

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
CURSO DE ENGENHARIA DE COMPUTAÇÃO

VINÍCIUS MACIEL HOFF

**Sistema de auxílio à locomoção de pessoas
com deficiência visual**

Monografia apresentada como requisito parcial
para a obtenção do grau de Bacharel em
Engenharia da Computação

Orientador: Prof. Dr. Edson Prestes
Co-orientador: Fernanda C. S. Rodrigues

Porto Alegre
2018

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitor: Prof. Rui Vicente Oppermann

Vice-Reitora: Prof^a. Jane Fraga Tutikian

Pró-Reitor de Graduação: Prof. Wladimir Pinheiro do Nascimento

Diretora do Instituto de Informática: Prof^a. Carla Maria Dal Sasso Freitas

Coordenador do Curso de Engenharia de Computação: Prof. Renato Ventura Henriques

Bibliotecária-chefe do Instituto de Informática: Beatriz Regina Bastos Haro

*“Of all men’s miseries, the bitterest is this:
to know so much and have control over nothing.”*

— HERODOTUS

AGRADECIMENTOS

Gostaria de agradecer a todos que de alguma forma colaboraram para a realização desse trabalho. Primeiramente, agradeço à toda minha família, em especial aos meus pais, Sonia e Paulo, minha madrinha, Auria, e minha irmã querida, Aline, por serem a base que me permitiu terminar esse projeto. Agradeço aos meus amigos e amigas pelo apoio e incentivo constante, especialmente à Carolina por me aguentar em todas as horas e à Natália por ser sempre solícita nos momentos que precisei. Agradeço à Fernanda e ao Diego por me guiarem durante a produção desse trabalho. Além disso, agradeço à UFRGS por todas as oportunidades proporcionadas e aos professores e professoras por todo o apoio e disponibilidade apresentado.

RESUMO

Pessoas podem encontrar muitas dificuldades para localizarem-se em ambientes com os quais não estão familiarizadas, mas a tarefa se torna ainda mais difícil quando não se pode contar com informações visuais. Com o objetivo de auxiliar pessoas com deficiências visuais a se deslocarem em ambientes internos desconhecidos, esse trabalho apresenta um sistema de locomoção como uma aplicação de tecnologia assistiva. O sistema proposto utiliza o celular como interface de forma a permitir ao usuário adicionar o destino desejado. Além disso, o aparelho tem o papel de capturar as imagens ao entorno do usuário. Essas imagens capturadas são enviadas para o servidor, no qual a localização da pessoa é encontrada e são calculados os passos até um destino desejado. A localização da pessoa é encontrada com o uso da funcionalidade de Rastreamento do ORB-SLAM e é representada por meio da grade de ocupação que tem por função descrever as características do ambiente em questão. A partir da posição da pessoa, um caminho que passa o mais distante possível de todos os obstáculos é calculado com um planejador de rotas BVP.

Palavras-chave: Planejador BVP. Sistema de navegação interno. ORB-SLAM. Pessoas com deficiência visual.

Locomotion assistance system to people with visual impairment

ABSTRACT

People could have difficulties finding themselves in an unknown environment, but this task becomes even more difficult when one cannot rely on visual information. With the aim of helping people with visual impairments to move around unknown indoor environments, this work presents a locomotion system as an assistive technology application. The proposed system uses a mobile phone as an interface to allow the user to set the desired destination. In addition, the device has the role of capturing images around the user. These captured images are sent to the server where the person's location is found and the steps are calculated to a desired destination. The person's location is found using the ORB-SLAM Tracking feature and is represented by an occupation grid, which is intended to describe the characteristics of the environment in question. From the person's position, a route going as far as possible from obstacles is calculated with a BVP path planner.

Keywords: BVP Path Planner, Indoor navigation system, ORB-SLAM, Visually impaired people.

LISTA DE ABREVIATURAS E SIGLAS

BA	Bundle Adjustment
BVP	Boundary Value Problems Path
BRIEF	Binary Robust Independent Elementary Features
CCD	Charge-Coupled Device
DBoW	Bag of Binary Words
EKF	Extended Kalman Filter
FAST	Features from Accelerated Segment Test
GPS	Global Positioning System
LSD-SLAM	Large-Scale Direct Monocular SLAM
LVC	Local View Cells
ORB	Oriented FAST and Rotated BRIEF
PCN	Pose Cells Networks
P_nP	Perspective-n-Point
RANSAC	RANdom SAmples Consensus
RRT	Rapidly exploring Random Tree
SIFT	Scale-Invariant Feature Transform
SLAM	Simultaneous Localization And Mapping
SOR	Successive Over-Relaxation
SURF	Speeded-Up Robust Features

LISTA DE FIGURAS

Figura 2.1	Janela de detecção de pontos de interesse pelo algoritmo FAST.....	17
Figura 2.2	Índices entre uma <i>bag of binary words</i> e uma imagem.	19
Figura 2.3	Esquema geral dos passos de localização de um robô.....	20
Figura 2.4	Representação do funcionamento do ORB-SLAM.	23
Figura 2.5	Representação da tarefa de Rastreamento do ORB-SLAM.....	26
Figura 4.1	Diagrama das partes do sistema.....	37
Figura 4.2	Correção do erro no mapa com o fechamento de <i>loop</i>	39
Figura 5.1	Câmera durante exploração.	45
Figura 5.2	Mapa gerado em um sentido do corredor.	46
Figura 5.3	Os dois sentidos que o corredor pode ser percorrido.....	46
Figura 5.4	Mapas gerados nos dois sentidos de ambiente pelo ORB-SLAM.....	48
Figura 5.5	Planta baixa e mapa de pontos do ambiente com os quatro pontos de referência destacados.	49
Figura 5.6	Nuvem de pontos original, em preto, e transformada, em vermelho.....	50
Figura 5.7	Projeção dos pontos sobre a grade de ocupação.....	51
Figura 5.8	Grade de ocupação.....	52
Figura 5.9	Grade de ocupação com os potenciais calculados pelo BVP.....	53
Figura 5.10	Gradientes na grade de ocupação.	54
Figura 5.11	Diferença entre rotas encontradas com e sem obstáculos.....	54
Figura 5.12	Aplicativo que recebe as instruções do servidor.....	55
Figura 5.13	Utilização do sistema.....	56
Figura 5.14	Quadros capturados com pessoas interferindo na localização.....	58

SUMÁRIO

1 INTRODUÇÃO	10
2 SLAM	12
2.1 Visual SLAM	13
2.2 Método baseado em primitivas geométricas	15
2.3 Representação de Ambientes	18
2.4 Localização	19
2.5 ORB-SLAM	21
2.5.1 Mapa	23
2.5.2 Rastreamento.....	25
2.5.3 Mapeamento Local.....	28
2.5.4 Fechamento de <i>Loop</i>	29
3 PLANEJAMENTO DE ROTAS	31
3.1 Planejador BVP	33
4 SISTEMA PROPOSTO	36
4.1 Criação do mapa	38
4.1.1 Mapeamento.....	38
4.1.2 Nuvem de Pontos e Planta Baixa	40
4.2 Navegação	40
4.2.1 Grade de Ocupação	40
4.2.2 Localização com ORB-SLAM.....	41
4.2.3 Cálculo de rotas.....	42
4.3 Trabalhos Relacionados	43
5 ANÁLISE DO SISTEMA	45
5.1 Geração do mapa	45
5.2 Grade de ocupação	49
5.3 Planejador BVP	53
5.4 Utilização	55
6 CONCLUSÃO	59
REFERÊNCIAS	61
APÊNDICE A — TRABALHO DE GRADUAÇÃO 1	66

1 INTRODUÇÃO

Tarefas como localização e mapeamento de robôs são alvos de pesquisas por décadas. Um robô usa informações capturadas do ambiente para aferir sua posição usando sensores e algoritmos, os quais podem ser de vários tipos. Recentemente, pesquisas tem mostrado a viabilidade do uso de câmeras como sensores em diversas aplicações, como explicam Scaramuzza e Fraundorfer (2011). De acordo com Aulinas et al. (2008), um exemplo desse tipo de aplicação é no uso de Localização e Mapeamento Simultâneos (SLAM da sigla em inglês *Simultaneous Localization And Mapping*).

As técnicas de SLAM com imagens, conhecido como Visual SLAM, têm progredido nos últimos anos e aos poucos estão sendo empregadas em produtos comerciais, segundo Platinsky, Davison e Leutenegger (2017). As aplicações possíveis não se limitam a embarcar a solução de Visual SLAM em robôs, podendo ser empregadas, por exemplo, em tecnologias assistivas não robóticas. De acordo com Lancioni e Singh (2014), tecnologias assistivas são equipamentos, comerciais ou customizados, que ajudam as pessoas à reduzir o impacto de suas incapacidades nas tarefas diárias e à alcançar uma melhor qualidade de vida. Já o governo brasileiro, por meio do Comitê de Ajudas Técnicas, define que “Tecnologia Assistiva diz respeito à pesquisa, fabricação, uso de equipamentos, recursos ou estratégias utilizadas para potencializar as habilidades funcionais das pessoas com deficiência” (BRASIL, 2009, 11).

Auxiliar pessoas com deficiência a se locomoverem em ambientes diversos é um dos potenciais usos de tecnologias assistivas. Pessoas eventualmente precisam se localizar em ambientes desconhecidos e descobrir rotas até os destinos desejados. Para essa tarefa são usadas diversas informações do espaço, como explicam Vermeersch, Heylighen e Leuven (2012), as quais costumam ser principalmente visuais. Lugares públicos e privados tendem a disponibilizar mapas, letreiros e setas informativas que auxiliam na localização de quem circula nesses locais. Por conta da impossibilidade de analisar essas informações, Dischinger (2006) aponta que pessoas com deficiência visual interpretam o ambiente ao seu redor de forma diferente. Assim, elas dependem de outras fontes de informações para encontrarem os caminhos desejados.

Para esses casos, é proveitoso o uso de uma tecnologia assistivas que interprete as informações visuais existentes no ambiente. Nesse sentido, a 3ª Conferência Nacional dos Direitos da Pessoa com Deficiência têm aprovadas na área de acessibilidade propostas para “garantir e implementar equipamentos e serviços com tecnologia assistiva que asse-

gurem a comunicação e a interação social entre as pessoas com deficiência e as demais” (BRASIL, 2012, 52). O advento dos celulares inteligentes e o amadurecimento nas técnicas de Visual SLAM trazem novas possibilidades para o desenvolvimento desses tipos de equipamentos. A câmera e as interfaces disponibilizadas presentes em um celular, assim como os algoritmos de Visual SLAM, podem fazer do próprio aparelho um equipamento para auxílio a pessoas com deficiência visual.

Uma solução de Visual SLAM, proposta por Mur-Artal e Tardós (2014), é o ORB-SLAM. Essa técnica utiliza as imagens monoculares para executar as tarefas de localização e mapeamento. De acordo com Huletski, Kartashov e Krinkin (2016), o ORB-SLAM é uma solução promissora para os casos em que não existem outras fontes de informação para odometria, além de uma câmera. Dessa forma, o sistema do ORB-SLAM pode ser usado em conjunto com um celular, usando as imagens provenientes da câmera do aparelho, para criação de novas soluções, como as de tecnologias assistivas.

O presente trabalho propõe o desenvolvimento de um sistema para planejamento de caminhos com o uso de plataformas de telefones celulares e as funcionalidades fornecidas pelo ORB-SLAM. Tal sistema pode servir como base para o desenvolvimento de tecnologias assistivas capazes de auxiliar pessoas com deficiências visuais a se deslocarem em ambientes previamente cadastrados. Para tanto, as imagens da câmera de um celular são usadas pelo sistema do ORB-SLAM para localização e mapeamento.

Inicialmente, é criado um mapa do ambiente desejado a partir da planta baixa do local e da nuvem de pontos fornecida pelo ORB-SLAM. Esse mapa é posteriormente utilizado para a localização da posição do usuário e o planejamento da rota até um destino desejado. A localização é feita com base nas imagens obtidas pelo celular do usuário que são repassadas ao módulo de localização do ORB-SLAM. O planejamento da rota desejada é realizado levando em consideração os obstáculos detectados pela nuvem de pontos do ORB-SLAM de modo que o usuário passe o mais distante possível deles.

A organização do trabalho está dividida em seis seções. Após a introdução, é apresentada a segunda seção que traz conceitos gerais sobre SLAM, aprofundando-se nos principais campos usados para esse trabalho. Os conceitos de Planejamento de Rotas são mostrados na terceira seção. Na quarta seção o sistema proposto é apresentado, contendo a explicação do que foi desenvolvido. Em seguida, estão as pesquisas relacionadas, destacando suas vantagens e limitações. Os resultados alcançados são detalhados na quinta seção. Por fim, na última seção são apresentadas as conclusões obtidas com o trabalho.

2 SLAM

Para algumas tarefas, robôs devem ser capazes de estimar sua posição em relação a um referencial, ao mesmo tempo que constroem um mapa do ambiente, para poderem se locomover de forma eficaz, como explica Shridhar e Kai-Yuan (2015). Em robótica, essa tarefa é conhecida como Localização e Mapeamento Simultâneos (SLAM da sigla em inglês). Segundo Huletski, Kartashov e Krinkin (2016), SLAM é um problema difícil de ser resolvido devido ao fato do robô precisar se localizar usando o mapa que está sendo construído e atualizado ao mesmo tempo em que calcula a sua posição. Já Aulinas et al. (2008) aponta que o problema se torna ainda mais difícil dada a imprecisão dos sensores usados para coletar as informações do ambiente e, dependendo da aplicação, o alto custo computacional para processar os dados de entrada.

A medida que se desloca em um determinado ambiente, um robô precisa estimar sua posição em relação a um mapa para realizar tarefas e interagir com o mundo ao seu redor. De acordo com Romero et al. (2010), o robô executa a tarefa de localização intercalando ciclos de movimento e percepção. Com base numa estimativa inicial, o robô se movimenta enquanto faz uma predição da sua posição. Ao finalizar esse movimento, informações do ambiente são comparadas com a predição feita anteriormente para corrigir a posição do robô.

Conforme Mur-Artal e Tardós (2014), realocação e fechamento de *loops* (laços, em inglês) são duas tarefas essenciais para aplicações reais de SLAM. A realocação é a tarefa de localizar o robô em um mapa existente quando a sua posição é desconhecida. O robô pode não conhecer a sua localização por ter se perdido, devido a movimentos muito rápidos, por exemplo, ou por estar iniciando uma nova seção com um mapa já criado. Já o fechamento de *loops* consiste em detectar quando um certo lugar já mapeado é revisitado pelo robô com o objetivo de corrigir o erro acumulado durante o percurso.

Simultaneamente ao processo de localização, o mapa é construído pelo robô enquanto ele explora o ambiente. Segundo Romero et al. (2010), o robô utiliza sensores para ler informações ao seu entorno e gera modelos computacionais que o descrevem, os quais são usados para construir um mapa do ambiente. Para ser consistente, o mapa precisa estar de acordo com um referencial fixo no mundo e é necessário calcular a posição e a orientação do robô em relação a esse referencial. Os ambientes descritos por esses mapas podem ser internos (*indoor*) ou externos (*outdoor*).

De acordo com Thrun (2002), os dois principais modelos de representação de ma-

pas são o métrico e o topológico. No mapa métrico, são armazenadas as propriedades geométricas do ambiente. Já no mapa topológico os locais são descritos na forma de um grafo. Independente do modelo usado, Romero et al. (2010) explica que uma questão importante na construção dos mapas são as imprecisões sensoriais. Os sensores apresentam erros devido a influência de diversas fontes de ruídos que se acumulam ao longo do tempo, aumentando a sua imprecisão. Esses erros devem ser analisados por meio de um modelos com o uso de algum modelo computacional durante o processo de mapeamento para que o mapa criado seja consistente com o ambiente.

Diversos métodos de filtragem foram desenvolvidos para lidar com a falta de precisão dos sensores e são aplicados nos algoritmos de SLAM. De acordo com Huletski, Kartashov e Krinkin (2016), métodos baseados em inferência Bayesiana são comumente usados nesses casos. Esses métodos permitem o desenvolvimento do SLAM com diferentes tipos de sensores de entrada. Um exemplo desses sensores são as câmeras, com as quais a tarefa de SLAM recebe imagens do ambiente.

2.1 Visual SLAM

O Visual SLAM, localização e mapeamento simultâneos com imagens, tem se popularizado por conta do baixo custo e riqueza de informações fornecidas pelas câmeras, defendem Mur-Artal e Tardós (2016). Nele, o posicionamento e mapeamento mantidos pelo robô dependem da pose da câmera. Segundo Romero et al. (2010), os métodos para estimar a pose da câmera e a estrutura da cena estão divididos em dois grupos principais. No primeiro grupo, os parâmetros são calculados de forma causal, utilizando apenas as informações disponíveis até o instante corrente. Já no segundo grupo, várias imagens são processadas conjunto. Em outras palavras, é feito um ajuste de parâmetros por blocos de imagens (do inglês *Bundle Adjustment* (BA)). Nesse caso, as imagens capturadas após um dado instante influenciam a estimativa da pose e da cena relativas às imagens capturadas anteriormente aquele instante, ou seja, os parâmetros são calculados de forma não causal.

No caso específico do Visual SLAM ser executado com apenas uma câmera como fonte de entrada, ele é denominado SLAM Monocular. Para esse tipo de SLAM, Mur-Artal e Tardós (2014) apontam desafios extras para a tarefa de localização e mapeamento, pois não é possível determinar a profundidade de objetos com uma única imagem. Para estabelecer a escala do mapa e calcular a trajetória realizada, Choi, Park e Yu (2013) explicam que é necessária uma sequência de imagens com múltiplos pontos de vista de uma

mesma cena. Segundo Engel, Sturm e Cremers (2013), os dois principais métodos para analisar imagens e obter as informações necessárias são o método baseado em primitivas geométricas (*Feature-based Method* em inglês) e o método direto (*Direct Method* em inglês).

No método direto, o sistema trabalha diretamente nas imagens recebidas da câmera tanto para a tarefa de localização como para a tarefa de mapeamento. O ambiente é modelado como uma superfície densa, onde os novos quadros (*frames* em inglês) são localizados usando toda a imagem. Essa abordagem elimina a necessidade de extração de apenas algumas características específicas da imagem, podendo assim explorar toda a informação contida nela. Como desvantagem, Engel, Sturm e Cremers (2013) apontam que esse modelo requer uma grande capacidade de processamento para analisar todas as imagens em tempo real.

Já no método baseado em primitivas geométricas, pontos de interesse são extraídos da imagem e associados às suas posições na cena observada. Em seguida são criados descritores para cada ponto de interesse que servem para identificá-los mais rapidamente e necessitam de menos memória para serem armazenados, como explica Calonder et al. (2010). Os descritores extraídos podem ser usados em diferentes etapas das tarefas de localização e mapeamento, como no cálculo da pose da câmera e para armazenar os quadros da imagem.

Recentemente, diversas soluções para SLAM Monocular tem sido apresentadas. Dentre elas, Shridhar e Kai-Yuan (2015) destacam duas consideradas estado atual da arte, o LSD-SLAM (da sigla em inglês para *Large-Scale Direct monocular SLAM*), apresentado por Engel, Schöps e Cremers (2014), e o ORB-SLAM, apresentado por Mur-Artal e Tardós (2014). Enquanto o LSD-SLAM utiliza um modelo monocular denso, o ORB-SLAM utiliza um modelo baseado em características.

No LSD-SLAM, cada quadro capturado pela câmera é analisado e comparado com o quadro anterior para criar um mapa topológico do ambiente. De acordo com Engel, Schöps e Cremers (2014), cada vértice do grafo criado é tido como um quadro chave (*keyframe* em inglês) representando a imagem capturada e seu respectivo mapa de densidade, sendo assim baseado no modelo monocular direto. Os vértices são conectados entre si por arestas que contém informações sobre o posicionamento das imagens, porém sem conter escalas. Segundo Shridhar e Kai-Yuan (2015), o LSD-SLAM possui a vantagem de poder reconstruir de forma mais completa uma cena que contém texturas que não seriam detectadas por métodos baseados em características, porém com um custo computacional

maior.

No ORB-SLAM é criado um mapa do ambiente explorado que pode ser reutilizado em seções distintas, conforme explica Mur-Artal, Montiel e Tardós (2015). O ORB-SLAM executa em tempo real três tarefas distintas: Rastreamento (*Tracking* em inglês), Mapeamento Local (*local Mapping* em inglês) e Fechamento de *Loop* (*loop Closer* em inglês). O Rastreamento recebe o quadro vindo da câmera, extrai os pontos de interesse, otimiza-os e decide se deve criar um novo quadro chave. O Mapeamento Local processa os novos quadros chaves e executa uma triangulação para determinar sua posição no mapa. Por fim, o Fechamento de *Loop* procura por *loops* e, caso algum seja encontrado, o mapa é atualizado e sincronizado com o Mapeamento Local. No ORB-SLAM, os quadros chaves são representados como um conjunto de descritores dos pontos de interesse da imagem, sendo assim baseado em primitivas geométricas, como aponta Huletski, Kartashov e Krinkin (2016).

2.2 Método baseado em primitivas geométricas

No método baseado em primitivas geométricas, pontos de interesse da imagem observada são extraídos e processados para se obter características associadas à cena. Os pontos de interesse são primitivas geométricas, podendo ser pontos, retas e contornos, como explica Romero et al. (2010). Podem ser armazenados de forma mais eficiente, além de serem usados para o cálculo da posição da câmera e nas tarefas de localização e mapeamento.

De acordo com Calonder et al. (2010), a extração de pontos de interesse da imagem reduz a complexidade do problema e permite que ele seja resolvido em tempo real. Por outro lado, essa abordagem pode perder informações importantes da imagem, pois tipicamente apenas cantos e linhas são utilizados para extrair as primitivas geométricas, como explica Engel, Sturm e Cremers (2013). Outro problema é que cada primitiva extraída da imagem precisa ser comparada com todas as primitivas geométricas de todas as outras imagens, o que pode ter um custo computacional elevado dependendo dos algoritmos usados e da quantidade de imagens.

A identificação das primitivas geométricas mais relevantes de uma imagem ocorre em dois passos. Primeiro são identificados pontos de interesse na imagem e em seguida são extraídas informações sobre cada um dos pontos de interesse e a relação entre eles. Diversos algoritmos tem sido desenvolvidos para realizar essas tarefas, entre eles, Chien

et al. (2016) destacam o SIFT (LOWE, 1999), o SURF (BAY; TUYTELAARS; GOOL, 2006) e o ORB (RUBLEE et al., 2011).

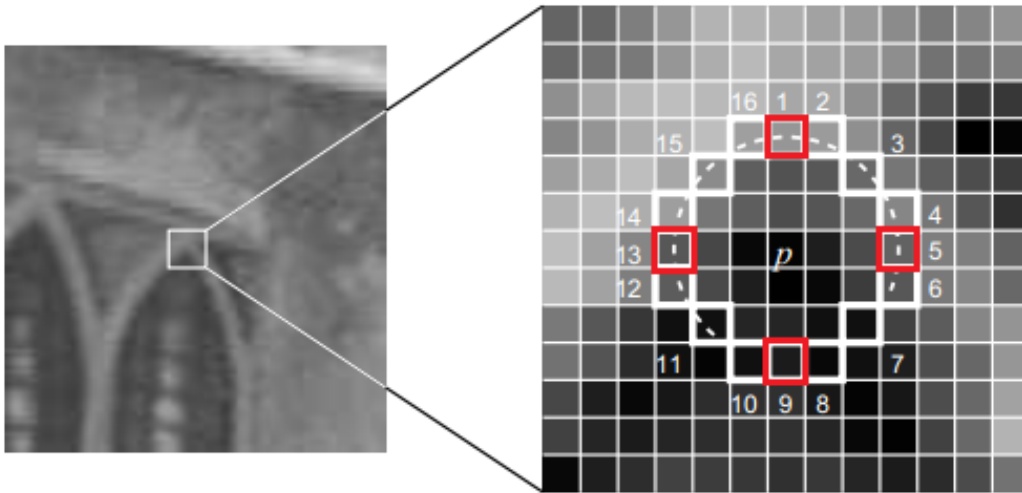
Um dos primeiros algoritmos propostos para a extração de primitivas geométricas de uma imagem foi o SIFT (da sigla em inglês *Scale-Invariant Feature Transform*). Para criar uma representação da imagem, o SIFT constrói uma pirâmide de múltiplas resoluções aplicando operadores de diferença de Gaussianas para localizar extremos locais. Após encontrar as primitivas geométricas na imagem, o algoritmo calcula os gradientes em janelas centradas em cada primitiva geométrica e então as agrupa em sub-regiões para quantizar seus histogramas. No fim, é obtido um vetor de descrição SIFT de 128 elementos. Segundo Chien et al. (2016), apesar de ser robusto e apresentar bons resultados, o SIFT possui um alto custo computacional, o que torna difícil sua aplicação em sistemas de tempo real.

O algoritmo SURF (da sigla em inglês *Speeded-Up Robust Features*) foi apresentado como uma alternativa ao SIFT. Uma das principais diferenças apresentadas é o uso de filtros derivativos de segunda ordem no lugar de operadores de diferença de Gaussianas. Os filtros usados podem ser implementados de forma mais eficiente sobre as imagens, o que deixa o algoritmo mais rápido que seu antecessor. De acordo com Calonder et al. (2010), o SURF requer muitos bytes para cada descritor, o que é um problema quando se tem milhões de descritores para serem armazenados.

Em 2011, Rublee et al. (2011) propuseram o ORB (sigla para *Oriented FAST and Rotated BRIEF* em inglês), como uma alternativa mais eficiente aos algoritmos SIFT e SURF. Ele combina as técnicas de dois algoritmos, o FAST (sigla para *Features from Accelerated Segment Test* em inglês) para localizar os pontos de interesse da imagem e o BRIEF (sigla para *Binary Robust Independent Elementary Feature* em inglês) para descrever e armazená-los. Segundo Rublee et al. (2011), a combinação desses algoritmos faz o ORB ser capaz de alcançar um resultado similar ao SIFT e superior ao SURF com um tempo de processamento muito menor que os dois. De acordo com Chien et al. (2016), essas primitivas geométricas fazem do ORB o mais indicado para aplicações de tempo real ou que tenham uma capacidade de processamento limitada.

Para detectar os pontos de interesse, Rosten e Drummond (2006) explicam que o algoritmo do FAST compara a intensidade de um *pixel* central com quatro *pixels* ao redor de uma janela circular, como ilustrado na Figura 2.1. De acordo com Galvez-López e Tardos (2012), o FAST é rápido o suficiente para ser usado em sistemas de tempo real, porém ele apresenta a limitação de não calcular a orientação dos pontos detectados.

Figura 2.1: Janela de detecção de pontos de interesse pelo algoritmo FAST.



Fonte: Adaptado de Rosten e Drummond (2006)

O Descritor BRIEF é uma sequência de bits que descrevem uma região de uma imagem por meio de um conjunto de testes binários da intensidade dos *pixels*, explicam Calonder et al. (2010). Considerando uma região da imagem \mathbf{p} , um teste binário τ é definido por:

$$\tau(p; x; y) := \begin{cases} 1 & p(x) < p(y) \\ 0 & p(x) \geq p(y), \end{cases} \quad (2.1)$$

onde $\mathbf{p}(\mathbf{x})$ é a intensidade \mathbf{p} em um pixel \mathbf{x} de uma imagem suavizada. O tamanho da região de imagem e da janela de testes afetam a performance e o resultado do algoritmos, como apontam Calonder et al. (2010).

Nos descritores ORB, Rublee et al. (2011) explicam que são usadas regiões de 31 x 31 *pixels* e janelas de testes de 5 x 5 *pixels* com descritores de 256 bits. Para calcular a orientação dos pontos FAST detectados, o algoritmo ORB utiliza uma escala de pirâmides da imagem, como mostrado em Klein e Murray (2008).

Cada imagem é descrita pelos seus descritores ORB. O conjunto dos descritores de cada imagem pode ser usadas para comparar semelhanças entre imagens e para organizá-las. Com uma correta organização, os descritores podem ser usados para a representação do ambiente explorado, como explicado na seção seguinte.

2.3 Representação de Ambientes

Representações de ambientes são realizadas por meio de mapas, os quais são utilizados em diversas atividades humanas. Mapas podem ser dos mais diversos tipos, dependendo da aplicação a que se destinam. Segundo Thrun (2002), uma forma de classificação de mapas é pelo nível de abstração que eles apresentam, podendo ser métricos ou topológicos. Os primeiros são construídos com um baixo nível de abstração enquanto os segundos com um nível de abstração mais elevado.

Os mapas métricos possuem elementos que representam características de um local com uma posição geográfica definida. Esse tipo de mapa pode ser usado em aplicações que necessitem de informações detalhadas sobre o ambiente, como a forma e o tamanho dos objetos, além de indicações dos espaços livres, como corredores e portas. Um exemplo de aplicação é a representação de plantas baixas.

Em robótica móvel, Choset et al. (2004) explicam que um tipo de mapa métrico muito utilizado é a grade de ocupação, que consiste em células uniformes que dividem o espaço. As células informam se a região está ocupada ou livre. De posse de um mapa do tipo grade de ocupação é possível definir os caminhos que podem ser percorridos. O espaço representado por esse tipo de mapa pode ser tanto 2D como 3D.

Os mapas topológicos representam relações entre os elementos de um ambiente na forma de um grafo. Os vértices do grafo indicam locais do espaço ou mesmo regiões completas com características em comum. As arestas representam relações entre esses locais. De acordo com Siegwart, Nourbakhsh e Scaramuzza (2011), uma aresta indica uma adjacência entre dois vértices e pode conter diferentes tipos de informações, como a existência de um caminho e a distância entre os dois locais representados pelo vértice.

Uma das principais vantagens dos mapas topológicos é que eles requerem menos memória de armazenamento quando comparados com os mapas métricos. Um mapa topológico armazena informações de forma mais abstrata e, por isso, mais eficaz e sem dados desnecessários. Por outro lado, Romero et al. (2010) apontam que o processo de construção de um mapa topológico é mais complexo e, por isso, tende a demandar um poder computacional maior. Isso acontece porque é necessário abstrair as informações do ambiente a fim de obter apenas as informações de interesse.

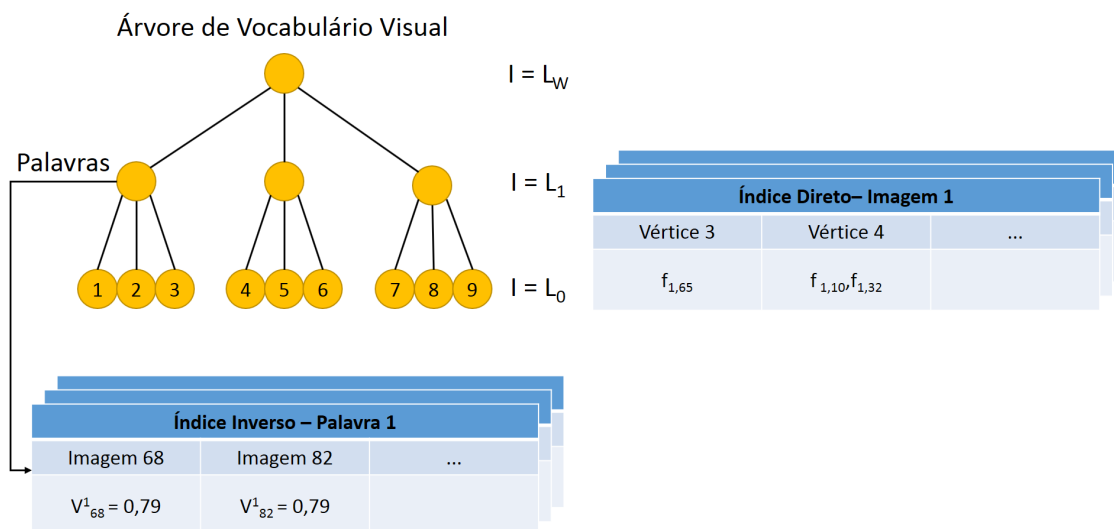
Segundo Mur-Artal e Tardós (2014), uma característica importante da representação de mapas é a capacidade de armazenar cada cenário de forma eficiente e conseguir recuperá-los quando necessário. Uma solução proposta por Galvez-López e Tardos (2012)

busca resolver esse problema por meio de um pacote de palavras binárias (*Bag of Binary Words* em inglês), chamado de DBoW2.

As palavras são uma discretização do espaço do descritor das imagens, que são conhecidas como o vocabulário visual. O vocabulário é *offline*, ou seja, é gerado antes de ser usado, com os descritores extraídos de um grande conjunto de imagens. Se as imagens forem gerais o suficiente, o mesmo vocabulário pode ser usado para diferentes ambientes, obtendo um bom desempenho.

Com o DBoW2, o sistema cria gradualmente um banco de dados que contém um índice invertido, o qual armazena para cada palavra visual no vocabulário a imagem em que foi detectado. De acordo com Galvez-López e Tardos (2012), com essa técnica é possível armazenar as posições percorridas pelo robô de forma eficiente, assim como verificar se uma determinada posição já foi percorrida anteriormente. A Figura 2.2 mostra como é realizado esse índice. O conhecimento das posições já percorridas pelo robô é importante para a tarefa de localização, descrita em seguida.

Figura 2.2: Índices entre uma *bag of binary words* e uma imagem.



Fonte: Adaptado de Galvez-López e Tardos (2012)

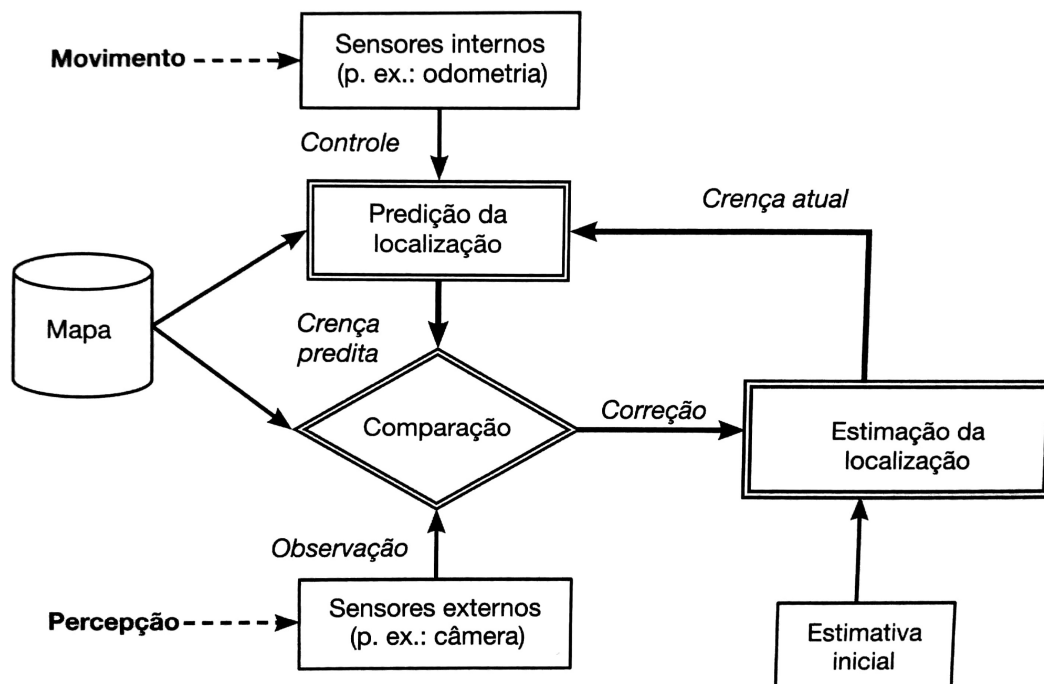
2.4 Localização

Para que um robô possa navegar em um ambiente e realizar tarefas é imprescindível que ele faça uma correta representação de sua posição em um mapa. De acordo com Romero et al. (2010), para representar a crença de sua localização, um robô pode fazer uso de mapas contínuos ou discretos, assim como pode usar uma hipótese absoluta de

localização ou múltiplas hipóteses de localização, entre as quais o robô deve lidar com uma incerteza em relação a hipótese correta.

Como demonstrado na Figura 2.3, durante a tarefa de localização o robô se movimenta, percebe o espaço ao seu redor e se localiza. Inicialmente é feita uma predição da posição com base no movimento e as informações da odometria. Então o robô observa o ambiente, por meio da interpretação das informações fornecidas pelos sensores, e determina a sua localização com base na percepção realizada.

Figura 2.3: Esquema geral dos passos de localização de um robô



Fonte: Adaptado de Romero et al. (2010)

Sensores são a base para a tarefa de percepção e, por isso, sua imprecisão e incompletude afetam diretamente a tarefa de localização. Como exemplo, Siegwart, Nourbakhsh e Scaramuzza (2011) mostram que sistemas de navegação para ambientes internos baseados em câmeras CCD podem sofrer interferência da iluminação do local e não reconhecer um mesmo espaço com diferentes níveis de iluminação.

Devido às incertezas e ruídos associados aos sensores, Romero et al. (2010) explicam que frequentemente torna-se impossível determinar uma única hipótese de localização. Por consequência, ao invés de representar a localização como uma pose (x,y) ou uma pose (x,y,θ) , no caso de mapas contínuos, ou uma única célula, no caso de mapas discretos, pode-se manter múltiplas hipóteses de localização.

Outra forma de representar uma localização é por meio de uma estimativa proba-

bilística da posição do robô. Segundo Choset et al. (2004), nesse caso, ao invés de usar hipóteses sobre a posição do robô, calcula-se uma distribuição de probabilidade de todas as possíveis posições. De acordo com Romero et al. (2010), dois métodos principais são utilizados para localização probabilística de robôs baseada em mapas, a Localização Markoviana (FOX, 2002) e o Filtro de Kalman (KALMAN, 1960).

Na Localização Markoviana, o processo de localização pode começar por qualquer posição desconhecida do mapa e pode ser executada mesmo em situações ambíguas. Para realizar essa tarefa, são rastreadas múltiplas posições possíveis, o que requer a utilização de um mapa discreto, como explica Siegwart, Nourbakhsh e Scaramuzza (2011). Assim, é necessário mais memória e um maior poder de processamento, por vezes limitando a precisão e o tamanho do mapa por falta de recursos.

Já a localização por Filtro de Kalman estima o posicionamento do robô à partir de uma posição inicial conhecida e precisa. Logo, esse tipo de localização pode usar tanto mapas discretos como contínuos para representar o ambiente. Entretanto, Siegwart, Nourbakhsh e Scaramuzza (2011) explicam que no Filtro de Kalman a incerteza da posição do robô tende a aumentar com o tempo e, assim, pode falhar em capturar a posição real do robô.

A tarefa de localização sempre se inicia a partir da análise das informações provenientes dos sensores. O sistema precisa extrair os dados necessários vindos das entradas para posteriormente poder analisá-los. Em sistemas que utilizam uma câmera como entrada, as informações provêm das imagens capturadas, como no caso do ORB-SLAM.

2.5 ORB-SLAM

Recentemente, foi desenvolvido por Mur-Artal e Tardós (2014) um novo método para SLAM Monocular, chamado de ORB-SLAM. Esse método utiliza descritores do tipo ORB, como o próprio nome sugere, para representar os pontos de interesse da imagem e um pacote de palavras binárias DBoW2 para armazená-los em um grafo de covisibilidade e poder recuperá-los rapidamente.

De acordo com Shridhar e Kai-Yuan (2015), o ORB-SLAM faz uso de algoritmos modernos para possibilitar o emprego de SLAM em novas aplicações. Segundo Mur-Artal, Montiel e Tardós (2015), as principais contribuições do ORB-SLAM são:

- utilização dos mesmos descritores ORB para todas as tarefas do sistema, o que o

torna mais eficiente, simples e confiável;

- possibilidade de uso do sistema em aplicações de tempo real em grandes ambientes graças ao uso de um grafo de covisibilidade;
- fechamento de *loops* em tempo real baseado em um grafo da pose da câmera;
- relocalização da pose da câmera em tempo real com pouca interferência de variações de iluminação e do ponto de vista da cena;
- um novo método de inicialização que permite a criação de um mapa inicial de cenas planares e não planares;
- um método de gerenciamento de quadros chaves e pontos de interesse que é generoso ao criar novos itens ao mesmo tempo que possui uma política rigorosa para removê-los e evitar redundâncias.

Tais melhorias permitem o ORB-SLAM operar plenamente em aplicações de tempo real, o que normalmente é uma limitação dos sistemas convencionais de SLAM devido ao alto custo computacional, como explica Huletski, Kartashov e Krinkin (2016). Outra característica dos sistemas convencionais de SLAM é que eles são pensados para se localizarem e criarem mapas do ambiente em uma única seção, porém, uma vez terminada a seção, os mapas criados não podem ser usados em outras seções. Com o ORB-SLAM, por outro lado, Mur-Artal e Tardós (2016) mostram que um mapa pode ser reutilizado em diferentes momentos, mesmo com câmeras diferentes da usada para a criação do mapa e com imagens de pontos de vistas diferentes.

Dessa forma, o ORB-SLAM reúne as características mínimas necessárias para ser usado nesse trabalho. O grafo de covisibilidade e o método de gerenciamento de quadros permite que grandes ambientes sejam mapeados com um crescimento sustentável do mapa gerado. A possibilidade de salvar o mapa gerado pelo ORB-SLAM para ser usado em outras seções é essencial, visto que a localização dos usuários no sistema proposto é realizada em um ambiente mapeado anteriormente.

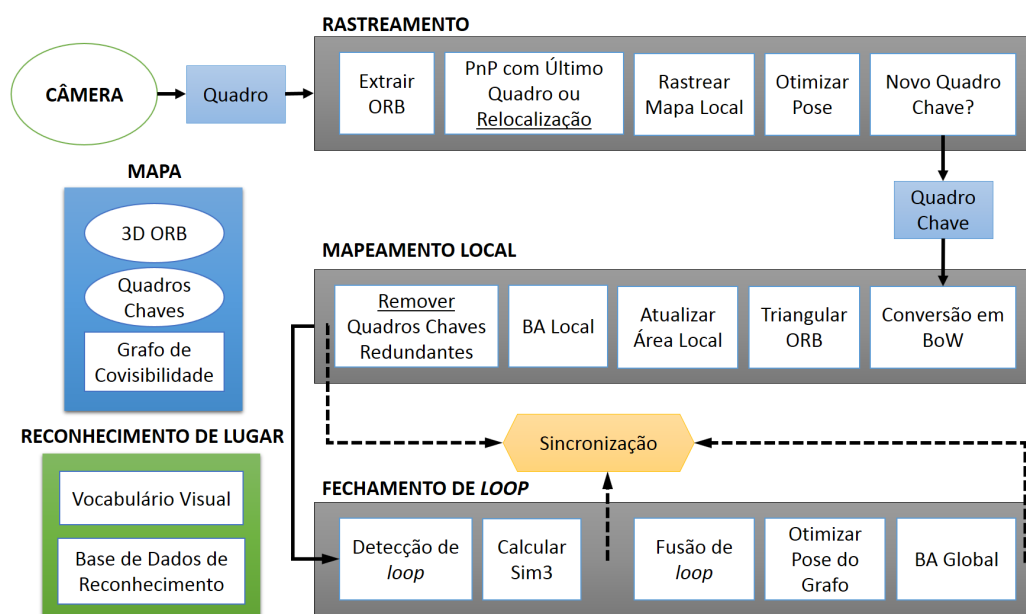
Segundo Huletski, Kartashov e Krinkin (2016), o ORB-SLAM não possui nenhum tipo de odometria para determinar a escala usada no mapa. Dessa forma, a escala utilizada em cada mapa é totalmente indeterminada e só pode ser usada como referência dentro do mesmo mapa. Essa limitação torna difícil a associação de um mapa criado pelo ORB-SLAM com pontos específicos do ambiente real.

O processamento realizado pelo ORB-SLAM é dividido em três tarefas independentes, as quais são:

- Rastreamento: localiza a câmera a cada quadro recebido e decide se um novo quadro chave deve ser inserido;
- Mapeamento Local: processa os novos quadros chaves recebidos e realiza um BA usando apenas um subgrafo com os quadros chaves próximos ao novo quadro chave;
- Fechamento de *Loop*: procura por fechamento de *loops* quando novos quadros chaves são adicionados e realiza o ajuste de todo o grafo.

A Figura 2.4 mostra passo a passo as três tarefas, e as suas subtarefas, executadas pelo ORB-SLAM. Em seguida, cada uma delas é explicada mais detalhadamente, assim como a estrutura de mapas utilizada.

Figura 2.4: Representação do funcionamento do ORB-SLAM.



Fonte: Adaptado de Mur-Artal e Tardós (2016)

2.5.1 Mapa

De acordo com Mur-Artal e Tardós (2014), o ORB-SLAM utiliza um grafo de covisibilidade para representar o ambiente explorado. As informações são armazenadas em um grafo não direcionado com pesos associados. Cada vértice desse grafo contém um quadro chave e o peso de cada aresta é o número de pontos de interesse em comum entre os dois quadros chaves conectados pela aresta. Para uma aresta existir entre dois quadros chaves, é necessário que o número de pontos de interesse compartilhados por eles seja maior que um valor θ . Quanto menor o valor de θ , mais interconectado é o grafo e maior

é o custo para percorrê-lo.

Com o objetivo de melhorar o desempenho nas tarefas que precisam percorrer o grafo, Mur-Artal, Montiel e Tardós (2015) explicam que o ORB-SLAM também constrói um Grafo Essencial (*Essencial Graph* em inglês). O Grafo Essencial armazena todos os vértices do grafo original, mas um número reduzido de arestas. Ele é constituído de uma árvore de extensão (em inglês, *spanning tree*) mais um conjunto de arestas do grafo de covisibilidade que possuam um peso alto, ou seja, que tenham um peso θ_{min} maior que um valor definido. A árvore de extensão é atualizada pela política de inserção e remoção de quadros chaves, um novo quadro chave é sempre inserido no vértice da árvore que possui o θ de maior valor.

Os quadros chaves são representações das cenas capturadas pela câmera. Cada quadro chave armazena todos os descritores ORB encontrados no quadro chave, a posição da câmera e os parâmetros intrínsecos dela, explicam Mur-Artal e Tardós (2014). Os quadros chaves são criados utilizando uma política generosa, o que permite uma expansão rápida do mapa durante o processo de exploração. Por outro lado, eles não são permanentes, podendo ser removidos se forem redundantes, ou seja, se possuírem muitos pontos de interesse em comum com outros quadros chaves.

Segundo Mur-Artal, Montiel e Tardós (2015), no ORB-SLAM, os pontos do mapa (*map points* em inglês) são características ORB extraídas da imagem e que correspondem a regiões da cena vistas de diferentes pontos de vista. O conjunto com todos os pontos do mapa forma a estrutura da representação em 3D do ambiente explorado. Cada ponto do mapa armazena sua posição 3D, um vetor unitário que representa a média de todas as direções de visualização da região da cena nos diferentes quadros chaves. Além disso, pontos do mapa armazenam os descritores ORB que descrevem a região e as distâncias máximas e mínimas que a região pode ser observada de acordo com o limite de invariância da escala da característica ORB. Um ponto do mapa pode possuir mais de um descritor ORB, pois ele pode ser associado a diversos quadros chaves nos quais foi identificado.

O reconhecimento de lugares para a detecção de fechamento de *loops* e para a realocação utiliza uma base de dados com um pacote de palavras binárias DBoW2, explicam Mur-Artal e Tardós (2014). Os quadros chaves são inseridos na base de dados, na qual são associados às palavras binárias, do dicionário criado previamente, em que são vistas no determinado quadro chave. Assim, um quadro chave pode ser consultado de forma eficiente por meio das palavras binárias que ele contém. Quando uma tarefa consulta a base de dados, todos os quadros chaves que possuem um número mínimo de palavras binárias

correspondentes são retornados. Dessa forma, podem ser consultados os quadros chaves com base em suas similaridades. A base de dados é utilizada nas tarefas de Rastreamento e de Fechamento de *loops*, como explicado a seguir.

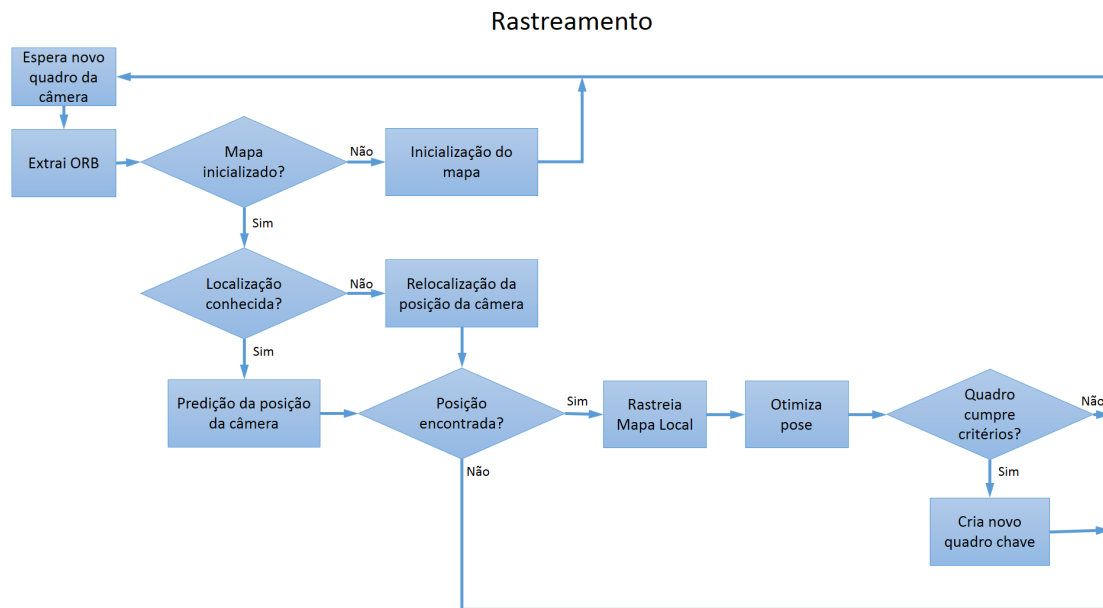
2.5.2 Rastreamento

A tarefa de Rastreamento recebe cada quadro lido da câmera e o processa para ser usado posteriormente. Inicialmente, são extraídos os pontos de interesse FAST do quadro e são calculados os descritores ORB da imagem e a sua orientação. Para garantir que os pontos de interesse estejam espalhados por toda a imagem, Mur-Artal, Montiel e Tardós (2015) explicam que o quadro chave é dividido em uma grade em que cada célula deve ter o mesmo número de pontos de interesse. Apenas se não for possível obter pontos de interesse suficientes em uma determinada célula as outras podem ter um número maior deles.

Após encontrar os descritores ORB da imagem, a tarefa de Rastreamento pode executar a inicialização do mapa, a realocação da câmera ou a predição da posição da câmera a partir do último quadro recebido. A Figura 2.5 mostra as sequências de passos que podem ser realizadas pela tarefa de Rastreamento. Caso o mapa já tenha sido inicializado, é feita a realocação da câmera quando a localização está perdida ou então é feita a estimativa da posição da câmera para o quadro recebido. Se a posição for encontrada com sucesso, são realizados os passos de rastreamento no mapa local, de otimização da pose e, por fim, a decisão de criar um novo quadro chave. Caso o mapa ainda não tenha sido inicializado, o ORB-SLAM utiliza os dois últimos quadros recebidos para tentar criar um novo mapa.

O processo de inicialização do mapa deve calcular a pose entre dois quadros para triangular um conjunto inicial de pontos do mapa. Se a pose for calculada corretamente, ela pode ser usada para representar a posição da câmera no mapa criado. É importante que a inicialização seja executada de forma automática, ou seja, sem a intervenção humana para selecionar dois quadros com diferentes pontos de vista da mesma cena, explicam Mur-Artal, Montiel e Tardós (2015). Ela também deve ser realizada independente do tipo de cena, se planar ou não planar, isto é, se os quadros foram capturados a partir do mesmo plano ou não. Para isso, são calculados dois modelos geométricos em paralelo, uma homografia considerando que a cena seja planar e uma matriz fundamental considerando que a cena não seja planar. Apenas o modelo que melhor representa a cena é selecionado,

Figura 2.5: Representação da tarefa de Rastreamento do ORB-SLAM.



a partir de uma heurística, e usado para tentar calcular a pose relativa entre os quadros. A fim de evitar a criação de um mapa corrompido, a inicialização apenas é executada quando o modelo escolhido apresenta um baixo erro de projeção para cada imagem.

Se a localização foi executada com sucesso no último quadro, o Rastreamento tenta prever a posição da câmera nesse quadro a partir de um modelo velocidade de movimento constante. Para tanto, para cada característica ORB encontrada no quadro anterior é realizada uma busca por essa característica no quadro atual em uma pequena área ao redor da posição do ORB. Segundo Mur-Artal, Montiel e Tardós (2015), movimentos rápidos da câmera e rotações podem violar o modelo de movimento usado e fazer com que poucas características sejam encontradas. Nesse último caso, a busca é refeita considerando uma área de busca maior ao redor da posição das características ORB. Com as correspondências encontradas, a pose da câmera é otimizada com o algoritmo de RANSAC (RANdom SAMple Consensus, em inglês) resolvendo a cada iteração um problema do tipo PnP (sigla para Perspective-n-Point, em inglês).

Por outro lado, se a localização for perdida, o Rastreamento converte o quadro em um pacote de palavras DBoW2 e busca por quadros chaves contendo características similares dentro da base de dados. Em seguida são procuradas as correspondências entre as características ORB de cada quadro chave encontrado com as características do quadro atual. Se um conjunto suficiente de características for encontrado, são realizadas iterações RANSAC para tentar encontrar a pose da câmera. Caso a pose seja encontrada com sucesso, a tarefa de Rastreamento pode continuar com a nova localização.

Com a estimativa da pose da câmera calculada nos passos anteriores, os pontos do mapa do mapa podem ser projetados no quadro para encontrar mais correspondências. De acordo com Mur-Artal, Montiel e Tardós (2015), projetar todos os pontos do mapa contidos no mapa possui um custo computacional muito alto, por isso é usado apenas um conjunto reduzido de pontos do mapa de um mapa local. O mapa local κ é formado pelo conjunto de quadros chaves κ_1 que possuem pontos do mapa em comum com o quadro atual e um conjunto κ_2 com os quadros chaves que são vizinhos de κ_1 no grafo de covisibilidade. O mapa local também possui um quadro chaves de referência κ_{ref} pertencente à κ_1 que compartilha o maior número de pontos do mapa com o quadro atual. Uma vez que o mapa local tenha sido criado, todos os pontos do mapa contidos nele são buscados no quadro atual seguindo os seguintes passos:

1. calcula a projeção do ponto do mapa no quadro. Descarta se ele ficar fora da imagem;
2. calcula o ângulo entre o vetor do ponto de vista no quadro e o vetor normal do ponto do mapa. Descarta se o produto escalar entre os dois for menor que $\cos(45^\circ)$;
3. calcula a distância \mathbf{d} entre o ponto do mapa e o centro da câmera. Descarta se ele ficar fora da região de invariância da escala do ponto do mapa $\mathbf{d} \notin [d_{min}, d_{max}]$;
4. calcula a escala no quadro usando a relação \mathbf{d}/d_{min} ;
5. compara o descritor do ponto do mapa com o descritor ORB do quadro próximo da projeção calculada, considerando a escala estimada;
6. associa o ponto do mapa com a melhor correspondência encontrada.

Por fim, a pose da câmera é otimizada levando em consideração a estimativa da pose e todas as correspondências dos pontos do mapa encontradas nos passos anteriores. Para minimizar o erro de projeção, a posição de todos os pontos do mapa é mantida fixa. O g2o, apresentado por Kümmerle et al. (2011) como um *framework* para otimizar funções que podem ser descritas como grafos, é usado para realizar todas as otimizações.

O último passo da tarefa de Rastreamento é decidir se um novo quadro chave deve ser criado a partir do quadro processado. O objetivo é fazer a inserção dos quadros chaves de forma mais rápida possível, dado que existem mecanismos para remover quadros chaves redundantes. A distância entre os quadros ou o tempo em que foram visualizados não são usados como critérios de decisão para criar um novo quadro chave. Em vez disso, é levado em consideração a similaridade dos quadros com base nos pontos do mapa em comum entre eles. Os seguintes critérios são usados para a inserção de um novo quadro

chave:

1. mais de 20 quadros terem sido processados desde a última realocização;
2. mais de 20 quadros terem sido processados desde o último quadro chave inserido ou a tarefa de Mapeamento Local não está sendo realizada;
3. o quadro atual ter pelo menos 50 pontos do mapa;
4. o quadro atual ter menos de 90% de pontos do mapa em comum com o κ_{ref} .

Apenas no caso de um quadro satisfazer todos os critérios ele é selecionado para a criação de um novo quadro chave. Nesse caso, a tarefa de Rastreamento coloca o quadro em uma fila para ser lido pela tarefa de Mapeamento Local, a qual cria o novo quadro chave.

2.5.3 Mapeamento Local

O Mapeamento Local processa o quadro recebido do Rastreamento para ser um novo quadro chave e atualiza o grafo de covisibilidade, a árvore de extensão e o Grafo Essencial, explicam Mur-Artal e Tardós (2014). Inicialmente é necessário converter o novo quadro chave em um pacote de palavras binárias DBoW2, que serão associadas a cada característica ORB do quadro chave na base de dados e serão usados posteriormente na tarefa de Fechamento de *Loop*.

O mapa de covisibilidade recebe o novo quadro chave e é atualizado com os pontos do mapa que são compartilhados com outros quadros chaves. Da mesma forma, a árvore de extensão é atualizada com a inserção do quadro chave no vértice com o qual ele possui mais pontos do mapa em comum.

Novos pontos do mapa são criados a partir da triangulação dos descritores ORB do novo quadro chave com os quadros chaves interconectados à ele no grafo de covisibilidade. A fim de garantir que os pontos do mapa são robustos, Mur-Artal, Montiel e Tardós (2015) explicam que eles precisam preencher, durante os três primeiros quadros após a sua criação, os seguintes critérios:

- serem encontrados em mais de 25% dos quadros em que são previstos para serem vistos;
- serem encontrados em pelo menos três quadros chaves depois da sua criação.

Após passarem por esses critérios, um ponto do mapa só é removido se ele for visto em

menos de três quadros chaves, o que pode acontecer quando quadros chaves são descartados.

O passo seguinte é realizar um BA local, utilizando o *framework* de g2o, com o novo quadro chave, todos os quadros chaves conectados à ele no grafo de covisibilidade e todos os pontos do mapa observados por esses quadros chaves. De acordo com Mur-Artal, Montiel e Tardós (2015), a complexidade para resolver o BA cresce com o aumento do número de vértices no grafo. Para evitar um crescimento ilimitado de quadros chaves que possam prejudicar o rendimento do sistema, o Mapeamento Local busca e remove quadros chaves redundantes, ou seja, quadros chaves que compartilham mais de 90% de pontos do mapa com pelo menos três outros quadros chaves.

A tarefa de Mapeamento Local não realiza um BA global, pois o tempo de processamento para um número maior de quadros chaves é demasiadamente alto para ser realizado a cada novo quadro chave inserido. Com a expansão do ambiente, o sistema ficaria muito tempo apenas realizando otimizações no mapa. Dessa forma, o sistema executa o BA global apenas na tarefa de Fechamento de *loops* quando um novo *loop* é detectado, conforme explicado na seção seguinte.

2.5.4 Fechamento de *Loop*

A tarefa de Fechamento de *Loop* recebe o quadro chave processado pela tarefa de Mapeamento Local e procura por *loops* que podem ser fechados no grafo de covisibilidade. De acordo com Mur-Artal e Tardós (2014), o Fechamento de *Loop* inicialmente procura por quadros chaves candidatos ao fechamento de *loops*. Em seguida são calculados os descritores ORB correspondentes entre o quadro chave atual e os quadros chaves candidatos. Se um candidato for validado, ele faz a combinação dos pontos do mapa duplicados e, por fim, realiza uma otimização sobre o Grafo Essencial.

A procura por candidatos começa com o cálculo da similaridade entre o pacote de palavras binárias DBoW2 do quadro chave atual com todos seus vizinhos no mapa de covisibilidade que possuem um peso $\theta_{min} = 30$, definem Mur-Artal, Montiel e Tardós (2015). O menor valor encontrado é guardado e usado para descartar todos os quadros chaves com menos similaridade em uma consulta na base de dados. Em seguida, todos os quadros chaves diretamente conectados ao quadro chave atual são descartados. Para garantir que o resultado seja mais robusto, são necessários no mínimo três quadros chaves candidatos que sejam conectados entre si no grafo de covisibilidade para que um candi-

dato seja aceito.

Se ao menos um candidato for encontrado no passo anterior, o Fechamento de *Loop* calcula os descritores ORB correspondentes entre os pontos do mapa do quadro chave atual e os candidatos. Segundo Strasdat, Montiel e Davison (2010), existem sete graus de liberdade no SLAM Monocular, três translações, três rotações e uma variação da escala. Dessa forma, são executadas iterações RANSAC para calcular a transformação do quadro chave atual com os candidatos para encontrar o erro acumulado com o *loop* e para uma validação geométrica do *loop*. Se um candidato tiver um número suficiente de pontos do mapa dentro de um determinado erro tolerável, ele é aceito para o fechamento de *loop*.

Com um quadro chave candidato aceito como um quadro chave de *loop*, a pose do quadro chave atual é corrigida a partir dos descritores ORB correspondentes do quadro chave atual calculado no passo anterior. A correção é propagada para todos os seus vizinhos no grafo de covisibilidade, de forma que os dois lados o *loop* fiquem alinhados. Todos os pontos do mapa visualizados no quadro chave de *loop* são projetados no quadro chave atual e em seus vizinhos para uma busca por combinações. Os pontos do mapa que estiverem dentro do erro tolerável são combinados em um único ponto do mapa e todos os quadros chaves envolvidos na combinação tem suas arestas atualizadas. A criação de novas arestas entre os quadros chaves que compartilham os mesmos pontos do mapa fecham efetivamente o *loop*.

Por fim, Mur-Artal, Montiel e Tardós (2015) explicam que se faz necessário uma otimização global do grafo de covisibilidade para corrigir o erro acumulado com o fechamento de *loop*. Antes da otimização global, é realizada a correção da pose no Grafo Essencial que distribui o erro do fechamento de *loop* ao longo do grafo. Após esse passo, é realizado um BA global, com o uso do *framework* de g2o assim como o BA local, mas sobre todos os quadros chaves do grafo.

O fechamento de *loops* e a correção dos erros acumulados garantem que o mapa seja consistente com o ambiente. Possuir um mapa robusto permite que ele seja usado para localização e para o cálculo de rotas entre as posições do mapa. Rotas podem ser encontradas por meio de algoritmos de planejamento de rotas, conforme explicado a seguir.

3 PLANEJAMENTO DE ROTAS

Na área de robótica, o planejamento de rotas refere-se ao cálculo e à descoberta de passos sequenciais a serem executados para que um robô saia da sua posição inicial e vá até uma posição final, evitando colisões com obstáculos conhecidos. De acordo com Romero et al. (2010), os cálculos devem levar em conta a localização e orientação do robô, as restrições de movimento que ele possui e os obstáculos existentes no trajeto.

O planejamento de rotas já é uma área estabelecida na robótica móvel, como explica Romero et al. (2010), e diversos trabalhos apresentam bons resultados tanto teóricos como práticos. Dessa forma, os algoritmos de planejamento de rotas podem ser vistos em outras áreas de pesquisa, como biologia molecular e animação de objetos artificiais.

As possibilidades de movimentos que um robô pode realizar são definidas pelos graus de liberdade que ele possui. Como explicam Choset et al. (2004), cada grau de liberdade define um tipo de movimento que o robô pode executar, como translação e rotação. Os graus de liberdade de um robô definem o seu espaço de configuração, isto é, o conjunto de todas as configurações que podem ser realizadas pelo robô. Já o espaço de trabalho representa o espaço pelo qual o robô pode se deslocar. Como exemplo, para um espaço de trabalho bidimensional (x,y) , um robô pode ter três graus de liberdade, dois para translação e um para orientação, e seu espaço de configuração tem três dimensões (x,y,θ) .

Segundo Romero et al. (2010), um dos desafios do planejamento de rotas é que o tempo de solução para encontrar as rotas cresce exponencialmente com o aumento dos graus de liberdade do robô. Outros fatores, como o tamanho e o tipo do mapa, também influenciam no desempenho dos algoritmos. Dessa forma, calcular uma solução determinística e completa pode ser inviável em certas aplicações.

De acordo com Siegwart, Nourbakhsh e Scaramuzza (2011), algumas simplificações podem ser realizadas para reduzir a complexidade do cálculo de rotas. Uma simplificação usual consiste em assumir o robô como um ponto dentro do espaço de trabalho, assim, o espaço de configurações é reduzido de três para duas dimensões. Essa modificação, entretanto, pode resultar em colisões do robô com obstáculos no caminho. Uma forma de evitar que isso ocorra, é aumentar a dimensão de cada obstáculo com o tamanho do raio do robô.

A discretização do mapa de representação do ambiente é outra forma de melhorar o desempenho do planejamento de rotas. Mapas discretos podem ser representados

como uma grade de ocupação, explicam Choset et al. (2004), no qual o espaço de trabalho é dividido em células que indicam espaços livres ou ocupados. Duas classes de algoritmos são comumente usadas para o planejamento de rotas em grades de ocupação: planejamento em grafos e planejamento em campos potenciais.

No planejamento de rotas baseado em grafos, os espaços livres do mapa são conectados para formarem um grafo de forma que seja possível realizar uma busca de caminhos entre dois vértices do grafo para o cálculo da rota. Conforme Siegwart, Nourbakhsh e Scaramuzza (2011), existem diversos algoritmos para construção e busca em grafos, dado que essa é uma área largamente estudada dentro da matemática e da robótica.

A construção do grafo consiste em criar um conjunto de vértices e arestas, que permitem o robô ir de um vértice representando um ponto livre até outro ponto vértice, sem colidir com obstáculos, explicam Choset et al. (2004). Entre as abordagens possíveis para a construção do grafo, estão: grafo de visibilidade, diagrama de Voronoi, decomposição exata de células e grafos de Lattice. Cada uma delas apresenta vantagens e desvantagens, as quais podem ser fatores limitadores para o projeto, como proximidade da rota com os obstáculos e capacidade de memória necessária para armazenar o mapa.

Após obter um grafo conectado, a escolha da rota pode ser realizada através do cálculo do melhor caminho entre dois vértices do grafo. Os diversos algoritmos existentes para realizar essa tarefa podem ser divididos em duas classes distintas: algoritmos determinísticos e algoritmos randômicos. Um dos algoritmos determinísticos largamente usados para o planejamento de rotas é o A*. Esse algoritmo é a combinação dos algoritmos de busca melhor-primeiro e de Dijkstra.

Os algoritmos de busca randômicos podem ser usados quando o espaço de busca é muito grande e uma busca exaustiva teria um custo muito alto. Uma classe largamente usada desses algoritmos é o RRT (sigla para *Rapidly Exploring Random Tree* em inglês). No RRT, é criada uma árvore inicial, à qual são adicionados nodos de forma dinâmica, até encontrar um caminho entre o ponto inicial e final. Esses algoritmos não tem garantia de encontrar o melhor caminho, conforme Siegwart, Nourbakhsh e Scaramuzza (2011) apontam, porém é provado que uma rota será eventualmente encontrada caso ela exista.

No planejamento de rotas baseado em campos potenciais são definidos potenciais de repulsão gerados pelos obstáculos um potencial de atração gerado pelo ponto de destino. Assim, Siegwart, Nourbakhsh e Scaramuzza (2011) explicam que, ao percorrer o gradiente gerado, o robô se desloca em direção ao destino ao passo que desvia dos obstáculos. Os campos potenciais se comportam como forças que atuam no robô até que o seu

valor seja nulo, ou seja, existe uma convergência até um mínimo no campo potencial. O ponto de destino deve corresponder ao mínimo do campo de forças, de forma que o robô sempre tende à ir em sua direção. Segundo Romero et al. (2010), as rotas geradas pelo planejamento baseado em campos potenciais são de boa qualidade e costumam passar longe dos obstáculos.

Existem dois métodos comumente utilizados para o cálculo dos campos potenciais, de acordo com Trevisan et al. (2006). No primeiro método, os campos potenciais são calculados como a soma dos campos individuais produzidos por cada obstáculo e pelo ponto de destino. Esse método requer um baixo custo computacional para ser calculado, porém pode resultar em mínimos locais que quebram as rotas calculadas, ou seja, as rotas terminam no mínimo local em vez de no destino desejado. No segundo método, os campos são calculados com a resolução de equações diferenciais parciais, como a Equação de Laplace, nas quais os obstáculos e o destino são tratados como condições de contorno. Dessa forma, o problema dos mínimos locais é evitado, porém com um custo maior para o cálculo das rotas. Um exemplo de método que resolve a Equação de Laplace para encontrar os campos potenciais é o planejador BVP, apresentado na seção seguinte.

3.1 Planejador BVP

Os métodos para planejamento de rotas que resolvem equações diferenciais com condições de contorno são conhecidos como BVP *Path Planning* (Planejamento de rotas com problemas de valor sobre o contorno, da sigla em inglês para *Boundary Value Problems Path Planning*). Um método para exploração de ambientes e planejamento de rotas com BVP foi proposto por Prestes et al. (2001). Nele, são usadas funções harmônicas em uma grade de ocupação para calcular um mapa potencial.

De acordo com Connolly e Grupen (1993), a solução de funções harmônicas com os valores de contornos adequados geram caminhos ótimos e sem mínimos locais. As funções harmônicas são a solução para a equação de Laplace definida por:

$$\nabla^2 p(r) = \sum_i \frac{\partial^2 p(r)}{\partial_i^2} = 0 \quad (3.1)$$

Com as condições de contorno de Dirichlet, lugares com obstáculos são definidos com um potencial alto enquanto destino é definido com um potencial baixo. Dessa maneira, as linhas de força do gradiente gerado pela solução nos espaços livres ficam alinhadas na

direção do destino e repelidas pelos obstáculos.

A equação de Laplace pode ser resolvida por métodos de relaxação, como método de Gauss-Siedel e o SOR (sigla para Sobre-Relaxação Sucessiva, em inglês *Successive Over-Relaxation*), explicam Silveira, Prestes e Nedel (2010).

De acordo com Prestes et al. (2001), os caminhos gerados pelo BVP possuem as mesmas propriedades que as linhas de força de campos potenciais trabalhadas pela física. Assim, podemos afirmar que os caminhos encontrados não apresentam ciclos, sempre conectam os pontos de alto potencial (obstáculos) aos pontos de baixo potencial (destino) e dois caminhos nunca se cruzam.

O algoritmo para planejar rotas com BVP exige que o mapa seja representado como uma grade de ocupação, explicam Trevisan et al. (2006). Os seguintes passos devem ser realizados para a inicialização do mapa:

1. localizar a posição de todos os obstáculos no mapa;
2. definir o valor das células correspondentes à obstáculos com um alto potencial (usualmente 1.0);
3. definir o valor das células correspondentes ao destino com um baixo potencial (usualmente zero);
4. definir o valor das células correspondentes à espaços livres com um valor intermediário entre os valores de alto e baixo potencial.

Com o mapa atualizado, os campos potenciais podem ser encontrados com os métodos de Gauss-Seidel ou de SOR. Seguindo o método escolhido, os valores das células do mapa são atualizados sucessivas vezes até convergirem. O método de Gauss-Seidel utiliza a seguinte regra para atualizar os valores das células:

$$p_{i,j}^{new} = \frac{1}{4}(p_{i-1,j}^{new} + p_{i+1,j}^{old} + p_{i,j-1}^{new} + p_{i,j+1}^{old}), \quad (3.2)$$

onde p é o valor do campo potencial da célula, i e j são os índices da célula atualizada, *new* significa que o valor da célula já foi atualizado na iteração atual e *old* que o valor ainda não foi atualizado.

Já o método de SOR utiliza a seguinte regra:

$$p_{i,j}^{new} = p_{i,j}^{new} + \frac{1}{4}w(p_{i-1,j}^{new} + p_{i+1,j}^{old} + p_{i,j-1}^{new} + p_{i,j+1}^{old} - 4p_{i,j}^{old}), \quad (3.3)$$

com o termo de sobre-relaxação w sendo:

$$w = \frac{2}{(1 + \sqrt{1 - \rho^2})}, \quad (3.4)$$

e:

$$\rho = \frac{1}{2} \left[\cos\left(\frac{\pi}{L_x}\right) + \cos\left(\frac{\pi}{L_y}\right) \right]. \quad (3.5)$$

A cada iteração, a diferença entre $p_{i,j}^{new}$ e $p_{i,j}^{old}$ tende a diminuir, convergindo para um determinado valor. Após executar um número suficiente de iterações para todo o mapa convergir, o gradiente negativo pode ser calculado da posição do robô até o destino. A direção que o robô deve tomar a cada passo é dada por:

$$\theta = \arctan(p_{i-1,j} - p_{i+1,j}, p_{i,j-1} - p_{i,j+1}). \quad (3.6)$$

De acordo com Prestes et al. (2001), o número de iterações necessárias para todas as células do mapa convergirem é menor para o método de SOR em comparação com o método de Gauss-Seidel. Por outro lado, o método de Gauss-Seidel produz rotas mais suaves, e por isso melhores, que o SOR.

O tamanho do mapa utilizado influencia diretamente no tempo para calcular rotas com o algoritmo de BVP. Segundo Silveira, Prestes e Nedel (2010), para um mapa de 513 x 513 células, são necessários 22 segundos para encontrar rotas com BVP, contra apenas 0,07 segundo usando um algoritmo tradicional A*. Entretanto, com BVP já são calculadas as rotas de todos os pontos livres do mapa até o destino, não sendo necessário recalculá-las caso o robô tome um caminho diferente do esperado. Além disso, as rotas encontradas pelo BVP já são suavizadas e tem a garantia de que passam o mais distante possível de obstáculos.

4 SISTEMA PROPOSTO

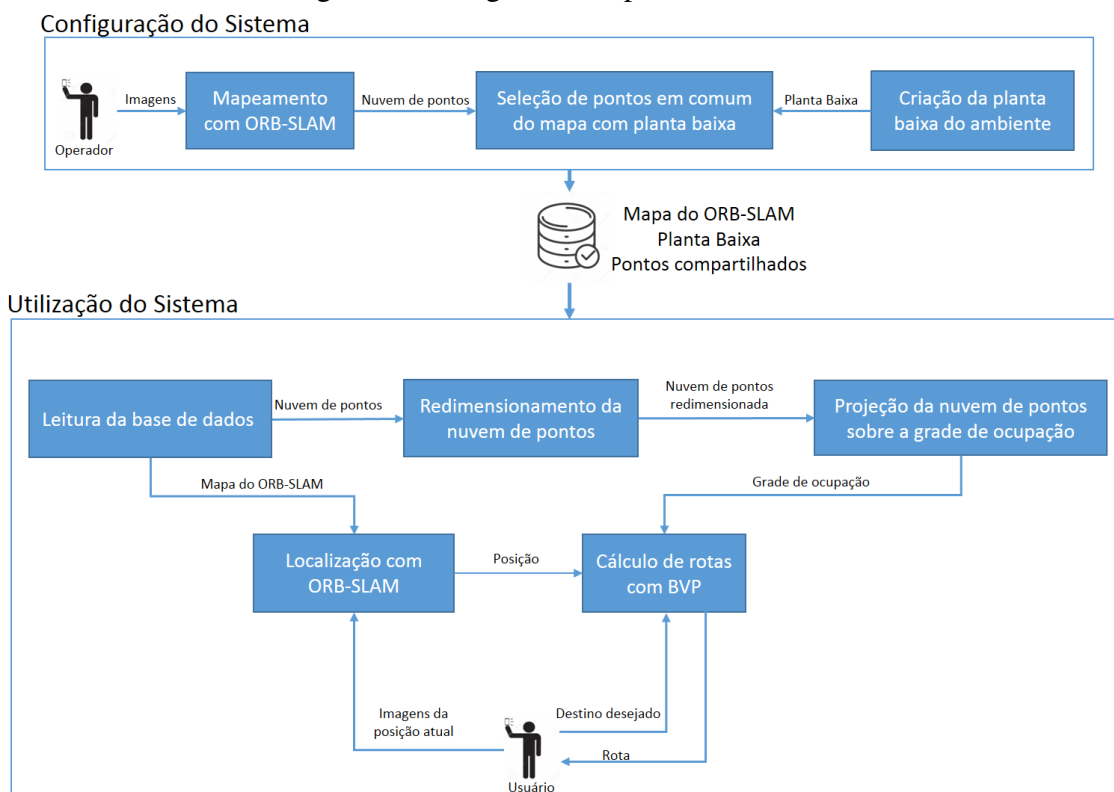
Nesse trabalho, são empregadas técnicas de navegação tipicamente usadas na área de robótica, como abordado nas seções anteriores. Esse sistema serve como tecnologia assistiva para auxiliar pessoas com deficiência visual à locomoverem-se em ambientes internos, aos quais elas não estão familiarizadas. O sistema proposto utiliza imagens provenientes da câmera de celular para localizar a pessoa em um ambiente previamente mapeado. Com a localização conhecida, uma rota é calculada até o destino desejado pela pessoa que está utilizando o sistema.

As tarefas de mapeamento do ambiente e de localização podem ser realizadas por técnicas de SLAM, como as mostradas na seção 2. Para esse trabalho é utilizado um aparelho celular como fonte de informações do ambiente, de modo que a localização deve ser realizada com as funcionalidades presentes em aparelhos celulares. O GPS (sigla para Sistema de Posicionamento Global do inglês *Global Positioning Systems*) é uma funcionalidade de localização presente nos aparelhos celulares atuais, entretanto a tecnologia de GPS tem seu espaço de trabalho limitado, não funcionando bem em áreas fechadas, como explicam Siegwart, Nourbakhsh e Scaramuzza (2011). Dessa forma, a utilização da câmera do celular com o uso de técnicas de SLAM Monocular é uma alternativa para ambientes internos.

De acordo com Mur-Artal, Montiel e Tardós (2015), o ORB-SLAM é umas das mais completas e confiáveis soluções de SLAM Monocular. Assim, ela pode ser empregada para a tarefa de localização utilizando apenas as imagens monoculares de uma câmera de celular. Como descrito na seção 2.5, com o ORB-SLAM é possível utilizar um mapa criado durante uma execução em outras execuções com diferentes condições. Essa característica é importante para permitir que um mapa do ambiente seja criado em um primeiro momento e que esse mapa possa ser posteriormente reutilizado pelo usuário para se localizar no ambiente. A criação do mapa é realizada durante a configuração do sistema, como mostrado na Figura 4.1. Já o mapa criado é utilizado durante a utilização do sistema, para a localização do usuário e o cálculo de rotas entre pontos do ambiente desejados pelo usuário.

O planejamento de rotas deve encontrar um caminho que leve a pessoa do seu ponto inicial até um ponto desejado sem colidir com os obstáculos, conforme abordado na seção 3. Não há garantias de que a rota será percorrida seguindo rigorosamente os passos calculados, pois eles serão executados por pessoas que podem sair do trajeto por

Figura 4.1: Diagrama das partes do sistema.



vários motivos. Dessa forma, o cálculo da rota deve contemplar cada desvio que ocorre durante o trajeto, atualizando o caminho estimado. O planejador de caminhos BVP se encaixa nesses requisitos, como explicado na seção 3.1, pois, uma vez definido os pontos de destino, são geradas rotas a partir de todos os pontos do mapa até o destino, como explicam Prestes et al. (2001). Além disso, todas as rotas geradas são suaves e passam o mais distante possível dos obstáculos. O cálculo de caminhos faz parte da segunda parte do sistema, ou seja, a utilização pelo usuário.

As duas partes do sistema, configuração e utilização, podem ser vistas na Figura 4.1. Na primeira parte, um mapa é criado com o ORB-SLAM por um operador do sistema e em seguida ele é transformado para ficar alinhado com pontos selecionados do ambiente. Na segunda parte, a funcionalidade de localização do ORB-SLAM é utilizada por um usuário para se localizar no ambiente em que ele está. Para isso, o mapa criado pelo ORB-SLAM é projetado sobre uma grade de ocupação bidimensional, na qual são detectados os obstáculos e os caminhos livres. Por fim, o planejador de caminhos BVP é utilizado sobre a grade de ocupação para encontrar uma rota entre a localização da pessoa e o destino desejado. As duas partes são explicadas mais detalhadamente nas seções seguintes.

4.1 Criação do mapa

A primeira parte do sistema consiste na criação de um mapa de todos os espaços do ambiente desejado. Essa tarefa deve ser executada uma única vez para cada ambiente onde se pretende utilizar o sistema. Na segunda parte do sistema, o mapa criado é utilizado na localização do usuário e no planejamento de rotas. Visto que a correta localização do usuário depende da qualidade do mapa criado, é essencial que ele seja condizente com o ambiente. Por isso, o ambiente deve ser mapeado pelo operador do sistema, como explicado a seguir.

4.1.1 Mapeamento

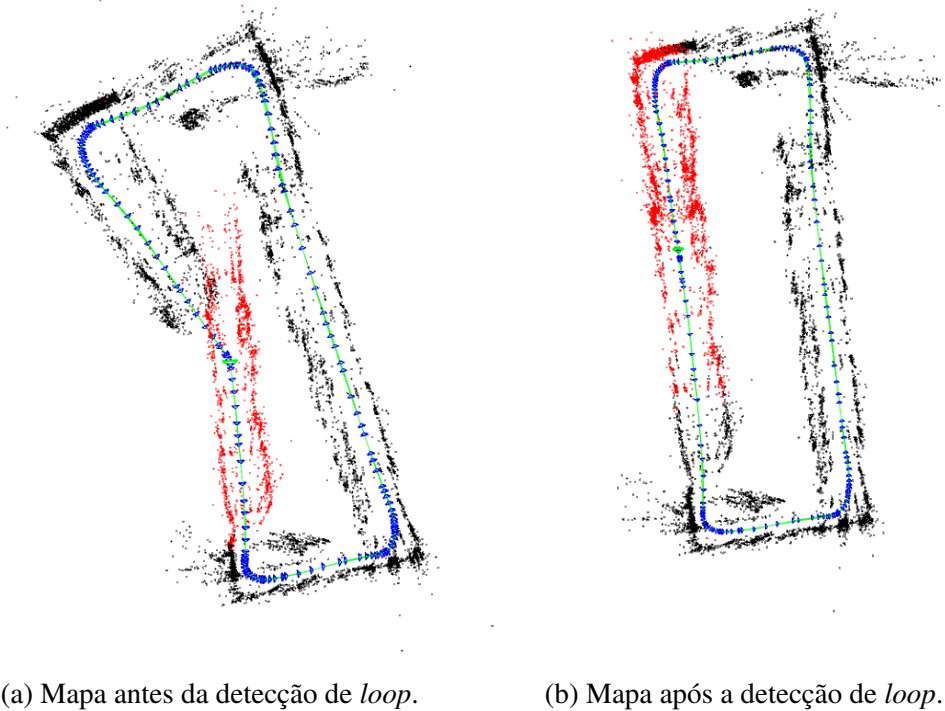
Como primeiro passo de execução do sistema é necessário gerar e salvar o mapa da região que posteriormente será explorada pelo usuário. Para tanto, o ambiente é explorado por um operador de posse de um celular, que envia as imagens capturas para o servidor. No servidor, o ORB-SLAM processa as imagens para realizar a localização e mapeamento do ambiente, conforme descrito na seção 2.5. Para isso, a exploração deve percorrer todos os espaços do ambiente em questão sob diferentes pontos de vista. Além disso, é importante que o operador explore o ambiente de modo a percorrer círculos completos, que possam fechar *loops*.

Segundo Mur-Artal, Montiel e Tardós (2015), o ORB-SLAM utiliza o fechamento de *loop* para corrigir os erros acumulados no mapa. Dessa forma, é importante que *loops* sejam percorridos e detectados pelo ORB-SLAM durante a exploração do ambiente para garantir a consistência do mapa. Na Figura 4.2 pode-se ver a correção do erro realizada em um mapa gerado pelo ORB-SLAM logo após o fechamento de um *loop*.

Para garantir que *loops* não sejam detectados incorretamente, o ORB-SLAM possui uma política rigorosa para aceitar que dois quadros detectados em momentos diferentes sejam do mesmo local e, portanto, fechem um *loop*, conforme foi mencionado na seção 2.5.4. Por outro lado, o não fechamento de *loops* pelo ORB-SLAM em pontos que efetivamente formam um *loop* no ambiente pode causar inconsistências no mapa por causa do erro acumulado e não corrigido.

Para garantir a geração de um mapa consistente pelo ORB-SLAM e que possa ser usado na tarefa de localização pelo sistema proposto, a exploração durante o mapeamento deve ser feita de forma planejada. Assim, a estratégia usada durante a exploração con-

Figura 4.2: Correção do erro no mapa com o fechamento de *loop*.



siste em realizar no mínimo uma volta completa nos dois sentidos em cada espaço do ambiente explorado. Para posterior localização do usuário, apenas as características do próprio ambiente devem ser consideradas na construção do mapa. Dessa forma, a exploração deve ser realizada em um momento que não tenha circulação de pessoas ou outras características no lugar que não possam ser visualizadas posteriormente.

Ao fim do mapeamento do ambiente, o mapa gerado precisa ser salvo para a sua utilização durante a tarefa de localização. O ORB-SLAM gera um mapa em forma de grafo, conforme descrito na seção 2.5. Nele, os vértices são os quadros chaves detectados e para cada quadro chave existe um conjunto de pontos do mapa relacionados. Durante a execução, o ORB-SLAM pode realizar diversas otimizações no grafo gerado, reorganizando-o, assim como pode excluir quadros chaves redundantes e pontos do mapa não utilizados.

O índice dos quadros chaves e dos pontos do mapa pode não ser sequencial ao fim da execução, o que pode causar falhas em execuções posteriores caso ocorra uma tentativa de acesso em uma posição inexistente. Para garantir a consistência do mapa e evitar esse tipo de erro, os índices de todos os quadros chaves e todos os pontos do mapa são reorganizados pelo sistema antes de serem salvos. Posteriormente, o mapa salvo deve ser lido para ser reutilizado durante a utilização do sistema, como explicado a seguir.

4.1.2 Nuvem de Pontos e Planta Baixa

De acordo com Huletski, Kartashov e Krinkin (2016), o ORB-SLAM não possui um sistema de odometria e, por isso, o mapa gerado não possui uma escala determinada. Para superar essa limitação, é necessário redimensionar o mapa de acordo com o ambiente em questão. Segundo Mur-Artal, Montiel e Tardós (2015), a precisão do mapa pode ser menor que 1cm para ambientes internos e apenas alguns metros para ambiente externos uma vez que o mapa seja redimensionado seguindo um referencial real.

No sistema proposto, é criada uma planta baixa do ambiente explorado que serve como referência para a transformação do mapa gerado pelo ORB-SLAM. Para isso, na parte de configuração do sistema são escolhidos pontos distintos da nuvem de pontos, que são associados aos seus respectivos pontos do ambiente. Durante a utilização do sistema, os conjuntos de pontos escolhidos são então usados para criar uma matriz de transformação que escala, rotaciona e translada a nuvem de pontos para a escala da planta baixa. Na planta baixa também são definidos os nomes e as posições de cada local do ambiente que podem ser usados como destino para a navegação do usuário na segunda parte do sistema.

4.2 Navegação

Na segunda parte é realizada a localização do usuário que está utilizando o sistema e é calculada uma rota da posição encontrada até um destino desejado. O mapa gerado na primeira parte da configuração do sistema e descrito na seção 4 é recuperado e usado pelo ORB-SLAM para a tarefa de localização enquanto a grade de ocupação, descrita a seguir, é usada para o cálculo de rotas pelo planejador BVP.

4.2.1 Grade de Ocupação

Uma grade de ocupação é criada para referenciar a localização da pessoa encontrada pelo ORB-SLAM e calcular rotas e os destinos selecionados. A escala utilizada na grade de ocupação é definida pela planta baixa criada na primeira etapa, descrito na seção 4.1.2. Os obstáculos, células ocupadas da grade de ocupação, são definidos tanto pela nuvem de pontos projetada na grade, como pelas informações da planta baixa.

A nuvem de pontos tridimensional é recuperada do mapa do ORB-SLAM e é escalada, rotacionada e transladada conforme a matriz de transformação calculada na primeira parte. Após a transformação, a nuvem de pontos fica com a mesma escala que a grade de ocupação. Os pontos tridimensionais então podem ser projetados no espaço bidimensional da grade de ocupação, levando em consideração a altura do ponto.

Pontos com altura menor que um valor h_{min} são tidos como partes do piso e não são considerados como obstáculos, da mesma forma, pontos com altura maior que h_{max} também não são considerados obstáculos, pois estão acima do espaço de passagem do usuário. Assim, o valor de h_{min} não pode ser muito alto de forma que um obstáculo real no chão não seja considerado. Da mesma forma, o valor de h_{max} não pode ser muito baixo de forma que obstáculos alcançáveis pelo usuário não apareçam.

De acordo com Strasdat, Montiel e Davison (2010), o SLAM Monocular pode apresentar erros no mapa gerado mesmo após as correções com o fechamento de *loops*. Dessa forma, determinados pontos do mapa podem não representar corretamente um obstáculo. Com o intuito de minimizar a influência desses erros na grade de ocupação, foi definido um valor mínimo de pontos que precisam ser projetados sobre determinada célula para que ela seja considerada como ocupada. Já a posição do usuário na grade de ocupação é obtida por meio da funcionalidade de localização do ORB-SLAM.

4.2.2 Localização com ORB-SLAM

Uma das vantagens do ORB-SLAM é a possibilidade de reaproveitar o mapa gerado em uma execução em outros momentos, explicam Mur-Artal e Tardós (2014). Essa característica permite que um mapa do ambiente seja criado uma única vez e depois seja reutilizado para a localização de pessoas, usuárias do sistema, nesse ambiente por diversas vezes. Assim, o mapa criado na primeira parte desse sistema pode ser usado a cada vez que um usuário deseja se localizar no ambiente.

O mapa previamente criado é carregado quando uma pessoa inicia o sistema para se localizar. No mapa estão salvas as informações de todos os quadros chaves e de todos os pontos do mapa que foram gerados pelo ORB-SLAM durante o mapeamento. Essas informações são lidas e usadas para reconstruir o grafo de covisibilidade e a base de dados dos quadros chaves usada pelo ORB-SLAM. Com o mapa carregado, ele pode continuar com as tarefas de localização e mapeamento.

O ORB-SLAM deve ser executado em modo “localização” nessa segunda parte

do sistema. Nesse modo, as tarefas de Mapeamento Local e de Fechamento de *Loop* não são executadas, ou seja, o ORB-SLAM executa apenas a tarefa de Rastreamento, descrita na seção 2.5.2. A execução nesse modo é importante porque, como explicam Mur-Artal, Montiel e Tardós (2015), durante as tarefas de Mapeamento Local e de Fechamento de *Loop* o ORB-SLAM realiza otimizações no grafo de covisibilidade e nos valores dos pontos do mapa, podendo alterar a escala e redimensionar partes do mapa, assim como modificar os quadros chaves e os pontos do mapa. Visto que o mapa do ORB-SLAM é projetado sobre a grade de ocupação, essas alterações fariam ela perder a confiabilidade.

Durante a execução da segunda parte do sistema, a pessoa deve se locomover com o celular de forma que o caminho percorrido seja filmado. Os quadros capturados são enviados para o servidor com o sistema, no qual o ORB-SLAM recebe os quadros e busca fazer a localização da posição da pessoa. Uma vez encontrada a posição, ela é informada para a tarefa do cálculo de rota e a localização segue normalmente. Caso a posição seja perdida durante o processo, a relocalização é realizada novamente. Dessa forma, a localização do usuário pode ser usada para o cálculo de rotas até um caminho desejado.

4.2.3 Cálculo de rotas

O cálculo de rotas é realizado por um planejador BVP sobre a grade de ocupação criada anteriormente. O planejador BVP gera rotas que desviam das células com um alto potencial (com valor 1.0) e tendem até as células com um baixo potencial (com valor zero). Assim, as células da grade de ocupação correspondentes a obstáculos tem o seu valor definido para 1.0 enquanto as células correspondentes ao destino tem o seu valor definido para zero.

A posição referente ao destino é recebida do usuário que está utilizando o sistema por meio do celular. O usuário decide o destino desejado, o sistema verifica na planta baixa a posição desse destino e define como zero o valor das células da grade de ocupação correspondentes a ele. A posição atual do usuário não é necessária durante o cálculo de rotas com BVP, pois, como explicam Silveira, Prestes e Nedel (2010), o BVP calcula simultaneamente rotas de todas as células até o destino, como explicado na seção 3.1.

Tendo definidas as células correspondentes aos obstáculos e as células de destino, o algoritmo do BVP é executado um número mínimo de vezes para a convergência de todas as células. Durante cada iteração, as células são atualizadas com o método de

Gauss-Seidel, pois, de acordo com Prestes et al. (2001), esse método produz rotas mais suaves. As dimensões das células da grade de ocupação influenciam diretamente no desempenho do algoritmo, como mostra Silveira, Prestes e Nedel (2010). Assim, calcular o potencial de todas as células para uma grade de ocupação de maiores dimensões pode ter um custo elevado de tempo. Por outro lado, uma vez que o planejador BVP tenha calculado os campos potenciais para todas as células da grade de ocupação, não é preciso recalculá-los a menos que o destino seja alterado. Para cada posição encontrada do usuário, uma instrução indicando o próximo passo que ele deve tomar é gerada. A instrução é obtida de acordo com o gradiente resultante da célula da grade de ocupação onde o usuário está localizado.

4.3 Trabalhos Relacionados

Apesar de ser uma área vastamente estudada na robótica, novas aplicações para os sistemas de localização e mapeamento tem surgido. Telefones celulares cada vez mais potentes e com sensores mais sofisticados expandem as possibilidades de utilização das tecnologias já existentes. Ao mesmo tempo, o desenvolvimento de tecnologias assistivas tem sido estimulado para auxiliar na acessibilidade de pessoas com deficiência. Assim, as técnicas de localização e mapeamento tem sido empregadas em propostas de aplicações para auxiliar pessoas com deficiência visual, assim como em diversas outras aplicações em telefones celulares.

As soluções que envolvem a área de localização e mapeamento com tecnologias assistivas para pessoas com deficiência visual despontam de variadas formas. As variações podem ser tanto no tipo de equipamentos usados como na maneira de auxiliar as pessoas. Como exemplo, Tapu, Mocanu e Zaharia (2013) propõem uma solução utilizando Monocular SLAM em um celular para auxiliar pessoas com deficiência visual à locomoverem-se. Entretanto, ele não tem como finalidade ajudar na localização da pessoa e sim na detecção de obstáculos que uma pessoa com deficiência visual possa encontrar ao percorrer um trajeto.

Outros autores propõem soluções com a finalidade de auxiliar pessoas à localizarem-se em ambientes diversos, porém não fazem uso apenas de celulares em seus projetos. Entre esses autores, Hesch e Roumeliotis (2007), Xiao et al. (2013), e Poggi e Mattoccia (2016) propõem bengalas inteligentes, utilizando diferentes técnicas. Já Tian, Liu e Tan (2013) apresentam um dispositivo para localização acoplado a um óculos e Serrão et al.

(2012) utilizam uma câmera que deve ser carregada pelo usuário.

Alguns autores propuseram soluções de localização e mapeamento utilizando apenas os recursos fornecidos por um celular, entretanto sem serem direcionados à pessoas com deficiência visual. Assim, essas soluções não realizam o cálculo de rotas e detecção de obstáculos necessárias para navegação de pessoas cegas, a exemplo das alternativas propostas por Dang, Ding e Zhou (2016) e Santos et al. (2016). Já Zhang, Ma e Tan (2016) apresentam uma solução que usa apenas a câmera do celular para obter informações do ambiente, porém apenas em corredores.

Entre os sistemas que utilizam apenas celulares como equipamentos e são focados para o auxílio de pessoas com deficiência visual, mais de uma abordagem pode ser utilizada para resolver o problema. A funcionalidade do GPS é utilizada por Basso, Frigo e Giorgi (2015) e por Endo et al. (2017), porém com a limitação do GPS para ambientes internos, como descrito na seção 4. Por isso, Apostolopoulos et al. (2012) propõem um sistema utilizando apenas a câmera de um celular como sensor. Nesse sistema o usuário deve dar um retorno para confirmar a sua posição em pontos de referências pré-estabelecidos. De acordo com Brady et al. (2015), esse tipo de verificação nem sempre é trivial, já que pessoas com deficiência visual tem seu próprio meio de compreender o ambiente ao seu redor.

5 ANÁLISE DO SISTEMA

Nessa seção são mostrados os resultados alcançados com o sistema proposto e a sua utilização. Primeiramente é mostrada a geração do mapa do ambiente com o ORB-SLAM e a transformação da nuvem de pontos obtida para ser utilizada com a grade de ocupação do mesmo ambiente. Em seguida, é explicada a obtenção das rotas geradas pelo planejador BVP e, por fim, são mostrados os exemplos de uso do sistema.

5.1 Geração do mapa

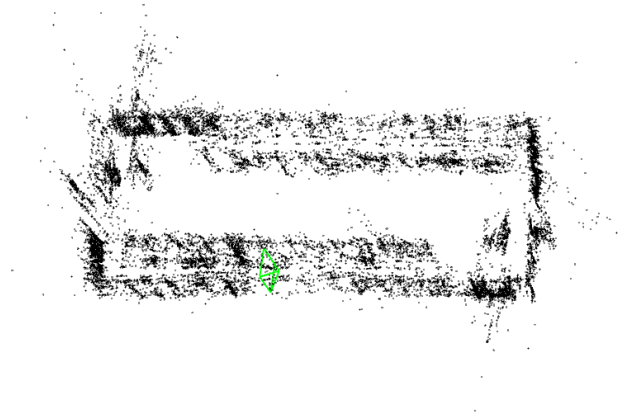
Como explicado na seção 4.1.1, a geração de um mapa consistente por meio do ORB-SLAM é essencial para o funcionamento do sistema. Durante a exploração do ambiente, o ORB-SLAM deve detectar os *loops* e corrigir os erros acumulados no grafo de covisibilidade. Alterações na altura da câmera e no ponto de vista das cenas durante a exploração podem fazer com que o ORB-SLAM não detecte um *loop* mesmo quando uma volta completa é realizada no ambiente, podendo ocasionar erros no mapa gerado.

Com o objetivo de garantir que o ORB-SLAM detecte os *loops*, um tripé foi utilizado para apoiar o celular com a câmera durante o mapeamento, como pode ser visto na Figura 5.1. O correto fechamento de *loops* em um sentido do ambiente corrige os erros acumulados e gera um mapa consistente com o ambiente, como o mapa mostrado na Figura 5.2.

Figura 5.1: Câmera durante exploração.

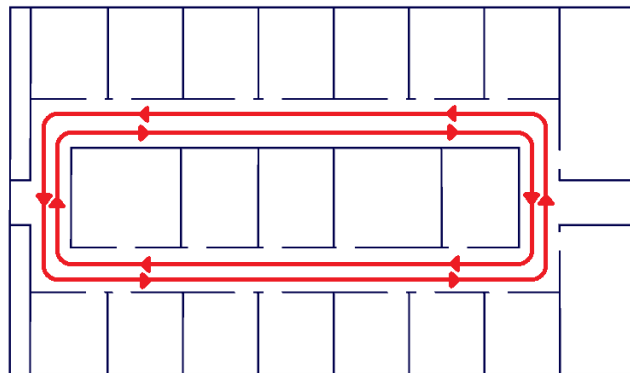


Figura 5.2: Mapa gerado em um sentido do corredor.



Para ser usado em situações reais, o sistema proposto deve funcionar durante o deslocamento em qualquer sentido nos corredores do ambiente desejado. Para isso, é preciso gerar um mapa com o ORB-SLAM que contemple o ambiente sendo percorrido em todas as direções, como mostrado na Figura 5.3. Dessa forma, uma pessoa pode ir e voltar pelo corredor enquanto o sistema realiza a sua localização e calcula a rota até o destino.

Figura 5.3: Os dois sentidos que o corredor pode ser percorrido.



A criação de um mapa pelo ORB-SLAM que contemple os dois sentidos de um corredor é uma tarefa difícil dado que os mesmos pontos vistos em cada um dos sentidos precisam ficar sobrepostos em um único mapa. Mesmo o ORB-SLAM sendo capaz de lidar bem com rotações e alterações nos pontos de vista das imagens, inverter o sentido do trajeto é um desafio, pois o ponto de vista da câmera muda completamente.

A Figura 5.4 mostra um conjunto de mapas criados pelo ORB-SLAM ao percorrer um corredor nos seus dois sentidos. Em todos os mapas criados, podemos ver que o ORB-SLAM faz o mapeamento para os dois sentidos percorridos como se não fossem o mesmo lugar, ou seja, sem estarem sobrepostos. Por não possuir odometria para mensurar

a distância e os ângulos das curvas percorridos, o ORB-SLAM pode não perceber que o sentido de exploração foi invertido após uma curva. Assim, o mapa gerado pode ficar rotacionado a partir do local onde foi realizada a inversão da trajetória, como pode ser visto na Figura 5.4a, na qual a inversão ocorreu no canto superior esquerdo do mapa, e na Figura 5.4b, na qual a inversão ocorreu no meio do corredor esquerdo do mapa.

O fato de não possuir uma escala definida, como explicado na seção 2.5, torna essa tarefa ainda mais difícil. Ao fechar um *loop* em um sentido oposto ao fechado anteriormente, o ORB-SLAM pode utilizar uma escala diferente da anterior. Sem a odometria, o ORB-SLAM utiliza apenas as informações das imagens capturadas pela câmera para se localizar. Porém, alterações no espaço e no posicionamento da câmera podem mudar a percepção do ambiente, impedindo o correto fechamento de *loops* e alterando a escala do mapa. Todos os mapas gerados possuem diferenças nas escalas das trajetórias de cada sentido. O mapa da Figura 5.4c é um exemplo extremo desse efeito, pois a escala do mapa foi reduzida em várias vezes após inverter o sentido de exploração, como pode ser visto no canto direito superior da imagem.

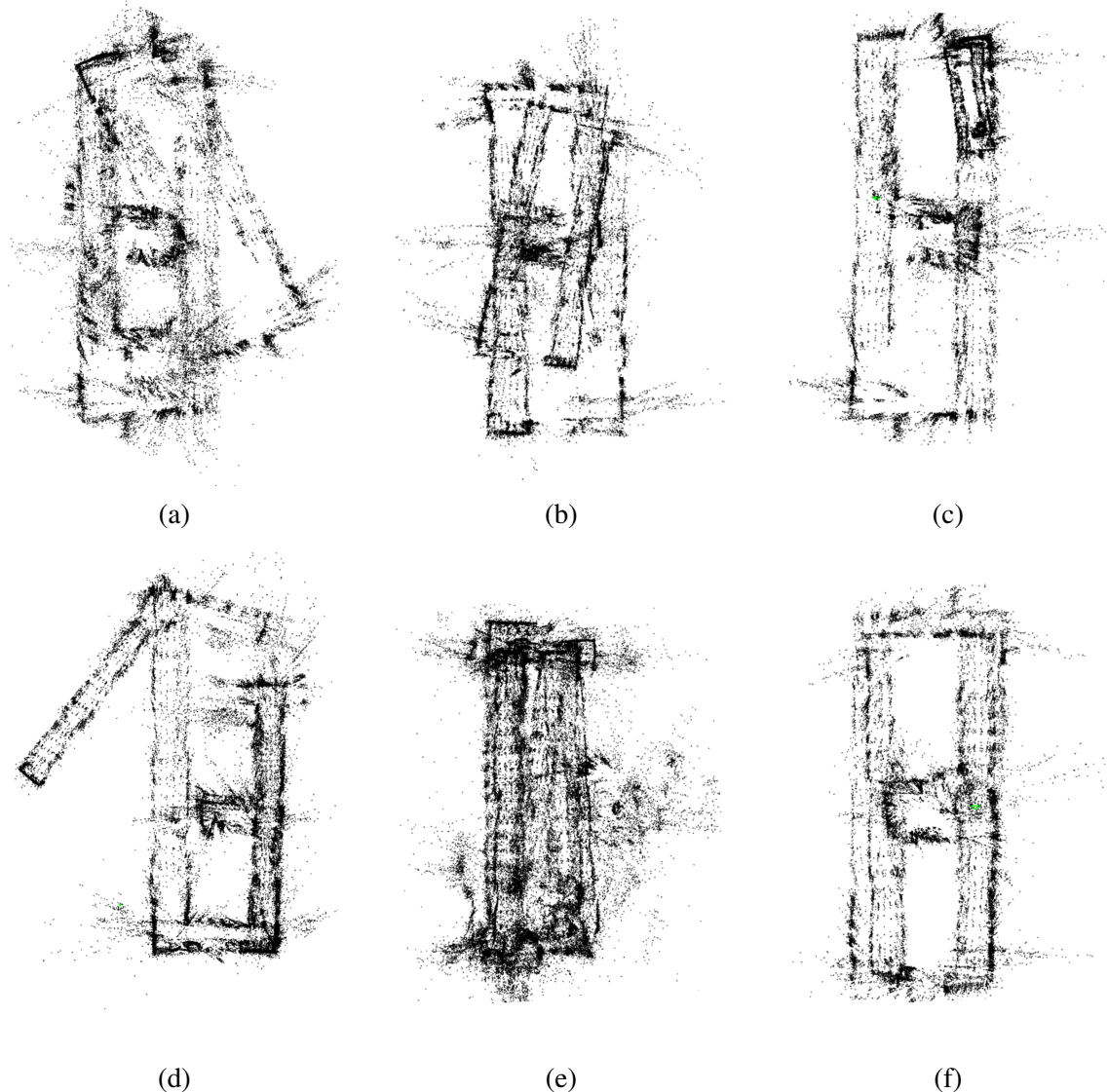
A realização de uma volta completa no ambiente sem que o ORB-SLAM detecte o fechamento de um *loop* pode deixar erros acumulados sem serem corrigidos no mapa de covisibilidade. Como explicado na seção 2.5, movimentos de rotação são críticos durante a tarefa de rastreamento do ORB-SLAM. Ao realizar uma volta completa no ambiente após a inversão do sentido de exploração de um corredor, um *loop* deve ser fechado para a correção do erro acumulado. Porém, o rastreamento pode ser perdido devido a rotação antes da detecção do *loop*. Nesse momento, o ORB-SLAM pode ter gerado pontos do mapa referentes ao mesmo local, porém em posições diferentes, como na Figura 4.2a. Ao fazer a realocação, a localização pode ocorrer nos pontos do mapa referentes ao início da volta e, dessa forma, não fechar o *loop*. Nesse caso, todo o trajeto realizado durante a volta pode ficar inconsistente no mapa, como mostrado na Figura 5.4d.

Por outro lado, o fechamento incorreto de um *loop* também pode causar incoerências no mapa gerado. Em ambientes com partes muito semelhantes, o ORB-SLAM pode detectar erroneamente um *loop*, ocasionando um erro na localização, além de corromper o mapa gerado. No mapa da Figura 5.4e, pode-se verificar o fechamento incorreto de *loops*, resultando em corredores deslocados e mesmo em um corredor na diagonal do mapa.

No mapa mostrado na Figura 5.4f, a criação dos mapas para cada sentido ficou o mais próximo de se sobreporem, porém o segundo mapa ainda apresenta uma escala menor e uma pequena translação. Como resultado, a parede detectada em um sentido de

exploração fica posicionada no meio do corredor detectado em outro sentido.

Figura 5.4: Mapas gerados nos dois sentidos de ambiente pelo ORB-SLAM.



A sobreposição incorreta dos mapas gerados em cada um dos sentidos impossibilita que eles sejam usados para o sistema proposto. A discrepância do mapa faz com que o ORB-SLAM retorne localizações diferentes para usuários no mesmo local, porém viradas para lados opostos. Outro problema gerado nesses casos é que uma sobreposição incorreta pode fazer com que pontos do mapa encontrados em um sentido sejam projetados sobre espaços livres do mapa no outro sentido. O caminho poderia parecer incorretamente bloqueado e impossibilitar o cálculo de rotas utilizando uma grade de ocupação, como explicado na seção seguinte. Dessa forma, não foi possível utilizar os mapas gerados com os dois sentidos no sistema proposto.

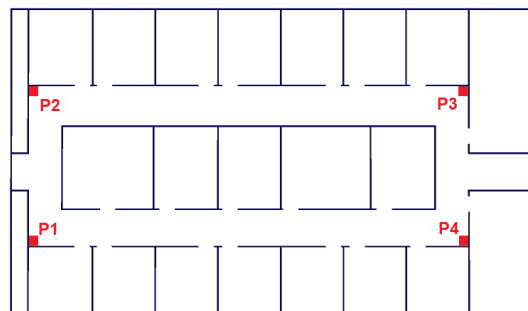
Para mostrar o funcionamento do sistema, optou-se por utilizar um mapa gerado

em apenas um sentido, como o da Figura 5.2. Assim, o mapa se mantém consistente com o ambiente, porém com a possibilidade de realizar caminhos em apenas um sentido do local. Como descrito na seção 4.2.1, o mapa gerado é então projetado sobre uma grade de ocupação.

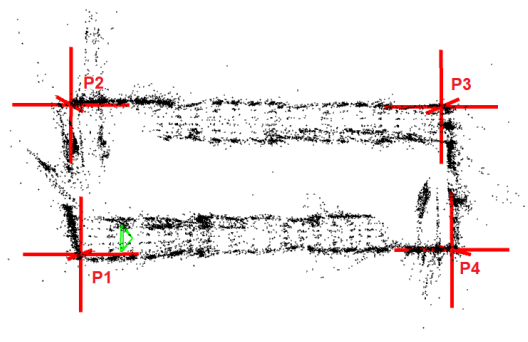
5.2 Grade de ocupação

O mapa gerado pelo ORB-SLAM não possui uma escala definida, conforme explicado na seção 2.5.1. Dessa forma, a escala do mapa precisa ser alterada para condizer com a escala real do ambiente. Para servir como referência da escala, uma planta baixa do ambiente foi criada e quatro pontos dela foram selecionados manualmente, como mostrado na Figura 5.5a. Foram escolhidos pontos nas extremidades dos corredores para facilitar o correto posicionamento deles na nuvem de pontos, como mostrado na Figura 5.5b.

Figura 5.5: Planta baixa e mapa de pontos do ambiente com os quatro pontos de referência destacados.



(a) Planta baixa do ambiente.

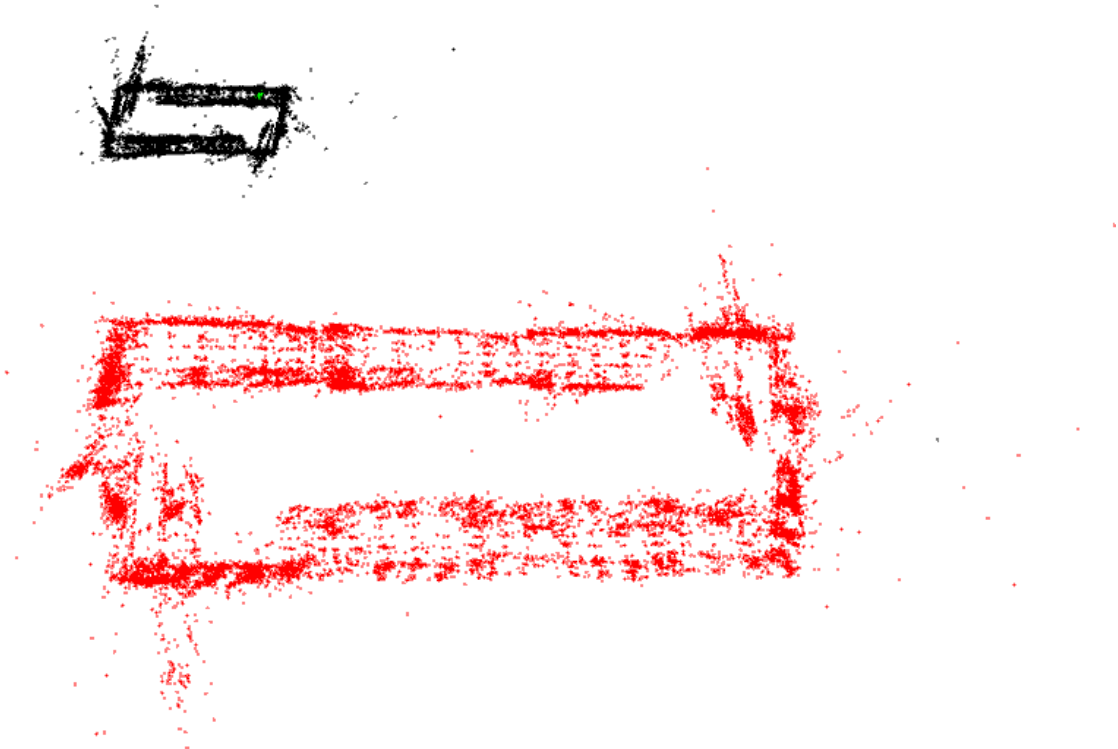


(b) Nuvem de pontos.

A partir da correspondência entre os pontos selecionados na planta baixa e os pontos da nuvem de pontos, é calculada uma matriz de transformação. A matriz calculada é utilizada para reescalar, rotacionar e transladar a nuvem de pontos original de modo que

a resultante possa ser projetada sobre a planta baixa. Na Figura 5.6 é possível ver a nuvem de pontos original e, abaixo dela, a transformada.

Figura 5.6: Nuvem de pontos original, em preto, e transformada, em vermelho.



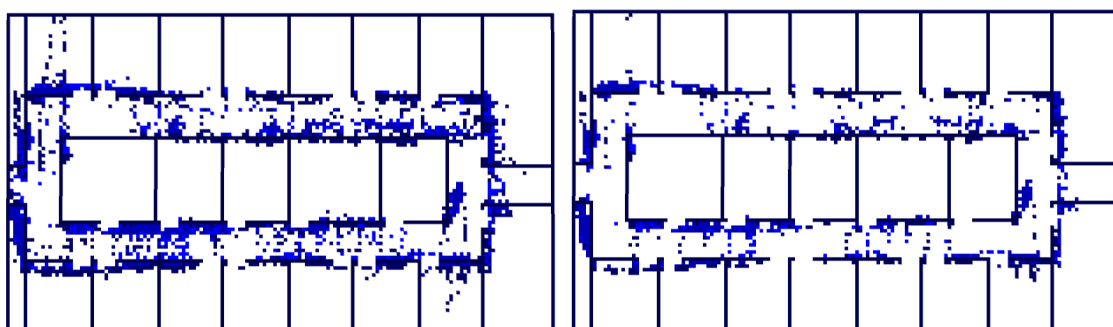
Inicialmente, é criada uma grade de ocupação bidimensional com as dimensões da planta baixa. A nuvem de pontos transformada possui a mesma escala da planta baixa, de modo que os pontos podem ser diretamente projetados sobre a grade de ocupação. Cada célula da grade de ocupação é considerada ocupada se uma parede é definida na planta baixa ou se ela possui um número mínimo de pontos da nuvem de pontos projetados sobre ela.

Os pontos de interesse encontrados pelo ORB-SLAM podem apresentar erros de posicionamento, como explicado na seção 2.5, resultando em células consideradas como ocupadas de forma indevida. Para minimizar esse problema, é definido um valor mínimo de pontos que uma célula deve ter projetado sobre ela para que ela seja considerada como ocupada. Por outro lado, a quantidade de pontos projetados sobre cada célula pode não ser muito alta, pois, como explicam Mur-Artal, Montiel e Tardós (2015), a nuvem de pontos criada pelo ORB-SLAM não é muito densa.

A Figura 5.7 mostra a nuvem de pontos projetada sobre a grade de ocupação com cada célula representando um espaço de 20 cm x 20 cm, considerando diferentes quantidades mínimas de pontos por célula. Quando uma célula é considerada ocupada com

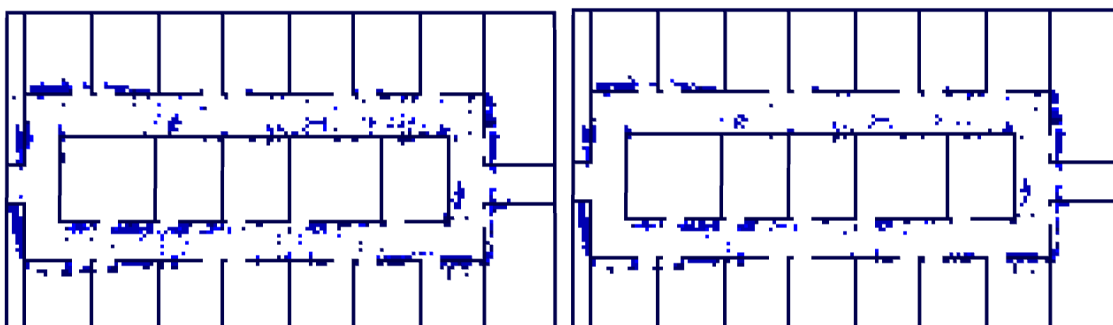
apenas dois pontos projetados sobre ela, como mostrado na Figura 5.7a, pode-se constatar um maior espalhamento das células ocupadas fora dos corredores. Com uma quantidade mínima de oito pontos projetados sobre uma célula para ela ser considerada como ocupada, como na Figura 5.7d, as células ocupadas ficam esparsas. Dessa forma, optou-se por considerar uma célula ocupada quando ao menos seis pontos sejam projetados sobre ela, como na Figura 5.7c.

Figura 5.7: Projeção dos pontos sobre a grade de ocupação.



(a) Ao menos dois pontos por célula.

(b) Ao menos quatro pontos por célula.



(c) Ao menos seis pontos por célula.

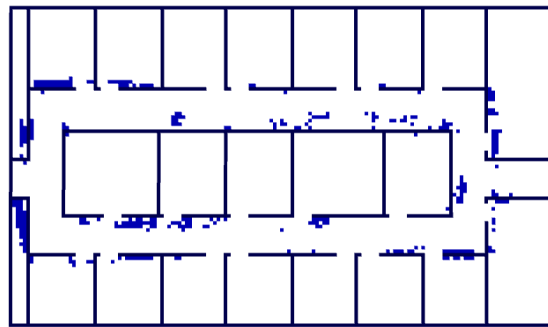
(d) Ao menos oito pontos por célula.

A altura dos pontos também é usada como critério para definir a ocupação das células. Apenas pontos com altura superior à h_{min} e inferior à h_{max} são considerados como obstáculos. Conforme descrito na seção 4.2.1, a altura h_{min} deve ter um valor alto o suficiente para que pontos com um erro na posição da sua altura não sejam erroneamente considerados como obstáculos. Por outro lado, o valor de h_{min} não pode ser tão alto que obstáculos verdadeiros sejam desconsiderados. Já o valor de h_{max} não deve gerar células ocupadas nos pontos referentes ao teto ou à obstáculos acima do espaço por onde uma pessoa pode passar.

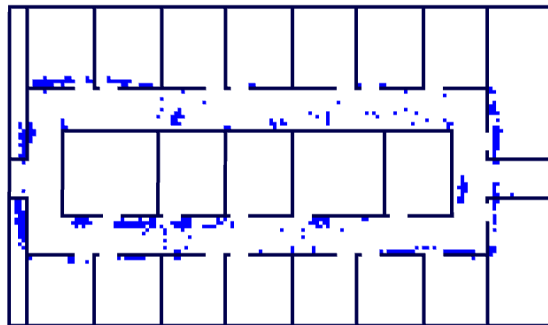
O valores de h_{min} e h_{max} podem ser alterados para se adequarem ao ambiente e a pessoa que está utilizando o sistema. Na Figura 5.8a é mostrada a grade de ocupação com ao menos seis pontos projetados em cada célula e com $h_{min} = 5cm$ e $h_{max} = 200cm$.

Já as Figuras 5.8b e 5.8c mostram a grade de ocupação com células com pontos menores que h_{min} como ocupadas e a grade de ocupação com células com pontos maiores que h_{max} como ocupadas, respectivamente. Casos essas células também fossem consideradas como ocupadas na grade de ocupação usada pelo sistema, o caminho seria quase que completamente bloqueado por falsos obstáculos. Com a grade de ocupação devidamente preenchida para representar os obstáculos do ambiente, ela pode ser usada para o planejamento de rotas, como explicitado em seguida.

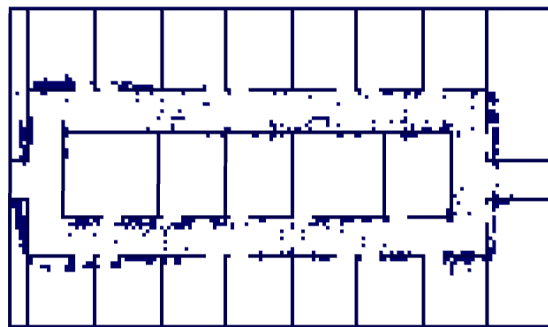
Figura 5.8: Grade de ocupação.



(a) Células consideradas como obstáculos.



(b) Células com pontos menores que $h_{min} = 5\text{cm}$.

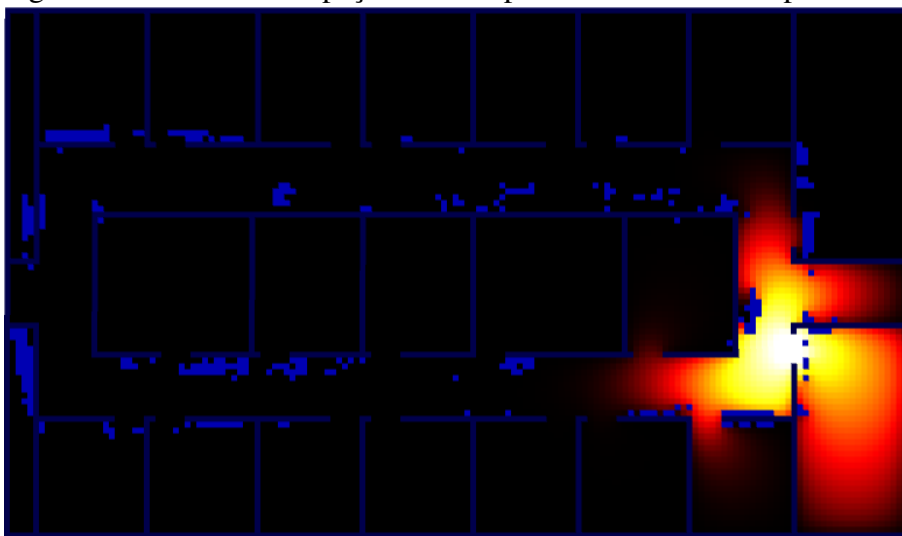


(c) Células com pontos maiores que $h_{max} = 200\text{cm}$.

5.3 Planejador BVP

A partir da grade de ocupação como a mostrada na Figura 5.8a e do destino recebido da pessoa que está utilizando o sistema, são calculados os potenciais relativos de cada célula com o método de Gauss-Seidel, como explicado na seção 3.1. A Figura 5.9 mostra a grade de ocupação com os valores potenciais de cada célula. Estes potenciais são representados pela variação de cor em direção a um ponto de destino que está localizado na parte inferior direita do corredor.

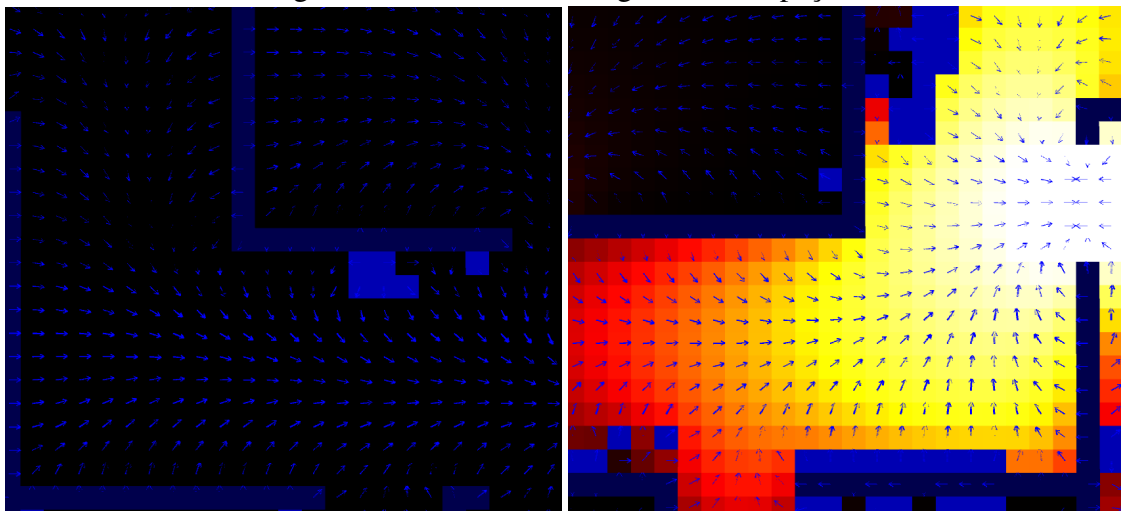
Figura 5.9: Grade de ocupação com os potenciais calculados pelo BVP.



Com os campos potenciais das células calculados, o trajeto até a posição desejada pode ser feito seguindo os gradientes de qualquer ponto até o ponto com potencial mínimo. A Figura 5.10 mostra os vetores dos gradientes em um ponto do mapa próximo do destino e em um ponto distante do destino. Os gradientes calculados são usados para orientar a pessoa que está utilizando o sistema a chegar a posição desejada desviando dos obstáculos encontrados pelo ORB-SLAM.

Na Figura 5.11 é possível verificar a importância da detecção dos obstáculos para o cálculo da rota até o destino. A rota definida na Figura 5.11a não leva em conta os obstáculos encontrados a partir da nuvem de pontos gerada pelo ORB-SLAM. Desse modo, a rota passa bem pelo centro do corredor e muito próximo dos obstáculos existentes. Já a rota definida na Figura 5.11b considera os obstáculos encontrados, assim, a rota é deslocada de modo a passar no espaço mais afastado entre a parede do corredor e os obstáculos. Dessa forma, a rota encontrada pode ser usada para guiar uma pessoa até um determinado destino, como explicado na seção seguinte.

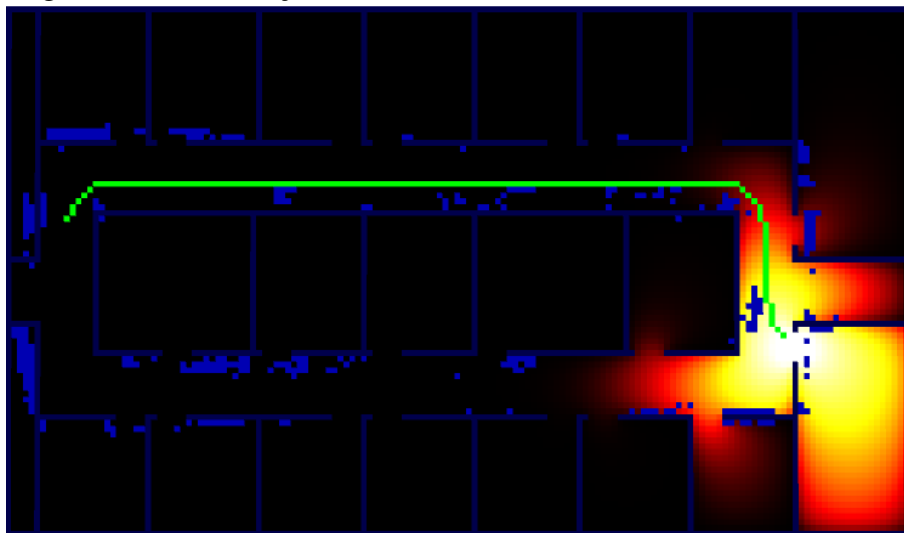
Figura 5.10: Gradientes na grade de ocupação.



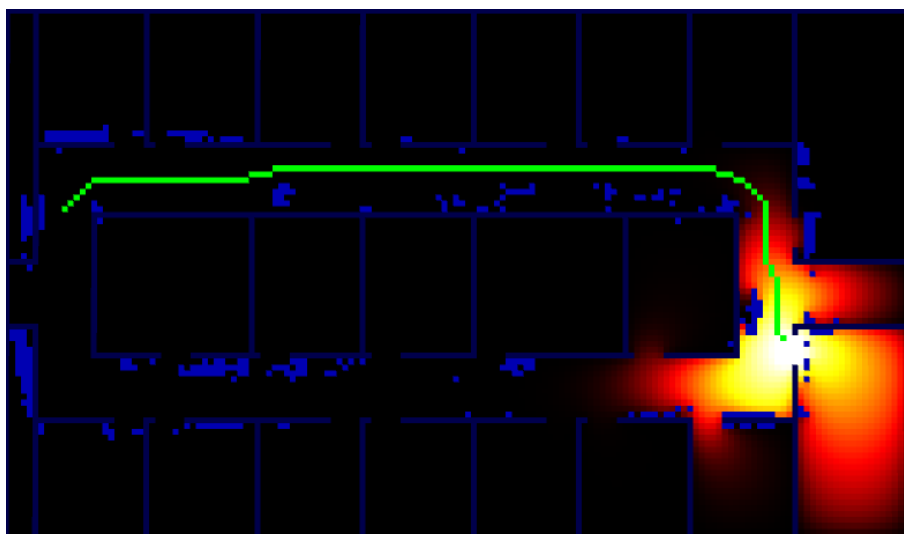
(a) Gradientes distantes do destino.

(b) Gradientes próximos do destino.

Figura 5.11: Diferença entre rotas encontradas com e sem obstáculos.



(a) Rota encontrada desconsiderando os obstáculos.

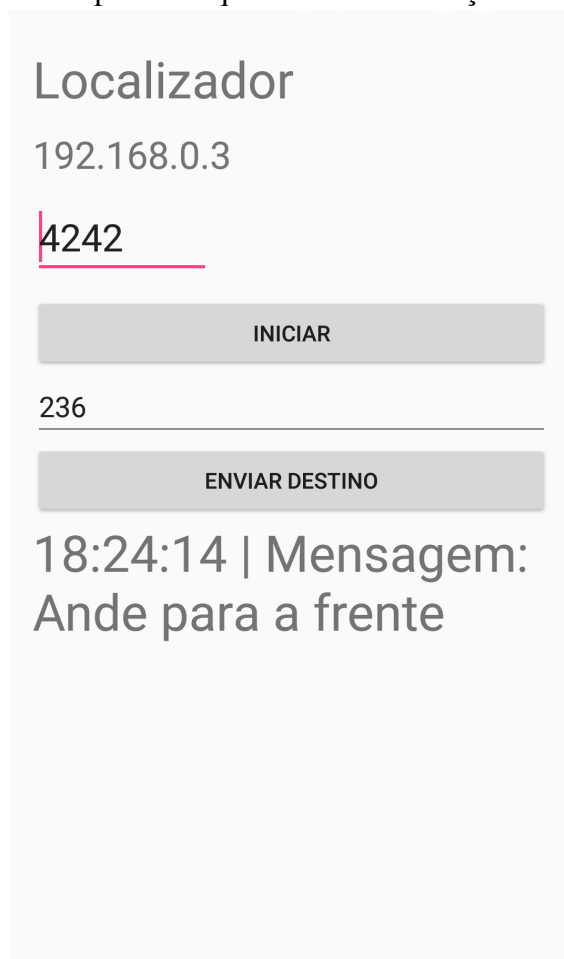


(b) Rota encontrada considerando os obstáculos.

5.4 Utilização

O sistema é utilizado com o auxílio de um celular que captura imagens do local e as envia para o servidor, assim como recebe as instruções de deslocamento do servidor. A Figura 5.12 mostra a interface do aplicativo que recebe as instruções do servidor. No aplicativo, o usuário configura o endereço IP e a porta para a comunicação com o servidor e clica em “INICIAR” para iniciar o sistema. Em seguida, ele seleciona o número da sala que deseja ir e clica em “ENVIAR DESTINO” para que o servidor calcule uma rota até o destino selecionado. Assim, o usuário, após enviar o destino desejado, recebe uma frase de instrução sobre como chegar ao destino para cada ponto em que a localização é estimada.

Figura 5.12: Aplicativo que recebe as instruções do servidor.

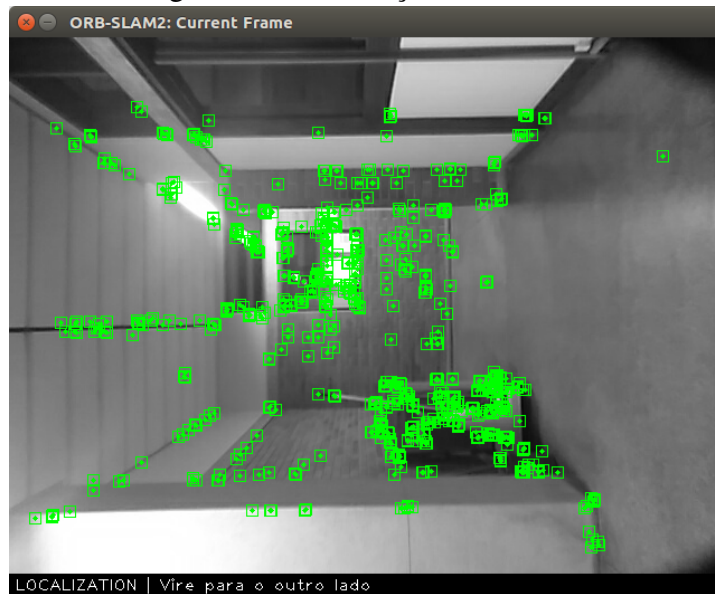


A instrução até o destino é encontrada ao comparar o vetor \vec{g} do gradiente da célula em que a pessoa se encontra com o vetor \vec{d} da direção que ela está se deslocando. O ângulo interno φ entre os dois vetores é usado para definir a mensagem enviada à pessoa utilizando o sistema, conforme os seguintes critérios:

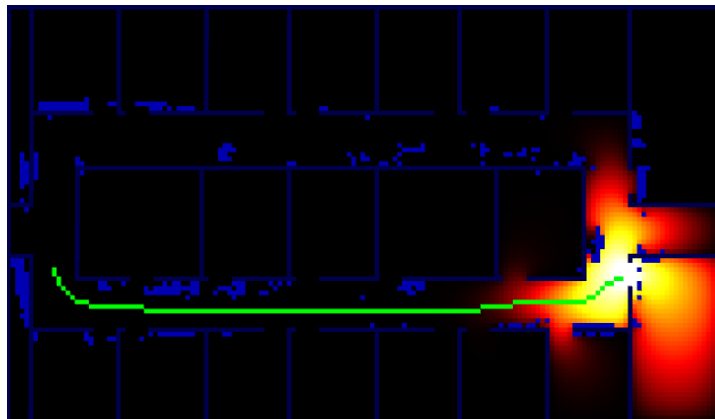
- se φ entre -5° e 5° , a mensagem é: “Ande para a frente”;
- se φ entre -45° e -5° , a mensagem é: “Ande para a direita”;
- se φ entre 5° e 45° , a mensagem é: “Ande para a esquerda”;
- se φ entre -45° e -225° , a mensagem é: “Vire para a direita”;
- se φ entre 45° e 225° , a mensagem é: “Vire para a esquerda”;
- se φ entre -225° e 225° , a mensagem é: “Vire para o outro lado”.

A rota gerada a partir da localização da pessoa que está utilizando o sistema e o respectivo quadro usado para definir a localização podem ser vistos na Figura 5.13. Como o quadro capturado pelo celular da pessoa está virado para o lado oposto do indicado pela rota encontrada, a instrução recebida indica para a pessoa se virar para o outro lado do qual ela está.

Figura 5.13: Utilização do sistema.



(a) Quadro registrado e instrução realizada.



(b) Rota encontrada.

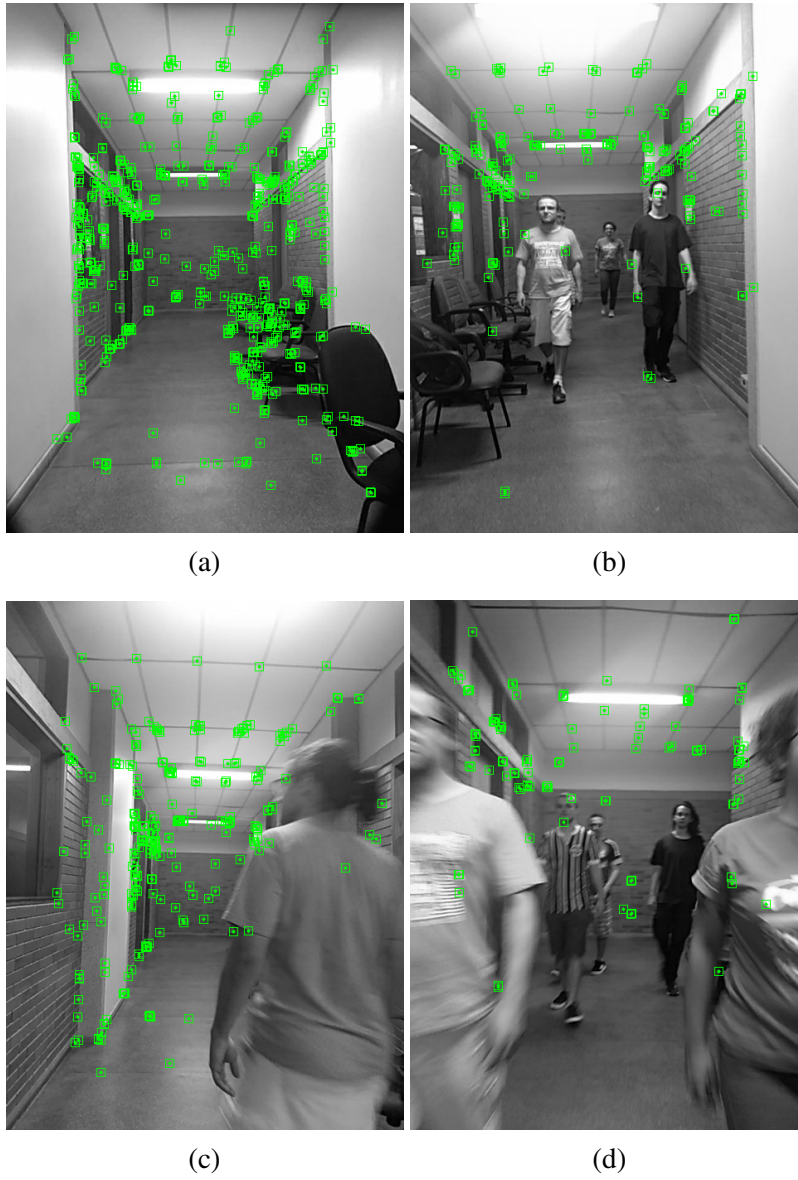
A geração do mapa pelo ORB-SLAM é feita em um momento que não existe circulação de pessoas pelo ambiente, como explicado na seção 4.1.1. Porém, durante a tarefa de localização, o ambiente é dinâmico e é possível que existam pessoas no caminho, assim como outras modificações nas características do espaço. As modificações são percebidas pelos quadros capturados e alteram os pontos de interesse reconhecidos pelo ORB-SLAM.

Como explicado na seção 2.5.2, o ORB-SLAM utiliza a quantidade de pontos de interesse em comum entre o quadro capturado e o grafo de covisibilidade para encontrar a posição atual da câmera. Desde que a quantidade mínima de pontos de interesse mapeados possa ser encontrada no quadro capturado, não importa que existam modificações na cena. Assim, a posição pode ser encontrada corretamente mesmo que tenham diferenças entre as cenas percebidas durante a localização e as cenas percebidas durante geração do mapa.

A Figura 5.14 mostra o ORB-SLAM realizando a localização em um ambiente com modificações da configuração inicial e com circulação de pessoas. Com o ambiente sem nenhuma alteração em relação ao momento da geração do mapa, mostrado na Figura 5.14a, o ORB-SLAM, como esperado, consegue se localizar e encontrar uma grande quantidade de pontos de interesse. Na Figura 5.14b o mesmo ambiente é percorrido, dessa vez com pessoas caminhando na direção contrária e alterações na disposição das cadeiras do corredor. Já nas Figuras 5.14c e 5.14d as pessoas aparecem muito próximas da câmera. Em todos os casos o ORB-SLAM conseguiu se localizar usando pontos de interesse que permaneceram inalterados no ambiente.

Apesar do bom desempenho em se localizar mesmo com alterações no ambiente, a diminuição dos pontos de interesse encontrados nos quadros dificulta a localização se houver rotações da câmera ou movimentos muito rápidos. Como explicam Mur-Artal, Montiel e Tardós (2015), uma quantidade maior de pontos de interesse encontrados deixa a localização mais robusta em condições adversas.

Figura 5.14: Quadros capturados com pessoas interferindo na localização.



6 CONCLUSÃO

O desenvolvimento de tecnologias assistivas pode auxiliar pessoas com deficiência a executarem tarefas que são complexas para elas. Entre essas tarefas está a auto-localização de pessoas com deficiência visual em ambientes internos desconhecidos. Pois, conforme visto na seção 1, pessoas com deficiência visual podem enfrentar dificuldades para se localizar em determinados ambientes, assim como encontrar uma rota até um destino desejado.

Nesse trabalho foi apresentado um sistema que pode ser usado para o desenvolvimento de tecnologia assistiva. Ele tem por objetivo auxiliar pessoas com deficiência visual a se localizarem e a se guiarem em ambientes previamente cadastrados. O sistema utilizou um telefone celular para capturar imagens do ambiente e enviá-las ao servidor. No servidor, as imagens foram processadas, com o uso do ORB-SLAM, com o intuito de inicialmente criar o mapa do ambiente e, em um segundo momento, encontrar a posição da pessoa nesse mapa. Um planejador BVP foi utilizado para encontrar rotas da posição da pessoa até o destino informado.

O produto desse trabalho foi um sistema capaz de determinar a localização de uma pessoa e criar rotas até o ponto desejado em um ambiente previamente cadastrado. Como mostraram os testes realizados, o ORB-SLAM conseguiu localizar a pessoa mesmo quando em ambientes dinâmicos, onde havia circulação de pessoas e modificações no espaço. O sistema também foi capaz de encontrar uma rota da localização do usuário até um destino desejado. Como resultado do uso da técnica de BVP, foram encontradas rotas suaves e que passam distantes dos obstáculos encontrados.

Diferentemente dos trabalhos correlatos apresentados, o sistema desenvolvido utiliza apenas a câmera de um celular como fonte de informações do ambiente. Assim, não é necessária a concepção e a utilização de equipamentos adicionais, como bengalas ou óculos inteligentes. Não são empregadas outras funcionalidades do telefone celular além da câmera, as quais podem não funcionar corretamente dependendo do ambiente, como o GPS. O sistema também não necessita de um retorno do usuário em pontos de referência pré-estabelecidos, que poderiam ser difíceis de ser localizados.

O sistema proposto serve como uma base para o desenvolvimento de uma solução para a locomoção de pessoas com deficiência visual. Para o desenvolvimento de uma solução final, a interface deve ser melhorada para atender o público alvo e testes reais devem ser realizados.

A geração de um mapa consistente de um ambiente sendo percorrido em ambas as direções se mostrou inviável. Por limitações do próprio ORB-SLAM, conforme descrito na seção 5.1, o mapeamento completo de um espaço gerou erros que impossibilitaram a utilização do mapa. Se faz necessário um aperfeiçoamento na tarefa de geração do mapa pelo ORB-SLAM para reduzir erros que podem corrompê-lo. Como alternativa, a tarefa de aquisição das imagens para a geração do mapa poderia ser realizada por um robô, por exemplo, melhorando a precisão e reduzindo oscilações causadas pela pessoa que faz o mapeamento.

A transformação da nuvem de pontos gerada pelo ORB-SLAM para a escala correta também é realizada de forma manual. Em um ambiente grande, no qual fosse necessário gerar um novo mapa com frequência, essa tarefa pode ser trabalhosa. Um método de transformação automática da nuvem de pontos para a escala desejada poderia melhorar a confiabilidade do sistema e otimizar a geração do mapa.

Apesar das limitações apresentadas, esse trabalho verificou a possibilidade de estimar o posicionamento do usuário com ORB-SLAM, mesmo em condições adversas, utilizando unicamente o celular como fonte de informações do ambiente. Além disso, o uso do planejador BVP se mostrou factível para a geração de rotas seguras considerando obstáculos presentes no ambiente. Portanto, as técnicas de ORB-SLAM e de planejamento de caminhos BVP se mostram efetivas no desenvolvimento de tecnologias assistivas. Como sugestão de continuidade ao trabalho, se faz interessante o aperfeiçoamento da interface de comunicação por meio de testes reais com usuários.

REFERÊNCIAS

- APOSTOLOPOULOS, I. et al. Integrated online localization and navigation for people with visual impairments using smart phones. In: **Robotics and Automation (ICRA), 2012 IEEE International Conference on**. [S.l.]: IEEE, 2012. p. 1322–1329.
- AULINAS, J. et al. The slam problem: A survey. In: **Proceedings of the 2008 Conference on Artificial Intelligence Research and Development: Proceedings of the 11th International Conference of the Catalan Association for Artificial Intelligence**. Amsterdam, The Netherlands, The Netherlands: IOS Press, 2008. p. 363–371. ISBN 978-1-58603-925-7.
- BASSO, S.; FRIGO, G.; GIORGI, G. A smartphone-based indoor localization system for visually impaired people. In: **Medical Measurements and Applications (MeMeA), 2015 IEEE International Symposium on**. [S.l.]: IEEE, 2015. p. 543–548.
- BAY, H.; TUYTELAARS, T.; GOOL, L. V. Surf: Speeded up robust features. In: LEONARDIS, A.; BISCHOF, H.; PINZ, A. (Ed.). **Computer Vision – ECCV 2006**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006. p. 404–417. ISBN 978-3-540-33833-8.
- BRADY, E. L. et al. Exploring interface design for independent navigation by people with visual impairments. In: **Proceedings of the 17th International ACM SIGACCESS Conference on Computers & Accessibility**. [S.l.]: ACM, 2015. (ASSETS '15), p. 387–388.
- BRASIL. **Subsecretaria Nacional de Promocao dos Direitos da Pessoa com Deficiencia. Comite de Ajudas Tecnicas. Tecnologia Assistiva**. [S.l.]: CORDE, 2009.
- BRASIL. **3a Conferencia Nacional dos Direitos da Pessoa com Deficiencia(Relatorio Final)**. [S.l.]: Secretaria Nacional de Promoção dos Direitos da Pessoa com Deficiencia (SNPD), 2012.
- CALONDER, M. et al. Brief: Binary robust independent elementary features. In: _____. **Computer Vision – ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part IV**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010. p. 778–792. ISBN 978-3-642-15561-1. Available from Internet: <https://doi.org/10.1007/978-3-642-15561-1_56>.
- CHIEN, H. J. et al. When to use what feature? sift, surf, orb, or a-kaze features for monocular visual odometry. In: **2016 International Conference on Image and Vision Computing New Zealand (IVCNZ)**. [S.l.: s.n.], 2016. p. 1–6.
- CHOI, S.; PARK, J.; YU, W. Resolving scale ambiguity for monocular visual odometry. In: **2013 10th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)**. [S.l.: s.n.], 2013. p. 604–608.
- CHOSSET, H. et al. **Principles of robot motion : theory, algorithms, and implementation**. [S.l.]: MIT Press, 2004.

CONNOLLY, C. I.; GRUPEN, R. A. The applications of harmonic functions to robotics. **Journal of Robotic Systems**, Wiley Subscription Services, Inc., A Wiley Company, v. 10, n. 7, p. 931–946, 1993. ISSN 1097-4563. Available from Internet: <<http://dx.doi.org/10.1002/rob.4620100704>>.

DANG, F.; DING, X.; ZHOU, P. A real independent centimeter-grade 3d indoor localization system on smartphone. In: **2016 International Wireless Communications and Mobile Computing Conference (IWCMC)**. [S.l.]: IEEE, 2016. p. 824–829.

DISCHINGER, M. The non-careful sight. In: _____. **Blindness and the Multi-sensorial City**. [S.l.]: Antwerp-Apeldoorn: Garant, 2006. p. 143–176.

ENDO, Y. et al. Indoor positioning and obstacle detection for visually impaired navigation system based on lsd-slam. In: **2017 International Conference on Biometrics and Kansei Engineering (ICBAKE)**. [S.l.: s.n.], 2017. p. 158–162.

ENGEL, J.; SCHÖPS, T.; CREMERS, D. Lsd-slam: Large-scale direct monocular slam. In: _____. **Computer Vision – ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part II**. Cham: Springer International Publishing, 2014. p. 834–849.

ENGEL, J.; STURM, J.; CREMERS, D. Semi-dense visual odometry for a monocular camera. In: **2013 IEEE International Conference on Computer Vision**. [S.l.: s.n.], 2013. p. 1449–1456. ISSN 1550-5499.

FOX, D. Markov localization: A probabilistic framework for mobile robot localization and navigation. 06 2002.

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

HESCH, J. A.; ROUMELIOTIS, S. I. An indoor localization aid for the visually impaired. In: **Proceedings 2007 IEEE International Conference on Robotics and Automation**. [S.l.]: IEEE, 2007. p. 3545–3551.

HULETSKI, A.; KARTASHOV, D.; KRINKIN, K. Evaluation of the modern visual SLAM methods. **Proceedings of Artificial Intelligence and Natural Language and Information Extraction, Social Media and Web Search FRUCT Conference, AINL-ISMW FRUCT 2015**, v. 7, p. 19–25, 2016.

KALMAN, R. E. A new approach to linear filtering and prediction problems. **Transactions of the ASME–Journal of Basic Engineering**, v. 82, n. Series D, p. 35–45, 1960.

KLEIN, G.; MURRAY, D. Improving the agility of keyframe-based slam. In: _____. **Computer Vision – ECCV 2008: 10th European Conference on Computer Vision, Marseille, France, October 12-18, 2008, Proceedings, Part II**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008. p. 802–815. ISBN 978-3-540-88688-4. Available from Internet: <https://doi.org/10.1007/978-3-540-88688-4_59>.

KÜMMERLE, R. et al. G2o: A general framework for graph optimization. In: **2011 IEEE International Conference on Robotics and Automation**. [S.l.: s.n.], 2011. p. 3607–3613. ISSN 1050-4729.

LANCIONI, G. E.; SINGH, N. N. **Assistive Technologies for People with Diverse Abilities**. [S.l.]: Springer New York, 2014.

LOWE, D. G. Object recognition from local scale-invariant features. In: **Proceedings of the International Conference on Computer Vision-Volume 2 - Volume 2**. Washington, DC, USA: IEEE Computer Society, 1999. (ICCV '99), p. 1150–. ISBN 0-7695-0164-8. Available from Internet: <<http://dl.acm.org/citation.cfm?id=850924.851523>>.

MUR-ARTAL, R.; MONTIEL, J. M. M.; TARDÓS, J. D. Orb-slam: A versatile and accurate monocular slam system. **IEEE Transactions on Robotics**, v. 31, n. 5, p. 1147–1163, Oct 2015. ISSN 1552-3098.

MUR-ARTAL, R.; TARDÓS, J. D. ORB-SLAM: Tracking and Mapping Recognizable Features. **Workshop on Multi View Geometry in Robotics (MVGRO) - RSS 2014**, 2014.

MUR-ARTAL, R.; TARDÓS, J. D. ORB-SLAM2: an Open-Source SLAM System for Monocular, Stereo and RGB-D Cameras. 2016.

MUR-ARTAL, R.; TARDÓS, J. D. Fast relocalisation and loop closing in keyframe-based slam. In: **2014 IEEE International Conference on Robotics and Automation (ICRA)**. [S.l.: s.n.], 2014. p. 846–853. ISSN 1050-4729.

PLATINSKY, L.; DAVISON, A. J.; LEUTENEGGER, S. Monocular visual odometry: Sparse joint optimisation or dense alternation? In: **2017 IEEE International Conference on Robotics and Automation (ICRA)**. [S.l.: s.n.], 2017. p. 5126–5133.

POGGI, M.; MATTOCCIA, S. A wearable mobility aid for the visually impaired based on embedded 3d vision and deep learning. In: **2016 IEEE Symposium on Computers and Communication (ISCC)**. [S.l.]: IEEE, 2016. p. 208–213.

PRESTES, E. et al. Exploration technique using potential fields calculated from relaxation methods. In: **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.: s.n.], 2001. v. 4, p. 2012–2017 vol.4.

ROMERO, R. A. F. et al. **Robotica movel**. [S.l.]: LTC, 2010.

ROSTEN, E.; DRUMMOND, T. Machine learning for high-speed corner detection. In: _____. **Computer Vision – ECCV 2006: 9th European Conference on Computer Vision, Graz, Austria, May 7-13, 2006. Proceedings, Part I**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006. p. 430–443. ISBN 978-3-540-33833-8. Available from Internet: <https://doi.org/10.1007/11744023_34>.

RUBLEE, E. et al. Orb: An efficient alternative to sift or surf. In: **2011 International Conference on Computer Vision**. [S.l.: s.n.], 2011. p. 2564–2571. ISSN 1550-5499.

SANTOS, D. D. D. et al. Breadcrumb: An indoor simultaneous localization and mapping system for mobile devices. In: **2016 IEEE Sensors Applications Symposium (SAS)**. [S.l.]: IEEE, 2016. p. 1–6.

SCARAMUZZA, D.; FRAUNDORFER, F. Visual odometry [tutorial]. **IEEE Robotics Automation Magazine**, v. 18, n. 4, p. 80–92, Dec 2011. ISSN 1070-9932.

SERRÃO, M. et al. Indoor localization and navigation for blind persons using visual landmarks and a gis. In: HADJILEONTIADIS, L. et al. (Ed.). **Proceedings of the 4th International Conference on Software Development for Enhancing Accessibility and Fighting Info-exclusion (DSAI 2012)**. [S.l.]: Elsevier B.V., 2012. p. 65–73.

SHRIDHAR, M.; KAI-YUAN, N. Monocular SLAM for Real-Time Applications on Mobile Platforms. p. 1–10, 2015.

SIEGWART, R.; NOURBAKHSI, I. R.; SCARAMUZZA, D. **Introduction to Autonomous Mobile Robots**. 2nd. ed. [S.l.]: MIT Press, 2011.

SILVEIRA, R.; PRESTES, E.; NEDEL, L. Fast path planning using multi-resolution boundary value problems. In: **2010 IEEE/RSJ International Conference on Intelligent Robots and Systems**. [S.l.: s.n.], 2010. p. 4710–4715. ISSN 2153-0858.

STRASDAT, H.; MONTIEL, J. M. M.; DAVISON, A. J. Scale drift-aware large scale monocular slam. In: **Robotics: Science and Systems**. [S.l.: s.n.], 2010.

TAPU, R.; MOCANU, B.; ZAHARIA, T. A computer vision system that ensure the autonomous navigation of blind people. In: **E-Health and Bioengineering Conference (EHB), 2013**. [S.l.]: IEEE, 2013. p. 1–4.

THRUN, S. Robotic mapping: A survey. In: LAKEMEYER, G.; NEBEL, B. (Ed.). **Exploring Artificial Intelligence in the New Millenium**. [S.l.]: Morgan Kaufmann, 2002.

TIAN, Y.; LIU, Y.; TAN, J. Wearable navigation system for the blind people in dynamic environments. In: **Cyber Technology in Automation, Control and Intelligent Systems (CYBER), 2013 IEEE 3rd Annual International Conference on**. [S.l.]: IEEE, 2013. p. 153–158.

TREVISAN, M. et al. Exploratory navigation based on dynamical boundary value problems. **Journal of Intelligent and Robotic Systems**, v. 45, n. 2, p. 101–114, Feb 2006. ISSN 1573-0409. Available from Internet: <<https://doi.org/10.1007/s10846-005-9008-2>>.

VERMEERSCH, P.-w.; HEYLIGHEN, A.; LEUVEN, K. U. Blindness and multi-sensoriality in architecture : the case of Carlos Mourão Pereira. Architectural Research Centers Consortium (ARCC), p. 393–400, 2012.

XIAO, J. et al. A low cost outdoor assistive navigation system for blind people. In: **2013 IEEE 8th Conference on Industrial Electronics and Applications (ICIEA)**. [S.l.]: IEEE, 2013. p. 828–833.

ZHANG, Y.; MA, L.; TAN, X. Smart phone camera image localization method for narrow corridors based on epipolar geometry. In: **2016 International Wireless Communications and Mobile Computing Conference (IWCMC)**. [S.l.]: IEEE, 2016. p. 660–664.

APÊNDICE A — TRABALHO DE GRADUAÇÃO 1

Sistema de auxílio à locomoção de pessoas com deficiência visual

Vinicius Maciel Hoff¹, Orientador Prof. Dr. Edson Prestes¹

¹Instituto de Informática – Universidade Federal do Rio Grande do Sul (UFRGS)
Caixa Postal 15.064 – 91.501-970 – Porto Alegre – RS – Brazil

{vmhoff, prestes}@inf.ufrgs.br

Abstract. *Anyone could have difficulties finding herself in an unknown environment, but this task becomes even more difficult when one cannot rely on visual information. The aim of this work is to help people with visual impairment to move around unknown indoor environments by using a navigation system based on the navigation techniques already known in robotics. Robot control systems use Visual SLAM and PnP Problem algorithms to estimate one's position in the environment and they use Path Planning algorithms to evaluate routes between two points. Environment Representations are used in every task. Thus, these techniques can be implemented in a mobile phone in an assistive technology application.*

Resumo. *Qualquer pessoa pode encontrar muitas dificuldades para localizar-se em ambientes com os quais não está familiarizada, mas a tarefa se torna-se ainda mais difícil quando não se pode contar com informações visuais. Com o objetivo de auxiliar pessoas com deficiências visuais se deslocarem em ambientes internos desconhecidos, esse trabalho propõe um sistema de locomoção baseado em técnicas de navegação que já são utilizadas na área de robótica. Sistemas de controle de robôs utilizam Visual SLAM e algoritmos de Problema PnP para estimar sua localização no ambiente, algoritmos de Path Planning para calcularem rotas entre dois pontos. além de Representações do Ambiente, as quais são usadas para cada tarefa. Assim, essas técnicas podem ser implementadas em um telefone celular para uma aplicação de tecnologia assistiva.*

1. Introdução

Localizar-se em ambientes desconhecidos e encontrar uma rota até um destino desejado pode ser um trabalho difícil para qualquer pessoa. No auxílio dessa tarefa, a tendência de lugares públicos e privados é a de disponibilizar placas informativas que possibilitem a localização das pessoas que por ali circulam. As informações fornecidas costumam ser do tipo visual, como plantas baixas, letreiros e setas informativas. Pessoas com deficiência visual não podem contar com esse tipo de informações e, por isso, podem encontrar muitas dificuldades de localização nesses espaços.

Uma das formas de auxiliar pessoas com deficiência à locomoverem-se em ambientes diversos é através de tecnologias assistivas. [Lancioni and Singh 2014] define tecnologias assistivas como equipamentos, comerciais ou customizados, que ajudam as pessoas à reduzir o impacto de suas incapacidades nas tarefas diárias e à alcançar uma melhor

qualidade de vida. Já o governo brasileiro, por meio do Comitê de Ajudas Técnicas, define que "Tecnologia Assistiva diz respeito à pesquisa, fabricação, uso de equipamentos, recursos ou estratégias utilizadas para potencializar as habilidades funcionais das pessoas com deficiência"[Brasil 2009, 11].

Com o advento dos *smartphones*, ou celulares inteligentes, surgem novas possibilidades para o desenvolvimento de tecnologias assistivas. Atualmente, os celulares já vêm equipados com sensores de tamanho reduzido, como câmeras, acelerômetros e giroscópios, os quais podem ser usados para localização e navegação à um preço acessível [Elloumi et al. 2016]. Um equipamento para auxílio de pessoas com deficiência visual pode fazer uso dos diversos tipos de sensores e interfaces que são disponibilizados. Tais características possibilitam a criação de equipamentos de tecnologias assistivas sem ser necessária a utilização de outros dispositivos além do próprio celular.

No Brasil, de acordo com o Censo Demográfico 2010, 3,6% da população é formada por pessoas com deficiência visual severa, [IBGE 2014]. O poder público tem buscado formas de proporcionar a inclusão de pessoas com deficiência física à sociedade. Um das formas que o governo federal utiliza para estimular a integração de pessoas com deficiência a todos os ambientes é por meio do estímulo às políticas de acessibilidade universal. À exemplo disso, a 3ª Conferência Nacional dos Direitos da Pessoa com Deficiência, tem, na área de acessibilidade, aprovadas propostas para "garantir e implementar equipamentos e serviços com tecnologia assistiva que assegurem a comunicação e a interação social entre as pessoas com deficiência e as demais"[Brasil 2012, 52] e "assegurar os meios de locomoção, criar instrumentos adequados para segurança das pessoas com deficiência em via pública e em espaços públicos e particulares"[Brasil 2012, 57].

O presente trabalho propõe o desenvolvimento de um sistema de localização com o uso de plataformas de telefones celulares. Tal sistema poderá ser utilizado para a implementação de um dispositivo de tecnologia assistiva capaz de auxiliar pessoas com deficiências visuais a se deslocarem em ambientes previamente cadastrados. À partir das informações cadastradas do ambiente e das informações capturadas pelos sensores do celular do usuário, o sistema deve estimar a posição do usuário no mapa, planejar uma rota até o destino desejado pelo usuário, indicar ao usuário como realizá-la, determinando constantemente a posição do utilizador e, caso necessário, replanejar a rota.

O presente artigo está organizado em cinco seções diferentes. Após a apresentação, desenvolvemos a explicação de conceitos importantes para a compreensão desse trabalho: Representação de Ambientes, Localização, Planejamento de Rotas e SLAM. Em seguida, são mostradas pesquisas relacionadas, destacando suas vantagens e limitações. Na quarta seção o sistema proposto é apresentado, com uma explicação do que deverá ser desenvolvido, o resultado final que espera-se alcançar e meios de verificação e validação do sistema. Na quinta seção é apresentado um plano de atividades com as etapas a serem realizadas para a conclusão do projeto.

2. Conceitos fundamentais

Um dispositivo de navegação destinado à auxiliar pessoas com deficiência visual deve executar tarefas como mapeamento, localização, e planejamento de caminhos. Essas tarefas são temas atuais de pesquisas na área de robótica. Ao passo que novos dispositivos são desenvolvidos, como sensores menores e mais precisos, algoritmos são criados e me-

lhorados para atender a demanda de novas aplicações.

Diversos conhecimentos já estabelecidos da área de robótica podem ser utilizados na construção de novas tecnologias assistivas. Assim, um sistema de localização para pessoas com deficiência visual pode aproveitar-se largamente de conceitos, algoritmos e modelos já trabalhados na área de robótica móvel.

2.1. Representação de Ambientes

Representações de ambientes são realizadas por meio de mapas, os quais são utilizados em diversas atividades humanas. Mapas podem ser dos mais diversos tipos, dependendo da aplicação a que se destinam. Uma forma de classificação de mapas é pelo nível de abstração que eles apresentam, podendo ser métricos e topológicos. Os primeiros são construídos com um baixo nível de abstração enquanto os segundos com um nível de abstração mais elevado [Romero et al. 2010].

Os mapas métricos possuem elementos que representam características de um local com uma posição geográfica definida. Esse tipo de mapa pode ser usado em aplicações que necessitem de informações detalhadas sobre o local, como a forma e o tamanho dos objetos, além de indicações dos espaços livres, como corredores e portas. Um exemplo de aplicação são as plantas baixas de prédios largamente utilizadas.

Em computação, um tipo de mapa métrico muito utilizado é a grade de ocupação, a qual consiste em células uniformes que dividem o espaço. As células informam se a região está ocupada ou livre. De posse de um mapa do tipo grade de ocupação é possível definir os possíveis caminhos que podem ser percorridos. O espaço representado por esse tipo de mapa podem ser tanto 2D como 3D [Choset et al. 2004].

Os mapas topológicos representam relações entre os elementos de um ambiente na forma de um grafo. Os elementos do grafo podem representar regiões, as quais possuem certas características em comum. Os grafos de um mapa topológicos especificam um conjunto de vértices e a conectividade entre eles. Os vértices indicam locais do espaço representado e as arestas relações entre esses locais. Uma aresta indica uma adjacência entre dois vértices e pode conter diferentes tipos de informações, como a existência de um caminho, a distância ou o tempo entre os dois locais representados pelo vértice [Siegwart et al. 2011].

Uma das principais vantagens dos mapas topológicos é que utilizam muito menos memória quando comparados com os mapas métricos. Isso se deve ao fato de um mapa topológico armazenar informações de forma mais abstratas, logo de forma mais eficaz e sem dados desnecessários. Por outro lado, o processo de construção de um mapa topológico é mais complexo e, por isso, tende a demandar um poder computacional maior. Isso acontece porque é necessário abstrair as informações do ambiente a fim de obter apenas as informações de interesse [Romero et al. 2010].

2.2. Localização

A representação da sua própria localização, em relação a um mapa, que um robô acredita possuir a cada momento é um item essencial para a navegação de robôs baseada em mapas. De acordo com [Romero et al. 2010], para representar a crença de sua localização, um robô pode fazer uso de mapas contínuos ou discretos, assim como pode usar uma

hipótese absoluta de localização ou múltiplas hipóteses de localização, entre as quais o robô deve lidar com uma incerteza em relação a hipótese correta.

Como demonstrado na (Figura 1) por [Siegwart et al. 2011], a tarefa de localização inclui quatro passos distintos: percepção; localização; cognição e controle de movimento. Existindo entre eles uma relação de dependência, a **percepção** é a interpretação das informações fornecidas pelos sensores, **localização** é o processo para determinar a localização com base na percepção realizada, **cognição** é a decisão sobre o que fazer com base na localização atual e o **controle de movimento** é a execução mecânica dos atuadores para realizar a tarefa determinada na cognição.

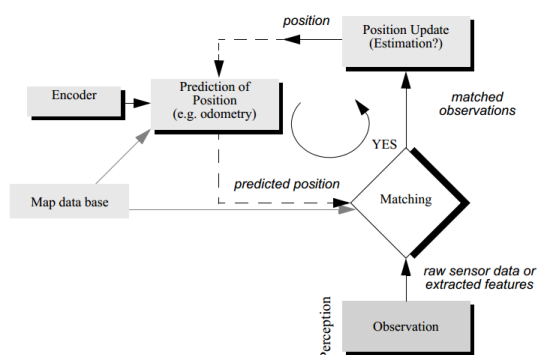


Figura 1. Esquema geral dos passos de localização de um robô

Sensores são a base para a tarefa de percepção e, por isso, sua imprecisão e incompletude afetam diretamente a tarefa de localização. Como exemplo, [Siegwart et al. 2011] mostra que sistemas de navegação para ambientes internos baseados em câmeras CCD podem sofrer interferência da iluminação do local e não reconhecer um mesmo espaço com diferentes níveis de iluminação.

Devido às incertezas e ruídos associados aos sensores, frequentemente torna-se impossível determinar uma única hipótese de localização [Romero et al. 2010]. Por consequência, ao invés de representar a localização como um posição (x,y) ou uma pose (x,y,θ) , no caso de mapas contínuos, ou uma única célula, no caso de mapas discretos, pode-se manter múltiplas hipóteses de localização.

Outra forma de representar uma localização é por meio de uma estimação probabilística da posição do robô. Nesse caso, ao invés de usar hipóteses sobre a posição do robô, calcula-se uma distribuição de probabilidade de todas as possíveis posições [Choset et al. 2004]. De acordo com [Romero et al. 2010], dois métodos principais são utilizados para localização probabilística de robôs baseada em mapas, a *Localização Markoviana* e o *Filtro de Kalman*.

Na *Localização Markoviana*, o processo de localização pode começar por qualquer posição desconhecida do mapa e pode ser executada mesmo em situações ambíguas. Para realizar essa tarefa, são rastreados múltiplas posições possíveis, o que requer a utilização de um mapa discreto, como explica [Siegwart et al. 2011]. Assim, é necessário mais memória e mais poder de processamento e pode limitar a precisão e o tamanho do mapa utilizado.

Já a localização por *Filtro de Kalman* estima o posicionamento do robô à partir de uma posição inicial conhecida, a qual precisa ser precisa. Logo, esse tipo de localização pode usar tanto mapas discretos como contínuos para representar o ambiente. Entretanto, no *Filtro de Kalman* a incerteza da posição do robô tende a aumentar com o tempo e, assim, pode falhar em capturar a posição real do robô [Siegwart et al. 2011].

2.3. Planejamento de Rotas

Na área de robótica, o planejamento de rotas refere-se ao cálculo e à descoberta de passos sequenciais a serem executados para que um robô saia de uma posição inicial e chegue a uma posição final, evitando colisões com obstáculos conhecidos. Os cálculos devem levar em conta a localização e orientação do robô no espaço de trabalho, assim como o seu espaço de configurações [Romero et al. 2010]. O espaço de configurações depende da quantidade de variáveis independentes utilizadas na configuração. Como exemplo, para um espaço de trabalho bidimensional (x,y) , um robô pode ter três graus de liberdade, dois para translação e um para orientação, e seu espaço de configuração tem três dimensões (x,y,θ) .

Segundo [Siegwart et al. 2011], para reduzir a complexidade do cálculo de rotas, é comum assumir o robô como um ponto, assim, o espaço de configurações é reduzido de três dimensões para um espaço bidimensional (x,y) . Com essa simplificação é importante aumentar o tamanho de cada obstáculo com o valor do raio do robô, outra modificação necessária para o planejamento de rotas é a discretização do mapa de representação do ambiente. Mapas discretos podem ser representados como uma grade de ocupação, o qual é adequado para a aplicação de duas classes de algoritmos comumente usados para o planejamento de rotas de robôs: *Planejamento em grafos* e *Planejamento em Campos Potenciais*.

O planejamento de rotas baseado em campos potenciais utiliza o conceito de gradientes para definir um potencial de atração gerado pelo ponto de destino e potenciais de repulsão gerados pelos obstáculos. Dessa forma, robô tende a, naturalmente, se dirigir ao destino, ao passo que desvia dos obstáculos [Romero et al. 2010]. A ideia é que os campos potenciais se comportem como forças, as quais atuam no robô até que o seu valor seja nulo, ou seja, existe uma convergência até um mínimo no campo potencial. O ponto de destino deve corresponder ao mínimo do campo de forças, de forma que o robô sempre tenda à ir em sua direção.

As rotas geradas pelo o planejamento de baseado em campos potenciais são de boa qualidade e que costumam passar longe dos obstáculos [Romero et al. 2010]. Entretanto, não há garantias que o destino seja um mínimo global, podendo haver mínimos locais no campo potencial. Assim, é necessário utilizar algoritmos que evitem mínimos locais que não correspondam ao destino e que façam o robô parar em pontos indesejados.

Para o planejamento de rotas baseado em grafos, os espaços livres do mapa são conectados, formando um grafo, e então é realizada uma busca de caminhos entre o ponto inicial e final. Técnicas de criação e busca em grafos é um campo largamente estudado em matemática e robótica [Siegwart et al. 2011]. À vista disso, existem diversos algoritmos que podem ser usados para essa tarefa. A maioria desses algoritmos pode ser dividido em duas etapas distintas, uma etapa de construção do grafo e uma etapa de busca do caminho no grafo.

A construção do grafo consiste em criar um conjunto de vértices e arestas de um grafo, que permitem o robô ir de qualquer ponto livre até outro ponto livre, sem colidir com obstáculos [Siegwart et al. 2011]. Entre as abordagens possíveis para a construção do grafo estão: grafo de visibilidade, diagrama de Voronoi, decomposição exata de células e grafos de Lattice. Cada uma delas apresenta vantagens e desvantagens, as quais podem ser fatores limitadores para o projeto, como proximidade da rota com os obstáculos e capacidade de memória necessária para armazenar o mapa.

Após obter um grafo conectado, a escolha da rota pode ser realizado através do cálculo do melhor caminho entre dois vértices do grafo. Novamente, existe uma grande quantidade de algoritmos diferentes que podem ser utilizado para realizar essa tarefa. Esses algoritmos podem ser divididos em duas classes distintas: algoritmos determinísticos e algoritmos randômicos. Um dos algoritmos probabilísticos largamente usados para o planejamento de rotas é o A*. Esse algoritmos é a combinação de outros dois tipicamente usados para essa tarefa, o algoritmo de busca melhor-primeiro e o algoritmo de Dijkstra.

Os algoritmos de busca randômicos podem ser usados quando o espaço de busca é muito grande e uma busca exaustiva teria um custo muito alto. Uma classe largamente usada desses algoritmos é o RTT (*Rapidly Exploring Random Tree*). Nele é criado uma árvore inicial, à qual são adicionados nodos de forma dinâmica, até encontrar um caminho entre o ponto inicial e final. Esses algoritmos não tem garantia de encontrar o melhor caminho, porém é provado que, se existir, uma rota será, eventualmente, encontrada [Siegwart et al. 2011].

2.4. SLAM

Para algumas tarefas, robôs devem ser capazes de estimarem sua posição ao mesmo tempo que coletam informações ao seu redor e constroem o mapa do ambiente [Huletski et al. 2016]. Em robótica, essa tarefa é conhecida como SLAM (*Simultaneous Localization and Mapping*) e, recentemente, tem sido alvo de muitas pesquisas. Essa incumbência apresenta muitos desafios devido à imprecisão dos sensores utilizados para coletar informações do ambiente, além do desafio de construir um mapa para localizar-se enquanto processa informações que atualizam o próprio mapa.

Diversos métodos de filtragem foram desenvolvidos para lidar com a falta de precisão dos sensores e são aplicados nos algoritmos de SLAM. Métodos baseados em inferência bayesiana são comumente usados, entre eles: Kalman Filter, Extended Kalman Filter (EKF) e Particle Filter [Huletski et al. 2016]. Esses métodos permitem o uso de SLAM de modo que todas as informações do ambiente são capturadas utilizando câmeras como sensores [Mur-Artal and Tardós 2016], o qual é conhecido por Visual Slam (V-SLAM).

Visual Slam pode ser executado com o uso de apenas uma câmera, o que é conhecido como Monocular SLAM [Mur-Artal and Tardós 2014]. A tarefa de localização e mapeamento com apenas uma câmera é muito mais difícil de ser realizada, pois não é possível determinar a profundidade de objetos com uma única imagem. Dessa forma, são necessárias imagens a partir de múltiplos pontos de vista e técnicas de filtragens para poder determinar a escala do mapa e calcular a trajetória realizada.

O ajuste dos parâmetros para estimar a pose da câmera e a estrutura da cena pode ser realizado apenas com as informações lidas até o momento ou então por blo-

cos de imagens (do inglês *bundle adjustment*) [Romero et al. 2010]. No primeiro caso, os parâmetros são calculados de forma incremental e, assim, de forma causal. Já no segundo grupo, os parâmetros são calculados de forma não causal, pois os resultados de imagens obtidas em tempos anteriores sofrem influência das imagens obtidas posteriormente.

Recentemente, foi desenvolvido por [Mur-Artal and Tardós 2016] um novo método para Monocular SLAM, chamado de ORB-SLAM. Nessa proposta, os parâmetros para estimar a pose da câmera e a estrutura da cena são calculados usando a abordagem de blocos de imagens. As limitações dessa abordagem para aplicações de tempo real, como o custo computacional, são corrigidas com uma série de otimizações e simplificações durante a coleta de imagens.

A partir dos parâmetros calculados pelo ORB-SLAM é possível determinar a pose e a estrutura da câmera resolvendo um problema do tipo *Perspective-n-Point* (PnP) [Huletski et al. 2016]. Um problema do tipo PnP consiste em determinar a posição e a orientação de uma câmera a partir dos seus parâmetros intrínsecos e as correspondências entre uma série de pontos 3D do ambiente e as suas projeções 2D na imagem [Moreno-moguer et al. 2007].

3. Trabalhos Relacionados

Apesar de ser uma área tipicamente da robótica, novas aplicações para os sistemas de localização e mapeamento tem surgido. Telefones celulares cada vez mais potentes e com sensores mais sofisticados expandem as possibilidades de utilização das tecnologias já existentes. Ao mesmo tempo, o desenvolvimento de tecnologias assistivas tem sido estimulado para auxiliar na acessibilidade de pessoas com deficiência. Assim, as técnicas de localização e mapeamento tem sido empregadas em propostas de aplicações para auxiliar pessoas com deficiência visual, assim como em soluções diversas executando em telefones celulares.

As soluções envolvendo as áreas de localização e mapeamento, tecnologias assistivas para pessoas com deficiência visual e aplicações sobre telefones celulares podem variar tanto nas tecnologias empregadas como no foco em uma ou em uma combinação dessas áreas ou na finalidade do sistema. Como exemplo, [Tapu et al. 2013] propõe um sistema utilizando Monocular SLAM em um celular para auxiliar pessoas com deficiência visual à locomoverem-se. Entretanto, ele não tem como finalidade ajudar na localização do usuário e sim na detecção de obstáculos que uma pessoa com deficiência visual possa encontrar ao percorrer um trajeto.

Outros autores propõem sistemas com a finalidade de auxiliar pessoas com deficiência visual à localizarem-se em ambientes diversos, porém não fazem uso apenas de celulares em seus projetos. Entre esses autores, [Hesch and Roumeliotis 2007], [Xiao et al. 2013], e [Poggi and Mattocchia 2016] propõem bengalas inteligentes, utilizando diferentes técnicas. Já [Tian et al. 2013] apresenta um dispositivo para localização acoplado em um óculos e [M. Serrão and du Buf 2012] utiliza uma câmera que deve ser carregada pelo usuário à mão.

Alguns autores propuseram sistemas de localização utilizando apenas os recursos fornecidos por um celular, porém sem serem direcionados à pessoas com deficiência visual. Esses trabalho fazem uso de diversas técnicas para calcular a posição do dispositivo,

porém todos devem fazer uso dos sensores disponíveis em telefones celular. Dessa forma, [Zhang et al. 2016] apresenta um sistema que usa apenas a câmera do celular para obter informações do ambiente. Já [Dang et al. 2016] e [Santos et al. 2016], além da câmera, eles também utilizam outros sensores, como giroscópios e acelerômetros.

Entre os sistemas que utilizam apenas celulares como equipamentos e são focados para o auxílio de pessoas com deficiência visual, mais de uma abordagem pode ser utilizada para resolver o problema. O sistema de GPS (Global Positioning Systems) é utilizado por [Basso et al. 2015], entretanto a tecnologia de GPS tem seu espaço de trabalho limitado, não funcionando em áreas internas [Siegwart et al. 2011]. Por isso, [Apostolopoulos et al. 2012] propõe um sistema utilizando a apenas câmera de um celular como sensor, nesse sistema o usuário deve dar um retorno ao sistema para confirmar a sua posição em pontos de referências pré-estabelecidos. Esse tipo de verificação nem sempre é trivial, já que pessoas com deficiência visual tem seu próprio meio de compreender o ambiente ao seu redor [Brady et al. 2015].

4. Proposta de Trabalho

Com esse trabalho, pretende-se empregar técnicas de navegação tipicamente usadas na área de robótica para o desenvolvimento de um sistema de tecnologia assistiva para auxiliar pessoas com deficiência visual à locomoverem-se em ambientes internos, aos quais elas não estão familiarizadas. Tecnologias desse tipo devem ser acessíveis ao maior número de pessoas possível, para isso, o sistema deve ser desenvolvido com o mínimo de recursos possíveis. Ao utilizar um telefone celular, outros dispositivos não são necessários. O sistema deve guiar o usuário desde sua posição inicial até o destino desejado. Para realizar a tarefa de navegação, uma sequencia de tarefas deve ser executada:

1. O usuário informa o destino desejado;
2. O sistema estima a posição atual do usuário no mapa;
3. O sistema planeja uma rota a partir da posição do usuário até a posição destino;
4. O sistema informa os passos a serem realizados até chegar à posição final;
5. O usuário locomove-se de acordo com as instruções recebidas;
6. Os passos 2 ao 5 são repetidos até o usuário alcançar o destino desejado.

O sistema pode ser dividido nas seguintes partes: Leitura de dados; Localização e Mapeamento (SLAM); Planejamento de Rota; Interface com o Usuário e Representação do Ambiente (mapas). A (Figura 2) mostra um diagrama com os componentes do sistema.

Requisitos de um projeto são importantes por diversos motivos, entre eles, a definição dos testes a serem empregados para validar o sistema final. A partir das funcionalidades descritas acima, pode-se definir os requisitos que o presente projeto deve apresentar:

- Um telefone celular deve ser o único recurso de hardware utilizado;
- O sistema deve estimar corretamente a posição do usuário;
- O sistema deve calcular uma rota possível até a posição desejada;
- O sistema não deve ser dependente de informações dadas pelo usuário;
- Em caso de incertezas, da posição ou da rota escolhida, o sistema deve informar ao usuário;

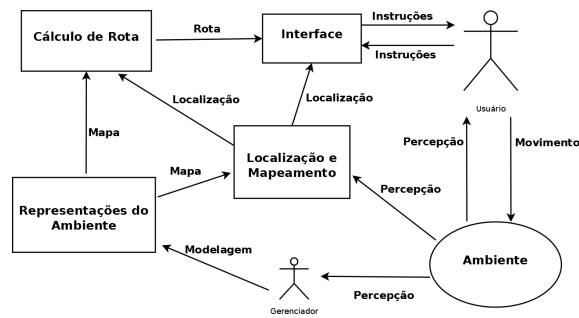


Figura 2. Diagrama mostrando os componentes do sistema proposto

Para que os requisitos sejam plenamente alcançados, o projeto deve ser desenvolvido com o estado da arte das técnicas de localização, mapeamento e planejamento de rotas. Cada uma das tecnologia e cada um dos algoritmo empregados deve ser integrado ao sistema de forma que formem um único produto. Durante esse processo, novas técnicas e adaptações de técnicas já desenvolvidas podem vir a serem necessárias.

A primeira tarefa a ser executada é a criação de mapas dos ambientes a serem explorados. A partir dos mapas, o sistema é capaz de estimar a posição do usuário e planejar rotas. Esses mapas devem ser cadastrados antes da utilização do sistema pelo usuário final. Dessa forma, o projeto deve prever o desenvolvimento de uma forma de criação de mapas topográficos dos ambientes desejados.

Por meio de SLAM é possível realizar a tarefa de localização e mapeamento do ambiente. Atualmente, ORB-SLAM é um métodos de SLAM baseados em características do ambiente com melhores resultados. Além disso, ORB-SLAM é otimizado para funcionar usando apenas uma câmera como sensor. Isso torna essa abordagem indicada para o uso em plataformas de celulares. Com s informações sobre o ambiente fornecidas pelo ORB-SLAM e dos mapas previamente cadastradas, é possível estimar a localização do usuário por meio de um algoritmo para resolver problemas PnP . A última etapa a ser realizada pelo sistema é cálculo de uma rota para levar o usuário até o destino desejado. A partir da posição estimada anteriormente e dos mapas cadastrados, o sistema pode executar um algoritmo de busca de rotas.

Por fim, um aplicativo de celular deve ser capaz de gerências todas as partes do sistema e realizar a interface com o usuário. Para esse trabalho, cada uma das partes do sistema deve ser implementada de forma a apresentar o melhor desempenho e melhor resultado para o usuário, que, nesse caso, são representadas por pessoas com deficiências visuais que precisam localizarem-se em ambientes desconhecidos.

Para a verificação do funcionamento do projeto e a validade do sistema deve-se testar a performance de cada uma das partes separadamente. Isto é, deve-se atestar que cada parte realize a tarefa a qual é destinada. Logo, os testes devem verificar: a porcentagem de erro entre a posição real do usuário e a posição estimada, a diferença entre a menor rota possível e a rota planejada pelo sistema, se o sistema entende corretamente os comandos, da mesma forma que se envia as instruções corretamente.

O foco desse trabalho é a implementação de tecnologias de navegação em uma plataforma celular. A comunicação do sistema com o usuário não está no escopo desse

projeto, sendo uma parte adicional dele. Assim, os testes de validação poderão ser realizados por meio de mensagens de texto entre o usuário e o sistema.

5. Plano de Atividades

O projeto proposto deve ser realizado no período máximo de um semestre, sendo o fim do primeiro semestre de 2016 o prazo máximo para a entrega do relatório final e sua apresentação. Dessa forma, o plano de atividades contido na Tabela 1 apresenta o período que cada uma das atividades referentes ao projeto deve ser realizada, começando já no mês de dezembro de 2015 e finalizando no mês de junho de 2016. Para facilitar a separação das tarefas, o cronograma está dividido em quinzenas, mostrando a primeira e a segunda quinzena de cada mês.

Definiu-se as seguintes tarefas a serem feitas para a realização do projeto:

1. Pesquisa Bibliográfica
2. Instalação e configuração do sistema ORB-SLAM
3. Desenvolvimento do sistema de mapas
4. Desenvolvimento de algoritmos para problema PnP
5. Desenvolvimento de algoritmos de cálculo de rotas
6. Integração das partes do sistema
7. Desenvolvimento do aplicativo Android
8. Testes funcionais
9. Testes de usabilidade do aplicativo
10. Validação e levantamento de dados do projeto
11. Desenvolvimento da monografia
12. Apresentação final do trabalho

Tabela 1. Plano de atividades por quinzenas

Período	2015				2016									
	Dezembro		Janeiro		Fevereiro		Março		Abril		Maio		Junho	
	1º	2º	1º	2º	1º	2º	1º	2º	1º	2º	1º	2º	1º	2º
Atividade														
1														
2														
3														
4														
5														
6														
7														
8														
9														
10														
11														
12														

6. Referências

Referências

- Apostolopoulos, I., Fallah, N., Folmer, E., and Bekris, K. E. (2012). Integrated online localization and navigation for people with visual impairments using smart phones. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 1322–1329. IEEE.
- Basso, S., Frigo, G., and Giorgi, G. (2015). A smartphone-based indoor localization system for visually impaired people. In *Medical Measurements and Applications (MeMeA), 2015 IEEE International Symposium on*, pages 543–548. IEEE.
- Brady, E. L., Sato, D., Ruan, C., Takagi, H., and Asakawa, C. (2015). Exploring interface design for independent navigation by people with visual impairments. In *Proceedings of the 17th International ACM SIGACCESS Conference on Computers & Accessibility, ASSETS '15*, pages 387–388. ACM.
- Brasil (2009). *Subsecretaria Nacional de Promocao dos Direitos da Pessoa com Deficiencia. Comite de Ajudas Tecnicas. Tecnologia Assistiva. CORDE.*
- Brasil (2012). *3a Conferencia Nacional dos Direitos da Pessoa com Deficiencia(Relatorio Final). Secretaria Nacional de Promoção dos Direitos da Pessoa com Deficiencia (SNPD).*

- Choset, H., Lynch, K., Hutchinson, S., Kantor, G., Burgard, W., Kavraki, L., and Thrun, S. (2004). *Principles of robot motion : theory, algorithms, and implementation*. MIT Press.
- Dang, F., Ding, X., and Zhou, P. (2016). A real independent centimeter-grade 3d indoor localization system on smartphone. In *2016 International Wireless Communications and Mobile Computing Conference (IWCMC)*, pages 824–829. IEEE.
- Elloumi, W., Latoui, A., Canals, R., Chetouani, A., and Treuillet, S. (2016). Indoor pedestrian localization with a smartphone: A comparison of inertial and vision-based methods. *IEEE Sensors Journal*, pages 5376–5388.
- Hesch, J. A. and Roumeliotis, S. I. (2007). An indoor localization aid for the visually impaired. In *Proceedings 2007 IEEE International Conference on Robotics and Automation*, pages 3545–3551. IEEE.
- Huletski, A., Kartashov, D., and Krinkin, K. (2016). Evaluation of the modern visual SLAM methods. *Proceedings of Artificial Intelligence and Natural Language and Information Extraction, Social Media and Web Search FRUCT Conference, AINL-ISMW FRUCT 2015*, 7:19–25.
- IBGE (2014). *Censo Demografico 2010 Caracteristicas gerais da populacao, religiao e pessoas com deficiencia*. Instituto Brasileiro Geografia e Estatistica.
- Lancioni, G. E. and Singh, N. N. (2014). *Assistive Technologies for People with Diverse Abilities*. Springer New York.
- M. Serrão, J.M.F. Rodrigues, J. R. and du Buf, J. (2012). Indoor localization and navigation for blind persons using visual landmarks and a gis. In Leontios Hadjileontiadis, Paulo Martins, R. T. H. P. J. R. and Barroso, J., editors, *Proceedings of the 4th International Conference on Software Development for Enhancing Accessibility and Fighting Info-exclusion (DSAI 2012)*, pages 65–73. Elsevier B.V.
- Moreno-moguer, F., Lepetit, V., and Fua, P. (2007). Accurate Non-Iterative O(n) Solution to the PnP Problem. *Iccv*.
- Mur-Artal, R. and Tardós, J. D. (2014). ORB-SLAM: Tracking and Mapping Recognizable. *Workshop on Multi View Geometry in Robotics (MVGRO) - RSS 2014*.
- Mur-Artal, R. and Tardós, J. D. (2016). ORB-SLAM2: an Open-Source SLAM System for Monocular, Stereo and RGB-D Cameras.
- Poggi, M. and Mattoccia, S. (2016). A wearable mobility aid for the visually impaired based on embedded 3d vision and deep learning. In *2016 IEEE Symposium on Computers and Communication (ISCC)*, pages 208–213. IEEE.
- Romero, R. A. F., Junior, E. P. S., Osorio, F. S., and Wolf, D. F. (2010). *Robotica movel*. LTC.
- Santos, D. D. D., Lam, A. J., Macatangay, J., Paner, I., and Ngo, C. A. (2016). Bread-crumbs: An indoor simultaneous localization and mapping system for mobile devices. In *2016 IEEE Sensors Applications Symposium (SAS)*, pages 1–6. IEEE.
- Siegwart, R., Nourbakhsh, I. R., and Scaramuzza, D. (2011). *Introduction to Autonomous Mobile Robots*. MIT Press, 2nd edition.

- Tapu, R., Mocanu, B., and Zaharia, T. (2013). A computer vision system that ensure the autonomous navigation of blind people. In *E-Health and Bioengineering Conference (EHB), 2013*, pages 1–4. IEEE.
- Tian, Y., Liu, Y., and Tan, J. (2013). Wearable navigation system for the blind people in dynamic environments. In *Cyber Technology in Automation, Control and Intelligent Systems (CYBER), 2013 IEEE 3rd Annual International Conference on*, pages 153–158. IEEE.
- Xiao, J., Ramdath, K., Losilevish, M., Sigh, D., and Tsakas, A. (2013). A low cost outdoor assistive navigation system for blind people. In *2013 IEEE 8th Conference on Industrial Electronics and Applications (ICIEA)*, pages 828–833. IEEE.
- Zhang, Y., Ma, L., and Tan, X. (2016). Smart phone camera image localization method for narrow corridors based on epipolar geometry. In *2016 International Wireless Communications and Mobile Computing Conference (IWCMC)*, pages 660–664. IEEE.