



UNIVERSIDADE FEDERAL DA BAHIA

DISSERTAÇÃO DE MESTRADO

**MAPEAMENTO DE AMBIENTES TRIDIMENSIONAIS
UTILIZANDO SENSORES RGB-D e LiDAR APLICADOS À
ROBÓTICA MÓVEL**

Ricardo Sutana de Mello

Programa de Pós-Graduação em Engenharia Elétrica

Salvador
2020

RICARDO SUTANA DE MELLO

**MAPEAMENTO DE AMBIENTES TRIDIMENSIONAIS
UTILIZANDO SENSORES RGB-D E LIDAR APLICADOS À
ROBÓTICA MÓVEL**

Esta Dissertação de Mestrado foi apresentada ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal da Bahia, como requisito parcial para obtenção do grau de Mestre em Engenharia Elétrica.

Orientador: Prof. Dr. André Gustavo Scolari Conceição
Co-orientador: Prof Dr. Paulo César Machado de Abreu Farias

Salvador
2020

Dedico este trabalho para minhas famílias: Sutana de Mello e Menezes Evangelista, à minha esposa, Neima, que sempre segurou minha mão para que eu pudesse seguir em frente e ao Samuel que sempre estará vivo em minha memória e em minhas fotografias.

AGRADECIMENTOS

Agradeço primeiramente aos meus orientadores que me deram a oportunidade de ampliar meu conhecimento compartilhando o conhecimento deles, agradeço pelas conversas e orientações.

Agradeço também à minha esposa Neima, por estar sempre me incentivando quando eu mesmo já não tinha condições de acreditar que chegar até aqui era possível e este trabalho é também por você. Muito obrigado por estar nessa e em outras jornadas comigo.

Não posso deixar de agradecer também ao Erick Suzart de Souza que gentilmente me cedeu o modelo 3D do laboratório. Agradeço também ao Henrique que sempre esteve disposto a debugar código ou me dar uma dica valiosa. Não poderia deixar de agradecer profundamente à Nara por ter sido uma companheira de pesquisa incrível nesta jornada, serei sempre muito grato!

Su e Aluska, vocês foram as melhores pessoas com quem eu pude dividir meus dias em Salvador! Sentirei saudade dos programas furados.

Ao meus pais, que sempre me incentivaram a buscar o desconhecido. Muito obrigado!

Agradeço à Fundação de Amparo à Pesquisa do Estado da Bahia (FAPESB) pelo apoio financeiro.

*"Como um velho boiadeiro
Levando a boiada
Eu vou tocando os dias
Pela longa estrada, eu vou
Estrada eu sou."*

—RENATO TEIXEIA E ALMIR SATER (Tocando em Frente)

RESUMO

Tarefas autônomas realizadas por robôs terrestres móveis, necessitam de uma forma de modelar o ambiente físico em que estão situados. Para isso, faz-se a utilização de mapas de ocupação bidimensionais. Para a construção destes mapas, o sensor mais empregado é o sensores do tipo LiDAR 2D. Entretanto, a utilização deste tipo de sensor associado aos objetos contidos em um ambiente como mesas e cadeiras podem gerar mapas inconsistentes. Este trabalho, atua na utilização de sensores do tipo RGB-D para construção de mapas que levem em consideração toda a geometria dos objetos contidos no ambiente e avalia a qualidade dos mapas construídos. Dessa maneira, mapas mais coerentes com o ambiente real podem ser utilizados para as mesmas tarefas como navegação autônoma com rotas que ofereçam segurança para a estrutura do robô. Para o desenvolvimento deste trabalho foi utilizado o *framework* ROS, o pacote RTAB-Map, o robô terrestre móvel HUSKY A200, os sensores: *Microsoft Kinect* e SICK LMS111 e o índice de similaridade estrutural. Os resultados obtidos mostraram que utilizar o sensor RGB-D para mapeamento criou mapas mais consistentes e que a persistência de consecutivas observações corrigiu erros inseridos pelo sensor.

Palavras-chave: ROS; Mapa de Ocupação; SLAM; RTAB-Map; HECTOR SLAM; vSLAM; RGB-D; LiDAR; Robótica.

ABSTRACT

For autonomous tasks performed by mobile robots is necessary a model of the physical environment in which they are located. For this, two-dimensional occupation maps are applied. For the construction of these maps, the most used sensor is the LiDAR 2D sensor. However, the use of this type of sensor associated with objects in an environment such as tables and chairs can generate inconsistent maps. This work uses RGB-D sensors to build maps that take into account all the geometry of objects contained in the environment and evaluate the quality of the built maps. In this way, maps more consistent with the real environment can be used for the same tasks with routes that offer security for the robot's structure. For the development of this work, were used the framework ROS, the RTAB-Map package, the HUSKY A200 mobile terrestrial robot, the sensors: Microsoft Kinect and SICK LMS111, and the structural similarity index. The results obtained showed that the use of the RGB-D sensor for the mapping task created more consistent maps and that the persistence of consecutive observations corrected errors inserted by the sensor.

Keywords: ROS; Occupancy grid; SLAM; RTAB-Map; HECTOR SLAM; vSLAM; RGB-D; LiDAR; Robotics;

SUMÁRIO

Capítulo 1—INTRODUÇÃO	1
1.1 Objetivo	6
1.1.1 Objetivos Específicos	6
1.2 Justificativa	7
1.3 Estrutura da dissertação	7
Capítulo 2—REVISÃO BIBLIOGRÁFICA	8
2.1 Definição matemática e revisão histórica	8
2.2 Soluções populares e disponíveis para o ROS	14
2.2.1 Sistemas baseados em LiDAR	14
2.2.2 Sistemas baseados em câmeras RGB, <i>Stereo</i> e RGB-D	15
2.3 HECTOR SLAM	16
2.4 RTAB-Map	17
2.5 Métricas de similaridade para imagens	20
Capítulo 3—METODOLOGIA	22
3.1 Montagem do sistema	22
3.2 Experimentos	24
3.2.1 Simulação computacional	26
3.2.2 Experimento Real	28
3.2.3 <i>Ground Truth</i> e análise métrica	29
Capítulo 4—RESULTADOS E DISCUSSÃO	31
4.1 Resultados das simulações computacionais	31
4.1.1 Simulação no GAZEBO com: Sensor LiDAR + pacote HECTOR SLAM	31
4.1.2 Simulação no GAZEBO com: Sensor LiDAR + Sensor RGB-D + Odometria filtrada + pacote RTAB-Map	35
4.2 Resultados dos experimentos em ambiente real	38
4.2.1 Experimento real com: Sensor LiDAR + pacote HECTOR SLAM	38
4.2.2 Experimento real com: Sensor LiDAR + Sensor RGB-D + Odometria filtrada + pacote RTAB-Map	41
4.3 Análise comparativa métrica	43

Capítulo 5—CONCLUSÃO	47
5.1 Publicação	49
Apêndice A—Anexo	51
A.1 MAPAS EXPERIMENTO SIMULAÇÃO 4.1.1	51
A.2 MAPAS EXPERIMENTO SIMULAÇÃO 4.1.2	60
A.3 MAPAS EXPERIMENTO REAL 4.2.1	69
A.4 MAPAS EXPERIMENTO REAL 4.2.2	78
A.5 MAPAS DE SIMILARIDADE LOCAL	87

LISTA DE FIGURAS

1.1	Vítimas de acidente fatal por setor da indústria nos EUA em 2018. Fonte: Adaptado de <i>US Bureau of Labor Statistics [Statistics 2018]</i>	2
1.2	Utilização de robôs pela indústria em três continentes. Fonte: Adaptado de <i>International Federation of Robotics(IFR)</i>	3
1.3	Sensores tipo exteroceptivos	3
1.4	Sensores tipo RGB-D.	4
1.5	Estrutura interna do sensor RGB-D Microsoft Kinect. Adaptado de [TechCrunch 2010].	4
1.6	Sensor tipo LiDAR 2D E 3D	5
1.7	Interceptação do sensor LiDAR com os objetos. Fonte: Próprio autor. . .	5
2.1	Estimação da posição e orientação de um robô baseada na observação de marcos no ambiente. Fonte: Próprio autor.	10
2.2	Representação de um grafo. Fonte: Próprio autor.	12
2.3	Fluxo de dados para o vSLAM baseado em grafos. Adaptado de [Frost 2016]	13
2.4	Usuários do ROS ate 2017. Adaptado de [Zhang et al. 2017]	14
2.5	Representação em diagrama de blocos do HECTOR SLAM no ROS. Fonte: [Kohlbrecher et al. 2013]	17
2.6	Diagrama de blocos do RTAB-Map. Adaptado de [Labbé e Michaud 2019].	18
2.7	Tempo de processamento de cada módulo interno do nó <i>rtabmap_ros</i> sem o gerenciamento de memória e com o gerenciamento de memória para um mapa com taxa de atualização de 2Hz usando sessões combinadas do MIT <i>Stata Center</i> . Fonte: [Labbé e Michaud 2019]	19
2.8	Diagrama do sistema de similaridade estrutural (SSIM). Adaptado de [Wang et al. 2004].	21
3.1	Posicionamento dos sensores LiDAR e RGB-D acoplados ao robô terrestre móvel. Fonte: Próprio autor.	22
3.2	Trajetória a ser percorrida. Em vermelho o ponto inicial e final. Fonte: Próprio autor.	23
3.3	Modelo 3D do Laboratório de Robótica. Fonte: Próprio autor.	24
3.4	Erro no mapeamento ao se utilizar o processo de integração da posição como fonte de odometria.	27
3.5	Relação de entradas e saídas utilizadas no pacote HECTOR SLAM. Fonte: Próprio autor.	27
3.6	Relação de entradas e saídas utilizadas no pacote RTAB-Map. Fonte: Próprio autor.	28

3.7	Medidas das paredes do Laboratório de Robótica. Fonte: Próprio autor. .	30
4.1	Relacionamento entre os tópicos subscritos e publicados pelo ROS no primeiro experimento simulado. Fonte: Próprio autor.	32
4.2	Mapa de ocupação adquiridos pelos sensor LiDAR em simulação no GAZEBO. Fonte: Próprio autor.	34
4.3	Relacionamento entre os tópicos subscritos e publicados na segunda simulação. Fonte: Próprio autor.	36
4.4	Mapa de ocupação adquiridos pelos sensor LiDAR em simulação no GAZEBO. Fonte: Próprio autor.	37
4.5	Relacionamento entre os tópicos subscritos e publicados no primeiro experimento real. Fonte: Próprio autor.	38
4.6	Resultado de sucessivas voltas pela trajetória usado sensor LiDAR e pacote HECTOR SLAM. Fonte: Próprio autor.	40
4.7	Relacionamento entre os tópicos subscritos e publicados no segundo experimento real. Fonte: Próprio autor.	41
4.8	Resultado de sucessivas voltas pela trajetória usado os sensores câmera RGB-D e LiDAR utilizando o pacote RTAB-Map. Fonte: Próprio autor.	42
4.9	Análise de similaridade dos mapas câmera RGB-D e LiDAR utilizando o pacote RTAB-Map. Fonte: Próprio autor.	45
4.10	Análise de similaridade local dos mapas a), b), c), e) f), g), h) e i) da Figura 4.8 contra o mapa da Figura 4.8 d). Fonte: Próprio autor.	46
A.1	Mapa da primeira volta em simulação usando sensor LiDAR e pacote HECTOR SLAM. Fonte: Próprio autor.	51
A.2	Mapa da segunda volta em simulação usando sensor LiDAR e pacote HECTOR SLAM. Fonte: Próprio autor.	52
A.3	Mapa da terceira volta em simulação usando sensor LiDAR e pacote HECTOR SLAM. Fonte: Próprio autor.	53
A.4	Mapa da quarta volta em simulação sensor usando LiDAR e pacote HECTOR SLAM. Fonte: Próprio autor.	54
A.5	Mapa da quinta volta em simulação usando sensor LiDAR e pacote HECTOR SLAM. Fonte: Próprio autor.	55
A.6	Mapa da sexta volta em simulação usando sensor LiDAR e pacote HECTOR SLAM. Fonte: Próprio autor.	56
A.7	Mapa da sétima volta em simulação usando sensor LiDAR e pacote HECTOR SLAM. Fonte: Próprio autor.	57
A.8	Mapa da oitava volta em simulação sensor LiDAR e pacote HECTOR SLAM. Fonte: Próprio autor.	58
A.9	Mapa da nona volta em simulação sensor LiDAR e pacote HECTOR SLAM. Fonte: Próprio autor.	59
A.10	Mapa da primeira volta em simulação usando os sensores LiDAR + RGB-D e pacote RTAB-Map. Fonte: Próprio autor.	60

A.11 Mapa da segunda volta em simulação usando os sensores LiDAR + RGB-D e pacote RTAB-Map. Fonte: Próprio autor.	61
A.12 Mapa da terceira volta em simulação usando os sensores LiDAR + RGB-D e pacote RTAB-Map. Fonte: Próprio autor.	62
A.13 Mapa da quarta volta em simulação usando os sensores LiDAR + RGB-D e pacote RTAB-Map. Fonte: Próprio autor.	63
A.14 Mapa da quinta volta em simulação usando os sensores LiDAR + RGB-D e pacote RTAB-Map. Fonte: Próprio autor.	64
A.15 Mapa da sexta volta em simulação usando os sensores LiDAR + RGB-D e pacote RTAB-Map. Fonte: Próprio autor.	65
A.16 Mapa da sétima volta em simulação usando os sensores LiDAR + RGB-D e pacote RTAB-Map. Fonte: Próprio autor.	66
A.17 Mapa da oitava volta em simulação usando os sensores LiDAR + RGB-D e pacote RTAB-Map. Fonte: Próprio autor.	67
A.18 Mapa da nona volta em simulação usando os sensores LiDAR + RGB-D e pacote RTAB-Map. Fonte: Próprio autor.	68
A.19 Mapa da primeira volta em ambiente real usando sensor LiDAR e pacote HECTOR SLAM. Fonte: Próprio autor.	69
A.20 Mapa da segunda volta em ambiente real usando sensor LiDAR e pacote HECTOR SLAM. Fonte: Próprio autor.	70
A.21 Mapa da terceira volta em ambiente real usando sensor LiDAR e pacote HECTOR SLAM. Fonte: Próprio autor.	71
A.22 Mapa da quarta volta em ambiente real usando sensor LiDAR e pacote HECTOR SLAM. Fonte: Próprio autor.	72
A.23 Mapa da quinta volta em ambiente real usando sensor LiDAR e pacote HECTOR SLAM. Fonte: Próprio autor.	73
A.24 Mapa da sexta volta em ambiente real usando sensor LiDAR e pacote HECTOR SLAM. Fonte: Próprio autor.	74
A.25 Mapa da sétima volta em ambiente real usando sensor LiDAR e pacote HECTOR SLAM. Fonte: Próprio autor.	75
A.26 Mapa da oitava volta em ambiente real usando sensor LiDAR e pacote HECTOR SLAM. Fonte: Próprio autor.	76
A.27 Mapa da nona volta em ambiente real usando sensor LiDAR e pacote HECTOR SLAM. Fonte: Próprio autor.	77
A.28 Mapa da primeira volta em ambiente real usando sensores LiDAR + RGB-D e pacote HECTOR SLAM. Fonte: Próprio autor.	78
A.29 Mapa da segunda volta em ambiente real usando sensores LiDAR + RGB-D e pacote HECTOR SLAM. Fonte: Próprio autor.	79
A.30 Mapa da terceira volta em ambiente real usando sensores LiDAR + RGB-D e pacote HECTOR SLAM. Fonte: Próprio autor.	80
A.31 Mapa da quarta volta em ambiente real usando sensores LiDAR + RGB-D e pacote HECTOR SLAM. Fonte: Próprio autor.	81
A.32 Mapa da quinta volta em ambiente real usando sensores LiDAR + RGB-D e pacote HECTOR SLAM. Fonte: Próprio autor.	82

A.33 Mapa da sexta volta em ambiente real usando sensores LiDAR + RGB-D e pacote HECTOR SLAM. Fonte: Próprio autor.	83
A.34 Mapa da sétima volta em ambiente real usando sensores LiDAR + RGB-D e pacote HECTOR SLAM. Fonte: Próprio autor.	84
A.35 Mapa da oitava volta em ambiente real usando sensores LiDAR + RGB-D e pacote HECTOR SLAM. Fonte: Próprio autor.	85
A.36 Mapa da nona volta em ambiente real usando sensores LiDAR + RGB-D e pacote HECTOR SLAM. Fonte: Próprio autor.	86
A.37 Mapa de similaridade local entre mapa 1 e mapa de referência (similaridade global: 0,7369). Fonte: Próprio autor.	87
A.38 Mapa de similaridade local entre mapa 2 e mapa de referência (similaridade global: 0,8199). Fonte: Próprio autor.	88
A.39 Mapa de similaridade local entre mapa 3 e mapa de referência (similaridade global: 0,9576). Fonte: Próprio autor.	89
A.40 Mapa de similaridade local entre mapa 5 e mapa de referência (similaridade global: 0,73439). Fonte: Próprio autor.	90
A.41 Mapa de similaridade local entre mapa 6 e mapa de referência (similaridade global: 0,7888). Fonte: Próprio autor.	91
A.42 Mapa de similaridade local entre mapa 7 e mapa de referência (similaridade global: 0,7924). Fonte: Próprio autor.	92
A.43 Mapa de similaridade local entre mapa 8 e mapa de referência (similaridade global: 0,8036). Fonte: Próprio autor.	93
A.44 Mapa de similaridade local entre mapa 9 e mapa de referência (similaridade global: 0,9268). Fonte: Próprio autor.	94

LISTA DE TABELAS

3.1	Resolução e valores de limiar ocupado e livre para a construção dos mapas de ocupação. Fonte: Próprio autor.	25
3.2	Lista de tópicos do ROS e seu respectivo tipo de dado utilizados para o <i>dataset</i> do Laboratório de Robótica. Fonte: Próprio autor.	25
3.3	Lista de tópicos do ROS e seu respectivo tipo de dado utilizados para o <i>dataset</i> em ambiente virtual. Fonte: Próprio autor.	26
4.1	Tempo gasto para percorrer cada volta e o tempo total gasto em ambiente virtual. Fonte: Próprio autor.	32
4.2	Tempo gasto para percorrer cada volta e o tempo total gasto em ambiente Real. Fonte: Próprio autor.	39
4.3	Erro associado ao mapeamento. Fonte: Próprio autor.	44

INTRODUÇÃO

A Revolução Industrial, iniciada no século XVIII, culminou na introdução de equipamentos mecânicos no processo produtivo pois as máquinas foram capazes de aumentar a produtividade, diminuir o número de trabalhadores para exercer determinada atividade, reduzir os custos da produção e padronizar do processo de produção. Desde então, as máquinas passaram por aprimoramentos e as evoluções tecnológicas desses equipamentos permitiram que atividades antes feitas por mecanismos manuais ou semi automáticos fossem substituídas por máquinas em um processo automático. Um dos instrumentos advindos da evolução tecnológica foi o computador. A invenção do primeiro computador eletrônico em 1945, chamado de ENIAC, provocou uma revolução no processamento de cálculos matemáticos, capaz de mudar os desdobramentos da Segunda Guerra Mundial. O desenvolvimento tecnológico foi capaz de apresentar, em 1966, o primeiro robô terrestre móvel, o *Shakey*, e junto dele uma gama de estudos foi desenvolvida, como o algoritmo de planejamento de caminho A* [SRI 2020], um importante passo para o planejamento de rotas de deslocamento dos robôs. Os robôs terrestres móveis ganharam grande importância na nossa sociedade atual e podemos encontrá-los para além do campo industrial. Em nossa vida cotidiana os robôs terrestres móveis já estão incorporados, como é o caso do robô aspirador que executa tarefas domésticas como aspirar e limpar.

Com o crescente aumento de tecnologia aplicada ao desenvolvimento de robôs, eles vêm sendo largamente utilizados no setor industrial. Características como repetibilidade, padronização, precisão e colaboração são interessantes pois conferem rapidez aos processos de produção das indústrias sem causar estafa ao trabalhador. Mas uma característica especialmente importante é o fato de conseguirem preservar a vida dos trabalhadores em atividades que conferem riscos à vida humana ou algum tipo de dano leve ou dano permanente à saúde, sobretudo em atividades repetitivas ou em locais de elevado risco. O setor da construção civil e o de transporte e armazenamento concentram grande parte dos acidentes de trabalho com vítimas fatais. Podemos ver na Figura 1.1 que, nos Estados Unidos da América, durante o ano de 2018, esses setores apresentaram 1008 e 874 acidentes com vítimas fatais, respectivamente. Diante de números alarmantes, podemos pensar

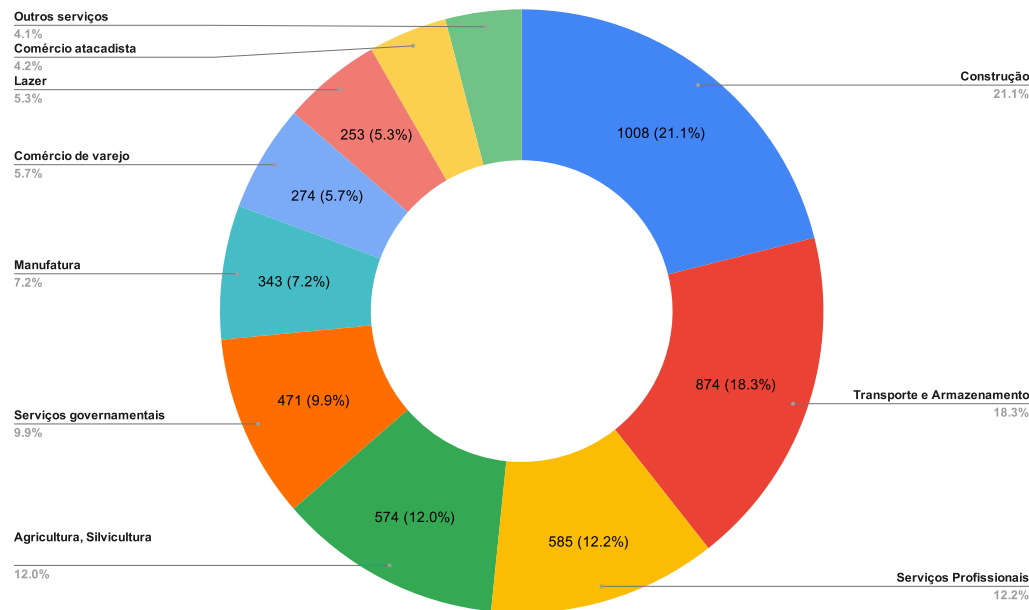


Figura 1.1: Vítimas de acidente fatal por setor da indústria nos EUA em 2018. Fonte: Adaptado de *US Bureau of Labor Statistics [Statistics 2018]*

que a utilização de robôs terrestres móveis tem potencial de colaborar com a redução de acidentes e criação de ambientes mais seguros aos trabalhadores.

A utilização dos robôs no setor industrial é tão versátil que tem sido ampliada de forma significativa nos últimos anos. A Figura 1.2 evidencia esses números em três continentes: Ásia/Austrália, Europa e América. Em 2018 existiam 414.000 unidades nesses continentes, sendo que a Ásia/Austrália concentravam 283.000 unidades, quase 70% da quantidade dos robôs mundiais.

Para que os robôs executem suas tarefas há o desenvolvimento de diversas tecnologias, que possibilitam que essas tarefas sejam executadas de forma a não acontecer erros, nem danos na estrutura física dos robôs. Para os robôs terrestres móveis, uma das principais tarefas consiste em o robô perceber o ambiente em que ele está inserido utilizando sensores. Estes sensores devem fornecer informações para o sistema computacional do sistema robótico, para que o robô possa compreender o ambiente no qual está inserido, sendo capaz de saber se há obstáculos à sua operação e, caso existam, a que distância se encontram esses obstáculos, o que garante uma perfeita utilização do robô.

Os sensores que capacitam o robô a perceber o ambiente externo são denominados sensores exteroceptivos. Estes sensores fornecem informações como distância, quantidade de luz ou amplitude do som. A Figura 1.3 apresenta três tipos de sensores exteroceptivos: uma câmera monocular, um sensor a *laser* e um sensor de ultrassom. Assim, estes sensores produzem informações que são interpretadas pelo sistema robótico através de equacionamentos matemáticos e, com isso, informações significantes como imagens podem ser produzidas. Outro tipo de sensor exteroceptivo que ganhou notoriedade ao longo

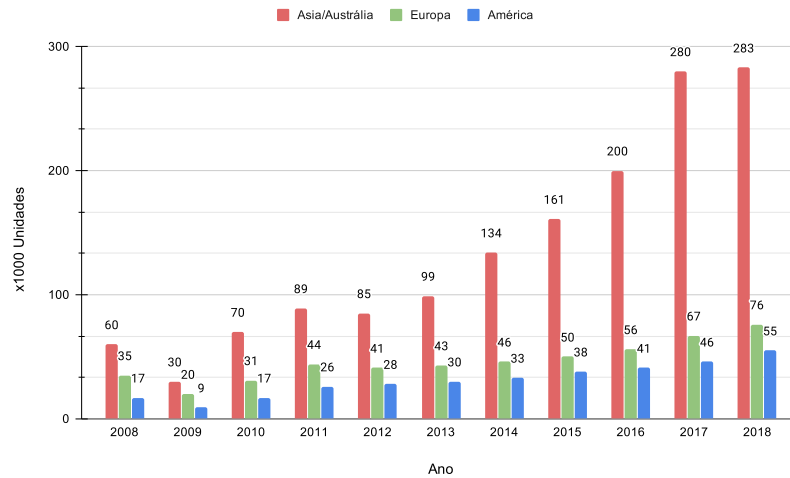


Figura 1.2: Utilização de robôs pela indústria em três continentes. Fonte: Adaptado de *International Federation of Robotics*(IFR)



(a) Câmera monocular SP-20000M-PMCL. Fonte: JAI



(b) Laser SICK S300 Mini. Fonte: SICK



(c) Sonar HC-SR04

Figura 1.3: Sensores tipo exteroceptivos

dos anos entre os pesquisadores devido ao seu baixo custo é o sensor do tipo RGB-D. Este sensor possui a capacidade de captar informações de luz e cor, como as câmeras monoculares, e também consegue captar informações de profundidade por meio de um sensor de infra-vermelho. A Figura 1.4 apresenta os principais sensores RGB-D, o sensor Microsoft Kinect, o sensor ASUS Xtion e o sensor Intel RealSense. Muitos trabalhos ao longo dos anos utilizando sensores RGB-D foram desenvolvidos [Munaro, Basso e Menegatti 2012], [Jafari, Mitzel e Leibe 2014], [Henry et al. 2014] e [Reis et al. 2019].

Apresentado em 2009, o sensor RGB-D Microsoft Kinect foi amplamente utilizado em diversos projetos em diferentes áreas como robótica móvel: [El-laithy, Huang e Yeh 2012], [Oliver et al. 2012], visão computacional: [Han et al. 2013] e [Izadi et al. 2011] e bioengenharia: [Alnowami et al. 2012], entre outros. Isto mostra a versatilidade deste tipo de sensor e ainda hoje estudos continuam sendo desenvolvidos por grupos de estudos em visão computacional, robótica e outros grupos. Os sensores RGB-D são compostos por uma câmera RGB e uma fonte de luz estruturada para a captação das distâncias [Villena-



Figura 1.4: Sensores tipo RGB-D.

Martínez et al. 2017]. A Figura 1.5 mostra a estrutura interna deste sensor.

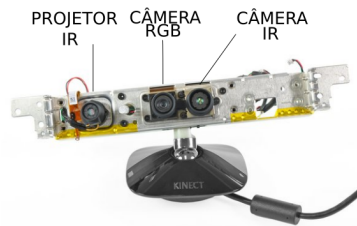


Figura 1.5: Estrutura interna do sensor RGB-D Microsoft Kinect. Adaptado de [TechCrunch 2010].

Outro tipo de sensor que ganhou notabilidade pela sua precisão e repetibilidade é o sensor a *laser* tipo LiDAR, do inglês *Light Detect And Range*, em português, Detectores de Distância por Luz. Este sensor foi e ainda é um dos sensores mais utilizados para a navegação autônoma de robôs terrestres [Grisetti, Stachniss e Burgard 2007], [?] e [Hess et al. 2016]. Hoje encontramos dois tipos de LiDAR os com detecção bidimensional(2D) e os com detecção tridimensional(3D). A Figura 1.6 abaixo mostra diferentes tipos de sensores 2D e 3D.

Com as informações oriundas dos sensores, cabe ao sistema robótico transformar estes dados primárias em dados significantes para reconhecimento de objetos, reconstrução de objetos, planejamento de trajetórias, produção de mapas, etc. A produção de mapas tem um papel relevante para a robótica terrestre móvel, para as principais técnicas de navegação autônoma é preciso que o robô tenha conhecimento do local em que está inserido, sabendo as delimitações do ambiente. Uma dessas formas de representar o ambiente é com mapas de ocupação bidimensionais. Esta modelagem foi primeiramente introduzida por [Elfes 1989], como forma de modelar o ambiente a partir de uma estimativa Bayesiana da leitura do sensor, se tornando um paradigma dominante para a robótica

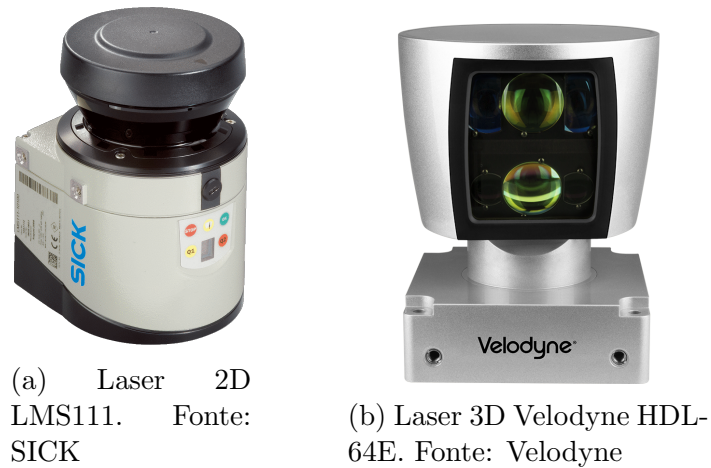


Figura 1.6: Sensor tipo LiDAR 2D E 3D

móvel. Uma vez adquirido, este mapa permite que várias tarefas sejam desenvolvidas como localização, planejamento de caminhos, desvio de obstáculos entre outros. [Thrun 2001].

Diversas aplicações em robótica móvel utilizam sensores do tipo LiDAR 2D. Esta escolha é feita principalmente devido ao custo deste sensor, que é mais baixo quando comparado a um sensor LiDAR 3D. Quando fisicamente acoplado a um robô, em geral, o sensor é estaticamente posicionado paralelo ao chão, assim as leituras são sempre realizadas na mesma altura e a única variável a ser lida é a distância captada pelo raio. Porém, dessa forma, objetos que estão posicionados acima ou abaixo da altura do sensor não são captados ou são incoerentemente captados. Objetos como mesas e cadeiras quando mapeadas e representadas em mapas de ocupação bidimensionais, ficam representadas geralmente apenas pelo seu suporte estrutural. A Figura 1.7 mostra como os raios do sensor LiDAR interceptam objetos como mesas e cadeiras

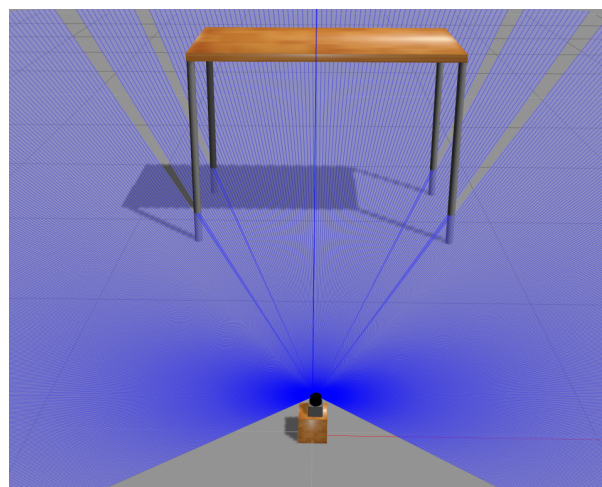


Figura 1.7: Intercepção do sensor LiDAR com os objetos. Fonte: Próprio autor.

Este problema poderia ser simplesmente resolvido ao se utilizar um sensor de mesma natureza porém com leitura tridimensional, porém o custo desse tipo de sensor muitas vezes inviabiliza sua aquisição. Para contornar este problema uma abordagem utilizando sensores RGB-D pode oferecer informações tridimensionais do ambiente por um baixo custo de aquisição.

Tendo em consideração a importância da utilização dos robôs na sociedade e a relevância de se produzir robôs que se desloquem de forma autônoma e segura, utilizando os avanços tecnológicos produzidos pelos sensores, esta dissertação aborda a utilização do *framework* ROS [Quigley et al. 2009] e GAZEBO [Koenig e Howard 2004] para a realização da atividade simulada e real de mapeamento *indoor* do Laboratório de Robótica da Universidade Federal da Bahia utilizando dois sensores de percepção de natureza distintas, um à *laser* (LiDAR) e um utilizando um sensor RGB-D fixados no robô terrestre móvel HUSKY A200 para representar objetos tridimensionais em mapas de navegação bidimensionais.

1.1 OBJETIVO

O objetivo principal deste trabalho é utilizar um robô terrestre móvel e os sensores de percepção RGB-D e LiDAR 2D para produzir um mapeamento *indoor* bidimensional que leve em consideração a tridimensionalidade e complexidade do ambiente e dos objetos.

1.1.1 Objetivos Específicos

Os objetivos específicos são:

- Analisar as principais características do método de mapeamento HECTOR SLAM utilizando um sensor LiDAR (*laser*).
- Analisar os principais características dos métodos de mapeamento RTAB-Map utilizando sensores tipo RGB-D e LiDAR(*laser*).
- Utilizar os métodos HECTOR SLAM e RTAB-Map no ROS (*Robot Operating System*).
- Realizar uma simulação computacional no GAZEBO para avaliar o mapeamento produzido por diferentes tipos de sensores utilizando uma abordagem por LiDAR e outra pela combinação de RGB-D e LiDAR.
- Realizar um experimento local utilizando um robô terrestre móvel para avaliar o mapeamento produzido por diferentes tipos de sensores utilizando uma abordagem por LiDAR e outra pela combinação de RGB-D e LiDAR.
- Gerar um *ground truth* baseado nas medidas reais do Laboratório de Robótica para avaliar a relação entre as medidas métricas dos mapas.
- Avaliar as diferenças entre os mapas reais produzidos utilizando o método de SLAM RTAB-Map utilizando a métrica Índice de Similaridade Estrutural, do inglês (*structural similarity index measure*) .

1.2 JUSTIFICATIVA

A navegação autônoma de robôs terrestres móveis tem sido alvo dos pesquisadores por várias décadas, sendo este um tópico exclusivo de pesquisa dentro todos os tópicos que envolvem a robótica móvel. Muitos trabalhos foram apresentados à sociedade acadêmica na tentativa de solucionar os problemas inerentes à este tópico. Muitos destes trabalhos foram desenvolvidos utilizando sensores tipo LiDAR 2D fixados ao robô, captando apenas as informações que estão no seu raio de atuação, e não a complexidade do dos objetos contidos no ambiente. Assim, certos objetos podem não ser representados devido à sua geometria. A incorporação de um sensor RGB-D em um robô terrestre móvel permite que uma grande quantidade de dados possa ser incorporado devido ao sensor RGB. Além disso, o sensor de profundidade presente no sensor RGB-D permite uma compreensão tridimensional sobre a profundidade vista pelo sensor, permitindo assim a representação de objetos 3D, desta forma os objetos que antes eram complexos para a câmrea RGB-D tornam-se menos complexos.

A utilização do sensor RGB-D faz-se uma opção uma vez que sensores tipo LiDAR 3D, possuem um alto valor agregado, o que o torna inviável em muitos projetos. Na contramão do custo do sensor LiDAR 3D, sensores RGB-D possuem um baixo valor agregado o que permite ser aplicado em grandes e também pequenos projetos. Assim, o problema da compreensão da complexidade tridimensional do ambiente é resolvida, permitindo que mapas de ocupação bidimensionais, como os produzidos pelos sensores LiDAR 2D, possam ser produzidos com informações mais precisas sobre o ambiente. Assim, aplicações de técnicas de planejamento de caminhos e evasão de obstáculos podem ser melhor executadas, garantindo mais segurança, qualidade e otimização de recursos do sistema robótico.

1.3 ESTRUTURA DA DISSERTAÇÃO

A dissertação está estruturada da seguinte maneira:

- O capítulo 2 aborda uma revisão bibliográfica sobre os principais métodos de mapeamento e localização contidos na literatura e também uma análise das características do métodos implementados no ROS. São apresentados também os métodos de mapeamento HECTOR SLAM e RTAB-Map. Ao fim do capítulo, métricas de comparação de imagens são apresentadas.
- O capítulo 3 apresenta a metodologia utilizada para a geração dos mapas e para a geração do *ground-truth*.
- O capítulo 4 apresenta os resultados obtidos através da metodologia aplicada e traz uma discussão dos dados obtidos.
- A conclusão e considerações finais são apresentadas no capítulo 5.

REVISÃO BIBLIOGRÁFICA

2.1 DEFINIÇÃO MATEMÁTICA E REVISÃO HISTÓRICA

Sistemas de mapeamento e localização simultâneos (SLAM) são sistemas computacionais equipados com sensores que respondem ao problema de se posicionar um robô móvel com uma postura arbitrária em ambiente desconhecido e, incrementalmente, construir um modelo geométrico do ambiente e se localizar dentro desse modelo.

A necessidade de construir um mapa do ambiente é dupla. A primeira necessidade reside no suporte de tarefas como planejamento de rotas seguras e para uma intuitiva teleoperação do robô. A segunda necessidade está em reduzir o erro de estimação de postura, evitando o conhecido erro por *dead reckoning*. Outra vantagem de se utilizar um mapa com marcos bem definidos é que o robô pode redefinir sua localização revisitando áreas conhecidas através da técnica de fechamento de *loop*.

A popularidade do SLAM está intimamente ligada ao fato de que em ambientes internos o sistema de localização global (GPS) não pode ser acessado pela dificuldade de comunicação entre o sistema computacional e a constelação de satélites. Assim, a técnica de mapeamento e localização simultânea emergiu como uma solução para a dificuldade do desenvolvimento de tarefas para robôs autônomos.

Matematicamente, o SLAM é definido utilizando o Teorema da probabilidade condicional com eventos independentes (Teorema de Bayes) 2.1, sejam os eventos A, B e E independentes entre si. A probabilidade do evento A dadas as informações dos eventos B e E é dada por:

$$P(A|B, E) = \frac{P(B|A, E)P(A|E)}{P(B|E)} \quad e \quad P(B|E) > 0 \quad (2.1)$$

$$Probabilidade = \frac{(hipótese).(priori)}{(evidência)}$$

Assim o problema de SLAM é entendido como a estimação do estado $\mathbf{x}_k$ e o conjunto de marcos $\mathbf{m}_k$ por meio de uma função de densidade de probabilidade *a priori* em cada

instante k e é corrigida por meio de uma distribuição de probabilidade condicional *a posteriori*, tendo em mão que todos os sinais de entrada $\mathbf{u}_k$ e as observações $\mathbf{z}_k$ são conhecidas. A equação 2.2 define esta distribuição de probabilidade que realiza a predição e a equação 2.3 realiza a correção da predição [Durrant-Whyte e Bailey 2006].

$$P(x_k, m_k | z_0, \dots, z_{k-1}, u_0, \dots, u_k, x_0) = \int P(x_k | x_{k-1}, u_k) \times P(x_{k-1}, m_k | z_0, \dots, z_{k-1}, u_0, \dots, u_{k-1}, x_0) dx_{k-1} \quad (2.2)$$

$$P(x_k, m_k | z_0, \dots, z_k, u_0, \dots, u_k, x_0) = \frac{P(z_k | x_k, m_k) \times P(x_k, m_k | z_0, \dots, z_{k-1}, u_0, \dots, u_k, x_0)}{P(z_k | z_0, \dots, z_{k-1}, u_0, \dots, u_k)} \quad (2.3)$$

Um modelo para descrever a probabilidade de se fazer uma observação $\mathbf{z}_k$ quando a posição e localização do marco é conhecida geralmente descrita pela equação 2.4:

$$P(z_k | x_k, m_k) \quad (2.4)$$

Assumindo que o movimento do robô é um processo Markoviano, a transição do estado atual x_{k-1} para o próximo estado x_k depende apenas do sinal de controle u_k aplicado sobre o robô. Desta maneira, o movimento do robô pode ser descrito utilizando a distribuição de probabilidade que é independente das observações e do mapa.

$$P(x_k | x_{k-1}, u_k) \quad (2.5)$$

O mapa é estimado também por meio de uma função de densidade de probabilidade condicional, presumindo que todos os x_k estados foram estimados para todos os k instantes de tempo. O problema de se determinar um mapa é dado pela equação 2.6.

$$P(m_k | x_0, \dots, x_k, u_1, \dots, u_k, z_0, \dots, z_k) \quad (2.6)$$

De modo inverso, conhecendo-se o mapa, a localização dos marcos e realizando observações sucessivas, o problema de localização é determinado usando a distribuição condicional 2.7. No SLAM, as correlações entre as estimativas dos marcos aumentam monotonicamente, assim, a estimativa da posição dos marcos fica sempre melhor independentemente dos movimentos que a plataforma robótica irá realizar [Santana 2011]. O SLAM sempre se apresentou como uma grande promessa para o futuro da robótica móvel e muitos trabalhos ainda buscam encontrar soluções para este problema.

$$P(x_k | z_0, \dots, z_k, m, u_0, \dots, u_k) \quad (2.7)$$

A estrutura do SLAM foi apresentada pela primeira vez no *International Symposium on Robotics* em 1995 [Durrant-Whyte, Rye e Nebot 1996] e em 1997 a teoria matemática formal do SLAM foi pela primeira vez apresentada e sua grande contribuição foi de não

se ignorar a relação entre os erros do veículo e a estimação do mapa. Teoricamente, é possível inserir um robô em um local e posição desconhecidos e este realizar a localização e mapeamento simultâneo. Entretanto, demonstrou-se também que uma grande capacidade de armazenamento de dados e de carga computacional era necessária e, com isso, o problema ainda estaria em aberto para futuras pesquisas [Csorba 1997]. A Figura 2.1 exemplifica a essência do problema, onde um robô tenta se localizar a partir de observações feitas no ambiente e marcos são utilizados para realizar a estimação da posição do robô.

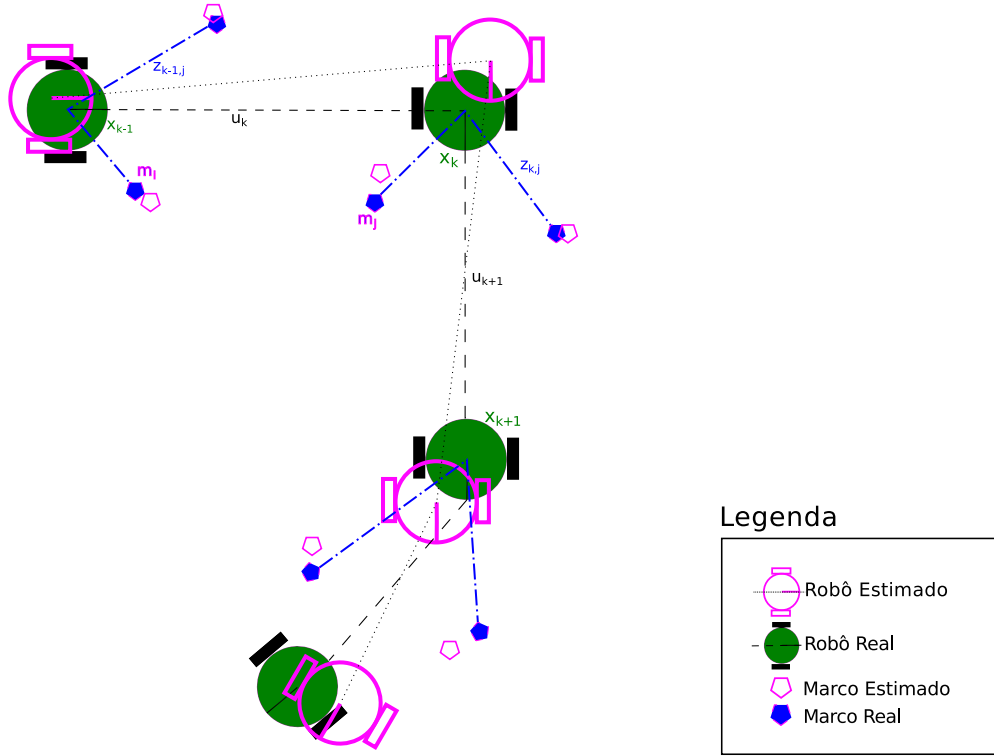


Figura 2.1: Estimação da posição e orientação de um robô baseada na observação de marcos no ambiente. Fonte: Próprio autor.

Onde:

- $\mathbf{x}_k$:representa o vetor de estado contendo a posição e orientação do robô.
- $\mathbf{u}_k$:representa o vetor contendo os comandos de deslocamento do robô de $k-1$ ate o estado $\mathbf{x}_k$ no tempo k .
- $\mathbf{m}_i$:representa o vetor contendo a posição do i -ésimo marco, sendo os marcos fixos e invariantes no tempo.
- $\mathbf{z}_k$:representa o vetor de estado contendo as informações sobre as observações do ambiente.

Nos anos seguintes, o principal avanço no tema foi alcançado quando a convergência entre o método de SLAM baseado em filtro de Kalman e a localização e mapeamento

probabilístico proposto foi atingido [Thrun, Burgard e Fox 1998] . O primeiro grande avanço foi contemplado por [Dissanayake et al. 2001] e consiste na utilização de um filtro de Kalman estendido (EKF-SLAM). Os resultados produzidos por [Montemerlo et al. 2002] (FastSLAM 1.0) impulsionaram ainda mais os avanços ao utilizarem um filtro de partículas numa aproximação ideal ao filtro recursivo de Bayes. O espaço é particionado e partículas representam a posição e orientação do robô e os marcos são estimados analiticamente por um filtro de Kalman. Este particionamento do espaço faz com que o FastSLAM seja um filtro de partículas do tipo Rao-Blackwell [Bailey, Nieto e Nebot 2006]. Uma atualização do FastSLAM 1.0 [Montemerlo et al. 2003] foi feita de modo que a estimação da distribuição de probabilidade do robô fosse uma solução mais eficiente. Muitas soluções novas e híbridas do FastSLAM 1.0 e do FastSLAM 2.0 foram publicadas por pesquisadores como Sebastian Thrun, John Leonard, Mathew Walter, Wolfram Burgard, Cyrill Stachniss, Andrew Davison, Hugh Durrant-Whyte, Simon Lacroix e Young-Ho Choi, cada uma trazendo inovações na forma de executar o SLAM ou de estimar os estados do robô de forma bidimensional(2D) e também tridimensional(3D).

Avançando sobre a questão da dimensionalidade, realizar o mapeamento e localização simultâneo para um ambiente tridimensional seria, a princípio, apenas estender as funcionalidades do SLAM bidimensional para o ambiente tridimensional. Entretanto, realizar o SLAM em 3D adiciona uma larga complexidade por se determinar o modelo probabilístico do robô, aumentando a dificuldade de detecção e modelagem dos recursos. Essencialmente existem três formas de realizar o SLAM-3D. A primeira forma é o uso de *laser-2D* para a realização de mapas tridimensionais como feitos por [Thrun, Burgard e Fox 2000] e [Fruh e Zakhor 2001]. A segunda forma é uma extensão do SLAM 2D para o 3D com a extração discreta de características dos marcos e estimação de funções de densidade condicional do mapa e veículo. Esta técnica foi implementada utilizando uma câmera monocular [Davison, Cid e Kita 2004]. A terceira técnica consiste numa formulação diferente das outras anteriores, onde os estados são compostos pelo histórico das posições e orientações do robô, e em cada posição e orientação o robô terrestre realiza uma varredura tridimensional do ambiente e as posições e orientações são alinhadas pelas correlações entre as varreduras. Estas formulações foram implementadas para ambientes abertos como feitos por [Eustice et al. 2005] [Newman, Cole e Ho 2006].

Soluções de SLAM que utilizam câmeras são denominadas como vSLAM, onde a letra “v” se refere à palavra visual, assim do inglês: *visual Simultaneous Localization and Mapping*. Soluções de vSLAM são, em geral, soluções construídas com o auxílio de grafos (*graph based SLAM*). Esta estrutura foi pela primeira vez apresentada por [Lu e Milios 1997], entretanto levou-se anos para que ela tornar-se factível devido à sua alta complexidade computacional para o problema de minimização do erro. Grafos são representações geométricas gráficas que possuem apenas nós e arestas. A Figura 2.2 apresenta uma simples representação de um grafo planar com 4 nós e 6 arestas. Para o caso do vSLAM, os nós 1,2,3 e 4 representam:

- O tempo t de criação do nó.
- Posição e orientação x_t no mapa.

- Posição e orientação da odometria x_t^{odom} .
- Conjunto de características (*features*) visuais extraídas da câmera no tempo t .
- Conjunto de arestas que conectam este nó a outros nós.

As arestas $a_1, \dots, a_6$ armazenam as transformações entre *features* de diferentes nós bem como as transformações necessárias entre o conjunto de referenciais (*frames*) além de:

- Transformações entre os a_i nós.

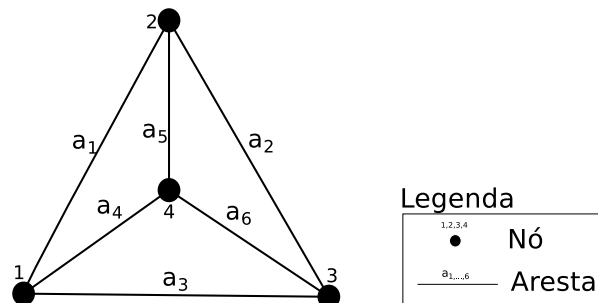


Figura 2.2: Representação de um grafo. Fonte: Próprio autor.

O vSLAM baseado em grafo é construído à medida que o robô se move através do ambiente, assim é dito que este modelo de vSLAM é *online*. O grafo é construído iterativamente por meio de um algoritmo dividido em duas partes: *front-end* e *back-end*. O *front-end* tem por característica trabalhar diretamente sobre os dados oriundos dos sensores empregados no robô, extrair dados relativos a estes sensores, realizar inferências sobre estes dados e impor restrições. Enquanto o *back-end* trabalha na otimização do grafo, minimizando os erros de posição e orientação introduzidos pelos sensores. A Figura 2.3 mostra as entradas, etapas e saída para sistemas SLAM baseado em grafos.

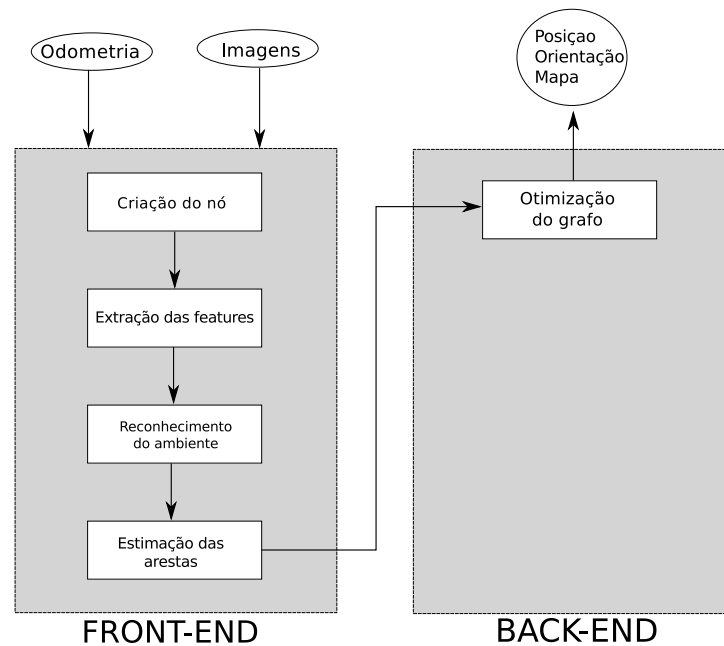


Figura 2.3: Fluxo de dados para o vSLAM baseado em grafos. Adaptado de [Frost 2016]

A solução baseada em grafos foi utilizada por diversos pesquisadores que implementaram esta técnica, não somente para o caso do SLAM 3D, mas também para o caso bidimensional [Gutmann e Konolige 1999], [Howard, Mataric e Sukhatme 2001], [Koenig e Howard 2004] entre outros. Entretanto, muitas destas soluções não são disponibilizadas em código aberto e livre, o que dificulta sua compreensão, teste e comparação. Desde sua criação o *framework* ROS (*Robot Operating System*) [Quigley et al. 2009] vem ganhando mais adeptos devido à sua simplicidade, abstração e disponibilidade de bibliotecas e *drivers* para o desenvolvimento de aplicações para robótica.

O ROS é um *framework* de código aberto baseado em Linux. Foi originalmente desenvolvido pela Willow Garagage e atualmente é mantido pela *Open Source Robotics Foundation*. Sua principal característica é realizar a comunicação entre o *hardware* e *software* por meio de bibliotecas que podem ser obtidas livremente e também modificadas pelos usuários, desta maneira é possível realizar uma customização para uma dada tarefa específica a ser realizada por um componente robótico como planejamento de trajetórias, reconhecimento de objetos, *pick and place*, etc. Sua estrutura de arquivos é dividida em:

- *Packages* - unidade de organização de *software*
- *Metapackages* - pacotes especializados para representar outros pacotes
- *Package Manifest* - arquivos do tipo XML que fornecem metadados sobre os pacotes
- *Message types* - Descrição da estrutura do tipo dado
- *Service types* - Descrição das solicitações de serviços e estrutura do dados de serviço

Ao longo dos anos, sua utilização cresceu em número de usuários chegando em 2017 a um total de 13,4 milhões de usuários. A Figura 2.4 mostra a evolução da utilização do *framework*. Um *plugin* muito utilizado para simulação é o GAZEBO.

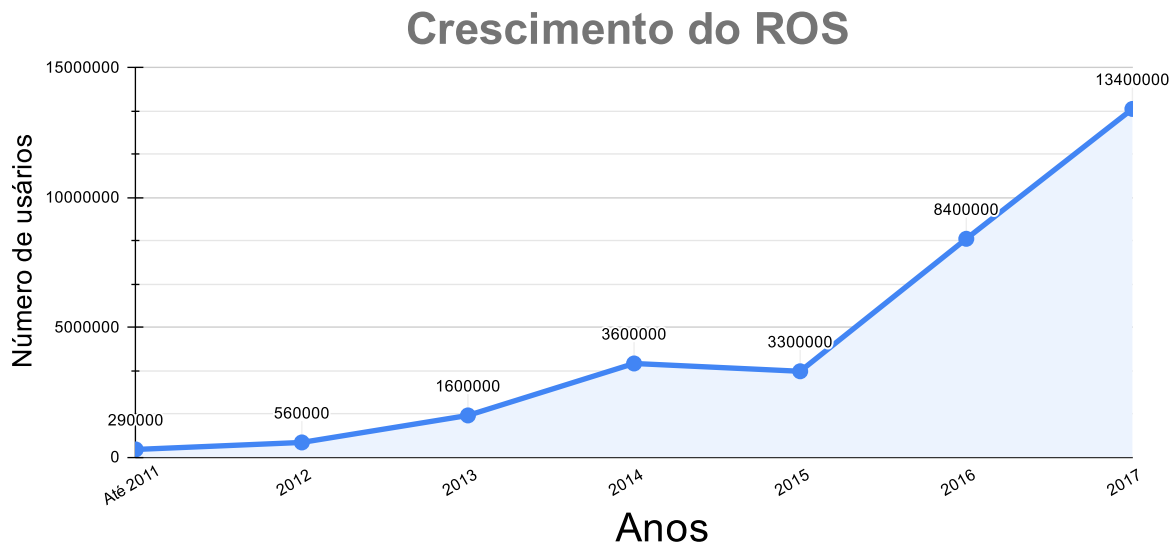


Figura 2.4: Usuários do ROS ate 2017. Adaptado de [Zhang et al. 2017]

O GAZEBO, é um ambiente de simulação 3D que permite criação de ambientes para desenvolvimento de estratégias de operação de robôs e sensores, permitindo replicar condições ambientais como: gravidade, sol, chuva, neblina, vento, etc. Em sua estrutura básica, o simulador provê uma série de sensores como câmeras RGB e RGB-D, LiDAR a *laser*, altímetro, IMU, etc. Possui também outros modelos 3D como: veículos, casas, robôs, etc e também permite a inserção de modelos 3D customizados. Este ambiente de simulação faz parte do pacote básico de *software* já incluídos no ROS, ou seja, uma vez que o ROS é instalado, o GAZEBO estará prontamente instalado.

2.2 SOLUÇÕES POPULARES E DISPONÍVEIS PARA O ROS

Devido à sua popularidade, atualmente existem muitas aplicações e variações de soluções para o SLAM e vSLAM. Nesta subseção serão apresentadas algumas das implementações mais populares analisando suas características, entradas e saídas. Primeiro serão apresentadas as soluções envolvendo a tecnologia LiDAR e posteriormente soluções envolvendo câmeras: Monoculares, *Stereo* e RGB-D.

2.2.1 Sistemas baseados em LiDAR

O trabalho proposto por [Grisetti, Stachniss e Burgard 2007], conhecido como GMapping, é o SLAM padrão pré-instalado no ROS e tem sido amplamente utilizado para a geração

de mapas de ocupação bidimensionais usando sensores a *laser* 2D. Esta abordagem requer uma fonte de odometria e utiliza um filtro de partículas para estimar a trajetória do robô, assim como no trabalho de [Steux e Hamzaoui 2010], o tinySLAM.

O HECTOR SLAM [Kohlbrecher et al. 2011], cria de forma rápida e com baixo custo computacional um mapa de ocupação usando um *laser* 2D. Apesar de não ser um SLAM completo por não detectar o fechamento de *loop*, apresenta um baixo erro de deslocamento. Esta implementação não exige o uso de uma fonte externa de odometria, o que pode ser uma vantagem quando o robô não possui esta capacidade. Entretanto, não utilizar uma fonte externa de odometria pode ser uma desvantagem quando operado em um ambiente com restrições geométricas, como acontece com largos corredores, limitando a capacidade de performance.

Implementações baseadas em grafos são apresentadas nos pacotes KartoSLAM [Vincent, Limketkai e Eriksen 2010], LagoSLAM [Carlone et al. 2012] e Google Cartographer [Hess et al. 2016]. Nestes casos, enquanto a atividade de mapeamento acontece, sub mapas são criados e ligados através de restrições por meio de um grafo. Quando um *loop* é detectado, a posição dos sub mapas é novamente otimizada para corrigir os erros introduzidos por ruídos.

Utilizando *lasers* 3D, a principal abordagem é proposta por [Dubé et al. 2017], o SegMatch. Nesta abordagem, os autores usam um LiDAR 3D para detecção de *loop* em tempo real de processamento. O fechamento do *loop* é detectado quando partes da aquisição atual são encontradas na nuvem de pontos acumulada anteriormente.

2.2.2 Sistemas baseados em câmeras RGB, Stereo e RGB-D

Utilizando câmeras monoculares RGB, *stereo* ou RGB-D, utilizam-se técnicas para estimação das posturas de uma câmera em um ambiente tridimensional. Este processo é conhecido como odometria visual [Scaramuzza e Fraundorfer 2011] e [Fraundorfer e Scaramuzza 2012]. Neste contexto, serão apresentadas implementações disponíveis para o *framework* ROS que utilizam câmeras.

O Maplab [Schneider et al. 2018], é um vSLAM baseado em grafo usando somente uma câmera e uma unidade de medição inercial (IMU). Com isto, os autores adquirem nuvens de pontos capazes de construir mapas sub-mapas que são fundidos utilizando um pré alinhamento baseado no fechamento de *loops*.

A abordagem baseada em grafos feita por [Endres et al. 2013], implementada sob o nome de RGBD-SLAMv2, é feita através da extração de *features*. Os *key points* de cada imagem são extraídos e localizados na imagem de profundidade. O algoritmo RANSAC [Fischler e Bolles 1981] é utilizado para realizar a associação dos *key points* e o grafo usando uma otimização não linear. Com isso, um mapa 3D é gerado.

Por muitos anos, o ORB-SLAM2 [Mur-Artal e Tardós 2017] e o S-PTAM [Pire et al. 2017] foram os sistemas mais utilizados pelos pesquisadores para mapeamento e localização. A técnica ORB-SLAM2 permite o uso de câmeras RGB-D destacando-se das demais técnicas em função do baixo custo computacional. Nesta técnica, a detecção do *loop* é feita usando a abordagem proposta por [Gálvez-López e Tardós 2012] na qual

uma otimização é feita utilizando o *bundle adjustment*. Entretanto, a medida que o mapa aumenta, o tempo de processamento da otimização do grafo também aumenta, o que faz com que a correção do erro não aconteça imediatamente após a detecção.

Desenvolvido por [Labbe e Michaud 2019], o RTAB-Map, é um sistema de vSLAM baseado em grafos e com um módulo de gerenciamento de memória [Labbe e Michaud 2013] fazendo com que o grafo sempre tenha um tamanho fixo e, com isso, podendo ser aplicado em ambientes de larga escala. Esta abordagem tem duas características principais: a possibilidade de uso de uma fonte externa de odometria e o conjunto de entradas e saídas fornecidas. Como entrada o sistema admite o uso de câmeras: RGB-D, *stereo* e multicâmeras RGB-D. Fornece como saída a postura da câmera, um mapa de ocupação 2D e 3D e também uma nuvem de pontos densa. Ao apresentar um mapa de ocupação 2D como os produzidos pelos sistemas baseados em LiDAR, este faz-se factível para ser embarcado em um robô terrestre móvel.

Analisando os métodos de localização e mapeamento acima citados, escolheu-se utilizar o método de mapeamento por LiDAR 2D o método HECTOR SLAM por ser um método já utilizado em outras atividades onde mostrou-se ser um método confiável quando se possui somente o sensor LiDAR 2D para mapeamento, fornecendo uma mapa de ocupação bidimensional. Para a utilização do sensor RGB-D optou-se por utilizar o método RTAB-MAP por ser o único método de vSLAM que fornece um mapa de ocupação bidimensional do ambiente como os mapas produzidos pelos métodos por LiDAR. Nas sessões seguintes, será feita uma breve apresentação teórica dos sistemas HECTOR SLAM e RTAB-Map respectivamente, apresentando suas características e estrutura teórica.

2.3 HECTOR SLAM

Apresentado em 2011 no *IEEE International Symposium on Safety, Security, and Rescue Robotics* o HECTOR SLAM [Kohlbrecher et al. 2011] é um algoritmo de código aberto disponível para o ROS para localização, mapeamento, navegação exploratória autônoma e também rastreamento de objetos (*object tracking*). A Figura 2.5 ilustra como a organização do método é feita. Os retângulos representam os nós, os tópicos são representados por flechas e os serviços são representados pelos diamantes.

O *scan matching*, técnica que alinha os dados oriundos do sensor LiDAR, é baseado em uma aproximação Gauss-Newton que tem como objetivo encontrar uma transformação rígida, $\vec{\zeta}$, que ajuste novos dados e os dados existentes no mapa. Esta transformação é dada pela equação 2.8, onde M é o mapa e $\vec{S}_i(\vec{\zeta})$ é o conjunto de coordenadas dos *beam end-points* do LiDAR. A covariância é dada por: $R = var\vec{\zeta}$ [Koch 2018].

$$\vec{\zeta}_* = \underset{\vec{\zeta}}{argmin} \sum_{i=1}^n [1 - M(\vec{S}_i(\vec{\zeta}))]^2 \quad (2.8)$$

Na figura acima, o processo de mapeamento se inicia no bloco *MAPPING* com o fornecimento dos dados do sensor LiDAR por meio de um tópico do ROS. Este dado é do tipo `\sensor_msgs\LaserScan.msg` e também é necessário fornecer a TF(*transformation library*) referente ao posicionamento do sensor em relação á base do robô. Com apenas estes dados iniciais o processo de mapeamento é iniciado. E como saída, um mapa de

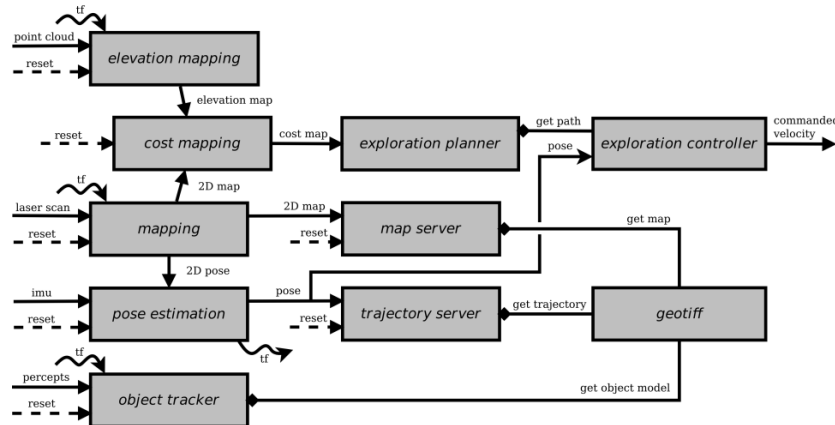


Figura 2.5: Representação em diagrama de blocos do HECTOR SLAM no ROS. Fonte: [Kohlbrecher et al. 2013]

ocupação é fornecido para o bloco *MAP SERVER*. Apesar de solicitar como entrada os dados de leitura do sensor LiDAR, um problema pode acontecer devido a não utilização de uma fonte de odometria. Ambientes sem características a serem extraídas como um longo corredor paralelo, poderão causar uma falha no mapeamento pois os dados coletados serão quase idênticos, e os dados sendo idênticos, não haverá uma atualização da posição bem como uma atualização do mapa, causando uma distorção.

2.4 RTAB-MAP

O pacote RTAB-Map, *Real-Time Appearance-Based Mapping*, é um método de SLAM baseado em grafos disponibilizado para o ROS em código aberto como *rtabmap-ros*. Este pacote é implementado com o fechamento de *loop* e também com uma abordagem com gerenciamento de memória que limita o tamanho do mapa, o que faz que as detecções de *loop* sejam executadas sempre em um tempo fixo. Após a sincronização dos sensores um módulo de memória de curto prazo (*ShortTerm Memory*-STM) cria-se um nó e armazena-se informações da pose da odometria, dados dos sensores e informações adicionais para outros módulos como exemplo o mapa de ocupação local. A Figura 2.6 ilustra a representação do pacote. As entradas requeridas são: TF para definições de posições do sensor em relação à base do robô, uma fonte qualquer de odometria, uma das opções de imagem (RGB-D ou *Stereo*). Opcionalmente, pode-se enviar dados de sensor LiDAR 2D e 3D (nuvem de pontos). Como saída, o pacote disponibiliza: dados do mapa contendo os últimos nós adicionados, um grafo representando o caminho do robô, odometria corrigida publicada em forma de TF e opcionalmente um mapa 3D (somente quando há uma nuvem de pontos como entrada) ou um mapa de ocupação 2D.

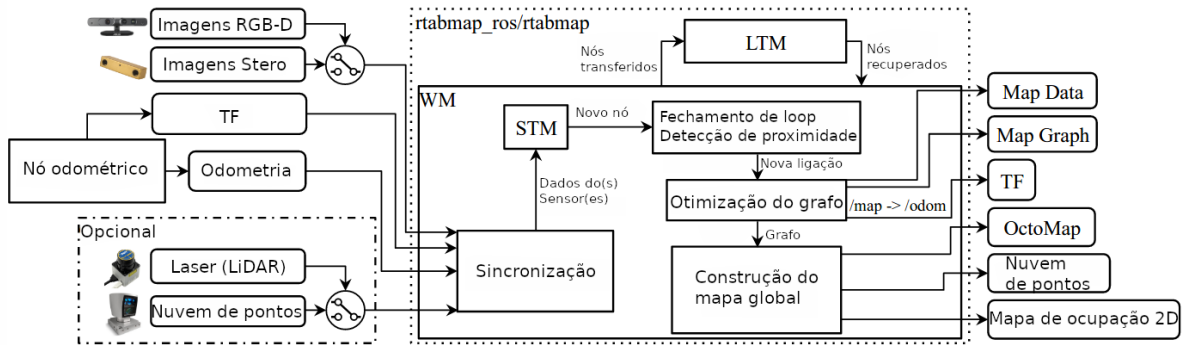
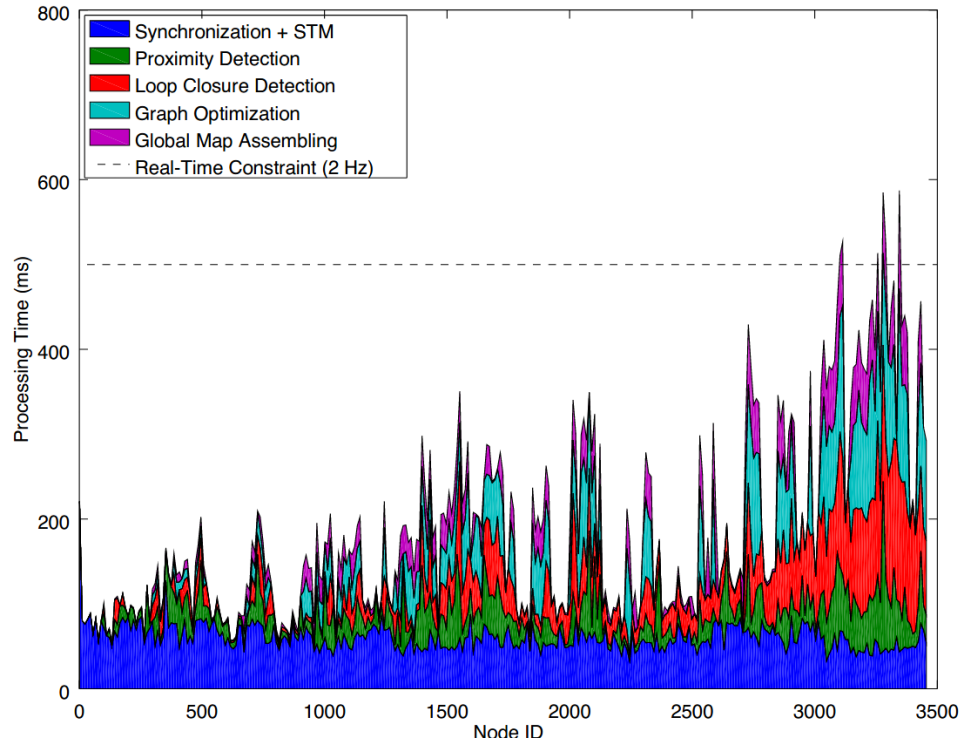
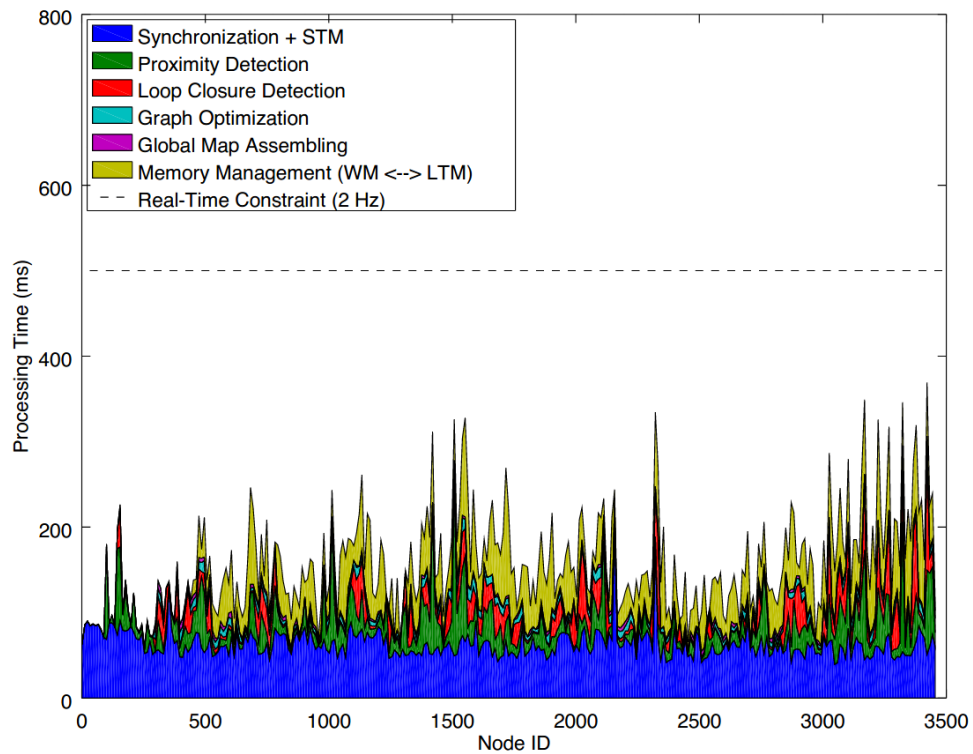


Figura 2.6: Diagrama de blocos do RTAB-Map. Adaptado de [Labbé e Michaud 2019].

O gerenciamento de memória é o ponto crucial deste método de SLAM, pois uma vez que o grafo cresce com a adição de novos nós, o tempo de processamento de seus módulos pode extrapolar as restrições de para operação em tempo real. Assim o gerenciamento de memória foi dividido em três módulos: *ShortTerm Memory*-STM, *Working Memory*-WM e *Long Term Memory* - LTM. A Figura 2.7 mostra como a implementação destes módulos influenciou o tempo de processamento quando testado no MIT *Stata Center*. Uma vez que um nó é criado pela STM, na STM informações da imagem RGB são extraídas e quantizadas em um vocabulário de palavras. Os nós são removido da STM e enviados para a WM em uma taxa fixa. Na WM o novo nó é comparado com outros nós existentes dentro da WM e processos como reconhecimento do ambiente para fechamento de *loop* e otimização do grafo são executados. De lá os nós podem ser descartados ou enviados para a LTM por meio de uma heurística que avalia a importância daquele nó [Labbé e Michaud 2019].



(a) Sem gerenciamento de memória.



(b) Com gerenciamento de memória.

Figura 2.7: Tempo de processamento de cada módulo interno do nó *rtabmap_ros* sem o gerenciamento de memória e com o gerenciamento de memória para um mapa com taxa de atualização de 2Hz usando sessões combinadas do MIT *Stata Center*. Fonte: [Labbé e Michaud 2019]

2.5 MÉTRICAS DE SIMILARIDADE PARA IMAGENS

As métricas de similaridade de imagem têm como objetivo comparar uma imagem de referência com uma outra imagem que foi digitalmente processada ou reconstruída. Esta comparação tenta traduzir de forma sistemática o que o sistema visual humano facilmente faz quando observa um grupo de imagens e decide qual delas é a mais similar à imagem se referencia. As métricas mais comumente utilizadas são: MSE - Erro Quadrático Médio (*Mean Square Error*), PSNR - Relação Sinal-Ruído de Pico (*Peak Signal-to-Noise Ratio*) e SSIM - Índice de Similaridade Estrutural (*Structural Similarity Index*).

A técnica de avaliação mais simples de ser compreendida e implementada é a MSE - Erro quadrático Médio. A equação 2.9 apresenta a definição matemática para esta estimativa comparando duas imagens X e Y , onde os valores M e N , respectivamente representam as dimensões da imagem e as variáveis $X_{i,j}$ e $Y_{i,j}$ representam o *pixel* relativo a cada imagem. Esta representação compara a diferença de cada pixel entre a imagem de referência e a imagem a ser analisada. Esta métrica não avalia condições de rotação e translação, permitindo uma interpretação falha do resultado obtido.

$$MSE(X, Y) = \frac{1}{NM} \sum_{i=1}^N \sum_{k=1}^M (X_{i,j} - Y_{i,j})^2 \quad (2.9)$$

A métrica PSNR - Relação Sinal-Ruído de Pico, é uma métrica mais avançada utilizada em trabalhos que envolvem compressão de imagens e vídeo como [Villena et al. 2013] e [Babacan, Molina e Katsaggelos 2010]. Esta métrica é definida de acordo com a equação 2.10. Apesar de ser simples de se calcular, não trás um relacionamento com a percepção humana. Apesar de ser possível obter altos valores de PSNR, este alto índice não garante que ao se visualizar a imagem comparada haverá uma real identificação entre as duas [Salomon 2004]. Onde MAX se refere ao valor máximo de um pixel, por exemplo: em uma imagem em escala de cinza de 8-bits, MAX assume o valor de 255.

$$PSNR(X, Y) = 10 \cdot \log \frac{MAX^2}{MSE} \quad (2.10)$$

Trazendo uma relação mais próxima do sistema visual humano, a técnica SSIM - Índice de Similaridade Estrutural [Wang et al. 2004] desenvolve uma forma de análise de similaridade estrutural comparando padrões locais de luminância pelo contraste. A Figura 2.8 representa o funcionamento do sistema de comparação. O SSIM é matematicamente definido de acordo com a equação 2.11.

$$SSIM(X, Y) = \frac{(2\mu_x\mu_y + c1)(2\sigma_{xy} + c2)}{(\mu_x^2 + \mu_y^2 + c1)(\sigma_x^2 + \sigma_y^2 + c2)} \quad (2.11)$$

Onde:

- μ_x : média de X

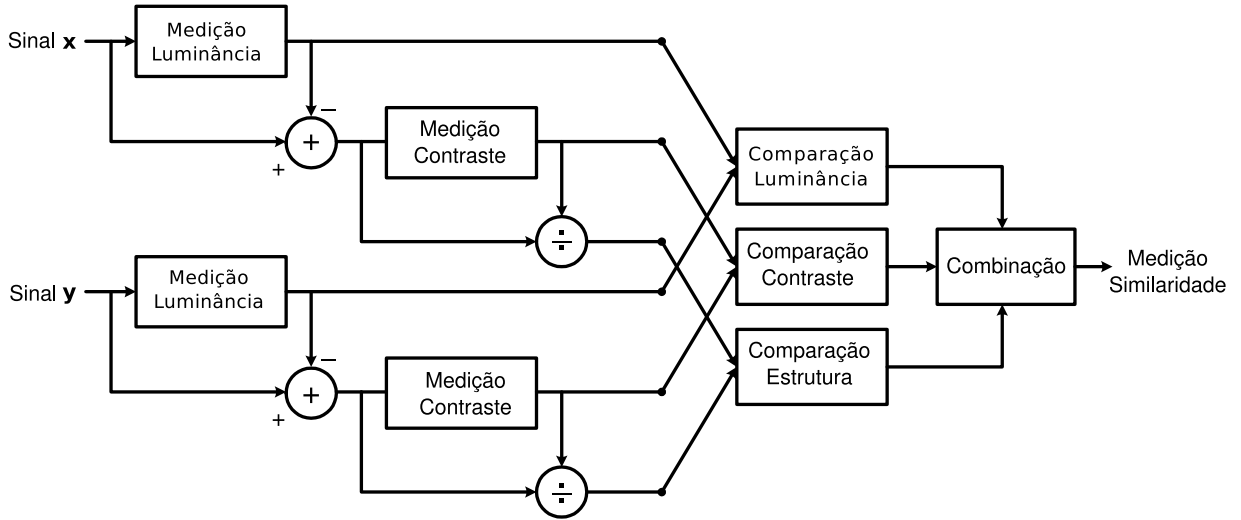


Figura 2.8: Diagrama do sistema de similaridade estrutural (SSIM). Adaptado de [Wang et al. 2004].

- μ_y : média de Y
- σ_x : variância de X
- σ_y : variância de Y
- σ_{xy} : covariância entre X e Y
- $c1$ e $c2$: constantes que estabilizam a divisão

Utilizando o SSIM deseja-se observar um mapa que seja considerado o melhor de todos entre os mapas produzidos utilizando o conjunto de sensores LiDAR e RGB-D utilizando o RTAB-Map. Assim, compara-se os mapas atuais fornecidos pelas voltas, afim de analisar numericamente estes mapas em relação ao melhor mapa alcançado, obtendo um nível de similaridade. Através disso, pode-se avaliar a quantidade de voltas executadas pelo robô terrestre móvel que vai resultar em um mapa mais próximo à realidade, tornando-se o melhor mapa.

METODOLOGIA

3.1 MONTAGEM DO SISTEMA

Para o desenvolvimento deste trabalho de mapeamento de um ambiente *indoor*, utilizou-se experimentos simulados e experimentos reais. A utilização de um ambiente simulado se faz necessário pois ele é capaz de fornecer uma projeção do que pode acontecer no experimento real. Dessa forma, os experimentos foram primeiramente simulados e, depois, foram testados no ambiente real.

Para os experimentos, tanto simulados quanto reais, os seguintes sistemas foram utilizados: a plataforma robótica terrestre móvel HUSKY A200, o sensor LiDAR SICK LMS111 e o sensor RGB-D Microsoft Kinect v1. O robô possui uma chapa metálica acima das suas rodas, formando uma plataforma. O sensor LiDAR foi acoplado sobre essa chapa e o sensor RGB-D foi acoplado sobre o para-choque frontal conforme mostrados na Figura 3.1.

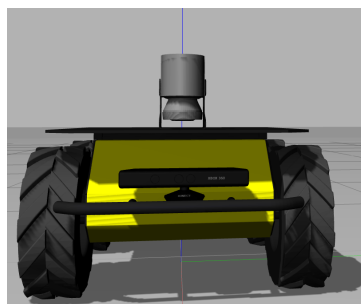


Figura 3.1: Posicionamento dos sensores LiDAR e RGB-D acoplados ao robô terrestre móvel. Fonte: Próprio autor.

Os Experimentos simulados e reais foram feitos utilizando os *frameworks* ROS versão *Kinetic* e Gazebo versão 7.0.0 instalados em um sistema computacional do tipo notebook DELL XPS L502X com o seguinte *hardware*: processador Intel®Core i5 2ª geração, placa

gráfica GForce ®GT540M 2Gb, 8Gb de memória RAM. O sistema operacional utilizado foi Ubuntu 16.04LTS.

Para que os mapas fossem sistematicamente construídos, uma rota foi empiricamente estabelecida de modo que, ao ser seguida por uma única vez, esta estaria abrangendo a maioria das regiões e direções observáveis pelos sensores tanto no ambiente simulado quanto no ambiente real conforme representado pela Figura 3.2.

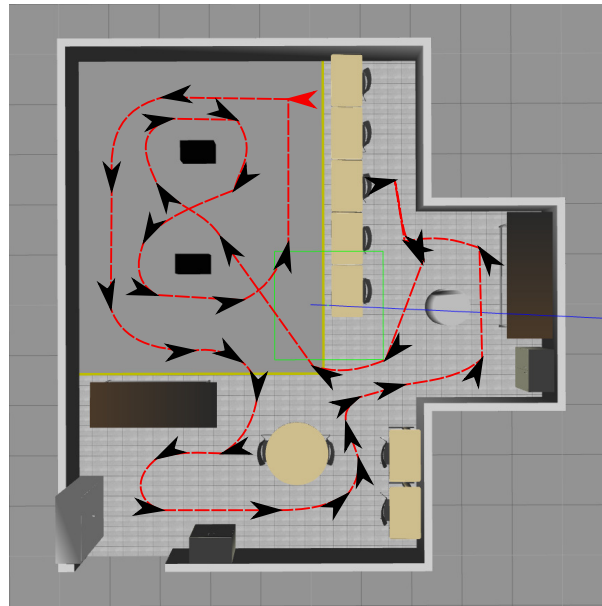


Figura 3.2: Trajetória a ser percorrida. Em vermelho o ponto inicial e final. Fonte: Próprio autor.

Para a realização das simulações computacionais, foi necessário a construção de um modelo geométrico que representasse com a maior fidelidade possível o ambiente que os testes reais seriam realizados. Garantir a maior fidelidade entre ambiente real e o modelo geométrico construído para os testes simulados minimiza possíveis divergências entre os mapas produzidos em ambiente simulado e os mapas produzidos no ambiente real. Dessa forma, construiu-se um modelo tridimensional do Laboratório de Robótica da Universidade Federal da Bahia¹ utilizando ferramentas de modelagem e este modelo foi exportado para o GAZEBO. As Figuras 3.3a, 3.3b e 3.3c mostram, em diferentes ângulos, o modelo geométrico do laboratório de robótica e a disposição objetos presentes.

No ambiente simulado, o modelo do robô HUSKY A200 é livremente fornecido pelo fabricante, a *Clearpath Robotics*, já com o sensor LIDAR. O modelo do sensor RGB-D é fornecido pelo próprio GAZEBO, podendo apenas ser importado para o robô simulado. Assim, tendo o modelo geométrico do ambiente, dos sensores e do robô.

Para ambos os ambientes, real e simulado, foi escolhido o pacote do ROS HECTOR SLAM como método de mapeamento ao se utilizar somente o sensor LiDAR. A escolha pelo pacote HECTOR SLAM se deu em função desse método apresentar um acurado

¹Desenvolvido por Erick Suzart de Souza

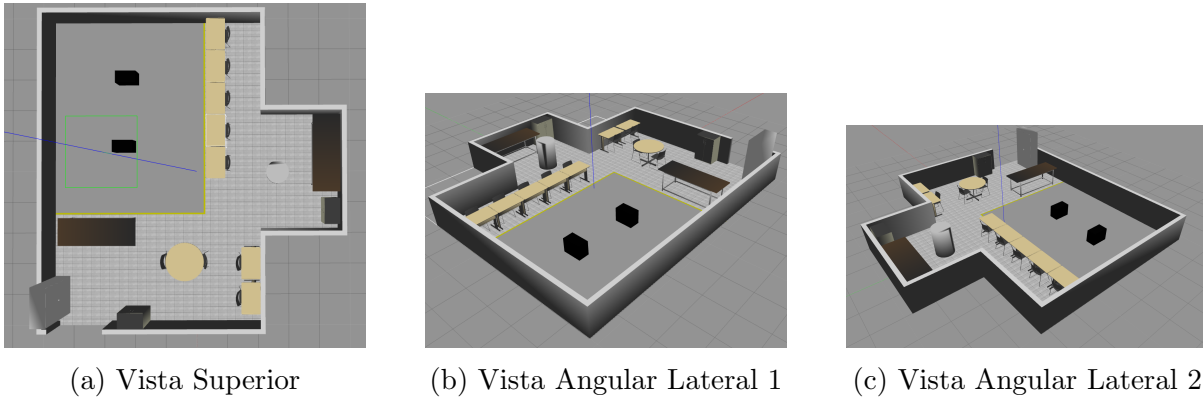


Figura 3.3: Modelo 3D do Laboratório de Robótica. Fonte: Próprio autor.

resultado utilizando a raiz do erro quadrático médio da trajetória [Filipenko e Afanasyev 2018], com simplicidade de implementação e consumo dos recursos computacionais.

Já para o sistema do sensor RGB-D foi escolhido o pacote RTAB-Map por permitir que um mapa de ocupação bidimensional fosse criado baseado nas observações feitas pelo sensor RGB-D igualmente os mapas criados pelos sistemas à LiDAR, o que não é possível ser feito a partir de outros sistemas de vSLAM presentes na biblioteca de ferramentas do ROS. As implementações disponíveis não permitem uma comparação de mapeamento 2D e 3D por fornecerem somente estruturas tridimensionais como nuvem de pontos 3D ou octo-mapas, que são estruturas diferentes das fornecidas pelos sistemas LiDAR.

3.2 EXPERIMENTOS

Foram realizados quatro experimentos, sendo dois deles executados em ambiente virtual, que será chamado de experimentos em ambiente simulado, utilizando primeiro o mapeamento através do HECTOR SLAM e o segundo através do RTAB-Map. E dois experimentos executados com o robô físico, de forma real, que chamamos de ambiente real, seguindo a mesma sequência: primeiramente o HECTOR SLAM e posteriormente o RTAB-Map.

Como esses mapeamentos trazem como resultado um mapa de ocupação bidimensional, é necessário que se faça um ajuste prévio, em ambos os métodos, de três parâmetros, de forma que estes estejam alinhados em todos os métodos. Esses parâmetros são: (i) resolução, que indica a quantidade de *pixels* por metro, (ii) limiar para ocupado, um valor probabilístico (entre 0 e 1) para estimar que a célula esteja ocupada, e (iii) limiar para livre, um valor probabilístico (entre 0 e 1) para a célula ser considerada não ocupada, neste caso, livre de obstáculos. A tabela 3.1 apresenta os valores adotados para que ambos os métodos HECTOR SLAM e RTAB-Map construíssem os mapas de ocupação utilizando as mesmas referências.

Ao se teleoperar o robô terrestre pela trajetória usando diferentes métodos, observou-se que a cada volta o robô não percorre o mesmo caminho, e com isso os mapas seriam inconsistentes a cada volta entre os métodos. Assim, para evitar desvios de trajetória quando os métodos HECTOR SLAM e RTAB-Map estivessem sendo executados, foi

Informação	Valor	Unidade
Resolução	0.050	m/célula
limiar ocupado	0.650	-
limiar livre	0.196	-

Tabela 3.1: Resolução e valores de limiar ocupado e livre para a construção dos mapas de ocupação. Fonte: Próprio autor.

montado um *dataset* a partir da simulação e do experimento real onde dados provenientes de sensores e informações próprias do robô foram obtidos e gravados simultaneamente utilizando a ferramenta *roslab* para que, posteriormente, os dados fossem processados sobre a mesma trajetória. A tabela 3.2 apresenta os tópicos e seu tipo de dado adquirido para o experimento real e a tabela 3.3 para a simulação envolvendo o GAZEBO.

Tópicos	Tipo de dado
/camera/depth_registered/image_raw	sensor_msgs/Image
/camera/rgb/camera_info	sensor_msgs/CameraInfo
/camera/rgb/image_rect_color	sensor_msgs/Image
/diagnostics	diagnostic_msgs/DiagnosticArray
/diagnostics_agg	diagnostic_msgs/DiagnosticArray
/diagnostics_toplevel_state	diagnostic_msgs/DiagnosticStatus
/husky_velocity_controller/cmd_vel	geometry_msgs/Twist
/husky_velocity_controller/odom	nav_msgs/Odometry
/husky_velocity_controller/parameter_descriptions	dynamic_reconfigure/ConfigDescription
/husky_velocity_controller/parameter_updates	dynamic_reconfigure/Config
/imu/data	sensor_msgs/Imu
/joint_states	sensor_msgs/JointState
/joy_teleop/cmd_vel	geometry_msgs/Twist
/joy_teleop/joy	sensor_msgs/Joy
/odometry/filtered	nav_msgs/Odometry
/rosout	roscpp_msgs/Log
/scan_inv	sensor_msgs/LaserScan
/status	husky_msgs/HuskyStatus
/tf	tf2_msgs/TFMessage
/tf_static	tf2_msgs/TFMessage
/twist_marker_server/update	visualization_msgs/InteractiveMarkerUpdate
/twist_marker_server/update_full	visualization_msgs/InteractiveMarkerInit

Tabela 3.2: Lista de tópicos do ROS e seu respectivo tipo de dado utilizados para o *dataset* do Laboratório de Robótica. Fonte: Próprio autor.

Nota-se que são publicados dois tipos diferentes de odometria, uma proveniente diretamente do robô e outra que foi filtrada. Estes diferentes tópicos de odometria são fundamentais para compreender o processo de mapeamento utilizando o método RTAB-Map.

O pacote RTAB-Map, por definição, necessita como entrada uma fonte qualquer que forneça informações de odometria (posição e orientação). O robô Husky A200 possui *encoders* em suas rodas, o que por si só já é suficiente para fornecer uma fonte odométrica. Entretanto, utilizar somente esses *encoders* como fonte de posição e orientação, implicaria em utilizar a técnica *Dead Reckoning*. Apesar de ser bem conhecida na literatura, esta técnica possui um erro de posição e orientação associado a um processo de integração, o que leva a uma incoerência entre a posição real e a posição estimada do robô. Este erro de integração aumenta ao longo do tempo e assim o mapeamento fica afetado conforme

Tópicos	Tipo de dado
/clock	rosgaph_msgs/Clock
/cmd_vel	geometry_msgs/Twist
/camera/depth_registered/image_raw	sensor_msgs/Image
/camera/rgb/camera_info	sensor_msgs/CameraInfo
/camera/rgb/image_rect_color	sensor_msgs/Image
/diagnostics	diagnostic_msgs/DiagnosticArray
/gazebo/link_states	gazebo_msgs/LinkStates
/gazebo/model_states	gazebo_msgs/ModelStates
/gazebo/parameter_descriptions	dynamic_reconfigure/ConfigDescription
/gazebo/parameter_updates	dynamic_reconfigure/Config
/gazebo_gui/parameter_descriptions	dynamic_reconfigure/ConfigDescription
/gazebo_gui/parameter_updates	dynamic_reconfigure/Config
/husky_velocity_controller/cmd_vel	geometry_msgs/Twist
/husky_velocity_controller/odom	nav_msgs/Odometry
/husky_velocity_controller/parameter_descriptions	dynamic_reconfigure/ConfigDescription
/husky_velocity_controller/parameter_updates	dynamic_reconfigure/Config
/imu/data	sensor_msgs/Imu
/joint_states	sensor_msgs/JointState
/odometry/filtered	nav_msgs/Odometry
/rosout	rosgaph_msgs/Log
/scan	sensor_msgs/LaserScan
/status	husky_msgs/HuskyStatus
/tf	tf2_msgs/TFMessage
/tf_static	tf2_msgs/TFMessage
/navsat/fix	sensor_msgs/NavSatFix
/navsat/fix/position/parameter_descriptions	dynamic_reconfigure/ConfigDescription
/navsat/fix/position/parameter_updates	dynamic_reconfigure/Config
/navsat/fix/status/parameter_descriptions	dynamic_reconfigure/ConfigDescription
/navsat/fix/velocity/parameter_updates	dynamic_reconfigure/Config
/navsat/vel	geometry_msgs/Vector3Stamped
/twist_marker_server/update	visualization_msgs/InteractiveMarkerUpdate
/twist_marker_server/update_full	visualization_msgs/InteractiveMarkerInit

Tabela 3.3: Lista de tópicos do ROS e seu respectivo tipo de dado utilizados para o *dataset* em ambiente virtual. Fonte: Próprio autor.

demarcado em vermelho na Figura 3.4, onde obstáculos foram incluídos no mapa mesmo não existindo no ambiente real. Por isso, optou-se por utilizar o método de odometria filtrado, capaz de corrigir este erro por meio de outros sensores.

Um dos métodos mais utilizados para realizar a filtragem de dados da odometria é o Filtro de Kalman Estendido, que, nesse contexto, emprega a Unidade de Medição Inercial - do inglês: *Inertial Measure Unity* (IMU), unidade que fornece dados de aceleração própria e de aceleração angular - e informações de *encoders* presentes no robô HUSKY A200 para realizar uma estimativa mais precisa da posição e orientação do robô.

O Husky A200 possui a implementação do Filtro de Kalman Estendido por meio do pacote *robot_localization* [Moore e Stouch 2014], que promove a fusão destas informações. Esses dados estão publicadas no tópico */odometry/filtered*, que foi escolhido como a fonte de odometria por fornecer informações mais precisas.

3.2.1 Simulação computacional

A primeira simulação tem como objetivo construir um mapeamento produzido somente pelo sensor LiDAR. Portanto, foram utilizados apenas os dados provenientes do sensor e o pacote do ROS HECTOR SLAM. Uma vez que o pacote não requer como entrada



Figura 3.4: Erro no mapeamento ao se utilizar o processo de integração da posição como fonte de odometria.

nenhuma fonte de odometria, mas, sim, as leituras de distância, nenhum outro dado do robô foi utilizado, assim a saída esperada é um mapa bidimensional do ambiente.

O robô foi virtualmente teleoperado pela trajetória proposta e um total de 9 voltas foram executadas. A Figura 3.5 mostra como a simulação foi montada. Nela, os dados oriundos do LiDAR são publicados no tópico `\scan` e subscritos pelo pacote HECTOR-SLAM, que faz o mapeamento e publica o tópico `\map`, com as informações necessárias sobre onde estão as áreas livres e ocupadas.

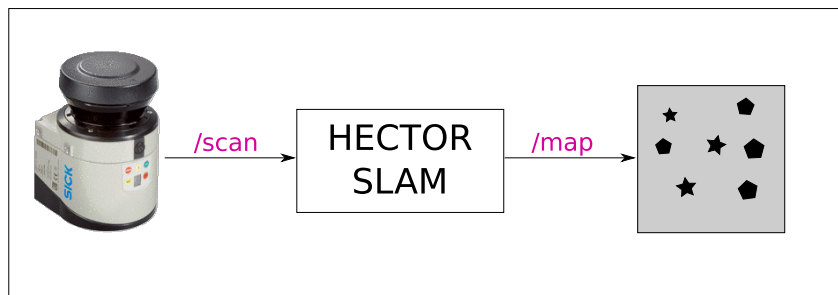


Figura 3.5: Relação de entradas e saídas utilizadas no pacote HECTOR SLAM. Fonte: Próprio autor.

A segunda simulação constrói um mapeamento produzido pelo conjunto de sensores RGB-D e LiDAR. Utilizou-se o pacote RTAB-Map em sua configuração máxima de sensores RGB-D e LiDAR. Este pacote requer como entrada minimamente uma fonte qualquer

de odometria, para isso foi utilizada a odometria existente no tópico `\filtered\odom`, que é uma fonte de odometria filtrada. Utilizou-se, também, o sensor LiDAR como entrada, para que este pudesse auxiliar no processamento dos dados provenientes da câmera RGB-D. A Figura 3.6 representa os sensores e os tópicos que foram utilizados como entrada e o tópico que publica o mapa produzido. Os dados coletados são provenientes da teleoperação virtual do robô por um total de 9 voltas.

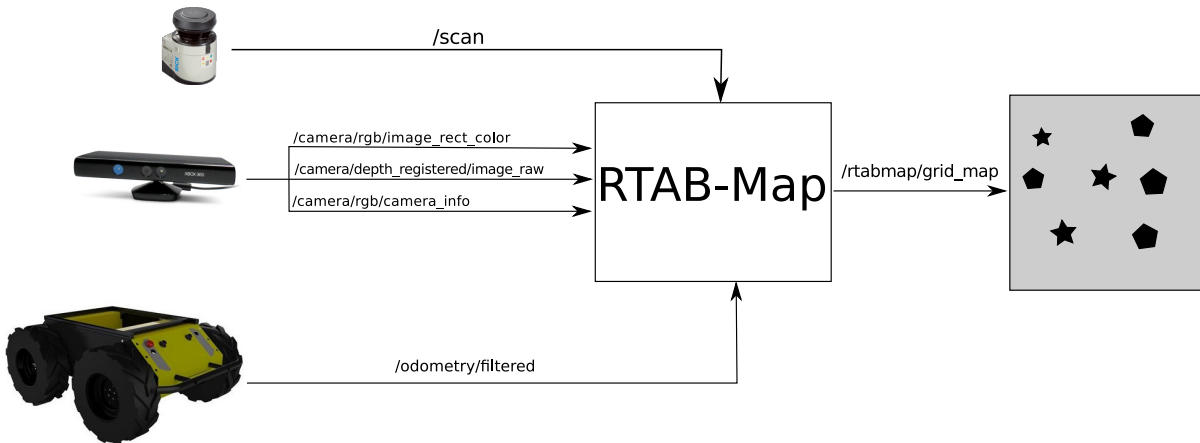


Figura 3.6: Relação de entradas e saídas utilizadas no pacote RTAB-Map. Fonte: Próprio autor.

3.2.2 Experimento Real

Após a realização dos experimentos em ambiente simulado, executou-se os experimentos no robô físico, em ambiente real no Laboratório de Robótica da Universidade Federal da Bahia. Para esses experimentos, o robô real foi alocado de forma que a sua posição e orientação fossem as mais próximas possíveis das praticadas no ambiente da simulação computacional, possibilitando que o mapa fosse construído com as mesmas orientações, tanto em ambiente simulado quanto em ambiente real, de forma a minimizar possíveis diferenças entre os mapas.

Assim como na simulação computacional, para evitar erros de trajetória, foi construído um *dataset* a fim de que se coletasse os dados de ambos os sensores e do robô de uma só vez, para que pudessem ser pós-processados por diferentes formas de mapeamento. Dessa forma, a trajetória é a idêntica para ambos os sensores utilizados. O robô terrestre móvel foi operado por 9 vezes sobre a trajetória e os dados foram capturados utilizando a ferramenta *roslaunch* do *framework* ROS e estão dispostos na tabela 3.2.

O segundo processamento do *dataset* foi feito utilizando as informações provenientes dos tópicos publicados pelo sensor LiDAR (`\scan`), sensor RGB-D (`/camera/rgb/image_rect_color`, `/camera/depth_registered/image_raw` e `/camera/rgb/camera_info`) e a fonte de odometria filtrada (`/odometry/filtered`) como entrada para o pacote RTAB-Map para a realização do mapeamento. Os dados foram processados e os mapas foram publicados no tópico: `/rtabmap/grid_map`.

3.2.3 Ground Truth e análise métrica

Para saber o quão próximos esses mapas gerados pelo RTAB-Map e o HECTOR SLAM no experimento em ambiente real estão das dimensões reais do Laboratório de Robótica da UFBA, comparou-se as medidas das paredes limítrofes desses mapas com as medidas das paredes retratadas no *ground truth*, estas últimas tomadas como exatas (Figura 3.7).

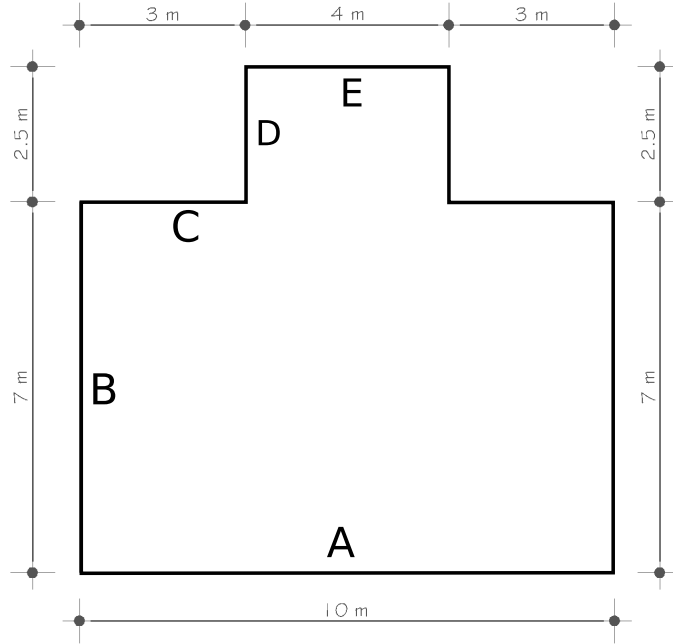


Figura 3.7: Medidas das paredes do Laboratório de Robótica. Fonte: Próprio autor.

Para medir as dimensões de cada parede produzida pelos mapeamentos foi utilizado a ferramenta de medição *measure*, presente no RVIZ. Cada parede foi medida por dez vezes e calculou-se o desvio padrão (σ) por meio da equação 3.1. Onde N é o total de amostras, $\bar{x}$ e x_i significam respectivamente a média e o valor da medição atual.

$$\sigma = \sqrt{\frac{\sum_{i=1}^N (x_i - \bar{x})^2}{N}} \quad (3.1)$$

A fim de compreender em qual volta o ambiente é melhor representado pelos mapeamentos produzidos pelo RTAB-Map, no ambiente real, e se há influência da quantidade de voltas na qualidade do mapa gerado, utilizou-se a métrica Índice de Similaridade Estrutural (SSIM, do inglês *Structural Similarity Index Measure*). A implementação da técnica de similaridade SSIM está disponível no *software* MatLab[®] na função *ssim(test, ref)*.

Para o cálculo desse índice, um mapa foi escolhido, através de uma inspeção visual, como o que melhor representa o ambiente real, baseado na menor quantidade de áreas não mapeadas e sem sobreposição de objetos. Este mapa foi chamado de mapa-referência, sendo comparado com os mapas de cada volta produzidos pelo método RTAB-Map e gerando um valor adimensional variável entre 0 e 1, sendo 0 quando não há similaridade e 1 quando há similaridade total. Mapas de similaridade local foram construídos em uma escala de cinza, onde a cor branca representa similaridade total e a cor preta representa nenhuma similaridade.

RESULTADOS E DISCUSSÃO

Nesta capítulo serão apresentados os resultados obtidos através dos experimentos executados, juntamente com a discussão sobre eles. Primeiro, serão mostrados os resultados obtidos por meio de simulação computacional, utilizando o sensor LiDAR, e em seguida os resultados produzidos pelo conjunto de sensores LiDAR e RGB-D, ainda em ambiente simulado. A seguir, expõe-se os resultados alcançados através do mapeamento em ambiente real, utilizando o sensor LiDAR, seguido pelos resultados provenientes do uso do conjunto de sensores LiDAR e RGB-D no ambiente real. Subsequentemente são apontados os resultados das análises métricas e da aplicação do Índice de Similaridade Estrutural. Por fim, são retratados os mapas de similaridade local.

Todas as figuras aqui apresentadas, encontram-se em alta resolução no anexo A.

4.1 RESULTADOS DAS SIMULAÇÕES COMPUTACIONAIS

4.1.1 Simulação no GAZEBO com: Sensor LiDAR + pacote HECTOR SLAM

O primeiro experimento foi realizado utilizando o ambiente virtual GAZEBO sobre o *dataset* coletado. Para o processamento destes dados foi criado um arquivo *.launch* que evoca o conjunto de arquivos necessários para se utilizar o HECTOR SLAM, bem como o tópico que continha a informação do sensor LiDAR conforme mostrado na Figura 3.5.

O relacionamento entre os tópicos, subscritos e publicados, são apresentados na Figura 4.1. Em vermelho está o tópico de entrada do sistema que contém as leituras provenientes do sensor LiDAR. Em azul, o nó responsável por transformar os dados de entrada em uma representação geométrica. Já em verde, a saída que contém o mapa bidimensional do ambiente em um formato de mapa de ocupação.

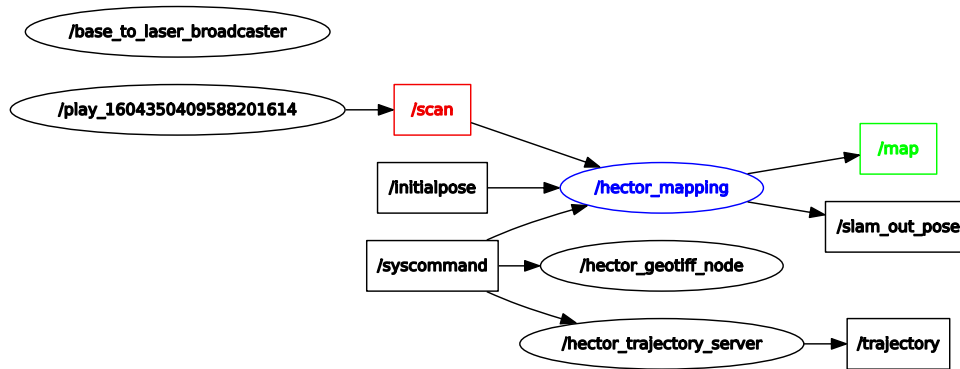


Figura 4.1: Relacionamento entre os tópicos subscritos e publicados pelo ROS no primeiro experimento simulado. Fonte: Próprio autor.

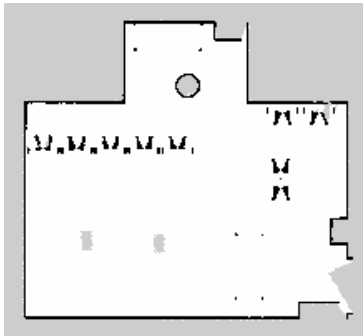
O tempo em cada volta não foi computado diretamente pelo sistema computacional, mas por um cronômetro independente. A tabela 4.1 mostra o tempo gasto para um total de nove voltas, bem como o tempo parcial de cada volta. Apesar de ser necessário 00 : 06 : 05,41 para completar a primeira volta, o mapa já estava suficientemente construído, contendo uma representação suficiente do ambiente antes do fim dessa volta. Isso acontece devido ao grande alcance do sensor LiDAR LMS11, que chega a ser de 20 metros e também da trajetória adotada, o que faz com que o robô complete o mapa antes mesmo de completar uma volta inteira, retornando para a posição inicial e também pela trajetória planejada.

Volta	Tempo[hh:min:seg]
1	00:06:05,41
2	00:05:38,56
3	00:06:17,33
4	00:06:33,14
5	00:06:19,12
6	00:06:27,71
7	00:06:03,07
8	00:05:52,98
9	00:06:11,74
Total	00:55:41,59

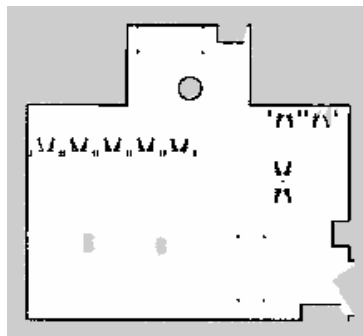
Tabela 4.1: Tempo gasto para percorrer cada volta e o tempo total gasto em ambiente virtual. Fonte: Próprio autor.

Analisando os mapas produzidos (Figura 4.2), pode-se perceber que já na primeira volta, o sensor LiDAR apresenta um mapa bem delimitado, coerente com a projeção bidimensional do ambiente simulado. Nesse mapa, apesar dos entornos bem delineados, objetos como mesas e cadeiras sofrem prejuízos de representação. Isso ocorre porque o sensor LiDAR captura os objetos de uma altura fixa, a altura do feixe de luz que emite. Assim, apenas o suporte estrutural de mesas e cadeiras (os pés) são apresentados na

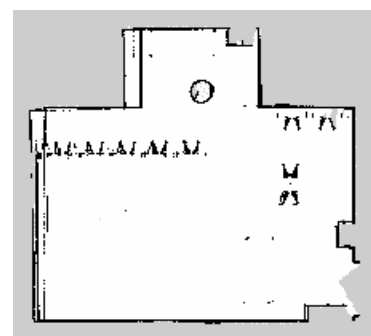
figura. A mesma análise pode ser feita para a segunda volta, na qual não há ganhos visuais significativos com relação à primeira volta.



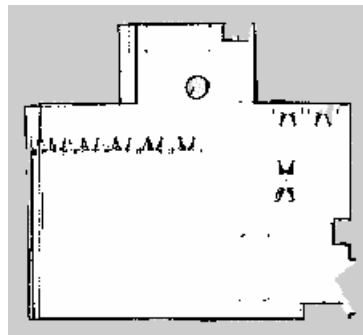
a) Mapa para a primeira volta em simulação LiDAR + HECTOR SLAM



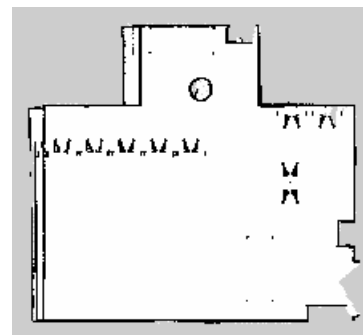
b) Mapa para a segunda volta em simulação LiDAR + HECTOR SLAM



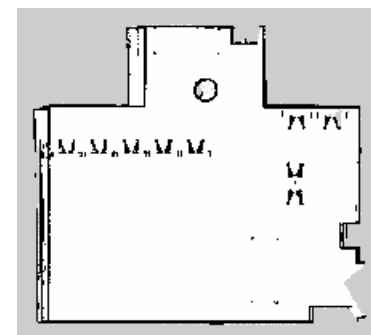
c) Mapa para a terceira volta em simulação LiDAR + HECTOR SLAM



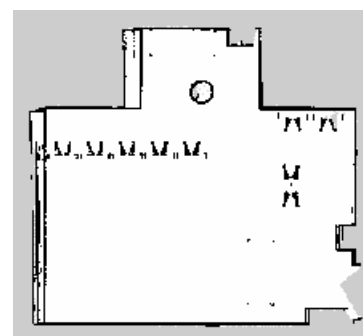
d) Mapa para a quarta volta em simulação LiDAR + HECTOR SLAM



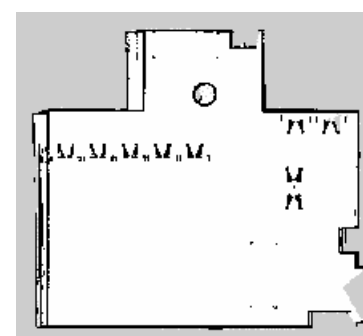
e) Mapa para a quinta volta em simulação LiDAR + HECTOR SLAM



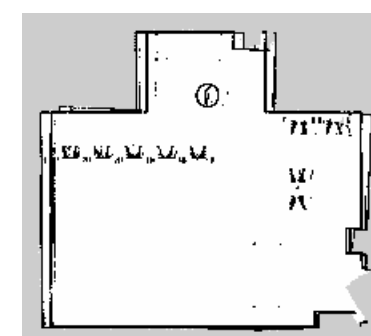
f) Mapa para a primeira volta em simulação LiDAR + HECTOR SLAM



g) Mapa para a sétima volta em simulação LiDAR + HECTOR SLAM



h) Mapa para a oitava volta em simulação LiDAR + HECTOR SLAM



i) Mapa para a nona volta em simulação LiDAR + HECTOR SLAM

Figura 4.2: Mapa de ocupação adquiridos pelos sensor LiDAR em simulação no GAZEBO.
Fonte: Próprio autor.

A partir da terceira volta, observam-se sobreposições de regiões. As paredes antes bem delimitadas, passam a não ser coerentes, como se o robô tivesse sobreposto um mapa a outro. Uma das hipóteses para que isto possa ter acontecido é de que o robô, ao receber um comando de translação ou rotação, não executava um controle do movimento das rodas suave. Com o início do arranque de rodas de forma abrupta, pode ter havido um pequeno deslocamento de sua posição em relação à posição anterior. Com sucessivos movimentos de translação e rotação este erro se propagou e impediu que um mapa uniforme para todas as voltas fosse produzido.

Observando os mapas produzidos na primeira e segunda voltas possuem qualidades visuais similares. Levando em consideração que a primeira volta já é suficiente para um mapa coerente e que é realizado com menos tempo, o que poupa o tempo de execução do mapeamento, o mapa produzido pela primeira volta representa o melhor mapa dentre os produzidos.

4.1.2 Simulação no GAZEBO com: Sensor LiDAR + Sensor RGB-D + Odometria filtrada + pacote RTAB-Map

Na segunda simulação, o sensor RGB-D foi utilizado juntamente com o sensor LiDAR para o mapeamento no simulador GAZEBO. Nesta simulação, utilizou-se como entrada os dados oriundos dos sensores: RGB-D e LiDAR. A odometria proveniente da fusão entre os dados odométricos dos *encoders* e IMU foi utilizada como entrada para o RTAB-Map. A Figura 4.3 mostra o relacionamento entre os tópicos lidos e publicados pelo ROS para a formação dos mapas de ocupação. Os tópicos em vermelho denotam os tópicos de entrada, em azul o pacote RTAB-Map e em verde o mapa de ocupação.

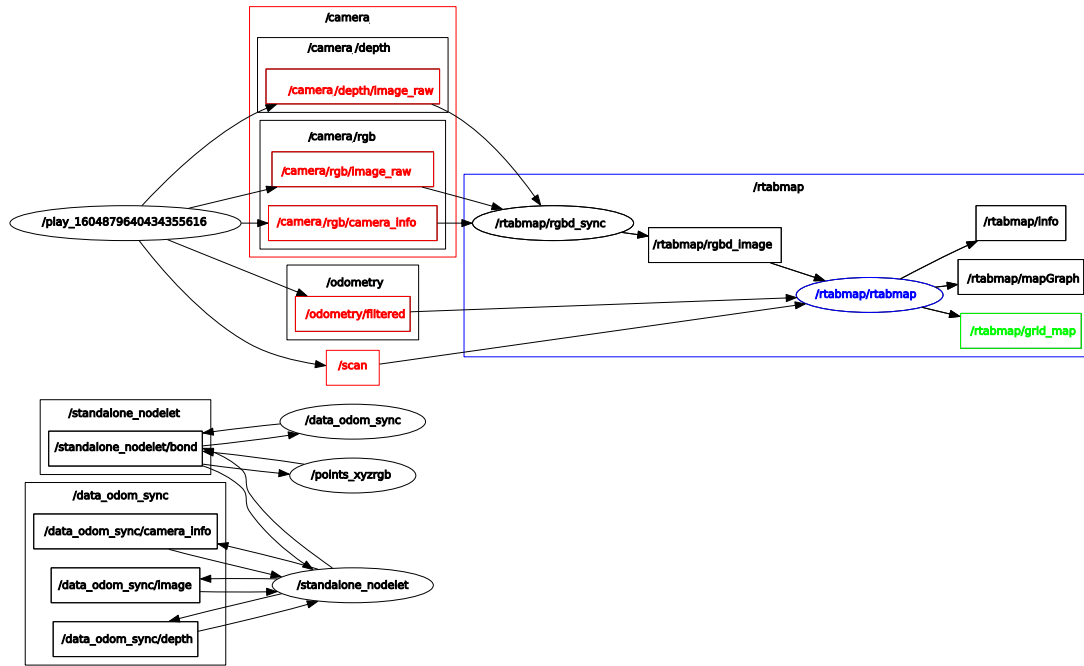


Figura 4.3: Relacionamento entre os tópicos subscritos e publicados na segunda simulação. Fonte: Próprio autor.

O tempo total e de cada volta foram os mesmos da simulação anterior, uma vez que a base de dados é a mesma para as duas simulações e está representada na tabela 4.1. Os mapas de ocupação resultantes desta combinação de sensores e método de SLAM são apresentados na Figura 4.4.



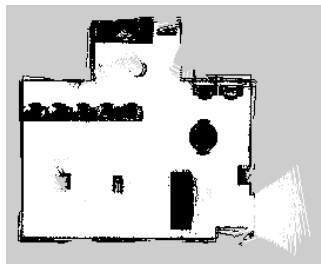
a) Mapa para a primeira volta em simulação LiDAR + RGB-D + Odometria + RTAB-Map.



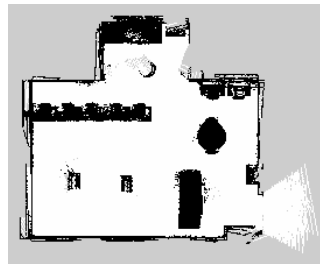
b) Mapa para a segunda volta em simulação LiDAR + RGB-D + Odometria + RTAB-Map.



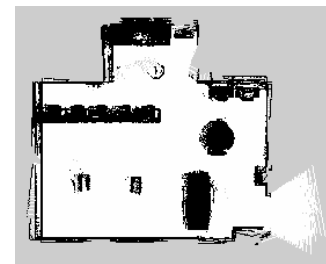
a) Mapa para a terceira volta em simulação LiDAR + RGB-D + Odometria + RTAB-Map.



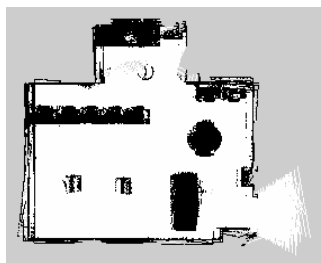
d) Mapa para a quarta volta em simulação LiDAR + RGB-D + Odometria + RTAB-Map



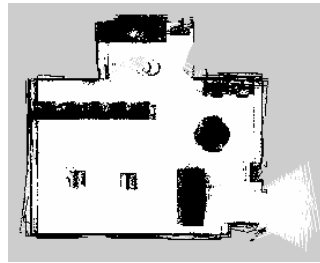
e) Mapa para a quinta volta em simulação LiDAR + RGB-D + Odometria + RTAB-Map



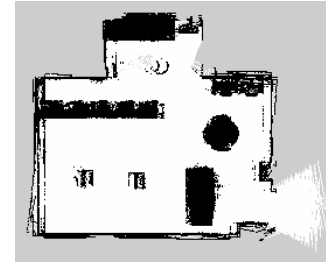
f) Mapa para a sexta volta em simulação LiDAR + RGB-D + Odometria + RTAB-Map



g) Mapa para a sétima volta em simulação LiDAR + RGB-D + Odometria + RTAB-Map.



h) Mapa para a oitava volta em simulação LiDAR + RGB-D + Odometria + RTAB-Map.



i) Mapa para a nona volta em simulação LiDAR + RGB-D + Odometria + RTAB-Map.

Figura 4.4: Mapa de ocupação adquiridos pelos sensor LiDAR em simulação no GAZEBO. Fonte: Próprio autor.

Pode-se observar que já na primeira volta, há, visualmente, uma representação do ambiente com qualidade. Diferente dos mapas utilizando somente o sensor LiDAR, aqui objetos como mesas e cadeiras passam a ser inteiramente representados. O mapeamento resultante da segunda volta apresenta uma melhoria na representação de algumas áreas, se comparado à primeira volta. A partir da terceira volta, sobreposições de regiões começam a ser observadas, o que invalida a utilização para atividades de localização e planejamento de caminhos pois as representações estão distorcidas e sem validade, uma vez que eles não representam mais a realidade, podem causar danos à estrutura física do robô caso

este mapa seja utilizado. Este erro foi transmitido aos mapas das voltas subsequentes, pois uma vez que uma correspondência incorreta entre os mapas aconteceu, ela dificilmente será removida.

Apesar do RTAB-Map ser um SLAM que possui a capacidade de reconhecer áreas já visitadas (fechamento de *loop*) os mesmos movimentos de deslocamento angular e linear que produziram incoerências na simulação 4.1.1 aqui também foram presentes uma vez que utilizou-se o mesmo *dataset*. Estas sobreposições de mapas reafirmam a hipótese de que movimentos abruptos que fazem as rodas deslizarem sem produzir movimento podem interferir no mapeamento, uma vez que produzem uma odometria inconsistente, não sendo removidas pelo Filtro de Kalman Estendido, e se propagam, assim, por todo o mapeamento.

4.2 RESULTADOS DOS EXPERIMENTOS EM AMBIENTE REAL

Para o desenvolvimento desta parte experimental, construiu-se um *dataset* com tópicos do ROS conforme a tabela 3.2 utilizando os dados provenientes do robô Husky A200 que compõem a odometria, sensor LiDAR LMS111, sensor RGB-D *Microsoft Kinect* utilizando o *framework* ROS e ferramenta de gravação de dados *rosvbag*. As informações para a composição do *dataset* foram coletadas enquanto foram percorridas um total de nove voltas pela trajetória.

4.2.1 Experimento real com: Sensor LiDAR + pacote HECTOR SLAM

O primeiro experimento no ambiente real foi conduzido utilizando os dados do sensor LMS111 (LiDAR) e o pacote HECTOR SLAM. A Figura 4.5 mostra o relacionamento entre os tópicos lidos e publicados pelo ROS enquanto o percurso era feito. Destaca-se em vermelho o tópico que contém as informações oriundas da leitura do sensor LiDAR LMS111. Em azul, o nó que executará o processamento dos dados provenientes do *laser*, transformando estes dados em um mapeamento bidimensional do ambiente. Em verde está o tópico que contém as informações para visualização do mapa.

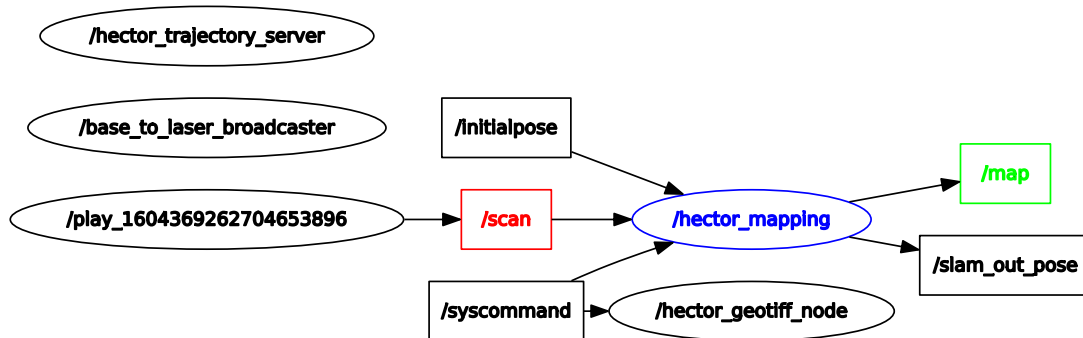


Figura 4.5: Relacionamento entre os tópicos subscritos e publicados no primeiro experimento real. Fonte: Próprio autor.

O tempo para completar 9 voltas, bem como o tempo de cada volta parcial também foi computado por um cronômetro externo e os valores estão representados na tabela

4.2. A velocidade para o deslocamento do robô pela trajetória foi ajustada como a mínima possível, usando um controlador manual (*joystick*), de forma que a velocidade de deslocamento não interferisse na aquisição de imagens do sensor RGB-D, uma vez que este publica imagens na frequência aproximada de 30Hz.

Volta	Tempo [hh:min:seg]
1	00:06:14,33
2	00:06:25,50
3	00:06:09,44
4	00:06:13,19
5	00:05:47,79
6	00:06:16,16
7	00:06:27,94
8	00:05:53,57
9	00:05:41,54
Total	00:55:11,27

Tabela 4.2: Tempo gasto para percorrer cada volta e o tempo total gasto em ambiente Real. Fonte: Próprio autor.

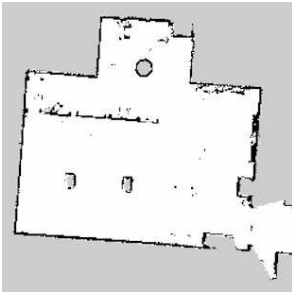
Executando suaves movimentos angulares e lineares para que estes não produzissem associações incorretas no mapa, ao fim de cada volta um mapa era coletado e salvo para análise. A Figura 4.6 mostra os mapas adquiridos volta após volta.

Observa-se que os mapas quase não apresentam diferenças entre as voltas, o que implica que, para a utilização somente do sensor LiDAR, apenas uma única volta foi capaz de representar todos os objetos contidos no ambiente. Apesar do método de mapeamento utilizado não possuir fechamento de *loop* ele foi robusto o suficiente para não deformar o mapa sobrepondo áreas já visitadas.

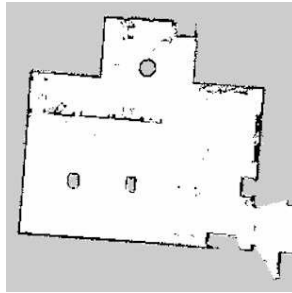
Analisando os mapas de ocupação nota-se que não houve sobreposição dos mapas, como aconteceu na simulação. Uma hipótese para isso é que a movimentação do robô não tenha sido rápida o suficiente para produzir associações erradas durante o processo de mapeamento. É interessante notar que, apesar de ter produzido um leve ruído no mapa de ocupação, os limites do ambiente foram bem definidos.

Apesar de ser capaz de representar os limites do ambiente, alguns objetos como as mesas e cadeiras ficaram mal representadas, não permitindo uma identificação visual do tipo e tamanho do objeto ali presente, assim como discutido no subtópico do ambiente simulado (4.1.1).

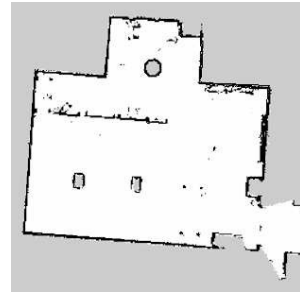
Esta dificuldade de interpretação visual do ambiente pode gerar uma interpretação errônea do ambiente, permitindo que um operador que não tenha um conhecimento prévio do mesmo possa executar um planejamento de rota que coloque a estrutura física do robô em risco, uma vez que há possibilidade de planejar uma rota que possa ir de encontro a um objeto.



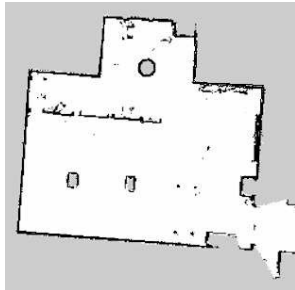
a) Mapa para a primeira volta em ambiente real com LiDAR + HECTOR SLAM.



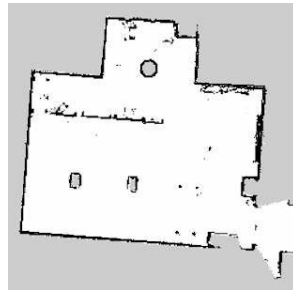
b) Mapa para a segunda volta em ambiente real com LiDAR + HECTOR SLAM.



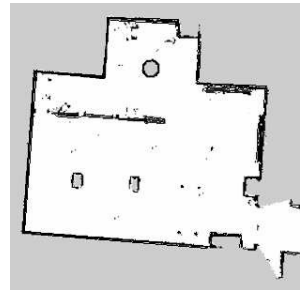
c) Mapa para a terceira volta em ambiente real com LiDAR + HECTOR SLAM.



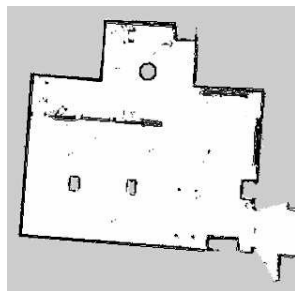
d) Mapa para a quarta volta em ambiente real com LiDAR + HECTOR SLAM.



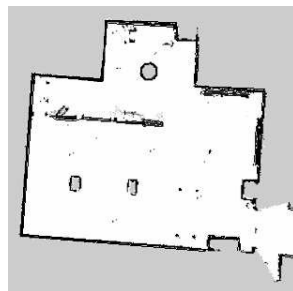
e) Mapa para a quinta volta em ambiente real com LiDAR + HECTOR SLAM.



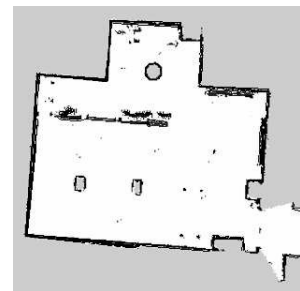
f) Mapa para a sexta volta em ambiente real com LiDAR + HECTOR SLAM.



g) Mapa para a sétima volta em ambiente real com LiDAR + HECTOR SLAM.



h) Mapa para a oitava volta em ambiente real com LiDAR + HECTOR SLAM.



i) Mapa para a nona volta em ambiente real com LiDAR + HECTOR SLAM.

Figura 4.6: Resultado de sucessivas voltas pela trajetória usado sensor LiDAR e pacote HECTOR SLAM. Fonte: Próprio autor.

4.2.2 Experimento real com: Sensor LiDAR + Sensor RGB-D + Odometria filtrada + pacote RTAB-Map

Este último experimento foi conduzido utilizando os mesmos sensores que na simulação 4.1.2. Os tópicos contendo as informações do sensor RGB-D (*Microsoft Kinect*), sensor LiDAR LMS111 e os dados de odometria filtrada por meio de um Filtro de Kalman Estendido, foram as entradas para o pacote do ROS RTAB-Map para serem processados e transformados em mapas de ocupação bidimensionais, igualmente aos produzidos pelos sistemas de SLAM 2D, porém, desta vez, contendo informações sobre a tridimensionalidade do ambiente.

A Figura 4.7 mostra o relacionamento dos tópicos dentro do ROS. Os tópicos de entrada do RTAB-MAP foram demarcados de vermelho, o nó responsável pelo mapeamento é representado em azul e o tópico que contém mapa atual é denotado na cor verde.

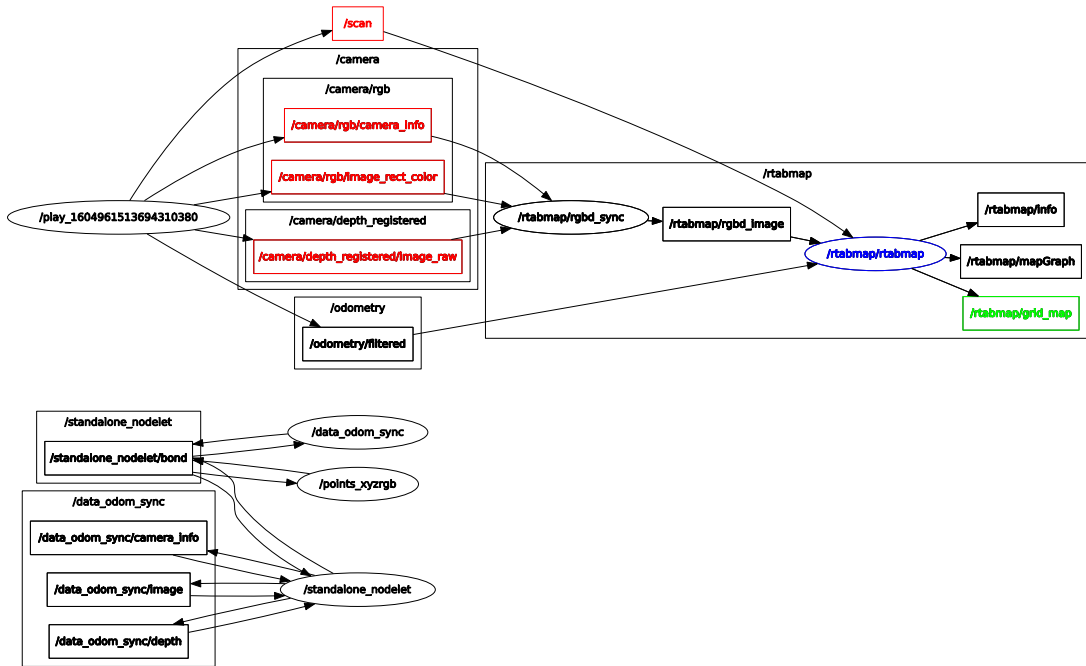
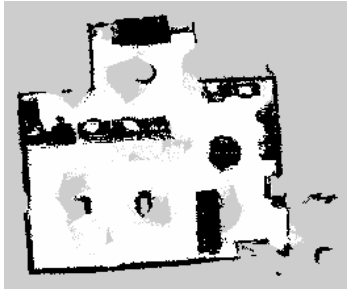
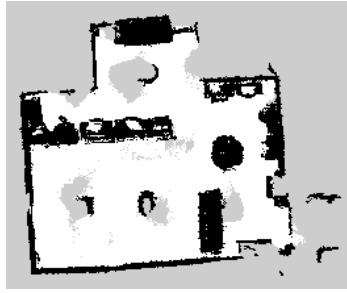


Figura 4.7: Relacionamento entre os tópicos subscritos e publicados no segundo experimento real. Fonte: Próprio autor.

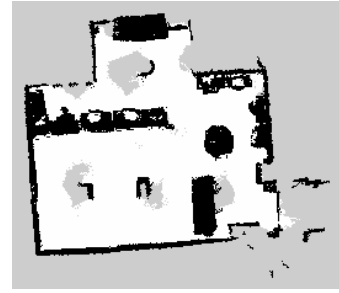
Os mapas adquiridos ao fim de cada volta estão dispostos na Figura 4.8. Nesta figura, é possível notar que diferenças substanciais foram observadas em relação ao mapeamento realizado somente pelo sensor LiDAR.



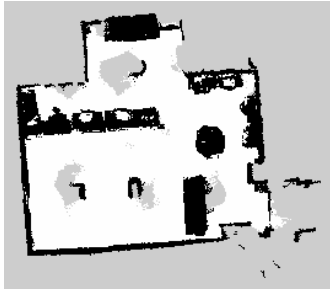
a) Mapa para a primeira volta em ambiente real LiDAR + RGB-D + Odometria + RTAB-Map.



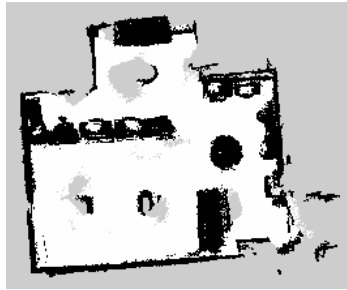
b) Mapa para a segunda volta em ambiente real LiDAR + RGB-D + Odometria + RTAB-Map.



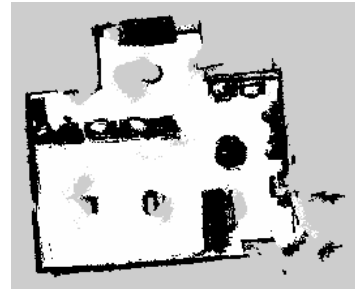
a) Mapa para a terceira volta em ambiente real LiDAR + RGB-D + Odometria + RTAB-Map.



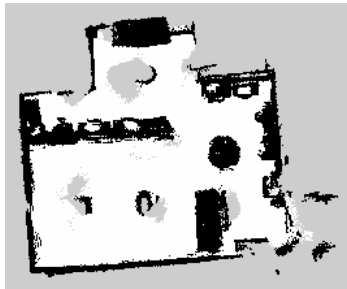
d) Mapa para a quarta volta em ambiente real LiDAR + RGB-D + Odometria + RTAB-Map.



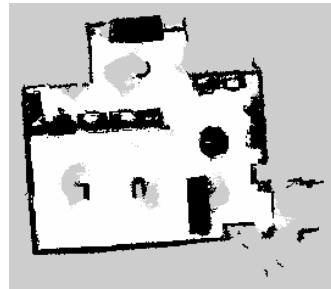
e) Mapa para a quinta volta em ambiente real LiDAR + RGB-D + Odometria + RTAB-Map.



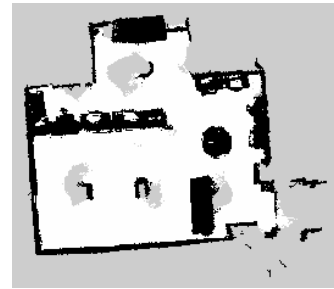
f) Mapa para a sexta volta em ambiente real LiDAR + RGB-D + Odometria + RTAB-Map.



g) Mapa para a sétima volta em ambiente real LiDAR + RGB-D + Odometria + RTAB-Map.



h) Mapa para a oitava volta em ambiente real LiDAR + RGB-D + Odometria + RTAB-Map.



i) Mapa para a nona volta em ambiente real LiDAR + RGB-D + Odometria + RTAB-Map.

Figura 4.8: Resultado de sucessivas voltas pela trajetória usado os sensores câmera RGB-D e LiDAR utilizando o pacote RTAB-Map. Fonte: Próprio autor.

Inicialmente, observa-se que novos locais demarcados como ocupados, surgiram em relação à Figura 4.6. Mesas e cadeiras contidas no ambiente foram mapeadas de forma que uma identificação visual pode ser feita. Apesar deste mapeamento apresentar uma

maior quantidade de objetos mapeados, ele também apresenta uma menor imunidade ao ruído provocado pelo sensor RGB-D, onde é possível observar pequenos pontos pretos não agrupados próximo dos objetos contidos no ambiente. Estes erros são esperados uma vez que o sensor RGB-D não possui uma grande precisão como os sensores LiDAR 3D à *laser*. Apesar destes erros, o algoritmo foi capaz de desempenhar sua função de mapeamento.

Observa-se, também, que algumas regiões não foram claramente mapeadas (espaços dentro do ambiente representados na cor cinza). Estas regiões não foram mapeadas devido ao fato do robô não tê-las observado por outros ângulos e posições. Para corrigir esta deficiência seria necessário que o robô pudesse circundar o objeto, o que faria com que ele fosse observado por diversas posições, mas isto implicaria um aumento do tempo de operação do processo de mapeamento, o que para muitos robôs operados por baterias é um fator limitante.

Nota-se que o mapa da primeira volta apresenta um bom mapeamento e os objetos do ambiente estão bem representados. Entretanto, algumas áreas do ambiente real não foram mapeadas pelo robô terrestre móvel. Já na segunda volta, tem-se a redução das áreas não mapeadas, continuando a boa definição do mapa. Até o mapa da quarta volta, as regiões não mapeadas foram diminuindo, mantendo a qualidade do mapeamento, sem degenerações. Dos mapas da quinta volta até à sétima volta, ocorre uma sobreposição dos mapas, o que compromete a qualidade desses mapas e os torna descartáveis. Por fim, nas voltas 8 e nove, as áreas de sobreposição foram corrigidas, mantendo a originalidade do ambiente.

Dessa forma, através de uma inspeção visual, conclui-se que o mapa correspondente à quarta volta é o melhor mapa obtido, pois, em progressão, ele teve a maior redução de áreas não mapeadas, sem que houvessem sobreposições de mapas que comprometessem a qualidade do mapeamento.

É interessante observar que, a todo momento, a utilização do sensor LiDAR, faz com que ele atue para corrigir as distorções de alinhamento do mapa produzido pelo sensor RGB-D, aumentando a consistência dos mapas.

4.3 ANÁLISE COMPARATIVA MÉTRICA

Os sensores utilizados nesta pesquisa, LiDAR e RGB-D, são sensores de natureza distintas: o primeiro opera por disparos de luz e o segundo por captação de luz. Ambos são utilizados para a produção de mapas de ocupação, porém não há como estabelecer um parâmetro para comparação direta entre os sensores. Por isso propõe-se, neste estudo, que, para compreender qual mapeamento representa melhor o ambiente real, é preciso que haja uma comparação entre a qualidade dos mapas produzidos, feita através da análise comparativa métrica.

Para a essa análise comparativa métrica, levou-se em consideração as medidas das partes comuns entre os mapas produzidos. Desta forma, os erros associados puderam ser comparados. Essas partes comuns são as paredes do ambiente. Para realizar essa comparação, foram eleitos os dois melhores mapas de cada sistema. Assim, utilizou-se mapa produzido pela quarta volta realizada pelo robô em ambiente real empregando o conjunto de sensores LiDAR e RGB-D. Já que os mapas produzidos utilizando somente

o sensor LiDAR não apresentaram muitas variações, escolheu-se, também, o mapa produzido na quarta volta feita pelo robô no ambiente real usando somente o sensor LiDAR. É importante lembrar que esses mapas estão no mesmo instante de tempo.

Nesses mapas, foram medidos os tamanhos das paredes, utilizando a ferramenta *measure*, presente no RVIZ. Cada parede, de cada mapa gerado, foi medida dez vezes e estimou-se o desvio padrão, através da equação 3.1. Essas medidas e desvio padrão estão exibidos na tabela 4.3.

Método	Lados(m)				
	A	B	C	D	E
Ground Truth	10,00	7,00	3,00	2,50	4,00
Ambiente Real usando LiDAR (4.2.1)	9,84±0,08	6,87±0,03	2,67±0,09	2,46±0,04	3,78±0,06
Ambiente real usando RGB-D e LiDAR (4.2.2)	9,52±0,27	6,58±0,14	2,52±0,13	2,27±0,29	3,53±0,36

Tabela 4.3: Erro associado ao mapeamento. Fonte: Próprio autor.

Nesta tabela, percebe-se que, no ambiente real, o sensor LiDAR obteve o menor desvio padrão quando comparado ao sistema composto por câmera RGB-D e LiDAR. Esse erro métrico maior apresentado pelo sistema composto pela câmera RGB-D e LiDAR pode causar um erro de posicionamento quando uma metodologia de localização for implementada sem o auxílio de um sensor que faça este erro tender a zero. Entretanto, esta diferença métrica entre os sistemas aplicados não implica na invalidação do sistema que utiliza o conjunto de sensores RGB-D e LiDAR, afinal é uma diferença baixa, que pode ser compensada por outras vantagens do emprego desse conjunto de sensores, como a maior compreensão da tridimensionalidade do ambiente.

Para compreender a evolução na qualidade dos mapeamentos produzidos na progressão de voltas, aplicou-se a técnica de similaridade estrutural. Como no sistema composto somente pelo sensor LiDAR não obteve-se diferenças substanciais entre os nove mapas, essa técnica foi aplicada somente no sistema composto pelo conjunto de sensores LiDAR e RGB-D.

Ao aplicar o SSIM nas imagens dos mapas, elaborou-se um comparativo, mostrado na Figura 4.9. Neste gráfico, o eixo das ordenadas representa o índice de similaridade estrutural (SSIM), que varia entre 0 e 1, sendo 0 quando não há qualquer similaridade e 1 quando há similaridade total entre os mapas. O eixo das abscissas representa os nove mapas coletados, cada um ao fim de cada volta. É importante salientar que, ao comparar o mapa de referência (mapa da volta quatro, demarcado pela barra na cor vermelha) com ele mesmo, o índice deve ser igual a 1, como foi validado nesta comparação.

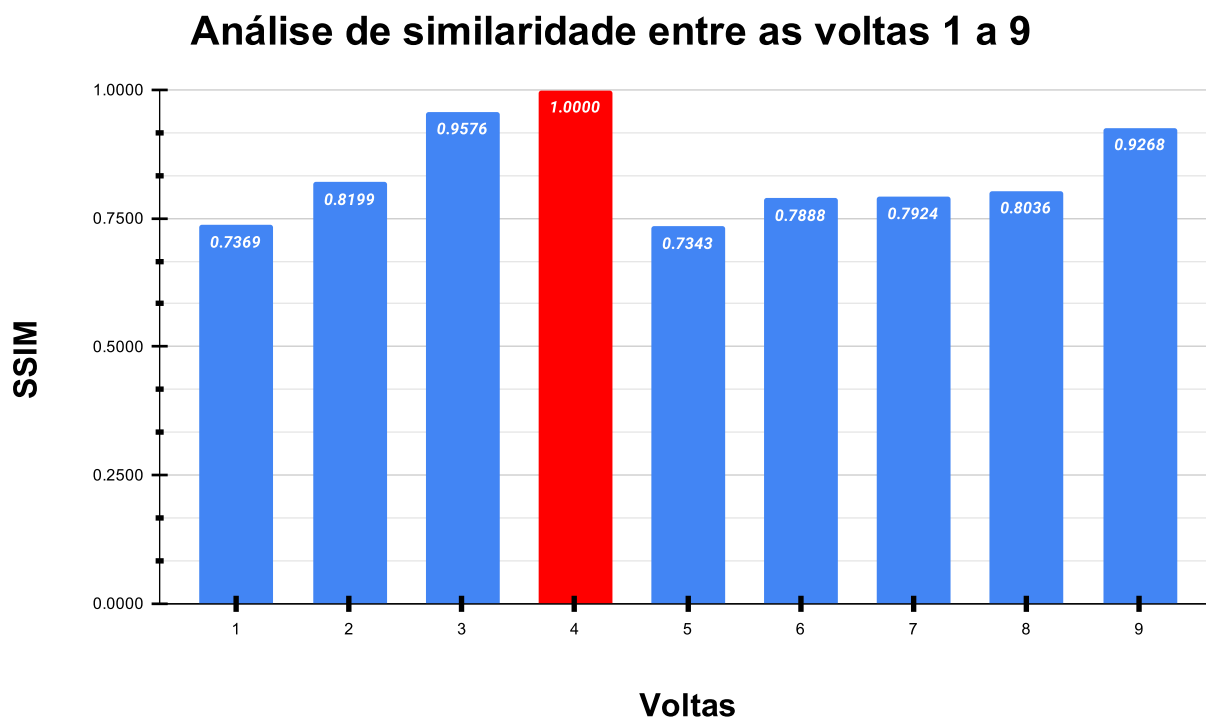


Figura 4.9: Análise de similaridade dos mapas câmera RGB-D e LiDAR utilizando o pacote RTAB-Map. Fonte: Próprio autor.

Observa-se que há uma similaridade crescente entre as voltas de 1 a 4. Se comparados com o mapa referência, o mapa da primeira volta apresenta similaridade de 0.736, o da segunda volta possui 0,8199 e o da terceira volta, o mais próximo ao mapa referência, tem similaridade de 0,9576.

Após a quarta volta, percebe-se um queda brusca no índice de similaridade, que passa a ser de 0,7343, sendo este pior do que quando comparado à primeira volta. Isto se deve, principalmente, à sobreposição de regiões no mapa. Da sexta volta em diante, o índice de similaridade volta a subir, chegando a 0,9268 na nona volta. Essa retomada da similaridade progressiva acontece porque o algoritmo do RTAB-Map começa a executar novas associações, de forma a corrigir o mapa atual, a cada nova volta.

A aplicação da técnica de similaridade estrutural, identifica não somente o nível de similaridade global entre as imagens, mas também permite uma análise da similaridade local entre as imagens, analisando regiões específicas dos mapas. Assim, foi elaborado um comparativo de similaridade local entre esses mapas, exibido na Figura 4.10 (imagens em alta resolução desta análise estarão disponíveis no Anexo A).

Observa-se, em uma escala de cinza, que, quanto mais escura a região, menor é o nível de similaridade, e quanto mais clara for a região mais ela será similar à imagem de referência. Podemos observar de forma mais precisa em quais locais houveram maiores divergências no comparativo entre mapa da volta e mapa referência. O mapa comparativo da volta referência com ela mesma não foi incluído pois a análise de similaridade local contra ele mesmo resultaria em um valor unitário (similaridade máxima) e sua análise local seria uma imagem em branco.



a) Similaridade local entre mapa 1 e mapa referência. (Similaridade global: 0.7369)



b) Similaridade local entre mapa 2 e mapa referência. (Similaridade global: 0.8199)



c) Similaridade local entre mapa 3 e mapa referência. (Similaridade global: 0.9576)



d) Similaridade local entre mapa 5 e mapa referência. (Similaridade global: 0.7343)



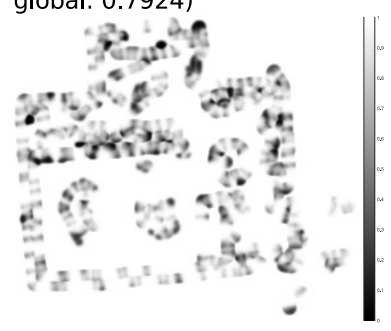
e) Similaridade local entre mapa 6 e mapa referência. (Similaridade global: 0.7888)



f) Similaridade local entre mapa 7 e mapa referência. (Similaridade global: 0.7924)



g) Similaridade local entre mapa 8 e mapa referência. (Similaridade global: 0.8036)



h) Similaridade local entre mapa 1 e mapa referência. (Similaridade global: 0.9268)

Figura 4.10: Análise de similaridade local dos mapas a), b), c), e f), g), h) e i) da Figura 4.8 contra o mapa da Figura 4.8 d). Fonte: Próprio autor.

CONCLUSÃO

O mapeamento e localização simultâneo (SLAM) por meio de robôs terrestres móveis foi e ainda é alvo de estudos de pesquisadores que queriam, principalmente, buscar soluções para o deslocamento do robô de um ponto A para um o ponto B, de forma que essa trajetória pudesse garantir que a rota fosse a mais segura o possível. As mais diversas técnicas foram publicas por renomados pesquisadores, entretanto em nenhuma delas foi possível comparar o resultado de uma técnica que utiliza sensores de natureza distintas.

Ao se utilizar o sensor LiDAR, para a realização de mapeamento, mostrou-se que, caso um robô tenha somente um sensor do tipo LiDAR, ele pode realizar um mapeamento seguro. A desvantagem de se utilizar somente este sensor é que objetos podem ficar mal representados na cena, criando uma confusão sobre quais locais são seguros ou não para deslocamento do robô, já que mesas e cadeiras ficam representadas apenas pelo seu suporte estrutural.

Com o advento de sensores RGB-D de baixo custo, como o sensor Microsoft Kinect, uma nova forma de explorar o mundo foi possível, pois agora, além de informações em RGB, é possível compreender a tridimensionalidade por meio do sensor de profundidade. Esta capacidade de aquisição de imagens, aliada ao seu baixo custo, fez com que pesquisadores tivessem interesse na capacidade de aplicação do sensor RGB-D, inclusive em conjunto com outros sensores, como no caso dos sensores do tipo LiDAR. Uma das técnicas de SLAM amplamente utilizada é o RTAB-Map, um SLAM baseado em grafos com uma grande capacidade de integração entre sensores tipo RGB-D e LiDAR.

Neste estudo, objetivou-se utilizar um robô terrestre móvel (Husky A200), associado a sensores de percepção do ambiente (LiDAR e RGB-D) para produzir um mapeamento *indoor* que, apesar de bidimensional, levasse em consideração a complexidade tridimensional do ambiente e dos objetos nele contidos. Foram produzidos mapas provenientes de ambiente simulado e ambiente real, cada um utilizando duas formas diferentes para mapeamento: somente o sensor LiDAR e o conjunto de sensores LiDAR e RGB-D.

Os resultados dos mapas de ocupação adquiridos mostraram que há poucas diferenças entre os mapeamentos produzidos no ambiente simulado e no ambiente real. Também

observou-se que, ao se utilizar somente o sensor LiDAR, um mapa coerente é produzido, sem diferenças significativas entre as voltas. Entretanto, nesse mapeamento, objetos inseridos no ambiente não puderam ser completamente caracterizados, sendo representados apenas por seus suportes estruturais. Isso pode representar um perigo à estrutura física de robôs terrestres móveis que utilizam esse modo de mapeamento para planejamento de rotas.

Constata-se, ainda, que o RTAB-Map foi capaz de compreender a tridimensionalidade do ambiente, produzindo um mapeamento através do conjunto de sensores LiDAR e RGB-D. Nesse mapeamento, objetos como mesas e cadeiras passam a ser representados pela sua forma geométrica e não apenas pelos seus suportes estruturais, o que garante maior confiabilidade no mapa, permitindo que o planejamento de rotas seja mais seguro e eficiente para a locomoção de robôs terrestres móveis, tanto de forma autônoma, quanto de forma teleoperada.

No ambiente do Laboratório de Robótica da UFBA, o RTAB-Map, utilizando o robô terrestre móvel Husky A200 e o conjunto de sensores LiDAR (LMS111) e RGB-D (*Microsoft Kinect*), teve seu melhor desempenho ao realizar quatro voltas. Pode-se observar que, na quinta volta, houve uma piora na qualidade do mapa. Isso sugere que realizar mais voltas não vai garantir um mapa que melhor represente a realidade. Entretanto, caso haja alguma associação incorreta dos dados durante o processo de mapeamento, dar mais voltas pode melhorar a qualidade do mapa que apresentou erros, como pudemos observar na nona volta, na qual a qualidade do mapa volta a ficar substancial em relação a todas as outras.

O índice de similaridade apontou que ao fim da primeira volta, o mapa coletado era de 73,69% similar ao mapa de referência, enquanto o maior índice aconteceu com o mapa relativo à terceira volta, obtendo um valor de 95,76% similaridade. O pior índice alcançado foi de 73,43%, um valor que ocorreu na volta posterior ao valor de referência (100%). Apesar de ter ocorrido logo após ao mapa de referência, este valor não decaiu sucessivamente, o que implicaria em uma degeneração do método. Ao contrário, eu foi sendo corrigido e na nona volta atingiu um índice de similaridade de 92,68%. Apontando uma robustez para correção de erros de ruído inseridos pelos sensores.

Em termos comparativos entre a utilização dos sistema do sensor LiDAR e do conjunto RGB-D e LiDAR, observa-se que, quando utiliza-se somente o sensor LiDAR, há menor divergência na análise métrica comparativa entre o mapa gerado e o *ground truth*. Porém essa divergência continua sendo baixa no conjunto RGB-D e LiDAR, o que não inviabiliza sua utilização.

Apesar de ser capaz de realizar a atividade de mapeamento, o agrupamento de sensores LiDAR e RGB-D ainda permite que regiões não sejam mapeadas, o que para algumas circunstâncias pode ser crucial no planejamento de rotas de robôs terrestres móveis. Esta limitação se dá devido ao campo de visão que o sensor RGB-D possui, pois é através deste que o mapa é iterativamente construído. Para suprir esta lacuna, pode-se sugerir a inclusão de mais um sensor RGB-D que, ao ser combinado com o outro sensor RGB-D, ofereçam um campo de visão mais amplo, permitindo que o mapa seja construído de forma mais rápida e sem a necessidade de percorrer várias vezes o mesmo percurso ou visitar um local.

Assim conclui-se que a utilização de sensores RGB-D em robôs terrestres móveis agrega uma grande quantidade de informação ao sistema robótico, permitindo que este desenvolva melhores ações sobre o ambiente no qual está inserido. No que tange à atividade de mapeamento, incorporar um sensor RGB-D permite que os objetos de diferentes alturas e formatos possam estar bem representados em um mapa de ocupação bidimensional. Os valores de similaridade apontaram que o método foi capaz de realizar um bom mapeamento com apenas uma volta pelo ambiente, o que, para robôs terrestres com baixa capacidade de suprimento energia, fornece uma boa estimativa inicial sobre o ambiente inserido.

5.1 PUBLICAÇÃO

Este trabalho de mestrado gerou a seguinte publicação:

- MELLO, RICARDO S.; DORIA, NARA S. F. ; CONCEIÇÃO, ANDRÉ G. S. ; FARIAS, PAULO C. M. A. . Qualitative study between two mapping methods RTAB-MAP and HECTOR SLAM applied to probabilistic roadmap path planning. In: III Brazilian Humanoid Robot Workshop (BRAHUR) and the IV Brazilian Workshop on Service Robotics (BRASERO), 2020, Online. Proceedings of the III Brazilian Humanoid Robot Workshop (BRAHUR) and IV Brazilian Workshop on Service Robotics (BRASERO), 2020.

Resumo: A ação autônoma de um robô esta diretamente relacionada com a forma que este percebe o ambiente ao seu redor. Para isso, utilizam-se diversos tipos de sensores como: ultrassom, *LiDAR*, câmeras monoculares RGB e RGB-D. Para a modelagem geométrica de um ambiente, grades de ocupação (do inglês *occupancy grid*) foram desenvolvidas para representar a bidimensionalidade do ambiente que o robô está localizado. Este trabalho faz uso de sensores *LiDAR* e RGB-D, da plataforma robótica virtual Husky A200 e do *framework* ROS para a elaboração de mapas bidimensionais. Utilizando este conjunto, foi elaborada uma simulação computacional na qual os sensores produziram mapas de ocupação e posteriormente os métodos simulados foram implementados em um robô real. Os mapas foram analisados e comparados com um *ground truth*. O mapeamento realizado pelo sensor a *laser* produziu o menor desvio padrão. Entretanto, os objetos contidos no ambiente simulado não foram completamente representados. Quando testou-se o sistema utilizando o sensor RGB-D associado a um sensor à *laser*, este teve um desvio padrão maior, porém seu desempenho para representar os objetos contidos no ambiente foi superior ao *laser* isoladamente.

A.1 MAPAS EXPERIMENTO SIMULAÇÃO 4.1.1

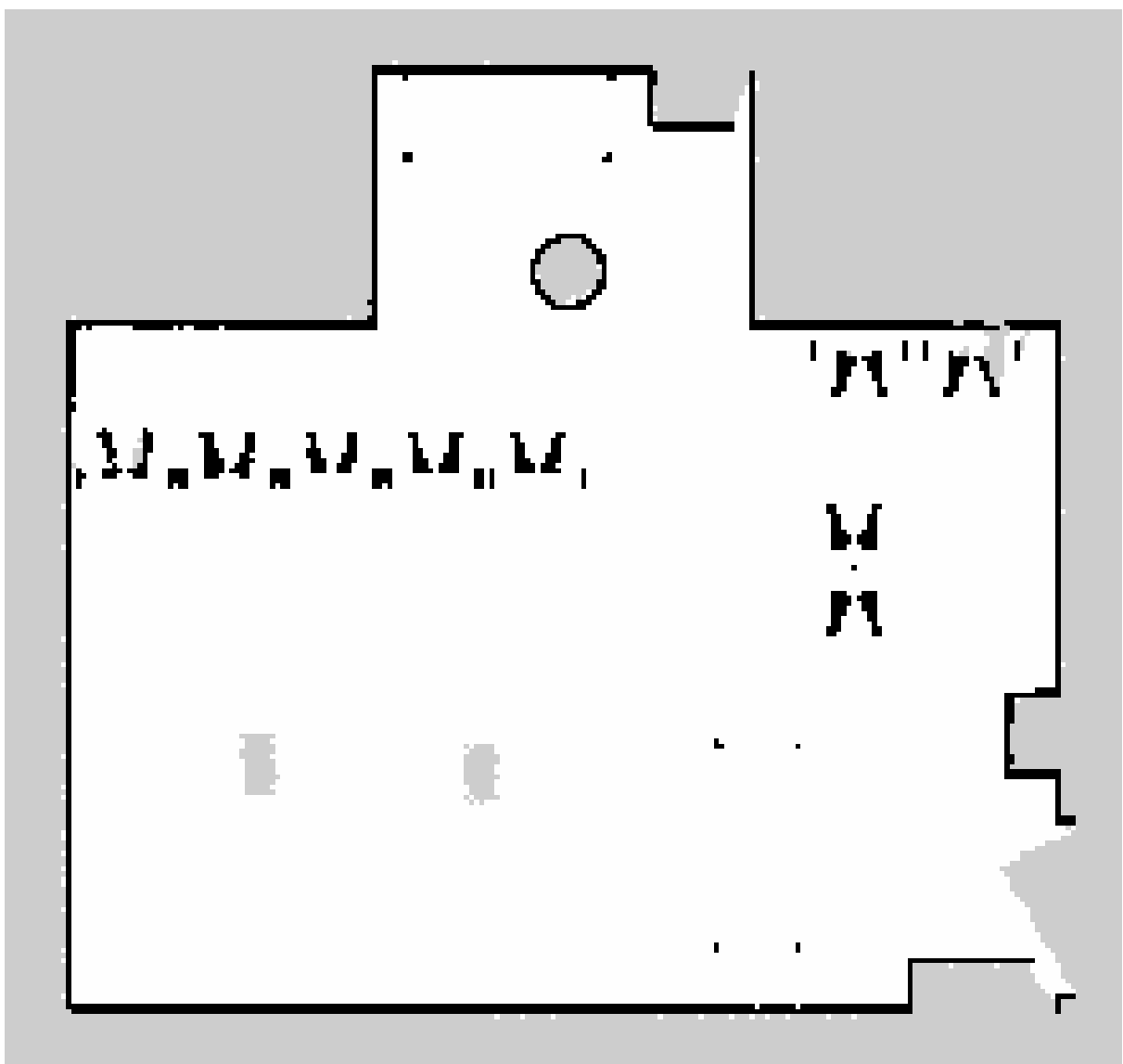


Figura A.1: Mapa da primeira volta em simulação usando sensor LiDAR e pacote HECTOR SLAM. Fonte: Próprio autor.

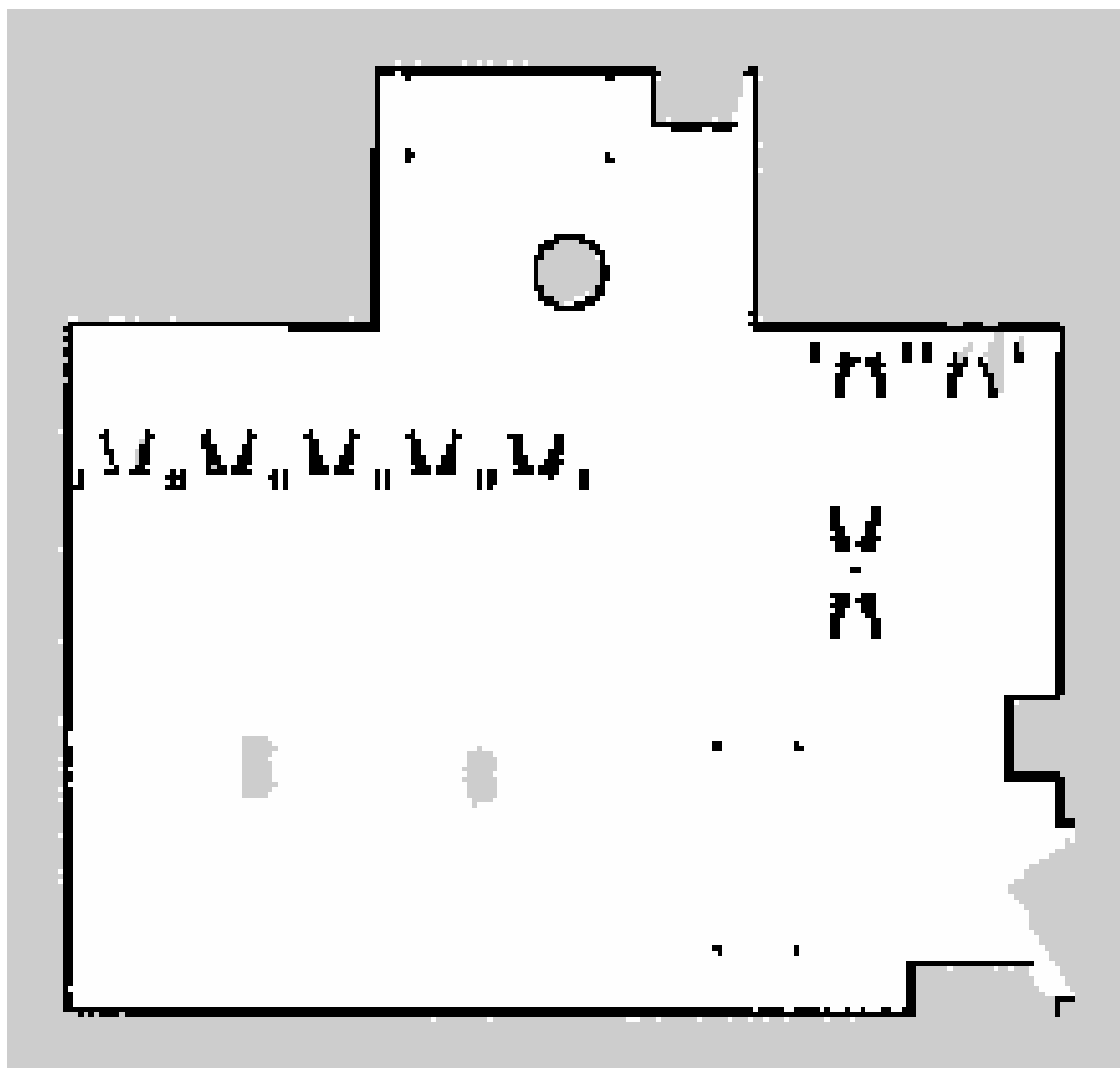


Figura A.2: Mapa da segunda volta em simulação usando sensor LiDAR e pacote HECTOR SLAM. Fonte: Próprio autor.

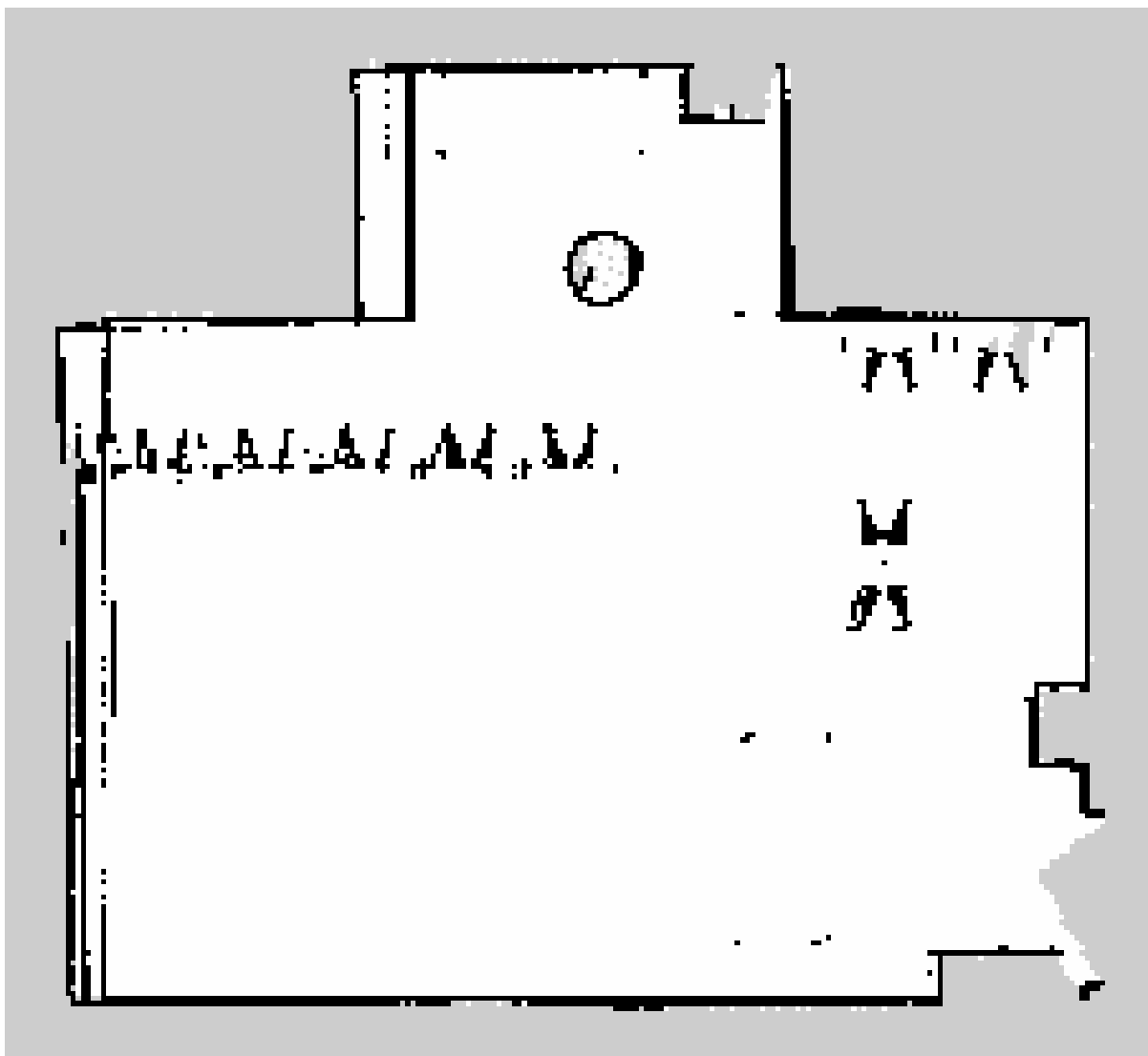


Figura A.3: Mapa da terceira volta em simulação usando sensor LiDAR e pacote HECTOR SLAM. Fonte: Próprio autor.

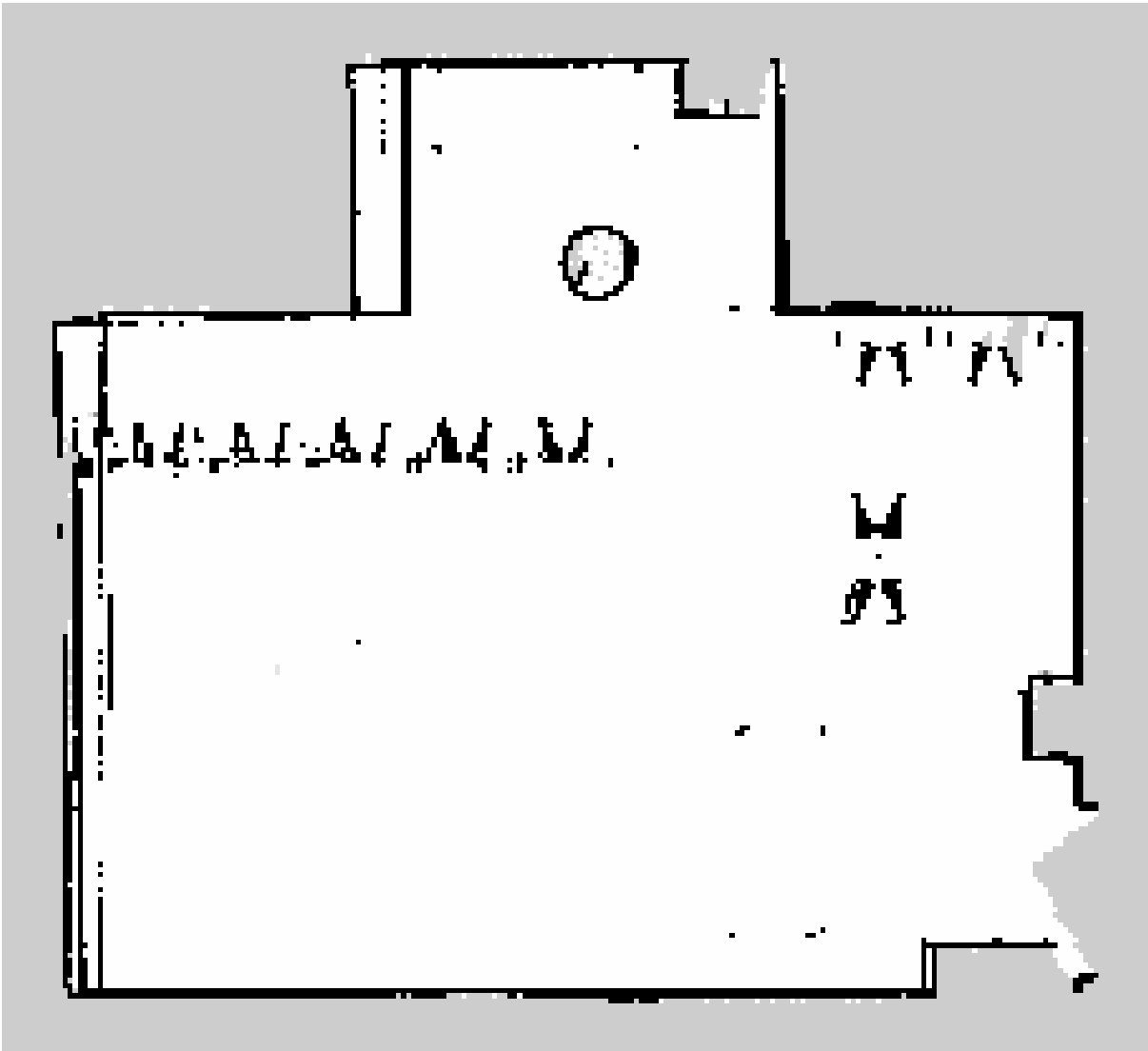


Figura A.4: Mapa da quarta volta em simulação sensor usando LiDAR e pacote HECTOR SLAM. Fonte: Próprio autor.

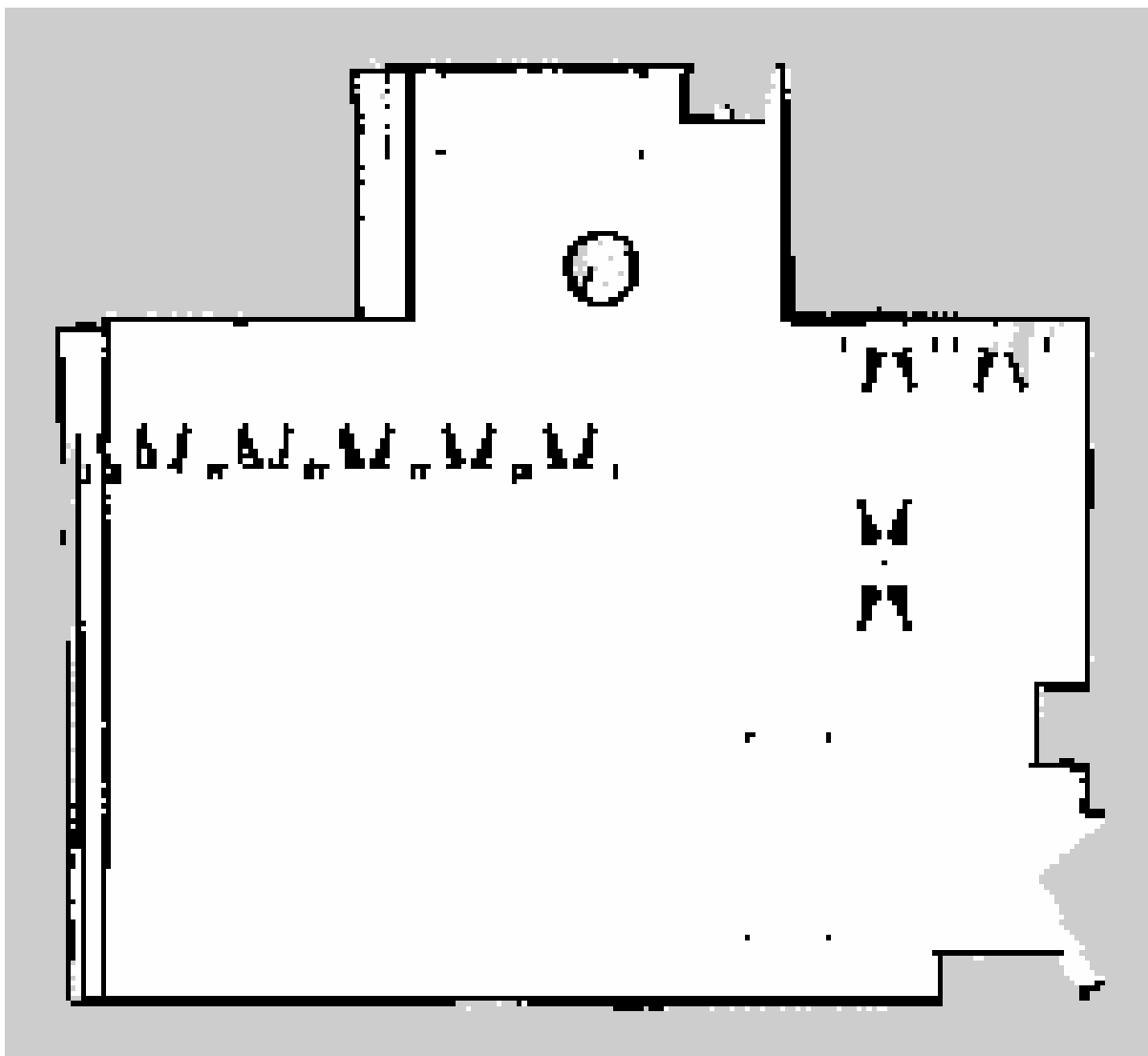


Figura A.5: Mapa da quinta volta em simulação usando sensor LiDAR e pacote HECTOR SLAM. Fonte: Próprio autor.

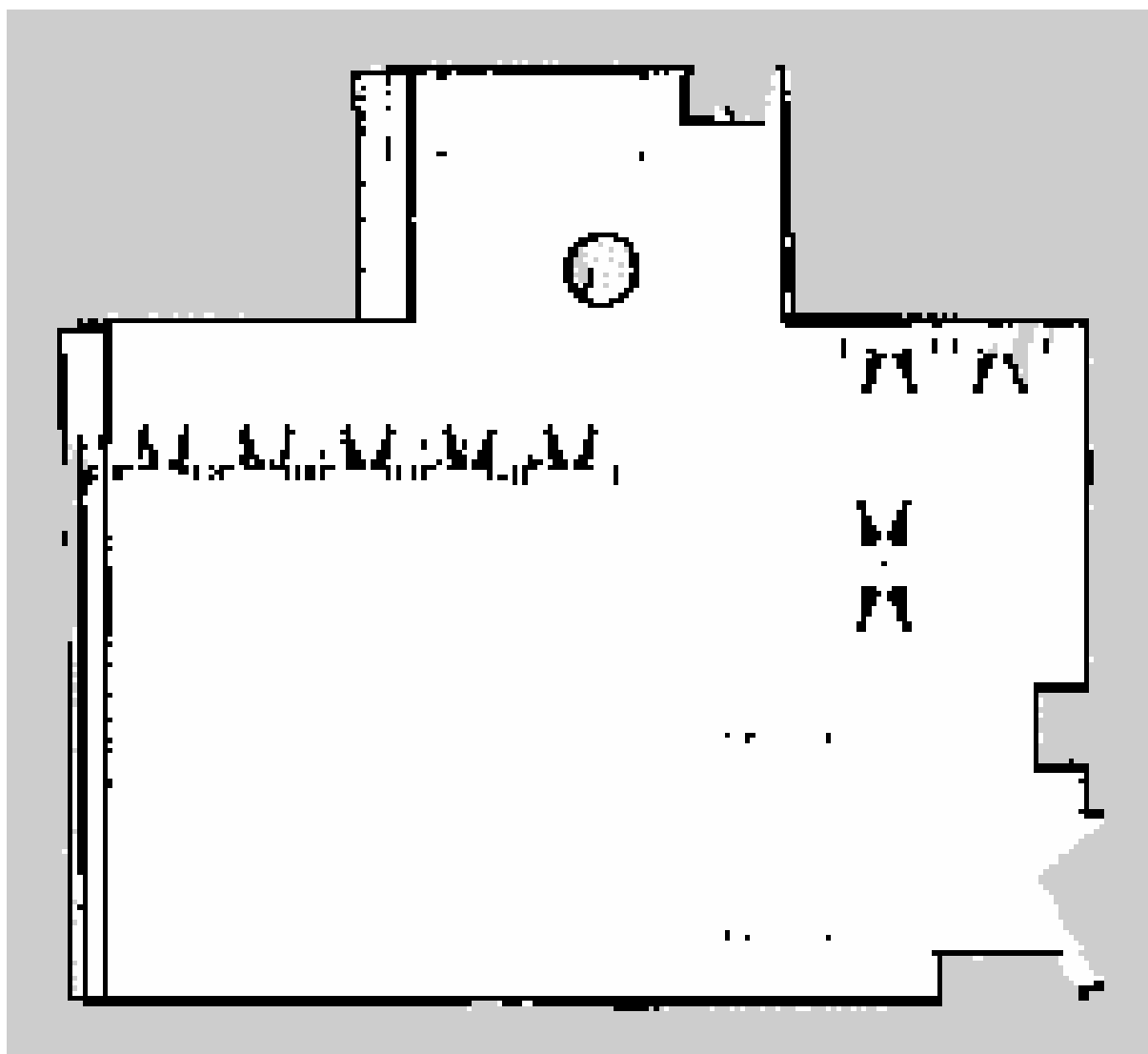


Figura A.6: Mapa da sexta volta em simulação usando sensor LiDAR e pacote HECTOR SLAM. Fonte: Próprio autor.

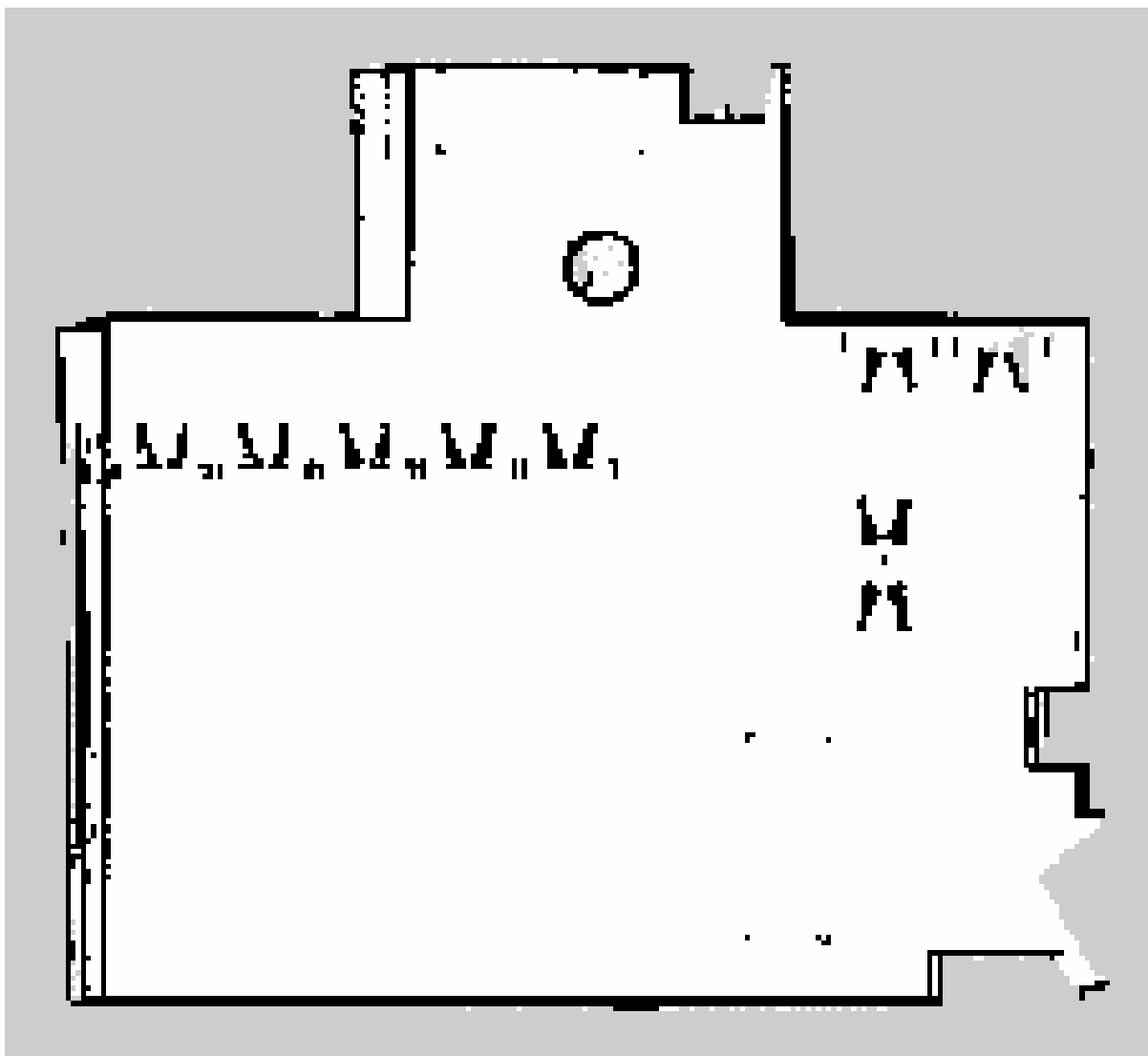


Figura A.7: Mapa da sétima volta em simulação usando sensor LiDAR e pacote HECTOR SLAM. Fonte: Próprio autor.

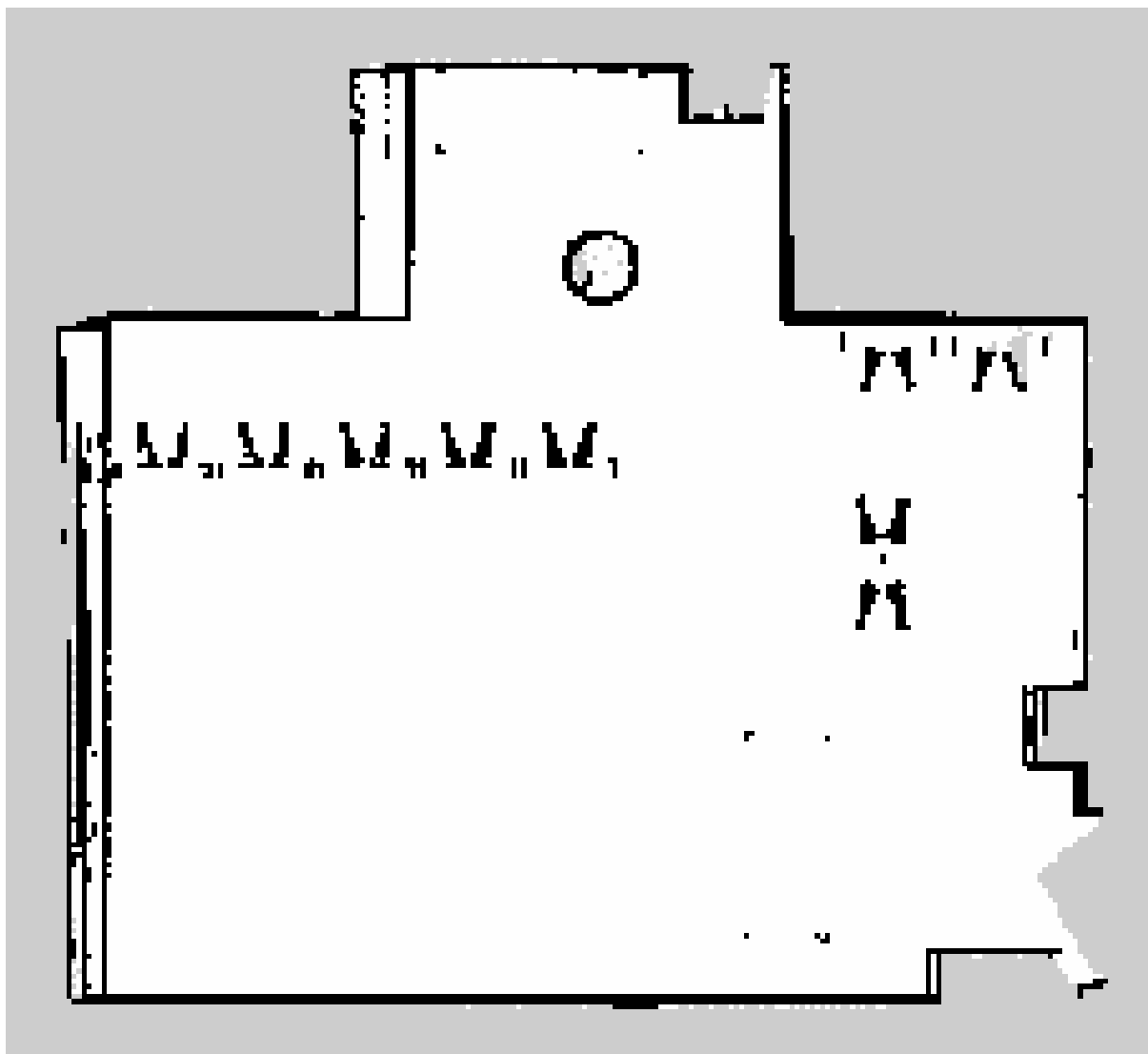


Figura A.8: Mapa da oitava volta em simulação sensor LiDAR e pacote HECTOR SLAM.
Fonte: Próprio autor.

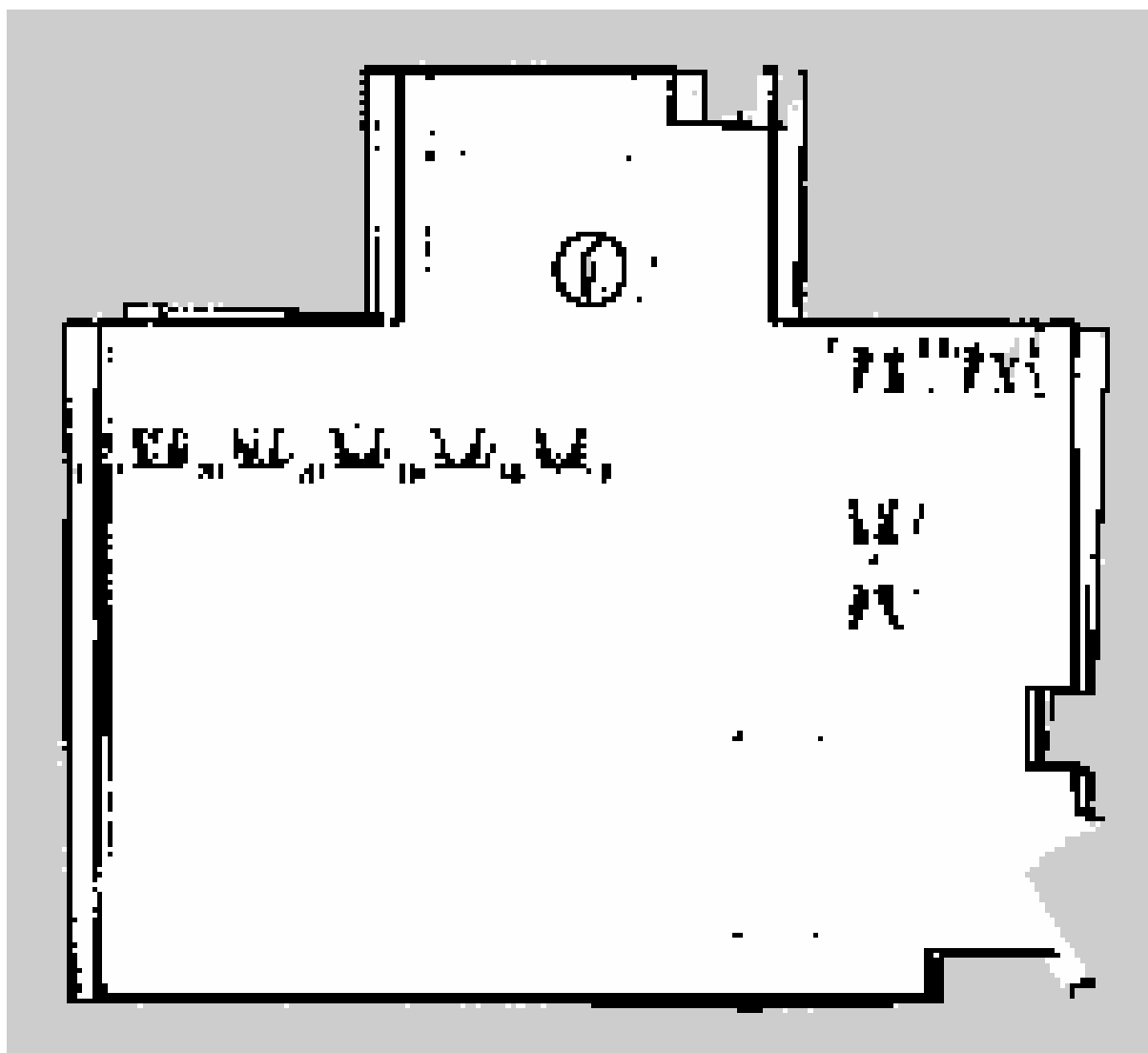


Figura A.9: Mapa da nona volta em simulação sensor LiDAR e pacote HECTOR SLAM.
Fonte: Próprio autor.

A.2 MAPAS EXPERIMENTO SIMULAÇÃO 4.1.2



Figura A.10: Mapa da primeira volta em simulação usando os sensores LiDAR + RGB-D e pacote RTAB-Map. Fonte: Próprio autor.



Figura A.11: Mapa da segunda volta em simulação usando os sensores LiDAR + RGB-D e pacote RTAB-Map. Fonte: Próprio autor.

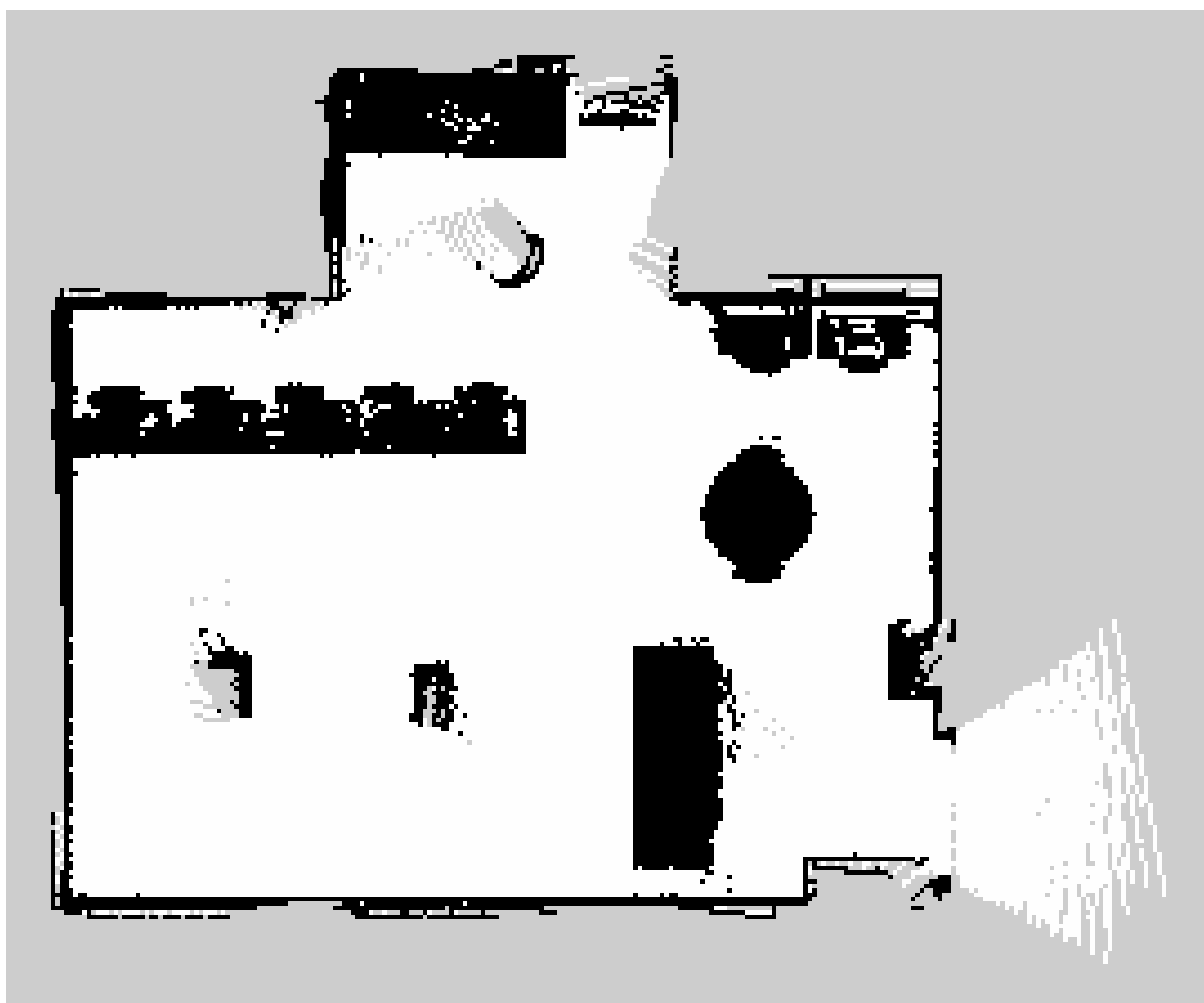


Figura A.12: Mapa da terceira volta em simulação usando os sensores LiDAR + RGB-D e pacote RTAB-Map. Fonte: Próprio autor.

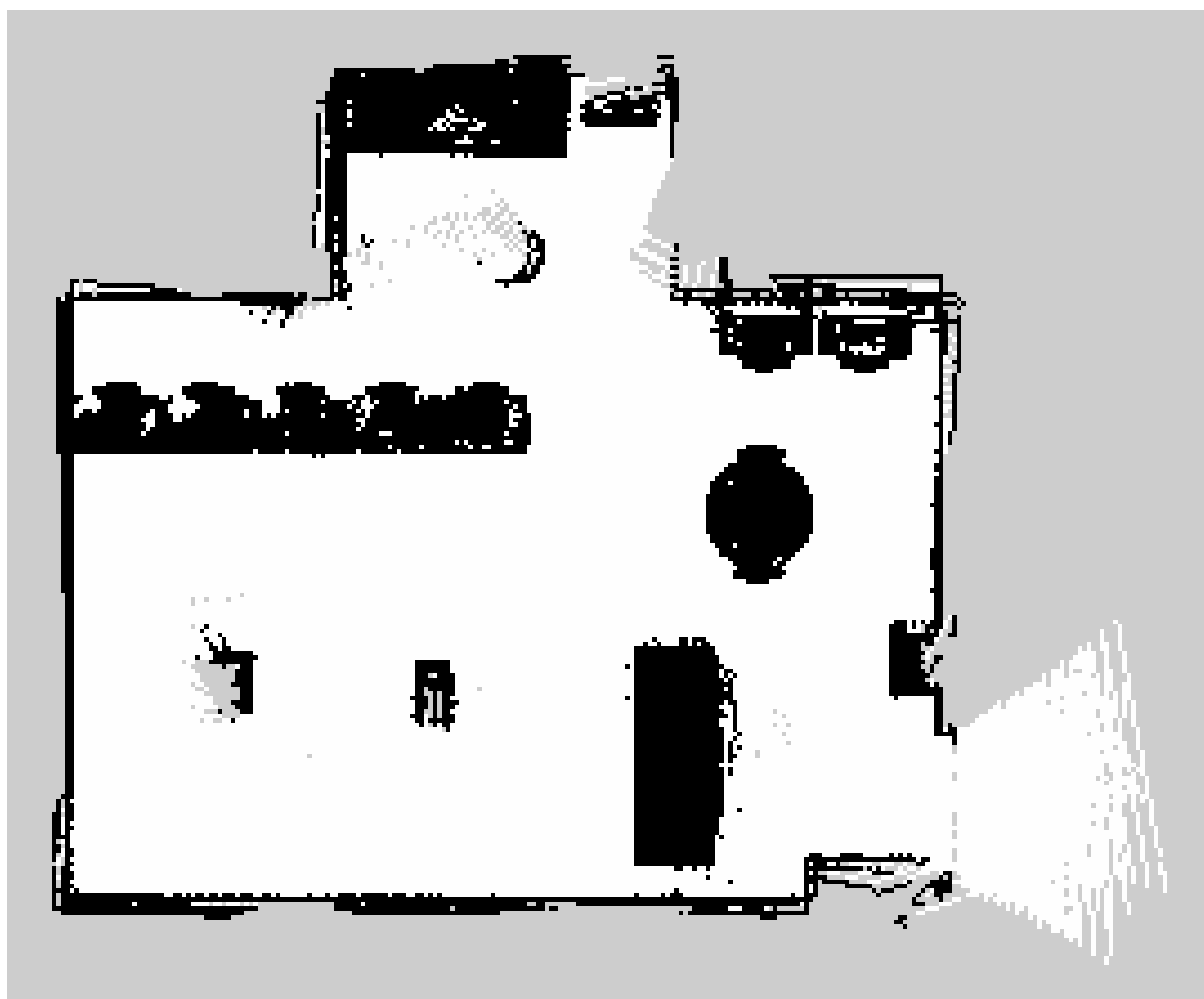


Figura A.13: Mapa da quarta volta em simulação usando os sensores LiDAR + RGB-D e pacote RTAB-Map. Fonte: Próprio autor.

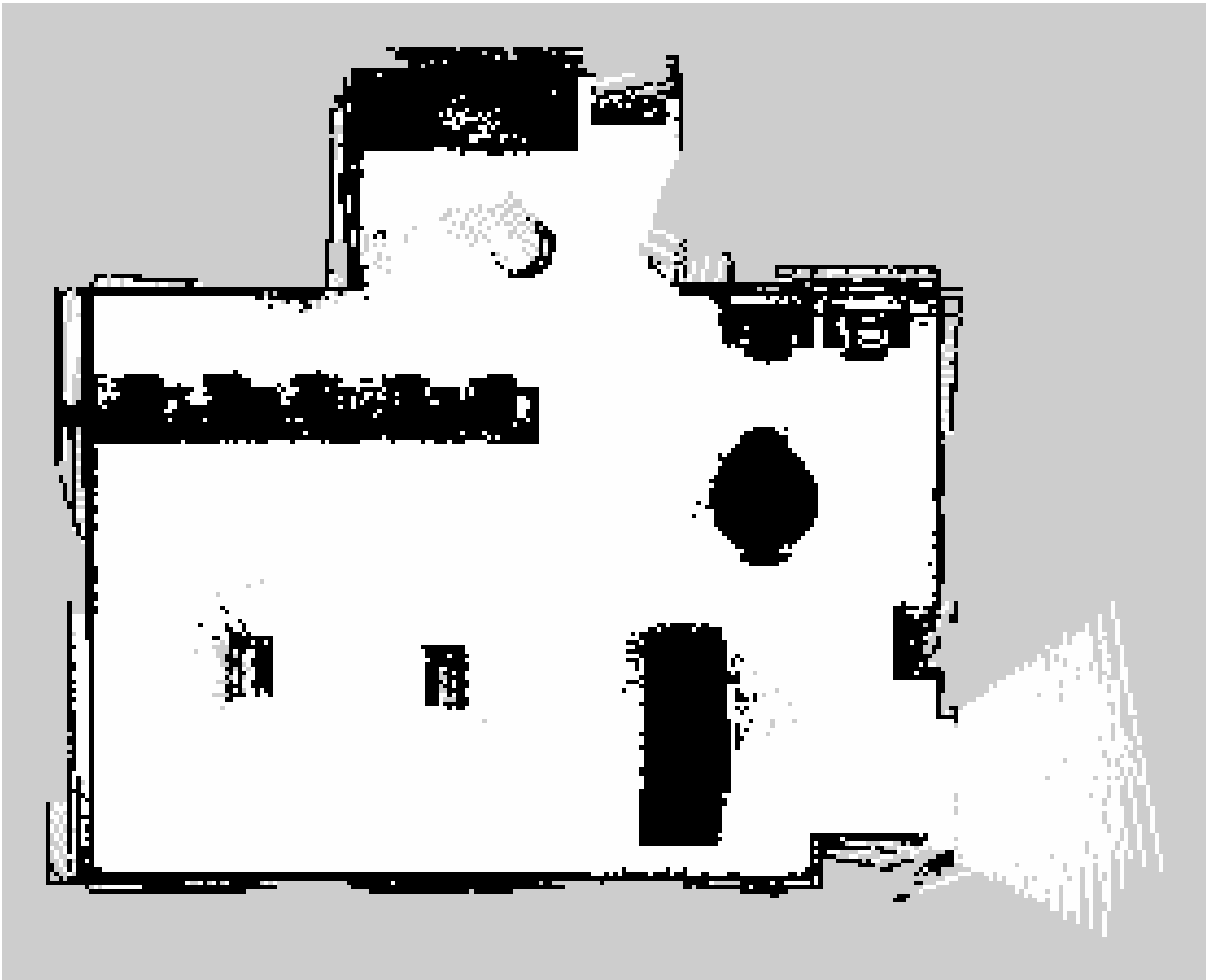


Figura A.14: Mapa da quinta volta em simulação usando os sensores LiDAR + RGB-D e pacote RTAB-Map. Fonte: Próprio autor.



Figura A.15: Mapa da sexta volta em simulação usando os sensores LiDAR + RGB-D e pacote RTAB-Map. Fonte: Próprio autor.

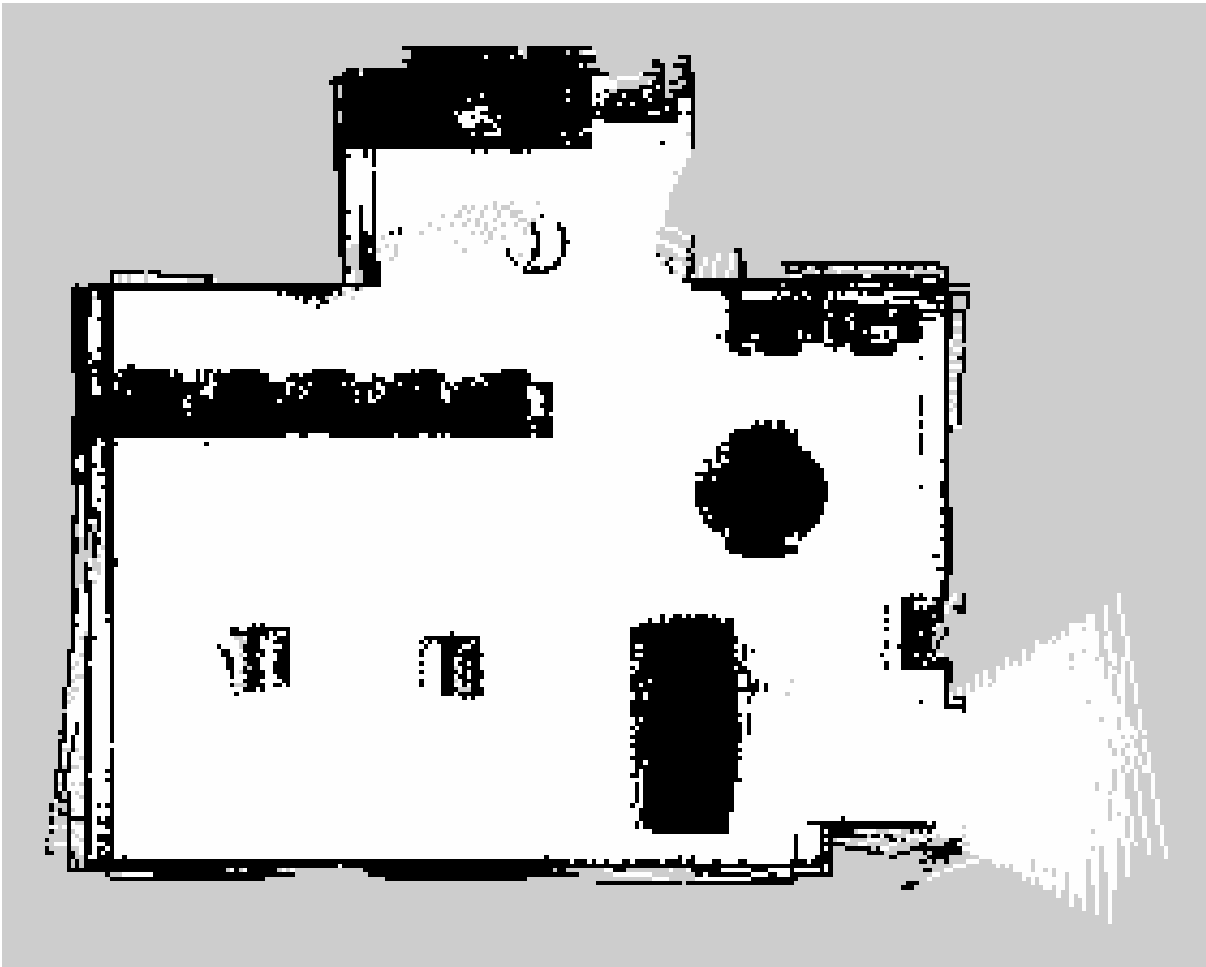


Figura A.16: Mapa da sétima volta em simulação usando os sensores LiDAR + RGB-D e pacote RTAB-Map. Fonte: Próprio autor.

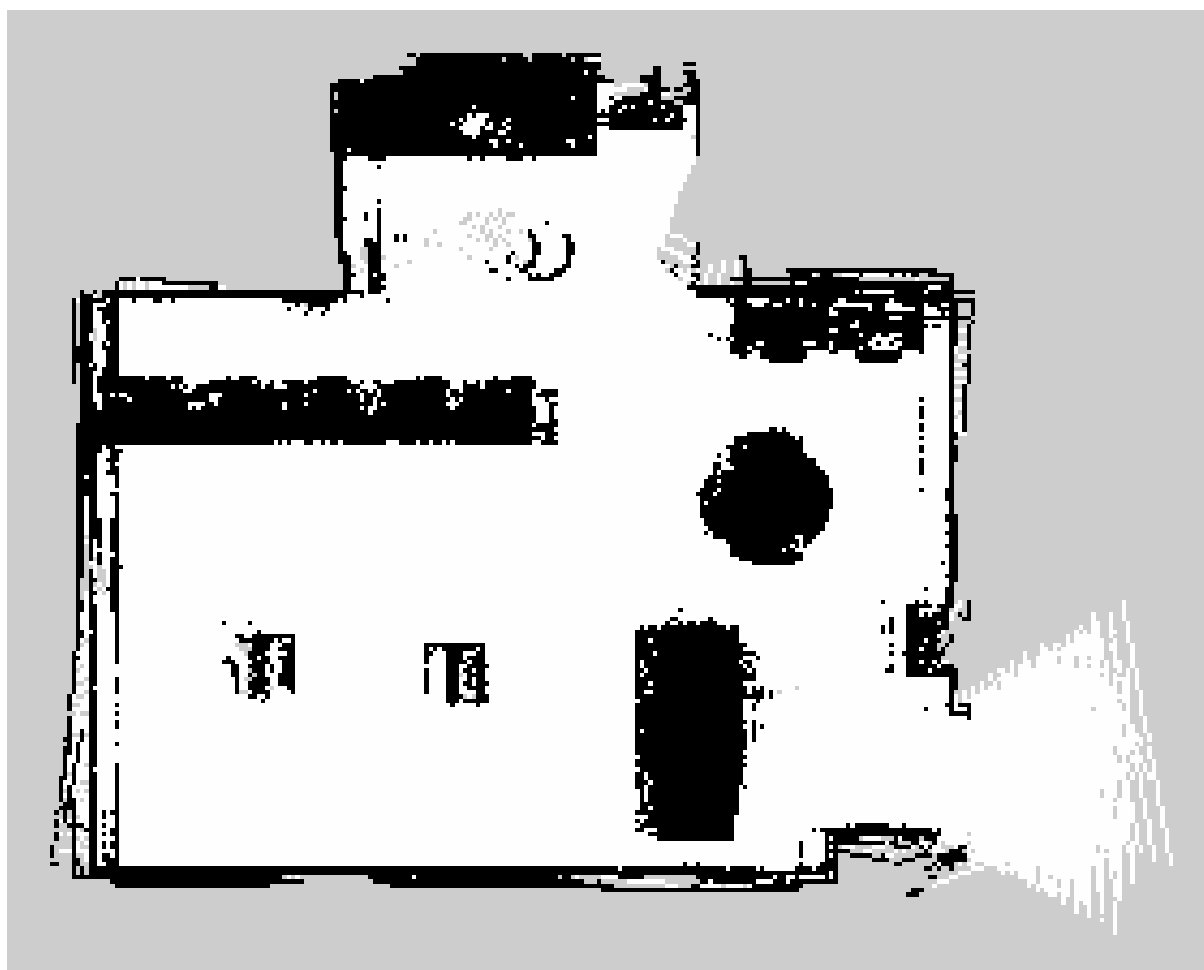


Figura A.17: Mapa da oitava volta em simulação usando os sensores LiDAR + RGB-D e pacote RTAB-Map. Fonte: Próprio autor.

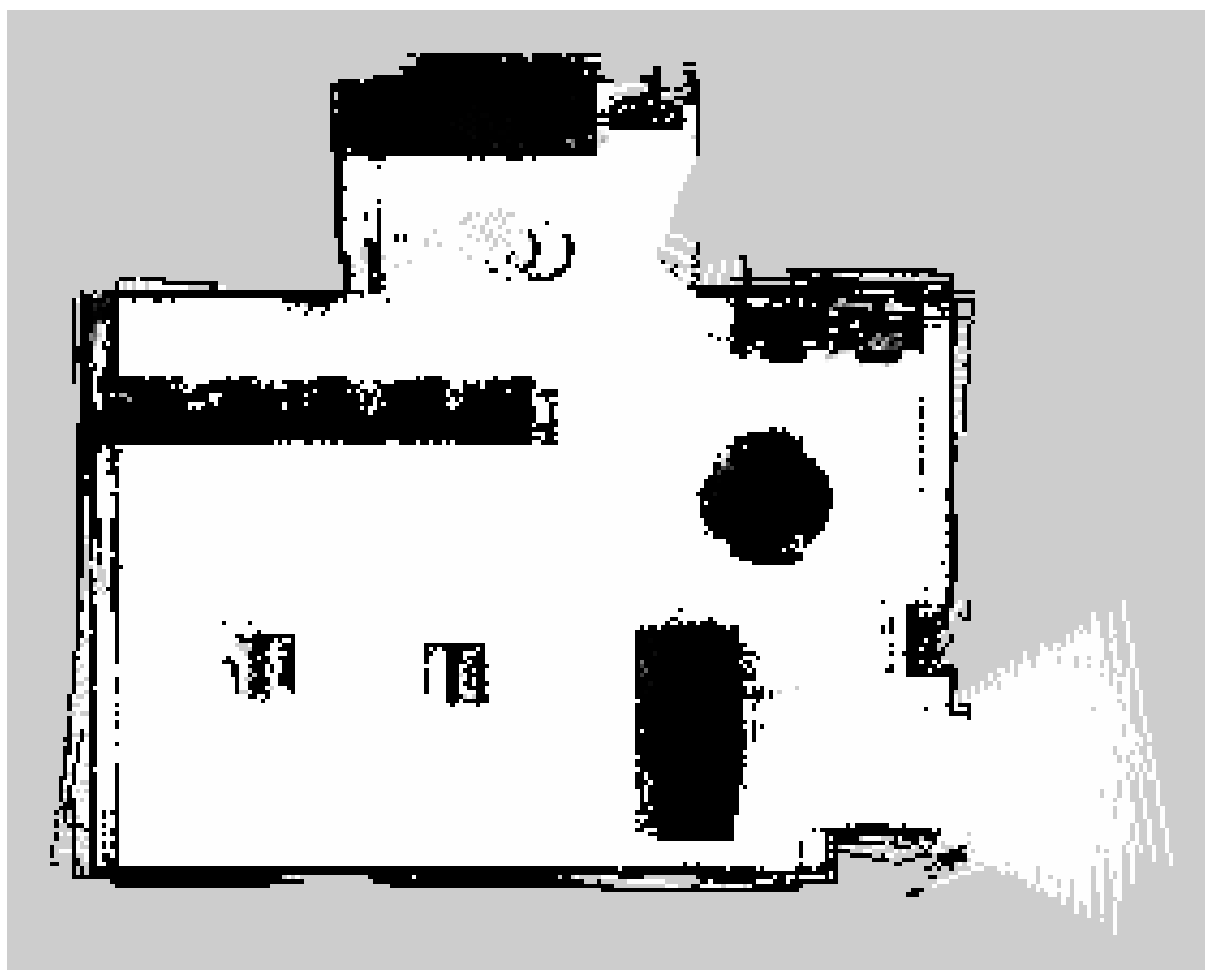


Figura A.18: Mapa da nona volta em simulação usando os sensores LiDAR + RGB-D e pacote RTAB-Map. Fonte: Próprio autor.

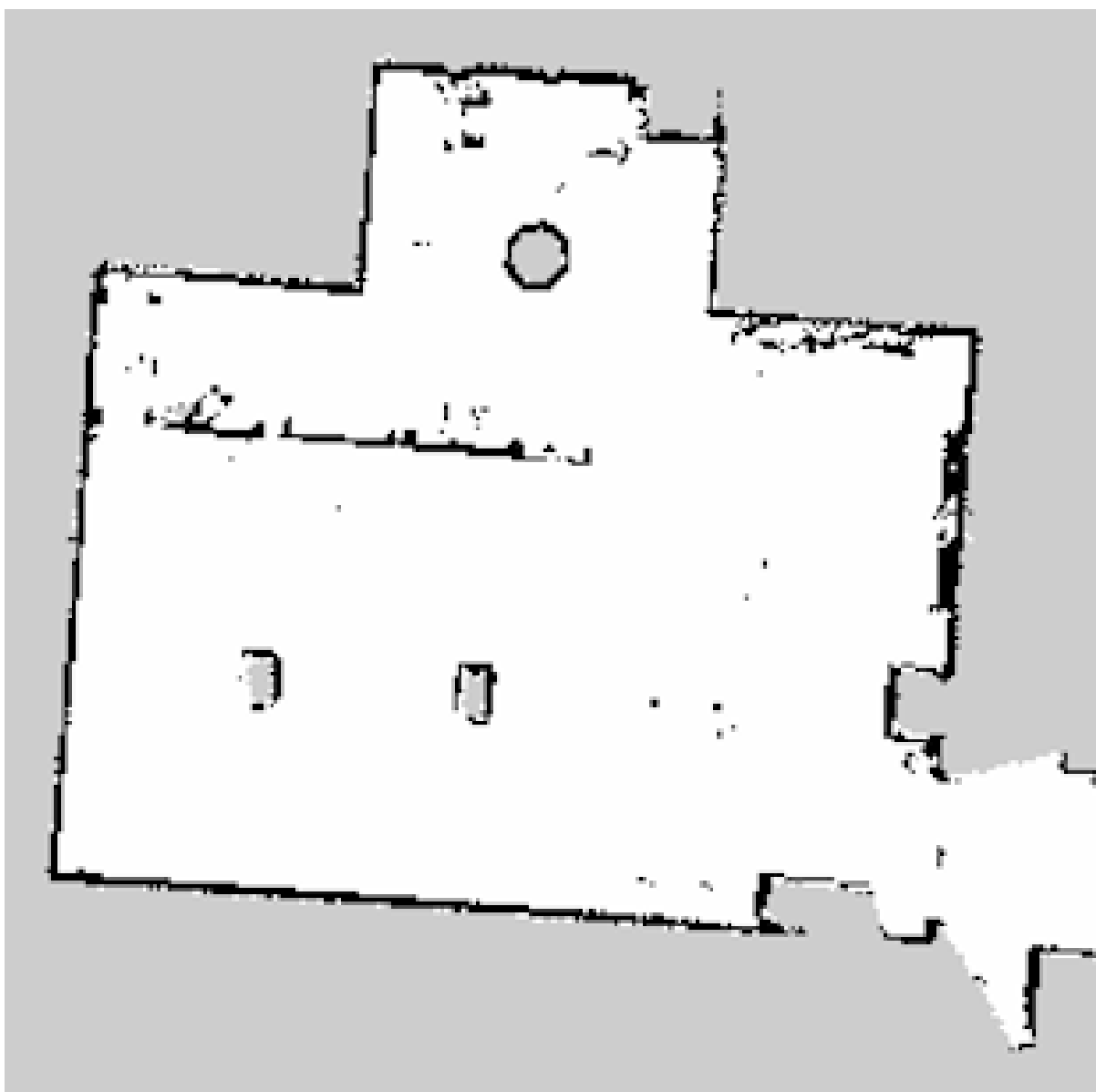
A.3 MAPAS EXPERIMENTO REAL 4.2.1

Figura A.19: Mapa da primeira volta em ambiente real usando sensor LiDAR e pacote HECTOR SLAM. Fonte: Próprio autor.

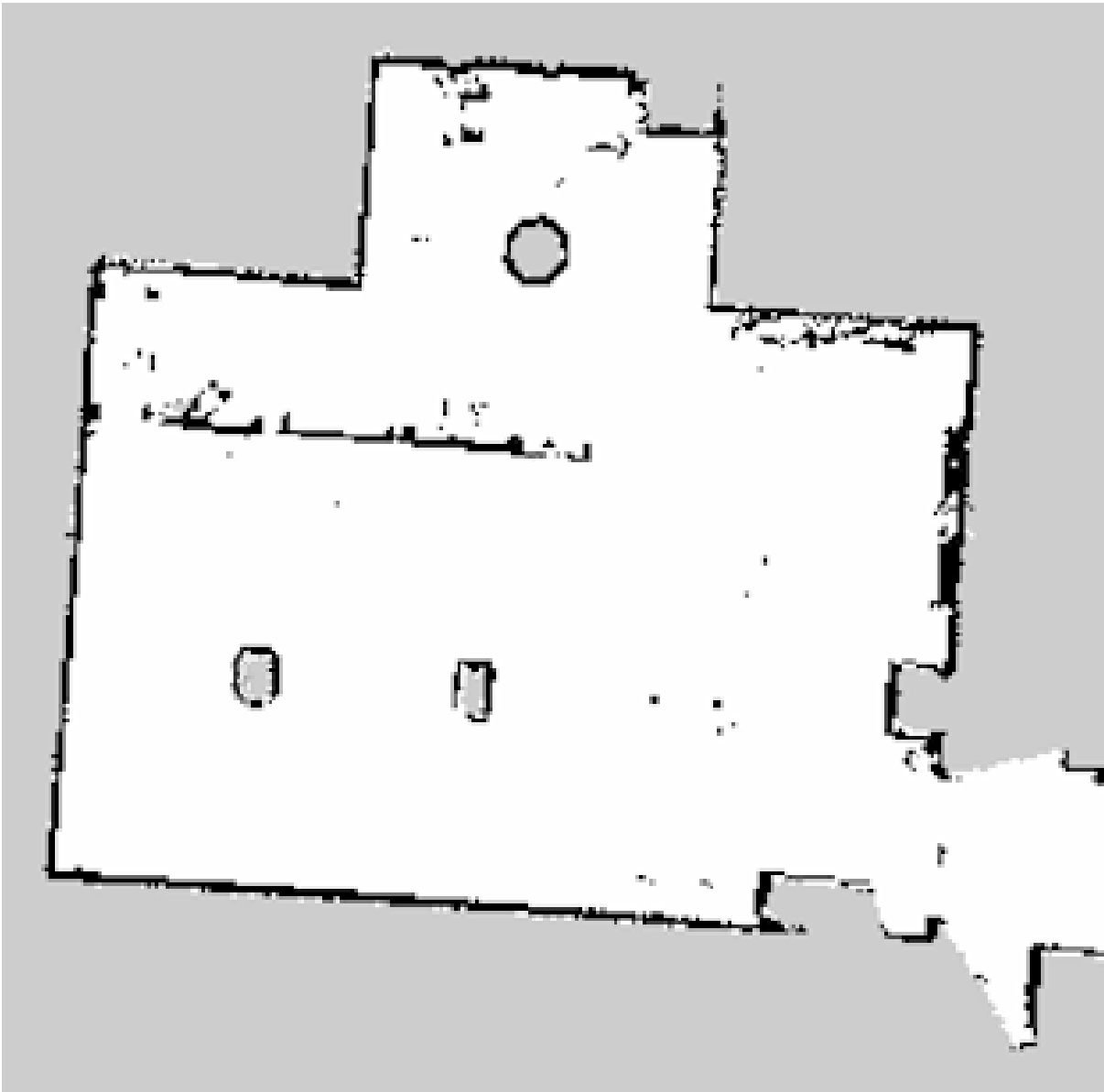


Figura A.20: Mapa da segunda volta em ambiente real usando sensor LiDAR e pacote HECTOR SLAM. Fonte: Próprio autor.

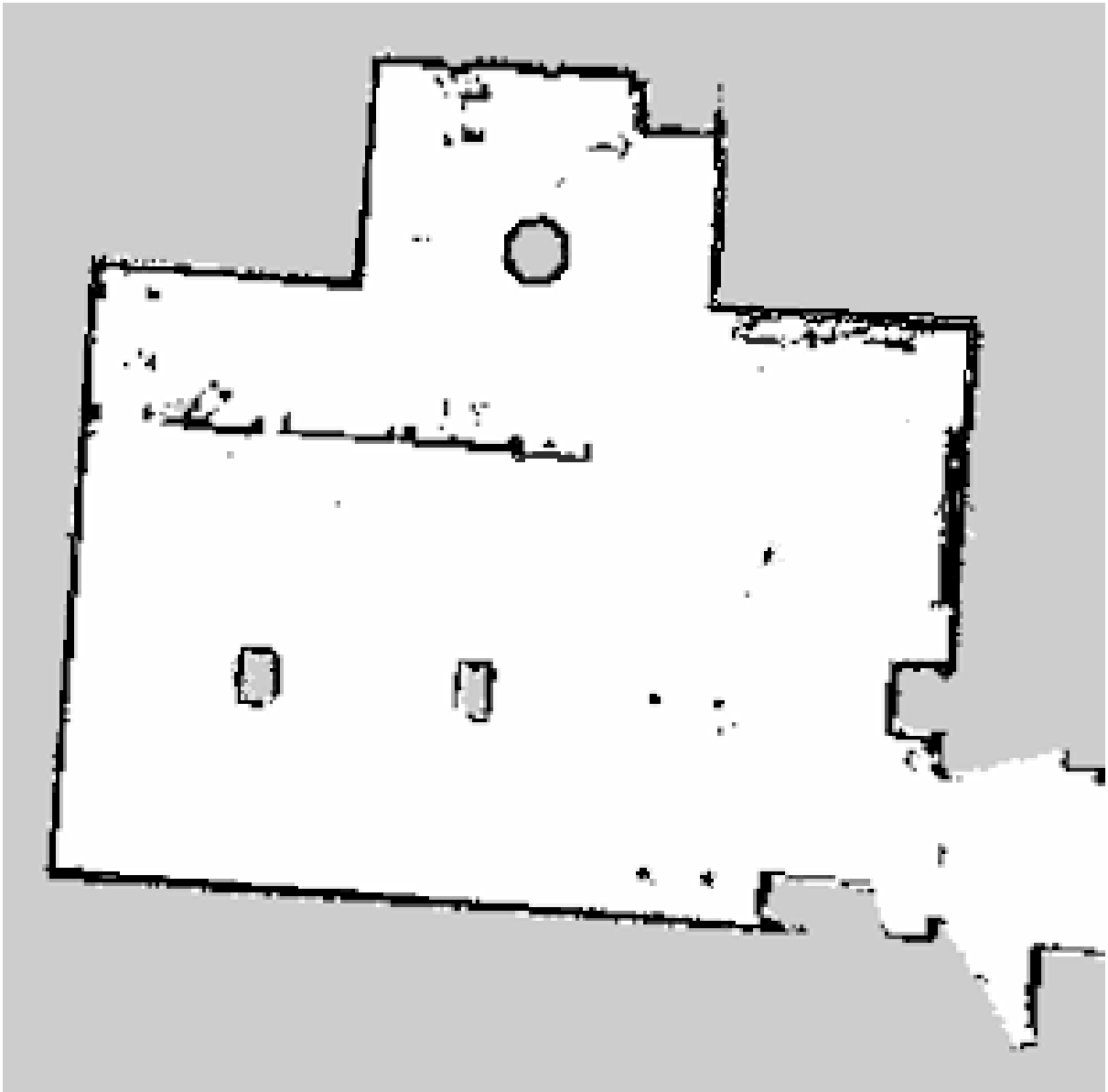


Figura A.21: Mapa da terceira volta em ambiente real usando sensor LiDAR e pacote HECTOR SLAM. Fonte: Próprio autor.

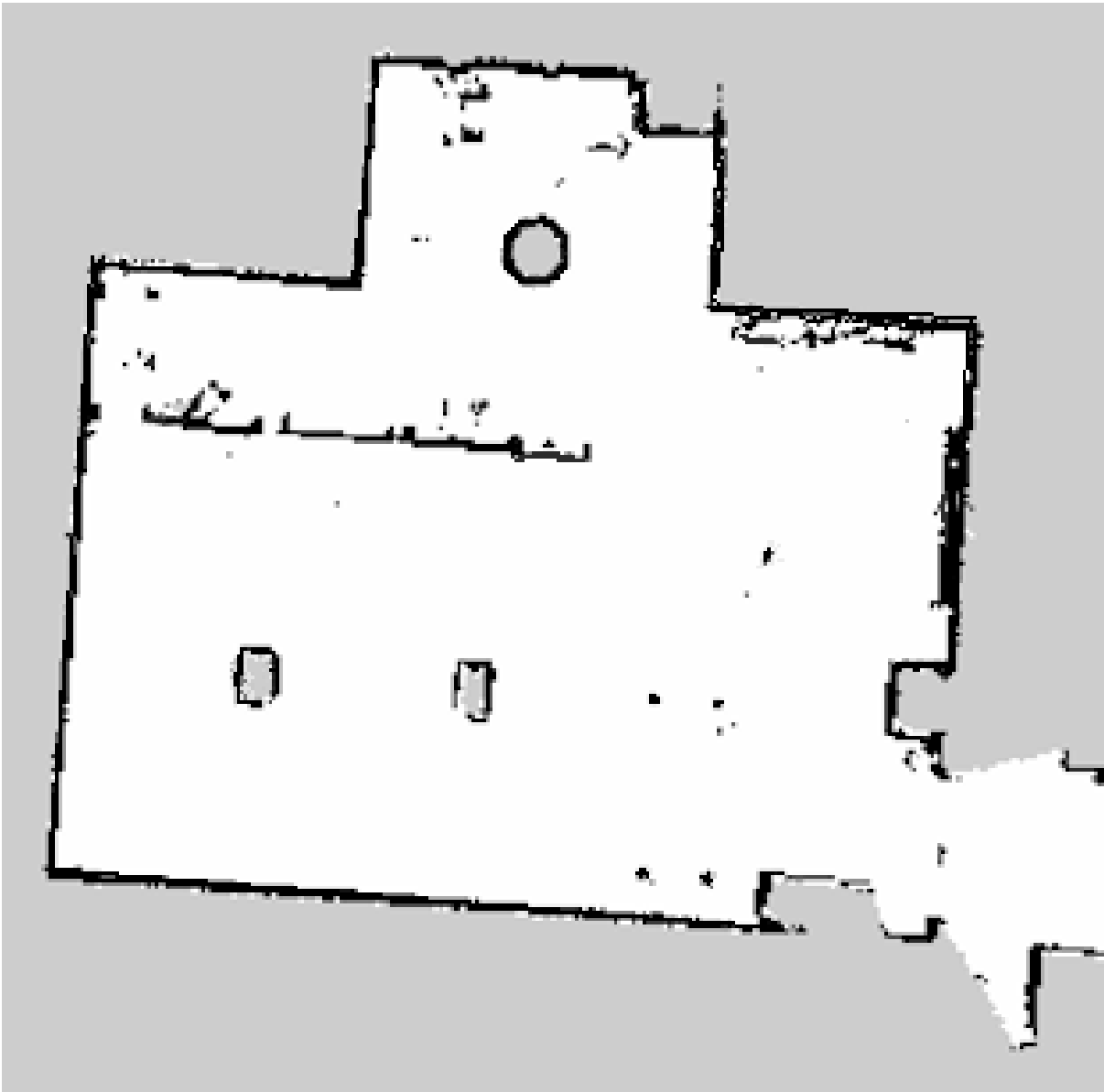


Figura A.22: Mapa da quarta volta em ambiente real usando sensor LiDAR e pacote HECTOR SLAM. Fonte: Próprio autor.

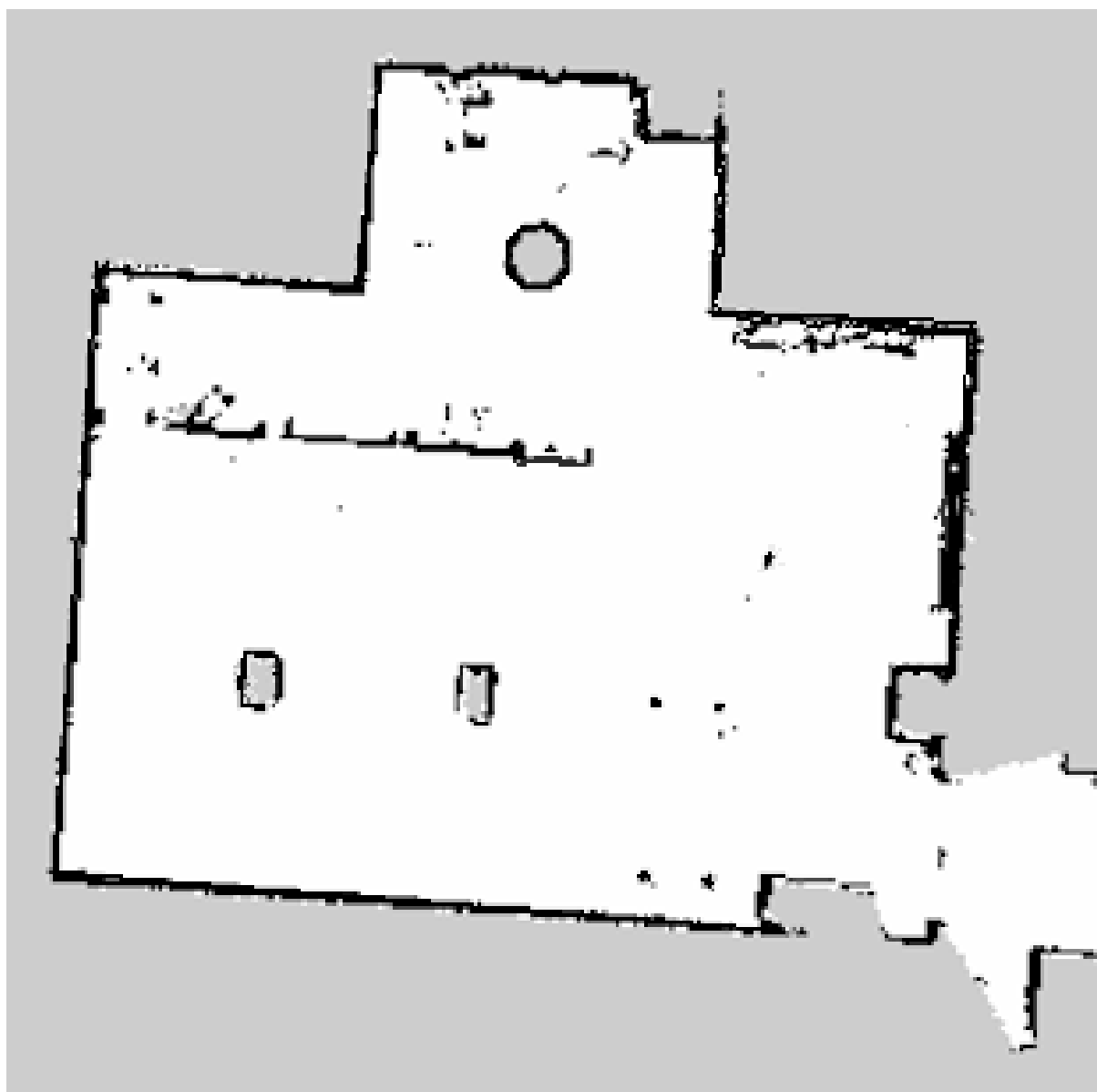


Figura A.23: Mapa da quinta volta em ambiente real usando sensor LiDAR e pacote HECTOR SLAM. Fonte: Próprio autor.

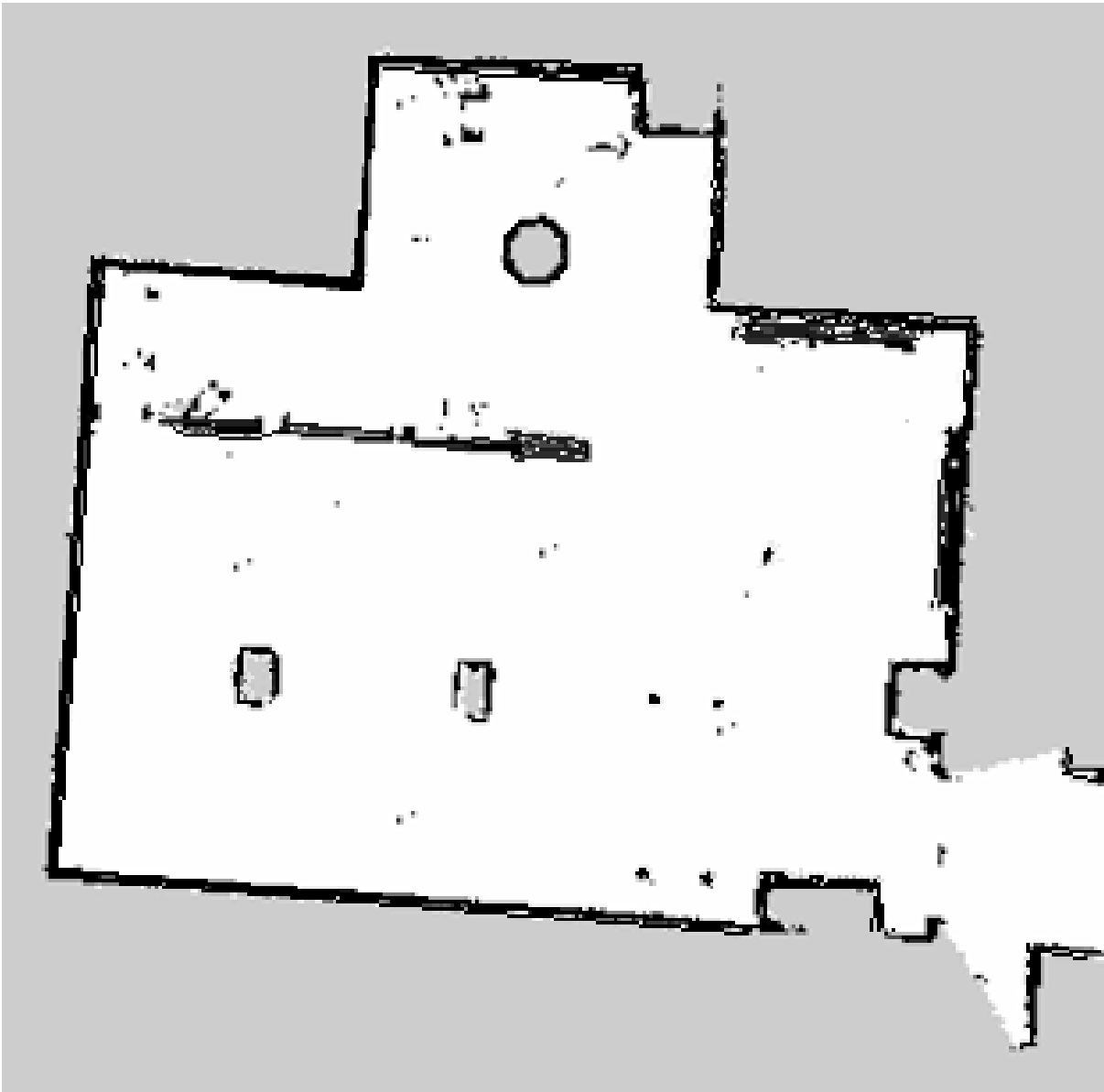


Figura A.24: Mapa da sexta volta em ambiente real usando sensor LiDAR e pacote HECTOR SLAM. Fonte: Próprio autor.

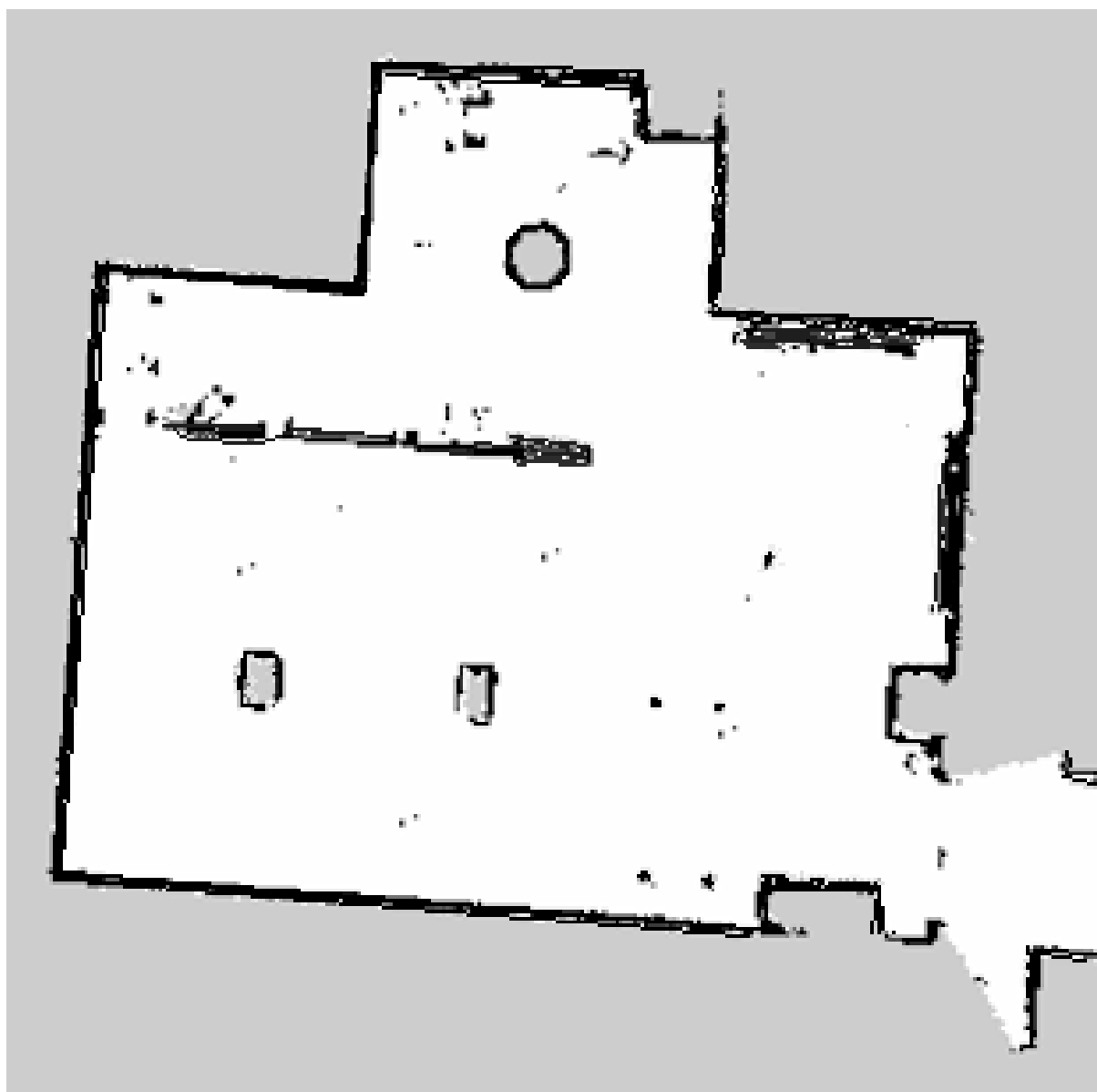


Figura A.25: Mapa da sétima volta em ambiente real usando sensor LiDAR e pacote HECTOR SLAM. Fonte: Próprio autor.

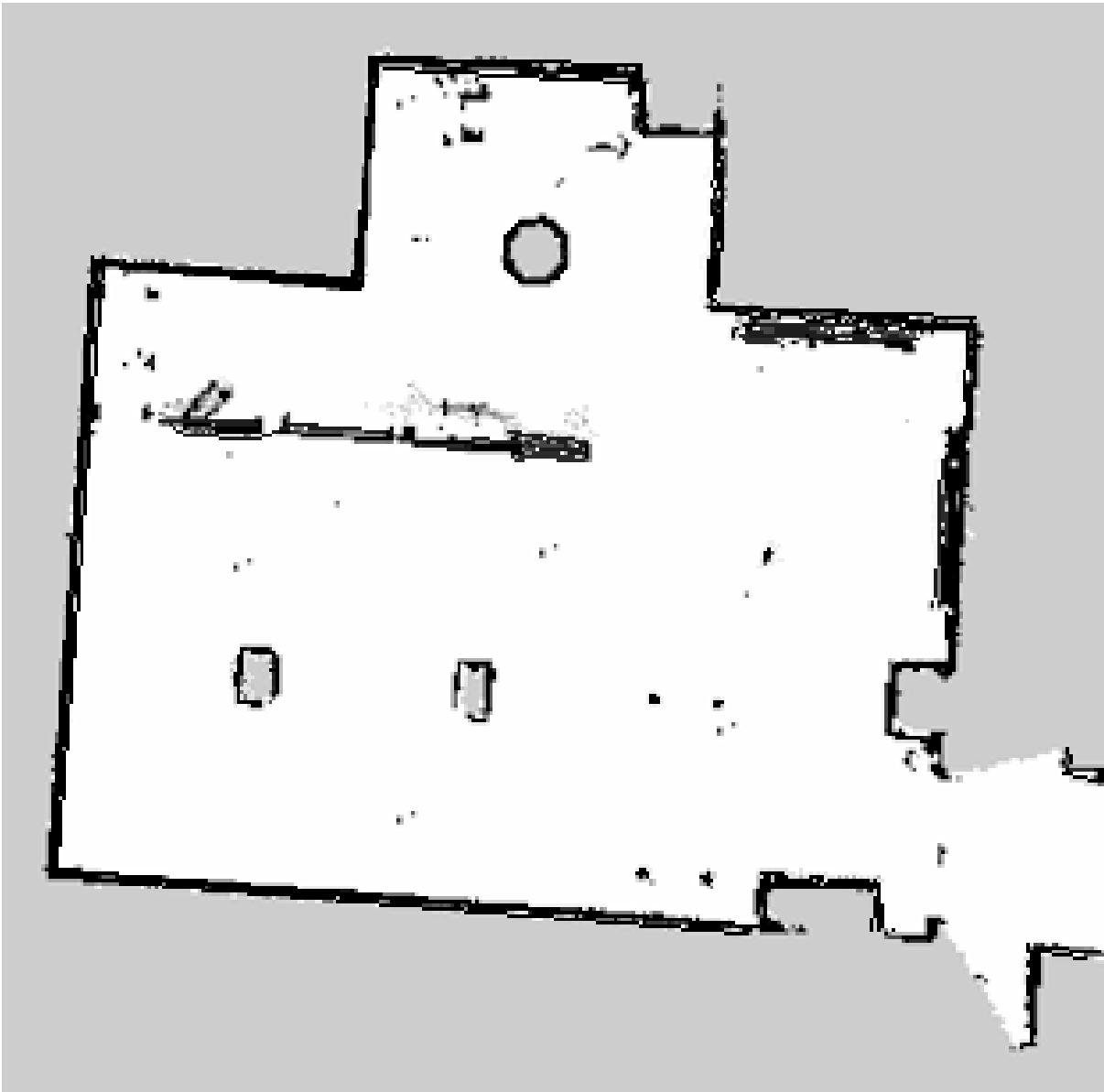


Figura A.26: Mapa da oitava volta em ambiente real usando sensor LiDAR e pacote HECTOR SLAM. Fonte: Próprio autor.

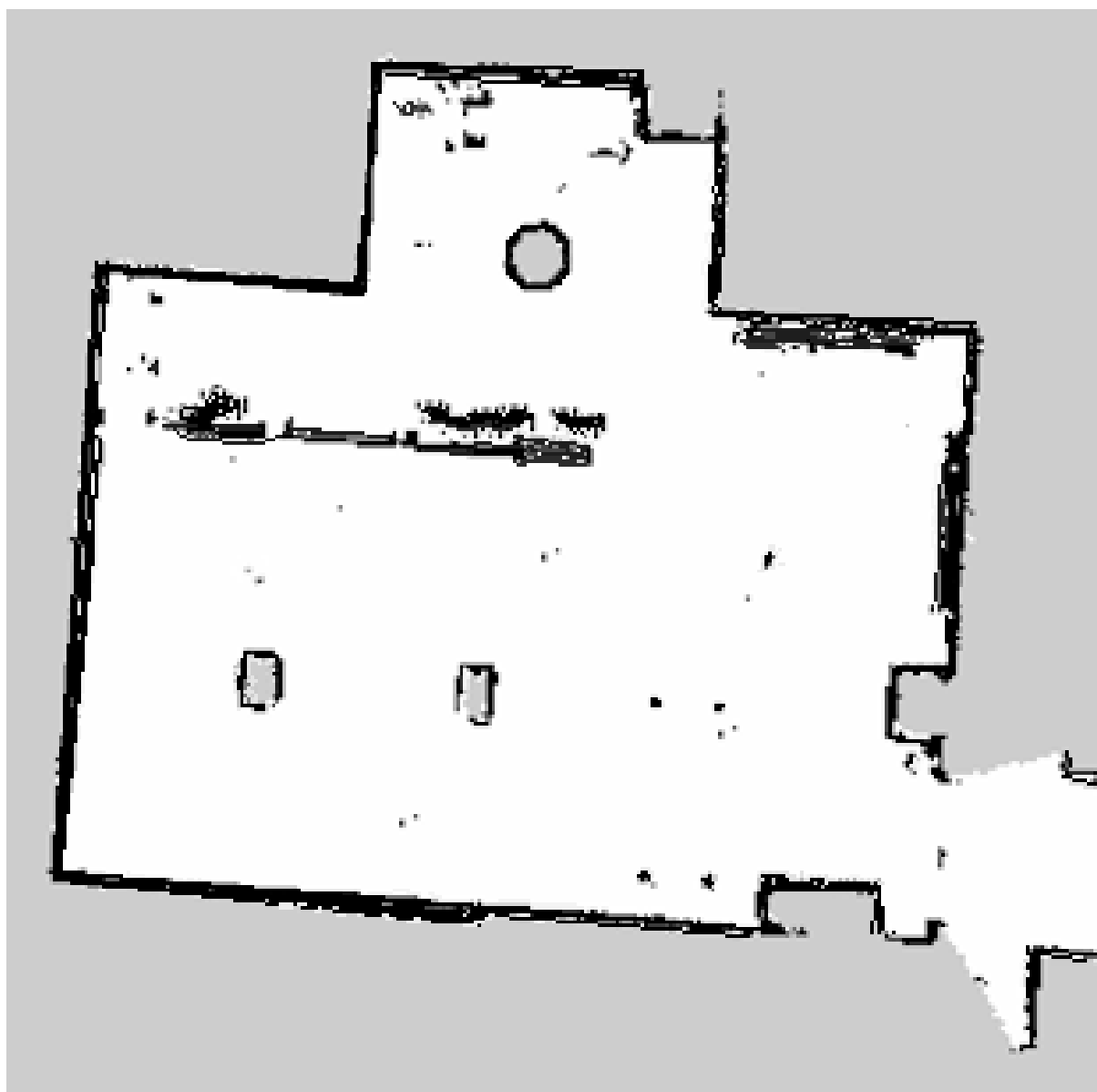


Figura A.27: Mapa da nona volta em ambiente real usando sensor LiDAR e pacote HECTOR SLAM. Fonte: Próprio autor.

A.4 MAPAS EXPERIMENTO REAL 4.2.2

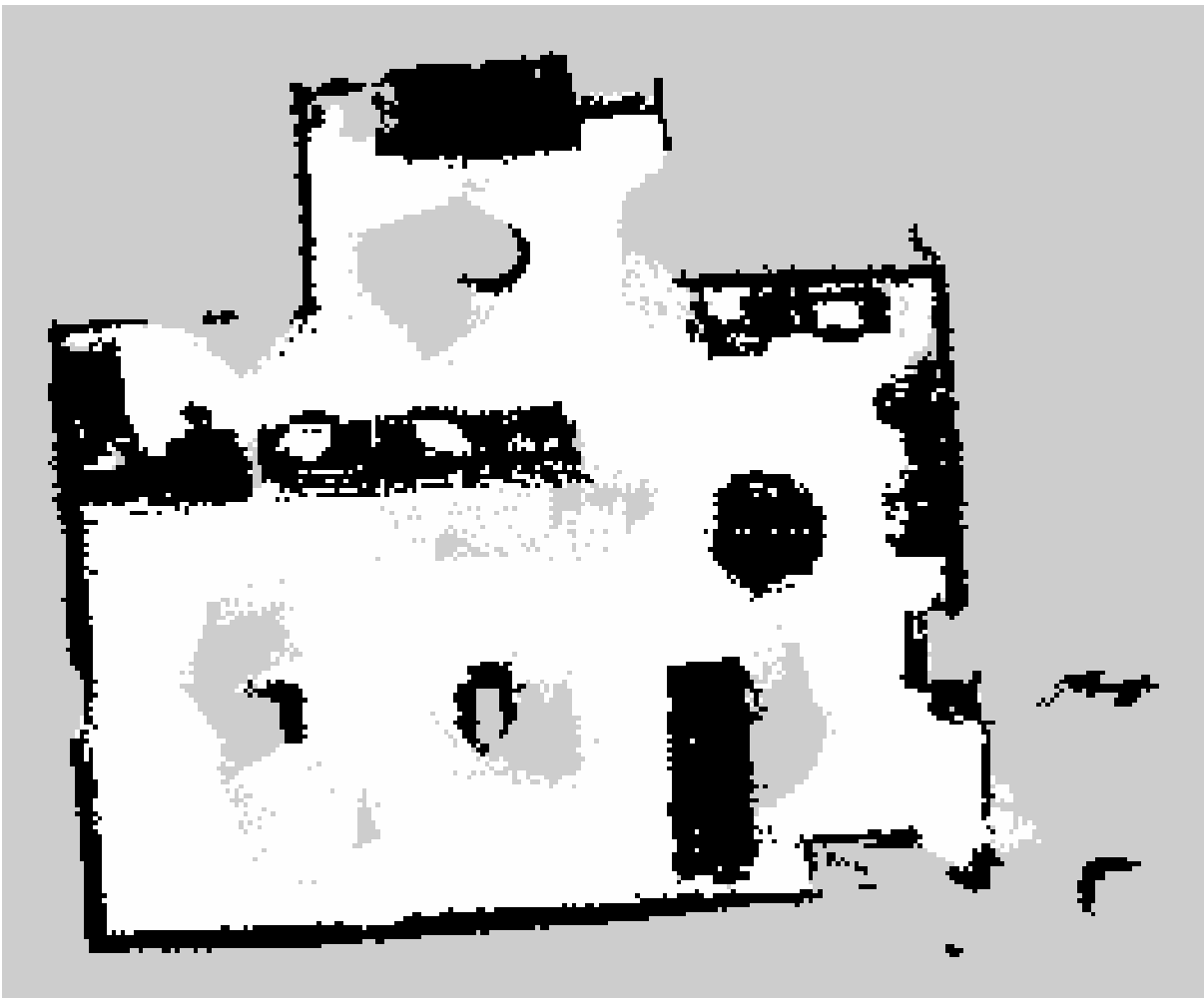


Figura A.28: Mapa da primeira volta em ambiente real usando sensores LiDAR + RGB-D e pacote HECTOR SLAM. Fonte: Próprio autor.

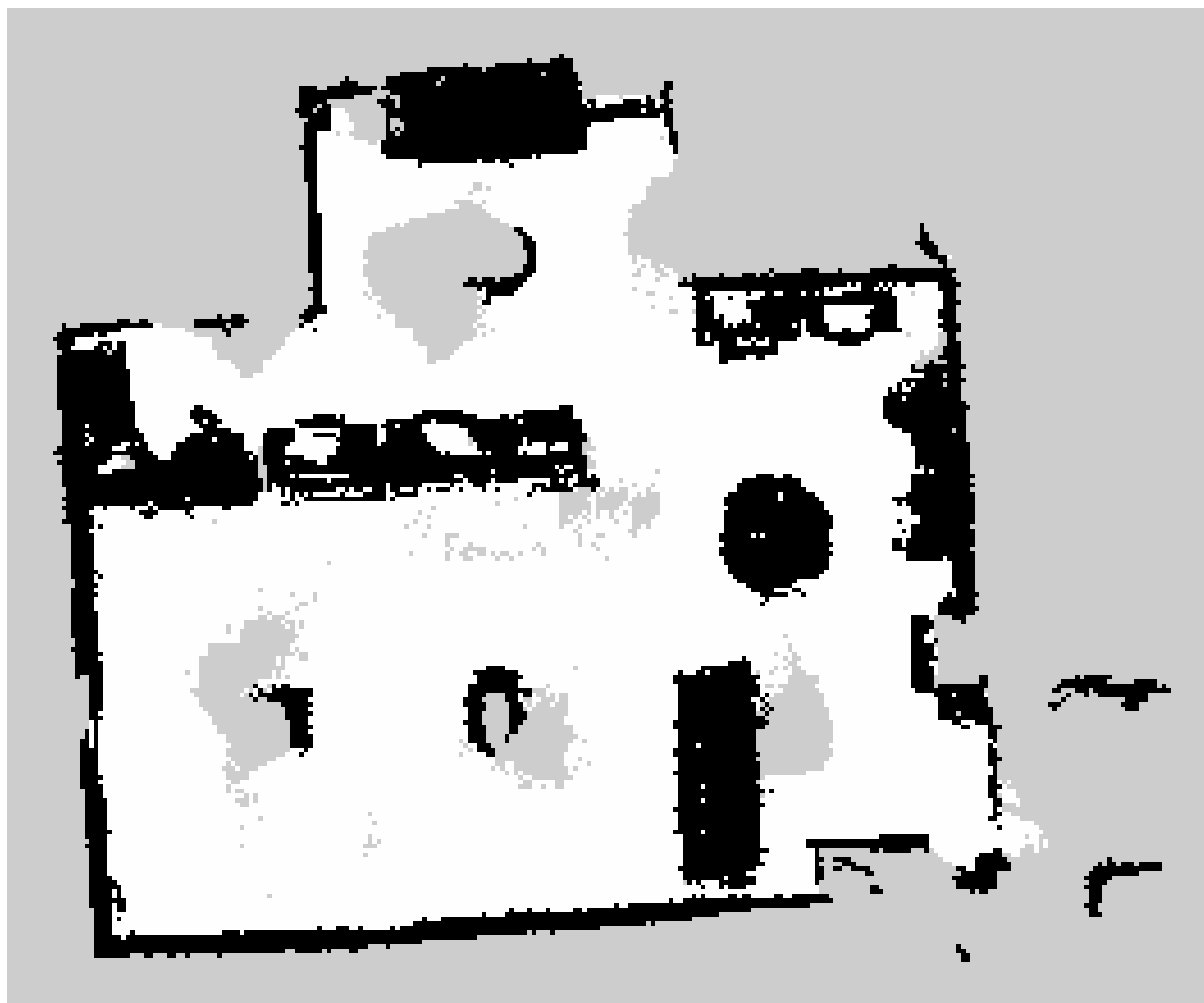


Figura A.29: Mapa da segunda volta em ambiente real usando sensores LiDAR + RGB-D e pacote HECTOR SLAM. Fonte: Próprio autor.

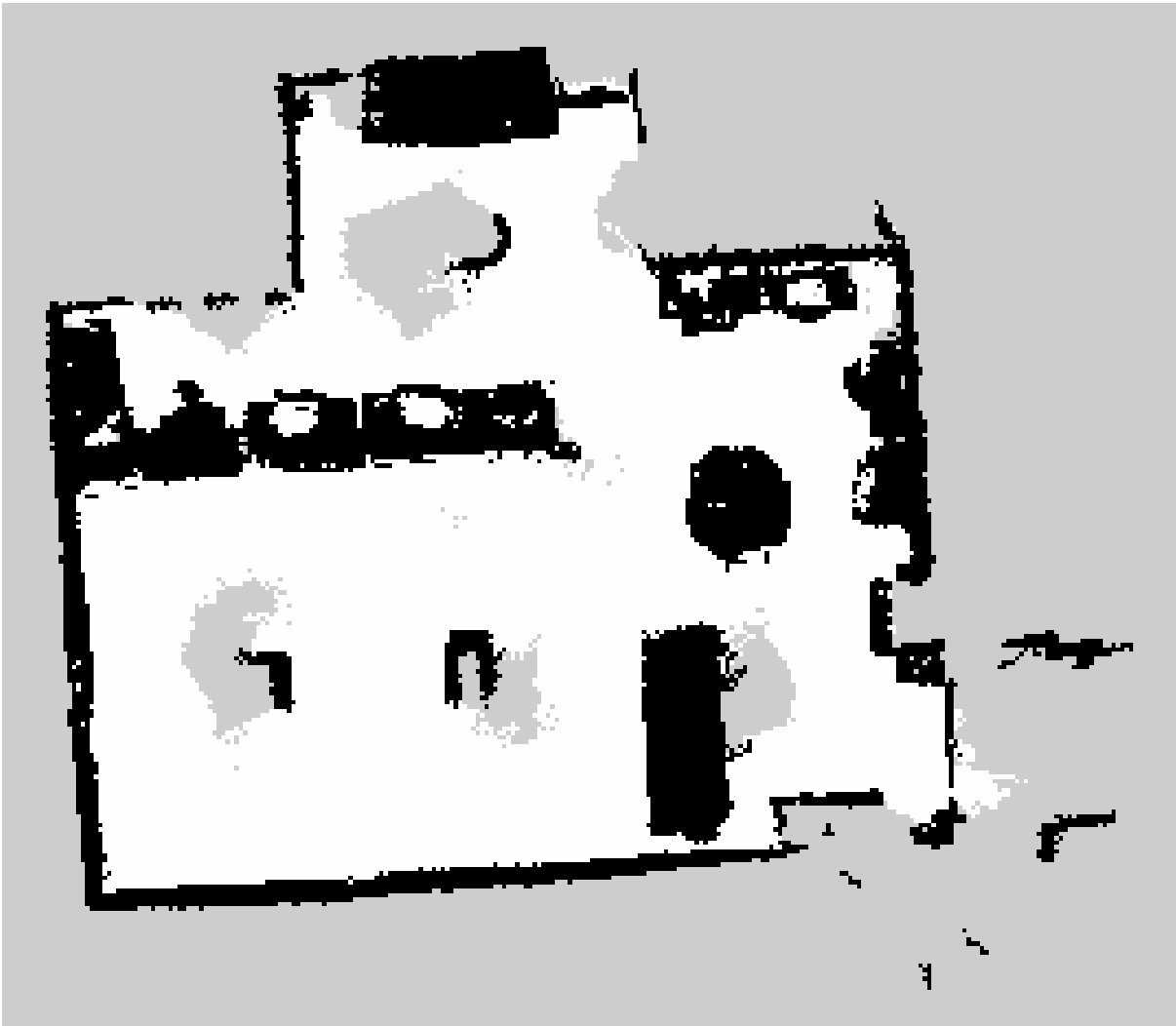


Figura A.30: Mapa da terceira volta em ambiente real usando sensores LiDAR + RGB-D e pacote HECTOR SLAM. Fonte: Próprio autor.

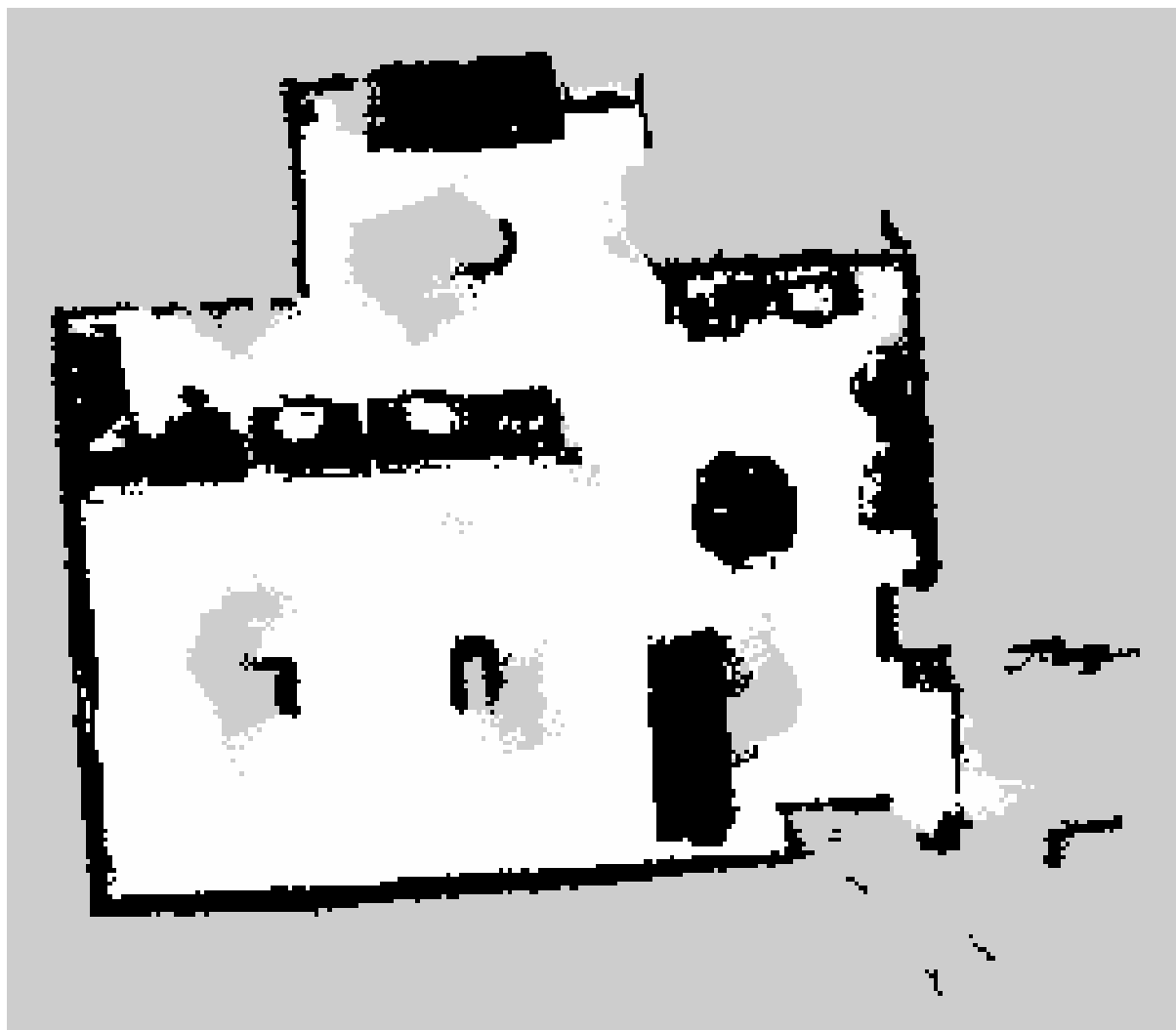


Figura A.31: Mapa da quarta volta em ambiente real usando sensores LiDAR + RGB-D e pacote HECTOR SLAM. Fonte: Próprio autor.

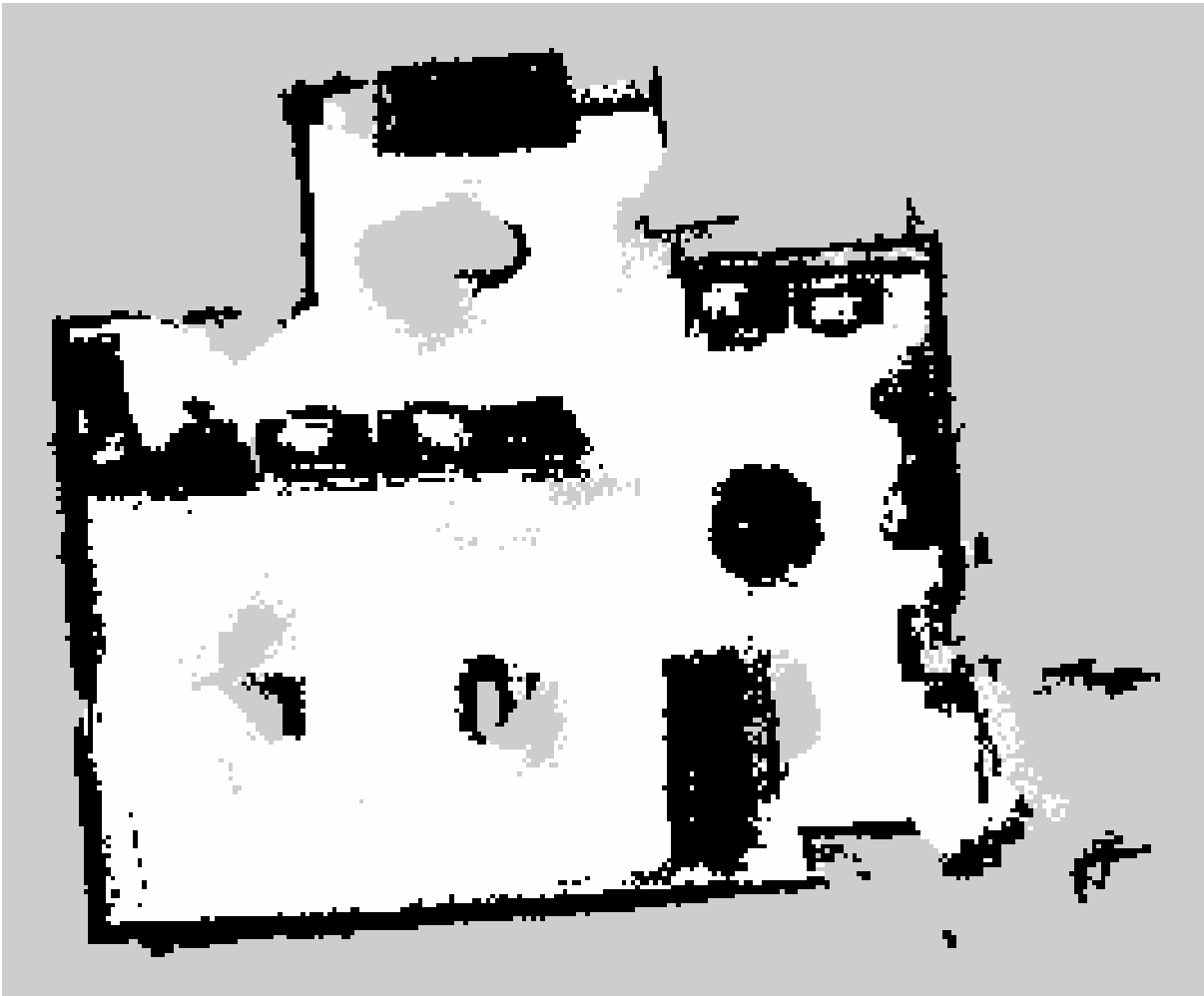


Figura A.32: Mapa da quinta volta em ambiente real usando sensores LiDAR + RGB-D e pacote HECTOR SLAM. Fonte: Próprio autor.

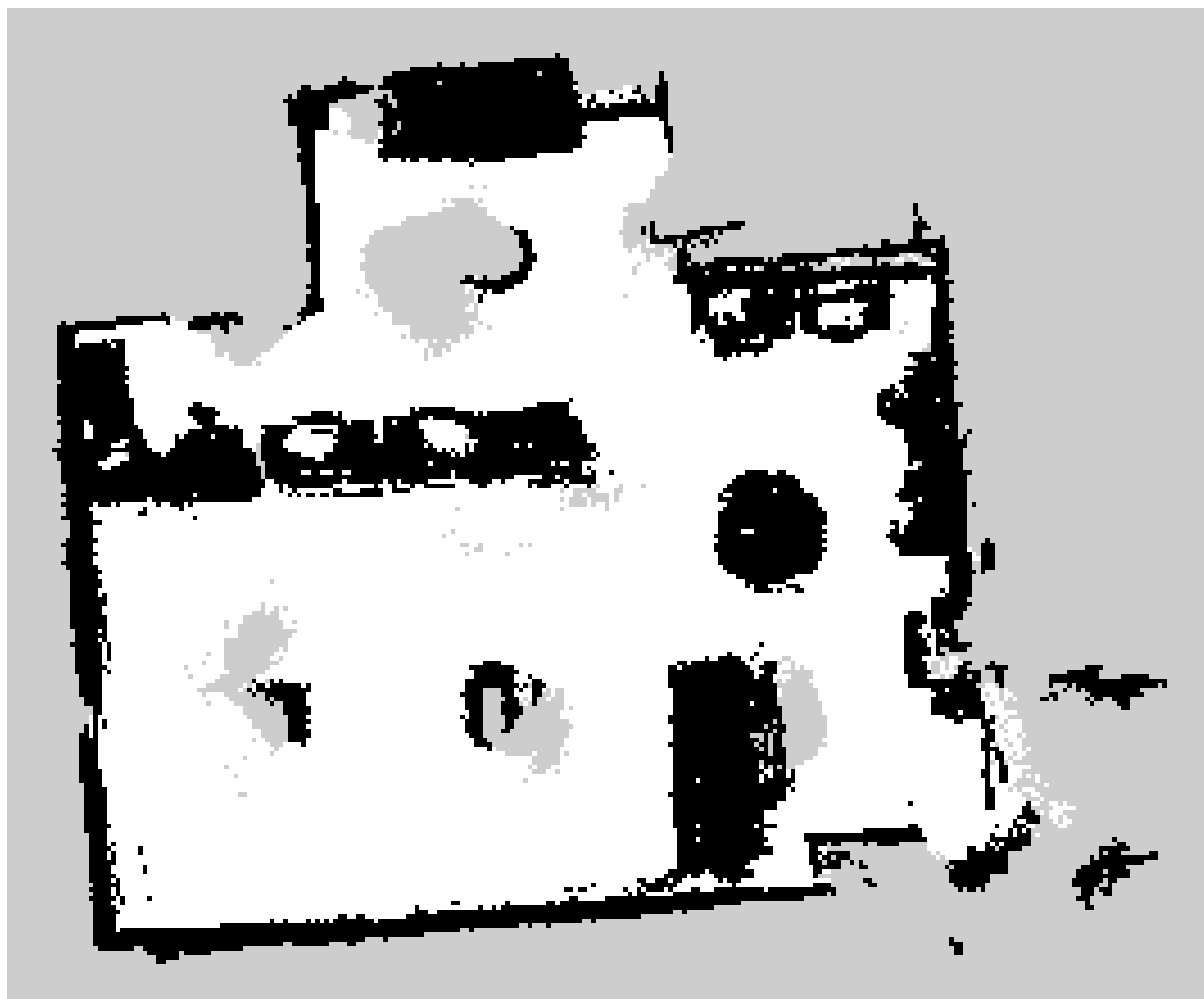


Figura A.33: Mapa da sexta volta em ambiente real usando sensores LiDAR + RGB-D e pacote HECTOR SLAM. Fonte: Próprio autor.

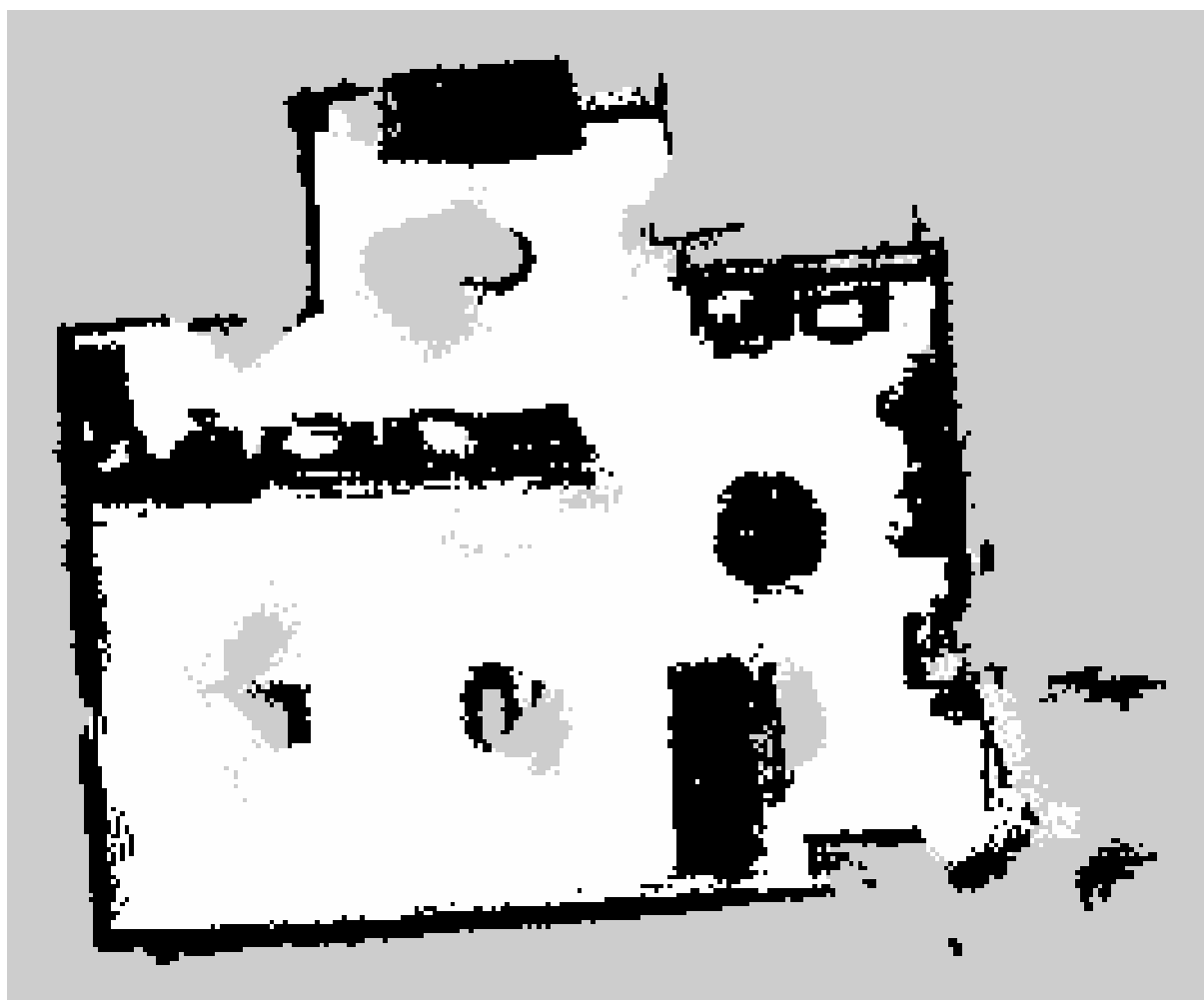


Figura A.34: Mapa da sétima volta em ambiente real usando sensores LiDAR + RGB-D e pacote HECTOR SLAM. Fonte: Próprio autor.

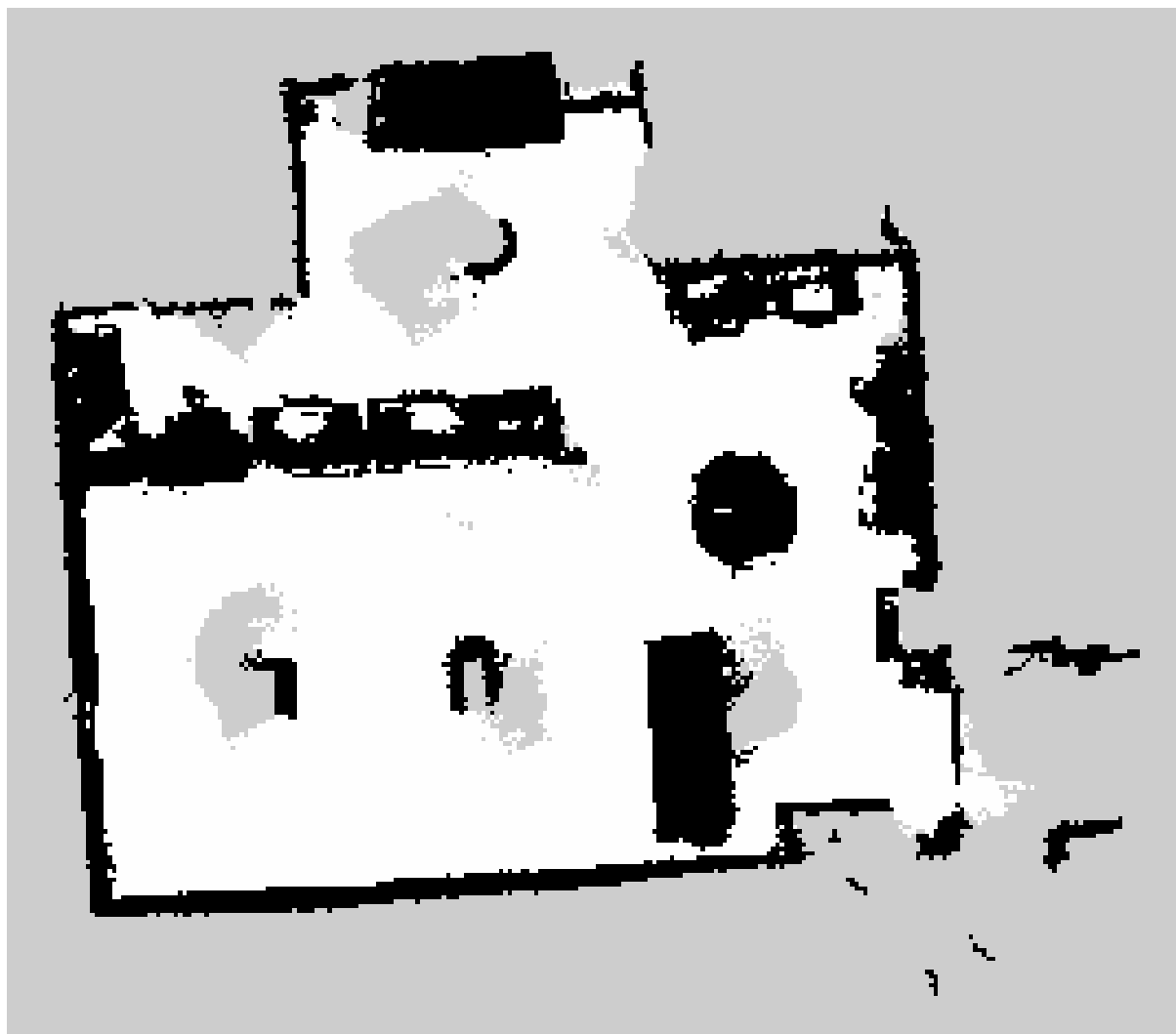


Figura A.35: Mapa da oitava volta em ambiente real usando sensores LiDAR + RGB-D e pacote HECTOR SLAM. Fonte: Próprio autor.

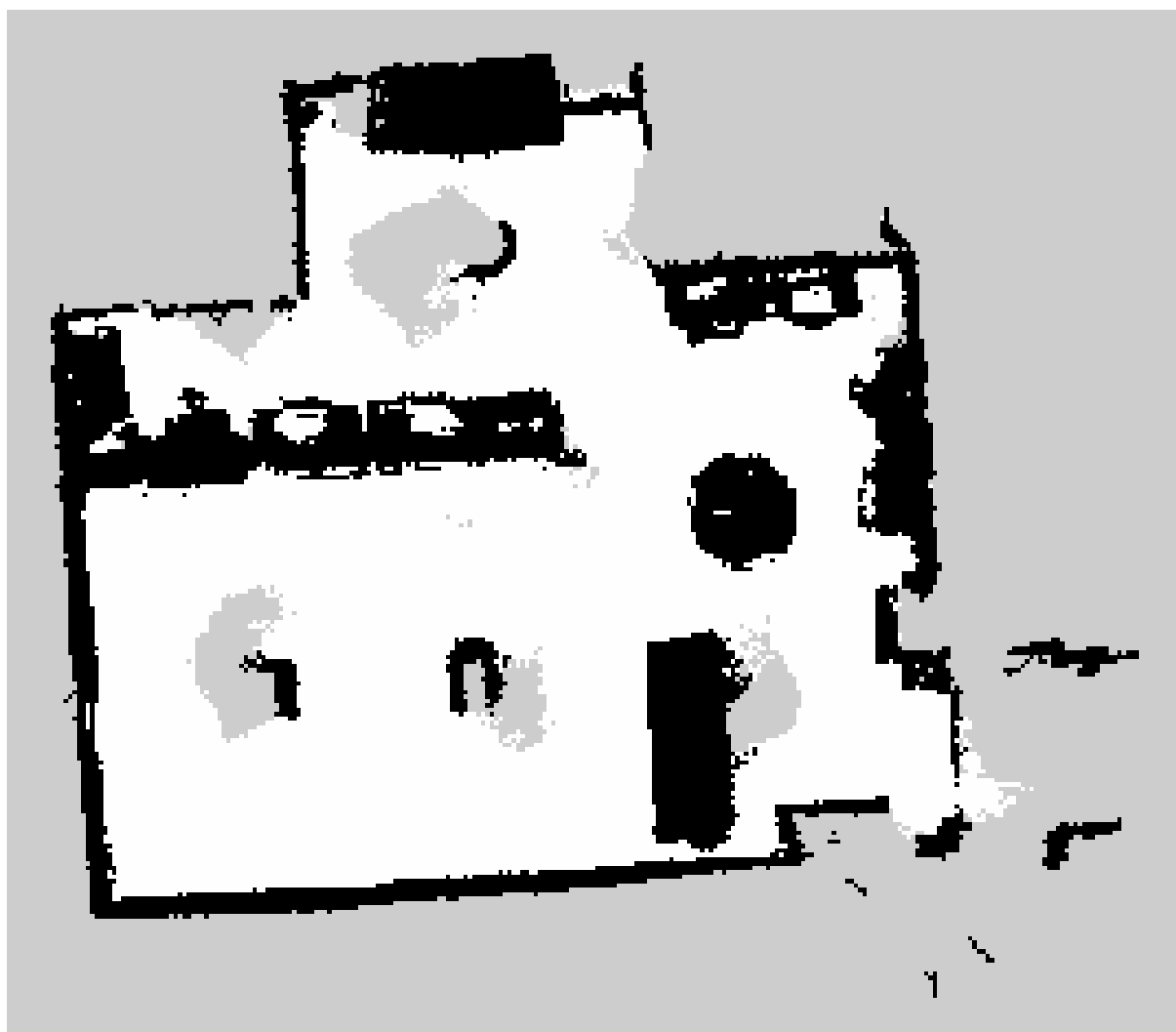


Figura A.36: Mapa da nona volta em ambiente real usando sensores LiDAR + RGB-D e pacote HECTOR SLAM. Fonte: Próprio autor.

A.5 MAPAS DE SIMILARIDADE LOCAL

Figura A.37: Mapa de similaridade local entre mapa 1 e mapa de referência (similaridade global: 0,7369). Fonte: Próprio autor.



Figura A.38: Mapa de similaridade local entre mapa 2 e mapa de referência (similaridade global: 0,8199). Fonte: Próprio autor.



Figura A.39: Mapa de similaridade local entre mapa 3 e mapa de referência (similaridade global: 0,9576). Fonte: Próprio autor.

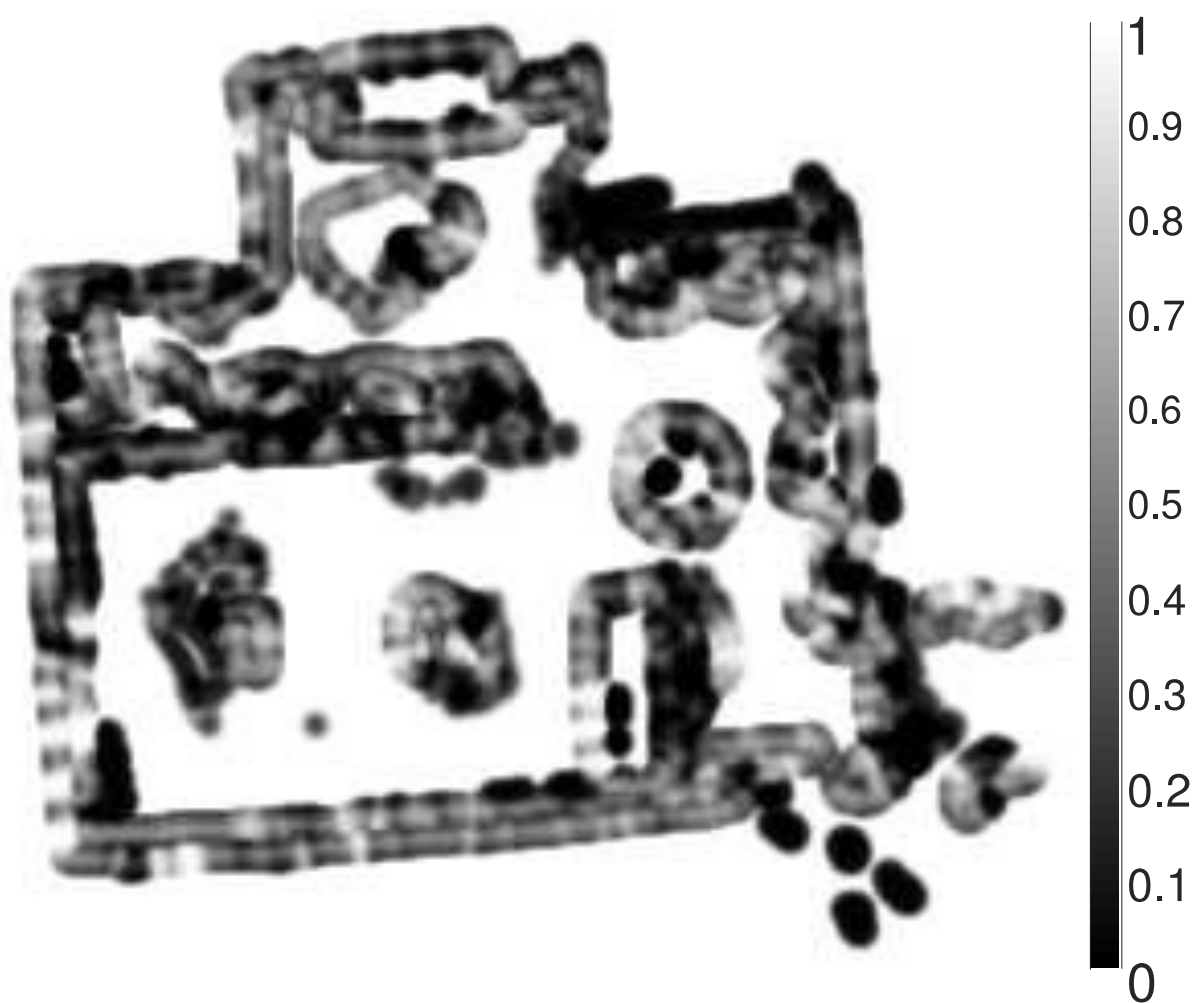


Figura A.40: Mapa de similaridade local entre mapa 5 e mapa de referência (similaridade global: 0,73439). Fonte: Próprio autor.



Figura A.41: Mapa de similaridade local entre mapa 6 e mapa de referência (similaridade global: 0.7888). Fonte: Próprio autor.



Figura A.42: Mapa de similaridade local entre mapa 7 e mapa de referência (similaridade global: 0,7924). Fonte: Próprio autor.



Figura A.43: Mapa de similaridade local entre mapa 8 e mapa de referência (similaridade global: 0,8036). Fonte: Próprio autor.

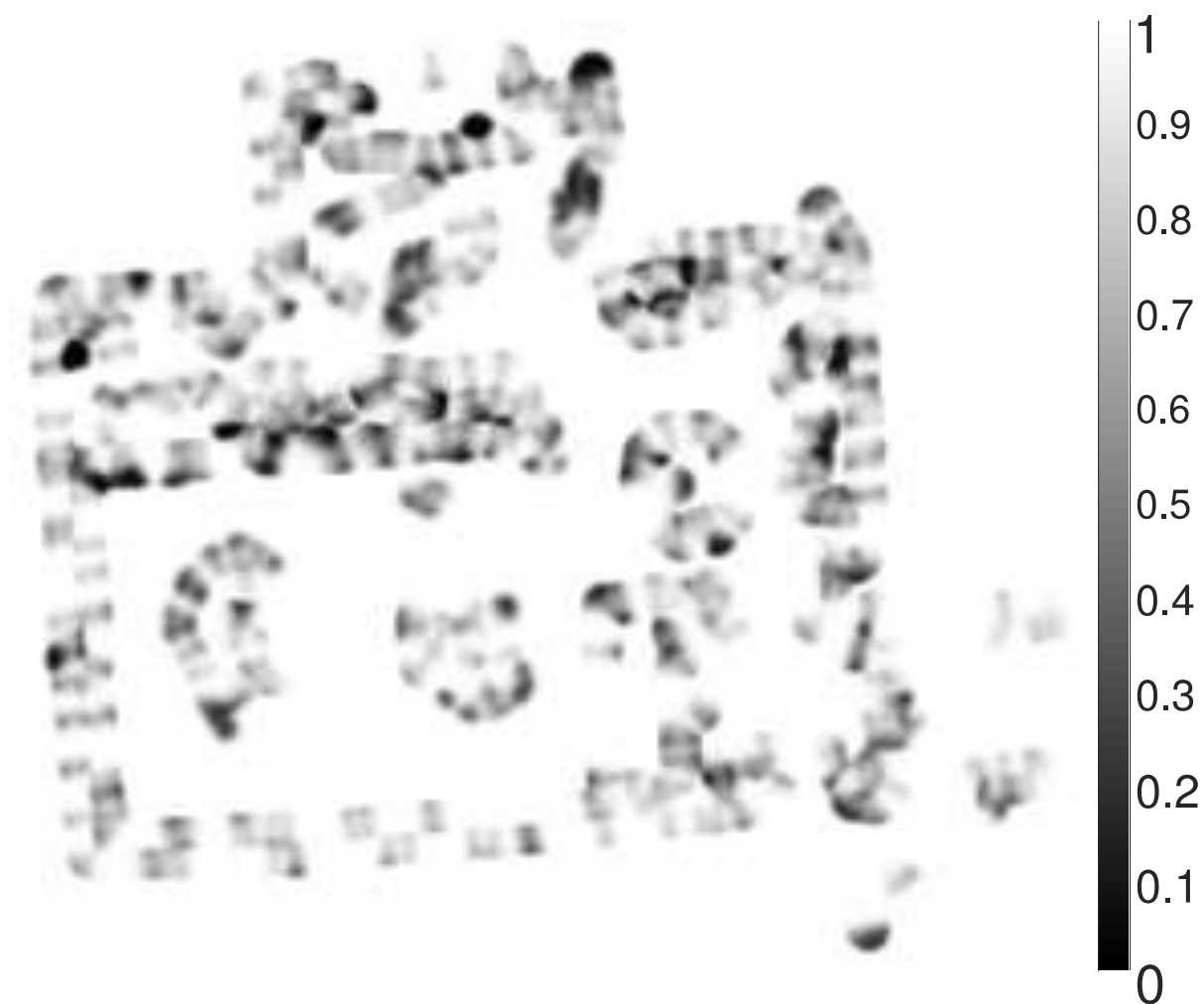


Figura A.44: Mapa de similaridade local entre mapa 9 e mapa de referência (similaridade global: 0.9268). Fonte: Próprio autor.

REFERÊNCIAS BIBLIOGRÁFICAS

- ALNOWAMI, M.; ALNWAIMI, B.; TAHAVORI, F.; COPLAND, M.; WELLS, K. A quantitative assessment of using the kinect for xbox 360 for respiratory surface motion tracking. In: INTERNATIONAL SOCIETY FOR OPTICS AND PHOTONICS. *Medical Imaging 2012: Image-Guided Procedures, Robotic Interventions, and Modeling*. [S.l.], 2012. v. 8316, p. 83161T.
- BABACAN, S. D.; MOLINA, R.; KATSAGGELOS, A. K. Variational bayesian super resolution. *IEEE Transactions on Image Processing*, IEEE, v. 20, n. 4, p. 984–999, 2010.
- BAILEY, T.; NIETO, J.; NEBOT, E. Consistency of the fastslam algorithm. In: IEEE. *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006*. [S.l.], 2006. p. 424–429.
- CARLONE, L.; ARAGUES, R.; CASTELLANOS, J. A.; BONA, B. A linear approximation for graph-based simultaneous localization and mapping. *Robotics: Science and Systems VII*, MIT Press, p. 41–48, 2012.
- CSORBA, M. *Simultaneous localisation and map building*. Tese (Doutorado) — University of Oxford, 1997.
- DAVISON, A. J.; CID, Y. G.; KITA, N. Real-time 3d slam with wide-angle vision. *IFAC Proceedings Volumes*, Elsevier, v. 37, n. 8, p. 868–873, 2004.
- DISSANAYAKE, M. G.; NEWMAN, P.; CLARK, S.; DURRANT-WHYTE, H. F.; CSORBA, M. A solution to the simultaneous localization and map building (slam) problem. *IEEE Transactions on robotics and automation*, IEEE, v. 17, n. 3, p. 229–241, 2001.
- DUBÉ, R.; DUGAS, D.; STUMM, E.; NIETO, J.; SIEGWART, R.; CADENA, C. Seg-match: Segment based place recognition in 3d point clouds. In: IEEE. *2017 IEEE International Conference on Robotics and Automation (ICRA)*. [S.l.], 2017. p. 5266–5272.
- DURRANT-WHYTE, H.; BAILEY, T. Simultaneous localization and mapping: part i. *IEEE robotics & automation magazine*, IEEE, v. 13, n. 2, p. 99–110, 2006.
- DURRANT-WHYTE, H.; RYE, D.; NEBOT, E. Localization of autonomous guided vehicles. In: *Robotics Research: The 7th International Symposium (ISRR'95)*. [S.l.]: Springer, 1996. p. 613–625.

EL-LAITHY, R. A.; HUANG, J.; YEH, M. Study on the use of microsoft kinect for robotics applications. In: IEEE. *Proceedings of the 2012 IEEE/ION Position, Location and Navigation Symposium*. [S.l.], 2012. p. 1280–1288.

ELFES, A. A probabilistic framework for robot perception and navigation. *PhD thesis, Carnegie-Mellon University*, 1989.

ENDRES, F.; HESS, J.; STURM, J.; CREMERS, D.; BURGARD, W. 3-d mapping with an rgb-d camera. *IEEE transactions on robotics*, IEEE, v. 30, n. 1, p. 177–187, 2013.

EUSTICE, R.; SINGH, H.; LEONARD, J. J.; WALTER, M. R.; BALLARD, R. Visually navigating the rms titanic with slam information filters. In: *Robotics: Science and Systems*. [S.l.: s.n.], 2005. v. 2005, p. 57–64.

FILIPENKO, M.; AFANASYEV, I. Comparison of various slam systems for mobile robot in an indoor environment. In: IEEE. *2018 International Conference on Intelligent Systems (IS)*. [S.l.], 2018. p. 400–407.

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

FRAUNDORFER, F.; SCARAMUZZA, D. Visual odometry: Part ii: Matching, robustness, optimization, and applications. *IEEE Robotics & Automation Magazine*, IEEE, v. 19, n. 2, p. 78–90, 2012.

FROST, J. *Robust and Scalable Visual Simultaneous Localization and Mapping in Indoor Environments using RGBD Cameras*. Tese (Doutorado) — Universität zu Lübeck, 2016.

FRUH, C.; ZAKHOR, A. 3d model generation for cities using aerial photographs and ground level laser scans. In: IEEE. *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*. [S.l.], 2001. v. 2, p. II–II.

GÁLVEZ-LÓPEZ, D.; TARDÓS, J. D. Bags of binary words for fast place recognition in image sequences. *IEEE Transactions on Robotics*, v. 28, n. 5, p. 1188–1197, October 2012. ISSN 1552-3098.

GRISSETTI, G.; STACHNISS, C.; BURGARD, W. Improved techniques for grid mapping with rao-blackwellized particle filters. *IEEE transactions on Robotics*, IEEE, v. 23, n. 1, p. 34–46, 2007.

GRISSETTI, G.; STACHNISS, C.; BURGARD, W. Improved techniques for grid mapping with rao-blackwellized particle filters. *IEEE transactions on Robotics*, IEEE, v. 23, n. 1, p. 34–46, 2007.

- GUTMANN, J.-S.; KONOLIGE, K. Incremental mapping of large cyclic environments. In: IEEE. *Proceedings 1999 IEEE International Symposium on Computational Intelligence in Robotics and Automation. CIRA'99 (Cat. No. 99EX375)*. [S.l.], 1999. p. 318–325.
- HAN, J.; SHAO, L.; XU, D.; SHOTTON, J. Enhanced computer vision with microsoft kinect sensor: A review. *IEEE transactions on cybernetics*, IEEE, v. 43, n. 5, p. 1318–1334, 2013.
- HENRY, P.; KRAININ, M.; HERBST, E.; REN, X.; FOX, D. Rgb-d mapping: Using depth cameras for dense 3d modeling of indoor environments. In: SPRINGER. *Experimental robotics*. [S.l.], 2014. p. 477–491.
- HESS, W.; KOHLER, D.; RAPP, H.; ANDOR, D. Real-time loop closure in 2d lidar slam. In: IEEE. *2016 IEEE International Conference on Robotics and Automation (ICRA)*. [S.l.], 2016. p. 1271–1278.
- HOWARD, A.; MATARIC, M. J.; SUKHATME, G. Relaxation on a mesh: a formalism for generalized localization. In: IEEE. *Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems. Expanding the Societal Role of Robotics in the the Next Millennium (Cat. No. 01CH37180)*. [S.l.], 2001. v. 2, p. 1055–1060.
- IZADI, S.; KIM, D.; HILLIGES, O.; MOLYNEAUX, D.; NEWCOMBE, R.; KOHLI, P.; SHOTTON, J.; HODGES, S.; FREEMAN, D.; DAVISON, A. et al. Kinectfusion: real-time 3d reconstruction and interaction using a moving depth camera. In: *Proceedings of the 24th annual ACM symposium on User interface software and technology*. [S.l.: s.n.], 2011. p. 559–568.
- JAFARI, O. H.; MITZEL, D.; LEIBE, B. Real-time rgb-d based people detection and tracking for mobile robots and head-worn cameras. In: IEEE. *2014 IEEE international conference on robotics and automation (ICRA)*. [S.l.], 2014. p. 5636–5643.
- KOCH, R. *Sensor Fusion for Precise Mapping of Transparent and Specular Reflective Objects*. Tese (doctoralthesis) — Universität Würzburg, 2018.
- KOENIG, N.; HOWARD, A. Design and use paradigms for gazebo, an open-source multi-robot simulator. In: IEEE. *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)(IEEE Cat. No. 04CH37566)*. [S.l.], 2004. v. 3, p. 2149–2154.
- KOHLBRECHER, S.; MEYER, J.; GRABER, T.; PETERSEN, K.; KLINGAUF, U.; STRYK, O. von. Hector open source modules for autonomous mapping and navigation with rescue robots. In: SPRINGER. *Robot Soccer World Cup*. [S.l.], 2013. p. 624–631.
- KOHLBRECHER, S.; STRYK, O. V.; MEYER, J.; KLINGAUF, U. A flexible and scalable slam system with full 3d motion estimation. In: IEEE. *2011 IEEE International Symposium on Safety, Security, and Rescue Robotics*. [S.l.], 2011. p. 155–160.

LABBE, M.; MICHAUD, F. Appearance-based loop closure detection for online large-scale and long-term operation. *IEEE Transactions on Robotics*, IEEE, v. 29, n. 3, p. 734–745, 2013.

LABBÉ, M.; MICHAUD, F. Rtab-map as an open-source lidar and visual simultaneous localization and mapping library for large-scale and long-term online operation. *Journal of Field Robotics*, Wiley Online Library, v. 36, n. 2, p. 416–446, 2019.

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

MONTEMERLO, M.; THRUN, S.; KOLLER, D.; WEGBREIT, B. et al. Fastslam: A factored solution to the simultaneous localization and mapping problem. *Aaai/iaai*, v. 593598, 2002.

MONTEMERLO, M.; THRUN, S.; KOLLER, D.; WEGBREIT, B. et al. Fastslam 2.0: An improved particle filtering algorithm for simultaneous localization and mapping that provably converges. In: *IJCAI*. [S.l.: s.n.], 2003. p. 1151–1156.

MOORE, T.; STOUCH, D. A generalized extended kalman filter implementation for the robot operating system. In: *Proceedings of the 13th International Conference on Intelligent Autonomous Systems (IAS-13)*. [S.l.]: Springer, 2014.

MUNARO, M.; BASSO, F.; MENEGATTI, E. Tracking people within groups with rgb-d data. In: IEEE. *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. [S.l.], 2012. p. 2101–2107.

MUR-ARTAL, R.; TARDÓS, J. D. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE Transactions on Robotics*, IEEE, v. 33, n. 5, p. 1255–1262, 2017.

NEWMAN, P.; COLE, D.; HO, K. Outdoor slam using visual appearance and laser ranging. In: IEEE. *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006*. [S.l.], 2006. p. 1180–1187.

OLIVER, A.; KANG, S.; WÜNSCHE, B. C.; MACDONALD, B. Using the kinect as a navigation sensor for mobile robotics. In: *Proceedings of the 27th conference on image and vision computing New Zealand*. [S.l.: s.n.], 2012. p. 509–514.

PIRE, T.; FISCHER, T.; CASTRO, G.; CRISTÓFORIS, P. D.; CIVERA, J.; BERLLES, J. J. S-PTAM: Stereo Parallel Tracking and Mapping. *Robotics and Autonomous Systems (RAS)*, v. 93, p. 27 – 42, 2017. ISSN 0921-8890.

QUIGLEY, M.; CONLEY, K.; GERKEY, B.; FAUST, J.; FOOTE, T.; LEIBS, J.; WHEELER, R.; NG, A. Y. Ros: an open-source robot operating system. In: KOBE, JAPAN. *ICRA workshop on open source software*. [S.l.], 2009. v. 3, n. 3.2, p. 5.

- REIS, D. H. D.; WELFER, D.; CUADROS, M. A. D. S. L.; GAMARRA, D. F. T. Mobile robot navigation using an object recognition software with rgb-d images and the yolo algorithm. *Applied Artificial Intelligence*, Taylor & Francis, v. 33, n. 14, p. 1290–1305, 2019.
- SALOMON, D. *Data compression: the complete reference*. [S.l.]: Springer Science & Business Media, 2004.
- SANTANA, A. M. Localização e mapeamento simultâneos de ambientes planos usando visão monocular e representação híbrida do ambiente. Universidade Federal do Rio Grande do Norte, 2011.
- SCARAMUZZA, D.; FRAUNDORFER, F. Visual odometry [tutorial]. *IEEE robotics & automation magazine*, IEEE, v. 18, n. 4, p. 80–92, 2011.
- SCHNEIDER, T.; DYMZYK, M. T.; FEHR, M.; EGGER, K.; LYNEN, S.; GILITS-CHENSKI, I.; SIEGWART, R. maplab: An open framework for research in visual-inertial mapping and localization. *IEEE Robotics and Automation Letters*, 2018.
- SRI, A. I. C. *Shakey*. 2020. Último acesso em 06 de Agosto de 2020. Disponível em: <http://www.ai.sri.com/shakey/>.
- STATISTICS, U. S. B. of L. *Fatal occupational injuries by industry and event or exposure, all United States*. 2018. Último acesso em 06 de Agosto de 2020. Disponível em: <https://www.bls.gov/iif/oshwc/cfoi/cftb0322.htm>.
- STEUX, B.; HAMZAoui, O. E. tinyslam: A slam algorithm in less than 200 lines c-language program. In: IEEE. *2010 11th International Conference on Control Automation Robotics & Vision*. [S.l.], 2010. p. 1975–1979.
- TECHCRUNCH. *Videos: The Best Kinect Hacks And Mods One Month In*. 2010. Último acesso em 06 de Agosto de 2020. Disponível em: <https://techcrunch.com/2010/12/07/videos-the-best-kinect-hacks-and-mods-one-month-in/>.
- THRUN, S. Learning occupancy grids with forward models. In: IEEE. *Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems. Expanding the Societal Role of Robotics in the the Next Millennium (Cat. No. 01CH37180)*. [S.l.], 2001. v. 3, p. 1676–1681.
- THRUN, S.; BURGARD, W.; FOX, D. A probabilistic approach to concurrent mapping and localization for mobile robots. *Autonomous Robots*, Springer, v. 5, n. 3-4, p. 253–271, 1998.
- THRUN, S.; BURGARD, W.; FOX, D. A real-time algorithm for mobile robot mapping with applications to multi-robot and 3d mapping. In: IEEE. *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065)*. [S.l.], 2000. v. 1, p. 321–328.

VILLENA-MARTÍNEZ, V.; FUSTER-GUILLÓ, A.; AZORÍN-LÓPEZ, J.; SAVAL-CALVO, M.; MORA-PASCUAL, J.; GARCIA-RODRIGUEZ, J.; GARCIA-GARCIA, A. A quantitative comparison of calibration methods for rgb-d sensors using different technologies. *Sensors*, Multidisciplinary Digital Publishing Institute, v. 17, n. 2, p. 243, 2017.

VILLENA, S.; VEGA, M.; BABACAN, S. D.; MOLINA, R.; KATSAGGELOS, A. K. Bayesian combination of sparse and non-sparse priors in image super resolution. *Digital Signal Processing*, Elsevier, v. 23, n. 2, p. 530–541, 2013.

VINCENT, R.; LIMKETKAI, B.; ERIKSEN, M. Comparison of indoor robot localization techniques in the absence of gps. In: INTERNATIONAL SOCIETY FOR OPTICS AND PHOTONICS. *Detection and Sensing of Mines, Explosive Objects, and Obscured Targets XV*. [S.l.], 2010. v. 7664, p. 76641Z.

WANG, Z.; BOVIK, A. C.; SHEIKH, H. R.; SIMONCELLI, E. P. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, IEEE, v. 13, n. 4, p. 600–612, 2004.

ZHANG, L.; MERRIFIELD, R.; DEGUET, A.; YANG, G.-Z. Powering the world's robots-10 years of ros. In: AMERICAN ASSOCIATION FOR THE ADVANCEMENT OF SCIENCE. [S.l.], 2017.