UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO

GEAN MARCIEL DOS SANTOS STEIN

NBD-BRIEF: Utilizando informações de proximidade de edifícios na localização global de VANT sobre imagens de satélite

Dissertação apresentada como requisito parcial para a obtenção do grau de Mestre em Ciência da Computação

Orientador: Prof. Dr. Renan de Queiroz Maffei
Co-orientador: Prof. Dr. Edson Prestes e Silva Júnior

Porto Alegre
2023

CIP — CATALOGAÇÃO NA PUBLICAÇÃO

Stein, Gean Marciel dos Santos

NBD-BRIEF: Utilizando informações de proximidade de edifícios na localização global de VANT sobre imagens de satélite / Gean Marciel dos Santos Stein. – Porto Alegre: PPGC da UFRGS, 2023.

63 f.: il.

Dissertação (mestrado) – Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Computação, Porto Alegre, BR–RS, 2023. Orientador: Renan de Queiroz Maffei; Coorientador: Edson Prestes e Silva Júnior.

1. Localização. 2. Segmentação. 3. VANT. 4. BRIEF. 5. MCL. I. Maffei, Renan de Queiroz. II. Júnior, Edson Prestes e Silva. III. Título.

“There ain’t no answer.

There ain’t gonna be any answer.

There never has been an answer.

That’s the answer.”

— Gertrude Stein

AGRADECIMENTOS

Inicialmente agradeço a minha família. Meus pais e meu irmão que sempre me apoiaram, motivaram e estiveram ao meu lado durante toda a minha vida.

Agradeço também aos professores e colegas do grupo Phi pela ajuda dada durante todo o período do trabalho, em especial ao Prof. Renan que não desistiu de mim e sem ele essa dissertação não aconteceria.

Aos amigos Ricardo Westhauser e Renato Peralta que me acompanharam desde o início do mestrado. E a Rodrigo Klanovicz pelo apoio e compreensão em todas as etapas.

RESUMO

Veículos Aéreos Não Tripulados (VANTs) dependem fortemente de sistemas de localização, geralmente GPS, para realizar suas tarefas com segurança e eficiência. No entanto, o uso de GPS não está isento de problemas, e depender de uma única fonte de medição pode ser desastroso para a operação do VANT. Para mitigar esse risco, este trabalho propõe um sistema de localização visual de VANT que serve como uma fonte adicional de estimativa de pose no caso de mau funcionamento do GPS.

A melhoria do nosso sistema está no modelo de observação usado na localização de Monte Carlo, que utiliza um novo descritor chamado NBD-BRIEF. Esse descritor é baseado em informações de construções presentes no ambiente, através de uma métrica chamada Distância de Construção Mais Próxima (NBD - Nearest Building Distance) obtidas na posição do veículo. Tal informação é mais estável e menos suscetível a mudanças de luz e cor do que os métodos tradicionais de correspondência baseados em cores. Extraímos o contorno dos prédios das imagens do VANT usando uma rede convolucional para fornecer informações semânticas ao nosso sistema.

Nossos experimentos demonstram que o descritor NBD-BRIEF supera outras abordagens para o mesmo problema. Além disso, nosso sistema de localização visual de VANT baseado em NBD-BRIEF estima com precisão a pose do VANT em três voos diferentes, enquanto os outros métodos de localização que comparamos, por serem semelhantes ao nosso, falharam. Em resumo, nosso sistema de localização visual proposto fornece uma solução robusta para a estimativa de pose de VANTs que pode aumentar a segurança e eficiência das operações de VANTs.

Palavras-chave: Localização. segmentação. VANT. BRIEF. MCL.

NBD-BRIEF: Using buildings proximity information in UAV global localization over satellite images

ABSTRACT

Unmanned Aerial Vehicles (UAVs) rely heavily on localization systems, typically GPS, to perform their tasks safely and efficiently. However, this approach is not without issues, and relying on a single source of measurement could be disastrous for the UAV's operation. To mitigate this risk, this work proposes a visual UAV localization system that serves as an additional source of pose estimation in the event of GPS malfunctioning.

The novelty of our system lies in the measurement model used in Monte Carlo Localization, which utilizes a new descriptor called NBD-BRIEF. This descriptor is based on Nearest Building Distance (NBD) information obtained at the vehicle position and is more stable and less susceptible to changes in light and color than traditional color-based matching methods. We extract the building footprint from the UAV images using a convolutional network to provide semantic information to our system.

Our experiments demonstrate that our NBD-BRIEF descriptor outperforms other approaches for the same problem. Furthermore, our visual UAV localization system based on NBD-BRIEF accurately estimates the UAV's pose in three flights, whereas other localization methods we compared, which are similiar to ours, failed. In summary, our proposed visual localization system provides a more robust solution to UAV pose estimation that could enhance the safety and efficiency of UAV operations.

Keywords: localization. segmentation. UAV. BRIEF. MCL.

LISTA DE ABREVIATURAS E SIGLAS

abBRIEF	*ab BRIEF* (BRIEF usando canais *ab*)
BRIEF	*Binary Robust Independent Elementary Features* (Características Elementares Independentes Binárias Robustas)
BRIEF-EB	BRIEF com Entrada Binária
CNN	*Convolutional Neural Network* (Rede Neural Convolucional)
EAM	Erro Absoluto Médio
GPS	*Global Positioning System* (Sistema de Posicionamento Global)
HOG	*Histogram of Oriented Gradient* (Histograma de Gradientes Orientados)
IMU	*Inertial Measuring Unit* (Unidade de Medições Inercial)
MCL	*Monte Carlo Localization* (Localização de Monte Carlo)
NBD	*Nearest Building Distance* (Distância para Construção Mais Próxima)
ORB	*Oriented FAST and Rotated BRIEF* (FAST orientado e BRIEF rotacionado)
SLAM	*Simultaneous Localization And Mapping* (Localização e Mapeamento Simultâneos)
SIFT	*Scale-invariant feature transform* (Transformação de Características Invariante a Escala)
SURF	*Speeded Up Robust Features* (Características Robustas Aceleradas)
U-NET	Rede neural U-Net
VANT	Veículo aéreo não tripulado

LISTA DE FIGURAS

LISTA DE TABELAS

SUMÁRIO

1 INTRODUÇÃO

1.1 Motivação

Nos últimos anos, veículos aéreos não tripulados (VANTs) têm sido utilizados em muitas tarefas diferentes em robótica e automação, como transporte de produtos médicos (THIELS et al., 2015), agricultura (COSTA et al., 2012) e até mesmo exploração de Marte (SERNA et al., 2020). Em todas essas aplicações, é crucial estimar com precisão a posição do VANT, especialmente quando ele está operando autonomamente. Embora a maioria dos VANTs dependa de seus sensores de sistema de posicionamento global, do inglês, *Global Positioning System* (GPS), incorporados para a estimativa de posição, problemas com o sinal GPS, como a instabilidade e a propagação multi-caminho (CABALLERO et al., 2006; COUTURIER; AKHLOUFI, 2021), podem aumentar a incerteza na estimativa de posição. Além disso, os VANTs podem ser sequestrados por sinais de GPS falsos (VISWANATHAN; PIRES; HUBER, 2016; CONTE; DOHERTY, 2008), representando um risco significativo à segurança do VANT e de pessoas próximas a ele.

Portanto, um sistema alternativo é necessário para fornecer uma fonte redundante de estimativa de pose. Os sistemas de localização baseados em visão computacional estão entre as soluções mais populares para atuar como sistema secundário de estimativa de pose quando o GPS falha. Não apenas as câmeras são mais portáteis e econômicas do que outras soluções, como os medidores de alcance a laser, mas as técnicas de localização visual também podem ser usadas em várias situações.

No entanto, a localização visual para VANTs é um problema desafiador por várias razões. As principais dificuldades surgem da comparação das leituras do sensor do VANT, geralmente imagens 2D, com o mapa do ambiente, geralmente uma imagem de satélite 2D. A imagem de satélite pode estar desatualizada em relação à data do voo, tornando as imagens do VANT e o mapa de satélite consideravelmente diferentes. Além disso, o voo do VANT ao ar livre durante a estimativa de pose pode levar a mudanças de iluminação em toda a sequência de imagens do VANT ou até mesmo efeitos de perspectiva em edifícios e estruturas altas.

Couturier e Akhloufi (2021) classificaram as técnicas de localização visual em categorias relativas e absolutas. Na primeira, estão incluídos métodos que comparam o quadro atual com o anterior. A odometria visual e o mapeamento e localização simultâneos, do inglês *simultaneous localization and mapping* (SLAM), são exemplos desses

métodos. Nos concentraremos na segunda categoria. Dentro da categoria de localização visual absoluta, Couturier e Akhloufi classificaram ainda os métodos em quatro grupos principais, sendo eles odometria visual, correspondência por *template*, correspondência por pontos de interesse e *Deep Learning*. Nosso interesse reside em técnicas que usam correspondência de pontos de características e *Deep Learning*.

No grupo de correspondência de pontos de característica, uma variedade de descritores são usados. Shan et al. (SHAN et al., 2015) extraíram recursos de histograma de gradiente orientado, do inglês *histogram of oriented gradient* (HOG), das imagens VANT e do mapa de referência e os usaram em combinação com um filtro de partículas para estimar a posição do VANT. No entanto, o fluxo óptico usado no método de Shan et al. requer dados de rotação e altitude do VANT. Para corresponder imagens de vista de rua e imagens de VANT, Majdik et al. (2015) usaram *bag of words* e pontos de interesse do tipo *scale-invariant feature transform* (SIFT), ou transformada de características invariantes à escala, mas devido a diferentes pontos de vista, quase 80% das correspondências de recursos eram outliers. Choi e Park (2020) usaram uma técnica de correspondência de modelos, mas ela exemplifica como mudanças de iluminação, como sombras, podem interferir no processo de localização. Ao mitigar o efeito de sombra, Choi e Park melhoraram seus resultados.

Mantelli et al. (2019) propuseram o método abBRIEF para corresponder mapas de imagens de satélite e imagens de VANT, como pode ser visto na Figura 1.1, que se mostrou extremamente robusto. Mantelli et al. desenvolveram um modelo de medição para o sistema de localização de Monte Carlo, *Monte Carlo localization* (MCL), usando o descritor abBRIEF, e seus experimentos mostraram que o sistema estimou corretamente a pose do VANT em múltiplos voos e mapas. No entanto, o método apresentou fraqueza em áreas homogêneas, como florestas e pastagens, devido à dependência do descritor abBRIEF da informação de cor para calcular a assinatura da imagem. Assim, em regiões sem textura, o sistema não pôde estimar eficientemente a pose do VANT.

Em contraste, um método de localização de VANTs menos afetado por variações de cor é o uso de informações semânticas, como prédios e estradas. Masselli, Hanten e Zell (2016) também usaram filtragem de partículas, mas classificaram o terreno do voo do VANT usando recursos ORB extraídos, que foram então classificados com uma abordagem de floresta aleatória. Embora seu trabalho tenha abordado a variação sazonal de vegetação e iluminação, os resultados mostraram uma alta chance de classificação incorreta para prédios e ruas.

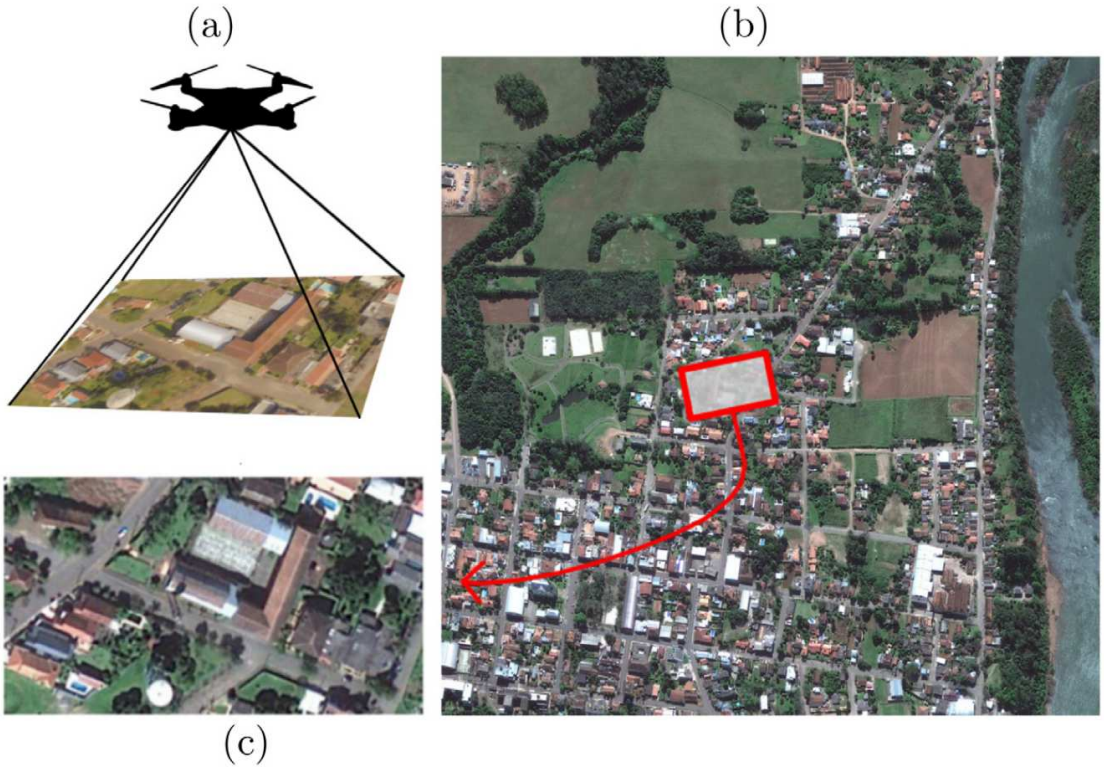


Figura 1.1 – Localização global de um VANT sobre mapa representado por imagem de satélite é feita comparando a imagem capturada pelo VANT, (a), com partes de tamanho equivalente, (c), retiradas da imagem de satélite, (b). A imagem (c) equivale ao recorte delimitado pelo retângulo vermelho em (b). Figura extraída de (MANTELLI et al., 2019).

Choi e Myung (2020) propuseram um sistema de localização visual de VANT que usa informações de construção obtidas por meio de segmentação de imagem e um conceito chamado razão de construção. O sistema proposto por Choi e Myung compara a imagem do VANT com *patches* do mapa, procurando possíveis correspondências até que haja uma convergência e a posição do VANT seja estimada. No entanto, a abordagem baseada na Relação de Construção simplificou demais as informações da imagem, dificultando a convergência em ambientes mais complexos. Além disso, os experimentos consideraram apenas voos em alturas fixas.

Por fim, há um número crescente de métodos de localização visual usando aprendizado profundo, seja como um *framework end-to-end* ou como uma ferramenta. Shetty e Gao (2019) usaram recursos de aprendizado profundo para comparar imagens de satélite com imagens de VANT, mesmo havendo diferenças nos pontos de vista. Li et al. (2021) usou um método de segmentação de estradas baseado em *U-Net* como ferramenta para geolocalização, embora não tenha sido especificamente projetado para localização de VANT.

Em resumo, existem vários métodos que abordam o problema de localização,

incluindo aqueles baseados em correspondência de características, *Deep Learning* e informações semânticas. Embora cada método tenha seus pontos fortes e fracos, todos eles buscam fornecer um sistema alternativo para a estimativa de posição redundante, o que é crucial para a operação segura e bem-sucedida de VANTs em várias aplicações. No entanto, ainda enfrentamos desafios significativos nessa área, especialmente em relação à comparação precisa entre as leituras dos sensores e os dados do mapa. Assim, este trabalho concentra-se em encontrar soluções para melhorar a precisão e confiabilidade da localização visual dos VANTs.

1.2 Objetivos e Contribuições

O principal objetivo do trabalho é investigar, no âmbito do problema de localização de VANTs sobre imagens de satélites, o uso de informação semântica associada a construções no ambiente, que tendem a ser mais estáveis quando comparadas a outros tipos de informação disponíveis no ambiente, como vegetação, corpos d'água, etc. Para tal, propõe-se um novo descritor binário, chamado NBD-BRIEF, que busca melhorar a precisão da localização de VANTs em áreas urbanas usando informação semântica. Esse descritor é baseado no descritor BRIEF e incorpora informações de proximidade de prédios, embora possa ser adaptado para outros tipos de objetos. A dissertação propõe um framework que é uma extensão do trabalho de Mantelli et al. (2019), que utilizou o abBRIEF para localização de VANTs.

Tanto o abBRIEF quanto o BRIEF são descritores binários, que são computados por um teste de diferença entre os valores de intensidade dos pixels de uma imagem. No entanto, devido à utilização de cores na descrição da imagem, o abBRIEF é suscetível a mudanças de luminosidade, mudanças na vegetação, estações do ano e outras variáveis. Para lidar com essa fragilidade, o novo descritor utiliza elementos que sofrem menos mudanças em períodos curtos e médios de tempo, como prédios, casas e outras construções.

Neste trabalho, propomos um novo sistema de localização de VANT fundamentado em visão computacional, que utiliza uma fonte de informação menos variável em comparação com as cores puras de uma imagem. A presença de construções em uma determinada região de uma área urbana tende a sofrer muito menos mudanças do que informações puramente visuais, como luz, cor ou vegetação de uma mesma área durante o dia, ao longo das estações ou ao longo dos anos. No entanto, conforme ilustrado na Figura 1.2b, a informação do que é ou não é um edifício normalmente não é muito descri-

(a) (b) (c)

Figura 1.2 – Informações sobre construções em áreas urbanas (**b**) tendem a ser uma fonte de informação mais confiável do que informações visuais puras, (**a**), que sofrem variações devido a mudanças de clima, estação do ano, etc. No entanto, tais informações acabam gerando grandes regiões homogêneas, aumentando indevidamente a similaridade entre diferentes regiões. Por outro lado, a proximidade da construção, ou seja, a distância ao edifício mais próximo, (**c**), é uma medida que varia suavemente ao longo do ambiente e pode ser utilizada para melhorar o processo de localização. Em (b), os edifícios são representados com pixels brancos. Em (c), quanto maior o valor de um pixel, mais distante ele está das bordas dos prédios.

tiva devido à sua natureza homogénea; assim, nosso método propõe o uso de uma medida de proximidade a edifícios, que varia gradativamente entre os pixels vizinhos, conforme mostrado na Figura 1.2c.

As principais contribuições do trabalho são:

- o desenvolvimento do descritor NBD-BRIEF, que utiliza *deep learning* para a segmentação dos prédios, permitindo que sejam selecionados apenas os pontos que estão próximos a eles;
- o desenvolvimento de um *framework* de localização de VANTs baseado no NBD-BRIEF, que utiliza imagens RGB e um mapa de referência com os contornos dos prédios para a localização precisa do VANT em áreas urbanas.

1.3 Organização

A estrutura desta dissertação é a seguinte. No Capítulo 2, apresentamos a fundamentação teórica do nosso trabalho na área de robótica móvel, que abrange conceitos essenciais e dificuldades encontradas na localização de robôs. No Capítulo 3, examinamos o problema de localização usando informações visuais, com enfoque no descritor BRIEF. O descritor BRIEF serviu como base para a construção do NBD-BRIEF. Também descrevemos a rede neural utilizada para segmentação das imagens. Esse capítulo também inclui uma revisão de métodos existentes propostos para a localização de VANTs. No Capítulo

4, apresentamos uma análise da importância de um bom modelo de observação para localização de VANTs. No Capítulo 5, é apresentado o descritor proposto, NBD-BRIEF: um descritor que utiliza imagens com informações de distância de edifícios próximos. Em seguida, é descrita a integração do descritor com o algoritmo MCL. O Capítulo 6 apresenta a validação experimental do método proposto. São descritos os equipamentos utilizados nos testes, incluindo as configurações e resultados obtidos. Também são realizadas comparações com outros algoritmos similares. Por fim, no Capítulo 7, apresentam-se as conclusões, bem como trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA EM ROBÓTICA MÓVEL

2.1 Definições e notações

Para que um robô móvel inteligente possa agir de forma autônoma, ele precisa ser capaz de resolver tarefas fundamentais, como se mover até o destino desejado (planejamento), conhecer o ambiente ao seu redor (mapeamento) e determinar sua posição no espaço (localização). Esses problemas, geralmente, precisam ser resolvidos simultaneamente, o que pode desencadear problemas mais complexos, como mostrado na Figura 2.1.

Figura 2.1 – Problemas fundamentais da robótica móvel e suas interseções. Figura adaptada de Makarenko et al. (2002).

Alguns desses problemas, como mapeamento, localização e SLAM, surgem porque o estado do sistema não é conhecido, o que significa que precisamos usar a percepção do robô - ou seja, os dados dos sensores - como fonte de informação para estimar o estado. Normalmente, as técnicas de estimativa de estado, como a filtragem Bayesiana, o filtro de Kalman e o filtro de partículas, são usadas para resolvê-los de maneira probabilística.

Outros problemas, como planejamento, exploração e localização ativa, também requerem uma abordagem eficiente para controlar o robô. Além disso, o SLAM ativo (Simultaneous Localization and Mapping ativo) é uma estratégia que combina técnicas de mapeamento e localização em tempo real com a capacidade do robô de tomar ações ativas para aprimorar sua estimativa de posição e o mapa construído. Essa abordagem envolve a seleção inteligente de movimentos e medições para obter informações mais relevantes e reduzir a incerteza durante o processo de mapeamento e localização. São problemas de

atuação[1]. Para encontrar a melhor solução, geralmente é necessário resolver um problema de otimização de uma função de custo, por exemplo, minimizar a distância percorrida pelo robô, maximizar a segurança do caminho percorrido ou maximizar a localizabilidade.

Um robô móvel autônomo tem como objetivo se mover em um ambiente para realizar suas tarefas. No entanto, para isso, é necessário que o robô saiba sua localização atual, bem como a localização de obstáculos próximos. Em cenários realistas, essa informação não é diretamente disponível e, portanto, o robô deve utilizar seus sensores, que fornecem informações parciais e ruidosas sobre o estado do ambiente (THRUN; BURGARD; FOX, 2005).

Na área de estimativa de estado em robótica móvel, há quatro variáveis[2] importantes a serem consideradas. A primeira delas é $\boldsymbol{x}_t$, que representa a pose do robô $(x, y, \theta)^T$ no instante t. Para nos referirmos às componentes de variáveis representadas por tuplas ou vetores, utilizamos a notação funcional *componente*(**variável**). Portanto, as componentes de $\boldsymbol{x}_t$ são referidas como $x(\boldsymbol{x}_t)$, $y(\boldsymbol{x}_t)$ e $\theta(\boldsymbol{x}_t)$. A trajetória seguida pelo robô é representada por $\boldsymbol{x}_{0:t} = \boldsymbol{x}_0, \boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_t$, onde $\boldsymbol{x}_0$ é a pose inicial do robô.

A segunda variável é $\boldsymbol{m}_i$, que representa a posição de um objeto i no ambiente. O mapa do ambiente é composto pelo vetor de todas as posições de objetos, ou seja, $\boldsymbol{m} = (\boldsymbol{m}_1, \boldsymbol{m}_2, \cdots, \boldsymbol{m}_N)^T$.

A terceira variável é $\boldsymbol{u}_t$, que representa o vetor de controle aplicado no instante $t-1$ que leva o robô à pose $\boldsymbol{x}_t$ no instante t. Geralmente, esse vetor de controle é dado pela medida de odometria entre os instantes $t-1$ e t. O histórico de comandos de controle é representado por $\boldsymbol{u}_{1:t} = \boldsymbol{u}_1, \boldsymbol{u}_2, \cdots, \boldsymbol{u}_t$.

Por fim, a quarta variável é z_t^i, que representa a i-ésima observação feita pelo robô no instante t. O vetor $z_t = (z_t^1, z_t^2, \cdots, z_t^K)^T$ é composto por todas as observações feitas pelo robô no instante t, enquanto o vetor $z_{1:t} = z_1, z_2, \cdots, z_t$ representa o histórico de todas as observações feitas pelo robô. Com essas quatro variáveis, é possível realizar a estimativa de estado em robótica móvel.

O conjunto de controles $\boldsymbol{u}_{1:t}$ que representam as ações realizadas pelo robô e as observações $z_{1:t}$ são sempre conhecidos. As ações são definidas com base em informações proprioceptivas, ou seja, medições de movimentos em relação a um quadro de referência interno do robô, como odometria em robôs terrestres equipados com rodas. As observações

[1]Embora os três últimos problemas citados (i.e. exploração, localização ativa e exploração integrada) envolvam tanto percepção quanto atuação, na maioria das vezes uma solução pronta é usada para a parte de percepção (i.e. solução de mapeamento no caso de exploração, solução de localização no caso de localização ativa e solução de SLAM no caso de exploração integrada) e o foco está na atuação.

[2]A notação utilizada neste trabalho é a definida por Thrun, Burgard e Fox (2005).

são informações exteroceptivas, ou seja, medições da disposição do ambiente e dos objetos em relação ao quadro de referência do robô, como imagens de câmera e medições de alcance de laser (MURPHY, 2000). O estado do robô $\boldsymbol{x}_{0:t}$ e o mapa $\boldsymbol{m}$ podem ser conhecidos ou desconhecidos. De fato, existem três problemas principais de estimativa de estado variando as informações conhecidas:

- *Localização*: estimar a pose do robô quando o mapa é conhecido;
- *Mapeamento*: estimar o mapa quando a pose do robô é conhecida;
- *SLAM*: estimar tanto a pose do robô quanto o mapa.

Neste trabalho, focaremos no problema de localização pois o mapa do ambiente da aplicação em questão é conhecido e pode ser obtido a partir de imagens de satélite. É importante notar que o mapa pode ter diferenças em relação às observações feitas pelo robô, no entanto, cabe aos métodos de localização tratar as informações usadas e associar incertezas a elas. Claro que quanto mais confiável é a informação disponível, mais provável é a obtenção de um bom resultado de localização.

2.2 Auto localização de robôs móveis

A tarefa de localização de robôs móveis envolve determinar a posição e orientação de um robô em relação a um mapa pré-existente do ambiente. Esse problema pode ser visto como um desafio de transformação de coordenadas, onde um sistema de coordenadas global, representando a posição de todos os objetos no ambiente, deve ser relacionado ao sistema de coordenadas local do robô (THRUN; BURGARD; FOX, 2005).

Conforme mostra a Figura 2.2, no problema de localização, é preciso estimar a pose do robô ao longo do tempo a partir das observações e das movimentações feitas. Como as observações são associadas a características conhecidas do ambiente, cabe estimar onde o robô deveria estar para que uma dada observação seja feita em relação aos obstáculos. O problema principal é considerar as incertezas associadas às observações feitas e à movimentação do robô, uma vez que ambos os aspectos não são perfeitos.

Existem dois tipos de problemas de localização dependendo do conhecimento disponível no início e durante a operação do robô. Na **localização local** ou *rastreamento de posição*, a pose inicial do robô é conhecida, e a abordagem de localização deve levar em conta apenas a incerteza no movimento do robô, geralmente modelada com uma distribuição Gaussiana. A **localização global**, por outro lado, é mais desafiadora e tem

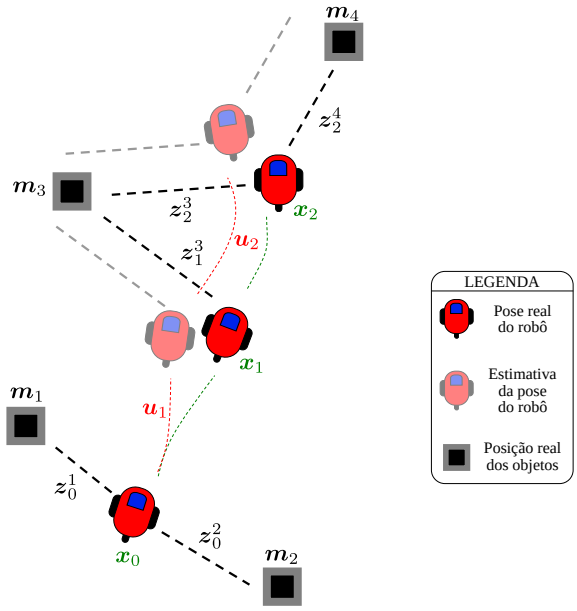


Figura 2.2 – O problema de Localização consiste no problema de estimativa de estado do robô, $\boldsymbol{x}_t$, em que a estimativa de localização prevista após a movimentação $\boldsymbol{u}_t$ a partir do ponto $\boldsymbol{x}_{t-1}$ apresenta erros, uma vez que a execução de tal ação nunca é perfeita. No entanto, como o mapa é conhecido, composto por objetos $\boldsymbol{m}_i$, as observações destes objetos, z_t^i, podem ser usadas para corrigir a estimativa de localização. Um dos desafios do problema é balancear as incertezas associadas à movimentação e às observações. Figura adaptada de Maffei (2017).

duas variações: o *problema do robô despertado* (normalmente só chamado de problema de localização global), em que a pose inicial do robô é desconhecida e requer uma distribuição de probabilidade multimodal para modelar a incerteza de localização; e o *problema do robô sequestrado*, em que um robô bem localizado é transportado repentinamente para uma localização diferente, testando a capacidade da estratégia de localização de se recuperar de falhas, uma vez que o robô pode acreditar que conhece sua posição atual, mas na realidade, não conhece. Neste trabalho, focamos no problema de localização global, em que não conhecemos a pose inicial.

O modelo gráfico para o problema de localização global, ilustrado na Figura 2.3, mostra a interdependência entre as variáveis e as relações causais entre elas. Nesse contexto, o objetivo é estimar a sequência de poses do robô $\boldsymbol{x}_{0:t}$ a partir das medidas $\boldsymbol{z}_{1:t}$, controles $\boldsymbol{u}_{1:t}$ e mapa $\boldsymbol{m}$. A Equação 2.1 representa a crença sobre a pose do robô no tempo t, que corresponde à distribuição de probabilidade posterior sobre o estado do robô dados o mapa e todas as medidas e controles passados:

$$bel(\boldsymbol{x}_t) = p(\boldsymbol{x}_t | \boldsymbol{u}_{1:t}, \boldsymbol{z}_{1:t}, \boldsymbol{m}). \tag{2.1}$$

O método clássico para resolver o problema de localização é através da localização

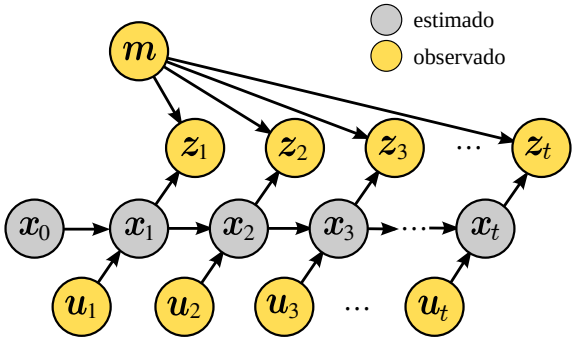


Figura 2.3 – Modelo gráfico do problema de Localização. A pose do robô ao longo do tempo, $\boldsymbol{x}_{0:t}$, precisa ser estimada em função do mapa, $\boldsymbol{m}$; das observações feitas pelo robô, $z_{1:t}$; e das ações tomadas por ele, $\boldsymbol{u}_{1:t}$. Pela premissa de Markov uma dada pose $\boldsymbol{x}_i$ só depende da pose anterior $\boldsymbol{x}_{i-1}$ e da última ação correspondente $\boldsymbol{u}_i$, e implica somente na observação daquele instante z_i. Figura adaptada de Thrun, Burgard e Fox (2005).

de Markov, aplicando o filtro de Bayes à crença da (2.1). O filtro de Bayes adota a Premissa de Markov, ilustrada na Figura 2.3, que implica que a crença anterior no tempo $t-1$ é suficiente para representar a história do robô até aquele momento. No entanto, é essencial observar que tal premissa é apenas uma aproximação na robótica, pois dinâmicas não modeladas e imprecisões podem ocorrer. Apesar disso, abordagens baseadas em filtragem Bayesiana têm se mostrado robustos na prática Thrun, Burgard e Fox (2005).

O filtro de Bayes fornece uma solução para $bel(\boldsymbol{x}_t)$ conforme dado por

$$bel(\boldsymbol{x}_t) = \eta\ p(z_t|\boldsymbol{x}_t,\boldsymbol{m}) \int p(\boldsymbol{x}_t|\boldsymbol{u}_t,\boldsymbol{x}_{t-1})\ bel(\boldsymbol{x}_{t-1})\ d\boldsymbol{x}_{t-1}, \tag{2.2}$$

onde $p(\boldsymbol{x}_t|\boldsymbol{u}_t,\boldsymbol{x}_{t-1})$ representa o modelo de movimento do robô, $p(z_t|\boldsymbol{x}_t,\boldsymbol{m})$ representa o modelo de observação dos sensores do robô, e η é um fator de normalização.

Na prática, separa-se a equação anterior em duas equações recursivas: uma de predição e uma de correção do estado. Na **predição**, estima-se o estado atual aplicando-se a última ação, $\boldsymbol{u}_t$, sobre o estado anterior, $\boldsymbol{x}_{t-1}$, mas sem utilizar a última observação. Essa distribuição é definida como $\overline{bel(\boldsymbol{x}_t)} = p(\boldsymbol{x}_t|\boldsymbol{u}_{1:t}, z_{1:t-1}, \boldsymbol{m})$, e note como ela não considera z_t:

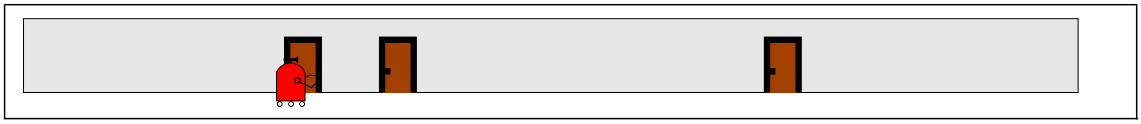
$$\overline{bel(\boldsymbol{x}_t)} = \int p(\boldsymbol{x}_t|\boldsymbol{u}_t,\boldsymbol{x}_{t-1})\ bel(\boldsymbol{x}_{t-1})\ d\boldsymbol{x}_{t-1}. \tag{2.3}$$

Na etapa de **correção**, corrige-se a estimativa feita aplicando a observação atual z_t:

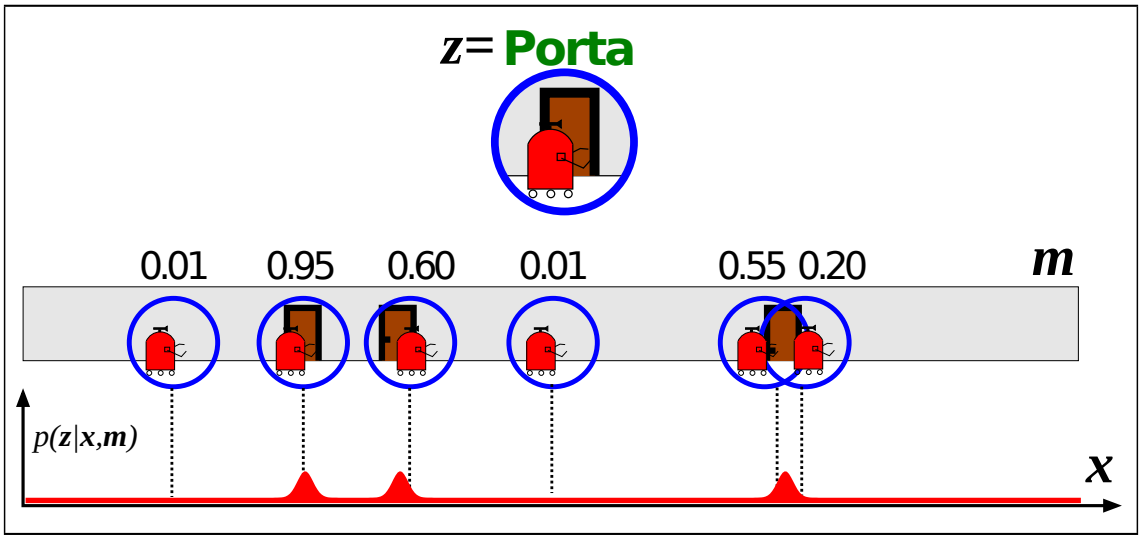
$$bel(\boldsymbol{x}_t) = \eta\ p(z_t|\boldsymbol{x}_t,\boldsymbol{m}).\overline{bel(\boldsymbol{x}_t)} \tag{2.4}$$

As Figuras 2.4 e 2.5 mostram, respectivamente, um exemplo simples de modelo

de observação e um exemplo correspondente de localização para ilustrar o processo de localização de Markov (THRUN; BURGARD; FOX, 2005). O exemplo mostra um robô que pode se mover para frente ou para trás em um ambiente de um corredor contendo portas (Figura 2.4a). A única observação que o robô pode fazer é observar ou não uma porta quando se encontra na frente de uma (Figura 2.4b). O modelo de observação baseado neste mapa diz que quanto mais próximo o robô estiver de uma posição contendo porta, maior a probabilidade de observar uma porta.



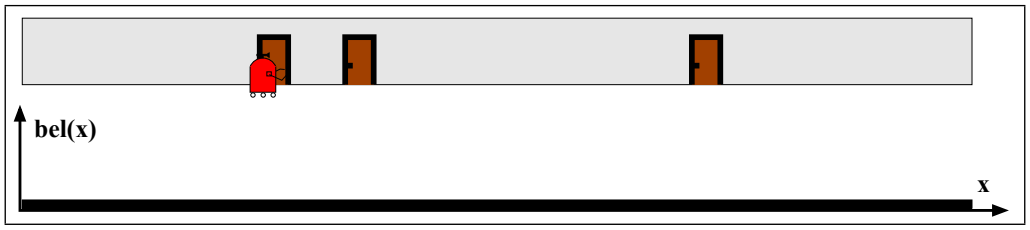
(a) Cenário de exemplo onde o mapa contém três portas em posições conhecidas.



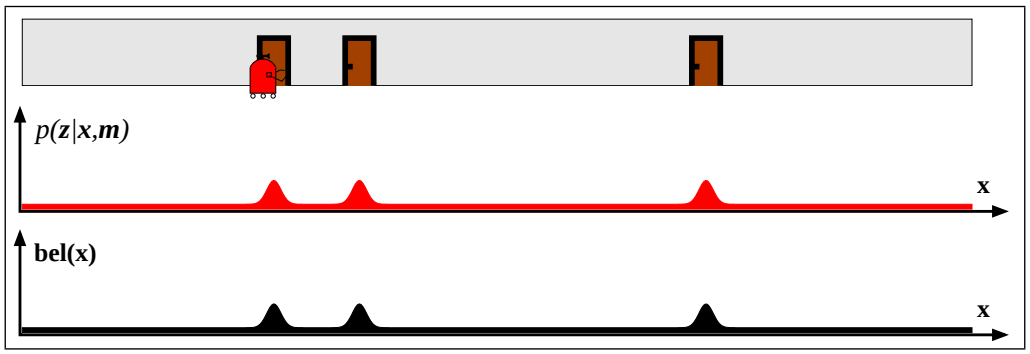
(b) Modelo de observação do robô neste exemplo, supondo que ele observa estar em frente a uma porta. Locais do mapa onde existem portas possuem chance maior de representarem à real posição do robô.

Figura 2.4 – Exemplo de modelo de observação de um robô movendo-se em um ambiente simples contendo portas. Figura adaptada de Thrun, Burgard e Fox (2005).

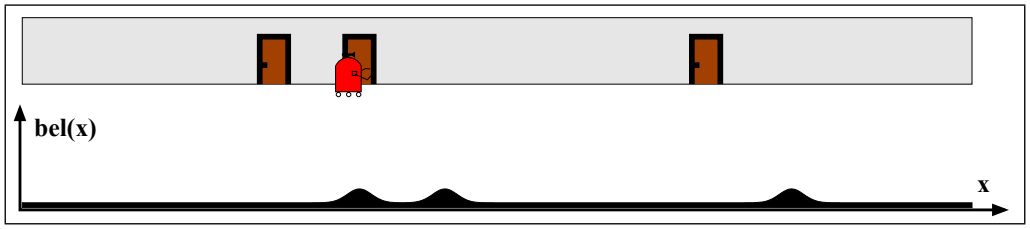
No início do processo de localização global, não sabemos a posição onde o robô se encontra logo todas as posições são igualmente prováveis (Figura 2.5a). Neste ponto, qualquer movimentação que seja feita não alterará as estimativas pois o robô continuará podendo estar em qualquer lugar. No entanto, se uma observação é feita e o robô enxerga uma porta (Figura 2.5b), uma correção do estado pode ser feita. Na sequência o robô se move e a estimativa do estado também se desloca (Figura 2.5c). Devido à incerteza de movimentação, a variância das estimativas aumentam. Porém ao observar novamente uma porta, uma nova correção pode ser feita (Figura 2.5d). Ao fim, quando o robô se move novamente, o filtro já convergiu para a localização correta (Figura 2.5e).



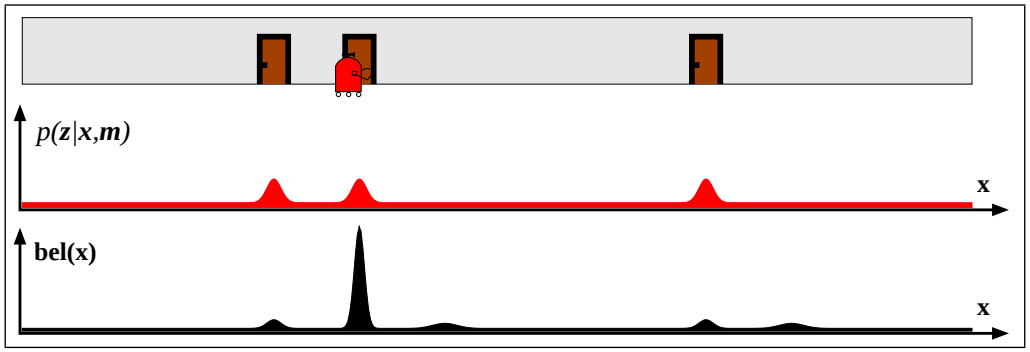
(a) Estimativa de localização inicial



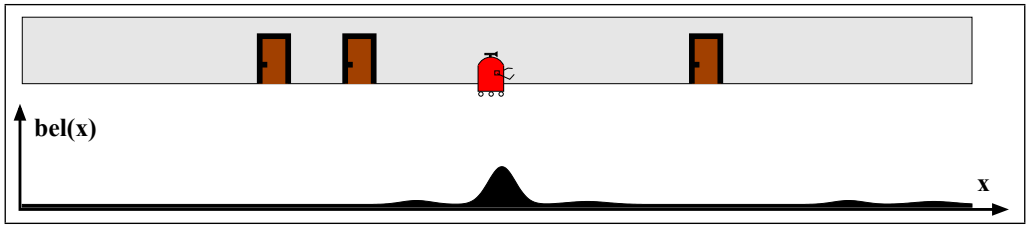
(b) Correção do estado usando modelo de observação.



(c) Nova predição do estado após movimentação para a direita.



(d) Nova correção do estado usando modelo de observação.



(e) Predição do estado após nova movimentação para a direita.

Figura 2.5 – Localização Markoviana: exemplo simplificado de localização de um robô móvel se deslocando em um corredor e realizando observações de existência ou não de portas ao seu redor. Figura extraída de Thrun, Burgard e Fox (2005).

Em resumo, o processo de localização de Markov é uma solução recursiva de predição e correção, conforme mostra a Figura 2.6. Sempre que uma predição ocorre baseada na última movimentação feita, aplica-se o modelo de movimento do robô, e a incerteza do estado tende a crescer devido à incerteza de movimento. No melhor caso, a incerteza se mantém igual caso a movimentação seja perfeita. Por outro lado, sempre que uma correção é feita baseada nas observações, aplica-se o modelo de observação do robô, e a incerteza tende a cair, pois se está usando as informações contidas no mapa para corrigir o estado. No pior caso, se a observação for muito imprecisa ou não existir, a incerteza se mantém igual uma vez que nenhuma correção é feita.

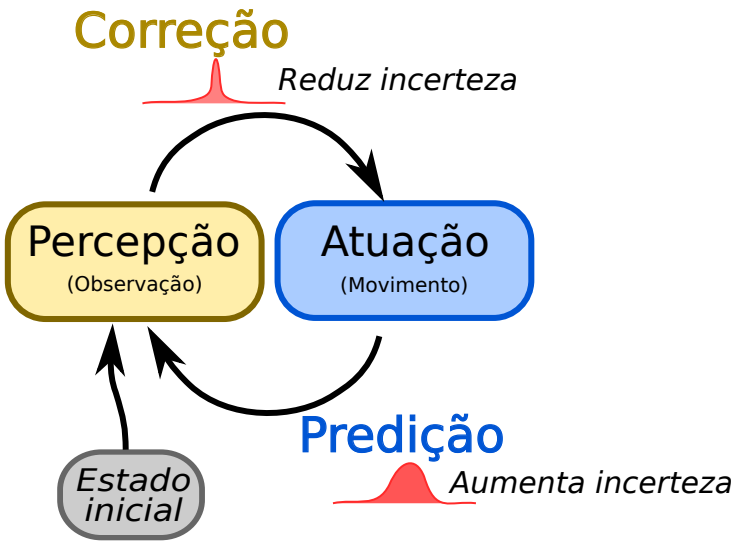


Figura 2.6 – Localização Markoviana é solucionada de forma cíclica em um processo recorrente de predição baseada na movimentação do robô e correção baseada nas observações feitas pelo robô. Figura adaptada de Maffei (2022).

Sendo assim, as implementações mais comumente usadas para a localização de Markov são os filtros de Kalman (LEONARD; DURRANT-WHYTE, 1991), que dependem de modelos de movimento lineares e assumem ruídos gaussianos. Por outro lado, temos os filtros baseados em grade ou histogramas (BURGARD et al., 1998), que utilizam um espaço de estados discretizado para lidar com distribuições multimodais, e os filtros de partículas (DELLAERT et al., 1999), que representam a distribuição posterior usando um conjunto ponderado de amostras (partículas). A abordagem de filtro de partículas, também conhecida como Localização de Monte Carlo (MCL), provou ser um método altamente eficaz para a localização de robôs e é usada neste estudo.

2.2.1 Localização de Monte Carlo - Filtro de partículas

O algoritmo de Localização de Monte Carlo (MCL) foi introduzido por Dellaert et al. (1999) como uma abordagem não-paramétrica para o filtro Bayesiano. O método aproxima a posteriori $p(\boldsymbol{x}_t|\boldsymbol{u}_{1:t}, \boldsymbol{z}_{1:t}, \boldsymbol{m})$ utilizando um conjunto de M partículas, como mostrado em

$$\mathcal{X}_t = \{\boldsymbol{p}_t^{[1]}, \boldsymbol{p}_t^{[2]}, \cdots, \boldsymbol{p}_t^{[M]}\}, \tag{2.5}$$

onde cada partícula $\boldsymbol{p}_t^{[m]} = \langle \boldsymbol{x}, w \rangle$ corresponde a uma pose $\boldsymbol{x}$ no tempo t e tem um peso de importância associado w.

Algorithm 2.1: Localização de Monte Carlo

Input: $\mathcal{X}_{t-1}, \boldsymbol{u}_t, \boldsymbol{z}_t, \boldsymbol{m}$
Output: $\mathcal{X}_t$

1 $\bar{\mathcal{X}}_t = \mathcal{X}_t = \emptyset$
2 **for** m *in* $1 \ldots M$ **do**
3 $\quad \boldsymbol{x}_{t-1} = \boldsymbol{x}(\boldsymbol{p}_{t-1}^{[m]})$
4 $\quad$ amostra $\boldsymbol{x} \propto p(\boldsymbol{x}_t \,|\, \boldsymbol{u}_t, \boldsymbol{x}_{t-1})$
5 $\quad w = p(\boldsymbol{z}_t \,|\, \boldsymbol{x}, \boldsymbol{m})$
6 $\quad \boldsymbol{p}_t^{[m]} = \langle \boldsymbol{x}, w \rangle$
7 $\quad \bar{\mathcal{X}}_t = \bar{\mathcal{X}}_t \cup \{\boldsymbol{p}_t^{[m]}\}$
8 **for** m *in* $1 \ldots M$ **do**
9 $\quad$ sorteia $\boldsymbol{p}_t^{[i]}$ de $\bar{\mathcal{X}}_t$ com probabilidade $\propto w(\boldsymbol{p}_t^{[i]})$
10 $\quad \mathcal{X}_t = \mathcal{X}_t \cup \{\langle \boldsymbol{x}_t^{[i]}, 1/M \rangle\}$
11 **return** $\mathcal{X}_t$

O Algoritmo 2.1 descreve o processo MCL, que estima recursivamente o conjunto de partículas $\mathcal{X}_t$ a partir do conjunto anterior $\mathcal{X}_{t-1}$ usando um processo de Amostragem-Importância-Reamostragem (*Sampling-Importance-Resampling*, SIR). O primeiro passo envolve a **amostragem** (linhas 3-4), onde cada partícula é propagada usando o modelo de movimento $p(\boldsymbol{x}_t, |, \boldsymbol{u}_t, \boldsymbol{x}_{t-1})$ em relação à pose anterior da partícula $\boldsymbol{x}(\boldsymbol{p}_{t-1}^{[m]})$.

No segundo passo, é realizada a **ponderação por importância** (linha 5), atribuindo pesos individuais a cada partícula com base no modelo de observação. A ideia é comparar as medidas reais do sensor com as medidas do sensor estimadas pela partícula, calculando a similaridade entre a distribuição alvo $p(\boldsymbol{x}_t, |, \boldsymbol{z}_{1:t}, \boldsymbol{u}_{1:t}, \boldsymbol{m})$ e a distribuição proposta $p(\boldsymbol{x}_t, |, \boldsymbol{z}_{1:t-1}, \boldsymbol{u}_{1:t}, \boldsymbol{m})$ obtida após a amostragem. As partículas propagadas e ponderadas compõem um conjunto temporário de partículas $\bar{\mathcal{X}}_t$ (linhas 6-7).

No terceiro passo, é realizada a **reamostragem** (linhas 9-10), onde a mesma quan-

tidade de M partículas é selecionada aleatoriamente com reposição de $\bar{X}_t$ e adicionada a um novo conjunto X_t. Uma técnica simples como o algoritmo da roleta é comumente usada para selecionar amostras e definir a probabilidade de selecionar cada partícula $\boldsymbol{p}_t^{[m]}$ proporcionalmente ao seu peso $w(\boldsymbol{p}_t^{[m]})$. A reamostragem é crítica para o filtro de partículas, pois aproxima a distribuição de partículas para a verdadeira posterior $p(\boldsymbol{x}_t \mid \boldsymbol{z}_{1:t}, \boldsymbol{u}_{1:t}, \boldsymbol{m})$. Durante a reamostragem, as partículas com pesos maiores têm mais probabilidade de serem replicadas, enquanto as com pesos menores tendem a ser descartadas, levando à convergência do filtro.

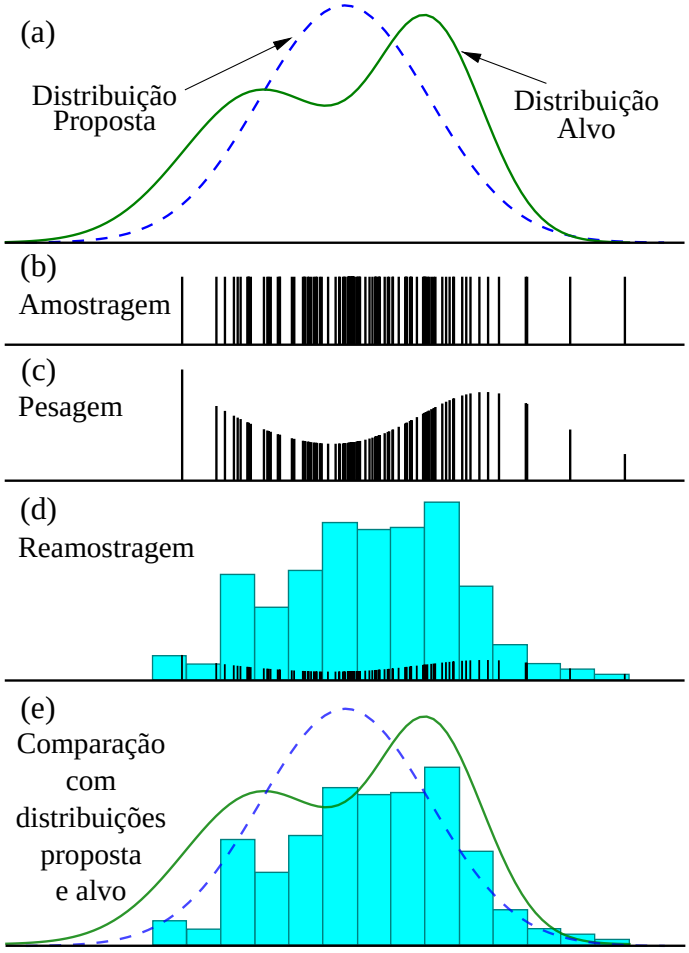
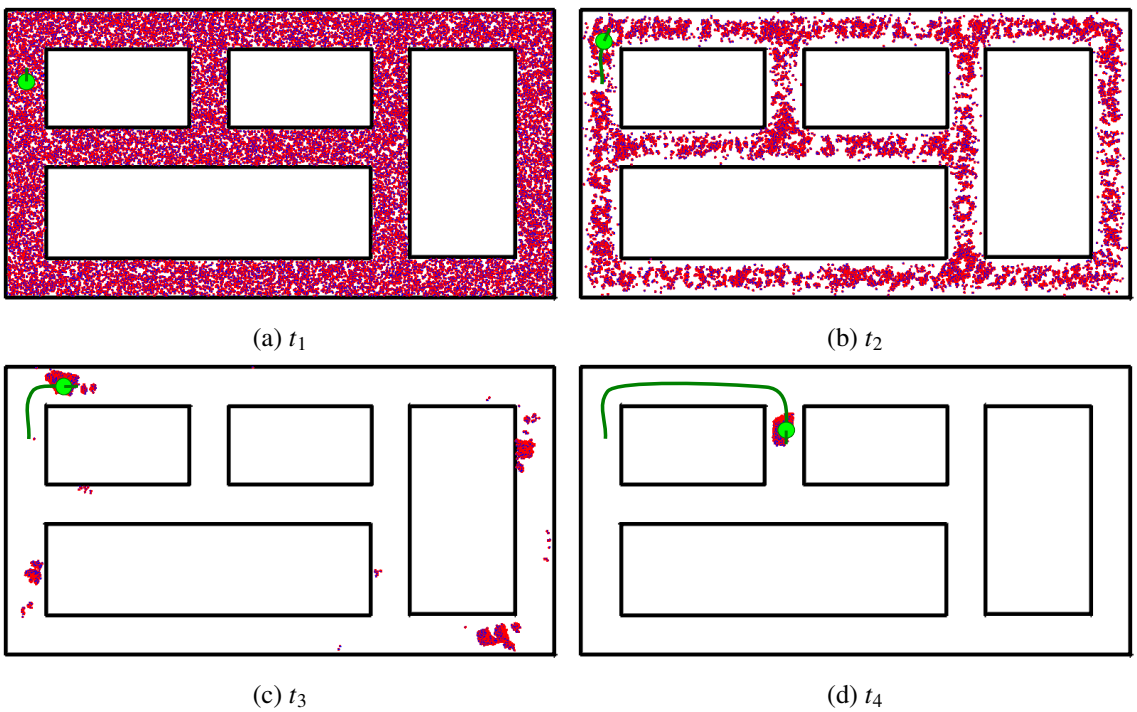


Figura 2.7 – Exemplo de amostragem, pesagem e reamostragem em um filtro de partículas. (a) Diferença entre as distribuições proposta e alvo. (b) Amostragem. (c) Pesagem. (d) Reamostragem. (e) Comparação da distribuição resultante com as distribuições proposta e alvo. Figura adaptada de Montemerlo e Thrun (2007).

Uma representação visual na Figura 2.7 demonstra a importância do passo de reamostragem no MCL. A distribuição-alvo (linha verde sólida) em (a) é multimodal, enquanto a distribuição proposta (linha pontilhada azul) é uma Gaussiana simples. Durante o passo de amostragem em (b), partículas são geradas com base na distribuição proposta. Consequentemente, as amostras obtidas (linhas verticais pretas) precisam ser ponderadas

para se aproximarem da distribuição-alvo. O processo de ponderação em (c) atribui pesos maiores (linhas pretas mais altas) às amostras de regiões onde a distribuição proposta é subestimada em comparação com a distribuição-alvo. Por outro lado, regiões onde a distribuição proposta é superestimada recebem pesos menores. Durante a reamostragem em (d), partículas são sorteadas de forma proporcional à a seus pesos (ou seja, a altura do recipiente correspondente no histograma azul de pesos). Por fim, a distribuição resultante (histograma azul) em (e) deve se aproximar da distribuição-alvo (linha verde). À medida que o número de partículas aumenta, o filtro de partículas se aproxima cada vez mais da distribuição-alvo.

Por fim, deve-se notar que o Algoritmo 2.1 descreve o processo de localização durante um único passo da trajetória do robô, ou seja, o movimento do instante $t-1$ para o instante t. Essas etapas são repetidas a cada instante, usando os valores atuais de ação e observações. Um exemplo completo do MCL é ilustrado na Figura 2.8. Inicialmente, em (a), partículas são geradas em todos os espaços livres, porque o robô pode estar em qualquer lugar no início do processo. À medida que o robô se move, a incerteza diminui. No entanto, após um pequeno deslocamento, mostrado em (b), o robô ainda não tem ideia de sua localização. Em (c), o robô vira à direita em uma esquina e a incerteza cai significativamente, porque há apenas quatro esquinas no ambiente. Finalmente, em (d), o



(a) t_1 (b) t_2

(c) t_3 (d) t_4

Figura 2.8 – Exemplo de convergência do filtro de partículas na Localização de Monte Carlo, com as partículas mostradas em rosa, e o robô e sua trajetória em verde. As partículas convergem quando o robô se move o suficiente para resolver as ambiguidades no ambiente.

robô vira novamente à direita e as últimas ambiguidades são resolvidas.

2.3 Modelos de observação

Os modelos de observação descrevem o processo de formação pelo qual as medições do sensor são geradas no mundo físico (THRUN; BURGARD; FOX, 2005). Ou seja, dado um mapa conhecido, $\boldsymbol{m}$, e uma pose $\boldsymbol{x}_t$ contida nele, o modelo de observação pondera como devem ser as observações tomadas daquele local. Conforme dito anteriormente, o modelo de observação é definido como uma distribuição de probabilidade condicional $p(z_t \mid \boldsymbol{x}_t, \boldsymbol{m})$. Quanto mais parecida uma medição z_t é da observação esperada a partir da pose $\boldsymbol{x}_t$ dado o mapa $\boldsymbol{m}$, mais alta é a probabilidade de z_t corresponder à observação daquele local. Portanto, mais provável de o robô estar na posição correta.

Cabe ao modelo de observação definir probabilisticamente o ruído nas medições do sensor em função de incertezas. Como regra geral, quanto mais preciso o modelo de sensor, melhores são os resultados. Na prática, no entanto, muitas vezes é impossível modelar um sensor com precisão. Além disso, ao se modelar o processo de medição como uma densidade de probabilidade condicional, $p(z_t \mid \boldsymbol{x}_t, \boldsymbol{m})$, em vez de uma função determinística $z_t = f(\boldsymbol{x}_t, \boldsymbol{m})$, a incerteza no modelo de sensor pode ser acomodada nos aspectos não determinísticos do modelo (THRUN; BURGARD; FOX, 2005).

Outro ponto importante é que modelos altamente precisos tendem a ser mais custosos computacionalmente do que modelos simples menos precisos. Entretanto, muitas vezes o segundo tipo de modelo é mais indicado para aplicações de alto grau de incerteza inicial, como no problema de localização global, pois esse tipo de situação exige um número muito alto de checagem de diferentes possibilidades de locais (MAFFEI et al., 2015).

Os robôs móveis atuais usam uma variedade de modalidades de sensores diferentes, como sensores de alcance ou câmeras. As especificidades do modelo de observação dependem do sensor, mas após modelá-los adequadamente é possível aplicá-los sem maiores dificuldades em técnicas como o filtro de partículas do método MCL. Neste trabalho, desenvolvemos um modelo de observação baseado em informação visual obtida por câmeras e portanto a seguir discutiremos mais sobre este assunto.

3 LOCALIZAÇÃO USANDO IMAGENS

3.1 Modelos de observação baseados em imagens e relação com casamento de *features*

Os modelos de observação são ferramentas que auxiliam na avaliação da qualidade de uma observação em relação ao ambiente em que ela ocorre. Comumente, técnicas de comparação são utilizadas, como a análise de imagens associadas às observações e partes do mapa. As técnicas de comparação de imagens não são apenas úteis para a avaliação da qualidade de observações em relação ao ambiente, mas também para o casamento de imagens. Esse processo é baseado na detecção e descrição de *features* distintivas em posições específicas das imagens, visando identificar correspondências entre elas. Quando as imagens compartilham características semelhantes, a correspondência de *features* tende a ocorrer com sucesso.

A detecção e correspondência de características são componentes essenciais de muitas aplicações de visão computacional. O primeiro tipo de característica que pode ser observado são locais específicos nas imagens, como cantos de prédios. Esses tipos de características localizadas são frequentemente chamados de pontos-chave ou pontos de interesse (ou até mesmo cantos) e são frequentemente descritos pela aparência de *patches* de pixels ao redor da localização do ponto. Outra classe importante de características são as bordas, que podem ser correspondidas com base em sua orientação e aparência local (perfis de borda) e também podem ser bons indicadores de limites de objetos e eventos de oclusão (SZELISKI, 2010).

Duas etapas importantes do *pipeline* de detecção e correspondência de *keypoints* são a detecção e a descrição de *features*. Durante a etapa de detecção de *features* (extração), cada imagem é examinada em busca de locais que provavelmente serão bem correspondidos em outras imagens. Na etapa de descrição de *features*, cada região ao redor das localizações de *keypoints* detectados é convertida em um descritor mais compacto e estável (invariante) que pode ser comparado com outros descritores.

Descritores de *features* são hoje fundamentais para muitas tecnologias que utilizam visão computacional, tais como SLAM visual, reconhecimento de objetos, reconstrução 3D, recuperação de imagem e localização de câmeras. Como essas tecnologias precisam lidar com cada vez mais dados ou executar em dispositivos móveis com recursos computacionais limitados, é importante que os descritores locais sejam rápidos para calcular, rápidos para serem correspondidos e eficientes em termos de memória (CALONDER et

al., 2010).

O detector e descritor de pontos-chave SIFT (LOWE, 2004) provou ser bem-sucedido em diversas aplicações que usam recursos visuais. O descritor realiza a detecção de características de escala-espacial construindo uma pirâmide de diferença de Gaussianas e, em seguida, procurando máximos no volume 3D resultante. Embora o SIFT funcione muito bem, ele impõe uma grande carga computacional, especialmente para sistemas em tempo real, como odometria visual, ou para dispositivos de baixa potência, como telefones celulares. Isso levou a uma busca intensiva por substitutos com menor custo computacional, como o descritor SURF (BAY; TUYTELAARS; GOOL, 2006).

Nos últimos anos, ORB (*Oriented FAST and Rotated BRIEF*) (RUBLEE et al., 2011) tornou-se bastante popular, especialmente após a introdução da técnica de SLAM visual chamada ORB-SLAM (MUR-ARTAL; MONTIEL; TARDÓS, 2015). ORB é um detector e descritor de características que se baseia no detector de *keypoints* FAST (ROSTEN; DRUMMOND, 2006) e no descritor BRIEF (CALONDER et al., 2010). O FAST é amplamente utilizado devido às suas propriedades computacionais, ou seja, é muito mais rápido do que outros detectores, já que leva apenas um parâmetro, o limiar de intensidade entre o pixel central e aqueles em um anel circular em torno do centro. O BRIEF é um descritor de características que usa testes binários simples entre pixels em um patch de imagem suavizado. Seu desempenho é semelhante ao do SIFT em muitos aspectos, incluindo robustez à iluminação, desfoque e distorção de perspectiva. No entanto, ele é muito sensível à rotação de plano.

Essa característica do BRIEF é um problema adicional para encontrar casamentos de *features* com diferentes orientações, no entanto é uma característica interessante para a construção de um modelo de observação conforme mostra o trabalho de. Mantelli et al. (2019) descrito na seção a seguir.

3.2 Usando o descritor abBRIEF para localização de VANTs

Em (MANTELLI et al., 2019), os autores propõem uma nova abordagem para resolver o problema de localização global de Veículos Aéreos Não Tripulados (VANTs) em imagens de satélite. Para isso, os autores propõem uma adaptação do descritor BRIEF para esse tipo de aplicação. Diferentemente de outras abordagens que aplicam o descritor em *patches* de imagens, a proposta do autor é aplicá-lo diretamente na imagem inteira. Essa abordagem tem a vantagem de eliminar a necessidade de usar um detector para

selecionar bons locais dentro da imagem. Em vez disso, o descritor BRIEF é aplicado em toda a imagem, gerando um descritor global que representa as características da imagem.

Além disso, a adaptação do descritor BRIEF apresentada pelo autor tem outra característica interessante. Apesar de não ser invariante a rotações, essa característica do descritor pode ser útil para diferenciar regiões observadas com orientações diferentes e, consequentemente, filtrar matchings errados. Isso é especialmente importante em aplicações de localização de VANTs, onde as imagens podem apresentar diferentes orientações. Ao usar o descritor BRIEF como um descritor global, é possível capturar as características da imagem de forma mais robusta, independentemente da orientação da imagem.

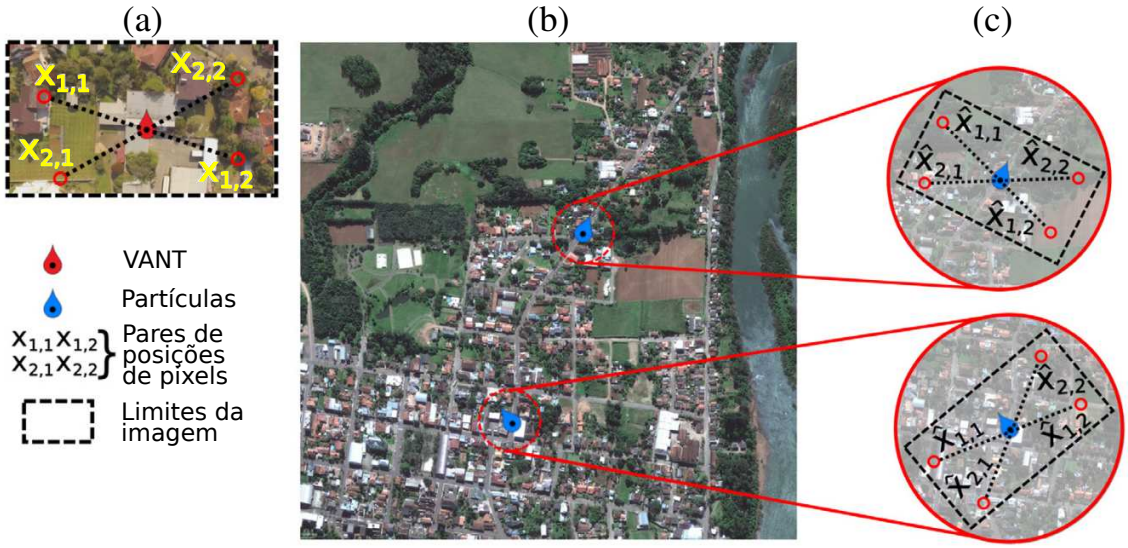


Figura 3.1 – Exemplo de uso do modelo de observação com abBRIEF. (a) Imagem I_t capturada pelo VANT, com os pares de pixels selecionados, K_t; (b) Imagem de satélite destacando duas partículas na mesma altitude mas com posições e orientações diferentes; e (c) duas partículas com suas respectivas imagens esperadas, $\hat{I}_t$, e pares de posições de pixels, $\hat{K}_t$. Figura adaptada de Mantelli et al. (2019).

O descritor abBRIEF é calculado seguindo um conjunto de etapas. Em primeiro lugar, a imagem é quantizada em n níveis para reduzir o ruído e garantir um descritor robusto mesmo quando as cores não são muito semelhantes. Em seguida, pares de pontos são distribuídos aleatoriamente pela imagem usando a mesma distribuição Gaussiana que o BRIEF. Cada ponto representa a posição de um pixel, resultando em um par de posições de pixel, como mostrado na Figura 3.1. O descritor binário é formado por um vetor binário, onde cada elemento do vetor armazena o resultado do teste binário computado usando a intensidade de cada par de pixels e canal de cor c. A similaridade entre duas imagens é medida usando a distância de Hamming computada sobre o descritor de ambas as imagens. O uso do descritor abBRIEF no problema de localizar VANTs em imagens

de satélite evita a necessidade de aplicar o descritor em patches de imagens, permitindo a aplicação do descritor em toda a imagem, tornando-o computacionalmente eficiente. O fato de o descritor não ser invariante à rotação o torna útil para distinguir regiões observadas com diferentes orientações e filtrar correspondências incorretas.

3.3 Segmentação de imagens usando *Deep Learning*

A segmentação de imagens é um processo essencial na área de visão computacional que consiste em dividir uma imagem em diferentes regiões ou segmentos, com o objetivo de identificar objetos, áreas de interesse ou características específicas.

Para realizar a segmentação de imagens, técnicas de aprendizado de máquina, como redes neurais convolucionais (CNNs), podem ser utilizadas. As CNNs têm a capacidade de aprender a reconhecer padrões e características nas imagens e, com base nessas informações, segmentar a imagem em diferentes regiões.

A U-Net (RONNEBERGER; FISCHER; BROX, 2015) é uma arquitetura de CNN desenvolvida para segmentação de imagens, especialmente em imagens médicas, onde é necessário segmentar estruturas complexas, como células, vasos sanguíneos e tumores. Além disso, a U-Net também pode ser adaptada para outras tarefas, como detecção de objetos, super-resolução de imagens e processamento de áudio.

A arquitetura da U-Net é dividida em duas partes principais: o *encoder* e o *decoder*. O *encoder* é responsável por capturar as características mais importantes da imagem, enquanto o *decoder* é responsável por transformar essas características em uma imagem segmentada. Essa arquitetura é simétrica e tem a forma de 'U', como mostrado na Figura 3.2, o que lhe deu o nome.

Uma característica importante da U-Net é o uso de conexões *skip*, que permitem a transmissão de informações de alto nível do *encoder* diretamente para o *decoder*, evitando a perda de informações importantes da imagem durante o processo de *downsampling* do *encoder*.

3.4 Trabalhos relacionados

A localização de VANTs é uma tarefa desafiadora devido à falta de sensores precisos e confiáveis para determinar a posição e orientação. Várias abordagens foram

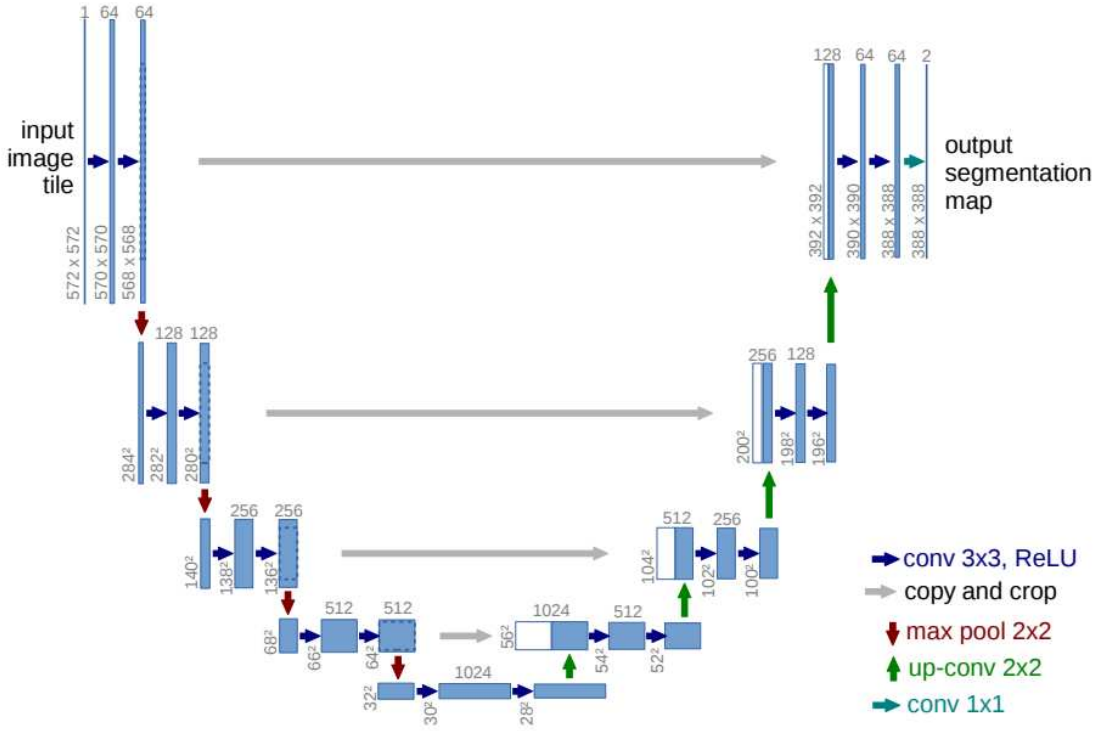


Figura 3.2 – Arquitetura da U-Net. Cada caixa azul representa um mapa de *features* com a quantidade de canais indicada no topo de cada caixa e o tamanho no lado esquerdo. As flechas indicam os diferentes tipos de operações. Figura extraída de (RONNEBERGER; FISCHER; BROX, 2015)

propostas para abordar esse problema, que vão desde o uso de GPS e unidades de medição inercial até técnicas de visão computacional.

A proposta de Yol et al. (2014) para localização absoluta baseada em visão para VANTs utiliza imagens aéreas georreferenciadas e uma câmera monocular montada no VANT para estimar sua posição. O método emprega um método de registro de modelo diferencial para calcular a transformação entre a imagem georreferenciada e a imagem atual capturada pela câmera. Os resultados indicam que o método é robusto e preciso na estimativa do movimento do VANT, mesmo em condições meteorológicas adversas, mas seu processamento é dispendioso.

Patel, Barfoot e Schoellig (2020) propõem um método para estimar a pose global de um VANT em ambientes sem GPS, utilizando imagens pré-renderizadas do Google Earth. O método emprega uma técnica densa de informação mútua, redes neurais convolucionais e odometria visual para localização precisa de imagens reais com imagens renderizadas georreferenciadas. O método tem bons resultados, no entanto, o modelo enfrenta dificuldades em cenas com muita textura auto-similar, ou seja, áreas onde partes menores da textura se assemelham às partes maiores em termos de padrões ou estruturas

repetitivas. Adicionalmente, enfrenta desafios em áreas com grandes sombras. Além disso, a utilização de uma reconstrução 3D da Terra do Google Earth como mapa para a estimativa de pose global apresenta desafios adicionais para a localização visual, devido às grandes diferenças de aparência causadas pela iluminação e mudanças sazonais, bem como alterações estruturais recentes no ambiente.

O método proposto por Shetty e Gao (2019) utiliza duas redes neurais para estimar a posição de um VANT por meio de geolocalização cruzada com imagens de satélite georreferenciadas. A técnica combina imagens de satélite com as capturadas por uma câmera terrestre para obter alta precisão de localização, sem a necessidade de GPS ou sensores externos.

Kim e Walter (2017) propuseram uma abordagem baseada em aprendizado de máquina para a localização de um veículo terrestre usando imagens de satélite. O método utiliza um modelo neural multi-visão que aprende *embeddings* para fazer a correspondência entre imagens em nível do solo com suas respectivas visualizações de satélite da cena. O modelo é avaliado em vários conjuntos de dados e demonstra a habilidade de localizar imagens em nível do solo em ambientes novos com variações significativas de ponto de vista e aparência.

Mollie et al. (BIANCHI; BARFOOT, 2021) propõem uma técnica de localização de UAV utilizando imagens de satélite codificadas automaticamente. Esse método emprega *autoencoders* para comprimir imagens de satélite em representações de baixa dimensão, que são então combinadas com as imagens capturadas pelo VANT para estimar a localização. Embora o método alcance alta precisão de localização mesmo em ambientes com baixa textura e não dependa de GPS ou de qualquer outro sensor externo, ele é intensivo em termos computacionais, e a precisão diminui quando há diferenças significativas entre as imagens capturadas pelo VANT e as do conjunto de dados de treinamento.

Além disso, outros trabalhos relacionados já foram discutidos na Seção 1.1, mas é importante destacar a relevância de dois deles: o trabalho de Mantelli et al. (2019), que serviu de base para o método proposto, e o trabalho de Choi e Myung (2020), que é semelhante ao nosso e usaremos alguns princípios do método para comparar os resultados obtidos.

4 A IMPORTÂNCIA DE UM BOM MODELO DE OBSERVAÇÃO NO PROBLEMA DE LOCALIZAÇÃO DE VANTS

Conforme comentado em seções anteriores, o algoritmo MCL é uma técnica amplamente utilizada para resolver problemas de localização em robótica (THRUN, 2002). A ideia básica por trás do MCL é usar partículas, que são cópias virtuais do robô, para estimar a pose do robô. O algoritmo funciona espalhando partículas no mapa do ambiente e usando um ciclo de previsão e correção para estimar a pose do robô.

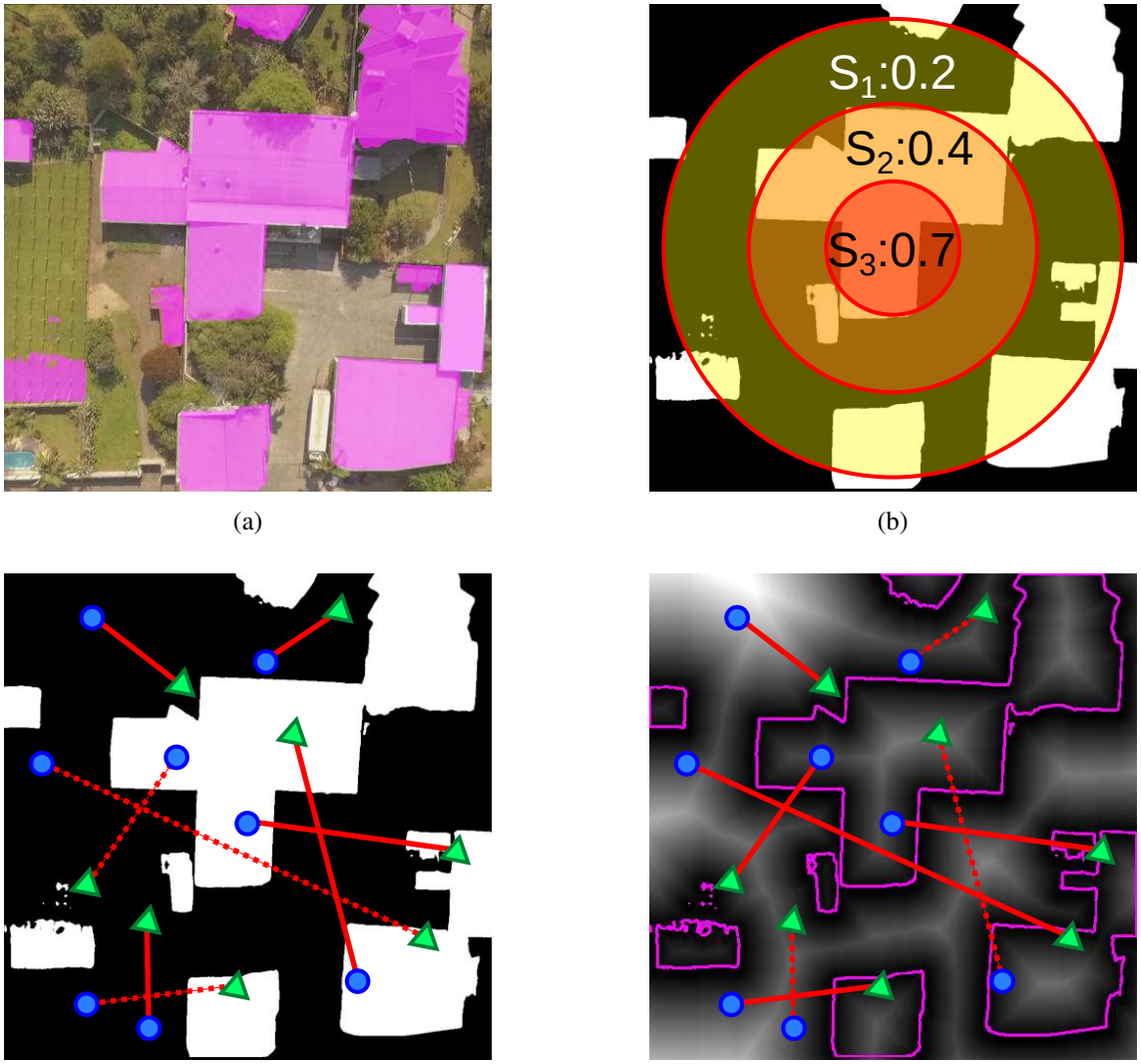
Na etapa de previsão do MCL, um modelo de movimento é usado para mover as partículas de acordo com as informações de odometria do robô. Na etapa de correção, um modelo de medição é usado para comparar as leituras do sensor do robô com as leituras das partículas. As partículas são então ponderadas de acordo com o quão bem suas leituras correspondem às leituras do robô, e um novo conjunto de partículas é amostrado com base em seus pesos.

O MCL é um dos algoritmos mais populares usados para localização de VANTs, conforme mencionado em (COUTURIER; AKHLOUFI, 2021). No contexto da localização de VANTs, o mapa de referência usado é tipicamente uma imagem de satélite, e o modelo de medição compara a imagem capturada pelo VANT com patches do mapa de referência simulando a visão das partículas.

As imagens RGB capturadas por um VANT contêm uma quantidade considerável de informações sobre o ambiente que o VANT voa sobre. No entanto, imagens segmentadas que classificam o ambiente de acordo com apenas uma classe, como prédios, têm informações limitadas, com apenas dois valores de pixel possíveis: verdadeiro ou falso. Isso cria um *trade-off* entre depender de informações suscetíveis a variações e não ter informações suficientes para realizar a correspondência de imagem de forma eficiente.

Para lidar com esse *trade-off*, diferentes estratégias de correspondência podem ser aplicadas a imagens binárias segmentadas como prédio/não prédio. Três abordagens diferentes foram selecionadas e estão ilustradas na Figura 4.1. A Figura 4.2 mostra as comparações entre as três diferentes estratégias de correspondência aplicadas a quatro imagens contendo diferentes configurações de prédios.

A primeira estratégia analisada é a aplicação do conceito de Razão de Construção (CHOI; MYUNG, 2020), ilustrado na Figura 4.1b. Ela calcula uma estimativa de densidade de núcleo indicando a proporção de uma área associada a edifícios sobre a área total



(a) (b)

(c) (d)

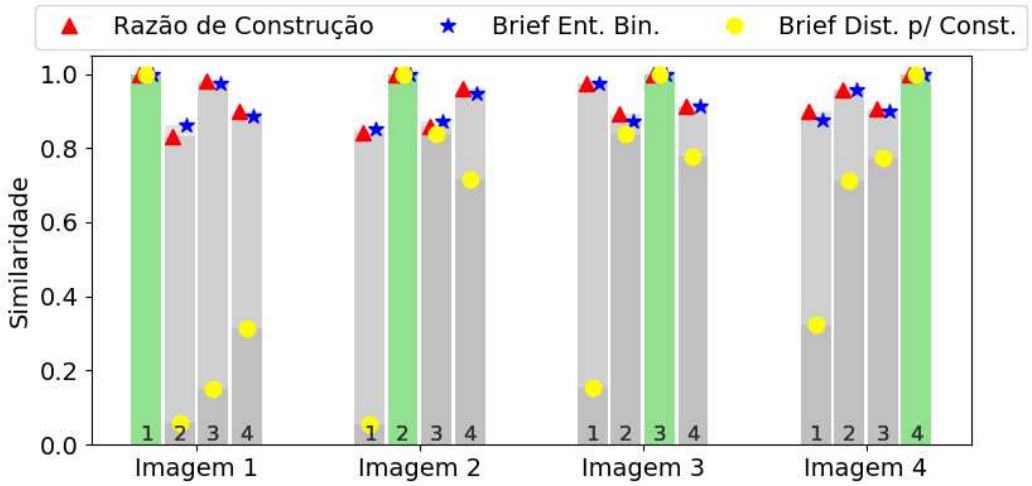
Figura 4.1 – Diferentes descritores considerando as informações de construções mostradas em (a): (b) **Razão de Construção**: densidade de células de construções dentro de um kernel (são mostrados os resultados com três tamanhos diferentes de kernel); (c) **BRIEF sobre entrada binária** (usando informação de construções e não-construções): comparações de valores binários extraídos de pares aleatórios de pontos; e (d) **BRIEF com Distância para Construções**: da mesma forma, mas comparando valores de distância para bordas de edifícios.

dentro de um núcleo centralizado no meio da imagem[1]. É possível observar na Figura 4.2 que configurações muito diferentes de edifícios têm razões de construção semelhantes porque o espaço sem edifícios nas imagens é muito maior do que o oposto. Portanto, é uma estratégia que serve para diferenciar bem apenas mudanças substanciais entre regiões.

A segunda estratégia analisada, que chamamos de BRIEF sobre entrada binária e é ilustrada na Fig. 4.1c, é uma variação do algoritmo de correspondência abBRIEF (MANTELLI et al., 2019), mas usando a imagem binária de classificação de edifícios

[1]Núcleos de tamanhos diferentes podem ser usados em paralelo para obter mais informações. Nos testes, foram utilizados três tamanhos, como sugerido em (CHOI; MYUNG, 2020).

(a) Imagem 1 (b) Imagem 2 (c) Imagem 3 (d) Imagem 4



(e) Similaridades entre imagens usando três diferentes abordagens de comparação

Figura 4.2 – Análise de três diferentes medidas de similaridade (***Razão de Construção***, ***BRIEF sobre entrada binária*** e ***Distância para Construções***) calculadas entre pares de quatro imagens contendo informações de prédios (Imagens 1 a 4). As imagens originais capturadas pelo VANT são mostradas em segundo plano (cores mais escuras). Apenas a informação da existência de prédios (rosa) ou sua ausência é utilizada pelos métodos para computar as similaridades.

como fonte. O descritor é construído comparando pares de pixels aleatórios dentro da mesma imagem e estabelecendo se o primeiro elemento do par tem uma intensidade maior do que o segundo. Para comparar duas imagens, o descritor é gerado em cada imagem, com os mesmos pares de pontos, e a diferença entre os descritores é o resultado de similaridade. Esse tipo de estratégia funciona bem para imagens coloridas (MANTELLI et al., 2019), já que qualquer variação de cores tende a ser consistente dentro da própria imagem (por exemplo, tudo é mais brilhante ou com maior contraste). Por outro lado, em imagens binárias, as regiões são principalmente homogêneas, então há uma grande chance de que pares distintos de pontos gerem o mesmo resultado. Podemos ver que, como a maior parte de cada imagem não contém edifícios, a maioria dos pares de pontos terá o mesmo valor, e os descritores gerados se tornarão muito semelhantes.

Finalmente, a terceira estratégia analisada, ilustrada na Figura 4.1d, é a que estamos propondo neste trabalho: o uso da proximidade com as construções em vez de simplesmente considerar se a região é ou não uma construção. A ideia é que, ao usar

os dados de proximidade com as construções, isto é, a distância até a construção mais próxima, tenhamos mais informações para usar no processo de correspondência, garantindo que essas informações sejam tão confiáveis quanto a imagem binária original. Com essa estratégia, as diferenças no resultado da correspondência de imagens são muito mais pronunciadas, como mostrado na Fig.4.2e. Por exemplo, a imagem na Figura 4.2a, que não contém construções, se torna muito diferente das outras. E embora não tão diferentes entre si, as três imagens restantes têm uma diferença mais significativa entre elas usando a estratégia proposta do que avaliando-as com as demais estratégias, o que é excelente para melhorar a qualidade da estimativa de localização.

Comparado aos descritores anteriores, especialmente ao conceito de Razão de Construção, o descritor baseado em Distância para Construções é muito mais sensível, mas menos robusto ao ruído. No entanto, isso é uma desvantagem menor em relação aos outros métodos cuja principal desvantagem é usar tão pouca informação que poderia impedir uma localização bem-sucedida. Com essa estratégia, as diferenças no resultado de correspondência de imagem são muito mais pronunciadas, mesmo para imagens que não contêm edifícios, como mostrado na Fig. 4.2e. As três imagens restantes também têm uma diferença mais significativa entre elas com essa estratégia do que avaliá-las com as anteriores, o que é excelente para melhorar a qualidade da estimativa de localização.

Em geral, a estratégia proposta oferece uma solução promissora para o *trade-off* entre depender de informações suscetíveis a variações e não ter informações suficientes para realizar a correspondência de imagem de forma eficiente na localização de VANT.

5 NBD-BRIEF - LOCALIZAÇÃO USANDO INFORMAÇÃO DE DISTÂNCIA PARA CONSTRUÇÕES

Nesta seção, descreveremos o modelo que propomos, chamado de NBD-BRIEF, bem como as etapas necessárias para calcular e implementa-lo. Depois, apresentaremos nosso modelo de observação completo, baseado no NBD-BRIEF.

5.1 Definindo o NBD-BRIEF (*Nearest Building* BRIEF)

Esta dissertação propõe uma nova abordagem para localização global usando MCL que compara imagens obtidas por um VANT olhando para baixo com uma imagem de satélite da região onde o robô está localizado. O algoritmo MCL pondera partículas com base em um novo descritor chamado **NBD-BRIEF** (*Nearest Building Distance BRIEF*) ou **BRIEF usando Distância para Construção mais Próxima**. Esse descritor é obtido a partir de imagens que descrevem a proximidade de construções, conforme ilustrado na seção anterior. Uma visão geral do *framework* é apresentada na Figura 5.1. Nela descrevemos como cálculo da NBD a aquisição dos contornos dos prédios e o uso da transformada de distância.

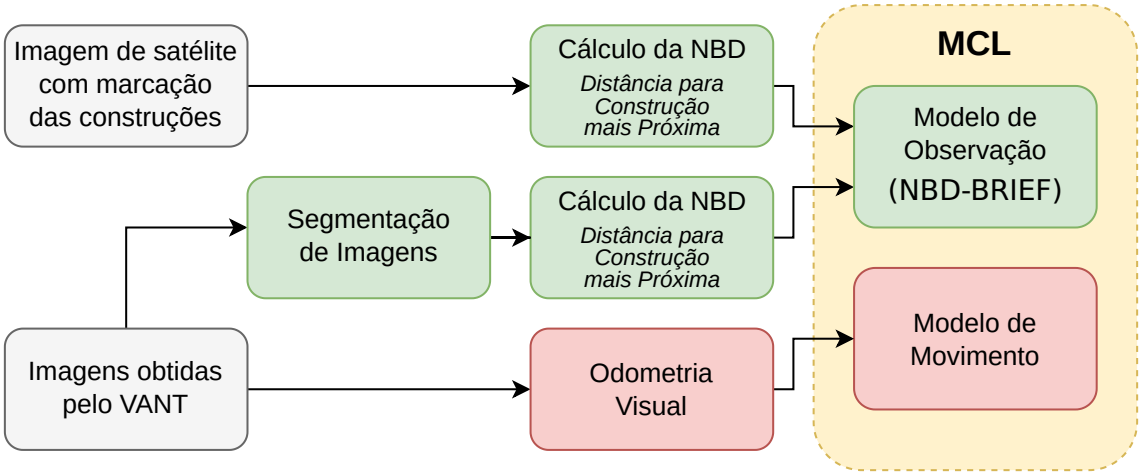


Figura 5.1 – Visão geral do *framework* proposto. A transformação de distância é aplicada na imagem do VANT segmentada e no mapa de referência, que é usado no MCL junto com a odometria.

No *framework* proposto, o mapa de referência, **M**, é uma imagem de satélite que contém informações sobre edifícios. Esta imagem é convertida em um mapa que descreve a proximidade de construções através da aplicação de uma transformação de distância. Este passo permite a construção do descritor baseado em BRIEF que é usado

no algoritmo MCL para estimar a pose do robô. O processo de obtenção de **M** é simples, pois mapas com contornos dos edifícios estão amplamente disponíveis em ferramentas como o Google Maps e o OpenStreetMap. A Figura 5.2 mostra uma imagem de satélite com uma sobreposição do contorno do edifício que foi obtida a partir do Google Maps.

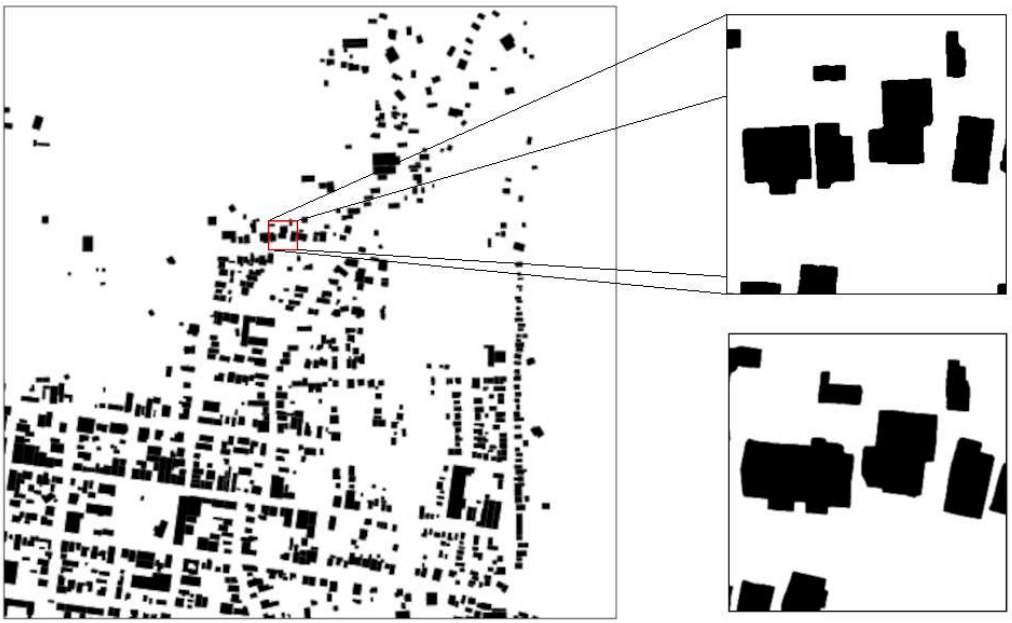


Figura 5.2 – Mapa de referência, **M**, usado no voo 1. O contorno dos edifícios foram extraídos do Google Maps. Ambas as imagens segmentadas na coluna da direita representam o mesmo lugar na Terra. A diferença é que a imagem superior é um patch de **M**, enquanto a inferior é a imagem do VANT segmentada associada à área destacada no mapa.

O framework proposto utiliza a informação de proximidade de edifícios para construir um descritor baseado em BRIEF que é altamente robusto às mudanças no ambiente. Esta abordagem supera descritores tradicionais, como o conceito de Razão de Edifícios, que são menos sensíveis às mudanças, mas menos robustos ao ruído. Além disso, a abordagem proposta fornece informações suficientes para realizar o casamento de imagens com eficiência, mesmo em cenários onde não há diferença significativa entre as imagens.

A seguir serão descritas as etapas necessárias para realizar localização de VANTs usando a estratégia proposta. Primeiramente, é explicada como é feita a segmentação das imagens para determinar o que é ou não é informação de construções no ambiente. Na sequência, a computação do novo descritor baseado na distância para construções é detalhada. Por fim, é explicado como aplicar o método proposto na estratégia de localização de Monte Carlo.

5.1.1 Segmentação das imagens

O primeiro passo na abordagem proposta é segmentar imagens de VANT como construções ou não-construções, usando um framework baseado na rede neural U-Net (RONNEBERGER; FISCHER; BROX, 2015). Embora o processo de segmentação não seja uma contribuição direta deste trabalho, selecionamos um dos métodos de segmentação mais populares e precisos disponíveis.

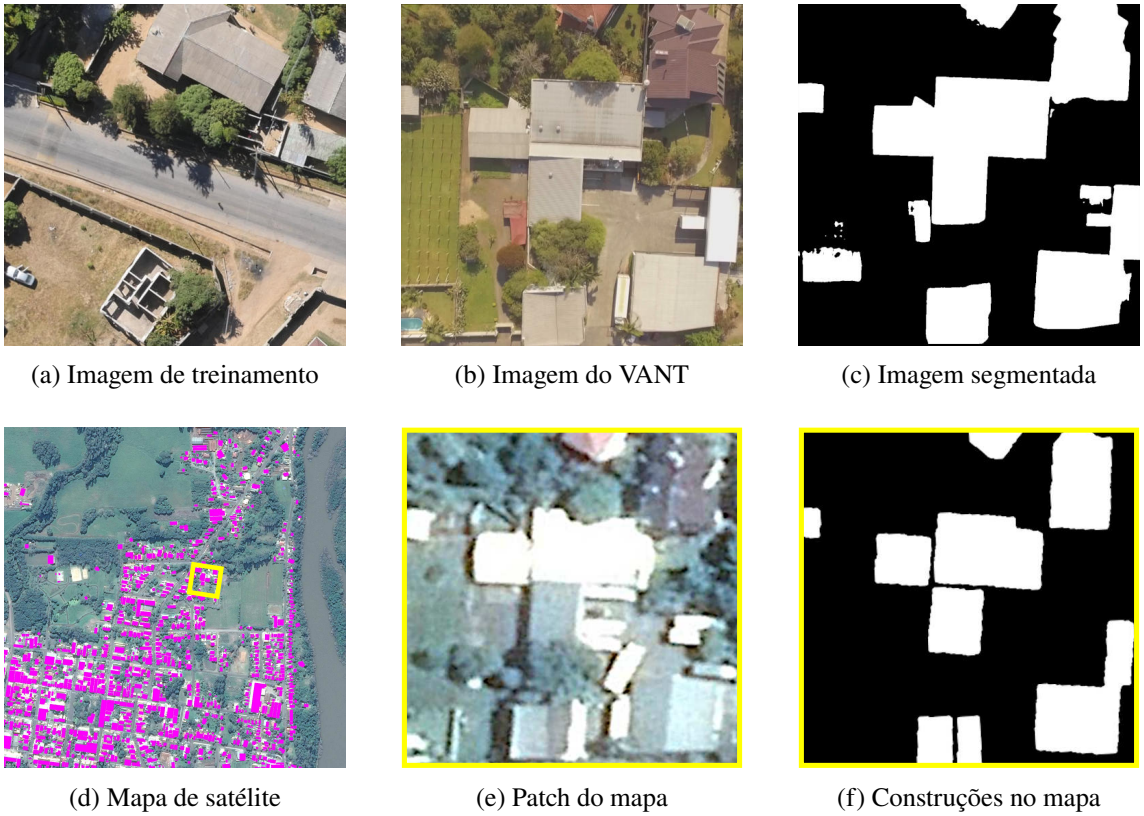
A rede U-Net compreende um caminho de contração e um caminho de expansão simétrico, em que ela codifica e decodifica características em uma imagem segmentada. Neste trabalho, usamos uma rede pré-treinada (DrivenData, 2020), que foi treinada em um conjunto de dados (GFDRR Labs, 2020) composto por imagens de cidades africanas e sua infraestrutura, contorno de edifícios e vegetação. Essas regiões possuem construções semelhantes àquelas onde foram conduzidos nossos experimentos. Na Figura 5.3 podemos ver um exemplo de imagem do dataset de treinamento, juntamente com uma imagem obtida pelo VANT, o resultado de sua segmentação e a comparação com as informações obtidas do mapa de referência no local onde o VANT estava.

É importante notar que, embora a segmentação de edifícios possa não ser 100% precisa, as informações de entrada para o filtro de partículas ainda são confiáveis devido ao pequeno impacto que pequenas mudanças nos contornos dos edifícios têm nos valores das distâncias até o edifício mais próximo. Claro, erros graves na classificação de edifícios ou mudanças nos edifícios presentes no mapa de referência podem prejudicar o processo de localização e fazê-lo falhar. No entanto, isso é um problema inerente à tarefa de localização, onde se espera que as observações correspondam adequadamente ao mapa disponível. O filtro de partículas é geralmente uma estratégia bem-sucedida quando as diferenças entre as informações esperadas e observadas são pequenas (COUTURIER; AKHLOUFI, 2021; THRUN, 2002).

5.1.2 Cálculo da distância do prédio mais próximo

Imagens segmentadas são difíceis de serem comparadas, como mostrado na Fig. 4.2. Nós propomos o uso de uma versão modificada do descritor BRIEF para melhorar os resultados. Ela utiliza informações sobre a distância até a borda mais próxima de um prédio, chamada de ***NBD-BRIEF***.

Primeiramente, as bordas dos prédios são extraídas da imagem segmentada utili-



(a) Imagem de treinamento (b) Imagem do VANT (c) Imagem segmentada

(d) Mapa de satélite (e) Patch do mapa (f) Construções no mapa

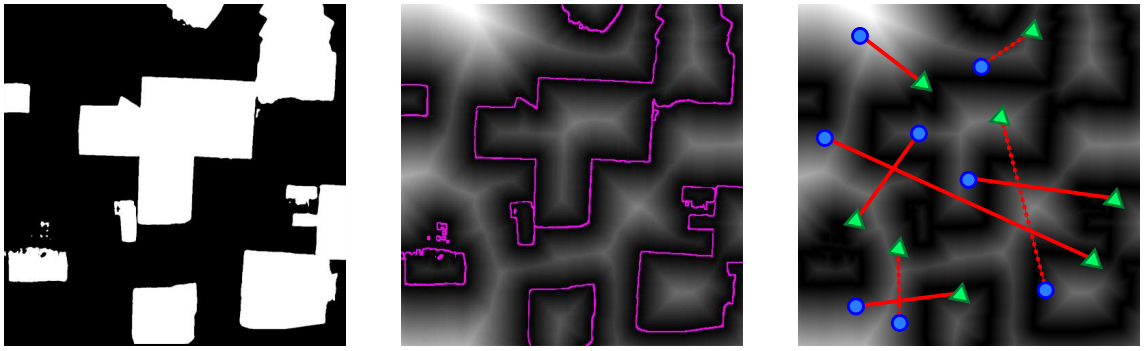
Figura 5.3 – Exemplo de segmentação de imagens para identificação de construções e comparação com informação contida no mapa. (a) Exemplo de imagem de treinamento do dataset escolhido (GFDRR Labs, 2020). (b) Exemplo de imagem obtida pelo VANT. (c) Resultado da segmentação usando a rede escolhida. (d) Mapa do ambiente de teste contendo o local sobrevoado pelo VANT, destacado por um quadrado amarelo. Em magenta estão as marcações de construções fornecidas junto com o mapa. (e) Destaque da imagem delimitada pelo quadrado amarelo. (f) Destaque da classificação dos prédios fornecida pelo mapa na área delimitada pelo quadrado amarelo.

zando o detector de bordas Canny (CANNY, 1986), resultando em uma imagem $\mathbf{I}$. Em seguida, $\mathbf{I}$ é utilizada para calcular a distância euclidiana de um determinado pixel $\mathbf{p}$ até a borda mais próxima do prédio, dada por

$$d(\mathbf{p},\mathbf{I}) = \min\left\{\min_{\mathbf{q}\in\mathbf{I}^e}\sqrt{(p_u-q_u)^2+(p_v-q_v)^2}\,,\,L\right\}, \tag{5.1}$$

onde p_u e p_v são as coordenadas do pixel $\mathbf{p}$, q_u e q_v são as coordenadas de um pixel de borda $\mathbf{q}$ pertencente ao conjunto de todos os pixels de borda $\mathbf{I}^e \subseteq \mathbf{I}$, e L é um limite máximo de distância. A Figura 5.4b apresenta um exemplo da imagem resultante ao aplicar a transformada de distância em todos os pixels da imagem $\mathbf{I}$.

O limite máximo L é necessário porque o VANT só observa prédios dentro de seu campo de visão atual; portanto, uma imagem sem prédios tem informações de distância iguais a zero. Isso não é o caso no mapa de referência, onde todos os prédios são



(a) Imagem segmentada (b) Transf. de distância (c) NBD-BRIEF

Figura 5.4 – Exemplo de obtenção do NBD-BRIEF. A partir da imagem segmentada (a), as bordas das construções são encontradas (b, em magenta). A seguir a transformação de distâncias é computada a partir das bordas (b, em tons de cinza). Por fim, comparando os valores de distância nas posições de pares de pixels em locais aleatórios da imagem transformada constrói-se o descritor.

conhecidos *a priori*. Se não houver limite de distância, até mesmo regiões distantes dos prédios terão informações de distância no mapa de referência, levando a inconsistências em relação aos descritores calculados nas imagens do VANT. Em nossos testes, o limite ideal para a transformada de distância nas imagens do VANT fica entre 25% e 50% da maior dimensão da imagem (comprimento ou largura). O limite no mapa é multiplicado por uma estimativa da escala entre o mapa e a imagem, que depende da altura do VANT[1].

No método NBD-BRIEF, um conjunto de pares de pixels $\mathbf{S} = \mathbf{s}_1, \ldots, \mathbf{s}_k$ é selecionado aleatoriamente, utilizando uma distribuição gaussiana, em uma imagem, onde cada par $\mathbf{s} = \mathbf{x_1}, \mathbf{x_2} \in \mathbf{S}$ é composto por dois pontos aleatórios, como ilustrado na Figura 5.4c. Durante a seleção, é utilizada uma distribuição gaussiana no centro da imagem para minimizar os efeitos de edifícios que aparecem subitamente perto das bordas da imagem.

O vetor de características binárias $\mathbf{B} = (\tau_1, \tau_2, \cdots, \tau_k)$ que descreve o NBD-BRIEF é composto por k comparações binárias τ, em nossos testes foi usado $k = 256$, mesmo valor usado no abBRIEF. Ao contrário do método abBRIEF (MANTELLI et al., 2019), onde é feita a comparação da intensidade de dois pixels, nós comparamos a distância desses pixels até a borda do edifício mais próximo, conforme dado por

$$\tau(\mathbf{I}; \mathbf{s}) = \left\{ \begin{array}{ll} 1 & : d(\mathbf{x_1}, \mathbf{I}) < d(\mathbf{x_2}, \mathbf{I}) \\ 0 & : otherwise \end{array} \right\} \tag{5.2}$$

[1]Como o filtro de partículas utilizado no MCL considera alturas máxima e mínima de partículas, também podemos calcular a diferença de escala máxima e mínima (em pixels × metros) entre o mapa de referência e as imagens do VANT. Durante os experimentos, selecionamos um valor de escala fixo dentro desse intervalo, o que se mostrou funcionar bem, apesar de ser uma aproximação.

A similaridade entre duas imagens é dada pelo cálculo da distância de Hamming (Hd) de cada descritor de imagem NBD-BRIEF, como em (MANTELLI et al., 2019).

5.2 Aplicação do NBD-BRIEF no MCL para localização de um VANT

Assim como em (MANTELLI et al., 2019), a localização do VANT é feita incorporando a medida de similaridade do descritor proposto como parte do modelo de observação do veículo. A probabilidade de uma dada observação no MCL, usada durante a etapa de pesagem do filtro de partículas, é modelada com uma distribuição gaussiana com média zero, computada em função da distância de Hamming entre descritores da imagem observada pelo VANT e da imagem esperada por cada partícula:

$$p(\mathbf{I}_t \,|\, \boldsymbol{x}_t^{[p]}, \mathbf{M}) = \mathcal{N}(Hd(\mathbf{B}_t, \hat{\mathbf{B}}_t^{[p]}), 0, \sigma), \tag{5.3}$$

Nessa equação, $\mathbf{B}_t$ é o descritor computado a partir da imagem do VANT $\mathbf{I}_t$, $\hat{\mathbf{B}}_t^{[p]}$ é o descritor computado da imagem esperada obtida a partir de $\mathbf{M}$ associada a uma partícula na pose $\boldsymbol{x}_t^{[p]}$, e o desvio padrão σ é um parâmetro de ruído intrínseco do modelo[2]. Uma vantagem desse tipo de abordagem baseada em BRIEF, é que não se faz necessário extrair um patch de imagem do mapa de referencia; basta transformar (i.e. transladar, rotacionar e escalar) o conjunto de pares de pixels usados pelo descritor diretamente do mapa de referência em função da pose de cada partícula.

A estimativa de localização proposta considera 4 graus de liberdade: a posição 2D sobre o mapa de referência, a altura do VANT e o ângulo yaw [3]. O modelo de movimento usado também é o mesmo de (MANTELLI et al., 2019), que usa como base uma odometria visual computada através do casamento de *features* usando SIFT entre imagens subsequentes obtidas pelo VANT.

[2]Nos nossos testes, utilizamos σ como 15% da distância de Hamming máxima.

[3]As outras orientações do VANT, roll e pitch, são consideradas zero, pois o VANT procura voar sempre paralelo ao solo.

6 EXPERIMENTOS E DISCUSSÃO

Nesta seção, apresentaremos a validação experimental realizada, incluindo informações sobre três datasets de voos realizados em duas regiões do estado do Rio Grande do Sul, Brasil, utilizando VANTs. As trajetórias dos voos serão mostradas em sobreposição aos mapas utilizados nos testes. Serão fornecidas informações adicionais sobre os voos, como a obtenção de imagens de satélite e contornos de edifícios por meio do Google Maps, bem como os desafios específicos enfrentados em cada teste. Além disso, serão apresentadas as comparações com outras abordagens, como o abBRIEF e a variação da abordagem da razão de construção, juntamente com os detalhes dos parâmetros utilizados em todos os experimentos.

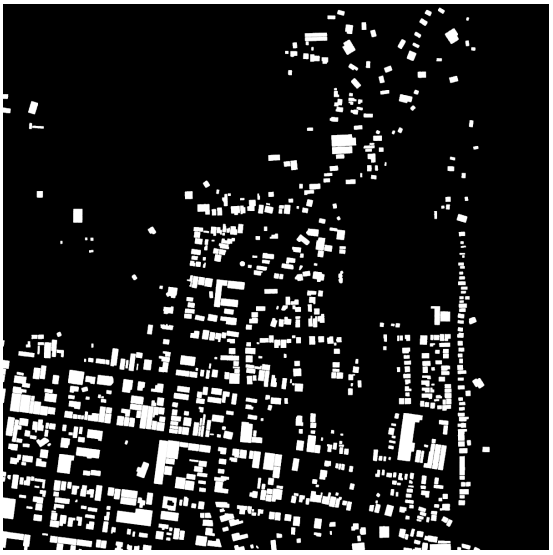
6.1 Configuração dos experimentos

A validação experimental foi realizada com três *datasets* de voos realizados por dois modelos diferentes de VANT equipados com GPS, câmera e IMU no estado do Rio Grande do Sul, Brasil. As trajetórias dos voos 1, 2 e 3 são ilustradas respectivamente nas Figuras 6.1, 6.2 e 6.3. Em cada uma das figuras mostramos o mapa de satélite da região, o mapa das construções e uma sobreposição dos dois com a trajetória do VANT. Informações adicionais sobre os voos podem ser encontradas na Tabela 6.1. Para obter imagens de satélite e contornos de edifícios para os experimentos, usamos o Google Maps. Embora outras fontes de imagens de satélite pudessem ter sido utilizadas, o Google Maps fornece as imagens mais detalhadas para a região onde os VANTs sobrevoaram. Os mapas usados datam de 2021, e cada voo foi testado em apenas um mapa, pois mapas com contornos de edifícios mais antigos não estavam disponíveis.

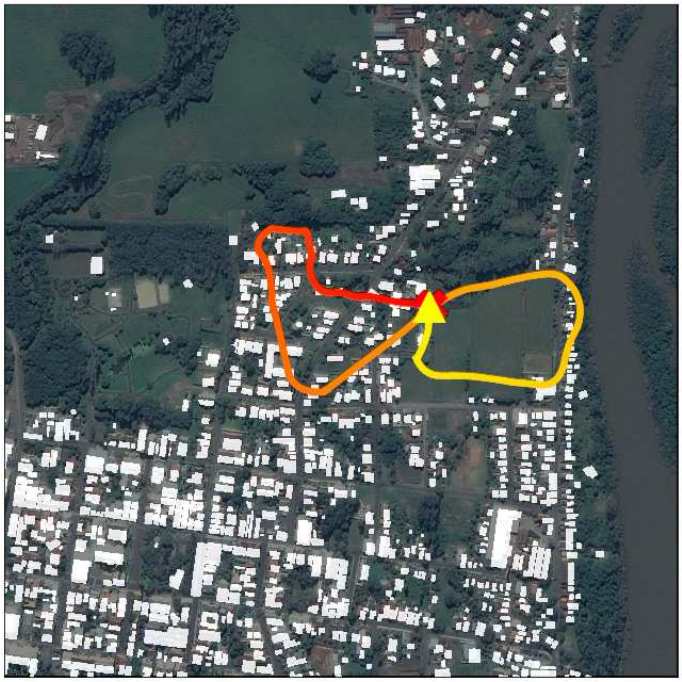
Tabela 6.1 – Detalhes dos voos

	Voo 1	Voo 2	Voo 3
Local	Arroio do Meio	Arroio do Meio	Porto Alegre
VANT	DJI Phantom 3	DJI Matrice 100	DJI Matrice 100
Gimbal	Sim	Não	Não
Tempo de voo	358 s	290 s	306 s
Distância percorrida	1800 m	1120 m	1085 m
Área do mapa	1.09 km²	0.81 km²	0.08 km²
Altitude	35m - 130m	35m - 180m	40m - 50m

(a) Mapa do Ambiente 1



(b) Mapa de construções do Ambiente 1

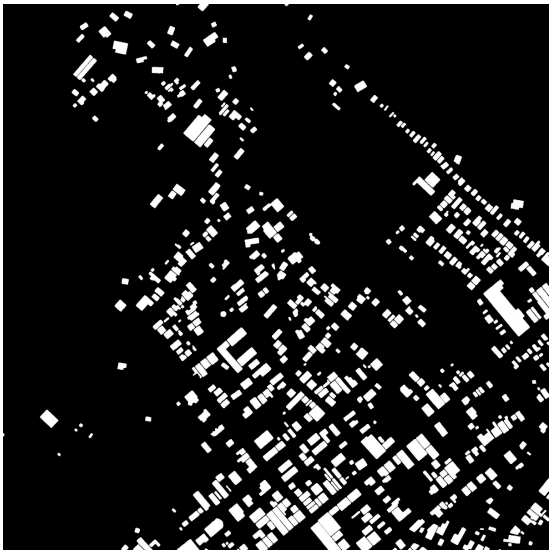


(c) Trajetória do Voo no Ambiente 1

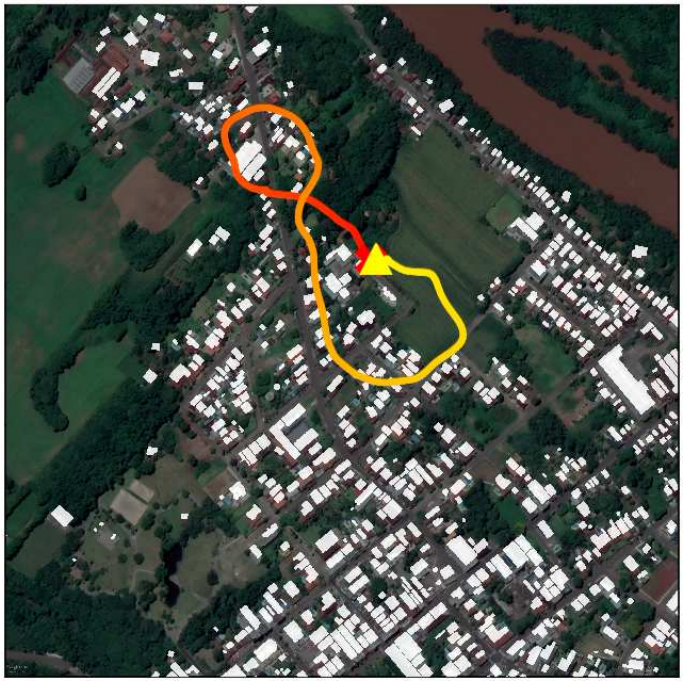
Figura 6.1 – Mapa e Trajetória do Ambiente 1. (a) Imagem de satélite da área do voo. (b) Imagem com áreas de construções destacadas em branco. (c) Trajetória do voo sobre imagem mesclando mapa de construções e imagem de satélite. O voo começa no 'X' vermelho e termina no triângulo amarelo.

Os três testes apresentaram desafios únicos. O primeiro teste usou um mapa de referência maior, enquanto o terceiro voo foi registrado em uma área com menos edifícios do que os outros dois. Além disso, o segundo e o terceiro voos foram registrados sem um *gimbal*, dispositivo usado na estabilização da câmera, o que afetou a odometria visual dos VANTs. As Figuras 6.4 e 6.5, mostram um exemplo de execução do método nos

(a) Mapa do Ambiente 2



(b) Mapa de construções do Ambiente 2

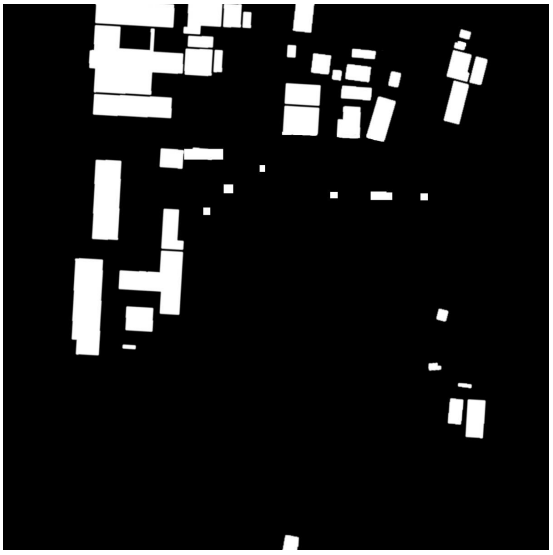


(c) Trajetória do Voo no Ambiente 2

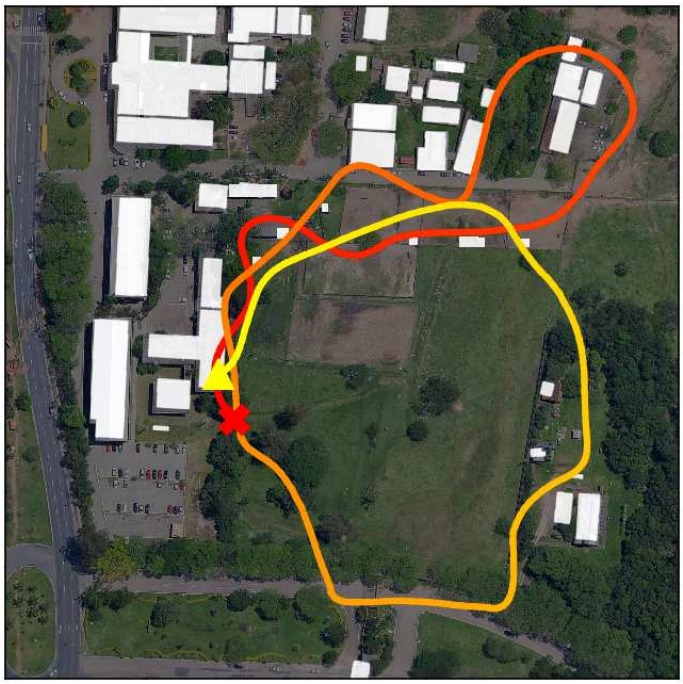
Figura 6.2 – Mapa e Trajetória do Ambiente 2. (a) Imagem de satélite da área do voo. (b) Imagem com áreas de construções destacadas em branco. (c) Trajetória do voo sobre imagem mesclando mapa de construções e imagem de satélite. O voo começa no 'X' vermelho e termina no triângulo amarelo.

ambientes 1 e 3, respectivamente. Pode-se ver no ambiente 1 que o filtro de partículas converge rapidamente conforme as primeiras observações e movimentos acontecem, e após isso a convergência se mantém até o fim do voo. Já no ambiente 3, o filtro até converge rapidamente, pois o voo inicia próximo a construções. No entanto, no meio da trajetória o VANT sobrevoa áreas sem qualquer construção, o que implica em nenhuma

(a) Mapa do Ambiente 3



(b) Mapa de construções do Ambiente 3

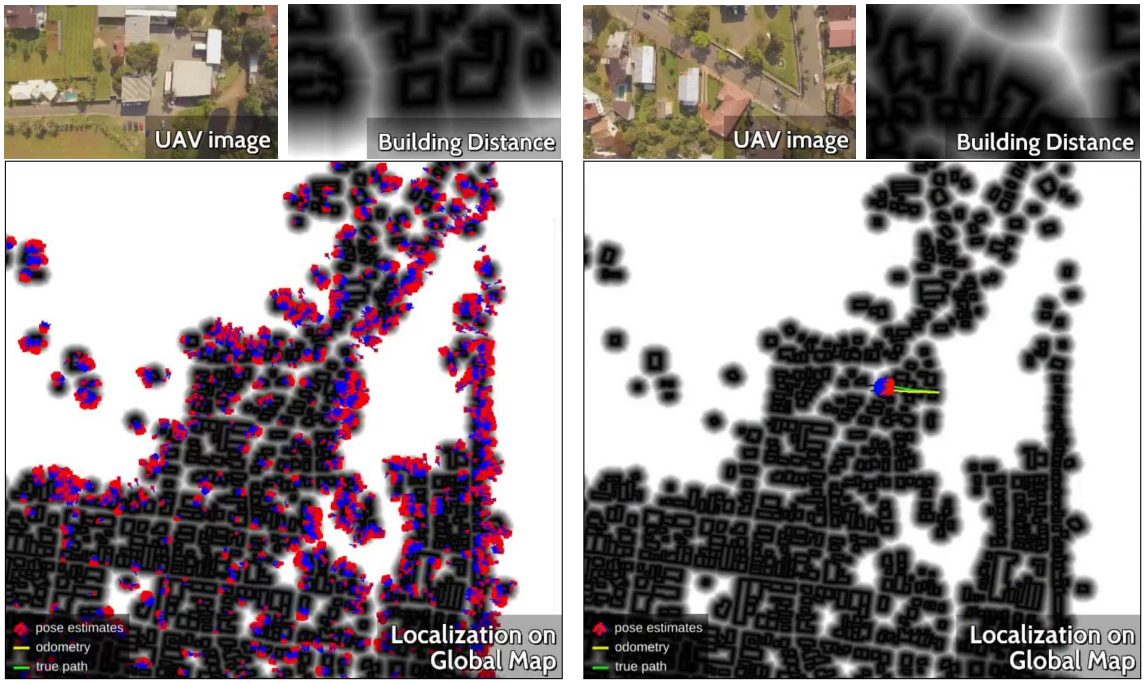


(c) Trajetória do Voo no Ambiente 3

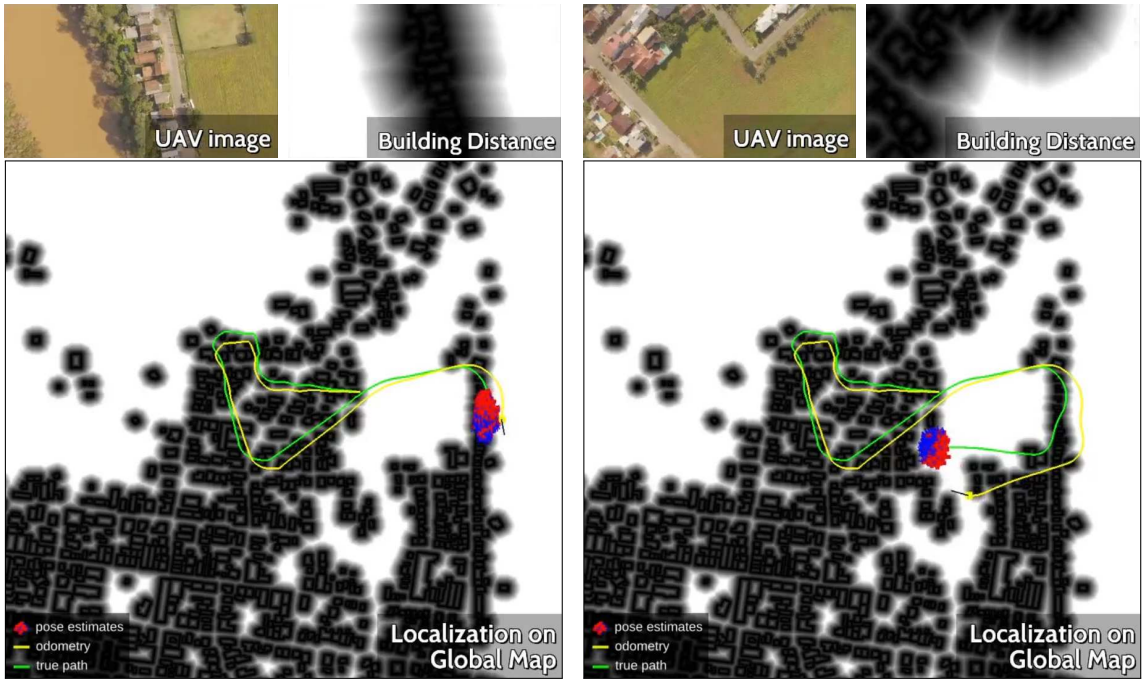
Figura 6.3 – Mapa e Trajetória do Ambiente 3. (a) Imagem de satélite da área do voo. (b) Imagem com áreas de construções destacadas em branco. (c) Trajetória do voo sobre imagem mesclando mapa de construções e imagem de satélite. O voo começa no 'X' vermelho e termina no triângulo amarelo.

fonte de informação para correções do filtro de partículas. Somente quando o VANT volta para áreas com obstáculos que a incerteza reduz novamente.

Em termos de comparações com outras abordagens, selecionamos o trabalho original do abBRIEF (MANTELLI et al., 2019), uma variação da abordagem da Razão de Construção (CHOI; MYUNG, 2020) e BRIEF com Entrada Binária (BRIEF-EB), que é



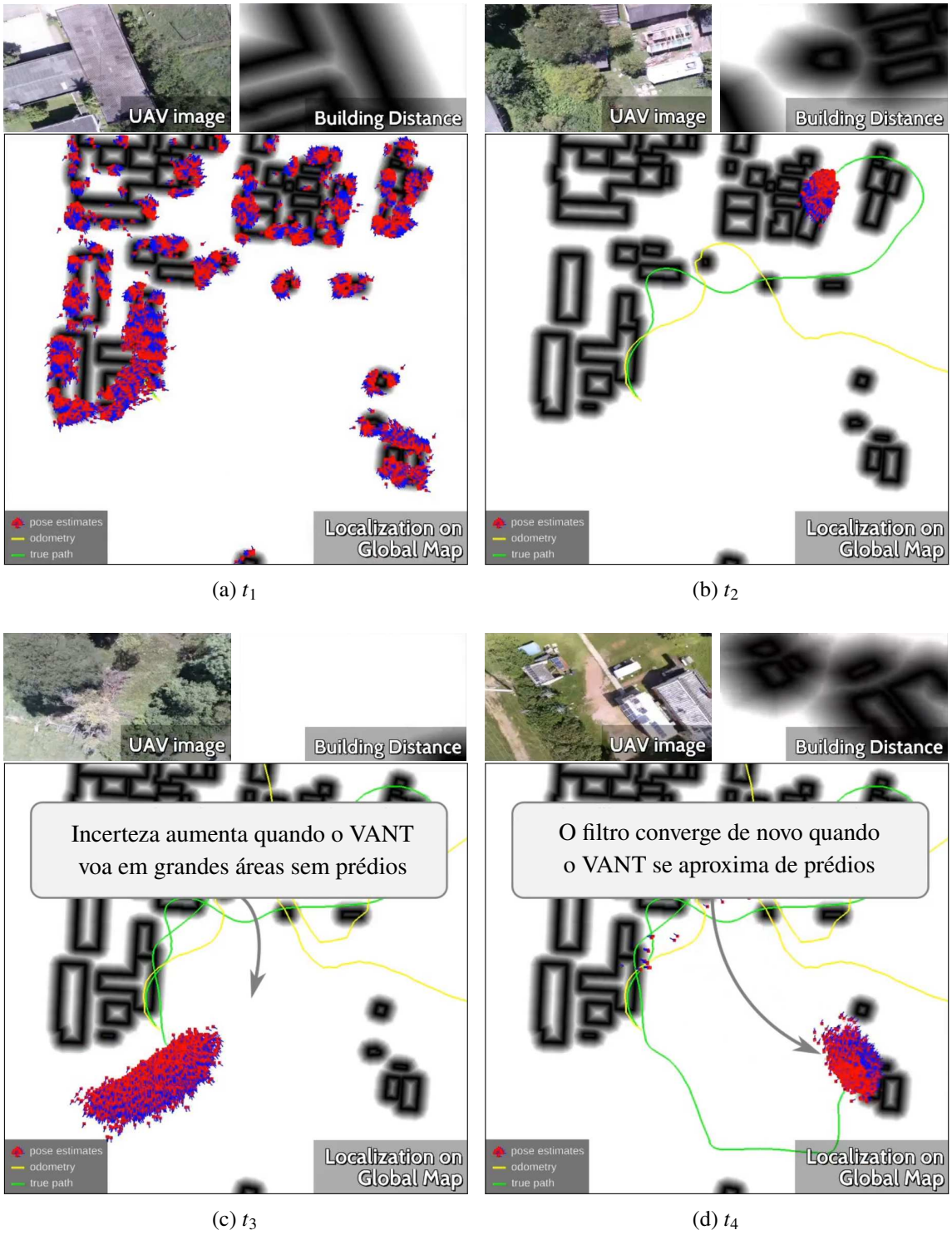
(a) t_1 (b) t_2



(c) t_3 (d) t_4

Figura 6.4 – Exemplo de resultado de teste no Ambiente 1. Pode-se ver que o filtro converge rapidamente para o local certo e mantém a convergência até o final.

a mesma variação de abBRIEF discutida na Seção 3.4, mas usa imagens segmentadas (construções x não-construções) ao invés de imagens coloridas. O abBRIEF foi testado com imagens no espaço de cor RGB, como no original. Para as versões modificadas usando a Razão de Construção e BRIEF-EB, usamos as imagens segmentadas do VANT e



(a) t_1 (b) t_2

(c) t_3 (d) t_4

Figura 6.5 – Exemplo de resultado de teste no Ambiente 3. Neste cenário mais complicado devido à pouca quantidade de construções, o método converge rapidamente enquanto sobrevoa os prédios (b). Porém a incerteza do filtro cresce bastante quando o VANT sobrevoa áreas sem construções (c). A incerteza só reduz novamente quando o VANT volta a se aproximar de prédios (d).

o mapa de referência contendo informações sobre os edifícios (o mesmo usado em nosso método antes de aplicar a transformada de distância).

Todos os experimentos foram realizados usando imagens capturadas a 1 FPS,

50.000 partículas no MCL e 30 execuções para cada configuração, conforme feito por Mantelli et al. (2019). A trajetória real usada para avaliação foi obtida diretamente a partir das coordenadas GPS durante os voos.

6.2 Resultados

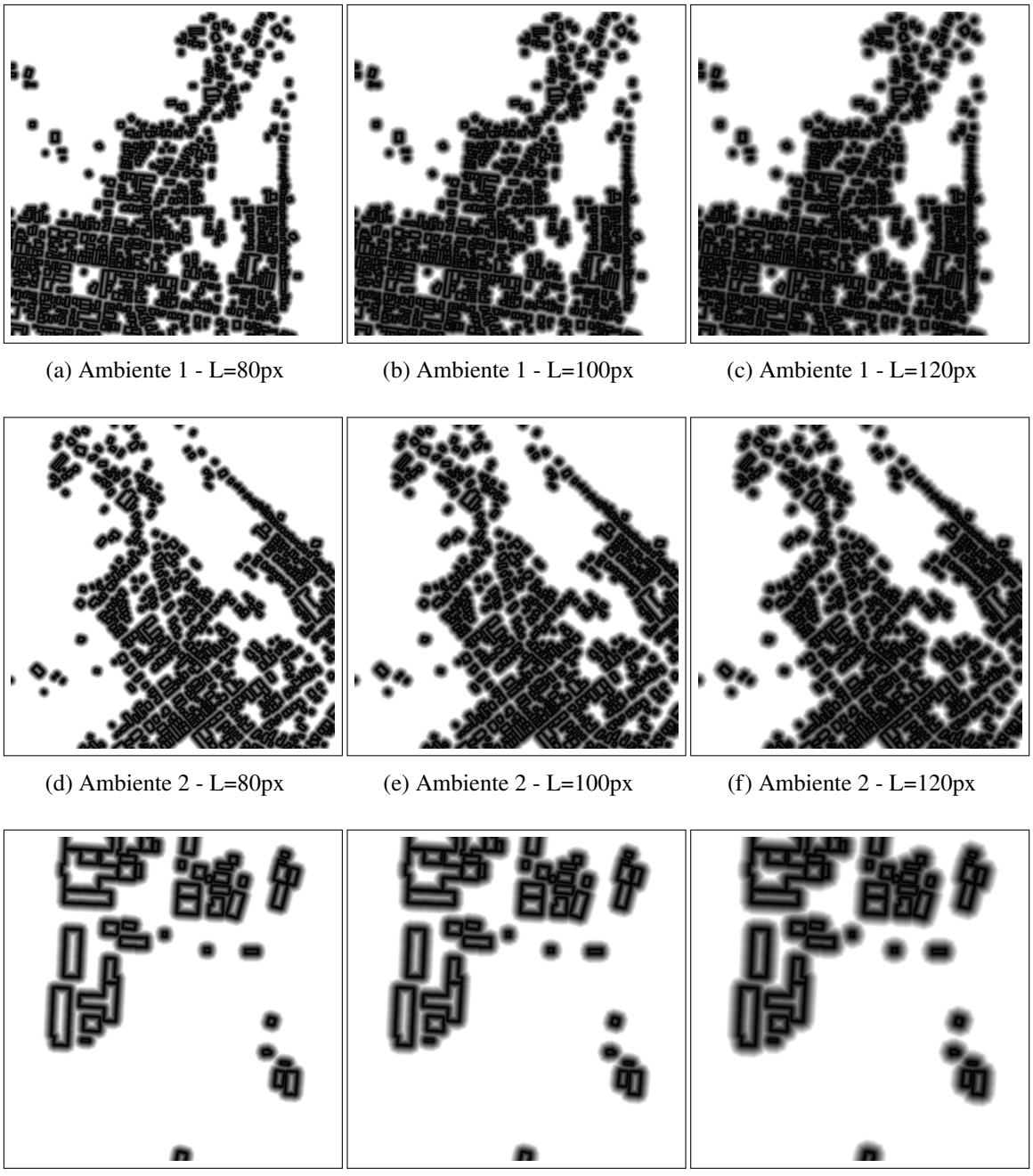
Como nosso trabalho é baseado em MCL, a localização do VANT é dita como completa após a convergência das partículas. Consideramos que as partículas convergem quando o erro de distância é inferior a 10% do tamanho do mapa de referência.

No primeiro conjunto de testes, realizamos a avaliação do nosso método utilizando três limites distintos para a transformada de distância: 80, 100 e 120 pixels. Observamos que valores levemente superiores ou inferiores a esses limites parecem ter pouca influência nos resultados obtidos. Por outro lado, valores muito altos ou muito baixos dificultaram ou até mesmo impossibilitaram a convergência do método. Diante disso, optamos por escolher esses três valores, a fim de simplificar a comparação entre os três ambientes. Na Figura 6.6 apresentamos os mapas de referências usados em cada ambiente e os diferentes limites na transformada de distância.

Tabela 6.2 – Resultados do NBD-BRIEF com diferentes limites para a transformada de distância

		Limite da transformada		
	Métrica	L=80	L=100	L=120
Voo 1	EAM (m)	21.74	**11.26**	**11.26**
	EAM Conv. (m)	20.04	9.16	**8.74**
	Conv. correta (%)	92.21	**95.19**	94.92
	Conv. errada (%)	**0**	0.02	**0**
	Não conv. (%)	7.79	**4.79**	5.08
Voo 2	EAM (m)	**61.32**	61.48	62.59
	EAM Conv. (m)	59.02	**57.57**	57.75
	Conv. correta (%)	71.35	**71.75**	71.02
	Conv. errada (%)	1.04	1.04	**0.90**
	Não conv. (%)	27.61	**27.21**	28.08
Voo 3	EAM (m)	17.49	17.15	**15.68**
	EAM Conv. (m)	**10.00**	10.84	10.81
	Conv. correta (%)	57.35	56.81	**58.02**
	Conv. errada (%)	4.97	4.34	**3.88**
	Não conv. (%)	**37.68**	38.85	38.01

A Tabela 6.2 apresenta os resultados do Erro Absoluto Médio (EAM), EAM após a convergência e as porcentagens de convergência correta, errada e sem convergência ao



(a) Ambiente 1 - L=80px (b) Ambiente 1 - L=100px (c) Ambiente 1 - L=120px

(d) Ambiente 2 - L=80px (e) Ambiente 2 - L=100px (f) Ambiente 2 - L=120px

(g) Ambiente 3 - L=80px (h) Ambiente 3 - L=100px (i) Ambiente 3 - L=120px

Figura 6.6 – Diferentes mapas de distâncias variando o limite de distância de construção nos três cenários de teste.

durante os testes das trinta execuções em cada ambiente. Podemos ver que a variação do parâmetro L geralmente não implica em grandes variações nos resultados, o que é bom porque mostra que o método pode funcionar com robustez mantendo o parâmetro fixo, mesmo sabendo que a altura do VANT varia durante os voos.

As Figuras 6.7, 6.8 e 6.9 exibem o EAM dos três ambientes avaliados, sendo que é possível notar que o método apresentou convergência praticamente nos mesmos instantes

para todos eles. No primeiro voo, constatamos uma discrepância no EAM entre o limite de distância L=80 em relação a L=100 e L=120 ao longo de todo o período, porém isso não ocorreu nos outros dois voos.

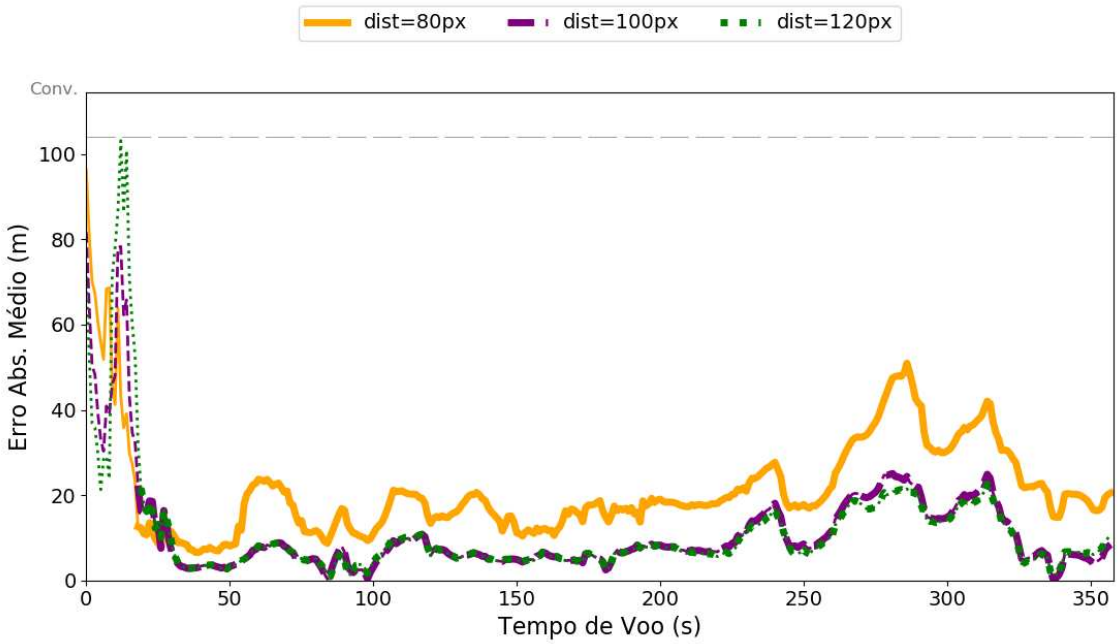


Figura 6.7 – Comparações de resultados obtidos com o método proposto no Ambiente 1, considerando três limites diferentes para a transformada de distância: 80, 100 e 120 pixels.

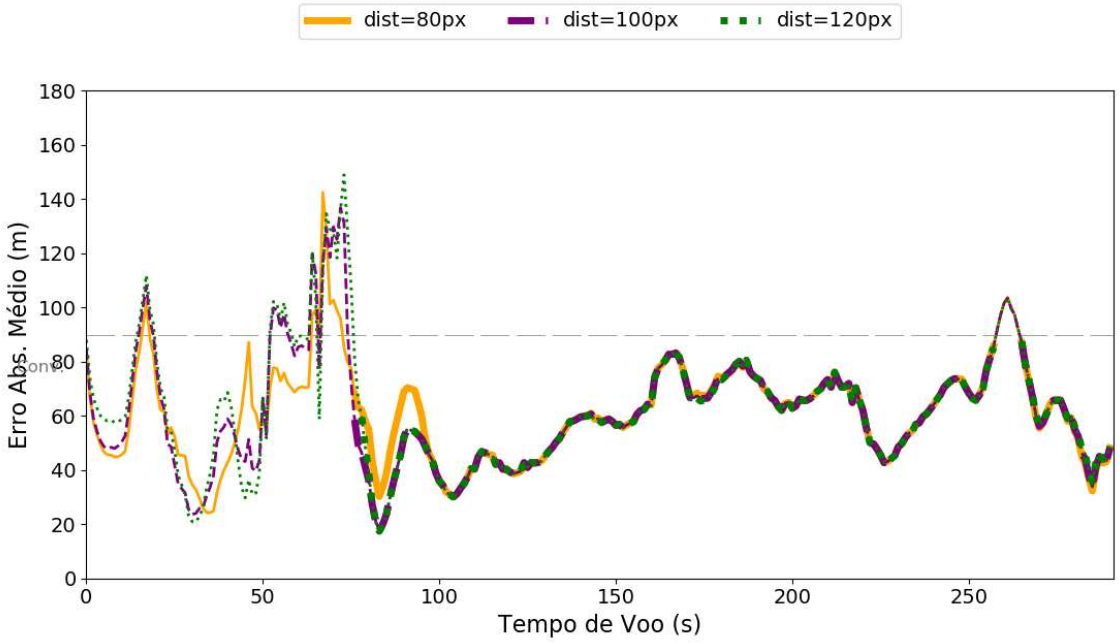


Figura 6.8 – Comparações de resultados obtidos com o método proposto no Ambiente 2, considerando três limites diferentes para a transformada de distância: 80, 100 e 120 pixels.

Isso acontece porque o método não precisa saber a distância exata de um ponto para um prédio, que seria uma medida que depende da escala atual da imagem e, con-

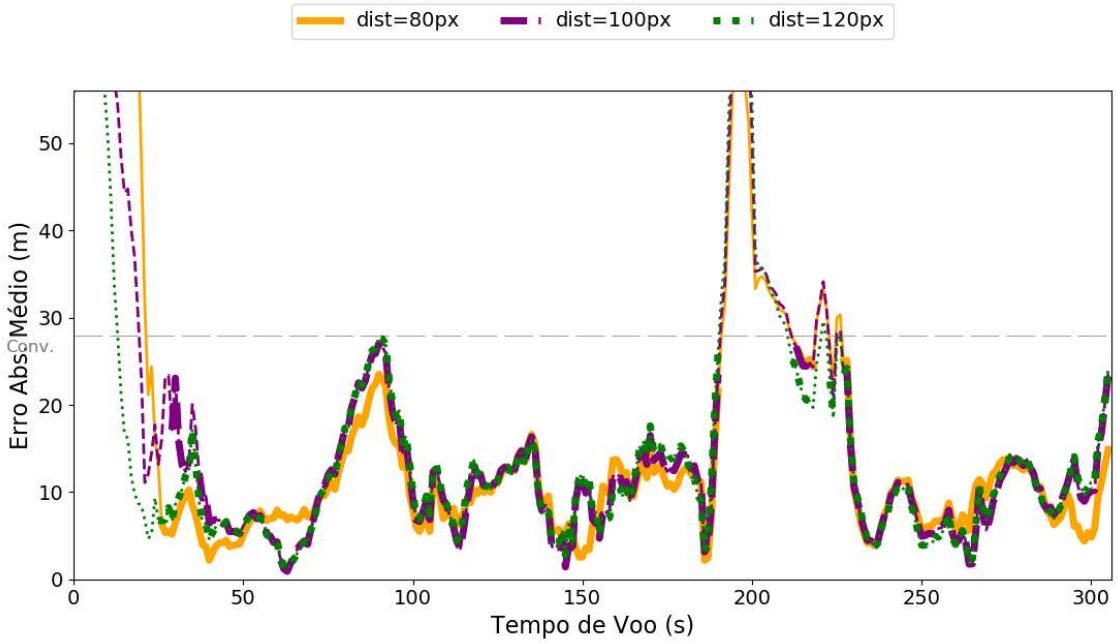


Figura 6.9 – Comparações de resultados obtidos com o método proposto no Ambiente 3, considerando três limites diferentes para a transformada de distância: 80, 100 e 120 pixels.

sequentemente, da altura exata do VANT. Ao contrário, é suficiente saber se o ponto está mais próximo do prédio do que outro ponto na mesma imagem, e essa relação de magnitude entre as distâncias é mantida mesmo que a escala esteja ligeiramente imprecisa (garantindo que o limite de distância seja suficientemente grande). Dito isso, L = 100 e L = 120 tiveram um desempenho ligeiramente melhor do que L = 80; portanto, fixamos L como 100 para comparar com outros métodos.

Um ponto importante de se analisar é que apesar dos voos no Ambiente 1 e Ambiente 2 terem sido feitos na mesma região, o erro obtido no segundo ambiente é muito maior do que no primeiro. Um dos fatores principais é a ausência de gimbal no segundo voo, que torna as imagens obtidas pelo drone muito mais instáveis, dificultando consideravelmente o cenário de teste. Com isso a odometria visual piora bastante e a performance do filtro de partículas também tende a piorar. No Ambiente 3, apesar de também não ser usado gimbal para estabilização das imagens, o erro é menor que no Ambiente 2. No entanto, isso decorre do fato do voo acontecer numa altitude bem mais baixa e cobrir uma área pequena.

6.3 Comparação com outras abordagens

As comparações entre nosso método, BRIEF-EB, Razão de Construção e abBRIEF são mostradas na Tabela 6.3. Nosso método teve um desempenho melhor do que os outros três métodos em todos os três cenários. Nosso método alcançou um erro após a convergência de 9,16 metros no primeiro voo, contra 39,09 metros do abBRIEF, o único dos três outros métodos com boa convergência. Este foi o mesmo conjunto de dados usado por Mantelli et al. (2019), portanto, esperava-se que o abBRIEF funcionasse bem. No segundo voo nosso método teve um erro após a convergência de 57,57 metros sendo 13,4% menor que o único método que convergiu, já no terceiro o erro após a convergência foi de 10,84 metros, 33,3% menor do que o método que convergiu.

Tabela 6.3 – Comparação com outros métodos

		Métodos			
	Métrica	BRIEF-EB	Razão de Construção	abBRIEF	Proposto L=100
Voo 1	EAM (m)	263.37	372.49	39.09	**11.26**
	EAM Conv. (m)	–	–	30.75	**9.16**
	Conv. correta (%)	0	4.49	84.96	**95.19**
	Conv. errada (%)	**0**	5.19	0.56	0.02
	Não conv. (%)	100	90.32	14.48	**4.79**
Voo 2	EAM (m)	160.36	105.46	336.27	**61.48**
	EAM conv. (m)	–	66.55	–	**57.57**
	Conv. correta (%)	0	46.21	0.26	**71.75**
	Conv. errada (%)	**0**	3.50	0.62	1.04
	Não conv. (%)	100	50.29	99.12	**27.21**
Voo 3	EAM (m)	86.15	333.00	107.13	**17.15**
	EAM conv. (m)	–	16.26	–	**10.84**
	Conv. correta (%)	0	16.11	4.03	**56.81**
	Conv. errada (%)	**0**	5.80	1.24	4.34
	Não conv. (%)	100	78.09	94.73	**38.85**

As Figuras 6.10, 6.11 e 6.12 mostram o EAM dos quatro métodos nos voos 1, 2 e 3, respectivamente. No Voo 1, nosso método alcançou convergência mais cedo do que o método abBRIEF e manteve um erro baixo ao longo da trajetória. O método abBRIEF aumentou significativamente o erro após cerca de 300s de voo, onde um pedaço de vegetação muda de cor ao longo das estações. Os outros métodos não atingiram a convergência. Na Figura 6.4 mostramos os momentos iniciais e de convergência do filtro, e como as partícuals se mantém concetradas durante todo o trajeto.

A convergência mais rápida também foi alcançada por nosso método no segundo

voo. Neste cenário, o método de Razão de Construção e o nosso mantiveram o EAM abaixo do limiar de convergência na maior parte do tempo, mas em algumas áreas com um número baixo de prédios, o erro aumentou acima do limiar. Nesse voo o abBRIEF, método que nossa proposta usa como base, não atingiu a convergência.

Durante o terceiro voo, o método *building ratio* atingiu a convergência antes do nosso método. No entanto, em torno dos 100 segundos de voo, o método de razão de prédios começou a apresentar perdas, enquanto nosso método conseguiu manter o erro abaixo de 20 metros em grande parte do tempo. Antes dos 200 segundos, nosso método proposto aumentou o EAM, uma vez que utilizou apenas a odometria visual como guia, pois voou sobre uma área sem edifícios. No entanto, assim que retornou às áreas com construções, o EAM do nosso método caiu abaixo do limite escolhido para definir a convergência. Isso também pode ser visto na Figura 6.5, nela são mostrados os momentos iniciais de convergência, quando a incerteza aumenta devido a falta de informação e quando o filtro converge novamente.

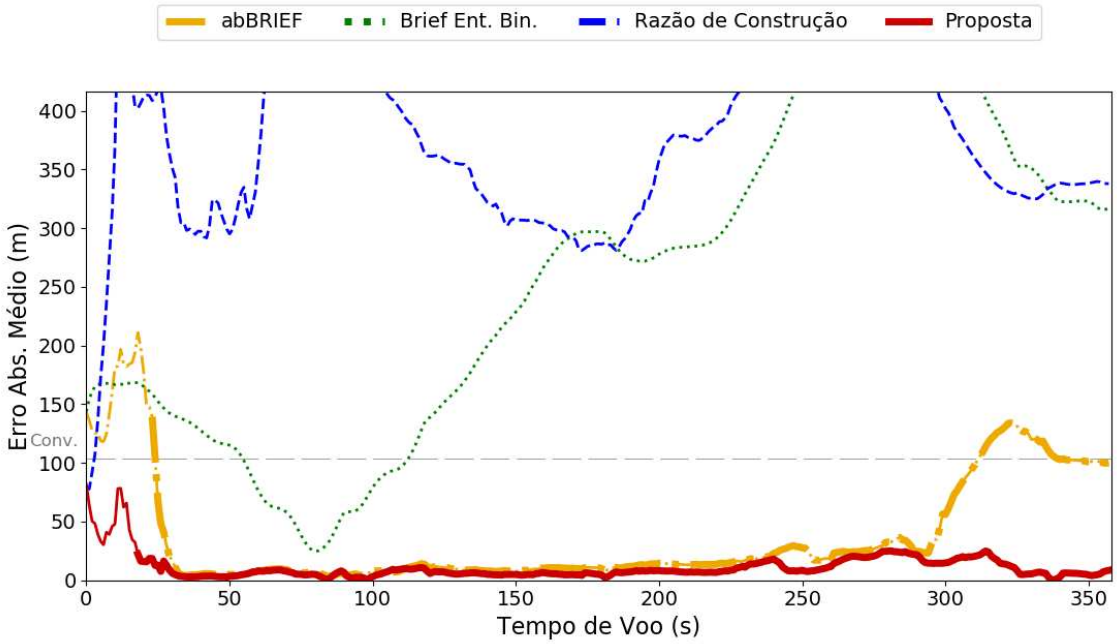


Figura 6.10 – Comparações do Erro Absoluto Médio (EAM) obtido com o método proposto e outras abordagens no **Ambiente 1**. A linha de erro é desenhada mais espessa quando o filtro de partículas converge para um único cluster. Nosso método manteve o EAM abaixo do limite de convergência na maior parte do tempo. O método abBRIEF também teve um bom desempenho, mas os demais métodos não.

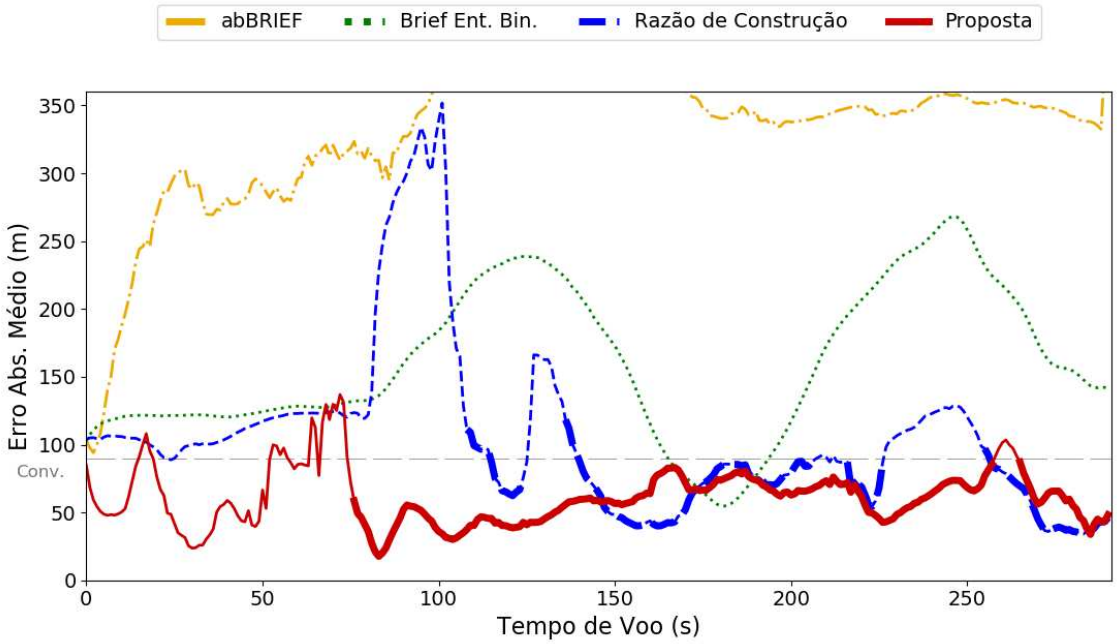


Figura 6.11 – Comparações do Erro Absoluto Médio (EAM) obtido com o método proposto e outras abordagens no **Ambiente 2**. A linha de erro é desenhada mais espessa quando o filtro de partículas converge para um único cluster. Nosso método manteve o EAM abaixo do limite de convergência na maior parte do tempo. Dessa vez, o método da razão de edifícios obteve bons resultados, mas os demais métodos não.

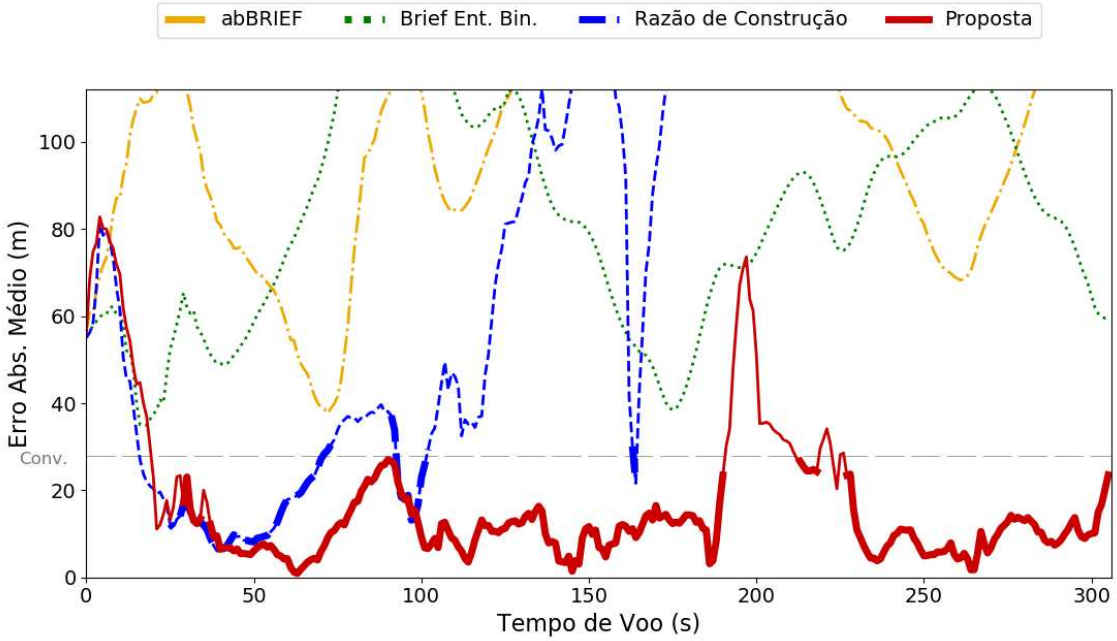


Figura 6.12 – Comparações do Erro Absoluto Médio (EAM) obtido com o método proposto e outras abordagens no **Ambiente 3**. A linha de erro é desenhada mais espessa quando o filtro de partículas converge para um único cluster. Nosso método manteve o EAM abaixo do limite de convergência na maior parte do tempo. Nenhum dos outros métodos convergiu adequadamente.

7 CONCLUSÃO

Neste trabalho, foi apresentado um sistema de localização de VANT baseado em visão computacional, o qual utiliza a presença de edifícios como fonte de informação, sendo menos mutável em relação às cores de uma imagem. O destaque do trabalho foi o desenvolvimento do descritor NBD-BRIEF, que emprega técnicas de *deep learning* para segmentar os prédios, aplicando em seguida uma transformada de distância para comparar as imagens capturadas pelo VANT e as do mapa de referência. Além disso, foi criado um *framework* de localização de VANTs baseado no NBD-BRIEF, proporcionando uma solução para a localização em áreas urbanas. Resumindo, as principais contribuições foram:

- a criação de um novo descritor, o NBD-BRIEF, que se baseia no BRIEF e utiliza a informação de proximidade de prédios para descrever uma imagem capturada pelo VANT e fazer o matching com o mapa de referência.
- um *framework* para a localização precisa de VANTs em áreas urbanas, que utiliza imagens RGB e um mapa de referência com os contornos dos prédios para a localização precisa do VANT em áreas urbanas.

Testamos nossa proposta em cenários desafiadores, com distâncias percorridas sempre superiores a 1 km e altitudes variando de 35m a 180m - situações em que virtualmente nenhuma outra abordagem testada funciona consistentemente bem.

O método proposto apresentou baixo erro médio ao longo dos experimentos, em dois deles o erro médio após a convergência ficou próximo dos dez metros, superando as abordagens concorrentes, como o método abBRIEF, que nos inspirou a desenvolver este trabalho. No voo 1, foi possível observar a invariância frente a diferentes estações do ano e cores da vegetação. Enquanto uma área com uma cor diferente causou divergência no método abBRIEF, nosso método proposto apenas aumentou a incerteza pela falta de construções, que diminuiu com o término da área sem prédios.

Na estratégia proposta, utilizamos apenas informações sobre proximidade com prédios, o que se mostrou adequado para aplicações em regiões urbanas. No entanto, como podemos ver no voo 3, a baixa contagem de prédios em alguns cenários pode ser um problema: a ausência de prédios em um dado local deixa o método às cegas naquela região. Outro problema que pode acontecer está relacionado a prédios muito próximos, que poderiam ser identificados como um único prédio. Tal situação impactaria

drasticamente na distância para a borda do prédio mais próximo, o que tornaria errada a avaliação da observação real em comparação a observação esperada. Felizmente esse foi um tipo de situação que não se mostrou presente nos ambientes testados, mas seria interessante investigar mais tal questão em trabalhos futuros. Além disso, pretendemos investigar o uso de outras classes semânticas, como estradas e vegetação alta ou baixa. O uso de mais classes pode melhorar a localização e diminuir a dependência de apenas um tipo de informação, e aumentando o ambiente de funcionamento do método para fora do ambiente urbano.

Outro ponto a ser melhor investigado é a aplicação da técnica proposta embarcada em VANTs. Os testes feitos neste trabalho foram executados *offline* em um computador, onde a capacidade de processamento computacional é maior do que em uma plataforma embarcada, portanto não se analisou a performance da proposta em termos de tempo de processamento. Vale destacar que as partes mais custosas do método, como a aplicação da segmentação usando *deep learning*, são feitas apenas sobre a imagem do VANT e não sobre a imagem das partículas, pois estas usam a informação de classificação de prédios pré-existente no mapa global. De qualquer forma, vale analisar com mais cuidado o quanto essa parte mais custosa do método, que na estratégia atual é computada uma vez a cada segundo, precisará ser adaptada em uma situação de restrição de recursos.

REFERÊNCIAS

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. v. 3951, cap. Lecture Notes in Computer Science, p. 404–417. ISBN 978-3-540-33832-1.

BIANCHI, M.; BARFOOT, T. D. Uav localization using autoencoded satellite images. **IEEE Robotics and Automation Letters**, v. 6, n. 2, p. 1761–1768, 2021.

BURGARD, W. et al. Integrating global position estimation and position tracking for mobile robots: the dynamic markov localization approach. In: **Proceedings of the 1998 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)**. Piscataway, NJ, USA: IEEE Press, 1998. v. 2, p. 730–735.

CABALLERO, F. et al. Improving vision-based planar motion estimation for unmanned aerial vehicles through online mosaicing. In: IEEE. **2006 IEEE ICRA**. [S.l.], 2006. p. 2860–2865.

CALONDER, M. et al. Brief: Binary robust independent elementary features. In: DANIILIDIS, K.; MARAGOS, P.; PARAGIOS, N. (Ed.). **Computer Vision – ECCV 2010**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010. p. 778–792. ISBN 978-3-642-15561-1.

CANNY, J. A computational approach to edge detection. **Transac. on pattern analysis and machine intelligence**, IEEE, n. 6, p. 679–698, 1986.

CHOI, J.; MYUNG, H. Brm localization: Uav localization in gnss-denied environments based on matching of numerical map and uav images. In: **IEEE/RSJ IROS 2020**. [S.l.: s.n.], 2020. p. 4537–4544.

CHOI, S. H.; PARK, C. G. Image-based monte-carlo localization with information allocation logic to mitigate shadow effect. **IEEE Access**, v. 8, p. 213447–213459, 2020.

CONTE, G.; DOHERTY, P. An integrated uav navigation system based on aerial image matching. In: IEEE. **Aerospace Conf.** [S.l.], 2008. p. 1–10.

COSTA, F. G. et al. The use of unmanned aerial vehicles and wireless sensor network in agricultural applications. In: **2012 IEEE International Geoscience and Remote Sensing Symposium**. [S.l.: s.n.], 2012. p. 5045–5048.

COUTURIER, A.; AKHLOUFI, M. A. A review on absolute visual localization for uav. **Robotics and Autonomous Systems**, v. 135, p. 103666, 2021. ISSN 0921-8890.

DELLAERT, F. et al. Monte carlo localization for mobile robots. In: **Proceedings of the 1999 IEEE International Conference on Robotics and Automation (ICRA)**. Piscataway, NJ, USA: IEEE Press, 1999.

DrivenData. **Open Cities AI Challenge: Segmenting Buildings for Disaster Resilience**. 2020. <https://github.com/drivendataorg/open-cities-ai-challenge/>. Accessed: 2021-07-29.

GFDRR Labs. **Open Cities AI Challenge Dataset, Version 1.0**. 2020. Radiant MLHub <https://doi.org/10.34911/rdnt.f94cxb>. Accessed: 2021-07-29.

KIM, D.-K.; WALTER, M. R. Satellite image-based localization via learned embeddings. In: **2017 IEEE International Conference on Robotics and Automation (ICRA)**. [S.l.: s.n.], 2017. p. 2073–2080.

LEONARD, J. J.; DURRANT-WHYTE, H. F. Mobile robot localization by tracking geometric beacons. **IEEE Transactions on Robotics and Automation**, v. 7, n. 3, p. 376–382, Jun 1991. ISSN 1042-296X.

LI, Y. et al. Road-network-based fast geolocalization. **IEEE Transactions on Geoscience and Remote Sensing**, v. 59, n. 7, p. 6065–6076, 2021.

LOWE, D. G. Distinctive image features from scale-invariant keypoints. **International Journal of Computer Vision**, Springer Netherlands, Dordrecht, Netherlands, v. 60, n. 2, p. 91–110, 2004. ISSN 0920-5691.

MAFFEI, R. **Translating sensor measurements into texts for localization and mapping with mobile robots**. Tese (Doutorado), Porto Alegre, Brazil, 2017.

MAFFEI, R. **Slides Aula 18 - Introdução a Localização, Robótica II, INF-UFRGS**. 2022. Accessed: 2022–11-06.

MAFFEI, R. et al. Fast monte carlo localization using spatial density information. In: **Proceedings of the 2015 IEEE International Conference on Robotics and Automation (ICRA)**. Piscataway, NJ, USA: IEEE Press, 2015. p. 6352–6358.

MAJDIK, A. L. et al. Air-ground matching: Appearance-based gps-denied urban localization of micro aerial vehicles. **Journal of Field Robotics**, v. 32, n. 7, p. 1015–1039, 2015.

MAKARENKO, A. A. et al. An experiment in integrated exploration. In: **Proceedings of the 2002 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)**. Piscataway, NJ, USA: IEEE Press, 2002. v. 1, p. 534–539.

MANTELLI, M. et al. A novel measurement model based on abbrief for global localization of a uav over satellite images. **Robotics and Autonomous Systems**, v. 112, p. 304–319, 2019. ISSN 0921-8890.

MASSELLI, A.; HANTEN, R.; ZELL, A. Localization of unmanned aerial vehicles using terrain classification from aerial images. In: MENEGATTI, E. et al. (Ed.). **Intelligent Autonomous Systems 13**. Cham: Springer International Publishing, 2016. p. 831–842. ISBN 978-3-319-08338-4.

MONTEMERLO, M.; THRUN, S. **FastSLAM: A Scalable Method for the Simultaneous Localization and Mapping Problem in Robotics**. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2007. (Springer Tracts in Advanced Robotics). ISBN 3540463992.

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.

MURPHY, R. R. **An Introduction to AI Robotics (Intelligent Robotics and Autonomous Agents)**. [S.l.]: The MIT Press, 2000. ISBN 0262133830.

PATEL, B.; BARFOOT, T. D.; SCHOELLIG, A. P. Visual localization with google earth images for robust global pose estimation of uavs. In: **2020 IEEE International Conference on Robotics and Automation (ICRA)**. [S.l.: s.n.], 2020. p. 6491–6497.

RONNEBERGER, O.; FISCHER, P.; BROX, T. U-net: Convolutional networks for biomedical image segmentation. In: NAVAB, N. et al. (Ed.). **MICCAI 2015**. Cham: Springer, 2015. p. 234–241. ISBN 978-3-319-24574-4.

ROSTEN, E.; DRUMMOND, T. Machine learning for high-speed corner detection. In: LEONARDIS, A.; BISCHOF, H.; PINZ, A. (Ed.). **Computer Vision – ECCV 2006**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006. p. 430–443. ISBN 978-3-540-33833-8.

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.

SERNA, J. G. et al. A review of current approaches for uav autonomous mission planning for mars biosignatures detection. In: **IEEE Aerospace Conf.** [S.l.: s.n.], 2020. p. 1–15.

SHAN, M. et al. Google map aided visual navigation for uavs in gps-denied environment. In: **2015 IEEE ROBIO**. [S.l.: s.n.], 2015. p. 114–119.

SHETTY, A.; GAO, G. X. Uav pose estimation using cross-view geolocalization with satellite imagery. In: **2019 International Conference on Robotics and Automation (ICRA)**. [S.l.: s.n.], 2019. p. 1827–1833.

SZELISKI, R. **Computer Vision: Algorithms and Applications**. 1st. ed. New York, NY, USA: Springer-Verlag New York, Inc., 2010. ISBN 1848829345, 9781848829343.

THIELS, C. A. et al. Use of unmanned aerial vehicles for medical product transport. In: . [S.l.: s.n.], 2015. v. 34, n. 2, p. 104–108.

THRUN, S. Probabilistic robotics. **Commun. ACM**, Association for Computing Machinery, New York, NY, USA, v. 45, n. 3, p. 52–57, mar 2002. ISSN 0001-0782.

THRUN, S.; BURGARD, W.; FOX, D. **Probabilistic Robotics**. Cambridge, MA, USA: MIT Press, 2005. (Intelligent robotics and autonomous agents). ISBN 9780262201629. Disponível em: <http://www.amazon.com/exec/obidos/redirect?tag=citeulike07-20&path=ASIN/0262201623>.

VISWANATHAN, A.; PIRES, B. R.; HUBER, D. Vision-based robot localization across seasons and in remote locations. In: IEEE. **IEEE ICRA**. [S.l.], 2016. p. 4815–4821.

YOL, A. et al. Vision-based absolute localization for unmanned aerial vehicles. In: **2014 IEEE/RSJ International Conference on Intelligent Robots and Systems**. [S.l.: s.n.], 2014. p. 3429–3434.