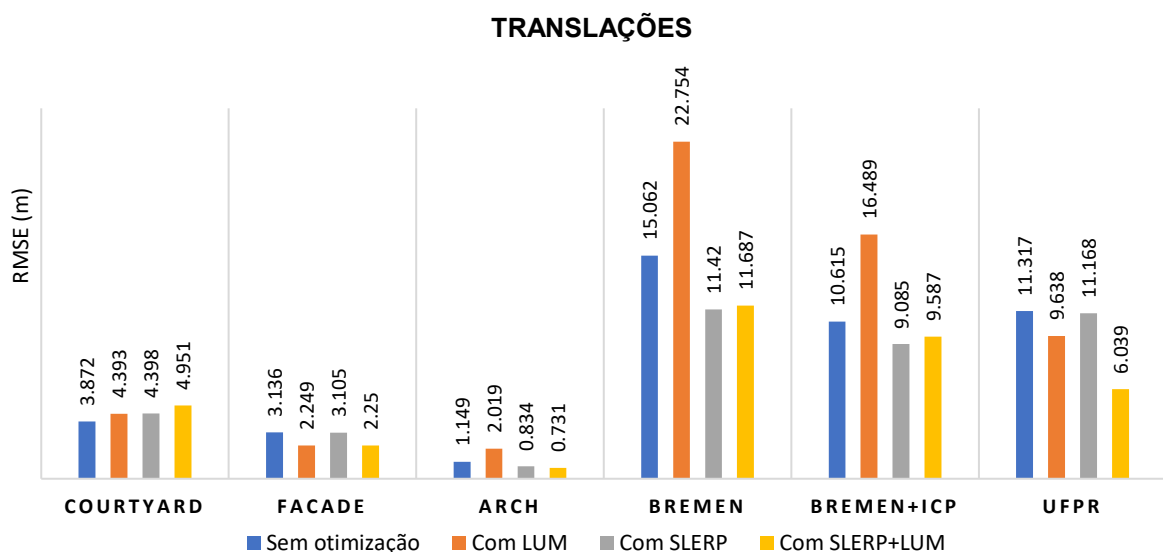


Conforme a Figura 42, no *dataset* UFPR, se os modelos forem aplicados isoladamente, o resultado é negativo no RMSE da translação das 5 primeiras poses. A aplicação do modelo completo, aumenta o erro nas três primeiras poses, mas diminui drasticamente nas últimas 4. Isto fica claro nas retas ajustadas ao RMSE sem otimização e após a etapa SLERP+LUM. Na Figura 43, com as rotações, o mesmo comportamento se repete, o RMSE das três primeiras rotações aumenta, mas o RMSE das quatro últimas diminuem. Esse efeito positivo nas últimas poses é bem maior que o efeito negativo nas primeiras, o que demonstra o comportamento nivelador do modelo de otimização por SLERP.

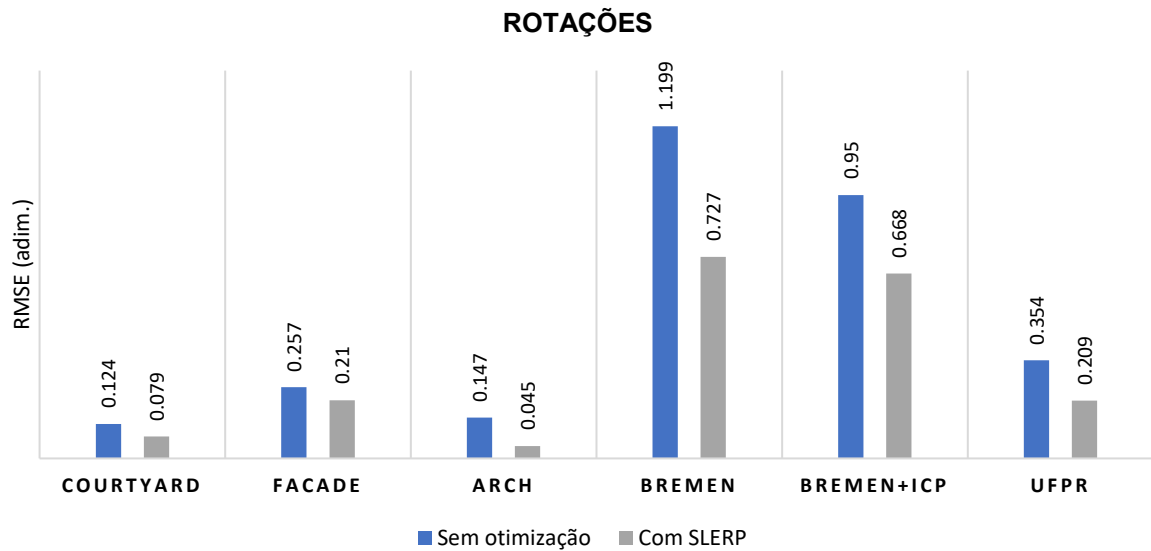
4.4.6 Resumo Quantitativo

A seguir, apresenta-se o erro total calculado em cada *dataset*, ou seja, o somatório do RMSE de todas as poses do *dataset*. Com exceção do *dataset Courtyard*, que contrariou o resultado esperado, todos os outros diminuíram RMSE em suas poses. Analisando cada *dataset* separadamente, em *Courtyard* houve aumento do erro nas translações, mas não nas rotações. Ainda assim, o impacto da otimização das rotações foi negativo nas translações, mesmo quando isoladamente. Não se tem uma hipótese provável para este erro, dado que não se observou erros grosseiros na análise qualitativa e quantitativa feita nos pares e em todo o circuito.

Figura 44 - Erro total na translação das poses cada *dataset*.



Fonte: o autor (2020)

Figura 45 - Erro total na rotação das poses de cada *dataset*.

Fonte: o autor (2020)

No *dataset Facade*, que contém um circuito nada convencional, ocorreram melhorias nas translações (Figura 44) e nas rotações (Figura 45) das poses. Praticamente toda a redução de erro nas translações adveio do modelo LUM. Se observa que as rotações também estão mais próximas do *groundtruth*. É provável que este *dataset* tenha se aproveitado mais do ajuste das translações porque os desníveis entre as estações do LST, no circuito, são elevados.

No *dataset Arch* os resultados são positivos, mas a dinâmica dos modelos se inverte nas translações. Desta vez foi a otimização SLERP que teve maior impacto. Por outro lado, o ajuste das translações contribuiu negativamente, aumentando o erro. O interessante neste *dataset* é que ele se beneficia mais da otimização SLERP+LUM, que é a proposta deste trabalho. O modelo LUM dobrou o erro total, mas quando combinado com a etapa SLERP, em vez de degradar o resultado, o melhorou.

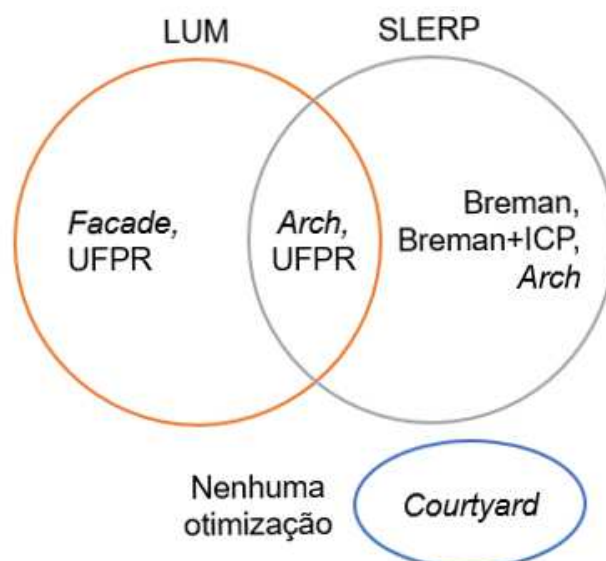
Ainda na Figura 44, no *dataset Breman*, o mesmo comportamento negativo se repetiu com LUM na otimização das translações. Apenas a otimização SLERP reduziu consideravelmente o erro. Desta vez, diferente do que ocorreu em *Arch*, a combinação dos modelos degradou o benefício obtido com a otimização das rotações. Se apenas o modelo SLERP fosse aplicado, os resultados teriam sido ligeiramente melhores que o modelo SLERP+LUM. No caso do circuito refinado pelo ICP, exatamente o mesmo comportamento se apresenta, o erro aumenta com a otimização LUM. Novamente, praticamente todo o impacto positivo ocorre na otimização SLERP, o resultado teria sido ligeiramente melhor se apenas as rotações fossem otimizadas.

No *dataset* UFPR tem-se o caso ideal, cada uma das etapas de otimização reduziu um pouco o erro na translação. A aplicação conjunta maximizou este benefício. Isoladamente, o modelo LUM teve maior impacto que o modelo SLERP, mas quando em conjunto, o modelo SLERP+LUM, maximizou a correção de 1,17 m obtida por LUM para 5,28 m com ambas as otimizações. Uma melhoria de quase 5 vezes em relação ao erro original, nas poses sem otimização.

De forma geral, o impacto do modelo SLERP foi mais positivo que o modelo LUM, para otimização das translações. Isto era esperado, pois rotações tem um impacto maior na composição de poses. Mesmo no *dataset Courtyard*, se olharmos apenas para a rotação (Figura 33), e ignorarmos os seus efeitos na translação (Figura 32), o resultado da técnica SLERP foi positivo. E isto se repete em todos os outros *datasets*.

Por outro lado, quando se observa o impacto das otimizações nas translações, apenas dois *datasets*, *Facade* e UFPR, se aproveitaram do modelo LUM isoladamente. Em Breman, Breman+ICP e *Arch*, os resultados foram negativos. Por fim, *Arch* e UFPR foram os únicos *datasets* que se beneficiaram da combinação SLERP+LUM. Esta informação encontra-se resumida no digrama de Venn da Figura 46. Foram feitas tentativas de correlacionar o resultado das otimizações com alguma característica presente no *dataset*, mas não se encontra nenhuma particularidade neles que justifique os resultados, além disso, a quantidade de *datasets* é muito pequena para este tipo de análise.

Figura 46 – Melhor modelo a ser aplicado, por *dataset*, segundo a maior melhoria obtida na translação das poses.



Fonte: o autor (2020)

5 CONCLUSÕES

Este trabalho investigou a otimização global do registro de nuvens de pontos 3D, obtidas por LST, por meio da proposta de um modelo linear de distribuição do erro de rotação entre poses. Cada rotação e translação, de cada pose, em cada circuito, de cada dataset, foi comparada com a respectiva pose *groundtruth*. Esta comparação considerou a aplicação individual e conjunta dos modelos SLERP e LUM. É nesta segunda configuração, SLERP+LUM, que acreditamos fornecer uma contribuição inédita através deste trabalho.

Assume-se uma confiança no modelo maior que 85%, dado que os resultados foram majoritariamente positivos em 41 poses, das 48 calculadas com 5 *datasets* e 6 circuitos. Isto significa que o erro acumulado, devido ao registro local dos pares, foi atenuado em 85% dos casos, garantindo uma melhor reconstrução de quase todos os *datasets*. Em muitos casos o erro também foi nivelado ao longo do circuito. Destes, apenas o circuito no *dataset Courtyard* apresentou resultados negativos, seja com a utilização de ambos os modelos, ou com o uso isolado.

Reduções do RMSE ocorreram nas poses das nuvens dos outros quatro *datasets*. As diferenças entre poses otimizadas e o *groundtruth*, em geral, diminuíram em relação às poses não otimizadas. Neste quesito, o modelo SLERP, de ajuste das rotações, teve maior impacto positivo que o modelo LUM, de ajuste das translações. Notou-se que o modelo SLERP tem um comportamento nivelador sobre as rotações das poses, distribuindo o erro ao longo do circuito. Outra observação é que a otimização das rotações sempre foi positiva, mas o seu impacto na translação, como demonstrou o *dataset Courtyard*, nem sempre é positivo. Por fim, a otimização LUM, se feita isoladamente, pode aumentar drasticamente o erro.

No registro local, onde se objetivou a reconstrução automática de todos os *datasets*, apenas dois deles, *Facade* e *Courtyard*, foram automaticamente reconstruídos. Neste quesito, o algoritmo FGR foi eficiente, pois o registro em pares teve qualidade similar ao ICP ponto-a-plano, conforme relatado pelos autores em (ZHOU et al., 2016). Se, por um lado, a automação do registro não foi possível, por outro, garantiu o uso de circuitos estimados por diferentes métodos de registro local. Isto possibilitou validar o modelo proposto sob diversas circunstâncias que poderiam ser questionadas, tais como, qual seria o resultado em circuitos registrados manualmente, automaticamente, ou em circuitos refinados com ICP. Os três tipos de

groundtruth, diferentes entre os cinco *datasets*, também contribuíram para a heterogeneidade dos testes. Isto põe mais confiança nos resultados, que sob todas estas circunstâncias, foi positivo em 85 % dos casos.

5.1 Recomendações de pesquisa

Muitas análises podem ser conduzidas futuramente com os modelos investigados neste trabalho. A mais direta delas consiste no estabelecimento de pesos para as translações otimizadas por MMQ no modelo LUM. A cada translação pode ser atribuído um valor de confiança, por exemplo, proporcional à sobreposição entregue pelo registro local do par, ou ainda, inversamente proporcional ao RMSE entre correspondências do par. Caso se descubra correlação entre a qualidade da translação estimada no registro local e a distância entre as estações no circuito, os pesos poderiam ser calculados com o inverso das distâncias, similarmente ao que ocorre no nivelamento geométrico. Em geral, inúmeras ponderações podem ser estabelecidas em cima das correlações que forem descobertas.

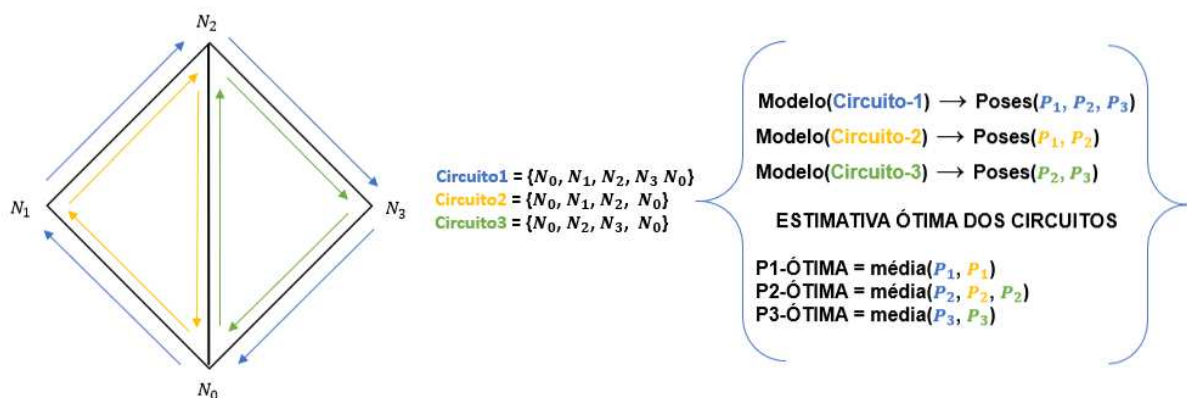
As estratégias anteriores, de pesos proporcionais ao RMSE ou à sobreposição do par, se estendem para a otimização das rotações por meio dos intervalos SLERP. Correlações poderiam ser investigadas entre a distância angular obtida no registro local e o RMSE em relação ao *groundtruth*. Não confundir este RMSE, que é das poses, com o RMSE do registro em pares. Por exemplo, se o registro local realizou uma grande rotação na nuvem de pesquisa, então o intervalo SLERP deste par seria levemente deslocado para o lado com maior erro no circuito, sendo o desvio proporcional à rotação que a nuvem de pesquisa sofreu. Analisando o RMSE das poses poderia ser inferido se esta mudança no intervalo foi positiva ou negativa.

Além da modificação de pesos e intervalos de interpolação, os modelos analisados neste trabalho podem ser expandidos para considerar múltiplos circuitos entre as estações/nuvens. Esta tarefa não é trivial, pois envolve a detecção de ciclos em grafos, um problema da computação não determinístico e não solucionável em tempo polinomial (NP-difícil). Todavia, estabelecidos os circuitos, em cada um deles é possível aplicar ambos os modelos. Theiler et al. (2015), provou que isto é possível.

Circuitos com n vértices retornam $(n - 1)$ poses, que são as estimativas ótimas daquele circuito. Então, a melhor estimativa entre vários circuitos, assumindo independência entre os erros de cada circuito, é simplesmente a média das poses em

comum. Um modelo que considere múltiplas sobreposições é de alto valor na literatura, portanto, daremos um exemplo simples. Suponha um grafo $g(v, e) = (4, 5)$ como na Figura 47. Se o vértice N_0 é a origem global, então este grafo possui três circuitos com início e fim nesta origem. Estes circuitos são apresentados com setas em cores diferentes. Aplicando o modelo deste trabalho em cada um deles, três conjuntos de poses são obtidos. A média destas poses será justamente a estimativa ótima dos circuitos.

Figura 47 - Estimativa ótima de poses em múltiplos circuitos de estações LST



Fonte: o autor (2021)

O grafo na Figura 47 é chamado *wheatstone bridge network* por (LU e MILIOS, 1997). Ele retorna a pose P_2 com a melhor estimativa, dado que ela é calculada por três circuitos diferentes. Uma demonstração similar a esse raciocínio, que utiliza o mesmo grafo, é dada em (Idem, 1997). A média entre poses, citada na Figura 47, reduz-se à média simples entre translações. Nas rotações, consiste em aplicar a técnica SLERP em $1/n$ a cada n rotação considerada. Para a estimativa ótima da rotação em P_2 , que equivale a uma média ponderada de 3 rotações, o cálculo é simplesmente:

$$R2_{\text{ótima}} = \text{Slerp}[\text{Slerp}(R2, R2, 1/2), R2, 1/3]$$

Nesta fórmula, a interpolação das duas primeiras rotações se dá em $1/2$, mas a interpolação do resultado desta com a terceira se dá em $1/3$. Isto é necessário para compensar o fato de que a terceira rotação não tem a mesma importância da média das duas primeiras. Desta forma, com base em (LU e MILIOS, 1997) e no que foi analisado neste trabalho, assume-se que o método proposto é perfeitamente

escalável para múltiplas sobreposições entre nuvens de pontos. A hipótese é de que os resultados do modelo, se assim aplicados, serão superiores ao aqui apresentado.

6 REFERÊNCIAS BIBLIOGRÁFICAS

Allied Market Research (2020). **Lidar Market Outlook – 2027**. Disponível em: <<https://www.alliedmarketresearch.com/lidar-market>>. Acesso em: 06/11/2020.

AIGER, D.; MITRA, N. J.; COHEN-OR, D. 4-points congruent sets for robust pairwise surface registration. **ACM SIGGRAPH 2008 papers**. p.1–10, 2008.

BABIN, P; GIGUERE, P; POMERLEAU, F. Analysis of robust functions for registration algorithms. In: **2019 International Conference on Robotics and Automation (ICRA)**. IEEE, 2019. p. 1451-1457.

BENJEMAA, R; SCHMITT, F. A solution for the registration of multiple 3D point sets using unit quaternions. In: **European Conference on Computer Vision**. Springer, Berlin, Heidelberg, 1998. p. 34-50.

BESL, P. J.; MCKAY, N. D. Method for registration of 3-D shapes. Sensor fusion IV: control paradigms and data structures. **Anais...** . v. 1611, p.586–606, 1992. International Society for Optics and Photonics.

BIBER, P.; STRASSER, W. The normal distributions transform: A new approach to laser scan matching. Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)(Cat. No. 03CH37453). **Anais...** . v. 3, p.2743–2748, 2003. IEEE.

BLACK, M. J.; RANGARAJAN, A. On the unification of line processes, outlier rejection, and robust statistics with applications in early vision. **International journal of computer vision**, v. 19, n. 1, p. 57–91, 1996. Springer.

BORRMANN, D.; ELSEBERG, J.; LINGEMANN, K.; NÜCHTER, A.; HERTZBERG, J. Globally consistent 3D mapping with scan matching. **Robotics and Autonomous Systems**, v. 56, n. 2, p. 130–142, 2008. Elsevier.

CHEN, Y.; MEDIONI, G. G. Object modeling by registration of multiple range images. **Image Vision Comput.**, v. 10, n. 3, p. 145–155, 1992.

CHOI, S.; ZHOU, Q.-Y.; KOLTUN, V. Robust reconstruction of indoor scenes. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. **Anais...** . p.5556–5565, 2015.

CONWAY, J. H.; SMITH, D. A. **On quaternions and octonions**. CRC Press, 2003.
 DAM, E. B.; KOCH, M.; LILLHOLM, M. **Quaternions, interpolation and animation**. Citeseer, 1998.

CONWAY J. H.; SMITH, D. **On quaternions and octonions their geometry, arithmetic and symmetry**. Livro (2003)

DÍEZ, Y.; ROURE, F.; LLADÓ, X.; SALVI, J. A qualitative review on 3D coarse registration methods. **ACM Computing Surveys (CSUR)**, v. 47, n. 3, p. 1–36, 2015. ACM New York, NY, USA.

DONG, Z.; YANG, B.; LIANG, F.; HUANG, R.; SCHERER, S. Hierarchical registration of unordered TLS point clouds based on binary shape context descriptor. **ISPRS Journal of Photogrammetry and Remote Sensing**, v. 144, p. 61–79, 2018. Elsevier B.V.

DONG, Z.; YANG, B.; LIU, Y.; et al. A novel binary shape context for 3D local surface description. **ISPRS JOURNAL OF PHOTOGRAMMETRY AND REMOTE SENSING**, v. 130, p. 431–452, 2017.

DOS SANTOS, D. R. et al. Mapping indoor spaces by adaptive coarse-to-fine registration of RGB-D data. **IEEE geoscience and remote sensing letters**, v. 13, n. 2, p. 262-266, 2016.

DAN, E. D.; KOCH, M.; LILLHOLM, M. Quaternions, interpolation and animation. **Technical Report DIKU-TR-98/5**, 1998. Department of Computer Science at the University of Copenhagen.

EVEN, S. **Graph algorithms** (livro). Cambridge University Press, 2011.

FARJAS, M.; ZANCAJO, J. J.; MOSTAZA, T. The Hidden World: Reality Through Laser Scanner Technologies—A Critical Approach to Documentation and Interpretation. **Laser Scanner Technology**, p. 151, 2012. BoD—Books on Demand.

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**, v. 24, n. 6, p. 381–395, 1981. ACM New York, NY, USA.

GALLIER, J. The Quaternions and the Spaces S^3 , $SU(2)$, $SO(3)$, and $\mathbb{R}P^3$. **Geometric Methods and Applications**. p.281–300, 2011. Springer.

FREY, C. B.; OSBORNE, M. A. The future of employment: How susceptible are jobs to computerisation?. **Technological forecasting and social change**, v. 114, p. 254-280, 2017.

GRISSETTI, G.; KUMMERLE, R.; STACHNISS, C.; BURGARD, W. A tutorial on graph-based SLAM. **IEEE Intelligent Transportation Systems Magazine**, v. 2, n. 4, p. 31–43, 2010. IEEE.

HAN, X.-F.; JIN, J. S.; WANG, M.-J.; et al. A review of algorithms for filtering the 3D point cloud. **Signal Processing: Image Communication**, v. 57, p. 103–112, 2017. Elsevier.

HARRIS, C. G.; STEPHENS, M. A combined corner and edge detector. Alvey vision conference. **Anais...** . v. 15, p.10–5244, 1988. Citeseer.

HARTLEY, R. I.; MUNDY, J. L. Relationship between photogrammetry and computer vision. In: **Integrating photogrammetric techniques with scene analysis and machine vision**. International Society for Optics and Photonics, 1993. p. 92-105.

HOLZ, D.; ICHIM, A.-E.; TOMBARI, F.; RUSU, R. B.; BEHNKE, S. Registration with the point cloud library: A modular framework for aligning in 3-D. **IEEE Robotics and Automation Magazine**, v. 22, n. 4, p. 110–124, 2015. Institute of Electrical and Electronics Engineers Inc.

HORN, B. K. P. Closed-form solution of absolute orientation using unit quaternions. **Josa a**, v. 4, n. 4, p. 629–642, 1987. Optical Society of America.

HUBER, D. F.; HEBERT, M. Fully automatic registration of multiple 3D data sets. **Image and Vision Computing**, v. 21, n. 7, p. 637–650, 2003. Elsevier.

HUYNH, D. Q. Metrics for 3D rotations: Comparison and analysis. **Journal of Mathematical Imaging and Vision**, v. 35, n. 2, p. 155–164, 2009. Springer.

KHOSHELHAM, K. Accuracy analysis of kinect depth data. In: **ISPRS workshop laser scanning**. 2011.

KÜMMERLE, R.; GRISSETTI, G.; STRASDAT, H.; KONOLIGE, K.; BURGARD, W. g 2 o: A general framework for graph optimization. 2011 IEEE International Conference on

Robotics and Automation. **Anais...** . p.3607–3613, 2011. IEEE.

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

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

MarketsAndMarkets (2020). **The LiDAR Market size is projected to reach USD 2.8 billion by 2025**. Disponível em: <<https://www.marketsandmarkets.com/Market-Reports/lidar-market-1261.html>>. Acesso em: 06/11/2020.

MAXVAL (2020). **Will LiDAR Scanners in Mobile Devices Be a Game Changer for Apple?**. Disponível em: <<https://www.maxval.com/blog/lidar-scanners-mobile-devices-game-changer-apple/>>. Acesso em: 16/11/2020.

MIAN, A.; BENNAMOUN, M.; OWENS, R. On the repeatability and quality of keypoints for local feature-based 3d object retrieval from cluttered scenes. **International Journal of Computer Vision**, v. 89, n. 2–3, p. 348–361, 2010. Springer.

MUJA, M.; LOWE, D. G. Fast approximate nearest neighbors with automatic algorithm configuration. **VISAPP (1)**, v. 2, n. 331–340, p. 2, 2009.

NEVES, R. C. **Os quatérnios de Hamilton e o Espaço**. Dissertação de Mestrado (Mestrado em ensino de matemática). 2008, UFRJ, Rio de Janeiro.

PAVAN, N. L.; DOS SANTOS, D. R. A global closed-form refinement for consistent TLS data registration. **IEEE Geoscience and Remote Sensing Letters**, v. 14, n. 7, p. 1131–1135, 2017. Institute of Electrical and Electronics Engineers Inc.

PAVAN, N. L.; DOS SANTOS, D. R.; KHOSHELHAM, K. Global Registration of Terrestrial Laser Scanner Point Clouds Using Plane-to-Plane Correspondences. **Remote Sensing**, v. 12, n. 7, p. 1127, 2020. Multidisciplinary Digital Publishing Institute.

Research and Markets (2020). **Smartphone Market - Growth, Trends, and Forecast (2020 - 2025)**. Disponível em: <<https://www.researchandmarkets.com/reports/4856120/smartphones-market-growth-trends-and-forecast>>. Acesso em: 1/11/2020.

RUSINKIEWICZ, S.; LEVOY, M. Efficient variants of the ICP algorithm. Proceedings Third International Conference on 3-D Digital Imaging and Modeling. **Anais...** . p.145–152, 2001. IEEE.

RUSU, R. B.; BLODOW, N.; BEETZ, M. Fast point feature histograms (FPFH) for 3D registration. 2009 IEEE international conference on robotics and automation. **Anais...** . p.3212–3217, 2009. IEEE.

RUSU, R. B.; BLODOW, N.; MARTON, Z. C.; BEETZ, M. Aligning point cloud views using persistent feature histograms. 2008 IEEE/RSJ international conference on intelligent robots and systems. **Anais...** . p.3384–3391, 2008. IEEE.

RUSU, R. B.; COUSINS, S. 3D is here: Point Cloud Library (PCL). 2011 IEEE INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION (ICRA). **Anais...** , IEEE International Conference on Robotics and Automation ICRA., 2011. 345 E 47TH ST, NEW YORK, NY 10017 USA: IEEE.

RUSU, R. B.; MARTON, Z. C.; BLODOW, N.; DOLHA, M.; BEETZ, M. Towards 3D point cloud based object maps for household environments. **Robotics and Autonomous Systems**, v. 56, n. 11, p. 927–941, 2008. Elsevier.

SCHOENKNECHT, E. **Medição de Distância com o uso de Laser**. UFRGS, Porto Alegre, 2010.

SCHULZ, A. Rotations and Interpolations. **Instituto Nacional de Matemática Pura e Aplicada IMPA**, 2010.

SHAN, J.; TOTH, C. K. **Topographic laser ranging and scanning: principles and processing**. CRC press, 2018.

SHOEMAKE, K. Animating rotation with quaternion curves. Proceedings of the 12th annual conference on Computer graphics and interactive techniques. **Anais...** . p.245–254, 1985.

SILVA A. M. **A representação das matrizes de rotações com o uso dos quatérnios: aplicações à fotogrametria**. Dissertação de mestrado (Ciências Geodésicas), Recife-PE, UFPE, 2014.

SLABAUGH, Gregory G. Computing Euler angles from a rotation matrix. **Retrieved on**

August, v. 6, n. 2000, p. 39-63, 1999.

SMITH, D. J.; VAMANAMURTHY, M. K. How small is a unit ball?. **Mathematics Magazine**, v. 62, n. 2, p. 101-107, 1989.

SOUDARISSANANE, S. S. **The geometry of terrestrial laser scanning; identification of errors, modeling and mitigation of scanning geometry**. 2016. 131 p. Tese (Doutorado em Sensoriamento Remoto). Delft University, Delft, 2016.

TASORA, A.; SERBAN, R.; MAZHAR, H.; et al. Chrono: An open source multi-physics dynamics engine. International Conference on High Performance Computing in Science and Engineering. **Anais...** . p.19–49, 2015. Springer.

THEILER, P. W.; WEGNER, J. D.; SCHINDLER, K. Keypoint-based 4-Points Congruent Sets - Automated marker-less registration of laser scans. **ISPRS Journal of Photogrammetry and Remote Sensing**, v. 96, p. 149–163, 2014. Elsevier B.V.

THEILER, P. W.; WEGNER, J. D.; SCHINDLER, K. Globally consistent registration of terrestrial laser scans via graph optimization. **ISPRS Journal of Photogrammetry and Remote Sensing**, v. 109, p. 126–138, 2015. Elsevier B.V.

VIRTANEN, P.; GOMMERS, R.; OLIPHANT, T. E.; et al. SciPy 1.0: fundamental algorithms for scientific computing in Python. **Nature methods**, v. 17, n. 3, p. 261–272, 2020. Nature Publishing Group.

VLAMINCK, M.; LUONG, H.; PHILIPS, W. Have i seen this place before? A fast and robust loop detection and correction method for 3D lidar SLAM. **Sensors (Switzerland)**, v. 19, n. 1, 2019. MDPI AG.

WEBER, T.; HÄNSCH, R.; HELLWICH, O. Automatic registration of unordered point clouds acquired by Kinect sensors using an overlap heuristic. **ISPRS Journal of Photogrammetry and Remote Sensing**, v. 102, p. 96–109, 2015. Elsevier B.V.

WEINMANN, M. **Reconstruction and analysis of 3D scenes**. Springer, 2016.

WILLIAMS, J.; BENNAMOUN, M. Simultaneous registration of multiple corresponding point sets. **Computer Vision and Image Understanding**, v. 81, n. 1, p. 117–142, 2001. Elsevier.

ZHANG, F. Quaternions and matrices of quaternions. **Linear algebra and its applications**, v. 251, p. 21–57, 1997. North-Holland.

ZHOU, Q.-Y.; PARK, J.; KOLTUN, V. Fast global registration. European Conference on Computer Vision. **Anais...** . p.766–782, 2016. Springer.

ZHOU, Q.-Y.; PARK, J.; KOLTUN, V. Open3D: A modern library for 3D data processing. **arXiv preprint arXiv:1801.09847**, 2018.

APENDICE A – TRANSFORMACOES DO CIRCUITO, LOOPCLOSURE E POSES DE COURTYARD

Neste apêndice, se entende por pose *loopclosure*, aquela que resulta da composição de todas as transformações em pares do circuito. Esta pose será tão mais próxima da identidade quanto menor forem os erros presentes entre as transformações em pares.

Transformações em pares do Circuito

T-10

0.9999689882 -0.0077786896 0.0012307176 13.7301474401
 0.0077749968 0.9999653417 0.0029773927 6.5454777836
 -0.0012538351 -0.0029677315 0.9999948102 2.6331268453
 0.0000000000 -0.0000000000 -0.0000000000 1.0000000000

T-21

0.9998155805 0.0190869580 0.0021196762 -9.0135356591
 -0.0190798819 0.9998124842 -0.0033097604 -15.8584423823
 -0.0021824519 0.0032687068 0.9999922762 4.2570956596
 -0.0000000000 -0.0000000000 -0.0000000000 1.0000000000

T-32

0.9999860641 -0.0047382392 -0.0023282206 2.0293370297
 0.0047315531 0.9999846907 -0.0028689191 -6.7355988715
 0.0023417785 0.0028578630 0.9999931743 -0.0476663303
 -0.0000000000 0.0000000000 -0.0000000000 1.0000000000

T-43

0.9998927626 -0.0144593996 -0.0023214145 -8.2334638300
 0.0144693296 0.9998859843 0.0043193602 -2.0568094852
 0.0022586945 -0.0043524863 0.9999879770 0.9301942050
 -0.0000000000 -0.0000000000 -0.0000000000 1.0000000000

T-54

0.9999065878 0.0032084064 0.0132861492 2.0953595359
 -0.0030765828 0.9999459588 -0.0099304594 -2.1929239002
 -0.0133172922 0.0098886558 0.9998624226 0.3694548733
 -0.0000000000 -0.0000000000 -0.0000000000 1.0000000000

T-65

0.9996237379 0.0263889119 -0.0074838467 -7.2792816947
 -0.0263970719 0.9996510427 -0.0009936614 4.7006306352
 0.0074550136 0.0011908392 0.9999715019 0.5150045825
 -0.0000000000 0.0000000000 -0.0000000000 1.0000000000

T-76

0.9965562603 -0.0828511023 -0.0033637692 3.0669479405

0.0828614068 0.9965564163 0.0030489998 16.9616873498
 0.0030995728 -0.0033172265 0.9999896943 0.3677147316
 -0.0000000000 -0.0000000000 -0.0000000000 1.0000000000
 T-07
 0.9986448526 0.0504865787 0.0126318512 2.7903217906
 -0.0505141620 0.9987215866 0.0018739804 -1.7320206187
 -0.0125210916 -0.0025095283 0.9999184589 -9.3443289774
 0.0000000000 -0.0000000000 -0.0000000000 1.0000000000
 T-LoopClosure
 0.9998422806 -0.0107629728 0.0141269964 -0.3212272980
 0.0108327832 0.9999294426 -0.0048744481 -0.2570038528
 -0.0140735360 0.0050267140 0.9998883275 0.0069261265
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

Poses do *Groundtruth* (G)

Pose-1

9.99949252e-001 -1.00655392e-002 -4.20963699e-004 1.41084377e+001
 1.00656771e-002 9.99949287e-001 3.26676452e-004 6.32330778e+000
 4.17654176e-004 -3.30897160e-004 9.99999858e-001 2.56439278e+000
 0.00000000e+000 0.00000000e+000 0.00000000e+000 1.00000000e+000

Pose-2

9.99936141e-001 1.13006712e-002 9.12331474e-005 5.22159374e+000
 -1.13006945e-002 9.99936112e-001 2.58986792e-004 -9.65604650e+000
 -8.83005945e-005 -2.60001252e-004 9.99999963e-001 6.85974164e+000
 0.00000000e+000 0.00000000e+000 0.00000000e+000 1.00000000e+000

Pose-3

9.99950170e-001 9.97996366e-003 -2.38957530e-004 7.24515074e+000
 -9.97985686e-003 9.99950101e-001 4.44026798e-004 -1.66540211e+001
 2.43376979e-004 -4.41619911e-004 9.99999873e-001 6.78143256e+000
 0.00000000e+000 0.00000000e+000 0.00000000e+000 1.00000000e+000

Pose-4

9.99978388e-001 -6.55634790e-003 -4.88518967e-004 -9.37203450e-001
 6.55659670e-003 9.99978375e-001 5.09450016e-004 -1.87782809e+001
 4.85168271e-004 -5.12642026e-004 9.99999751e-001 7.56275778e+000
 0.00000000e+000 0.00000000e+000 0.00000000e+000 1.00000000e+000

Pose-5

9.99971229e-001 -7.56884639e-003 -5.03380544e-004 1.12801868e+000
 7.56906993e-003 9.99971255e-001 4.43667687e-004 -2.10544329e+001
 5.00008023e-004 -4.47465044e-004 9.99999775e-001 8.11941176e+000
 0.00000000e+000 0.00000000e+000 0.00000000e+000 1.00000000e+000

Pose-6

9.99775112e-001 2.12058201e-002 1.96597734e-004 -6.74314344e+000
 -2.12058533e-002 9.99775116e-001 1.68462233e-004 -1.62114576e+001

-1.92981143e-004 -1.72593371e-004 9.99999967e-001 9.00624074e+000
0.00000000e+000 0.00000000e+000 0.00000000e+000 1.00000000e+000
Pose-7
9.98625179e-001 -5.24171340e-002 -4.40889562e-004 -3.45786056e+000
5.24172306e-002 9.98625249e-001 2.10659888e-004 1.39006780e+000
4.29241261e-004 -2.33480478e-004 9.99999881e-001 9.53832322e+000
0.00000000e+000 0.00000000e+000 0.00000000e+000 1.00000000e+000

Poses O obtidas sem nenhuma otimização (O = Originais)

Pose-1
0.9999689882 -0.0077786896 0.0012307176 13.7301474401
0.0077749968 0.9999653417 0.0029773927 6.5454777836
-0.0012538351 -0.0029677315 0.9999948102 2.6331268453
0.0000000000 0.0000000000 0.0000000000 1.0000000000
Pose-2
0.9999303177 0.0113027508 0.0034069852 4.8454884903
-0.0113016014 0.9999360714 -0.0003563908 -9.3698201374
-0.0034107954 0.0003178616 0.9999941327 6.9485654978
0.0000000000 0.0000000000 0.0000000000 1.0000000000
Pose-3
0.9999778736 0.0065639169 0.0010804194 6.7983909173
-0.0065604197 0.9999733307 -0.0032091673 -16.1279061817
-0.0011014553 0.0032020082 0.9999942669 6.8918368058
0.0000000000 0.0000000000 0.0000000000 1.0000000000
Pose-4
0.9999680552 -0.0079032341 -0.0011946950 -1.4473864623
0.0079045801 0.9999681241 0.0011261670 -18.1336309834
0.0011857566 -0.0011355746 0.9999986522 7.8245085492
0.0000000000 0.0000000000 0.0000000000 1.0000000000
Pose-5
0.9999157612 -0.0047092792 0.0120951611 0.6647959429
0.0048158933 0.9999496764 -0.0088006643 -20.3095059770
-0.0120531076 0.0088581719 0.9998881213 8.1989377397
0.0000000000 0.0000000000 0.0000000000 1.0000000000
Pose-6
0.9997568206 0.0216137835 0.0043753608 -6.6297800726
-0.0215686588 0.9997162458 -0.0101104203 -15.6487005211
-0.0045926436 0.0100135910 0.9999393158 8.8432616639
0.0000000000 0.0000000000 0.0000000000 1.0000000000
Pose-7
0.9981163541 -0.0613219251 0.0018343876 -3.1953628281
0.0613328684 0.9980951192 -0.0066642409 1.2383061747
-0.0014222291 0.0067641960 0.9999761111 9.3667160817

0.0000000000 0.0000000000 0.0000000000 1.0000000000

Poses S obtidas com otimização das rotações (S = SLERP)

Pose-1

0.9999791892 -0.0064291276 -0.0005362670 13.7301474401
 0.0064310091 0.9999728992 0.0035837115 6.5454777836
 0.0005132123 -0.0035870857 0.9999934347 2.6331268453
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-2

0.9999019885 0.0140001065 -0.0001023329 4.8164723707
 -0.0140000059 0.9999015665 0.0009258239 -9.3552447489
 0.0001152845 -0.0009243005 0.9999995662 6.9424542902
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-3

0.9999348637 0.0106160779 -0.0041913223 6.7513162782
 -0.0106216408 0.9999427343 -0.0013072360 -16.1186354727
 0.0041772046 0.0013516696 0.9999903619 6.9012476488
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-4

0.9999627967 -0.0024924379 -0.0082579007 -1.5073452481
 0.0025219458 0.9999904659 0.0035648076 -18.0890902612
 0.0082489369 -0.0035855009 0.9999595488 7.7942598987
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-5

0.9999925953 0.0020157574 0.0032781344 0.5903511386
 -0.0019969880 0.9999816531 -0.0057188696 -20.2753918351
 -0.0032896021 0.0057122809 0.9999782739 8.1888470464
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-6

0.9995398442 0.0297131367 -0.0061015937 -6.6777130695
 -0.0297505020 0.9995385759 -0.0061272235 -15.5632560476
 0.0059167193 0.0063059295 0.9999626131 8.7546377028
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-7

0.9985962443 -0.0518693061 -0.0107292087 -3.1104351148
 0.0518395058 0.9986508098 -0.0030373915 1.2971084601
 0.0108722804 0.0024769308 0.9999378272 9.2474441621
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

Poses L obtidas com otimização das translações (L = LU & MILIUS, 1997)

Pose-1

0.9999689882 -0.0077786896 0.0012307176 13.7703008524
 0.0077749968 0.9999653417 0.0029773927 6.5776032652
 -0.0012538351 -0.0029677315 0.9999948102 2.6322610795
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-2

0.9999303177 0.0113027508 0.0034069852 4.9257953149
 -0.0113016014 0.9999360714 -0.0003563908 -9.3055691742
 -0.0034107954 0.0003178616 0.9999941327 6.9468339662
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-3

0.9999778736 0.0065639169 0.0010804194 6.9188511540
 -0.0065604197 0.9999733307 -0.0032091673 -16.0315297369
 -0.0011014553 0.0032020082 0.9999942669 6.8892395083
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-4

0.9999680552 -0.0079032341 -0.0011946950 -1.2867728133
 0.0079045801 0.9999681241 0.0011261670 -18.0051290570
 0.0011857566 -0.0011355746 0.9999986522 7.8210454859
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-5

0.9999157612 -0.0047092792 0.0120951611 0.8655630042
 0.0048158933 0.9999496764 -0.0088006643 -20.1488785690
 -0.0120531076 0.0088581719 0.9998881213 8.1946089106
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-6

0.9997568206 0.0216137835 0.0043753608 -6.3888595991
 -0.0215686588 0.9997162458 -0.0101104203 -15.4559476315
 -0.0045926436 0.0100135910 0.9999393158 8.8380670691
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-7

0.9981163541 -0.0613219251 0.0018343876 -2.9142889423
 0.0613328684 0.9980951192 -0.0066642409 1.4631845459
 -0.0014222291 0.0067641960 0.9999761111 9.3606557210
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

Poses SL obtidas com otimização das translações e rotações (SL = SLERP+L)

Pose-1

0.9999791892 -0.0064291276 -0.0005362670 13.7468892264
 0.0064310091 0.9999728992 0.0035837115 6.5779207893
 0.0005132123 -0.0035870857 0.9999934347 2.6419089437
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-2

0.9999019885 0.0140001065 -0.0001023329 4.8499559433
 -0.0140000059 0.9999015665 0.0009258239 -9.2903587375

0.0001152845 -0.0009243005 0.9999995662 6.9600184869
0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-3

0.9999348637 0.0106160779 -0.0041913223 6.8015416371
-0.0106216408 0.9999427343 -0.0013072360 -16.0213064556
0.0041772046 0.0013516696 0.9999903619 6.9275939439
0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-4

0.9999627967 -0.0024924379 -0.0082579007 -1.4403781029
0.0025219458 0.9999904659 0.0035648076 -17.9593182384
0.0082489369 -0.0035855009 0.9999595488 7.8293882922
0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-5

0.9999925953 0.0020157574 0.0032781344 0.6740600702
-0.0019969880 0.9999816531 -0.0057188696 -20.1131768066
-0.0032896021 0.0057122809 0.9999782739 8.2327575383
0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-6

0.9995398442 0.0297131367 -0.0061015937 -6.5772623516
-0.0297505020 0.9995385759 -0.0061272235 -15.3685980134
0.0059167193 0.0063059295 0.9999626131 8.8073302930
0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-7

0.9985962443 -0.0518693061 -0.0107292087 -2.9932426106
0.0518395058 0.9986508098 -0.0030373915 1.5242095000
0.0108722804 0.0024769308 0.9999378272 9.3089188507
0.0000000000 0.0000000000 0.0000000000 1.0000000000

APENDICE B – TRANSFORMACOES DO CIRCUITO, LOOPCLOSURE E POSES DE FACADE

Transformações em pares do Circuito

T-10

0.9917988819 0.1274740342 -0.0092384204 5.8040641888
 -0.1276406893 0.9916081011 -0.0205238477 -2.2096017048
 0.0065446348 0.0215347275 0.9997466795 -1.6070391771
 -0.0000000000 0.0000000000 -0.0000000000 1.0000000000

T-21

0.9993832240 0.0345375392 0.0063505874 -5.0700339207
 -0.0345290751 0.9994026596 -0.0014376783 5.4834012133
 -0.0063964478 0.0012175116 0.9999788013 -0.4445454338
 -0.0000000000 -0.0000000000 -0.0000000000 1.0000000000

T-32

0.9951534208 -0.0967981052 0.0173146150 -0.0583827729
 0.0966014074 0.9952524962 0.0118590426 -3.5089805215
 -0.0183803466 -0.0101289507 0.9997797594 1.8307232412
 0.0000000000 -0.0000000000 -0.0000000000 1.0000000000

T-43

0.9952627989 0.0966952897 -0.0100986219 1.8720656201
 -0.0967245236 0.9953081839 -0.0024465618 1.3194050657
 0.0098146700 0.0034117564 0.9999460146 1.1087342690
 -0.0000000000 0.0000000000 -0.0000000000 1.0000000000

T-54

0.9708527334 -0.2396697542 -0.0018381884 -0.1508673394
 0.2396752932 0.9708452879 0.0038962601 -2.8013412924
 0.0008507808 -0.0042232631 0.9999907201 -0.0325432338
 -0.0000000000 -0.0000000000 -0.0000000000 1.0000000000

T-65

0.9846608869 -0.1744790468 -0.0000105154 -1.9708949419
 0.1744746283 0.9846355183 0.0071763598 3.2554014430
 -0.0012417706 -0.0070681155 0.9999742495 -0.8779831719
 -0.0000000000 0.0000000000 -0.0000000000 1.0000000000

T-06

0.9697586547 0.2439373758 0.0079188596 -0.1543891202
 -0.2438711743 0.9697706024 -0.0084752065 -1.9823637512
 -0.0097468969 0.0062877233 0.9999327290 0.1353907849
 -0.0000000000 -0.0000000000 -0.0000000000 1.0000000000

T-LoopClosure

0.9998517772 -0.0082798223 0.0150953032 0.3394287689
 0.0084379677 0.9999098677 -0.0104430490 -0.0260561950
 -0.0150074761 0.0105688748 0.9998315231 0.1027148142

0.0000000000 0.0000000000 0.0000000000 1.0000000000

Poses do *Groundtruth* (G)

Pose-1

9.88035971e-001 1.54222655e-001 5.41374166e-004 5.34167810e+000
 -1.54223348e-001 9.88034647e-001 1.64157616e-003 -2.56153095e+000
 -2.81728199e-004 -1.70542883e-003 9.99998506e-001 -1.46640894e+000
 0.00000000e+000 0.00000000e+000 0.00000000e+000 1.00000000e+000

Pose-2

9.80605051e-001 1.95993656e-001 4.70903652e-004 1.04304568e+000
 -1.95993837e-001 9.80605055e-001 3.75638389e-004 4.07405159e+000
 -3.88147761e-004 -4.60647115e-004 9.9999819e-001 -1.94609556e+000
 0.00000000e+000 0.00000000e+000 0.00000000e+000 1.00000000e+000

Pose-3

9.96681420e-001 8.13831140e-002 1.71362402e-003 4.91583521e-001
 -8.13854036e-002 9.96681843e-001 1.31156586e-003 3.88824961e-001
 -1.60119863e-003 -1.44667730e-003 9.99997672e-001 -3.35355926e-001
 0.00000000e+000 0.00000000e+000 0.00000000e+000 1.00000000e+000

Pose-4

9.82209556e-001 1.87788121e-001 9.50915135e-005 2.29462248e+000
 -1.87788135e-001 9.82209218e-001 8.17323244e-004 1.51587036e+000
 6.00838351e-005 -8.20639759e-004 9.99999661e-001 9.35602846e-001
 0.00000000e+000 0.00000000e+000 0.00000000e+000 1.00000000e+000

Pose-5

9.97605046e-001 -6.91592370e-002 1.08270146e-003 1.79245353e+000
 6.91585782e-002 9.97605477e-001 6.34633264e-004 -1.30765863e+000
 -1.12399966e-003 -5.58235253e-004 9.99999212e-001 7.24916111e-001
 0.00000000e+000 0.00000000e+000 0.00000000e+000 1.00000000e+000

Pose-6

9.75868011e-001 -2.18361121e-001 2.16189487e-004 -5.92242377e-001
 2.18360648e-001 9.75867222e-001 1.33850835e-003 2.31991733e+000
 -5.03250419e-004 -1.25900020e-003 9.99999081e-001 -8.31434416e-002
 0.00000000e+000 0.00000000e+000 0.00000000e+000 1.00000000e+000

Poses O calculadas sem nenhuma otimização (O = Originais)

Pose-1

0.9917988819 0.1274740342 -0.0092384204 5.8040641888
 -0.1276406893 0.9916081011 -0.0205238477 -2.2096017048
 0.0065446348 0.0215347275 0.9997466795 -1.6070391771
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-2

0.9868203312 0.1617798731 -0.0035925869 1.4787083866
 -0.1618197515 0.9865832530 -0.0216299079 3.8840497670
 0.0000451023 0.0219261844 0.9997595913 -1.9665699680
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-3

0.9977022545 0.0658760481 0.0158290114 0.8468356236
 -0.0657227439 0.9977876322 -0.0100180772 0.3919974601
 -0.0164539431 0.0089547321 0.9998245249 -0.2132280357
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-4

0.9867870207 0.1619549137 0.0046884755 2.8190670724
 -0.1618764046 0.9867124585 -0.0139482603 1.5743388418
 -0.0068851665 0.0130050087 0.9998917263 0.8763237357
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-5

0.9968344106 -0.0792746674 0.0060568060 2.2163495745
 0.0793246975 0.9968124031 -0.0085220528 -1.1649037279
 -0.0053619165 0.0089755298 0.9999453434 0.8083913045
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-6

0.9677034136 -0.2519816366 0.0074403048 -0.0116949636
 0.2519897488 0.9677298907 -0.0001583893 1.9312623815
 -0.0071602942 0.0020281545 0.9999723080 -0.0297571531
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

Poses S obtidas com otimização das rotações (S = SLERP)

Pose-1

0.9916239829 0.1286736682 -0.0111787099 5.8040641888
 -0.1288670401 0.9914844205 -0.0187597917 -2.2096017048
 0.0086696255 0.0200432266 0.9997615249 -1.6070391771
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-2

0.9864075756 0.1641527449 -0.0073465070 1.4870357515
 -0.1642608853 0.9862532777 -0.0179675763 3.8888050143
 0.0042960896 0.0189300971 0.9998115798 -1.9855288401
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-3

0.9975422821 0.0693927061 0.0096978272 0.8399883364
 -0.0693458256 0.9975796955 -0.0050899503 0.4047578202
 -0.0100275609 0.0044049368 0.9999400205 -0.2218267038
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-4

0.9860102204 0.1666611891 -0.0028095193 2.8097623486

-0.1666765227 0.9859894528 -0.0066133076 1.5855061856
 0.0016679747 0.0069890698 0.9999741850 0.8738807077
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-5

0.9972944174 -0.0733241282 -0.0052361576 2.1942221699
 0.0733152944 0.9973070784 -0.0018598075 -1.1512255204
 0.0053584258 0.0014708852 0.9999845618 0.8215079012
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-6

0.9694850453 -0.2450418956 -0.0072949565 -0.0054425673
 0.2450827947 0.9694873460 0.0053581356 1.9525455187
 0.0057594003 -0.0069825006 0.9999590362 -0.0622342886
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

Poses L obtidas com otimização das translações (L = LU & MILIUS, 1997)

Pose-1

0.9917988819 0.1274740342 -0.0092384204 5.7555743647
 -0.1276406893 0.9916081011 -0.0205238477 -2.2058793913
 0.0065446348 0.0215347275 0.9997466795 -1.6217127220
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-2

0.9868203312 0.1617798731 -0.0035925869 1.3817287383
 -0.1618197515 0.9865832530 -0.0216299079 3.8914943942
 0.0000451023 0.0219261844 0.9997595913 -1.9959170577
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-3

0.9977022545 0.0658760481 0.0158290114 0.7013661512
 -0.0657227439 0.9977876322 -0.0100180772 0.4031644008
 -0.0164539431 0.0089547321 0.9998245249 -0.2572486704
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-4

0.9867870207 0.1619549137 0.0046884755 2.6251077758
 -0.1618764046 0.9867124585 -0.0139482603 1.5892280961
 -0.0068851665 0.0130050087 0.9998917263 0.8176295562
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-5

0.9968344106 -0.0792746674 0.0060568060 1.9739004539
 0.0793246975 0.9968124031 -0.0085220528 -1.1462921600
 -0.0053619165 0.0089755298 0.9999453434 0.7350235801
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-6

0.9677034136 -0.2519816366 0.0074403048 -0.3026339084
 0.2519897488 0.9677298907 -0.0001583893 1.9535962629

-0.0071602942 0.0020281545 0.9999723080 -0.1177984224
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

Poses SL obtidas com otimização das translações e rotações (SL = SLERP+L)

Pose-1

0.9916239829 0.1286736682 -0.0111787099 5.7569707616
 -0.1288670401 0.9914844205 -0.0187597917 -2.2086811722
 0.0086696255 0.0200432266 0.9997615249 -1.6193396939
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-2

0.9864075756 0.1641527449 -0.0073465070 1.3928488970
 -0.1642608853 0.9862532777 -0.0179675763 3.8906460795
 0.0042960896 0.0189300971 0.9998115798 -2.0101298736
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-3

0.9975422821 0.0693927061 0.0096978272 0.6987080546
 -0.0693458256 0.9975796955 -0.0050899503 0.4075194180
 -0.0100275609 0.0044049368 0.9999400205 -0.2587282541
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-4

0.9860102204 0.1666611891 -0.0028095193 2.6213886395
 -0.1666765227 0.9859894528 -0.0066133076 1.5891883160
 0.0016679747 0.0069890698 0.9999741850 0.8246786406
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-5

0.9972944174 -0.0733241282 -0.0052361576 1.9587550335
 0.0733152944 0.9973070784 -0.0018598075 -1.1466228574
 0.0053584258 0.0014708852 0.9999845618 0.7600053174
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-6

0.9694850453 -0.2450418956 -0.0072949565 -0.2880031309
 0.2450827947 0.9694873460 0.0053581356 1.9580687143
 0.0057594003 -0.0069825006 0.9999590362 -0.1360373893
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

APENDICE C – TRANSFORMACOES DO CIRCUITO, LOOPCLOSURE E POSES DE ARCH

Transformações em pares do Circuito

T-10

0.9328766971 -0.3601858955 -0.0026811895 6.5358480601
 0.3601551588 0.9328573234 -0.0080917051 12.5946185553
 0.0054156853 0.0065829189 0.9999636671 -0.8651592526
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

T-21

-0.7510797564 0.6602093803 0.0016654579 -7.2005254260
 -0.6601827352 -0.7510682045 0.0074369598 11.4224821496
 0.0061608231 0.0044862434 0.9999709585 1.2346205463
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

T-32

0.6312555121 -0.7752602921 -0.0220897705 -7.2043873812
 0.7754536130 0.6314044711 0.0002966456 8.3769388693
 0.0137176023 -0.0173168515 0.9997559472 -0.3566657179
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

T-43

0.1080818793 -0.9941289109 -0.0051005681 14.5362785986
 0.9941278275 0.1081063079 -0.0047842301 11.0808369193
 0.0053075450 -0.0045535281 0.9999755474 0.3444111191
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

T-04

0.9545738393 0.2973663323 -0.0190276055 -25.3853128510
 -0.2976128318 0.9546145939 -0.0117294282 -6.8450989410
 0.0146760929 0.0168594649 0.9997501542 -0.8778380287
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

T-LoopClosure

0.9988813179 0.0030735817 -0.0471875610 0.1256425220
 -0.0051237482 0.9990451627 -0.0433879043 0.2373634191
 0.0470091483 0.0435811442 0.9979432969 -0.7992743928
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

Poses do Groundtruth (G)

Pose-1

0.9317580000 -0.3630790000 -0.0000720000 6.4963680000
 0.3630790000 0.9317580000 -0.0005320000 12.4993460000
 0.0002600000 0.0004700000 1.0000000000 -0.8025260000
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-2

-0.4604560000 0.8876820000 -0.0002590000 -4.3808110000
 -0.8876820000 -0.4604560000 -0.0008750000 20.5297810000
 -0.0008960000 -0.0001730000 1.0000000000 0.5158960000
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-3

0.3979520000 0.9174060000 0.0001980000 6.5419220000
 -0.9174050000 0.3979520000 -0.0013230000 23.3298790000
 -0.0012920000 0.0003450000 0.9999990000 0.2282580000
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-4

0.9546000000 -0.2978910000 -0.0006270000 22.4311220000
 0.2978910000 0.9546000000 -0.0002040000 14.3294860000
 0.0006590000 0.0000080000 1.0000000000 0.4651040000
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

Poses O calculadas sem nenhuma otimização (O = Originais)

Pose-1

0.9328766971 -0.3601858955 -0.0026811895 6.5358480601
 0.3601551588 0.9328573234 -0.0080917051 12.5946185553
 0.0054156853 0.0065829189 0.9999636671 -0.8651592526
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-2

-0.4628779686 0.8864204536 -0.0016630351 -4.2988815302
 -0.8863299017 -0.4628020084 0.0152841870 20.6468681163
 0.0127785600 0.0085487111 0.9998818070 0.4056139303
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-3

0.3946598339 0.9181609785 -0.0349861830 6.4619537659
 -0.9185692653 0.3951652220 0.0086575080 23.1500165977
 0.0217743088 0.0287204617 0.9993502962 0.0285407018
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-4

0.9557207784 -0.2937550984 -0.0174853057 22.3607815017
 0.2929350182 0.9553518267 -0.0386259286 14.1791809692
 0.0280511821 0.0317935442 0.9991007465 1.0074922271
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

Poses S obtidas com otimização das rotações (S = SLERP)

Pose-1

0.9325795439 -0.3609522903 0.0029729379 6.5358480601
 0.3609422748 0.9325818719 0.0034244057 12.5946185553

-0.0040085550 -0.0021204717 0.9999897175 -0.8651592526
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-2

-0.4614550517 0.8871491619 0.0050596167 -4.2985153006
 -0.8871428008 -0.4613976910 -0.0094774297 20.6522721543
 -0.0060733984 -0.0088620103 0.9999422876 0.3740912504
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-3

0.3974111265 0.9176406504 0.0001826456 6.4557753568
 -0.9176179325 0.3974026732 -0.0069602662 23.1818725760
 -0.0064596070 0.0025984883 0.9999757604 -0.0130352874
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-4

0.9549257251 -0.2967298239 0.0082626312 22.4009435116
 0.2966896751 0.9549566343 0.0057500779 14.2442796819
 -0.0095966740 -0.0030394599 0.9999493315 0.2662622613
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

Poses L obtidas com otimização das translações (L = LU & MILIUS, 1997)

Pose-1

0.9328766971 -0.3601858955 -0.0026811895 6.5107195557
 0.3601551588 0.9328573234 -0.0080917051 12.5471458715
 0.0054156853 0.0065829189 0.9999636671 -0.7053043740
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-2

-0.4628779686 0.8864204536 -0.0016630351 -4.3491385390
 -0.8863299017 -0.4628020084 0.0152841870 20.5519227487
 0.0127785600 0.0085487111 0.9998818070 0.7253236874
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-3

0.3946598339 0.9181609785 -0.0349861830 6.3865682527
 -0.9185692653 0.3951652220 0.0086575080 23.0075985462
 0.0217743088 0.0287204617 0.9993502962 0.5081053375
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-4

0.9557207784 -0.2937550984 -0.0174853057 22.2602674841
 0.2929350182 0.9553518267 -0.0386259286 13.9892902339
 0.0280511821 0.0317935442 0.9991007465 1.6469117413
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

Poses SL obtidas com otimização das translações e rotações (SL = SLERP+L)

Pose-1

0.9325795439 -0.3609522903 0.0029729379 6.4990986639
0.3609422748 0.9325818719 0.0034244057 12.5604387199
-0.0040085550 -0.0021204717 0.9999897175 -0.7957369902
0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-2

-0.4614550517 0.8871491619 0.0050596167 -4.3720140929
-0.8871428008 -0.4613976910 -0.0094774297 20.5839124834
-0.0060733984 -0.0088620103 0.9999422876 0.5129357751
0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-3

0.3974111265 0.9176406504 0.0001826456 6.3455271683
-0.9176179325 0.3974026732 -0.0069602662 23.0793330697
-0.0064596070 0.0025984883 0.9999757604 0.1952314997
0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-4

0.9549257251 -0.2967298239 0.0082626312 22.2539459270
0.2966896751 0.9549566343 0.0057500779 14.1075603401
-0.0095966740 -0.0030394599 0.9999493315 0.5439513107
0.0000000000 0.0000000000 0.0000000000 1.0000000000

APENDICE D – TRANSFORMACOES DO CIRCUITO, LOOPCLOSURE E POSES DE BREMEN

Transformações em pares do Circuito

T-10

-0.9213735900 0.3884060700 -0.0145408600 -1.6707991300
 -0.3886781600 -0.9207240300 0.0345910700 41.1916831800
 0.0000472600 0.0375230100 0.9992957600 -0.1519919700
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

T-21

0.7288861700 -0.6843948199 0.0181295810 -23.6093203784
 0.6844420557 0.7290538073 0.0044292593 -23.8073477737
 -0.0162488021 0.0091802218 0.9998258348 0.7168828538
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

T-32

0.9106173993 0.4130161841 -0.0139134347 -9.6086160440
 -0.4131956737 0.9105287270 -0.0143795891 -35.9228948627
 0.0067295790 0.0188432750 0.9997998018 0.1272293945
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

T-43

-0.9302171188 -0.3669841369 0.0043307190 36.5296708267
 0.3665708499 -0.9284632824 0.0598476841 0.8599493793
 -0.0179422372 0.0572588556 0.9981981264 0.4316488308
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

T-54

0.2971512039 -0.9547649406 0.0111834821 -22.1457963457
 0.9547523176 0.2972563321 0.0093104779 0.1317306302
 -0.0122136787 0.0079108358 0.9998941168 0.1587002812
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

T-65

-0.6747070288 0.7380301260 0.0090530865 -21.4313777391
 -0.7377345165 -0.6739570174 -0.0391116583 5.9296120237
 -0.0227641909 -0.0330676851 0.9991938350 0.2327961546
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

T-76

0.1981524925 0.9797064988 -0.0301788989 5.4672240806
 -0.9800490485 0.1975477824 -0.0218800389 -20.2746012130
 -0.0154742418 0.0339123854 0.9993050075 -0.3228641143
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

T-87

0.6198229238 0.7842700620 0.0272031809 25.1031175441
 -0.7841691888 0.6203223860 -0.0166979278 -7.8892841631
 -0.0299704269 -0.0109821379 0.9994904533 -0.7284164335
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

T-98

-0.5641875832 0.8255794601 -0.0105321453 -8.3980470349
 -0.8246870646 -0.5628712007 0.0553828211 26.9403776212
 0.0397946783 0.0399320240 0.9984096439 -0.6718899619
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

T-10_9

-0.8961162143 -0.4422963792 0.0367374938 -11.8694452382
 0.4437682300 -0.8941957032 0.0590237446 -22.6200436515
 0.0067445206 0.0691950672 0.9975803497 1.2687897528
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

T-11_10

0.1893017172 0.9819189653 -0.0000736448 -12.3617868941
 -0.9814945441 0.1892220980 0.0293846494 20.8445550879
 0.0288672798 -0.0054902826 0.9995681752 -0.9338117308
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

T-12_11

-0.7676193094 0.6409017427 -0.0023562898 5.2634013219
 -0.6408450709 -0.7675917824 -0.0109750023 -37.7429597120
 -0.0088425668 -0.0069146070 0.9999369966 0.0483731477
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

T-0_12

0.9999060627 -0.0107183350 0.0085430161 26.5831997562
 0.0106901199 0.9999372757 0.0033415621 16.9591161662
 -0.0085782962 -0.0032499223 0.9999579245 0.2674544988
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

T-LoopClosure

0.9974690596 0.0008993401 0.0710519157 0.0910237817
 0.0012964278 0.9995242686 -0.0308192436 0.3509997594
 -0.0710458729 0.0308331824 0.9969970304 -2.4865809026
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

Poses do *Groundtruth* (G)

Pose-1

-0.921438 0.388207 -0.0156965 -1.68238
 -0.388524 -0.920689 0.0371498 41.2186
 -2.97576e-05 0.0403297 0.999186 -0.262795

0 -0 0 1

Pose-2

-0.407744 0.912534 -0.0320468 10.8881
 -0.91306 -0.407164 0.0232124 72.5396
 0.00813376 0.0387254 0.999217 -0.413881

0 -0 0 1

Pose-3

-0.749543 0.66092 -0.0370166 -18.0654
-0.661956 -0.748393 0.0415084 96.1054
-0.000269256 0.0556157 0.998452 -1.66796
0 -0 0 1

Pose-4

0.940001 -0.34117 0.000543572 -44.9102
0.34117 0.939992 -0.00422287 71.2065
0.000929766 0.00415496 0.999991 -1.09916
0 -0 0 1

Pose-5

-0.0466467 -0.998883 0.00760308 -65.7978
0.99887 -0.0465743 0.00943839 63.7159
-0.00907374 0.00803477 0.999927 -0.992734
0 -0 0 1

Pose-6

0.770275 0.636505 0.0392143 -70.7102
-0.637335 0.770476 0.0130475 42.1765
-0.0219089 -0.0350429 0.999146 -0.559518
0 -0 0 1

Pose-7

-0.474543 0.880136 0.0130239 -79.6015
-0.879989 -0.474707 0.0164908 23.1039
0.0206967 -0.0036353 0.999779 -0.131855
0 -0 0 1

Pose-8

-0.98457 0.174868 -0.00654765 -98.6852
-0.174862 -0.984592 -0.00138895 4.67199
-0.00668964 -0.000222585 0.999978 -0.114828
0 -0 0 1

Pose-9

0.41369 -0.910352 0.0109021 -85.6062
0.90935 0.412596 -0.0533588 -20.0621
0.0440771 0.0319878 0.998516 -0.934659
0 -0 0 1

Pose-10

-0.776493 0.629291 -0.0324143 -69.779
-0.629796 -0.776723 0.00764482 -40.3554
-0.0203661 0.0263506 0.999445 -1.41587
0 -0 0 1

Pose-11

-0.766165 -0.64255 -0.0110213 -47.1943
0.642643 -0.766017 -0.0151054 -48.7825
0.00126351 -0.018656 0.999825 -1.39232
0 -0 0 1

Pose-12

0.999909 0.0101107 -0.0089379 -26.7516
 -0.0101315 0.999946 -0.00228384 -16.7652
 0.00891432 0.00237419 0.999957 -0.568388
 0 -0 0 1

Poses O obtidas sem nenhuma otimização (O = Originais)

Pose-1

-0.9213735900 0.3884060700 -0.0145408600 -1.6707991300
 -0.3886781600 -0.9207240300 0.0345910700 41.1916831800
 0.0000472600 0.0375230100 0.9992957600 -0.1519919700
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-2

-0.4055662910 0.9139228459 -0.0161557675 10.8248626654
 -0.9139939170 -0.4052497114 0.0196925152 72.3129053143
 0.0114503172 0.0227528906 0.9996755427 -0.3300530988
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-3

-0.7469693143 0.6645427848 -0.0204873158 -18.1110163582
 -0.6648041318 -0.7469476471 0.0102311705 95.6553701725
 -0.0085039043 0.0212624198 0.9997377594 -1.1302363837
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-4

0.9387793859 -0.3439590555 0.0196325580 -44.8349296992
 0.3429201110 0.9383879842 0.0428226707 70.7323731820
 -0.0331522045 -0.0334686442 0.9988897717 -0.9910609682
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-5

-0.0488194317 -0.9985220914 -0.0238806805 -65.6671410436
 0.9979283059 -0.0497655442 0.0407736814 63.2685446523
 -0.0419018593 -0.0218406617 0.9988829835 -0.1027637572
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-6

0.7690606259 0.6367836772 0.0552476423 -70.5472812908
 -0.6349061536 0.7710382745 -0.0489300763 41.5959677586
 -0.0737559236 0.0025531243 0.9972730521 0.8982802848
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-7

-0.4674045323 0.8814944307 -0.0670862583 -79.2710271134
 -0.8775276549 -0.4718181982 -0.0856318005 22.5080977763
 -0.1071364761 0.0188453524 0.9940657038 0.1212928659
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-8

-0.9808411650 0.1768502214 -0.0816983091 -97.9098313478

-0.1760368586 -0.9842348415 -0.0171111834 4.2641013570
 -0.0834364419 -0.0024014421 0.9965102049 -3.4409368209
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-9

0.4089247565 -0.9123154757 0.0214711398 -84.8533770896
 0.9033521537 0.4080183584 0.1321964774 -20.7616942895
 -0.1293655150 -0.0346624146 0.9909909568 -3.4744746168
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-10

-0.7707460562 0.6358022375 -0.0413041357 -69.0432288472
 -0.6339414360 -0.7717507923 -0.0501893153 -40.5456467499
 -0.0637869812 -0.0124988164 0.9978852619 0.1024467804
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-11

-0.7683779733 -0.6374373636 -0.0571742734 -46.2238452995
 0.6346529625 -0.7704360046 0.0603653648 -48.7489323467
 -0.0825282609 0.0100975919 0.9965375657 -0.3014013820
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-12

0.9967664194 -0.0044883420 0.0802282125 -26.2121199072
 0.0061625860 0.9997681174 -0.0206331088 -16.3270439592
 -0.0801170036 0.0210608004 0.9965629451 -1.0686880848
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

Poses S obtidas com otimização das rotações (S = SLERP)

Pose-1

-0.9219214176 0.3872999611 -0.0077226902 -1.6707991300
 -0.3873165615 -0.9212425162 0.0360292598 41.1916831800
 0.0068396603 0.0362072721 0.9993208957 -0.1519919700
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-2

-0.4079544766 0.9129656559 -0.0081765591 10.8690178511
 -0.9126578305 -0.4075382413 0.0311169771 72.2941336942
 0.0250764709 0.0201567108 0.9994823048 -0.4590748047
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-3

-0.7496248621 0.6618582323 -0.0024993058 -18.0085137902
 -0.6617551110 -0.7494292147 0.0208812150 95.7074247549
 0.0119473513 0.0173070063 0.9997788397 -1.2969488585
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-4

0.9406022665 -0.3394187627 -0.0078917495 -44.8239774921
 0.3394637701 0.9398330838 0.0384463697 70.8982705460

-0.0056324920 -0.0388417055 0.9992295016 -0.4140795332
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-5

-0.0436497951 -0.9986289675 -0.0288943021 -65.7003279993
 0.9990182749 -0.0438491975 0.0063035202 63.5104812798
 -0.0075618698 -0.0285907885 0.9995625968 -0.1358821527
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-6

0.7661197816 0.6420521985 0.0288002563 -70.6930615673
 -0.6418626544 0.7666393720 -0.0166254643 41.8416019695
 -0.0327538263 -0.0057487119 0.9994469166 0.0893411807
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-7

-0.4773716299 0.8779504826 -0.0363218547 -79.5311638998
 -0.8766821998 -0.4786682718 -0.0480104793 22.7944552321
 -0.0595369429 0.0089238827 0.9981862135 -0.2958640299
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-8

-0.9854679540 0.1678522246 -0.0260488445 -98.4146234386
 -0.1674161571 -0.9857197882 -0.0181198634 4.5983205601
 -0.0287183209 -0.0134955472 0.9994964373 -2.5879251949
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-9

0.4190925049 -0.9078220397 -0.0148531675 -85.5991129370
 0.9054160079 0.4166501002 0.0813913183 -20.5391994476
 -0.0677002589 -0.0475587871 0.9965715412 -3.3818741475
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-10

-0.7791511208 0.6265144409 0.0200794985 -70.0573798538
 -0.6268176647 -0.7789741877 -0.0172867006 -40.6073601538
 0.0048110434 -0.0260551365 0.9996489302 -0.2380880324
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-11

-0.7622683537 -0.6472001606 -0.0088830765 -47.3850154340
 0.6472188757 -0.7623036138 0.0009630029 -49.0800016054
 -0.0073948569 -0.0050152281 0.9999600810 -1.7741527525
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-12

0.9999536527 0.0094672857 -0.0017501253 -26.9703198090
 -0.0094753182 0.9999443439 -0.0046397669 -16.9017877530
 0.0017061019 0.0046561349 0.9999877047 -1.5754140836
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

Poses L obtidas com otimização das translações (L = LU & MILIUS, 1997)

Pose-1

-0.9213735900 0.3884060700 -0.0145408600 -1.6885268232
 -0.3886781600 -0.9207240300 0.0345910700 41.1311876016
 0.0000472600 0.0375230100 0.9992957600 0.0460654601
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-2

-0.4055662910 0.9139228459 -0.0161557675 10.7894072790
 -0.9139939170 -0.4052497114 0.0196925152 72.1919141576
 0.0114503172 0.0227528906 0.9996755427 0.0660617614
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-3

-0.7469693143 0.6645427848 -0.0204873158 -18.1641994378
 -0.6648041318 -0.7469476471 0.0102311705 95.4738834374
 -0.0085039043 0.0212624198 0.9997377594 -0.5360640934
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-4

0.9387793859 -0.3439590555 0.0196325580 -44.9058404719
 0.3429201110 0.9383879842 0.0428226707 70.4903908685
 -0.0331522045 -0.0334686442 0.9988897717 -0.1988312478
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-5

-0.0488194317 -0.9985220914 -0.0238806805 -65.7557795094
 0.9979283059 -0.0497655442 0.0407736814 62.9660667605
 -0.0419018593 -0.0218406617 0.9988829835 0.8875233933
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-6

0.7690606259 0.6367836772 0.0552476423 -70.6536474498
 -0.6349061536 0.7710382745 -0.0489300763 41.2329942885
 -0.0737559236 0.0025531243 0.9972730521 2.0866248655
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-7

-0.4674045323 0.8814944307 -0.0670862583 -79.3951209656
 -0.8775276549 -0.4718181982 -0.0856318005 22.0846287278
 -0.1071364761 0.0188453524 0.9940657038 1.5076948766
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-8

-0.9808411650 0.1768502214 -0.0816983091 -98.0516528931
 -0.1760368586 -0.9842348415 -0.0171111834 3.7801367301
 -0.0834364419 -0.0024014421 0.9965102049 -1.8564773800
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-9

0.4089247565 -0.9123154757 0.0214711398 -85.0129263282
 0.9033521537 0.4080183584 0.1321964774 -21.3061544948
 -0.1293655150 -0.0346624146 0.9909909568 -1.6919577458

0.0000000000 0.0000000000 0.0000000000 1.0000000000
 Pose-10
 -0.7707460562 0.6358022375 -0.0413041357 -69.2205057789
 -0.6339414360 -0.7717507923 -0.0501893153 -41.1506025335
 -0.0637869812 -0.0124988164 0.9978852619 2.0830210814
 0.0000000000 0.0000000000 0.0000000000 1.0000000000
 Pose-11
 -0.7683779733 -0.6374373636 -0.0571742734 -46.4188499244
 0.6346529625 -0.7704360046 0.0603653648 -49.4143837086
 -0.0825282609 0.0100975919 0.9965375657 1.8772303491
 0.0000000000 0.0000000000 0.0000000000 1.0000000000
 Pose-12
 0.9967664194 -0.0044883420 0.0802282125 -26.4248522253
 0.0061625860 0.9997681174 -0.0206331088 -17.0529908995
 -0.0801170036 0.0210608004 0.9965629451 1.3080010764
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

Poses SL obtidas com otimização das translações e rotações (SL = SLERP+L)

Pose-1
 -0.9219214176 0.3872999611 -0.0077226902 -1.6532404075
 -0.3873165615 -0.9212425162 0.0360292598 41.2068170774
 0.0068396603 0.0362072721 0.9993208957 -0.0609423321
 0.0000000000 0.0000000000 0.0000000000 1.0000000000
 Pose-2
 -0.4079544766 0.9129656559 -0.0081765591 10.9041352960
 -0.9126578305 -0.4075382413 0.0311169771 72.3244014891
 0.0250764709 0.0201567108 0.9994823048 -0.2769755289
 0.0000000000 0.0000000000 0.0000000000 1.0000000000
 Pose-3
 -0.7496248621 0.6618582323 -0.0024993058 -17.9558376228
 -0.6617551110 -0.7494292147 0.0208812150 95.7528264472
 0.0119473513 0.0173070063 0.9997788397 -1.0237999447
 0.0000000000 0.0000000000 0.0000000000 1.0000000000
 Pose-4
 0.9406022665 -0.3394187627 -0.0078917495 -44.7537426022
 0.3394637701 0.9398330838 0.0384463697 70.9588061358
 -0.0056324920 -0.0388417055 0.9992295016 -0.0498809815
 0.0000000000 0.0000000000 0.0000000000 1.0000000000
 Pose-5
 -0.0436497951 -0.9986289675 -0.0288943021 -65.6125343869
 0.9990182749 -0.0438491975 0.0063035202 63.5861507671
 -0.0075618698 -0.0285907885 0.9995625968 0.3193660369
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-6

0.7661197816 0.6420521985 0.0288002563 -70.5877092323
-0.6418626544 0.7666393720 -0.0166254643 41.9324053542
-0.0327538263 -0.0057487119 0.9994469166 0.6356390082
0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-7

-0.4773716299 0.8779504826 -0.0363218547 -79.4082528424
-0.8766821998 -0.4786682718 -0.0480104793 22.9003925142
-0.0595369429 0.0089238827 0.9981862135 0.3414834355
0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-8

-0.9854679540 0.1678522246 -0.0260488445 -98.2741536587
-0.1674161571 -0.9857197882 -0.0181198634 4.7193917397
-0.0287183209 -0.0134955472 0.9994964373 -1.8595280916
0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-9

0.4190925049 -0.9078220397 -0.0148531675 -85.4410844346
0.9054160079 0.4166501002 0.0813913183 -20.4029943706
-0.0677002589 -0.0475587871 0.9965715412 -2.5624274063
0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-10

-0.7791511208 0.6265144409 0.0200794985 -69.8817926289
-0.6268176647 -0.7789741877 -0.0172867006 -40.4560211793
0.0048110434 -0.0260551365 0.9996489302 0.6724083468
0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-11

-0.7622683537 -0.6472001606 -0.0088830765 -47.1918694867
0.6472188757 -0.7623036138 0.0009630029 -48.9135287335
-0.0073948569 -0.0050152281 0.9999600810 -0.7726067354
0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-12

0.9999536527 0.0094672857 -0.0017501253 -26.7596151391
-0.0094753182 0.9999443439 -0.0046397669 -16.7201809836
0.0017061019 0.0046561349 0.9999877047 -0.4828184286
0.0000000000 0.0000000000 0.0000000000 1.0000000000

APENDICE E – TRANSFORMACOES DO CIRCUITO, LOOPCLOSURE E POSES DE BREMEN (ICP)

Poses do *Groundtruth* (G)

Pose-1

-0.921438 0.388207 -0.0156965 -1.68238
 -0.388524 -0.920689 0.0371498 41.2186
 -2.97576e-05 0.0403297 0.999186 -0.262795
 0 -0 0 1

Pose-2

-0.407744 0.912534 -0.0320468 10.8881
 -0.91306 -0.407164 0.0232124 72.5396
 0.00813376 0.0387254 0.999217 -0.413881
 0 -0 0 1

Pose-3

-0.749543 0.66092 -0.0370166 -18.0654
 -0.661956 -0.748393 0.0415084 96.1054
 -0.000269256 0.0556157 0.998452 -1.66796
 0 -0 0 1

Pose-4

0.940001 -0.34117 0.000543572 -44.9102
 0.34117 0.939992 -0.00422287 71.2065
 0.000929766 0.00415496 0.999991 -1.09916
 0 -0 0 1

Pose-5

-0.0466467 -0.998883 0.00760308 -65.7978
 0.99887 -0.0465743 0.00943839 63.7159
 -0.00907374 0.00803477 0.999927 -0.992734
 0 -0 0 1

Pose-6

0.770275 0.636505 0.0392143 -70.7102
 -0.637335 0.770476 0.0130475 42.1765
 -0.0219089 -0.0350429 0.999146 -0.559518
 0 -0 0 1

Pose-7

-0.474543 0.880136 0.0130239 -79.6015
 -0.879989 -0.474707 0.0164908 23.1039
 0.0206967 -0.0036353 0.999779 -0.131855
 0 -0 0 1

Pose-8

-0.98457 0.174868 -0.00654765 -98.6852
 -0.174862 -0.984592 -0.00138895 4.67199
 -0.00668964 -0.000222585 0.999978 -0.114828

0 -0 0 1

Pose-9

0.41369 -0.910352 0.0109021 -85.6062

0.90935 0.412596 -0.0533588 -20.0621

0.0440771 0.0319878 0.998516 -0.934659

0 -0 0 1

Pose-10

-0.776493 0.629291 -0.0324143 -69.779

-0.629796 -0.776723 0.00764482 -40.3554

-0.0203661 0.0263506 0.999445 -1.41587

0 -0 0 1

Pose-11

-0.766165 -0.64255 -0.0110213 -47.1943

0.642643 -0.766017 -0.0151054 -48.7825

0.00126351 -0.018656 0.999825 -1.39232

0 -0 0 1

Pose-12

0.999909 0.0101107 -0.0089379 -26.7516

-0.0101315 0.999946 -0.00228384 -16.7652

0.00891432 0.00237419 0.999957 -0.568388

0 -0 0 1

Poses O obtidas sem nenhuma otimização (O = Originais)

Pose-1

-0.9212007425 0.3887914760 -0.0151467229 -1.6621281121

-0.3890858829 -0.9204513456 0.0371582670 41.2484543831

0.0005050210 0.0401235946 0.9991941705 -0.2874162482

0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-2

-0.4074580903 0.9130712226 -0.0166763560 10.8700821907

-0.9131049327 -0.4070444965 0.0234913161 72.5595637485

0.0146612418 0.0247989869 0.9995843056 -0.4644520709

0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-3

-0.7493730272 0.6618480638 -0.0199087217 -18.0770102878

-0.6621332851 -0.7492106601 0.0161868574 96.0969041679

-0.0042026342 0.0253122724 0.9996701232 -1.2916120083

0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-4

0.9399401496 -0.3410335893 0.0143699459 -44.9188051847

0.3403364453 0.9395763055 0.0369825869 71.1790016416

-0.0261140459 -0.0298708551 0.9992117997 -0.8907183741

0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-5

-0.0457746543 -0.9987595020 -0.0196027621 -65.8231253343
 0.9983354722 -0.0464244875 0.0341826427 63.7548268033
 -0.0350503708 -0.0180054946 0.9992230630 -0.1669000442
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-6

0.7700446332 0.6366166244 0.0418435637 -70.7362781986
 -0.6354029055 0.7711717630 -0.0394247532 42.2108336116
 -0.0573670453 0.0037711201 0.9983465688 0.6804759629
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-7

-0.4718535687 0.8800588983 -0.0533692162 -79.6230263053
 -0.8779079935 -0.4745679334 -0.0637280103 23.1199617520
 -0.0814116869 0.0167827392 0.9965399551 0.1074705441
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-8

-0.9828876406 0.1748780124 -0.0578378840 -98.6338970766
 -0.1747831837 -0.9845837517 -0.0067358720 4.8176183820
 -0.0581242937 0.0034882382 0.9983041810 -2.6327970373
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-9

0.4133734426 -0.9103645432 0.0188949922 -85.4974287343
 0.9046169362 0.4129556465 0.1055076503 -19.9257121637
 -0.1038532545 -0.0265216706 0.9942403922 -2.9246188017
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-10

-0.7760061059 0.6301767060 -0.0261962298 -69.6926480508
 -0.6298047790 -0.7764510139 -0.0216330833 -40.0450233362
 -0.0339731445 -0.0002890849 0.9994239595 -0.3193232241
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-11

-0.7656353000 -0.6426467204 -0.0283844352 -47.1042515347
 0.6408609794 -0.7658415909 0.0527154107 -48.4522219158
 -0.0556155514 0.0221701867 0.9982068968 -0.6861517705
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-12

0.9980116800 0.0104144556 0.0621135801 -26.6524930293
 -0.0083220565 0.9993935543 -0.0338135904 -16.4319178780
 -0.0624281065 0.0332292882 0.9974963316 -1.6857645428
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

Poses S obtidas com otimização das rotações (S = SLERP)

Pose-1

-0.9212737316 0.3888063359 -0.0091839352 -1.6621281121
 -0.3888689477 -0.9205457241 0.0371013668 41.2484543831
 0.0059710142 0.0377518619 0.9992693050 -0.2874162482
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-2

-0.4076206885 0.9131173209 -0.0078824281 10.8762089341
 -0.9127925080 -0.4072033557 0.0315478120 72.5566319242
 0.0255971023 0.0200545622 0.9994711616 -0.5372716258
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-3

-0.7495911843 0.6618947641 -0.0028944195 -18.0691293014
 -0.6617885661 -0.7493759545 0.0217157217 96.0983453840
 0.0122045141 0.0181934073 0.9997599961 -1.2996855694
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-4

0.9400159126 -0.3409999948 -0.0094386187 -44.9098203449
 0.3411041844 0.9392389742 0.0384458662 71.1958529233
 -0.0042449216 -0.0393592784 0.9992161067 -0.3045523288
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-5

-0.0463786261 -0.9984674727 -0.0301948506 -65.8193481019
 0.9988941804 -0.0465893476 0.0063126158 63.7548110175
 -0.0077097000 -0.0298686901 0.9995240977 -0.0673583894
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-6

0.7710941296 0.6362058475 0.0256117725 -70.7200961682
 -0.6362468726 0.7714484277 -0.0075657473 42.1921808708
 -0.0245715343 -0.0104615068 0.9996433347 0.1274701177
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-7

-0.4733667657 0.8806236223 -0.0206383371 -79.5897839107
 -0.8798056789 -0.4738164179 -0.0379469305 23.0850867234
 -0.0431957463 0.0001949104 0.9990666091 0.0187819014
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-8

-0.9844277986 0.1754159544 -0.0114521773 -98.6614451147
 -0.1751993404 -0.9843725356 -0.0177736378 4.7145499964
 -0.0143909885 -0.0154904493 0.9997764477 -1.6232773003
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-9

0.4134502242 -0.9102864380 -0.0209168093 -85.5384623529
 0.9088757746 0.4112089132 0.0696567010 -20.0098212054
 -0.0548063719 -0.0478103599 0.9973517088 -2.7971342952
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-10

-0.7759676402 0.6299991699 0.0312292718 -69.7668788571
 -0.6304322032 -0.7762242349 -0.0055833978 -40.1676751111
 0.0207233816 -0.0240204746 0.9994966524 -0.2903077213
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-11

-0.7663550375 -0.6424165793 0.0009461056 -47.2272280814
 0.6424013138 -0.7663455466 -0.0059207497 -48.5749593053
 0.0045286316 -0.0039296168 0.9999820246 -1.8183908837
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-12

0.9999472450 0.0097573958 -0.0032094189 -26.7844670038
 -0.0097725266 0.9999410461 -0.0047330902 -16.5345157213
 0.0031630470 0.0047642047 0.9999836486 -1.5383920008
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

Poses L obtidas com otimização das translações (L = LU & MILIUS, 1997)

Pose-1

-0.9212007425 0.3887914760 -0.0151467229 -1.6691299415
 -0.3890858829 -0.9204513456 0.0371582670 41.2214544016
 0.0005050210 0.0401235946 0.9991941705 -0.0961407941
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-2

-0.4074580903 0.9130712226 -0.0166763560 10.8560785320
 -0.9131049327 -0.4070444965 0.0234913161 72.5055637856
 0.0146612418 0.0247989869 0.9995843056 -0.0819011628
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-3

-0.7493730272 0.6618480638 -0.0199087217 -18.0980157759
 -0.6621332851 -0.7492106601 0.0161868574 96.0159042234
 -0.0042026342 0.0253122724 0.9996701232 -0.7177856462
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-4

0.9399401496 -0.3410335893 0.0143699459 -44.9468125022
 0.3403364453 0.9395763055 0.0369825869 71.0710017157
 -0.0261140459 -0.0298708551 0.9992117997 -0.1256165579
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-5

-0.0457746543 -0.9987595020 -0.0196027621 -65.8581344811
 0.9983354722 -0.0464244875 0.0341826427 63.6198268958
 -0.0350503708 -0.0180054946 0.9992230630 0.7894772261
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-6

0.7700446332 0.6366166244 0.0418435637 -70.7782891748

-0.6354029055 0.7711717630 -0.0394247532 42.0488337227
 -0.0573670453 0.0037711201 0.9983465688 1.8281286872
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-7

-0.4718535687 0.8800588983 -0.0533692162 -79.6720391108
 -0.8779079935 -0.4745679334 -0.0637280103 22.9309618816
 -0.0814116869 0.0167827392 0.9965399551 1.4463987224
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-8

-0.9828876406 0.1748780124 -0.0578378840 -98.6899117115
 -0.1747831837 -0.9845837517 -0.0067358720 4.6016185301
 -0.0581242937 0.0034882382 0.9983041810 -1.1025934049
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-9

0.4133734426 -0.9103645432 0.0188949922 -85.5604451986
 0.9046169362 0.4129556465 0.1055076503 -20.1687119972
 -0.1038532545 -0.0265216706 0.9942403922 -1.2031397153
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-10

-0.7760061059 0.6301767060 -0.0261962298 -69.7626663444
 -0.6298047790 -0.7764510139 -0.0216330833 -40.3150231512
 -0.0339731445 -0.0002890849 0.9994239595 1.5934313164
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-11

-0.7656353000 -0.6426467204 -0.0283844352 -47.1812716577
 0.6408609794 -0.7658415909 0.0527154107 -48.7492217123
 -0.0556155514 0.0221701867 0.9982068968 1.4178782240
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-12

0.9980116800 0.0104144556 0.0621135801 -26.7365149816
 -0.0083220565 0.9993935543 -0.0338135904 -16.7559176560
 -0.0624281065 0.0332292882 0.9974963316 0.6095409058
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

Poses SL obtidas com otimização das translações e rotações (SL = SLERP+L)

Pose-1

-0.9212737316 0.3888063359 -0.0091839352 -1.6605966582
 -0.3888689477 -0.9205457241 0.0371013668 41.2309379757
 0.0059710142 0.0377518619 0.9992693050 -0.2044697412
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-2

-0.4076206885 0.9131173209 -0.0078824281 10.8792718418
 -0.9127925080 -0.4072033557 0.0315478120 72.5215991095

0.0255971023 0.0200545622 0.9994711616 -0.3713786118
0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-3

-0.7495911843 0.6618947641 -0.0028944195 -18.0645349397
-0.6617885661 -0.7493759545 0.0217157217 96.0457961619
0.0122045141 0.0181934073 0.9997599961 -1.0508460484
0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-4

0.9400159126 -0.3409999948 -0.0094386187 -44.9036945293
0.3411041844 0.9392389742 0.0384458662 71.1257872937
-0.0042449216 -0.0393592784 0.9992161067 0.0272336992
0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-5

-0.0463786261 -0.9984674727 -0.0301948506 -65.8116908325
0.9988941804 -0.0465893476 0.0063126158 63.6672289806
-0.0077097000 -0.0298686901 0.9995240977 0.3473741455
0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-6

0.7710941296 0.6362058475 0.0256117725 -70.7109074449
-0.6362468726 0.7714484277 -0.0075657473 42.0870824265
-0.0245715343 -0.0104615068 0.9996433347 0.6251491596
0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-7

-0.4733667657 0.8806236223 -0.0206383371 -79.5790637335
-0.8798056789 -0.4738164179 -0.0379469305 22.9624718717
-0.0431957463 0.0001949104 0.9990666091 0.5994074504
0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-8

-0.9844277986 0.1754159544 -0.0114521773 -98.6491934836
-0.1751993404 -0.9843725356 -0.0177736378 4.5744187374
-0.0143909885 -0.0154904493 0.9997764477 -0.9597052444
0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-9

0.4134502242 -0.9102864380 -0.0209168093 -85.5246792679
0.9088757746 0.4112089132 0.0696567010 -20.1674688718
-0.0548063719 -0.0478103599 0.9973517088 -2.0506157323
0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-10

-0.7759676402 0.6299991699 0.0312292718 -69.7515643182
-0.6304322032 -0.7762242349 -0.0055833978 -40.3428391849
0.0207233816 -0.0240204746 0.9994966524 0.5391573486
0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-11

-0.7663550375 -0.6424165793 0.0009461056 -47.2103820886
0.6424013138 -0.7663455466 -0.0059207497 -48.7676397865

0.0045286316 -0.0039296168 0.9999820246 -0.9059793068

0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-12

0.9999472450 0.0097573958 -0.0032094189 -26.7660895571

-0.0097725266 0.9999410461 -0.0047330902 -16.7447126099

0.0031630470 0.0047642047 0.9999836486 -0.5430339170

0.0000000000 0.0000000000 0.0000000000 1.0000000000

APENDICE F – TRANSFORMACOES DO CIRCUITO, LOOPCLOSURE E POSES DE UFPR

Poses do *Groundtruth* (G)

Pose-1

0.9994309808 -0.0337243974 -0.0006161855 -23.1704877172
 0.0337243467 0.9994311681 -0.0000924865 -14.3545280596
 0.0006189541 0.0000716534 0.9999998059 -2.2765052130
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-2

0.9995858256 -0.0287683782 -0.0007468440 -45.9965040952
 0.0287680480 0.9995860131 -0.0004491828 -21.8600625261
 0.0007594570 0.0004275115 0.9999996202 -3.1997155159
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-3

0.9987080553 0.0508092945 -0.0007973854 -51.9884155852
 -0.0508091421 0.9987083593 0.0002102974 -18.9281428622
 0.0008070405 -0.0001695112 0.9999996600 -2.8844906460
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-4

0.9982099928 -0.0597742539 -0.0019618481 -57.9664085336
 0.0597701979 0.9982100279 -0.0020648491 -3.2594889650
 0.0020817613 0.0019438930 0.9999959438 -2.3220556161
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-5

0.9993644157 -0.0356200313 -0.0014064004 -65.9757408873
 0.0356177925 0.9993642266 -0.0015860349 21.0287216096
 0.0014620008 0.0015349340 0.9999977533 -4.1463613398
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-6

0.9972164670 0.0745496386 -0.0012920438 -36.4676979141
 -0.0745513226 0.9972163200 -0.0013082452 21.4877249113
 0.0011909179 0.0014009272 0.9999983096 -5.7825521876
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-7

0.9958605234 0.0908792371 -0.0016680151 -19.8990062063
 -0.0908822979 0.9958599016 -0.0018612516 6.2536976348
 0.0014919602 0.0020051400 0.9999968767 -5.1545698473
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

Poses O obtidas sem nenhuma otimização (O = Originais)

Pose-1

0.9993531529 -0.0351402372 -0.0076445764 -22.8310371488
 0.0350066268 0.9992431468 -0.0169608251 -13.5959102331
 0.0082347980 0.0166822432 0.9998269305 -2.1744956739
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-2

0.9994455117 -0.0315981240 -0.0104989396 -45.4680404033
 0.0314626719 0.9994226366 -0.0128255080 -21.3249855180
 0.0108981398 0.0124880718 0.9998626299 -3.3932827227
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-3

0.9989294430 0.0429555481 -0.0171694153 -51.2303540172
 -0.0431587007 0.9990003948 -0.0116420714 -18.4848331291
 0.0166521610 0.0123706177 0.9997848135 -3.2567182847
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-4

0.9983510959 -0.0498519685 0.0284582273 -58.8819808075
 0.0496523147 0.9987370338 0.0076801835 -3.6465284137
 -0.0288051579 -0.0062545026 0.9995654777 -2.6915209005
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-5

0.9985535931 -0.0247415094 0.0477344685 -67.4818217063
 0.0237222163 0.9994808059 0.0218031010 20.0499966272
 -0.0482491268 -0.0206391972 0.9986220733 -4.5995956043
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-6

0.9949971802 0.0864594402 0.0500537390 -38.2034686090
 -0.0863902061 0.9962550561 -0.0035490435 21.0448960273
 -0.0501731389 -0.0007928643 0.9987402202 -7.7204518859
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-7

0.9935641763 0.1021709530 0.0489011656 -21.6901790664
 -0.1017309853 0.9947465158 -0.0114094803 5.8987289383
 -0.0498099816 0.0063612874 0.9987384541 -7.7796159302
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

Poses S obtidas com otimização das rotações (S = SLERP)

Pose-1

0.9992236673 -0.0367670578 -0.0141508375 -22.8310371488
 0.0365277465 0.9991911807 -0.0168139288 -13.5959102331
 0.0147575907 0.0162839774 0.9997584936 -2.1744956739
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-2

0.9991161009 -0.0348471412 -0.0235094360 -45.4478219573
 0.0345603807 0.9993244896 -0.0124957846 -21.3595969313
 0.0239289975 0.0116722445 0.9996455181 -3.5398442540
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-3

0.9986039148 0.0380956950 -0.0365915217 -51.2201775872
 -0.0384746613 0.9992124106 -0.0097086999 -18.5372468668
 0.0361928429 0.0111029922 0.9992831439 -3.4796427978
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-4

0.9984056551 -0.0563936731 0.0024293007 -58.9497587371
 0.0563731522 0.9983791552 0.0078185823 -3.7336127700
 -0.0028662817 -0.0076691694 0.9999664836 -3.0947788863
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-5

0.9993409025 -0.0329539964 0.0152248053 -67.6564267022
 0.0326025097 0.9992086478 0.0227849648 19.9045863738
 -0.0159636128 -0.0222735804 0.9996244548 -5.2285046639
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-6

0.9969939854 0.0766346642 0.0114070697 -38.3024870619
 -0.0766611199 0.9970553966 0.0018996918 21.1588993345
 -0.0112278982 -0.0027684601 0.9999331327 -7.4020858851
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-7

0.9958674667 0.0907342736 0.0039090123 -21.6512844079
 -0.0907170143 0.9958670469 -0.0043872854 6.1788875198
 -0.0042909337 0.0040145409 0.9999827355 -6.7423090081
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

Poses L obtidas com otimização das translações (L = LU & MILIUS, 1997)

Pose-1

0.9993531529 -0.0351402372 -0.0076445764 -22.6168888705
 0.0350066268 0.9992431468 -0.0169608251 -13.5060634421
 0.0082347980 0.0166822432 0.9998269305 -1.7095824910
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-2

0.9994455117 -0.0315981240 -0.0104989396 -45.0397438467
 0.0314626719 0.9994226366 -0.0128255080 -21.1452919360
 0.0108981398 0.0124880718 0.9998626299 -2.4634563569
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-3

0.9989294430 0.0429555481 -0.0171694153 -50.5879091822

-0.0431587007 0.9990003948 -0.0116420714 -18.2152927561
 0.0166521610 0.0123706177 0.9997848135 -1.8619787361
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-4

0.9983510959 -0.0498519685 0.0284582273 -58.0253876942
 0.0496523147 0.9987370338 0.0076801835 -3.2871412497
 -0.0288051579 -0.0062545026 0.9995654777 -0.8318681690
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-5

0.9985535931 -0.0247415094 0.0477344685 -66.4110803146
 0.0237222163 0.9994808059 0.0218031010 20.4992305822
 -0.0482491268 -0.0206391972 0.9986220733 -2.2750296899
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-6

0.9949971802 0.0864594402 0.0500537390 -36.9185789391
 -0.0863902061 0.9962550561 -0.0035490435 21.5839767733
 -0.0501731389 -0.0007928643 0.9987402202 -4.9309727887
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-7

0.9935641763 0.1021709530 0.0489011656 -20.1911411182
 -0.1017309853 0.9947465158 -0.0114094803 6.5276564753
 -0.0498099816 0.0063612874 0.9987384541 -4.5252236501
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

Poses obtidas com otimização das translações e rotações (SL = SLERP+LUM)

Pose-1

0.9992236673 -0.0367670578 -0.0141508375 -22.6053314850
 0.0365277465 0.9991911807 -0.0168139288 -13.5729086981
 0.0147575907 0.0162839774 0.9997584936 -1.9569798434
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-2

0.9991161009 -0.0348471412 -0.0235094360 -44.9964106297
 0.0345603807 0.9993244896 -0.0124957846 -21.3135938614
 0.0239289975 0.0116722445 0.9996455181 -3.1048125931
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-3

0.9986039148 0.0380956950 -0.0365915217 -50.5430605958
 -0.0384746613 0.9992124106 -0.0097086999 -18.4682422619
 0.0361928429 0.0111029922 0.9992831439 -2.8270953065
 0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-4

0.9984056551 -0.0563936731 0.0024293007 -58.0469360819
 0.0563731522 0.9983791552 0.0078185823 -3.6416066301
 -0.0028662817 -0.0076691694 0.9999664836 -2.2247155645

0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-5

0.9993409025 -0.0329539964 0.0152248053 -66.5278983832

0.0326025097 0.9992086478 0.0227849648 20.0195940487

-0.0159636128 -0.0222735804 0.9996244548 -4.1409255116

0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-6

0.9969939854 0.0766346642 0.0114070697 -36.9482530790

-0.0766611199 0.9970553966 0.0018996918 21.2969085443

-0.0112278982 -0.0027684601 0.9999331327 -6.0969909023

0.0000000000 0.0000000000 0.0000000000 1.0000000000

Pose-7

0.9958674667 0.0907342736 0.0039090123 -20.0713447613

-0.0907170143 0.9958670469 -0.0043872854 6.3398982647

-0.0042909337 0.0040145409 0.9999827355 -5.2196981949

0.0000000000 0.0000000000 0.0000000000 1.0000000000