

Robô móvel teleguiado de mapeamento tridimensional

VINÍCIUS JOÃO DE BARROS VANZIN⁵
JOÃO LUÍS TAVARES DA SILVA⁶

Data de submissão: 20/10/2021. Data de publicação: 30/10/2021.

RESUMO

Considerado uma das questões fundamentais na concessão de autonomia a agentes robóticos, o problema de localização e mapeamento simultâneos (SLAM) envolve a determinação da postura do robô concorrentemente à elaboração de um mapa do ambiente no qual ele se encontra. O problema de SLAM foi solucionado com a aplicação de técnicas baseadas na teoria de probabilidades na abordagem denominada robótica probabilística e, recentemente, tem incorporado técnicas de visão computacional, formando a área de SLAM visual. O objetivo deste projeto é apresentar a especificação de *hardware* e de *software* e a construção de um protótipo de robô móvel teleguiado que realiza o processo de mapeamento robótico tridimensional. Em adição, o robô é avaliado em experimentos em diferentes cenários a fim de determinar o desempenho da aplicação.

Palavras-Chave: SLAM visual. Robótica probabilística. Visão computacional.

ABSTRACT

Considered one of the fundamental issues regarding the granting of autonomy to robotic agents, the problem of simultaneous localization and mapping (SLAM) involves the determination of the robot's pose concurrently with the construction of a map of its environment. The SLAM problem was solved through the application of probability theory-based techniques in an approach known as probabilistic robotics and, recently, has incorporated computer vision techniques, thus originating the visual SLAM field. This project's goal is to present the hardware and software specifications and the construction of a teleoperated mobile robot prototype that performs the tridimensional robotic mapping process. In addition, the robot is evaluated in experiments under different scenarios to determine the application's performance.

Keywords: Visual SLAM. Probabilistic robotics. Computer vision.

1 INTRODUÇÃO

Ao contrário dos animais, nos quais os processos de extração de informações do ambiente e de navegação são naturais, nos robôs a tarefa de determinação de sua própria localização e do mapeamento de seus arredores é complexa. Em muitos casos, o robô não possuirá dados sobre sua postura (posição e orientação) e não contará com um modelo do

⁵ Pós-graduado em Especialização em Docência pelo Centro Universitário Uniftec de Caxias do Sul, graduado em Engenharia de Computação pelo Centro Universitário Uniftec de Caxias do Sul.

⁶ Doutor em Ciência da Computação; Professor no Centro Universitário Uniftec.

ambiente, devendo, assim, obter ambos simultaneamente. Além disso, a dificuldade desse problema é exacerbada devido à dinamicidade e imprevisibilidade dos ambientes humanos.

Denominado de localização e mapeamento simultâneos (*simultaneous localization and mapping – SLAM*), este problema é considerado uma das questões fundamentais da robótica, porque está intimamente conectado à concessão de autonomia plena aos agentes robóticos. Dentre os paradigmas da robótica, aquele que mais obteve sucesso na produção de soluções para o SLAM foi a abordagem probabilística, já que é baseada nos ramos da matemática especializados em lidar com a incerteza: a teoria de probabilidades e a inferência bayesiana.

O presente trabalho foi realizado em três etapas: na primeira etapa, discutiu-se acerca da fundamentação matemática do problema de SLAM e analisou-se as primeiras soluções em cenário bidimensional; na segunda, avaliou-se uma técnica recente, o RTAB-Map (LABBÉ; MICHAUD, 2014), que incorpora métodos de visão computacional para elaboração de mapas tridimensionais densos dos ambientes do robô, com objetivo de obter parâmetros para execução em plataformas com especificações limitadas. Este artigo apresenta a terceira etapa, a especificação, construção e avaliação de um protótipo de robô móvel teleguiado que realiza o processo de SLAM e, portanto, é capaz de manter sua localização relativa ao ambiente e de gerar modelos tridimensionais de suas imediações.

A escolha do tema deste projeto (SLAM) justifica-se pela relevância da área de robótica probabilística na conferência de real autonomia a sistemas robóticos, em particular os de uso não-industrial. Thrun, Burgard e Fox (2005) argumentam que, para alguns problemas de estimativa, as únicas soluções conhecidas são os algoritmos probabilísticos.

Observa-se recentemente, a introdução de algoritmos de mapeamento em produtos destinados ao mercado consumidor. Como exemplo, citam-se as últimas versões do robô de aspiração de pó Roomba (IROBOT, 2017) e a família LD de veículos inteligentes para o setor industrial da Omron (OMRON, 2017). Destaca-se também a ascensão da tecnologia de carros autônomos, impulsionada principalmente pelas indústrias automobilística e de tecnologia da informação. Apesar das barreiras legais existentes, os analistas estimam que, até 2035, mais de vinte milhões de carros sem motorista estejam em funcionamento. Ressalta-se, no entanto, que em termos tecnológicos, ainda há muito o que ser desenvolvido até o atingimento da plena autonomia dos veículos (COREN, 2017).

Além disso, ressaltam-se as previsões positivas da *International Federation of Robotics* (IFR) acerca do crescimento do campo de robótica em geral. Em adição ao aumento



percentual do número total de robôs e do valor de venda nos últimos anos, a IFR projeta que no futuro haverá acréscimo significativo da quantidade de robôs de serviço em operação e que as áreas de automação e robótica permanecerão responsáveis pela geração de novos empregos de alta qualidade no mundo inteiro (IFR, 2016; IFR, 2017).

Por fim, o trabalho motiva-se também pelo avanço da área de pesquisa em robótica móvel no Centro Universitário UNIFTEC, visto que este campo já fora previamente abordado em projetos de conclusão anteriores do curso de Engenharia de Computação.

Neste artigo, a Seção 2 apresenta o referencial teórico no qual este trabalho está baseado, a Seção 3 apresenta a metodologia de pesquisa empregada neste trabalho, a Seção 4 aborda os projetos de *hardware* e de *software* do protótipo de robô móvel, a Seção 5 apresenta os experimentos e resultados obtidos com o robô em quatro *datasets* com condições distintas e a Seção 6 descreve as conclusões e considerações finais.

2 REFERENCIAL TEÓRICO

Para Rosário (2005), a robótica é a ciência que une a engenharia mecânica, a engenharia elétrica e a inteligência artificial para a criação de dispositivos mecatrônicos. De maneira complementar, Thrun, Burgard e Fox (2005) definem a robótica como a ciência que lida com a percepção e manipulação de dispositivos controlados por computadores. A robótica é, portanto, uma área multidisciplinar que envolve, além dos elementos tradicionais de engenharia e ciência da computação, contribuições dos campos de ciência cognitiva, psicologia e filosofia (NEHMZOW, 2012).

Desde a sua concepção, a robótica passou por diferentes paradigmas. A cibernética, ciência que relaciona a teoria de controle aos sistemas complexos, precede a robótica e foi utilizada como sua primeira abordagem. Nesse contexto, o único objetivo do controlador do robô é minimizar a discrepância entre o estado observado pelos sensores e o estado desejado através de uma função de controle (NEHMZOW, 2012).

Derivado da cibernética, o paradigma baseado em modelo (model-based paradigm) adotava um estado determinístico do robô cujas variáveis estivessem contidas plenamente em um único modelo global. Essa abordagem assumia um funcionamento realimentado e tinha por fundamentação a representação simbólica das entidades, também encontrada no campo da inteligência artificial (THRUN; BURGARD; FOX, 2005; NEHMZOW, 2012).



Apesar de impulsionar a área de pesquisa em robótica móvel durante os anos 1960 e 1970, logo encontraram-se desvantagens e fraquezas no paradigma baseado em modelo. Em primeiro lugar, a suposição de um modelo global significava que o robô poderia lidar somente com situações em que as fontes de incerteza pudessem ser mitigadas. Além disso, a construção e manutenção de uma representação completa do estado requeria poder de processamento e requisitos de memória que não estavam disponíveis na época. Por último, a própria abordagem simbólica sofreu críticas pela comunidade de inteligência artificial (THRUN; BURGARD; FOX, 2005; NEHMZOW, 2012).

Para remediar essas dificuldades, surgiu, na década de 1980, o paradigma baseado em comportamento (behavior-based paradigm). Nele, a utilização de um modelo interno do estado é completamente eliminada e, em seu lugar, aplica-se o conceito de acoplamento forte (tight coupling), no qual cada comportamento do robô tem acesso direto aos sensores e controla diretamente os atuadores. Portanto, essa abordagem vale-se do princípio do comportamento emergente (emergent behavior) que propõe que comportamentos complexos podem originar-se de tarefas simples aparentemente desconexas (THRUN; BURGARD; FOX, 2005; NEHMZOW, 2012).

Ainda assim, o paradigma baseado em comportamento restringiu-se a tarefas relativamente simples, já que a ausência de um modelo interno tornava difícil a especificação de planos de ação para um robô que somente respondia diretamente a estímulos sensoriais do ambiente. O problema dessa abordagem encontra-se no complexo mapeamento entre uma sequência de controles e sua tradução para comportamentos que executem as tarefas (THRUN; BURGARD; FOX, 2005; NEHMZOW, 2012).

Para atender às limitações presentes nessas abordagens, foi desenvolvido, na década de 1990, um paradigma na intersecção dos dois anteriores, denominado de paradigma híbrido (hybrid paradigm). Essa abordagem aproveitou as vantagens dos métodos anteriores de forma complementar, ou seja, as técnicas baseadas em comportamento eram utilizadas para controle em baixo nível, enquanto o planejamento baseado em modelo era aplicado à coordenação em alto nível (THRUN; BURGARD; FOX, 2005).

Na mesma década, surgiu um quarto paradigma chamado de robótica probabilística (probabilistic robotics). Assim como a abordagem híbrida, utiliza as medições dos sensores e representações internas por modelo, porém assume que ambas são insuficientes para o controle do robô. Portanto, a robótica probabilística aplica a teoria de probabilidades e a



inferência bayesiana para lidar com as incertezas do ambiente e para integrar os dados dos sensores aos modelos (THRUN; BURGARD; FOX, 2005).

A menos que o ambiente tenha sido construído ou modificado especificamente para uso por robôs, predomina a incerteza por conta da imprevisibilidade e dinamicidade dos espaços humanos. Além disso, os próprios componentes do sistema mecatrônico realimentado (sensor, atuador e controlador) estão sujeitos à fenômenos físicos que impedem seu desempenho ideal (THRUN; BURGARD; FOX, 2005).

Os sensores, por exemplo, estão suscetíveis ao ruído e seus transdutores apresentam limitações de alcance e de precisão. Os atuadores, além de também sofrerem de ruído, não estão livres de falhas elétricas e mecânicas. O software embarcado no controlador do robô também é fonte de incerteza, visto que qualquer modelo de representação interna, por mais complexo que seja, é somente uma abstração do mundo real e, portanto, aproxima os processos físicos. Em adição, os recursos computacionais de processamento, memória e interconexão são limitados (especialmente em sistemas embarcados) e, muitas vezes, opta-se por desempenho ao invés de precisão (THRUN; BURGARD; FOX, 2005).

Devido ao acúmulo de incertezas de fontes internas e externas ao robô, surgiu o paradigma da robótica probabilística, cuja representação do estado é baseada na teoria de probabilidades, de modo a prover ferramentas matematicamente rigorosas para a modelagem das incertezas. Em particular, a abordagem probabilística adota como filosofia a inferência bayesiana, através da qual as observações de sensoriamento e as ações de controle são integradas continuamente às estimativas de estado do robô (THRUN; BURGARD; FOX, 2005).

A fim de conferir autonomia real a um robô móvel, é imprescindível que ele possua a capacidade de construir representações internas do ambiente que serão utilizadas nos processos subsequentes de raciocínio (NEHMZOW, 2012). A criação de um mapa do mundo físico concomitante a determinação da postura do robô dá-se o nome de problema de localização e mapeamento simultâneos (simultaneous localization and mapping – SLAM). Esse assunto também é abordado na literatura sob os nomes de mapeamento e localização concorrentes (concurrent mapping and localization – CML), mapeamento robótico (robotic mapping) e está intimamente ligado ao conceito de structure from motion (SFM) do campo de visão computacional (THRUN; BURGARD; FOX, 2005).

Além de todas as fontes de incerteza com as quais deve-se lidar, o fato de o robô necessitar localizar-se sem possuir informações prévias acerca do ambiente e precisar mapeá-

lo sem conhecer sua verdadeira postura tornam o SLAM “um dos problemas mais fundamentais da robótica” (THRUN; BURGARD; FOX, 2005, p. 309, tradução nossa). No que diz respeito aos processos de mapeamento e navegação, os robôs sofrem de adversidades que, para os humanos e outros animais, são naturais (THRUN; BURGARD; FOX, 2005; MURPHY, 2000).

Em adição aos componentes de localização e mapeamento, há uma terceira questão inerente ao problema de SLAM: o problema de correspondência (correspondence problem) ou problema de associação de dados (data association problem), que se refere à detecção de locais no ambiente que já foram anteriormente visitados, ou, em termos técnicos, a identificação de fechamentos de ciclo (loop closures) dentro do mapa (THRUN; BURGARD; FOX, 2005; MURPHY, 2000).

De todos os quatro paradigmas da robótica mencionados anteriormente, o que obteve maior sucesso no desenvolvimento de soluções para o problema de SLAM foi a abordagem probabilística, devido, em grande parte, a adequação da fundamentação matemática em cálculo de probabilidades para o tratamento das fontes de incerteza (THRUN; BURGARD; FOX, 2005). A aplicação da estatística deve-se a Smith e Cheeseman (1986), que formularam uma solução para o problema de SLAM dentro do contexto da inferência bayesiana ao expressar as variáveis de estado do robô (e suas incertezas associadas) por distribuições de probabilidade integradas por meio da filtragem estocástica.

Em robótica probabilística, as soluções para o problema de SLAM, referidas simplesmente por algoritmos ou técnicas de SLAM, são categorizadas de diversas maneiras em relação a suas especificidades. Quando a solução mantém somente o estado (postura do robô e mapa) do instante de tempo atual e descarta as variáveis passadas, trata-se de uma técnica de SLAM online. No entanto, se a solução mantém a trajetória completa do robô desde o início da sessão de mapeamento, diz-se que é uma técnica de SLAM completo (full SLAM) (THRUN; BURGARD; FOX, 2005).

No SLAM volumétrico (volumetric SLAM) ou SLAM denso (dense SLAM), aproveitam-se os dados brutos dos sensores em totalidade; em contraste, no SLAM baseado em características (feature-based SLAM), pontos de interesse são extraídos das observações dos sensores antes da incorporação ao estado do robô. Em geral, a qualidade dos resultados no SLAM volumétrico é maior, porém, devido a menor quantidade de informações a serem processadas, o SLAM baseado em características é mais eficiente (THRUN; LEONARD, 2008).



As técnicas de SLAM topológico representam somente as relações básicas entre as localizações do mapa, enquanto no SLAM métrico, há indicação da disposição dos elementos do ambiente dentro de um sistema de referência. Os algoritmos estáticos supõem que não ocorrem mudanças no mapa ao longo do tempo, por outro lado, os algoritmos dinâmicos são capazes de lidar com alterações no ambiente. Também se divide as soluções de SLAM em relação à identidade dos objetos do mundo físico: os pontos de referência podem ser identificados unicamente, ou, em caso contrário, o algoritmo provê maneiras de estimar as correspondências (THRUN; LEONARD, 2008).

Durrant-Whyte e Bailey (2006) observam que o problema de SLAM pode ser considerado teoricamente e conceitualmente solucionado. O consenso na área é que ainda permanecem desafios para a aplicação prática dos algoritmos, em especial em ambientes volumosos, complexos e dinâmicos. Destaca-se a evolução das técnicas em paralelo ao avanço dos recursos computacionais de processamento e memória, bem como a introdução de contribuições do campo de visão computacional. Em consonância ao objetivo de conferir real autonomia aos robôs móveis, o SLAM visa a persistência, ou seja, a execução contínua por longo prazo sem intervenção humana (DURRANT-WHYTE; BAILEY, 2006; BAILEY; DURRANT-WHYTE, 2006; THRUN; LEONARD, 2008).

Dentre os algoritmos de SLAM, a técnica selecionada como objeto de estudo deste projeto foi o RTAB-Map, em continuidade ao projeto de Estágio II realizado previamente, cujo objetivo foi analisar o funcionamento do algoritmo e avaliá-lo em diferentes especificações de hardware. Sucintamente, as motivações para escolha do algoritmo em particular foram: licença open-source BSD de 3 cláusulas (3-clause BSD), desenvolvimento contínuo até a presente data, suporte a câmera RGB-D de baixo custo Microsoft Kinect, suporte a plataformas com hardware limitado (Raspberry Pi) e rica documentação (INTROLAB, 2017).

O RTAB-Map (Real-Time Appearance-Based Mapping), introduzido por Labbé e Michaud (2014), classifica-se como SLAM visual baseado em grafos e com detecção de fechamento de ciclos por aparência com foco no cumprimento das restrições de processamento em tempo real. A técnica é uma combinação de dois algoritmos: detecção de fechamento de ciclos com mecanismo de gerenciamento de memória e SLAM baseado em grafo de posturas com otimização.

3 MATERIAIS E MÉTODOS

Segundo Marconi e Lakatos (2003), a metodologia científica é o conjunto de técnicas sistemáticas que descrevem o modo que a pesquisa é conduzida para atingir seus objetivos a fim de detectar os erros e embasar as decisões do pesquisador. O presente trabalho tem por objeto de estudo a área de localização e mapeamento robótico, com ênfase na aplicação de técnicas baseadas em visão computacional a plataformas com especificações limitadas de hardware, de acordo com os objetivos de pesquisa supracitados. Em particular, estabelece-se como hipótese de pesquisa a possibilidade de execução de um algoritmo de mapeamento robótico tridimensional em um computador de placa única com cumprimento dos requisitos de tempo real.

Quanto a finalidade, este trabalho classifica-se como pesquisa aplicada que, de acordo com Gil (2002, p. 18), refere-se aos estudos realizados “para a solução de problemas práticos”. O projeto assim se caracteriza devido a construção de um protótipo funcional para experimentação no mundo real, em contraste aos trabalhos relacionados anteriores que trataram de questões puramente teóricas. Em relação à modalidade, o trabalho enquadra-se como pesquisa experimental, segundo o critério de Köche (2011), que envolve a manipulação das variáveis associadas as hipóteses. Especificamente neste projeto, os parâmetros da aplicação de mapeamento robótico serão configurados de modo que a hipótese de pesquisa possa ser avaliada.

Nesse contexto, a coleta de dados dos experimentos é de natureza quantitativa, já que todas as informações foram obtidas de maneira puramente numérica, sem análise subjetiva do autor. Os experimentos envolveram a execução da aplicação de mapeamento robótico e a manipulação dos parâmetros a fim de alcançar os objetivos de pesquisa. Durante as sessões de execução, foram extraídas diversas métricas que permitiram posterior análise estatística dos resultados e resposta a hipótese de pesquisa. Na aplicação, ao término do processo de mapeamento, os dados de desempenho são gravados em logs que contém os tempos de processamento, quantidade de localizações nas memórias, quantidade de nodos adicionados aos grafos, tamanho do dicionário de palavras visuais e uso de memória. Para análise e visualização dos dados coletados, os arquivos de log são processados por um script em MATLAB que gera os gráficos das métricas.

As principais métricas observadas são a taxa de atualização, a quantidade de detecções de fechamento de ciclo aceitas e rejeitadas, o tamanho da memória de trabalho, o



limite de tempo de processamento e o tempo total de processamento para cada novo nodo. As métricas de precisão (precision), que indica a taxa de itens selecionados que são relevantes, e de revocação (recall), que representa a taxa de itens relevantes que são selecionados, não foram aplicadas a este estudo, visto que os datasets obtidos não contam com a ground truth, ou seja, a verdade-base. No contexto de SLAM visual baseado em aparência, os falsos positivos representam localizações distintas as quais se foi erroneamente atribuído um fechamento de ciclo; e os falsos negativos representam localizações iguais nas quais o fechamento de ciclo não foi detectado. Enquanto os falsos negativos são prejudiciais para o funcionamento do algoritmo, os falsos positivos são destrutivos, porque, a partir do momento de uma detecção incorreta, o mapa não mais reflete o mundo físico e não há sentido em continuar o processo de localização e mapeamento. Por essa razão, na área de SLAM em geral, todos os experimentos requerem uma precisão que não pode ser diferente de 100%, enquanto a métrica comumente mensurada para avaliação da qualidade do algoritmo é, portanto, a revocação.

As métricas supracitadas foram escolhidas por estarem relacionados ao critério de avaliação de cumprimento do requisito de funcionamento em tempo real. Na definição de Laplante (2004, p. 4, tradução nossa), um sistema de tempo real é “um sistema que deve satisfazer explicitamente restrições (limitadas) de tempo de resposta ou arriscar consequências severas, incluindo a falha”. No contexto do projeto, a restrição de tempo de resposta (deadline) é o intervalo entre cada observação extraída pelo sensor, ou seja, a taxa de atualização do algoritmo. Se o tempo de processamento de uma imagem ultrapassar o período de aquisição, diz-se que o sistema não foi capaz de cumprir o requisito de tempo real. Mesmo assim, a perda de alguns deadlines não implica na falha total do sistema, visto que o algoritmo continua o processamento e procura retomar a execução em tempo abaixo do limite. Por isso, de acordo com a classificação de Laplante (2004), o sistema pode ser classificado como firm real-time, isto é, há tolerância para um número limitado de perdas de deadline. Por exemplo, se a taxa de captura do sensor está parametrizada em 2Hz, cada nova observação deve necessariamente ser processada em, no máximo, 500ms; caso contrário, o estado do grafo de posturas do robô não estará atualizado com a observação mais recente.

A fim de extrair as métricas, foram elaborados experimentos em diferentes cenários que influenciassem em suas características de tal maneira que permitissem a inferência de explicações para justificar os resultados obtidos. Os testes envolveram a variação do ambiente (interno e externo) e da presença de fechamento de ciclos com o propósito de determinar o

impacto das quatro combinações possíveis nas métricas, com ênfase na quantidade de fechamentos de ciclo detectados e no tempo total de processamento por nodo. Os dados coletados foram analisados a fim de elaborar conclusões que embasassem o comportamento observado para que assim fosse possível a avaliação do cumprimento dos objetivos estabelecidos.

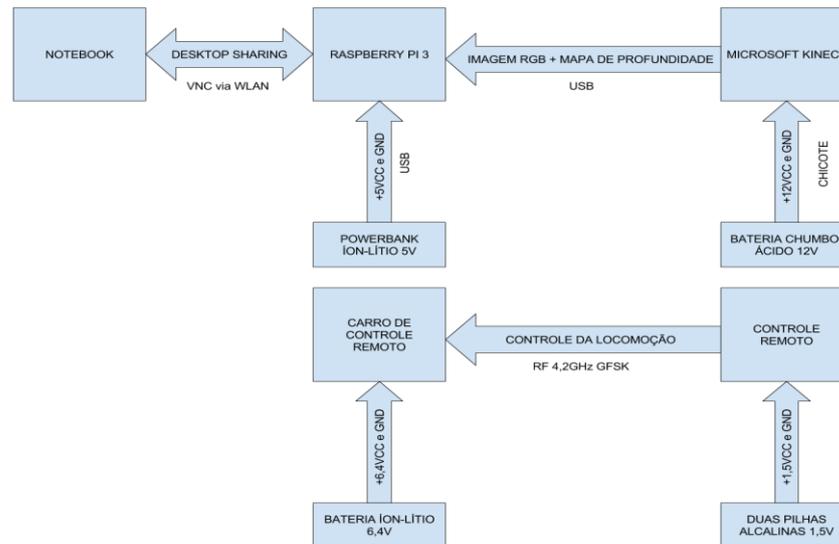
4 PROJETO

O projeto consiste em um robô móvel de localização e mapeamento visual tridimensional. Classifica-se como robô de autonomia fraca, ou seja, carrega consigo suas fontes de alimentação e seus recursos de processamento, porém não é capaz de decidir seu percurso de forma autônoma. Na escala de níveis de autonomia, enquadra-se no nível mais básico de teleoperação, no qual todas as ações de controle são transmitidas por um operador humano. Em relação à locomoção, o projeto enquadra-se na categoria de robô móvel baseado em rodas.

De acordo com as categorias da robótica probabilística, o *software* de SLAM embarcado (RTAB-Map) é classificado como: visual, tridimensional, completo, volumétrico, métrico, estático, sem correspondências conhecidas, passivo, de único robô, de tempo real e com detecção global de fechamento de ciclos. Apesar de estar limitado a ambientes planares (três graus de liberdade) por conta do sistema de locomoção, o processo de SLAM visual escolhido assume ambientes volumétricos (seis graus de liberdade); portanto, são extraídas das observações as três variáveis cartesianas e as três variáveis de rotação, embora o robô não tenha controle sobre todas elas.

A Figura 1 descreve a arquitetura geral do robô móvel do projeto, com os principais componentes e a conexão entre eles. Nota-se que o sistema de locomoção do robô, composto pelo carro e seu controle remoto, estão fisicamente separados do módulo de sensoriamento e processamento, que consiste no computador de placa única Raspberry Pi 3 e na câmera RGB-D Microsoft Kinect. A visualização do processo de mapeamento e a transferência dos dados coletados é realizada por meio de uma aplicação de *desktop sharing* com um *notebook* através de conexão sem fio (Wi-Fi).

Figura 1 – Diagrama de blocos da arquitetura do robô



Fonte: Elaborado pelo autor (2017).

A Figura 2 apresenta o aspecto externo do robô em seu estado final.

Figura 2 – Aspecto externo do robô móvel teleguiado



Fonte: Elaborado pelo autor (2017).

O *software* do robô móvel consiste nas aplicações instaladas no sistema operacional embarcado no computador de placa única Raspberry Pi 3. Em particular, duas aplicações são essenciais para o funcionamento do projeto: o programa de *desktop sharing* RealVNC, através do qual o andamento do processo de mapeamento é monitorado; e a aplicação de SLAM visual tridimensional RTAB-Map, que realiza o processo de localização e mapeamento propriamente dito.

A escolha das especificações dos projetos de *hardware* e de *software* ocorreu por meio de avaliação e experimentação dos componentes. A contribuição do trabalho, portanto, está ligada a determinação da configuração de montagem para construção de um protótipo de



robô móvel teleguiado baseado em rodas com objetivo de execução de aplicação de localização e mapeamento simultâneos tridimensional.

5 RESULTADOS E DISCUSSÃO

Com o intuito de cumprir o objetivo de avaliar o desempenho da aplicação de mapeamento (RTAB-Map) embarcada no robô móvel teleguiado, foram realizados quatro experimentos em *datasets* distintos extraídos pelo autor. O primeiro consiste na avaliação da solução de SLAM visual em um ambiente externo, sem fechamento de ciclo, com a determinação dos parâmetros necessários para execução satisfatória, ou seja, em conformidade com os requisitos de tempo real e ausência de falsos positivos. O segundo experimento avalia a execução do algoritmo em um ambiente externo, com detecção de fechamento de ciclo. O terceiro experimento, por sua vez, testa a aplicação em um ambiente interno, sem detecção de fechamento de ciclo, e busca o mesmo objetivo dos anteriores: cumprimento dos objetivos de execução. O quarto experimento busca a extração das métricas em um ambiente interno, com detecção de fechamento de ciclo.

Para fins de reprodutibilidade, os dados das sessões de mapeamento foram capturados diretamente através da aplicação RTAB-Map e armazenados no formato de banco de dados padrão do *software* para posterior reprocessamento. Desse modo, as observações mantinham-se exatamente como o robô móvel as obtivera e permitia a experimentação dos *datasets* com diferentes parâmetros de configuração até o cumprimento dos objetivos.

Os parâmetros iniciais da execução do algoritmo são baseados na configuração padrão da aplicação e nas recomendações dos autores (INTROLAB, 2017). Os parâmetros foram modificados caso não provesses execução satisfatória da sessão, ou seja, se falhassem em garantir tempo real ou se não se mostrassem adequados para a detecção de fechamentos de ciclo.

Uma importante alteração das configurações padrão é a aplicação dos processos de pré e pós-decimação de imagem. Observou-se que o maior ganho de desempenho resulta da redução da resolução das imagens capturadas (RGB e de profundidade). Na pré-decimação, o tamanho da imagem é reduzido antes da extração de *features* e, na pós-decimação, as observações são salvas no banco de dados em resolução menor. Para ambos os processos, foi empregado um fator de duas vezes, ou seja, o tamanho de cada imagem era reduzido pela metade. Em contrapartida, devido a menor quantidade de *pixels* por medição, perdeu-se

qualidade na extração de *features* das imagens.

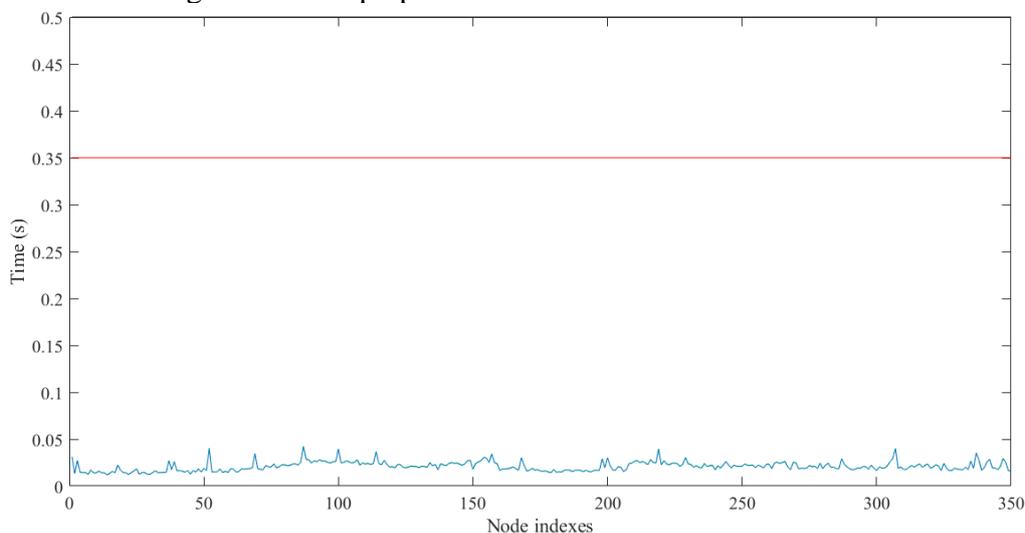
5.1 DATASET EXTERNO ACÍCLICO

Em 11 de novembro de 2017, o robô foi teleguiado por uma seção retangular de pátio externo no turno da manhã com clima ensolarado. Em seu percurso, o robô não retornou em nenhum momento a uma localização previamente visitada, isto é, não houve fechamento de ciclo. Já que a arquitetura do robô utiliza uma câmera passiva (sem movimentação independente), todas as imagens estão necessariamente na direção de locomoção, salvo nos momentos em que o robô anda no sentido inverso (ré).

As imagens RGB e de profundidade foram capturadas pelo sensor Kinect a uma taxa de 2 Hz e nenhum dos demais parâmetros padrão foi alterado, já que foram adequados para execução satisfatória do algoritmo. A taxa de captura de 2 Hz implica em um tempo máximo de processamento por nova observação de 500 ms (meio segundo); portanto, seguiu-se a orientação dos autores Labbé e Michaud (2013) em utilizar-se um valor de limite de processamento T_{time} entre 60 e 80% do tempo máximo. Por isso, o parâmetro T_{time} foi ajustado em 350 ms, a partir do qual o processo de transferência passa a tomar efeito.

Durante a sessão de mapeamento, foram capturadas 350 imagens em um percurso total de 27,1 metros. A Figura 3 apresenta o gráfico de tempo total de processamento por nodo de localização (em azul). O limite T_{time} de 350 ms é indicado em vermelho e o tempo máximo de processamento por observação (500 ms) está em preto. O tempo médio de processamento foi de 21,1 ms com tempo máximo de 41,34 ms.

Figura 3 – Tempo por nodo no *dataset* externo acíclico



Fonte: Elaborado pelo autor (2017).



Observa-se que em nenhum instante o tempo de processamento por nodo ultrapassou o tempo máximo de processamento, de modo que sequer atingiu o parâmetro limite T_{time} . Esse comportamento pode ser explicado pela baixa quantidade média de palavras visuais extraídas por assinatura e respectiva reduzida carga computacional para computação da similaridade entre as observações. A média de palavras visuais por assinatura foi de 7,79 palavras com média do tamanho do vocabulário de 373,97 palavras e tamanho máximo de 586 palavras. A quantidade limitada de *features* retiradas das imagens indica a baixa densidade de objetos típica de ambientes externos, agravada pelo curto alcance da câmera Kinect (quatro metros).

Como esperado do *dataset*, não houve detecções de fechamentos de ciclo aceitas nem rejeitadas, o que indica que os parâmetros configurados foram adequados para correta identificação das localizações sem ocorrência de falsos positivos. Portanto, observa-se que nesse *dataset*, os objetivos de cumprimento dos requisitos de tempo real e ausência de detecções espúrias foram atingidos, dados os parâmetros mencionados.

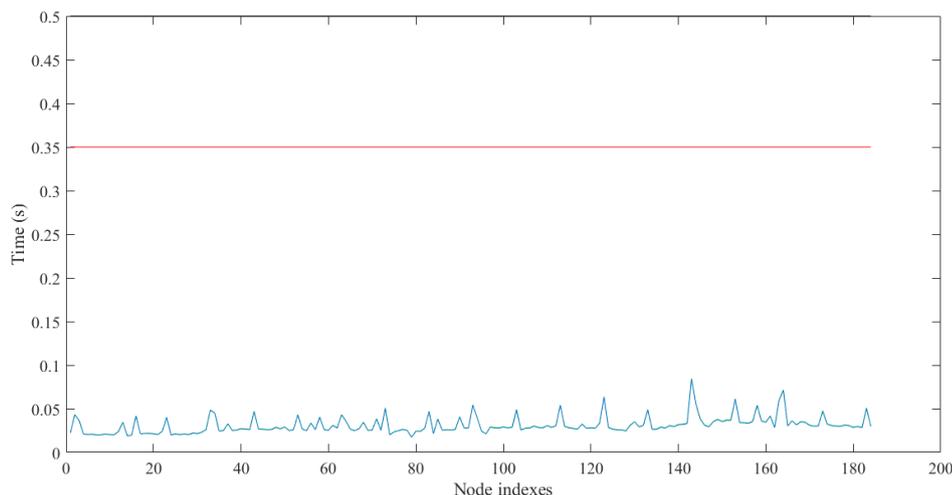
5.2 DATASET EXTERNO CÍCLICO

No dia 30 de novembro de 2017, o robô móvel foi teleguiado por um corredor de pátio externo no turno da tarde com clima nublado. A fim de avaliar a detecção de fechamento de ciclo, a trajetória do robô envolveu percorrer toda a extensão do corredor e, em seguida, retornar para o ponto inicial da sessão em sentido inverso (de ré) para testar a capacidade de o algoritmo corretamente identificar o retorno para uma localização previamente visitada.

Assim como no processamento do primeiro *dataset*, a taxa de captura das observações foi especificada na frequência de 2 Hz, portanto utilizou-se o mesmo parâmetro limite de tempo de processamento T_{time} de 350 ms. Dentre as demais configurações da aplicação de mapeamento, foram alterados dois parâmetros padrão: a quantidade mínima de *inliers* das *features* para aceitação da odometria foi reduzida de 20 para 9 palavras, visto que a quantidade média de palavras extraídas das imagens (como será mencionado adiante) ficou abaixo desse número, impossibilitando a detecção de fechamento de ciclos com o parâmetro padrão; e o limite de aceitação da hipótese de fechamento de ciclo T_{loop} foi reduzido de 11% para 6%, a fim de facilitar a detecção durante a sessão de mapeamento. Nota-se que a redução do limite T_{loop} , embora auxilie na detecção de verdadeiros positivos, tem o efeito colateral de

aumentar as chances de ocorrência de falsos positivos, já que as condições de aceitação se tornam menos restritivas; portanto, o limite utilizado representa um equilíbrio entre a correta identificação de fechamentos de ciclo e a ausência de falsos positivos.

Figura 4 – Tempo por nodo no *dataset* externo cíclico



Fonte: Elaborado pelo autor (2017).

No *dataset* externo cíclico, foram extraídas 184 imagens em uma trajetória total de 16,6 metros (considerando-se as partes do percurso revisitadas). Na Figura 4, apresenta-se o gráfico de tempo total de processamento por nodo de localização (em azul). Novamente, o limite T_{time} de 350 ms está em vermelho e o tempo máximo de 500 ms está em preto. A média do tempo de processamento foi de 31,3 ms e o maior tempo foi de 73,24 ms.

De maneira semelhante ao ocorrido no *dataset* anterior, nem o parâmetro de tempo limite nem o tempo máximo foram ultrapassados em nenhum momento. A maior dificuldade de extração de *features* em ambientes externos devido à baixa densidade de objetos resulta em menor carga computacional. Ainda assim, a quantidade média de palavras visuais por assinatura foi superior a do primeiro *dataset*: 15,66 palavras com média do tamanho do vocabulário de 252,78 palavras e tamanho máximo de 555 palavras. No entanto, como citado anteriormente, a média de palavras ficou abaixo do parâmetro padrão de aceitação de fechamento de ciclos (20 palavras); por essa razão, o número foi reduzido para possibilitar a correta identificação.

Durante o processamento do *dataset*, uma detecção de fechamento de ciclo foi aceita, a qual constatou-se tratar de um verdadeiro positivo. O número de detecções rejeitadas foi de 122 localizações, dentre as quais estavam falsos negativos e verdadeiros negativos. Portanto, observa-se que com a redução dos parâmetros de número mínimo de *inliers* para aceitação da odometria e de limite de detecção de fechamento de ciclo, a sensibilidade do algoritmo

aumentou tal que a quantidade de detecções rejeitadas foi considerável. Apesar da perda da qualidade com a rejeição de falsos negativos, ressalta-se que um possível aumento da sensibilidade acarretaria com maior probabilidade a ocorrência de um falso positivo destrutivo. Assim, considera-se que os objetivos de tempo real e de detecção de fechamento de ciclos foram atingidos para esse *dataset*, visto que o fechamento de ciclo foi corretamente identificado.

5.3 DATASET INTERNO ACÍCLICO

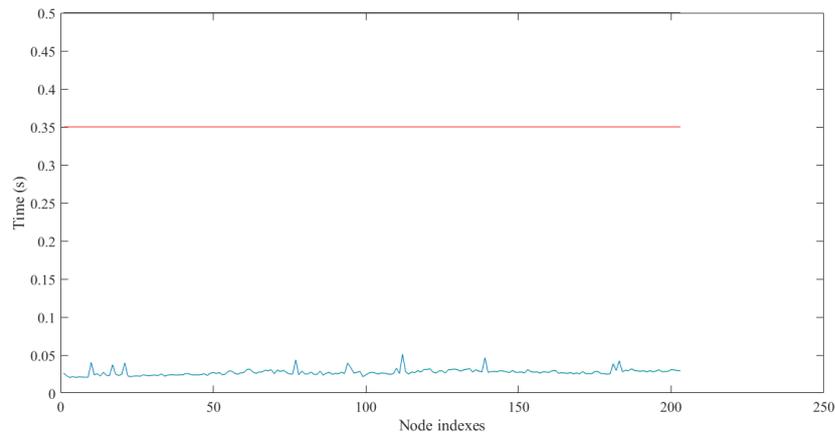
Em 24 de novembro de 2017, no turno da tarde, o robô foi teleguiado por uma seção do corredor do primeiro andar do bloco principal do Centro Universitário UNIFTEC. O robô seguiu uma trajetória linear, sem voltar para nenhuma localização anterior durante seu percurso. Assim como nos dois *datasets* anteriores, a taxa de captura das imagens RGB e de profundidade foi de 2 Hz; portanto, os parâmetros de tempo limite T_{time} e de tempo máximo de processamento ficaram novamente em 350 ms e 500 ms, respectivamente. Por resultarem em execução satisfatória da aplicação, nenhum dos parâmetros padrão foi modificado.

O robô percorreu 32,4 metros de distância, capturando no caminho 204 imagens. A Figura 5 expõe o gráfico de tempo total de processamento por nodo de localização em azul, o limite T_{time} em vermelho e o tempo máximo em preto. O tempo médio de processamento ficou em 27,55 ms com tempo máximo de 49,55 ms.

Verifica-se através do gráfico que o tempo de processamento por nodo não ultrapassou o limite de tempo T_{time} . A média de palavras visuais por assinatura foi de 42,7 palavras com média do tamanho do vocabulário visual de 895,44 palavras e tamanho máximo de 1653 palavras. Embora a quantidade de palavras visuais extraídas das observações foi superior à dos *datasets* anteriores (por tratar-se de um ambiente interno no qual a densidade de objetos dentro do raio de alcance do sensor é maior), a aplicação foi capaz de manter o tempo de processamento dentro do requisito de tempo real, dada a correta parametrização.

Em relação às detecções de fechamento de ciclo, houve zero detecções aceitas e uma detecção rejeitada. Portanto, o algoritmo foi capaz tanto de identificar corretamente que o robô não retornou para uma localização previamente visitada, quanto de reconhecer que a hipótese de fechamento de ciclo detectada se tratava de um verdadeiro negativo. Embora o ambiente do *dataset* apresentasse fortemente a característica de *aliasing* perceptual, o robô conseguiu cumprir os objetivos de tempo real e de ausência de falsos positivos.

Figura 5 – Tempo por nodo no *dataset* interno acíclico



Fonte: Elaborado pelo autor (2017).

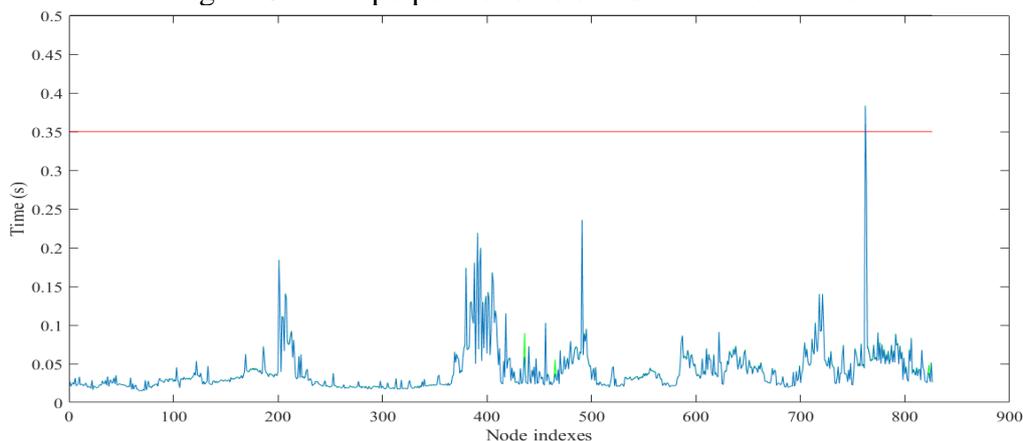
5.4 DATASET INTERNO CÍCLICO

Em 27 de novembro de 2017, o robô foi teleguiado pelo segundo andar de um sobrado. Durante o percurso, o robô passou por diversos cômodos e em várias ocasiões passou por locais em que já havia estado. O objetivo do processamento desse *dataset* foi, portanto, a avaliação da capacidade de a aplicação detectar fechamentos de ciclo em ambientes internos.

Assim como nos demais *datasets*, a frequência de captura de imagens foi de 2 Hz, mantendo-se o limite T_{time} em 350 ms e o tempo máximo de processamento por imagem em 500 ms. Com a exceção de um dos parâmetros, toda a configuração do algoritmo permaneceu em seu estado padrão. O limite de aceitação da hipótese de fechamento de ciclo T_{loop} foi decrescido de 11% para 10% para relaxar o requerimento de detecção sem influenciar de forma significativa na ocorrência de falsos positivos.

A Figura 6 descreve o gráfico de tempo de processamento por localização (em azul) com o limite de tempo T_{time} de 350 ms (em vermelho) e o tempo máximo de 500 ms (em preto). A média do tempo de processamento por nodo foi de 41,63 ms e o maior tempo foi de 380,64 ms. Observa-se que a maior carga computacional comparada aos *datasets* anteriores deve-se, principalmente, a dois fatores: a maior quantidade de *features* extraídas das imagens em razão da natureza do ambiente interno amostrado e a duração da sessão de mapeamento que resultou na captura de uma quantidade de observações (826 imagens em 70,8 metros) consideravelmente superior à das outras sessões.

Figura 6 – Tempo por nodo no *dataset* interno cíclico



Fonte: Elaborado pelo autor (2017).

A quantidade média de palavras visuais por assinatura foi de 45,05 palavras com média do tamanho do vocabulário visual de 3829,2 palavras e tamanho máximo do dicionário de 8443 palavras. Embora uma maior quantidade de *features* tenha sido extraída do *dataset*, o algoritmo foi capaz de manter a execução em tempo real (abaixo do limite máximo de 500 ms), ultrapassando o parâmetro limite T_{time} de 350 ms somente durante o processamento de um dos nodos, ou seja, o processo de transferência de localizações da WM para a LTM foi ativado por um curto período.

Em relação aos fechamentos de ciclo, houve sete detecções aceitas e dezoito detecções rejeitadas. Constatou-se que todos os fechamentos aceitos foram verdadeiros positivos e dentre os rejeitados estavam falsos negativos e verdadeiros negativos. Portanto, a redução no parâmetro limite de aceitação de hipótese facilitou a identificação de verdadeiros positivos, porém não influenciou a sensibilidade do algoritmo a ponto de permitir falsos positivos. Dessa forma, considera-se que nesse *dataset*, os objetivos de tempo real e de ausência de detecções espúrias foram atingidos, em especial, dada a complexidade e duração da sessão de mapeamento do ambiente.

5.5 DISCUSSÃO DOS RESULTADOS

Com a avaliação do robô móvel teleguiado com aplicação de mapeamento embarcada nos quatro *datasets* em cenários diferentes, fica verificada a capacidade de execução em conformidade com os objetivos de tempo real e de ausência de falsos positivos, dados os parâmetros adequados de configuração. Observa-se que, em geral, os parâmetros padrão da aplicação estavam corretamente dimensionados, somente houve necessidade de alteração de alguns parâmetros para facilitar o reconhecimento de fechamentos de ciclo nos



datasets onde a detecção era esperada. Em particular, o parâmetro com maior contribuição para os resultados foi a decimação das imagens, visto que efetivamente a quantidade de dados a serem processados decaiu pela metade.

Uma característica evidente nos gráficos de tempo por nodo dos três primeiros *datasets* é a presença de picos de processamento entre períodos de computação relativamente constante, que podem ser atribuídos aos momentos durante a sessão de mapeamento nos quais o algoritmo perdeu a referência de odometria e, em seguida, foi capaz de recuperá-la na localização seguinte. Corroborando essa hipótese, observa-se que a quantidade de picos nos *datasets* externos é superior à do *dataset* interno acíclico, já que o menor número de *features* extraídas por assinatura de observação nos primeiros é inferior ao do segundo, propiciando a perda de odometria. O quarto *dataset* (interno cíclico), no entanto, exibe um comportamento mais errático do que o dos demais, possivelmente devido à considerável carga computacional a qual ele estava submetido em vista da duração do processo de mapeamento e da densidade de objetos do ambiente amostrado.

Comparando-se a quantidade de palavras visuais extraídas das medições, observa-se novamente a clara distinção entre os *datasets* internos e externos. Nos ambientes externos, os objetos estão mais espaçados uns dos outros (menor densidade) e, dado o curto alcance do sensor utilizado, existem menos *features* que sirvam de referência para o robô, tanto em termos de odometria quanto de aparência. Por outro lado, nos ambientes internos é mais provável que existam objetos dentro do raio de observação do sensor e, conseqüentemente, mais *features* são extraídas por assinatura.

Em relação ao mecanismo de gerenciamento de memória do algoritmo, verificou-se que o processo de transferência de nodos da memória de trabalho (WM) para a memória de longo prazo (LTM) não tomou efeito, exceto em um único caso isolado durante o quarto *dataset*, ou seja, já que o parâmetro de tempo limite de processamento não foi ultrapassado, não houve necessidade de reduzir artificialmente a quantidade de observações verificadas. Portanto, o preenchimento da LTM deu-se somente pela fusão de nodos de acordo com o parâmetro de similaridade de observações, o que evidencia o *aliasing* perceptual dos ambientes selecionados.

Empiricamente, notou-se que a utilização das imagens em sua resolução original ou o aumento da frequência de captura do sensor acima de 5 Hz torna o processo de mapeamento insustentável, já que ambos restringem a operação a sessões de curta duração. Também se percebeu que a utilização de uma plataforma robótica móvel (em vez de operação humana)



elimina a suscetibilidade do algoritmo a aceleração brusca, já que o controle de velocidade é mais preciso. Curiosamente, observou-se que as áreas nos ambientes com alta luminosidade (como os diretamente expostos a luz do sol) interferem com a computação do mapa de profundidade do Kinect, visto que esse é um produto destinado para uso em ambientes internos.

Não obstante, fica afirmada a possibilidade de execução do algoritmo no *hardware* limitado do protótipo de robô móvel. Embora a redução de alguns parâmetros tenha representado a perda de qualidade do modelo gerado, a plataforma é capaz de acomodar os requisitos de tempo real sem a presença de danos irreversíveis nos mapas na forma de falsos positivos.

6 CONSIDERAÇÕES FINAIS

Através dos resultados dos experimentos realizados com o protótipo de robô móvel em diferentes cenários, considera-se que os objetivos deste trabalho foram cumpridos. Verifica-se que a escolha dos componentes de *hardware* e de *software* do projeto foi assertiva, visto que, juntamente com a correta parametrização do algoritmo de SLAM visual, permitiu a execução do processo de localização e mapeamento em tempo real, com reconhecimento dos fechamentos de ciclo nos *datasets* que os continham e sem detecções de falsos positivos.

O detalhamento das especificações do robô móvel, incluindo estrutura física, alimentação, sensores, plataforma de processamento, controle, acoplamento, comunicação e aplicação, explicitou a importância da integração entre *software* e *hardware* nos projetos de robótica, evidenciando a complementariedade dos componentes.

Por meio da execução do algoritmo nos *datasets* obtidos pelo autor, demonstrou-se a utilização do robô e foi possibilitada a extração de métricas que comprovassem o desempenho da aplicação. Desta forma, foram reconhecidos na prática os *tradeoffs* identificados na fundamentação teórica do algoritmo. Notou-se o comportamento do mecanismo de gerenciamento de memória no cumprimento dos *deadlines* de tempo real e o funcionamento do parâmetro de sensibilidade de detecção de fechamento de ciclos: quanto mais permissível for o limite, maior é a probabilidade de aceitação de hipóteses de fechamento, porém maior é a chance de ocorrência de falsos positivos. Verificou-se também as diferenças entre as sessões de mapeamento em ambientes internos e externos de maneira que o impacto das peculiaridades de cada cenário pode ser mensurado nos resultados. Sobretudo, destaca-se a



confirmação da possibilidade de execução do RTAB-Map nas especificações do protótipo de robô móvel.

Por fim, ressalta-se o encaminhamento da área de SLAM para soluções visuais tridimensionais na intersecção entre as áreas de visão computacional e robótica probabilística. Os mais recentes algoritmos do estado da arte expandem o mapeamento robótico com o paradigma baseado em objetos, com a embarcação em plataformas de *hardware* cada vez mais limitado e com a aplicação em agentes com diferentes formas de locomoção, como os *drones* e robôs aquáticos.

Com a demonstração da aplicação de localização e mapeamento simultâneos em um robô móvel teleguiado, o autor abre espaço para trabalhos futuros nas áreas de decisão, planejamento e navegação que avancem o estado do projeto para um contexto de autonomia forte do robô.

REFERÊNCIAS

BAILEY, Tim; DURRANT-WHYTE, Hugh. Simultaneous localization and mapping (SLAM): Part II. **IEEE Robotics & Automation Magazine**, v. 13, n. 3, p. 108-117, 2006. Disponível em: <https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=1678144>. Acesso em: 3 dez. 2017.

COREN, Michael J. **A robotics expert predicts that kids born in 2017 will never drive a car**. 2017. Disponível em: <https://qz.com/879454/a-robotics-expert-predicts-that-kids-born-in-2017-will-never-drive-a-car/>. Acesso em: 3 dez. 2017.

DURRANT-WHYTE, Hugh; BAILEY, Tim. Simultaneous localization and mapping: part I. **IEEE Robotics & Automation Magazine**, v. 13, n. 2, p. 99-110, 2006. Disponível em: <https://ieeexplore.ieee.org/document/1638022?arnumber=1638022>. Acesso em: 3 dez. 2017.

GIL, Antonio Carlos. **Como elaborar projetos de pesquisa**. 4. ed. São Paulo: Atlas, 2002. 176 p.

INTERNATIONAL FEDERATION OF ROBOTICS. **History of Industrial Robots**. 2012a. Disponível em: http://www.ifr.org/fileadmin/user_upload/downloads/forms___info/History_of_Industrial_Robots_online_brochure_by_IFR_2012.pdf. Acesso em: 07 set. 2017.

INTERNATIONAL FEDERATION OF ROBOTICS. **History**. 2012b. Disponível em: <http://www.ifr.org/history/>. Acesso em: 07 set. 2017.

INTERNATIONAL FEDERATION OF ROBOTICS. **Execute Summary World Robotics 2016 Service Robots**. 2016. Disponível em: https://ifr.org/downloads/press/02_2016/Executive_Summary_Service_Robots_2016.pdf. Acesso em: 3 dez. 2017.



INTERNATIONAL FEDERATION OF ROBOTICS. **Robots Create Jobs!** 2017. Disponível em: <https://ifr.org/robots-create-jobs>. Acesso em: 3 dez. 2017.

INTROLAB. **RTAB-Map library and standalone application**. 2017. Disponível em: <https://github.com/introlab/rtabmap>. Acesso em: 16 set. 2017.

IROBOT. **Roomba Robot Vacuums**. 2017. Disponível em: <http://www.irobot.com/For-the-Home/Vacuuming/Roomba.aspx>. Acesso em 3 dez. 2017.

KÖCHE, José Carlos. **Fundamentos de metodologia científica: teoria da ciência e iniciação à pesquisa**. Petrópolis: Vozes, 2011. 185 p.

LABBÉ, Mathieu; MICHAUD, François. Appearance-based loop closure detection for online large-scale and long-term operation. **IEEE Transactions on Robotics**, v. 29, n. 3, p. 734-745, 2013. Disponível em: <https://ieeexplore.ieee.org/document/6459608>. Acesso em: 3 dez. 2017.

LABBÉ, Mathieu; MICHAUD, François. Memory management for real-time appearance-based loop closure detection. *In: PROCEEDINGS OF THE IEEE/RSJ INTERNATIONAL CONFERENCE ON INTELLIGENT ROBOTS AND SYSTEMS*. **Anais eletrônicos...** 2011, p. 1271-1276. Disponível em: <https://ieeexplore.ieee.org/document/6094602>. Acesso em: 3 dez. 2017.

LABBÉ, Mathieu; MICHAUD, François. Online global loop closure detection for large-scale multi-session graph-based SLAM. *In: PROCEEDINGS OF THE IEEE/RSJ INTERNATIONAL CONFERENCE ON INTELLIGENT ROBOTS AND SYSTEMS*. **Anais eletrônicos...** 2014. Disponível em: <https://ieeexplore.ieee.org/document/6942926>. Acesso em: 3 dez. 2017.

LAPLANTE, Phillip A. **Real-time systems design and analysis**. 3. ed. New York: Wiley, 2004. 529 p.

MARCONI, Marina de Andrade; LAKATOS, Eva Maria. **Fundamentos de metodologia científica**. 5. ed. São Paulo: Atlas, 2003. 310 p.

MICROSOFT. **Kinect for Windows Sensor Components and Specifications**. 2017. Disponível em: <https://msdn.microsoft.com/en-us/library/jj131033.aspx>. Acesso em: 12 nov. 2017.

NEHMZOW, Ulrich. **Mobile robotics: a practical introduction**. 2. ed. Springer Science & Business Media, 2012, 280 p.

OMRON. **LD Series**. 2017. Disponível em: <http://www.ia.omron.com/products/family/3664/>. Acesso em: 3 dez. 2017.

OPENCV. **Non-free 2D Features Algorithms**. 2016. Disponível em: http://docs.opencv.org/3.2.0/d2/dca/group__xfeatures2d__nonfree.html. Acesso em: 12 nov. 2017.



RASPBERRY PI FOUNDATION. **Raspberry Pi 3 Model B**. 2017. Disponível em: <https://www.raspberrypi.org/products/raspberry-pi-3-model-b/>. Acesso em: 12 nov. 2017.

RASPBIAN. **About Raspbian**. 2017. Disponível em: <https://www.raspbian.org/RaspbianAbout>. Acesso em: 13 nov. 2017.

ROSÁRIO, João Maurício. **Princípios de mecatrônica**. São Paulo: Prentice Hall, 2005, 350 p.

SMITH, Randall C.; CHEESEMAN, Peter. On the representation and estimation of spatial uncertainty. **The international journal of Robotics Research**, Califórnia, v. 5, n. 4, p. 56-68, 1986. Disponível em: <https://journals.sagepub.com/doi/10.1177/027836498600500404>. Acesso em: 13 nov. 2017.

THRUN, Sebastian. Robotic mapping: A survey. **Exploring artificial intelligence in the new millennium**, Califórnia, v. 1, p. 1-35, 2002. Disponível em: <chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/viewer.html?pdfurl=http%3A%2F%2Frobotic.s.stanford.edu%2F~latombe%2Fcs326%2F2002%2Fthrun-mapping.pdf&clem=1484293&chunk=true>. Acesso em: 13 nov. 2017.

THRUN, Sebastian; BURGARD, Wolfram; FOX, Dieter. **Probabilistic robotics**. 3. ed. Cambridge: The MIT Press, 2005. 647 p.

THRUN, Sebastian; LEONARD, John J. Simultaneous localization and mapping. *In: SPRINGER HANDBOOK OF ROBOTICS*. Springer Berlin Heidelberg, **Anais eletrônicos...** 2008. p. 871-889. Disponível em: https://link.springer.com/referenceworkentry/10.1007%2F978-3-540-30301-5_38. Acesso em: 3 dez. 2017.