

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

MATHIAS FASSINI MANTELLI

**Um novo modelo de observação para  
o algoritmo de Monte Carlo aplicado  
ao problema de localização global de  
VANTs sobre imagens de satélite**

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

Orientador: Prof<sup>a</sup>. Dr<sup>a</sup>. Mariana Luderitz  
Kolberg

Porto Alegre  
2017

## CIP — CATALOGAÇÃO NA PUBLICAÇÃO

Mantelli, Mathias Fassini

Um novo modelo de observação para o algoritmo de Monte Carlo aplicado ao problema de localização global de VANTs sobre imagens de satélite / Mathias Fassini Mantelli. – Porto Alegre: PPGC da UFRGS, 2017.

85 f.: il.

Dissertação (mestrado) – Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Computação, Porto Alegre, BR-RS, 2017. Orientador: Mariana Luderitz Kolberg.

1. VANT, MCL, BRIEF, localização, imagens. I. Kolberg, Mariana Luderitz. II. Título.

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitor: Prof. Rui Vicente Oppermann

Vice-Reitora: Prof<sup>a</sup>. Jane Fraga Tutikian

Pró-Reitor de Pós-Graduação: Prof. Celso Giannetti Loureiro Chaves

Diretora do Instituto de Informática: Prof<sup>a</sup>. Carla Maria Dal Sasso Freitas

Coordenador do PPGC: Prof. Luigi Carro

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

*“Intelligence is the ability  
to adapt to change.”*

— STEPHEN HAWKING

## AGRADECIMENTOS

Inicialmente, agradeço minha família. Meus pais e meus irmãos, pela força que me deram, pelas palavras de motivação e carinho não apenas durante este trabalho, mas durante toda minha vida.

Agradeço à minha orientadora, Mariana Kolberg. Obrigado pela oportunidade que me ofereceu, por ter acreditado que eu seria capaz e por toda a ajuda que me dispensou durante estes dois anos.

Ao professor Edson Prestes, o mestre deve ser seguido e o conselheiro deve ser ouvido. Obrigado por ter sido, para mim, um grande Mestre-Conselheiro desde o começo.

A família Phi, agradeço toda a ajuda, atenção e momentos que foram compartilhados comigo. Em especial, agradeço o Diego e a Renata por todo auxílio e co-orientação. Vocês dois, além de amigos de laboratório, se tornaram amigos para toda a vida, os quais tenho grande apreço e admiração.

Por fim, agradeço todos aqueles que me ajudaram de uma forma ou de outra, estando perto ou longe, física ou virtualmente.

## RESUMO

A cada dia novos modelos de Veículos Aéreos Não Tripulados (VANTs) estão sendo lançados no mercado para serem utilizados em diversas aplicações, tais como mapeamento de ambientes e vigilância. Geralmente, estes robôs utilizam um sensor GPS como fonte de estimativa de localização. Contudo, para um bom funcionamento, este sensor depende de um número mínimo de satélites sincronizados com ele e que o sinal emitido pelos satélites seja recebido com boa qualidade, o que pode ser considerado um fator negativo. Uma forma de contornar este problema é empregar um sistema de localização baseado em visão computacional utilizando a câmera que já está embarcada no robô e imagens de satélite como mapa. Este sistema estima a localização do VANT através de comparações entre as imagens capturadas por ele e uma imagem de satélite, buscando encontrar a sua posição nesta imagem de satélite. Neste contexto, apresentamos uma variação do descritor BRIEF, o abBRIEF, para ser utilizado em um novo modelo de observação que também está sendo proposto. O modelo de observação é responsável por medir quão parecidas são as leituras do robô com diversas partes do mapa, para estimar a sua localização correta. Devido ao grande número de comparações necessárias, é importante que o descritor utilizado no processo seja rápido, não consuma muitos recursos computacionais e seja robusto para lidar com as várias diferenças entre as imagens. O modelo proposto foi utilizado no algoritmo de Monte Carlo (Monte Carlo Localization, MCL) para realizar a localização de VANTs e apresentou resultados satisfatórios que corroboram a eficácia do modelo e do descritor.

**Palavras-chave:** VANT, MCL, BRIEF, localização, imagens.

## ABSTRACT

New models of Unmanned Aerial Vehicles (UAVs) are developed on a daily basis and applied to several tasks, such as mapping terrains and surveillance. GPS sensors are usually the main source of information to estimate the robot position, however, a downside of these sensors is that a minimum number of satellites must be available and emitting high quality signals. A vision-based system can be used to overcome this problem by using a robot embedded camera and satellite images as maps. Computational vision systems estimate the UAV localization through the comparison between the robot extracted image and several parts of the satellite image. These comparisons are performed in order to localize the part of the map which is most similar to the robot perspective. Taking into account all this information, we propose BRIEF descriptor variation, called abBRIEF, to be used on a novel observation model, also proposed in this master thesis. An observation model is responsible for evaluate how similar the robot measurements and the different map parts are. The used descriptor must run fast, consume low computational resources and be robust regarding image changes, all to compensate the large number of necessary comparisons. The proposed model is applied with Monte Carlo Localization (MCL) method to the auto localization of UAVs and presented solid results that corroborate the model and descriptor efficiency.

**Keywords:** UAV, MCL, BRIEF, localization, images.

## LISTA DE ABREVIATURAS E SIGLAS

VANT	Veículo Aéreo Não Tripulado
SLAM	Simultaneous Localization and Mapping
GPS	Global Positioning System
INS	Inertial Sensor
GIS	Geographical Information Systems
QGIS	Quantum Geographic Information System
MCL	Monte Carlo Localization
BRIEF	Binary Robust Independent Elementary Features
DoF	Degree of Freedom
SIR	Sampling Importance Resampling
SIFT	Scale-Invariant Feature Transform
SURF	Speeded Up Robust Features
ORB	Oriented FAST and Rotated BRIEF
PCA	Principal Component Analysis
LDA	Linear Discriminant Analysis
MI	Mutual Information
CIE	Comission Internationale de l'éclairage
IMU	Inertial Measurement Unit
AdM	Arrio do Meio
UFRGS	Universidade Federal do Rio Grande do Sul
FLANN	Fast Library for Approximation Nearest Neighbors
RANSAC	Random Sample Consensus
NCC	Normalized Cross Correlation
OpenCV	Open Source Computer Vision

## LISTA DE FIGURAS

Figura 1.1 Ilustração do problema de localização global de VANTs.....	16
Figura 2.1 Exemplos de modelo de movimento. ....	22
Figura 2.2 Modelo de localização de robôs móveis. ....	25
Figura 2.3 Exemplos de aproximação de distribuições arbitrárias.....	27
Figura 2.4 Exemplo do processo SIR no filtro de partículas.....	28
Figura 2.5 Exemplo de localização utilizando MCL. ....	30
Figura 2.6 Processo de construção do descritor BRIEF. ....	33
Figura 2.7 Representação do processo de localização proposto por Yol et al. (2014). ....	34
Figura 2.8 Representação do processo de localização proposto por Merino et al. (2006). ....	36
Figura 2.9 Esquema de localização global de um VANT proposto por Majdik et al. (2015) .....	37
Figura 3.1 Sistema do espaço de cor CIE $L^*a^*b^*$ .....	40
Figura 3.2 Exemplo do modelo de observação proposto nesta dissertação. ....	43
Figura 3.3 Exemplo dos pares de pixels do descritor abBRIEF aplicado às partículas.....	45
Figura 4.1 Trajetórias dos vôos utilizados durante os testes deste trabalho.....	48
Figura 4.2 Exemplos de variações entre os mapas e a imagem capturada pelo VANT.....	49
Figura 4.3 Exemplo de correspondência entre <i>feature</i> . ....	50
Figura 4.4 Teste de correspondência de <i>features</i> usando uma imagem captu- rada pelo VANT e uma de satélite. ....	51
Figura 4.5 Comparação entre número de partículas e quantidade de pares de pixels. ....	59
Figura 4.6 Resultados de localização para 10.000 e 20.000 partículas. ....	60
Figura 4.7 Estimativa de localização no vôo <i>Flight 1</i> . ....	62
Figura 4.8 Imagem da trajetória <i>Flight 1</i> com sua respectiva imagem do mapa <i>AdM- 03-2014</i> . ....	63
Figura 4.9 Estimativa de localização no vôo <i>Flight 1</i> com 50.000 partículas e 512 pares de pixels.....	63
Figura 4.10 Estimativa de localização no vôo <i>Flight 2</i> . ....	64
Figura 4.11 Imagem da trajetória <i>Flight 2</i> com sua respectiva imagem do mapa <i>AdM-01-2014</i> . ....	64
Figura 4.12 Estimativa de localização no vôo <i>Flight 3</i> . ....	65
Figura 4.13 Imagem da trajetória <i>Flight 3</i> com sua respectiva imagem do mapa <i>UFRGS-05-2015</i> . ....	66
Figura 4.14 Desvio padrão das três trajetórias.....	67
Figura 4.15 Erros dos ângulos estimados pelo MCL.....	70
Figura 4.16 Demonstração qualitativa da diferença entre o <i>Ground-Truth</i> , odo- metria visual e localização estimada.....	71
Figura B.1 Mapas de Arroio do Meio-RS. ....	84
Figura B.2 Mapas da Universidade Federal do Rio Grande do Sul, Porto Alegre- RS.....	85

## LISTA DE TABELAS

Tabela 4.1	Análise do desempenho de métodos de comparação de imagens.....	54
Tabela 4.2	Tempo gasto por cada método para comparar duas imagens. ....	56

## LISTA DE ALGORITMOS

1 Algoritmo de Localização de Monte Carlo .....	29
---	----

## SUMÁRIO

1 INTRODUÇÃO .....	12
1.1 Motivação .....	13
1.2 Objetivos .....	17
1.3 Organização .....	18
2 FUNDAMENTAÇÃO TEÓRICA .....	20
2.1 Definições e Notações .....	20
2.2 Auto Localização de robôs móveis .....	23
2.3 Filtro de Partículas para Localização de Robôs Móveis .....	26
2.4 Binary Robust Independent Elementary Features - BRIEF .....	30
2.5 Trabalhos relacionados .....	33
3 NOVO MODELO DE OBSERVAÇÃO BASEADO NO DESCRITOR ABBRIEF .....	38
3.1 Descritor abBRIEF .....	38
3.2 Modelo de observação baseado no descritor abBRIEF .....	40
3.3 Aplicação no MCL .....	43
4 EXPERIMENTOS E DISCUSSÃO .....	47
4.1 Especificação do ambiente de experimentação .....	48
4.2 Correspondência entre <i>Features</i> .....	50
4.3 Métodos de comparação de imagens .....	52
4.4 Número de partículas e quantidade de pares de pixels .....	56
4.5 Comparação com o BRIEF original .....	61
4.6 Estimativa do ângulo .....	68
5 CONCLUSÃO .....	72
5.1 Trabalhos Futuros .....	75
REFERÊNCIAS .....	77
APÊNDICE A — TEORIA DA PROBABILIDADE .....	81
A.1 Probabilidade de uma variável aleatória .....	81
A.2 Regra do Produto .....	81
A.3 Probabilidade Condicional .....	81
A.4 Independência .....	82
A.5 Regra de Bayes .....	82
A.5.1 Probabilidade <i>a priori</i> .....	82
A.5.2 Probabilidade <i>a posteriori</i> .....	83
A.6 Lei da Probabilidade Total .....	83
A.7 Suposição de Markov .....	83
APÊNDICE B — IMAGENS .....	84
B.1 Arroio do Meio-RS .....	84
B.2 Universidade Federal do Rio Grande do Sul, Porto Alegre-RS .....	85

## 1 INTRODUÇÃO

As últimas décadas têm sido palco de uma série de avanços na área da robótica, principalmente pela evolução nas áreas de *hardware* e *software*. Os computadores e outros dispositivos se tornaram menores, mais acessíveis e com maior recurso computacional disponível. Dessa forma, novos algoritmos puderam ser desenvolvidos em decorrência da maior quantidade e precisão das informações providas por estes novos recursos de *hardware* (ROMERO et al., 2014). Estes avanços permitem o uso de robôs em tarefas que eram inimagináveis no passado, como o limpeza de residências e piscinas. Atualmente já existem, inclusive, robôs adaptados para operarem em mais de um tipo de ambiente, conhecidos como híbridos (YANG et al., 2015; SIDDALL; KOVAC, 2014; CHEN et al., 2015).

Independente do tipo de ambiente da aplicação, quando o robô executa tarefas mais elaboradas em locais extensos, não controlados e complexos, as arquiteturas de *software* frequentemente utilizadas usam representações internas do local como mapas, um sistema de localização e um outro sistema que corresponde à aplicação, tais como mapeamento ou planejamento de caminhos (ROMERO et al., 2014).

O sistema de localização busca resolver o problema de localização do robô e desempenha um papel importante em várias tarefas executadas por eles, tendo como exceção, por exemplo, tarefas em que o robô apenas precisa reagir a acontecimentos do ambiente e não é necessário um planejamento de suas ações (THRUN; BURGARD; FOX, 2005). Neste problema o objetivo é estimar a localização do robô no mapa comparando as leituras dos seus sensores com os elementos presentes neste mapa em determinadas posições. Contudo, dificuldades como leituras dos sensores carregadas de incertezas, mapas desatualizados e regiões ambíguas (longos corredores e ambientes simétricos, por exemplo) são encontradas neste problema e precisam ser tratadas. Para lidar com estes desafios e construir um sistema de localização robusto, é necessário utilizar as leituras dos sensores a fim de obter informações do ambiente e reduzir o erro de estimação da postura do robô.

Esta dissertação lida com o problema de localização de um Veículo Aéreo Não Tripulado (VANT) equipado com uma câmera<sup>1</sup> RGB operando em ambientes externos. Apesar de existirem trabalhos na literatura (JAYATILLEKE; ZHANG, 2013; YOL et al., 2014; KONOMURA; HORI, 2014; MAJDIK et al., 2015; MASSELLI;

---

<sup>1</sup>A não ser que expreso no texto, sempre que mencionado que uma câmera está embarcada em um VANT, considere que ela está perpendicular a este, apontada para baixo.

HANTEN; ZELL, 2016), ainda há espaço para melhorias, tais como sistemas de localização que não dependam de métodos de aprendizagem de máquina, que reduzam o consumo de recursos computacionais e tempo de processamento, que exijam uma menor quantidade de informações iniciais e que não necessitem de alterações no ambiente para seu funcionamento.

## 1.1 Motivação

Existem três problemas fundamentais na robótica: localização, controle de movimento e mapeamento (MAKARENKO et al., 2002). No problema de localização, o robô deve ser capaz de usar as leituras providas pelos seus sensores para estimar sua postura em uma mapa do ambiente. Esse problema em particular tem duas subdivisões. Quando o robô conhece sua postura inicial no mapa, o problema é chamado *localização local*. Por outro lado, quando a postura inicial do robô no mapa é desconhecida, o problema é dito *localização global*.

O problema de controle de movimento, ou navegação, tem o objetivo de guiar o robô pelo ambiente desviando de obstáculos, partindo de sua postura atual e levando-o até o destino. O planejamento dos movimentos do robô pode ser voltado para caminhos mais curtos ou mais seguros, dependendo da aplicação. Por isso, se faz necessário um mapa preciso do ambiente e o conhecimento da postura do robô para estabelecer a melhor opção.

O conhecimento sobre a postura correta do robô também é essencial para o tratamento do problema de mapeamento de ambientes. Este problema se resume na construção de uma representação adequada do ambiente, podendo esta ser a localização de obstáculos, de espaços livres ou áreas seguras para que o robô possa se locomover. Este problema seria mais fácil se não houvessem erros nas leituras dos sensores, mas isso não acontece na prática. Portanto, se faz necessário uma maior análise e tratamento destas leituras a fim de melhorar o processo de estimação do mapa.

Uma grande variedade de aplicações executadas por robôs resolve, em sua base, pelo menos um desses três problemas. Por exemplo, o aspirador de pó Roomba<sup>TM</sup>, da empresa iRobot, trata estes três problemas e um outro mais complexo (TAYLOR et al., 2004). Considerando que ele opera em residências que jamais esteve, este robô deve se deslocar por espaços livres e acessíveis, construir um mapa do ambiente e se

localizar neste mapa que está construindo para garantir uma limpeza eficiente. Ele é equipado com um algoritmo para tratar o problema de localização e mapeamento simultâneos (*Simultaneous Localization and Mapping*, SLAM), em que o robô deve construir o mapa do ambiente enquanto tenta se localizar simultaneamente. Isso é feito para garantir que o mapa do ambiente seja o mais coerente possível.

Como visto, apesar dos problemas de controle de movimento e mapeamento não serem o problema de localização em si, em aplicações reais eles dependem de uma boa estimativa da postura do robô para atingirem bons resultados, reforçando a dependência e importância do problema de localização de robôs.

No caso dos VANTs, a localização e navegação também são elementares para a execução de tarefas complexas (SENLET; ELGAMMAL, 2012). Os modelos de asas rotativas têm sido apresentados como uma tecnologia promissora em aplicações que requerem manobras difíceis e devem ser executadas em ambientes com terrenos desfavoráveis, onde existam aclives, declives e buracos (OLLERO; MERINO, 2004). Embora atualmente tenham pouca autonomia de vôo, sua liberdade de movimento e a independência em relação ao terreno têm chamado atenção de pesquisadores em todo o mundo. Hoje eles estão sendo usados em muitos tipos de aplicações, incluindo exploração, inspeção e modelagem de indústrias e monumentos históricos, entrega de comida ou medicamentos e vigilância (YOL et al., 2014; MAJDIK et al., 2015).

Em geral, estes veículos dependem de sensores como Sistema de Posicionamento Global (*Global Positioning System*, GPS) ou Inerciais (*Inertial Sensors*, INS) para estimar sua localização (ZHENG et al., 2014). Considerando que estas são as opções disponíveis para obterem a sua localização, a falha destes sensores pode ser catastrófica. Inclusive, como apontado por Carroll (2003) e Caballero et al. (2006), estes sensores possuem algumas fraquezas. Segundo eles, o número de satélites conectados ao GPS que está embarcado no VANT e a qualidade dos sinais emitidos por estes satélites são importantes para estimar uma boa localização. Como consequência, a posição estimada pelo GPS pode ser degradada pelo baixo número de satélites disponíveis ou por ruídos no sinal. Além disso, de acordo com Conte e Doherty (2008) e Viswanathan, Pires e Huber (2016), aparelhos que neutralizam o sinal de GPS são facilmente encontrados e seu uso pode fazer a postura estimada não ser confiável.

Como consequência, VANTs que se baseiam apenas em sinais de GPS para se localizarem podem perder sua localização e colocar o sucesso da aplicação em risco.

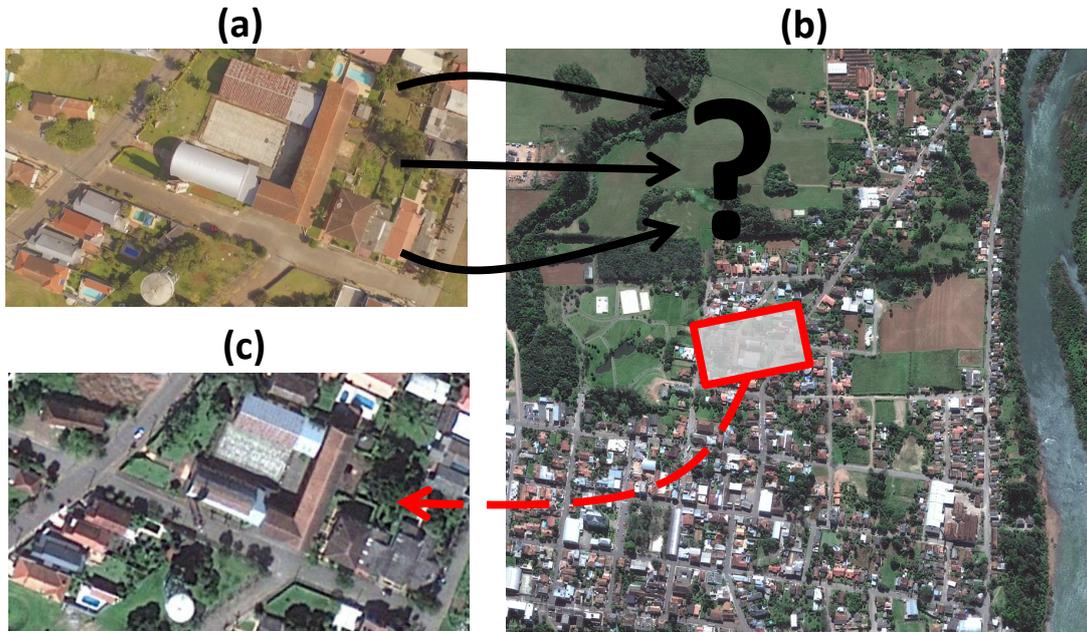
Em aplicações militares, por exemplo, em que veículos como estes são usados para monitorar um perímetro, o inimigo pode torná-lo inoperante através de interferências no sinal de GPS recebido, impossibilitando-o de se localizar. Uma solução em potencial para esta dependência do GPS por parte dos robôs é embarcar neles sistemas de localização baseados em visão computacional para estimar a sua postura, já que o funcionamento destes sistemas não depende de *hardware* externo, como satélites, por exemplo. Este tipo de sistema de localização tem recebido atenção da comunidade de pesquisa e produzido resultados promissores, mas que podem ser melhorados. Diante disso, o foco desta dissertação é o problema de localização baseado em visão computacional, especificamente a localização global de VANTs de asas rotativas utilizando imagens de satélite como mapas. A abordagem proposta neste trabalho pode ser utilizada como uma segunda fonte de estimativa de localização para estes veículos, no intuito de proporcionar um sistema redundante.

Atualmente muitos VANTs comercializados possuem uma câmera embarcada, podendo esta ser considerada um sensor padrão nestes robôs. A popularidade do uso de câmeras, neste caso, é justificada pelo seu tamanho, peso e custo, quando comparada com sensores laser, por exemplo. Além disso, as imagens podem ser utilizadas por mais de uma aplicação, como monitoramento de vegetações, determinar a saúde de pastagens e estratégias de pastoreio (RANGO et al., 2006).

Para o caso de sistema localização baseado em visão computacional, o processo mais comum é utilizar a câmera embarcada no VANT e determinar a postura das imagens tiradas por esta câmera em um mapa conhecido previamente. O mapa utilizado é, geralmente, uma imagem de satélite que abrange a área em que o robô irá sobrevoar. Este esquema é representado na Figura 1.1. Imagens de satélite podem ser encontradas através de diferentes fontes, incluindo Google<sup>TM</sup> Earth, Bing<sup>TM</sup> Maps, e também Sistemas de Informações Geográficas (*Geographical Information Systems*, GIS), como o QGIS (TEAM, 2009). Além disso, este tipo de imagem cobre boa parte do planeta, não se restringindo apenas a regiões urbanas. O acesso fácil e gratuito às imagens de satélite apoia a escolha desta abordagem de localização de VANTs, uma vez que existem diversas variedades de imagens e a ideal para a aplicação pode ser usada como mapa.

Apesar das vantagens destas câmeras embarcadas e a disponibilidade de mais de uma fonte de imagens de satélite, desenvolver um sistema de localização baseado em visão computacional é muito desafiador por diversas razões. As imagens de

Figura 1.1: Ilustração do problema de localização global de VANTs de asas rotativas baseado em visão computacional. (a) representa uma imagem capturada pela câmera embarcada no robô, (b) corresponde a imagem de satélite que é usada como mapa e (c) ilustra a parte do mapa que equivale a postura do robô. O objetivo é encontrar uma imagem menor, (c), na imagem de satélite, (b), que mais se assemelha a imagem capturada por ele, (a).



satélite normalmente não têm uma boa resolução e não são atualizadas com muita frequência. Ainda, imagens capturadas por estas câmeras podem ter diferenças quando comparadas com as imagens de satélite, como efeitos de projeção, condições de luminosidade e até mesmo modificações de curto e longo prazo no terreno, como é ilustrado pela Figura 1.1-(a) e Figura 1.1-(c). Então, quando se propõe um sistema como esse, é necessário garantir que ele seja robusto em relação a todas estas diferenças e desafios, sendo capaz de localizar o VANT.

Até aqui fica claro que a localização de um robô tem um papel importante em várias tarefas executadas por eles. Além disso, também está evidente que a dependência do GPS por parte dos VANTs como sistema de localização não é uma abordagem totalmente segura, sendo os sistemas de localização baseados em visão computacional uma alternativa viável. Então, a grande motivação para este trabalho de mestrado é propor um sistema de localização global para VANTs baseado em visão computacional. Não apenas isso, o sistema se apresenta como uma solução computacionalmente barata, rápida e que não necessita de alterações no ambiente para seu funcionamento.

## 1.2 Objetivos

Propõe-se neste trabalho um novo modelo de observação para ser incorporado a um sistema de localização global para VANTs que utiliza imagens de satélite como mapa. Quando pronto, ele poderá ser considerado como um sistema redundante de localização para estes veículos nos casos de falha do GPS, uma vez que este sensor é a principal fonte de localização utilizado até hoje nestes robôs.

A abordagem proposta é baseada no algoritmo de localização de Monte Carlo (*Monte Carlo Localization*, MCL) (DELLAERT et al., 1999). O MCL é um método de localização popular que representa as hipóteses sobre a localização do robô através de partículas (THRUN; BURGARD; FOX, 2005). Estas partículas simulam o comportamento do robô em diferentes posturas do mapa, a fim de descobrir em qual região deste mapa o robô está. As partículas se movimentam e fazem as leituras do ambiente da mesma maneira que o robô. Ele será discutido com mais detalhes no Capítulo 2. Para desenvolver o modelo de observação do MCL, propõe-se também uma variação do descritor *Binary Robust Independent Elementary Features*, BRIEF (CALONDER et al., 2010), denominada abBRIEF, tornando-o suficientemente robusto. O abBRIEF, assim como o BRIEF, é um descritor que representa um *patch*<sup>2</sup> da imagem como um vetor binário e é altamente discriminativo até mesmo com poucos bits. Este descritor é computado por um teste de diferença entre os valores de intensidade dos pixels de uma imagem.

O descritor BRIEF é variante a rotações e escalas, ou seja, se existem duas imagens iguais mas uma delas está rotacionada em 180° e com um fator de escala, o resultado do descritor BRIEF é que são imagens diferentes. Inicialmente, isso parece uma fraqueza ou falha, mas para este trabalho é totalmente adequado. Como já dito, no MCL muitas partículas são espalhadas pelo mapa para estimar qual é a postura do robô. No contexto de localização global de VANTs, a postura é composta pelas coordenadas 3D e o ângulo *yaw*, ou seja, 4 graus de liberdade (*Degrees of Freedom*, DoF). Partículas com postura mais similar à postura destes robôs receberão um peso maior, e o descritor BRIEF é capaz de pesar as partículas ao ponto de atribuir um peso alto apenas àquelas que estão com a postura similar a postura dele, e não apenas à posição similar.

---

<sup>2</sup>Um *patch* de uma imagem corresponde a uma parte da imagem que geralmente contém uma informação importante como um canto, uma borda ou um gradiente alto. Os *patches* têm tamanho fixo.

A abordagem proposta por Calonder et al. (2010) é contruir um descritor para cada *patch* da imagem utilizando apenas o valor de intensidade de cada pixel do *patch* para definir o valor do bit do descritor. Como mencionado anteriormente, neste trabalho propõe-se algumas mudanças no BRIEF, gerando o novo descritor abBRIEF, com objetivo de torná-lo mais robusto para o contexto de localização de VANTs, como utilizar as informações de cores da imagem, afim de gerar descritores mais singulares.

Sobre as vantagens da abordagem proposta neste trabalho em relação ao estado da arte em localização de VANTs utilizando imagens, pode-se citar: não é necessário uma etapa de treinamento *a priori* com as imagens do terreno, como proposto por Masselli, Hanten e Zell (2016); apenas parte dos pixels da imagem são utilizados pelo modelo de observação, diferente do modelo de Yol et al. (2014) que utiliza todos os pixels da imagem em seu modelo; não é necessário espalhar pontos de referência pelo ambiente, como a abordagem adotada por Jayatilleke e Zhang (2013); e as imagens de satélite que são utilizadas como mapa são encontradas com facilidade e não limitam os locais de aplicação, diferente do trabalho de Majdik et al. (2015) no qual depende de um tipo específico de imagens que não abrange muitas cidades do mundo e nem regiões não urbanas. A proposta do uso deste novo modelo de observação é que mesmo utilizando poucos pixels das imagens para reduzir o consumo de recursos computacionais e o tempo de processamento, os resultados sobre a estimativa da postura do VANT continuem sendo promissores.

Ainda sobre os resultados, dentre todos os métodos de comparações de imagens testados, o descritor abBRIEF se mostrou mais rápido que todos e com um bom desempenho em relação ao número de acertos nas comparações. Além disso, também é possível afirmar que as mudanças propostas no abBRIEF em relação ao BRIEF proporcionaram excelentes resultados. As estimativas da localização do VANT tiveram menos erros quando o abBRIEF foi utilizado no modelo de observação. De maneira geral, o sistema final foi capaz de estimar a localização do VANT em três trajetórias, utilizando cinco mapas e com variação de altura.

### 1.3 Organização

Esta dissertação é dividida como apresentado a seguir. O Capítulo 2 apresenta os métodos usados neste trabalho com mais detalhes e formalismo. Primeiro,

apresenta alguns conceitos importantes para o contexto do problema de localização de robôs móveis, seguido pelos desafios e variações deste problema. Ainda, apresenta o algoritmo MCL e o descritor BRIEF. Por fim, discute as principais abordagens já propostas para lidar com o problema de localização de VANTs.

No Capítulo 3 é apresentada a proposta deste trabalho. Inicialmente é descrito o abBRIEF, uma variação do descritor BRIEF. Na sequência, explica-se como este novo descritor é utilizado pelo modelo de observação proposto no MCL.

O Capítulo 4 apresenta e discute comparações feitas entre o abBRIEF e outros descritores. Também discute os testes e os resultados experimentais do modelo de observação proposto para o problema de localização global de VANTs. Não há testes provenientes de simulação, todos os resultados correspondem a vôos reais, abrangendo diferentes mapas, alturas e locais de vôo. Estes testes tentam cobrir os desafios existentes no problema abordado. Por fim, o Capítulo 5 contém nossas conclusões e o que pode ser feito no futuro.

## 2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo apresenta o problema de auto localização de robôs móveis, explica as duas principais técnicas utilizadas neste trabalho e aborda os trabalhos encontrados na literatura sobre localização de VANTs. Primeiro, na Seção 2.1 são introduzidos alguns conceitos básicos para o problema de localização e na Seção 2.2 é explorado as características e dificuldades deste problema. Então, na Seção 2.3 é explicado o MCL, uma implementação do filtro de partículas para o problema de localização de robôs móveis. O descritor BRIEF, que é a base para a proposta do abBRIEF, é apresentado na Seção 2.4. Por fim, na seção 2.5 é discutido alguns dos trabalhos de localização de VANTs utilizando imagens como mapa. Algumas leis elementares no contexto de teoria da probabilidade podem ser encontradas no Apêndice A deste trabalho.

### 2.1 Definições e Notações

Robôs móveis autônomos realizam suas tarefas sem interferência humana e para isso devem conhecer sua localização no ambiente. Seria fácil atender esta exigência se não houvessem erros nas leituras dos sensores do robô e o ambiente fosse totalmente estático, mas isso não acontece no mundo real. Por isso, o robô precisa usar seus sensores para estimar seu estado no ambiente (THRUN; BURGARD; FOX, 2005). Para estimar este estado, algumas informações são necessárias<sup>1</sup>:

- $\mathbf{x}_t$  é a postura do robô no instante  $t$ . É composta pela localização e orientação relativas ao sistema de coordenada global. Ainda, a postura também pode ser composta pelas três coordenadas cartesianas e pelos ângulos *roll*, *pitch*, e *yaw*, mas isso depende do robô utilizado e dos DoF que se deseja estimar na aplicação. O conjunto  $\mathbf{x}_{0:t} = \{\mathbf{x}_0, \mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_t\}$  representa a trajetória viajada pelo robô da postura inicial,  $\mathbf{x}_0$ , até a atual,  $\mathbf{x}_t$ .
- $\mathbf{u}_t$  é a informação de controle de movimento e corresponde a mudança da postura do robô no instante  $t-1$  para o instante  $t$ , normalmente dado por medidas de odometria. A sequência destas informações é representada por  $\mathbf{u}_{1:t} = \{\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3, \dots, \mathbf{u}_t\}$ .

---

<sup>1</sup>A notação que será usada nesta dissertação para se referir a componentes de variáveis ou vetores é: *componente(variável)*. Por exemplo, a coordenada  $y$  da postura  $\mathbf{x}$ , é dada por  $y(\mathbf{x})$ .

- $\mathbf{z}_t$  é a leitura do ambiente obtida pelo sensor do robô no instante  $t$ . Todas as leituras juntas geram um conjunto representado por  $\mathbf{z}_{1:t} = \{\mathbf{z}_1, \mathbf{z}_2, \mathbf{z}_3, \dots, \mathbf{z}_t\}$ .
- $\mathbf{m}$  é o mapa do ambiente dado previamente.

Os conjuntos  $\mathbf{u}_{1:t}$  e  $\mathbf{z}_{1:t}$  são estimados pelos sensores do robô ou por algoritmos<sup>2</sup>. Sobre o mapa  $\mathbf{m}$  e as posturas do robô  $\mathbf{x}_{0:t}$ , nem sempre é necessário estimá-los, depende do problema que se está tratando. Por exemplo, no problema de localização o mapa  $\mathbf{m}$  é conhecido, mas é necessário estimar o conjunto  $\mathbf{x}_{0:t}$ . Por outro lado, no problema de mapeamento, o que precisa ser estimado é o mapa  $\mathbf{m}$ , uma vez que o conjunto  $\mathbf{x}_{0:t}$  é conhecido. Diferente destes dois problemas, o SLAM busca estimar o mapa  $\mathbf{m}$  e o conjunto  $\mathbf{x}_{0:t}$  simultaneamente.

Além destas variáveis, no contexto de estimação do estado de um robô, os modelos de observação e o de movimento também são muito importantes. Esta importância está na forma de se trabalhar com a incerteza, i.e, propor um modelo probabilístico que trata, com acurácia, o tipo específico de incerteza que pode existir nos controles de movimento ou nas leituras, para ser usado nos algoritmos de localização (THRUN; BURGARD; FOX, 2005).

O **modelo de movimento** representa a probabilidade sobre o estado de transição de um robô de  $\mathbf{x}_{t-1}$  para  $\mathbf{x}_t$  com a execução de um controle de movimento  $\mathbf{u}_t$ , no mapa  $\mathbf{m}$ ,

$$p(\mathbf{x}_t \mid \mathbf{u}_t, \mathbf{x}_{t-1}, \mathbf{m}). \quad (2.1)$$

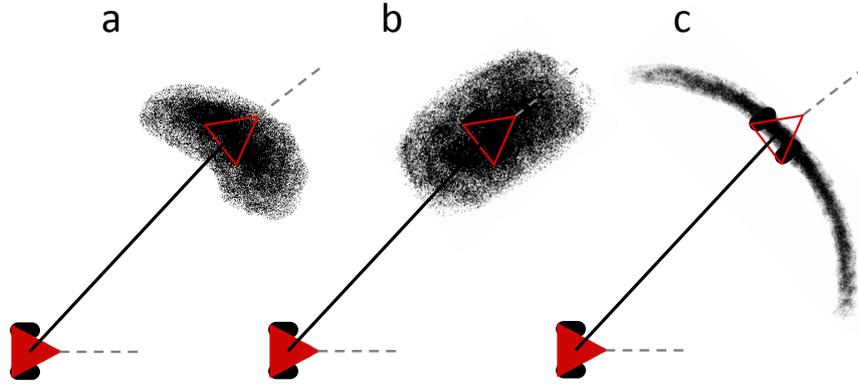
Em resumo, o modelo de movimento expressa a distribuição *a posteriori* sobre as configurações cinemáticas que um robô assume quando executa um controle de movimento  $\mathbf{u}_t$  no instante  $t - 1$ . Com essa informação, este modelo estima o movimento do robô levando em consideração as incertezas associadas a essa movimentação.

A Figura 2.1 representa três exemplos de modelos de movimento considerando diferentes parâmetros de incerteza de translação e de rotação. Apesar do movimento aplicado para todos os modelos ter sido o mesmo, estes parâmetros geraram resultados diferentes. Quanto mais preta uma localização, mais provável que o robô esteja nela (THRUN; BURGARD; FOX, 2005).

---

<sup>2</sup>Um exemplo de algoritmo é a Odometria Visual, que usa uma sequência de imagens para estimar a odometria de um robô (HOWARD, 2008; SCARAMUZZA; SIEGWART, 2008; WHELAN et al., 2013; FORSTER; PIZZOLI; SCARAMUZZA, 2014).

Figura 2.1: Exemplos de modelos de movimento baseado em medidas de odometria com variação de parâmetros de incerteza: (a) representa o modelo com erros de translação e rotação moderados, (b) muito erro de translação e (c) muito erro de rotação. O robô é representado pelo triângulo vermelho. Adaptado de (THRUN; BURGARD; FOX, 2005).



O **modelo de observação** compõe o outro modelo de domínio específico em robótica probabilística. Neste caso, o modelo define como as leituras do sensor são criadas no mundo real. A probabilidade condicional que define este modelo é dada pela leitura do sensor  $z_t$ , a postura do robô  $x_t$  no instante  $t$  e o mapa  $m$ ,

$$p(z_t | x_t, m). \quad (2.2)$$

O modelo de observação deve ser compatível com o sensor utilizado, uma vez que cada tipo de sensor requer um modelo de observação específico (CHOSSET et al., 2005). Os sensores de alcance como sonar e laser, por exemplo, são modelados como um cone e apenas um raio, respectivamente. Por outro lado, quando o sensor disponível é uma câmera ou outro tipo de sensor visual, é necessário modelar através de alguma correspondência entre imagens.

Este modelo de observação representa quão similar é a leitura do sensor com a postura atual do robô no mapa, ou seja, qual a probabilidade da leitura  $z_t$ , dada a postura  $x_t$ , no mapa  $m$  (LOWRY; MILFORD, 2015). Por exemplo, leituras de uma câmera (imagens) que está embarcada em um robô podem ser estimadas através da similaridade com a imagem que corresponde à leitura do robô no mapa  $m$  dada sua postura  $x_t$ . Esta comparação pode ser chamada de correspondência entre imagens.

A abordagem proposta por Lowry e Milford (2015) interpreta as informações visuais do ambiente, no contexto de localização de robôs, para gerar um modelo

de observação apropriado. Toda a abordagem proposta é baseada na ideia de que o ambiente pode sofrer drásticas mudanças em relação ao mapa, causando dois modos de operação possíveis: quando as condições entre o mapa e o ambiente atual são *similares*, e outro quando há *diferentes* condições entre ambos (por exemplo, as imagens que representam o mapa foram tiradas a noite, enquanto que o robô está tentando se localizar durante o dia). Então, no caso *similar*, a correspondência entre imagens é mais rigorosa e requer uma alta semelhança entre as imagens comparadas (uma imagem sendo a leitura,  $\mathbf{z}_t$ , e a outra sendo a imagem que corresponde a postura do robô,  $\mathbf{x}_t$ , no mapa  $\mathbf{m}$ ). Por outro lado, se uma correspondência rigorosa entre imagens é usada no caso *diferente*, não haverão imagens parecidas e o robô não será capaz de se localizar. Uma correspondência de imagens mais permissiva deveria ser usada neste caso.

## 2.2 Auto Localização de robôs móveis

O problema de auto localização de robôs móveis possui algumas características e conceitos que são importantes para o entendimento da sua complexidade e serão discutidos. Este é um problema básico no campo da robótica, podendo ser definido como a estimação da postura de um robô relativo a um mapa do ambiente previamente conhecido (THRUN; BURGARD; FOX, 2005). Apesar de parecer único, existem diferentes variações deste problema. A diferença pode estar no conhecimento inicial disponível ao robô, no ambiente, no número de robôs que estão simultaneamente realizando a auto localização e entre outros fatores.

O primeiro fator está relacionado ao conhecimento da postura inicial do robô, podendo gerar duas variações do problema de auto localização com diferentes níveis de dificuldade. A *localização local*, ou *rastreamento*, é a mais fácil, pois assume que a postura inicial do robô é conhecida. O erro inicial sobre a estimativa da localização do robô é baixo e é necessário, apenas, estimar a sua próxima postura de acordo com os controles de movimento e leituras dos sensores realizados por ele. Na *localização global* o robô não tem conhecimento sobre sua postura inicial e por isso ele deve se localizar considerando todo o mapa do ambiente. Por meio dos seus sensores, ele deve estimar sua postura considerando o mapa completo e reduzir o erro de estimativa que é alto desde o início. Ainda, apesar de não estar relacionada ao conhecimento da postura inicial do robô, outro problema é o problema do *robô raptado*. Nesse

caso, enquanto o robô se descola pelo ambiente, influências externas podem movê-lo para outro lugar sem que ele perceba. Ele deve notar que sua estimativa está errada e começar o processo de localização global, já que não sabe em que região do mapa foi parar.

O segundo é sobre o ambiente em que o robô vai operar, podendo ser ambiente *estático* ou *dinâmico*. Em ambientes *estáticos*, o robô é a única entidade que se move, e todo o resto permanece na mesma posição. O grau de dificuldade deste problema pode aumentar caso o mapa do ambiente esteja desatualizado. Por outro lado, em ambientes *dinâmicos*, juntamente com o robô, outras entidades mudam ou se movem com o tempo. Assim, estes ambientes se tornam mais difíceis que os *estáticos* para que o robô se localize. Se a mudança ocorre apenas durante uma leitura do sensor, isso não é tão catastrófico quanto mudanças permanentes. Mudanças permanentes geram discrepâncias com o mapa e dificultam o processo de localização.

Outro aspecto importante é a maneira que o robô se desloca pelo ambiente, de forma *passiva* ou *ativa*, enquanto realiza o processo de localização. Na abordagem *passiva*, o movimento do robô não é planejado para auxiliar a localização. Não há um planejamento de caminho que o auxilie a se localizar rapidamente. Na abordagem *ativa* é o oposto, já que ele é guiado para regiões singulares do ambiente que ajudem a reduzir o erro de estimativa da sua postura. Apesar da melhora no processo de localização, abordagens *ativas* necessitam de um módulo de controle de movimento que não apenas guie o robô pelo ambiente, mas que o guie em direção a regiões singulares. Esta busca por regiões singulares acaba sendo um custo extra nestas abordagens.

O último aspecto que será discutido é a respeito do número de robôs que estão realizando o processo de localização simultaneamente. Na abordagem *individual* apenas um robô realiza o processo de localização, e apenas as informações de seus sensores são utilizadas para a localização. Na abordagem *múltipla*, vários robôs estão se localizando e eles podem ajudar uns aos outros neste processo. Os primeiros que se localizam podem auxiliar os demais e diminuir o tempo total necessário para a localização. Contudo, existe a necessidade de compartilhar informações entre os robôs e isso pode causar erros por falhas na comunicação.

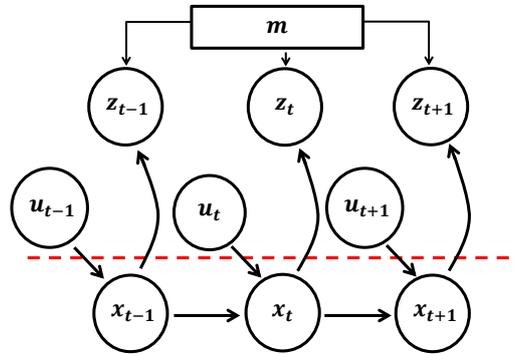
A localização de robôs móveis engloba outros fatores, como ambientes simétricos e assimétricos, e ruídos nas leituras dos sensores. Contudo, os fatores discutidos

aqui são os mais importantes e é o suficiente para explicar quão complexo este problema pode ser. Esta dissertação aborda o problema de localização global, passivo, realizado em ambientes externos, dinâmicos e com um único VANT.

Além da estimação da postura do robô, a localização de robôs móveis também pode ser vista como a tentativa de estabelecer uma correspondência entre o sistema de coordenadas global do mapa (que é independente da postura do robô) e o sistema de coordenada local do robô. Pela transformação do sistema de coordenada do robô para o sistema de coordenada do mapa, ele pode expressar a localização de pontos de interesse em seu sistema de coordenada, como obstáculos, por exemplo.

A Figura 2.2 apresenta o modelo gráfico para o problema de localização de robôs móveis, os ciclos intercalam movimento e percepção do robô. As quatro variáveis ilustradas na figura são muito importantes neste contexto, e elas já foram introduzidas na Seção 2.1. Os elementos acima da linha tracejada, isto é,  $\mathbf{m}$ ,  $\mathbf{z}_{1:t}$ , e  $\mathbf{u}_{1:t}$ , são conhecidos pelo robô, e com estas informações, o problema está em estimar o conjunto de posturas,  $\mathbf{x}_{0:t}$ <sup>3</sup>.

Figura 2.2: Modelo de localização de robôs móveis. Adaptado de (THRUN; BURGARD; FOX, 2005)



Considerando todo o histórico de controles de movimento e o das leituras dos sensores, a distribuição de probabilidade *a posteriori* da postura do robô corresponde a crença sobre a sua postura no instante  $t$ . Ela é dada pela equação

$$bel(\mathbf{x}_t) = p(\mathbf{x}_t \mid \mathbf{u}_{1:t}, \mathbf{z}_{1:t}, \mathbf{m}). \quad (2.3)$$

A maneira mais geral para calcular esta crença, *bel*, é usando o filtro de Bayes, que considera a leitura dos sensores, controles de movimento do robô e o mapa. Ainda, por ser recursivo, o cálculo da crença no tempo  $t$ ,  $bel(\mathbf{x}_t)$ , é feito

<sup>3</sup>No caso de localização local, a postura  $\mathbf{x}_0$  é conhecida.

considerando a crença anterior,  $bel(\mathbf{x}_{t-1})$  (THRUN; BURGARD; FOX, 2005),

$$\eta \cdot p(\mathbf{z}_t | \mathbf{x}_t, \mathbf{m}) \cdot \int_{\mathbf{x}_{t-1}} p(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{u}_t, \mathbf{m}) \cdot bel(\mathbf{x}_{t-1}) d\mathbf{x}_{t-1} \quad (2.4)$$

A Equação 2.4 é uma atualização recursiva que permite estimar a postura de um robô baseado na postura anterior, no controle de movimento e na leitura do sensor atuais. Em outras palavras, mostra que se existe uma distribuição de probabilidade sobre a postura anterior do robô,  $bel(\mathbf{x}_{t-1})$ , é possível computar sua postura atual no instante  $t$  considerando um controle de movimento executado,  $\mathbf{u}_t$ , e a leitura do sensor,  $\mathbf{z}_t$ , e isso é o Filtro de Bayes. Note que  $p(\mathbf{z}_t | \mathbf{x}_t, \mathbf{m})$  é o modelo de observação,  $p(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{u}_t, \mathbf{m})$  é o modelo de movimento, e  $\eta$  é um fator de normalização (THRUN; BURGARD; FOX, 2005).

Diferentes métodos de localização implementam a filtragem bayesiana. Os mais conhecidos são os filtros de Kalman (LEONARD; DURRANT-WHYTE, 1991) e o filtro de partículas (DELLAERT et al., 1999). O filtro de Kalman é, provavelmente, a melhor técnica estudada para implementar o filtro de Bayes, criado como uma técnica para predição e filtragem em sistemas Gaussianos lineares. Por outro lado, o filtro de partículas é um filtro não paramétrico, que não se baseia em uma forma funcional fixa de distribuição *a posteriori*, tal como o filtro de Kalman com a distribuição Gaussiana. Neste caso, o filtro de partículas aproxima a distribuição *a posteriori* por um número finito de partículas pesadas, em que cada partícula corresponde a uma postura no ambiente (THRUN; BURGARD; FOX, 2005).

### 2.3 Filtro de Partículas para Localização de Robôs Móveis

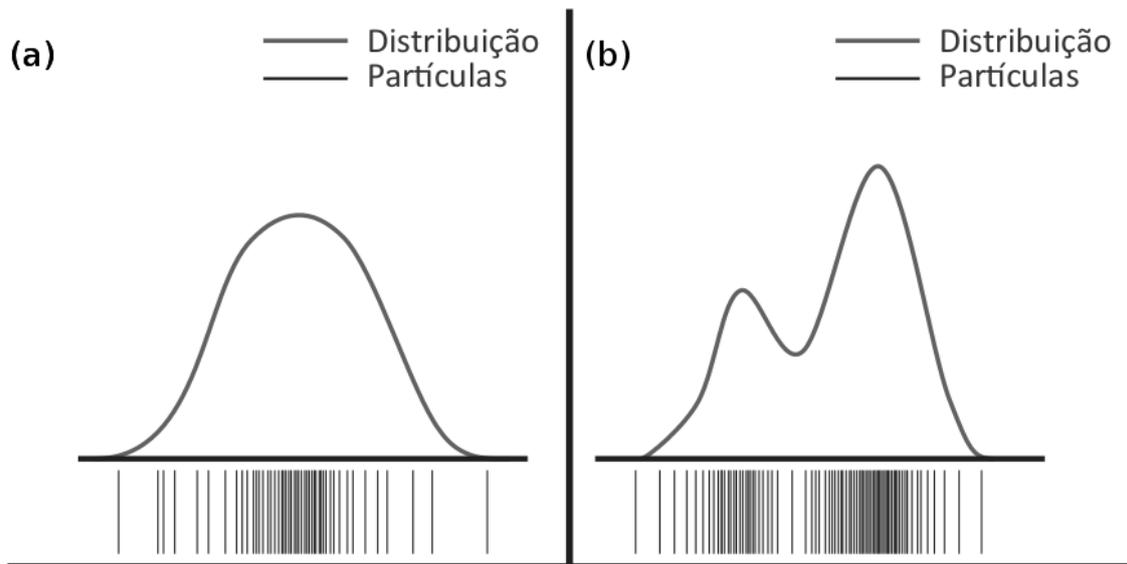
O filtro de partículas, ou MCL, se tornou um dos algoritmos de localização mais populares na robótica. Possui uma implementação não paramétrica do filtro de Bayes e é usado para estimar a postura de um robô em problemas de localização local ou global. Aqui, a ideia principal é representar a crença,  $bel(\mathbf{x}_t)$ , por um conjunto de partículas (THRUN; BURGARD; FOX, 2005). Este conjunto,  $X_t$ , de  $M$  partículas, é dado por

$$X_t = \{\mathbf{p}_t^{[i]} | i = 1, \dots, M\}, \quad (2.5)$$

em que  $\mathbf{p}_t^{[i]} = \langle \mathbf{x}, w \rangle$  corresponde a uma partícula composta por uma postura,  $\mathbf{x}$ , e um peso de importância,  $w$ . Este peso é um valor maior que zero, e a soma do peso de todas as partículas é igual a 1.

Através das partículas, o MCL pode distribuí-las de acordo com uma distribuição desejada e aproximá-las a uma outra distribuição arbitrária. A Figura 2.3 ilustra esta aproximação em dois exemplos de distribuições arbitrárias e seus correspondentes conjuntos de partículas. Em geral, quanto mais denso é o conjunto utilizado, melhor a aproximação. Porém, aumentar o número de partículas não gera apenas vantagens, mas desvantagens também, já que aumentar o número de partículas também aumenta o custo computacional. Quando o MCL é comparado do filtro de Kalman, a característica do MCL de modelar distribuições multi-modais é uma vantagem, já que o Kalman é restrito a distribuições Gaussianas (STACHNISS, 2006).

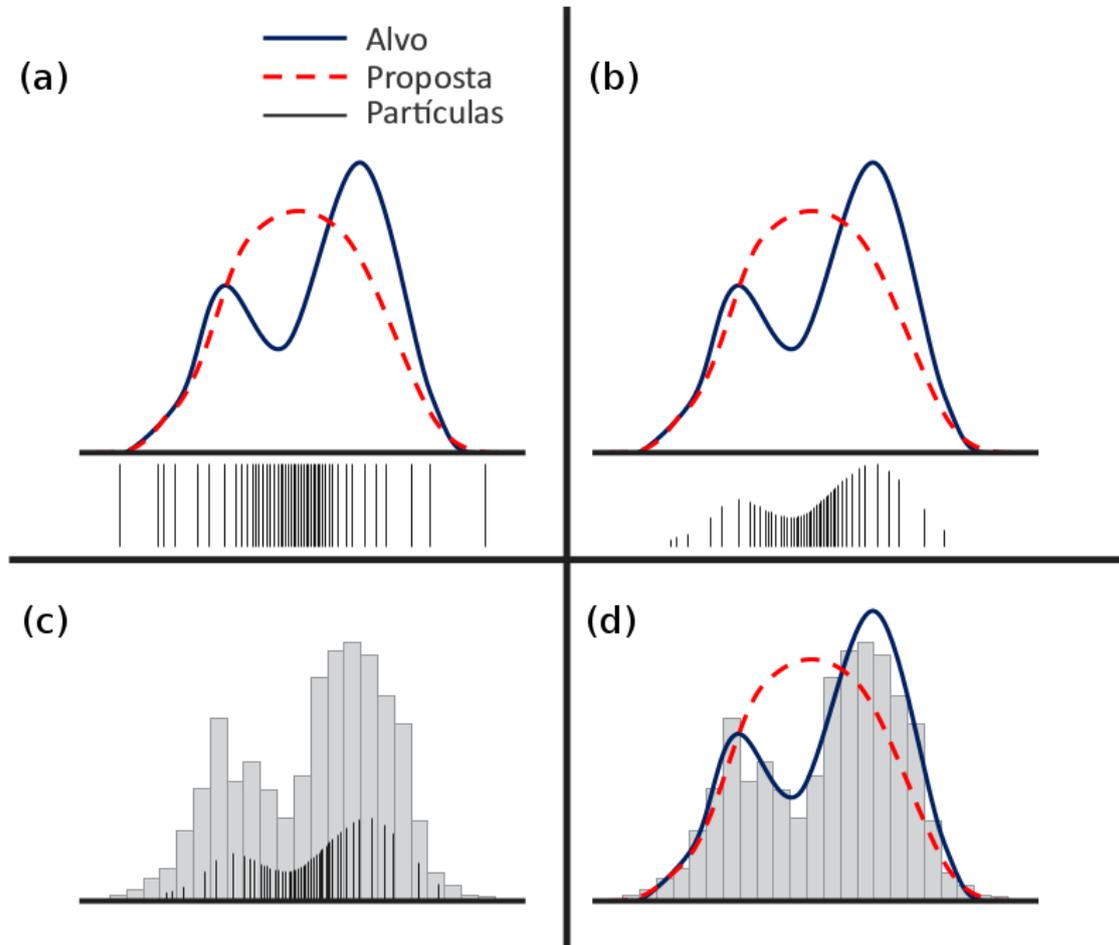
Figura 2.3: Exemplos de aproximação de distribuições arbitrárias através de um conjunto de partículas. As barras verticais pretas abaixo dos gráficos representam as partículas e as linhas cinzas são as distribuições. Em (a) as linhas verticais que representam as partículas estão concentradas em apenas um lugar, enquanto que em (b) há duas concentrações diferentes na mesma distribuição.



O MCL estima o conjunto de partículas  $X_t$  recursivamente, baseado no conjunto anterior,  $X_{t-1}$ , através do processo de Amostragem-Importância-Reamostragem (*Sampling-Importance-Resampling*, SIR), que é ilustrado na Figura 2.4.

Na Figura 2.4, existe a distribuição alvo (linha sólida azul), e a distribuição proposta (linha pontilhada vermelha), e através das partículas a distribuição proposta deverá se aproximar da distribuição alvo. Cada linha vertical abaixo das

Figura 2.4: Exemplo do processo SIR no filtro de partículas. (a) As partículas são amostradas de acordo com a distribuição proposta e deverão se aproximar da distribuição alvo (*Sampling*). (b) Elas são pesadas (*Importance*). (c) Passo de reamostragem (*Resampling*). (d) Distribuição das partículas aproximada a distribuição alvo.



distribuições representam uma partícula. Os passos do processo SIR podem ser vistos como:

- **Amostragem:** A próxima geração de partículas,  $X'_t$ , é criada baseada no conjunto anterior,  $X_{t-1}$ , e as partículas são propagadas de acordo com a distribuição proposta. Este passo é ilustrado pela Figura 2.4-(a).
- **Importância:** Cada partícula do conjunto  $X'_t$  é pesada através do modelo de observação e um valor de importância é atribuída a cada uma. É neste modelo que a leitura do sensor do robô e as leituras simuladas pelas partículas são comparadas. Quanto mais similar as leituras, maior o peso. Este passo é ilustrado pela Figura 2.4-(b).
- **Reamostragem:**  $M$  partículas são selecionadas a partir de  $X'_t$ , considerando

uma seleção aleatória com substituição. A probabilidade de selecionar uma partícula é proporcional ao seu peso. Então, as partículas com peso alto são mais prováveis de estarem no novo conjunto  $X_t$  do que aquelas com peso baixo. Este passo é ilustrado pela Figura 2.4-(c), em que as barras cinzas representam a soma dos pesos das partículas naquela região.

O Algoritmo 1 apresenta o funcionamento do MCL. Ele atua da mesma maneira que a Figura 2.4. Nas linhas 3 e 4, está representado o passo de Amostragem, em que são aplicados os controles de movimento,  $\mathbf{u}_t$ , no modelo de movimento,  $p(\mathbf{x}_t | \mathbf{u}_t, \mathbf{x}_{t-1}, \mathbf{m})$ . Na linha 5 está representado o passo de Importância e o modelo de observação, i.e, a leitura  $\mathbf{z}_t$  é aplicada em  $p(\mathbf{z}_t | \mathbf{x}_t, \mathbf{m})$ . Finalmente, as linhas 9 e 10 correspondem ao passo de Reamostragem, um novo conjunto de partículas  $X_t$  é gerado, considerando o peso das partículas.

---

**Algoritmo 1: Algoritmo de Localização de Monte Carlo**

---

**Entrada:**  $X_{t-1}, \mathbf{u}_t, \mathbf{z}_t, \mathbf{m}$   
**Saída:**  $X_t$

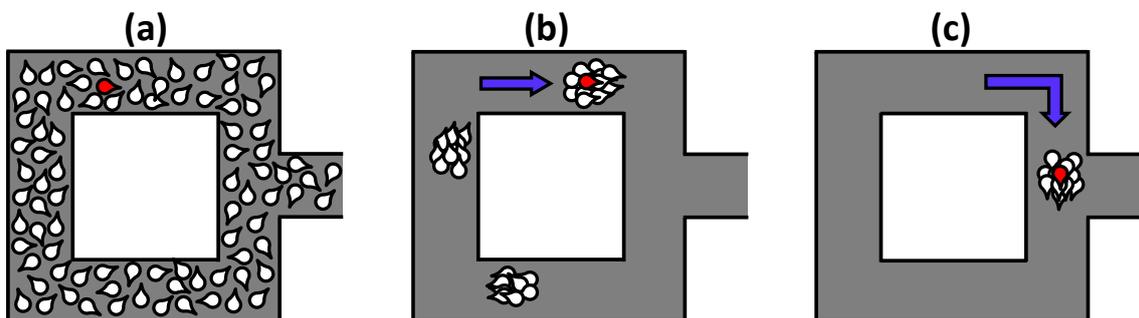
- 1  $X'_t = X_t = \emptyset$ ;
- 2 **para** cada  $i$  de 1 até  $M$  **hacer**
- 3  $\mathbf{x}_{t-1} = \mathbf{x}(\mathbf{p}_{t-1}^{[i]})$ ;
- 4 amostrar  $\mathbf{x} \propto p(\mathbf{x}_t | \mathbf{u}_t, \mathbf{x}_{t-1})$ ;
- 5  $w = p(\mathbf{z}_t | \mathbf{x}, \mathbf{m})$ ;
- 6  $\mathbf{p}_t^{[i]} = \langle \mathbf{x}, w \rangle$ ;
- 7  $X'_t = X'_t \cup \{\mathbf{p}_t^{[i]}\}$
- fin**
- 8 **para** cada  $i$  de 1 até  $M$  **hacer**
- 9 sortear  $\mathbf{p}_t^{[i]}$  de  $X'_t$  com probabilidade  $\propto w(\mathbf{p}_t^{[i]})$ ;
- 10  $X_t = X_t \cup \{\langle \mathbf{x}_t^{[i]}, \frac{1}{M} \rangle\}$ ;
- fin**
- 11 **retorna**  $X_t$ ;

---

O Algoritmo 1 descreve o processo de localização para um único passo. Enquanto o robô está se movendo, é necessário executar o algoritmo para continuar estimando a sua postura atual, que é definida como a média ponderada pelo peso das partículas de todas as posturas delas no mapa. A Figura 2.5 ilustra a execução do MCL para o problema de localização global de um robô e a evolução da crença sobre a sua localização. Na Figura 2.5-(a),  $M$  partículas com pesos iguais são distribuídas uniformemente pelo espaço livre do mapa. A Figura 2.5-(b) apresenta o filtro após algumas iterações iniciais, com concentrações de partículas em possíveis

localizações do robô. Finalmente, na Figura 2.5-(c), conforme o robô se descola pelo ambiente e realiza novas observações, diminui-se a variedade de posturas possíveis sobre onde ele estaria e aumenta-se a certeza sobre sua postura real.

Figura 2.5: Exemplo do MCL executando no contexto de localização global. Partículas estão representadas pelos pontos brancos e o robô representado pelo ponto vermelho. Em (a) as partículas são espalhadas pelas regiões livres do mapa. Já em (b), após algumas iterações e movimentos do robô, o filtro ainda não convergiu devido a existência de rigões ambíguos (corredores com curvas para direita). Por fim, em (c) o filtro já convergiu para a postura do robô.



## 2.4 Binary Robust Independent Elementary Features - BRIEF

Muitas aplicações em Visão Computacional, principalmente aquelas que estão focadas em descritores de *features*<sup>4</sup>, como localização utilizando câmeras, reconstrução de ambientes em 3D e reconhecimento de objetos, estão tendo que lidar com uma grande quantidade de dados a serem executados em dispositivos com recursos computacionais limitados, como *smartphones* e robôs construídos com computadores de baixo custo, como RaspBerry Pi<sup>5</sup>, Orange Pi PC<sup>6</sup> ou BeagleBone<sup>7</sup>. Esta maior quantidade de dados a serem processados e o reduzido poder de processamento requerem formas de representar imagens que sejam rápidas para serem computadas, rápidas para encontrarem correspondências entre elas quando comparadas e que consumam poucos recursos computacionais.

Estas representações das imagens são conhecidas como descritores, que expressam as características elementares das imagens, como formas, cores, texturas

<sup>4</sup>*Feature* é uma parte da imagem que contém informação relevante para ser utilizada em tarefas computacionais. Na imagem, uma *feature* pode ser uma borda, um canto, linhas e até mesmo objetos.

<sup>5</sup><https://www.raspberrypi.org/learning/hardware-guide/components/raspberry-pi/>

<sup>6</sup><http://www.orangepi.org/orangepipc2/>

<sup>7</sup><http://beagleboard.org/black>

e movimentos, de uma maneira mais compacta, estável e invariante, sendo normalmente utilizados para comparar ou descrever imagens (SZELISKI, 2010). Por exemplo, ao invés de comparar diretamente duas imagens, faz-se a extração dos descritores de cada imagem e então realiza-se uma comparação mais rápida e menos custosa entre eles para saber se as duas imagens são similares. Existem diferentes descritores de imagens disponíveis na literatura, como o SIFT (LOWE, 2004), SURF (BAY; TUYTELAARS; GOOL, 2006) ou ORB (RUBLEE et al., 2011).

Alguns descritores representam várias características de uma imagem, mas é possível que apenas uma parte delas já seja suficiente para a aplicação. Para isso existem os descritores pequenos, que podem ser obtidos por duas maneiras: pela redução de descritores originais, como os já citados SIFT ou SURF, através de processos de redução de dimensionalidade, como PCA (MIKOLAJCZYK; SCHMID, 2005) e LDA (HUA; BROWN; WINDER, 2007), ou pela construção direta de um descritor pequeno. Sobre esta última opção, ao invés de computar um descritor todo para depois reduzi-lo, Calonder et al. (2010) propuseram um descritor curto chamado de *Binary Robust Independent Elementary Features*, BRIEF. Ele é representado por um vetor binário que descreve um *patch*  $\mathbf{p}$ , de tamanho  $S \times S$ , tendo os valores vetor definidos por meio de um teste  $\tau$ :

$$\tau(\mathbf{p}; \mathbf{x}, \mathbf{y}) := \begin{cases} 1 & \text{se } \mathbf{p}(\mathbf{x}) < \mathbf{p}(\mathbf{y}) \\ 0 & \text{caso contrário} \end{cases}, \quad (2.6)$$

em que  $\mathbf{p}(\mathbf{x})$  é o valor de intensidade já suavizado de  $\mathbf{p}$  no pixel  $\mathbf{x} = (u, v)^\top$ . Quando um conjunto de  $n(\mathbf{x}, \mathbf{y})$  pares são escolhidos através de uma distribuição, eles definem um conjunto de testes binários. Então, o descritor BRIEF é o vetor binário de tamanho  $n$ ,

$$f_n(\mathbf{p}) := \sum_{1 \leq i \leq n} 2^{i-1} \tau(\mathbf{p}; \mathbf{x}_i, \mathbf{y}_i). \quad (2.7)$$

A Equação 2.6 retorna um valor binário apenas levando em conta a informação de intensidade de um único pixel, e isso significa que o resultado pode ser muito sensível a ruídos na imagem. Uma abordagem para evitar que os ruídos influenciem no descritor é diminuir as pequenas variações dos valores de intensidade dos pixels através de um processo de suavização no *patch* da imagem em que se está computando o descritor BRIEF. Neste caso, um *kernel* Gaussiano é usado, e de

acordo com os testes apresentados pelos autores, o melhor desempenho alcançado na correspondência dos descritores foi quando usaram o *kernel* Gaussiano com valor 2 de desvio padrão, e uma janela de tamanho  $9 \times 9$  pixels. Este *kernel*, para cada pixel da imagem, calcula uma média ponderada por uma Gaussiana considerando seus vizinhos, que neste caso são 80. O resultado deste cálculo se torna o valor do pixel testado, que está no centro da janela.

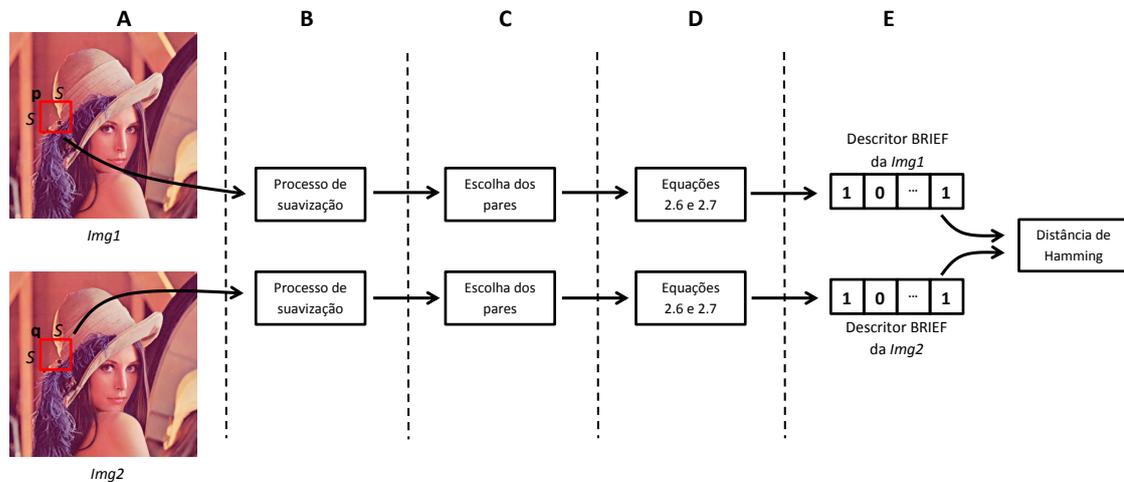
A Equação 2.6 compara dois valores de intensidade de um par de pixels. Estes pares são escolhidos de um mesmo *patch* seguindo uma abordagem proposta pelos autores:

- $(\mathbf{X}, \mathbf{Y}) \sim \text{i.i.d. Gaussian} \left(0, \frac{1}{25}S^2\right)$ : Os pixels são amostrados a partir de uma distribuição Gaussiana isotrópica. Com mais detalhes, os parâmetros da Gaussiana que atingiu os melhores resultados é  $\frac{S}{2} = \frac{5}{2}\sigma \Leftrightarrow \sigma^2 = \frac{1}{25}S^2$  (CALONDER et al., 2010).

Depois de gerados os vetores binários, é necessário compará-los. No caso do BRIEF, a Distância de Hamming (HAMMING, 1950) é usada para medir quão parecidos eles são. Considerando dois vetores de mesmo tamanho, a Distância de Hamming entre estes dois é o número de elementos que, na mesma posição de cada vetor, tem valores diferentes.

A Figura 2.6 apresenta o processo de computação do BRIEF e como os descritores são construídos. No passo **A**, suponha duas imagens de mesmo tamanho, *Img1* e *Img2*. Ainda, considere também um *patch* em cada imagem,  $\mathbf{p}$  para *Img1* e  $\mathbf{q}$  para *Img2*, com tamanhos  $S \times S$ . Depois, ambos os *patches* são submetidos a um processo de suavização pelo *kernel* Gaussiano, representado pelo passo **B**. No passo **C** os pares de pixels de cada *patch* são selecionados e usados como argumentos para as Equações 2.6 e 2.7 no passo **D**. Finalmente, no passo **E** os descritores são construídos, resultado do passo anterior, e comparados usando distância de Hamming. Cada posição do vetor de um descritor é comparada com a respectiva posição do vetor de um outro descritor. Quanto maior o número de igualdade nestas comparações, mais similares são os *patches* das imagens que estes descritores representam.

Figura 2.6: Processo de construção do descritor BRIEF. Todos estes passos são repetidos para cada *patch* das imagens *Img1* e *Img2*.



## 2.5 Trabalhos relacionados

Câmeras estão sendo usadas como sensores para muitos tipos de aplicações, assim como os VANTs. A combinação de câmeras e VANTs permite o tratamento de diferentes problemas. Como exemplo, o sistema que Sim et al. (2002) propuseram estima alguns parâmetros de navegação, tais como posição e velocidade do VANT, através de uma sequência de imagens. Ainda, outro exemplo é o trabalho de Conte e Doherty (2008), no qual apresenta um *framework* que utiliza imagens de satélite georeferenciadas para melhorar o sistema de navegação de um VANT em caso de falhas do GPS. Neste *framework*, eles combinam sensores inerciais, algoritmo de odometria visual e técnicas de registro de imagens<sup>8</sup> entre o vídeo da câmera e imagens aéreas georeferenciadas.

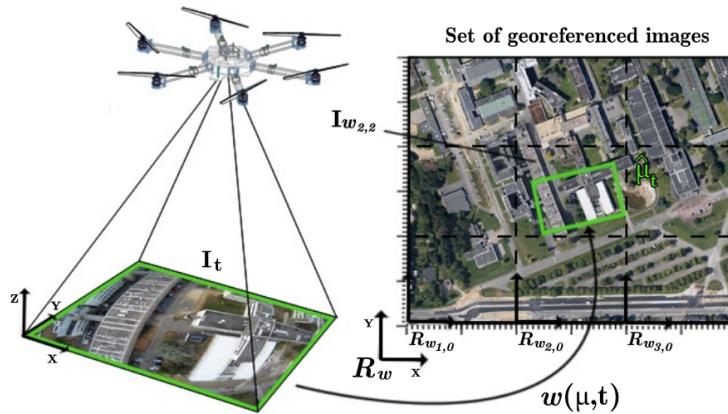
O foco desta dissertação é propor um modelo de observação, baseado em visão computacional, para lidar com o problema de localização global de VANTs, utilizando imagens de satélite como mapa. Muitos trabalhos que tratam este problema estão centrados em localizar, em imagens de satélite conhecidas *a priori*, imagens tiradas por câmeras que estão embarcadas em VANTs. Abaixo serão discutidos alguns dos mais relevantes trabalhos para esta dissertação.

Alguns trabalhos focam no problema de localização local de VANTs. A abordagem proposta por Yol et al. (2014) localiza um VANT, com uma câmera embar-

<sup>8</sup>Técnicas de registro de imagens, *Image Registration*, tentam alinhar geometricamente duas ou mais imagens da mesma cena, tiradas em diferentes momentos e pontos de vista (ZITOVÁ; FLUSSER, 2003).

cada, em um conjunto de imagens georeferenciadas que estão juntas e são usadas como mapa. Como mostra a Figura 2.7, o mapa é dividido em *frames*,  $\mathbf{R}_{W_{i,j}}$ , e para cada *frame* há uma imagem georeferenciada correspondente,  $\mathbf{I}_{W_{i,j}}$ . Como a localização é local, neste caso, a postura inicial do robô é conhecida, e para estimar a próxima postura é utilizada uma transformação 2D,  $\mu$ , entre a imagem georeferenciada,  $\mathbf{I}_{W_{i,j}}$ , e a imagem atual que está sendo tirada pela câmera do VANT,  $\mathbf{I}_t$ . Através desta transformação, é possível extrair a posição relativa entre o atual ponto de vista e a posição da câmera que está capturando a imagem de referência.

Figura 2.7: Representação da localização absoluta de um VANT proposto por Yol et al. (2014). Figura extraída de (YOL et al., 2014).



Há muitas transformações geométricas que podem ser aplicadas na imagem  $\mathbf{I}_t$ , mas o objetivo é encontrar a transformação que torna a imagem  $\mathbf{I}_{W_{i,j}}$  mais similar a  $\mathbf{I}_t$ , ou seja,

$$\hat{\mu} = \arg \max_{\mu} \text{MI}(\mathbf{I}_{W_{i,j}}(\mathbf{x}), \mathbf{I}_t(w(\mathbf{x}, \mu))), \quad (2.8)$$

em que MI, Informação Mútua (*Mutual Information*), é a função de similaridade, e  $w(\mathbf{x}, \mu)$  é a função de deformação que define a transformação geral do ponto  $\mathbf{x}$  da imagem  $\mathbf{I}_t$  usando o parâmetro  $\mu$ . Neste caso, a função de similaridade é responsável por informar quão parecidas são as duas imagens que estão sendo comparadas.

Neste contexto, MI (DAME; MARCHAND, 2012; VIOLA; III, 1997) compara duas imagens e é utilizada devido a sua robustez em relação a mudanças locais e globais entre imagens. MI expressa a medida da quantidade de informação compartilhada entre duas variáveis aleatórias, ou imagens para o caso do trabalho de Yol et al. (2014). Em outras palavras, de acordo com Zhou e Boulanger (2011), MI usa informação proveniente da entropia para medir a relação entre duas variáveis.

Ainda, quanto maior o valor de MI, mais informação sobre uma variável pode ser obtida conhecendo a outra, ou seja, quando as duas forem iguais, o MI atingirá seu valor máximo. A medida de MI entre duas imagens é dada por

$$\text{MI}(\mathbf{I}, \mathbf{I}^*) = h(\mathbf{I}) + h(\mathbf{I}^*) - h(\mathbf{I}, \mathbf{I}^*), \quad (2.9)$$

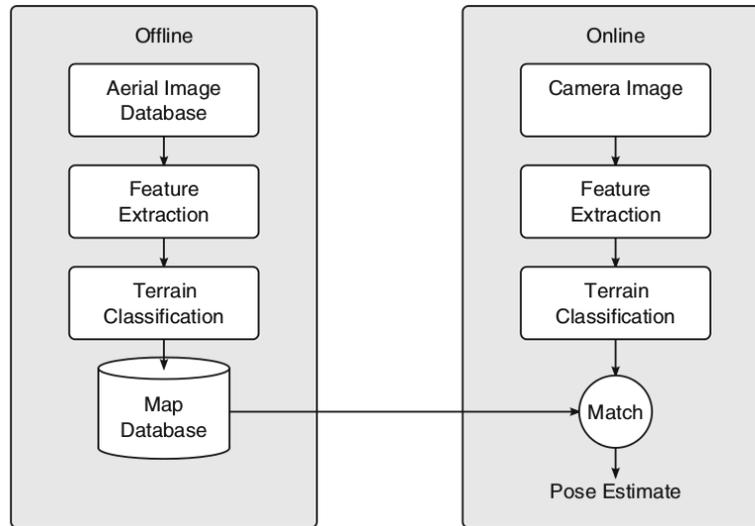
em que  $h(\mathbf{I})$  é a entropia da imagem  $\mathbf{I}$  e mede a variabilidade da imagem, enquanto que  $h(\mathbf{I}, \mathbf{I}^*)$  corresponde a entropia conjunta de duas imagens (YOL et al., 2014).

Nos experimentos do trabalho de Yol et al. (2014), o VANT voa com aproximadamente 150 metros de altura. Isso os permite assumir que efeitos de perspectiva são praticamente inexistentes e que a cena é planar, já que este efeito impacta no desempenho do seu sistema de localização. Esta fraqueza relacionada ao efeito de perspectiva restringe a utilização do sistema apenas para aplicações que não necessitam de vôos com baixa altitude. Além disso, o problema tratado por Yol et al. (2014) é o problema de localização local. Isso implica que ou é necessário conhecer a localização do VANT quando ele decola sem GPS, ou detectar uma falha no GPS quando este para de funcionar durante o vôo. Ambas abordagens não são muito adequadas para aplicações reais, pois seria necessário saber as coordenadas de partida do VANT, ou equipá-lo com um módulo extra para a verificação do funcionamento do GPS.

Além dos trabalhos de localização local de VANTs, também já foram propostos trabalhos que lidam com a localização global destes robôs. Masselli, Hanten e Zell (2016) usam o MCL para estimar a postura de um VANT com 4 DoF. O modelo de observação proposto por eles é baseado em classificação de terrenos, i.e., o objetivo é usar classificação de terrenos para realizar o alinhamento entre a imagem tirada pelo VANT e a imagem de satélite usada como mapa. Como apontado pelos autores, a vantagem de usar este tipo de abordagem está na robustez em relação às várias mudanças que podem acontecer no ambiente, como as condições das águas, tempo durante o dia e diferentes estações do ano.

A Figura 2.8 apresenta o esquema do sistema proposto. Há uma fase de treinamento *a priori*, em que imagens do ambiente no qual o VANT irá sobrevoar são usadas para treinar o classificador. A classificação é baseada em *features* extraídas destas imagens. Para extraí-las, cada imagem é dividida em uma grade, e para cada célula da grade um único descritor ORB (RUBLEE et al., 2011) é calculado, tendo o centro da célula como ponto chave da *feature*. Então, o classificador *Random*

Figura 2.8: Representação da localização absoluta de um VANT proposto Merino et al. (2006). Figura extraída de (MASSELLI; HANTEN; ZELL, 2016).



*Forest* atribui, para cada célula da grade, um vetor de probabilidade que contém as probabilidades de cada classe de terreno. No caso deste trabalho, há quatro classes diferentes, grama, arbustos/árvores, ruas/asfalto e prédios. O sistema proposto foi submetido a dados de vôos reais e simulados, e os resultados mostraram que é robusto às mudanças típicas de iluminação e mudanças no próprio ambiente.

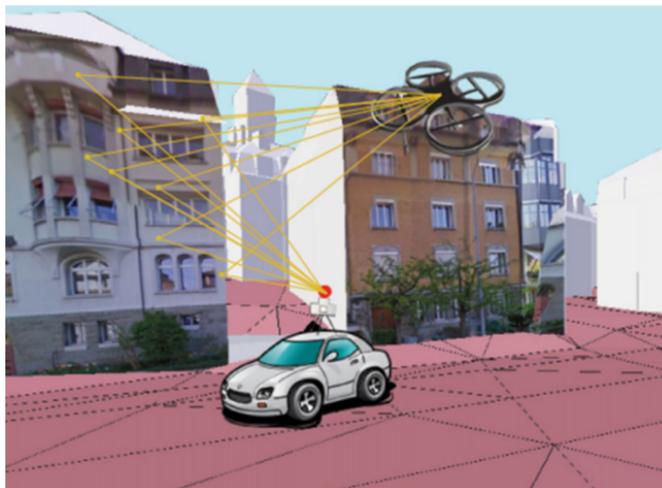
Apesar de tratarem o problema de localização global, o sistema proposto por Masselli, Hanten e Zell (2016) é totalmente baseado em aprendizagem de máquina com aprendizado supervisionado. Isso significa que antes de executar o sistema para estimar a localização do VANT, é necessário submetê-lo a uma fase de treinamento. Em outras palavras, é preciso classificar, manualmente, um conjunto de imagens do ambiente de acordo com as classes declaradas e então usar este conjunto para treinar o algoritmo de aprendizagem de máquina. Só depois disso, então, é que o algoritmo estará pronto para ser utilizado e realizar a classificação das imagens que serão capturadas.

Apesar do esquema de localização de VANTs mais comum ser o ilustrado pela Figura 1.1 apresentado na Seção 1.1, há outras formas de realizar este tipo de localização, como é o caso do trabalho desenvolvido por Majdik et al. (2015), que busca localizar globalmente o robô enquanto ele voa em baixas altitudes (10~20 metros do chão) por ambientes urbanos. Atualmente existem mapas de cidades criados usando imagens em 360° capturadas por carros enquanto estes trafegam pelas ruas. Como exemplo, podemos citar o *Google Street View*<sup>9</sup>. Então, ao invés

<sup>9</sup><https://www.google.com/streetview/>

de usar imagens de satélite como mapa, Majdik et al. (2015) utilizam este tipo de imagem e tentam localizar o VANT enquanto ele se desloca pelas ruas, como ilustrado pela Figura 2.9.

Figura 2.9: Esquema de localização global de um VANT proposto por Majdik et al. (2015). As imagens capturadas por um carro são utilizadas como mapa e são comparadas com as imagens capturadas pelo robô. Figura extraída de (MAJDIK et al., 2015).



Para saber a localização do VANT, a abordagem proposta por Majdik et al. (2015) procura no mapa a imagem que mais se assemelha com a imagem capturada pelo robô. Como o mapa é representado por um conjunto de imagens e não apenas uma só, é necessário comparar esta imagem capturada com todas as imagens do mapa. Apesar de haver uma etapa de conversão das imagens em *features*, ainda é um número grande de comparações pelo número de imagens que compõem o mapa. Quando a imagem do mapa mais semelhante à imagem capturada pelo VANT é encontrada, sabe-se, então, a sua localização. Apesar da novidade em localizar o robô em baixas altitudes, esta abordagem pode trazer um certo risco para as pessoas que trafegam ou caminham pelas ruas e calçadas por conta da pequena distância entre eles. Pelo mapa utilizado ser fornecido pelo Google Street View, o sistema fica limitado às regiões nas quais ele já mapeou<sup>10</sup>, e dependendo do país estas regiões são a minoria. Ainda, outra limitação é que este tipo de sistema só funciona em ambientes urbanos, já que o Google Street View não tem a mesma cobertura que as imagens de satélite.

---

<sup>10</sup><https://www.google.com/streetview/understand/>

### 3 NOVO MODELO DE OBSERVAÇÃO BASEADO NO DESCRITOR ABBRIEF

Neste capítulo propõe-se um modelo de observação baseado em uma variação do descritor BRIEF, o abBRIEF. Este modelo é incorporado ao MCL e aplicado ao problema de localização global de um VANT. Algumas características deste problema devem ser levadas em conta para a decisão do modelo de observação: considerando que a estimativa da postura do VANT, neste trabalho, é em 4 DoF, um grande número de partículas é necessário e por isso o modelo de observação não pode ser custoso computacionalmente; também é importante que o modelo de observação proporcione uma forma rápida de encontrar semelhanças na comparação entre as imagens que apresentam diferenças entre si. Dentre os vários descritores disponíveis na literatura, optou-se por utilizar o descritor BRIEF como base, devido a sua adequação aos fatores acima mencionados, sendo rápido para ser computado, pequeno para ser armazenado e razoavelmente resistente às imagens com grandes diferenças, determinando assim a similaridade entre a imagem da partícula e do VANT com uma certa robustês.

#### 3.1 Descritor abBRIEF

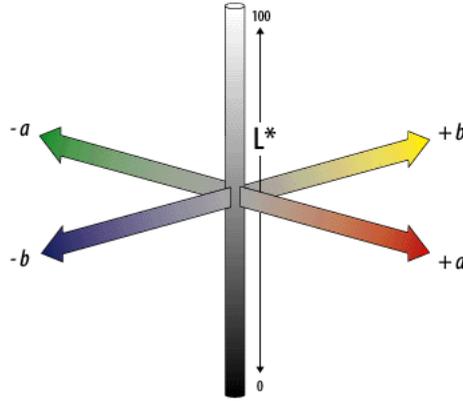
O descritor proposto neste trabalho, abBRIEF, o qual será usado no modelo de observação, possui algumas mudanças em relação ao descritor BRIEF original, sendo necessárias para obter um melhor desempenho para o problema em questão deste trabalho. Com exceção dos itens citados, o descritor proposto, abBRIEF, segue a descrição do descritor BRIEF apresentada na Seção 2.4. As mudanças feitas foram:

- O espaço de cor utilizado é o CIE  $L^*a^*b^*$  e para cada par de pixels escolhido são feitas apenas comparações para os canais  $a^*b^*$  do CIE  $L^*a^*b^*$ .
- Para lidar com o ruído nos valores dos pixels e deixar mais uniforme os valores das cores nas imagens, foi aplicado um processo de quantização com 25 níveis nas imagens capturadas pelo VANT e no próprio mapa.

Como apontado por Merino et al. (2006), *features* dependem de certas características na cena para serem detectadas, como bordas, cantos ou até mesmo

linhas, dependendo do descritor que está se utilizando. Por isso, para o problema de localização de VANTs, recorrer a técnicas que as utilizam é uma limitação, já que o robô só poderia sobrevoar regiões que tivessem estes tipos de características. As imagens de satélite que são utilizadas como mapa também não apresentam uma resolução boa o suficiente para que o processo de extração de *features* funcione de forma satisfatória. Além destas, outra limitação é o custo extra necessário para extraí-las das imagens. O uso do descritor abBRIEF neste trabalho funciona sem o uso de *features*. Aqui, descreve-se toda a imagem através de um único descritor, sem extrair *patches* a serem descritos. Desta forma, o método funciona muito mais rápido que utilizando *features*, e não depende da presença de cantos ou bordas bem definidos. Por fim, como os *patches* não são mais utilizados e o descritor considera toda a imagem, aplica-se a ela um processo de quantização para lidar com os ruídos e grandes variações nas cores. Este passo substitui o passo de suavização por *kernel* Gaussiano utilizado por Calonder et al. (2010).

A outra mudança proposta está relacionada às informações da imagem utilizada, especificamente os valores de cada pixel. Quando se utiliza cores é possível obter informações importantes e que auxiliem no processo de comparação de imagens (MEHTRE et al., 1995). O espaço de cor CIE  $L^*a^*b^*$ , ou CIELAB, é um bom meio de representação das cores e é independente de dispositivo. Esta independência significa que as cores são definidas sem levar em conta sua natureza de criação ou o dispositivo em que serão impressas. Sua acurácia e portabilidade é muito útil para impressões, fábricas automotivas, indústrias têxtil e plásticas. Como ilustrado pela Figura 3.1, cada cor é composta por três canais que se cruzam no centro,  $L^*$ ,  $a^*$  e  $b^*$ .  $L^*$  representa a luminosidade, variando de preto, 0, até branco, 100.  $a^*$  e  $b^*$  correspondem aos componentes de cores, com  $a^*$  variando de vermelho, 127, até verde, -128, e  $b^*$  variando de amarelo, 127, até azul, -128. Neste espaço de cor existe a teoria das cores opostas, em que duas cores não podem ser vermelhas e verdes ou azul e amarelo ao mesmo tempo. Essa maneira de representação das cores é muito importante para as comparações apresentadas na Equação 2.6. O canal  $L^*$ , como já dito, representa a luminosidade. Em ambientes externos, como é o caso da aplicação deste trabalho, o sol pode variar muito a luminosidade, além das variações das posições das sombras de prédios e árvores. Por isso, este canal não está sendo considerado, apenas os valores correspondentes as cores,  $a^*$  e  $b^*$ .

Figura 3.1: Sistema do espaço de cor CIE  $L^*a^*b^*$ .

### 3.2 Modelo de observação baseado no descritor abBRIEF

O modelo de observação de um robô é definido como a distribuição de probabilidade condicional,  $p(\mathbf{z}_t|\mathbf{x}_t, \mathbf{m})$ , da leitura do sensor  $\mathbf{z}_t$ , dado a postura  $\mathbf{x}_t$  e o mapa do ambiente,  $\mathbf{m}$ . A abordagem proposta neste trabalho considera um VANT voando sempre paralelo ao chão<sup>1</sup>, e sua postura é definida como  $\mathbf{x}_t = (x_t, y_t, z_t, \theta_t)^T$ . O VANT é equipado com uma câmera que captura uma imagem  $\mathbf{I}_t$  de tamanho  $w \times h$ , com  $c$  canais de cores.

Neste novo modelo, a leitura do sensor  $\mathbf{z}_t$ , associada à imagem  $\mathbf{I}_t$  é computada através de um único descritor abBRIEF sobre  $\mathbf{I}_t$ . Para isso, definimos um conjunto,  $\mathbf{K}$ , de  $k$  pares de pontos,

$$\mathbf{K} = \{(\mathbf{a}_1, \mathbf{b}_1), (\mathbf{a}_2, \mathbf{b}_2), \dots, (\mathbf{a}_k, \mathbf{b}_k)\}, \quad (3.1)$$

em que  $\mathbf{a}_i = (u, v)^T$  e  $\mathbf{b}_i = (u, v)^T$ ,  $i = 1, \dots, k$ , são pixels aleatoriamente selecionados a partir de  $\mathbf{I}_t$  e  $u$  e  $v$  são coordenadas do pixel da imagem.

No instante  $t$ ,  $\mathbf{z}_t$  é o conjunto de leituras associadas a todos os pares de pontos,

$$\mathbf{z}_t = \{\mathbf{z}_t^1, \mathbf{z}_t^2, \dots, \mathbf{z}_t^k\}, \quad (3.2)$$

em que  $\mathbf{z}_t^i = (\tau_t^{i,1}, \dots, \tau_t^{i,c})^T$ ,  $i = 1, \dots, k$ , é a leitura associada ao  $i$ -ésimo par de ponto,  $(\mathbf{a}_i, \mathbf{b}_i)$ . Cada componente  $\tau_t^{i,j}$  de  $\mathbf{z}_t^i$  é o resultado de um teste binário feito

<sup>1</sup>Não são consideradas variações nos ângulos *roll* e *pitch*, apenas no *yaw*.

sobre as intensidades dos pixels nas posições dadas por  $\mathbf{a}_i$  e  $\mathbf{b}_i$ ,

$$\tau_t^{i,j} = \begin{cases} 1 & , \text{ se } \delta(\mathbf{I}_t, \mathbf{a}_i, j) > \delta(\mathbf{I}_t, \mathbf{b}_i, j) \\ 0 & , \text{ caso contrário} \end{cases}, \quad (3.3)$$

sendo  $\delta(\mathbf{I}_t, \mathbf{a}_i, j)$  a função que retorna a intensidade do pixel na posição  $\mathbf{a}$ , no canal  $j$ , da imagem  $\mathbf{I}_t$ . Para imagens em escala de cinza, esta comparação é feita para o único canal disponível, mas para o nosso caso, considerando o espaço de cor CIE  $L^*a^*b^*$ , ela é aplicada para cada um dos dois canais,  $a^*b^*$ . Então, para cada par de pixels de uma imagem que é testada, serão acrescentados dois valores ao descritor, um do canal  $a^*$  e outro do  $b^*$ .

O mapa do ambiente,  $\mathbf{m}$ , corresponde a uma grande imagem de satélite,  $\mathbf{M}$ , do lugar onde o VANT irá sobrevoar. É possível recortar, de  $\mathbf{M}$ , uma imagem  $\mathbf{I}_t^*(\mathbf{x}_t, \mathbf{M})$  que deveria ser similar à imagem  $\mathbf{I}_t$  observada pelo VANT na postura  $\mathbf{x}_t$ . Esta imagem recortada será identificada como imagem estimada, já que representa, no mapa, a imagem capturada pelo robô considerando a sua postura estimada. Esta operação de recorte é muito custosa computacionalmente falando, e felizmente, na prática, ela não é necessária para determinar a leitura estimada  $\mathbf{z}_t^*$ . Apenas deve-se transformar o conjunto de pares de pontos,  $\mathbf{K}$ , em função de  $\mathbf{x}_t$  e  $\mathbf{M}$ , em um novo conjunto,  $\mathbf{K}^*$ :

$$\mathbf{K}^*(\mathbf{x}_t, \mathbf{M}) = \{(\mathbf{a}_1^*, \mathbf{b}_1^*), (\mathbf{a}_2^*, \mathbf{b}_2^*), \dots, (\mathbf{a}_k^*, \mathbf{b}_k^*)\}, \quad (3.4)$$

com cada ponto  $\mathbf{a}_i^*$  sendo definido como<sup>2</sup>:

$$\mathbf{a}_i^* = \mathbf{x}_t + \begin{bmatrix} \cos \theta_t & \sin \theta_t \\ -\sin \theta_t & \cos \theta_t \end{bmatrix} \cdot s \cdot \left( \mathbf{a}_i - \begin{pmatrix} w/2 \\ h/2 \end{pmatrix} \right), \quad (3.5)$$

em que  $\mathbf{x}_t = (u_t, v_t)^T$  é a posição do pixel correspondente da postura do VANT,  $\mathbf{x}_t$ , em  $\mathbf{M}$ ;  $\theta(\mathbf{x}_t)$  é o ângulo *yaw* do VANT;  $s$  é um fator de escala que relaciona a altura de vôo do VANT e faz parte da postura da partícula através da componente  $z(\mathbf{x}_t)$ , e a área de  $\mathbf{M}$  observada por ele; e  $(w/2, h/2)^T$  é a posição do VANT em  $\mathbf{I}_t$ , ou seja, o centro da imagem. Mais detalhadamente, o fator de escala de cada partícula é usado para afastar ou aproximar os pares de pontos do centro da imagem (posição  $x(\mathbf{x}_t)$  e  $y(\mathbf{x}_t)$  da partícula), simulando diferentes alturas das partículas, e é inicialmente atribuído a cada uma através de uma distribuição uniforme. Conforme o filtro vai

---

<sup>2</sup>A operação é análoga para determinar  $\mathbf{b}_i^*$ . Apenas substitua  $\mathbf{a}_i$  por  $\mathbf{b}_i$ .

convergindo, a componente  $z(\mathbf{x}_t)$  também vai convergindo e se aproximando do valor que corresponde a altura do VANT. Análogo à Equação 3.2 e Equação 3.3, a leitura estimada  $\mathbf{z}_t^* = \{\mathbf{z}_t^{*1}, \mathbf{z}_t^{*2}, \dots, \mathbf{z}_t^{*k}\}$  é o conjunto de leituras associadas a todos os pares de pontos em  $\mathbf{K}^*(\mathbf{x}_t, \mathbf{M})$ , em que  $\mathbf{z}_t^{*i} = (\tau_t^{*i,1}, \dots, \tau_t^{*i,c})^T$  é a leitura associada ao  $i$ -ésimo par de ponto,  $(\mathbf{a}_i^*, \mathbf{b}_i^*)$ . Define-se a similaridade entre a leitura observada e a leitura estimada como,

$$\text{sim}(\mathbf{z}_t, \mathbf{z}_t^*) = \frac{1}{kc} \sum_{i=1}^k \sum_{j=1}^c \tau_t^{i,j} \odot \tau_t^{*i,j}, \quad (3.6)$$

no qual  $f \odot g$  é o operador lógico XNOR<sup>3</sup>, e  $c$  é a dimensão do espaço de cor usado – neste caso,  $c = 2$ . Esta operação é o inverso da Distância de Hamming, já que a Equação 3.6 retornará a porcentagem de valores iguais nas mesmas posições de ambos descritores. Note que a imagem do VANT e a imagem de satélite devem estar no mesmo espaço de cor.

Idealmente,  $\mathbf{z}_t^*$  deveria ser igual a  $\mathbf{z}_t$ , i.e.,  $\text{sim}(\mathbf{z}_t, \mathbf{z}_t^*) = 1$ . Contudo, as leituras estão sujeitas a uma série de erros: variações no ambiente (por exemplo, objetos dinâmicos e novas construções), diferenças de iluminação, discretização da imagem e outros. Isso é ilustrado na Figura 3.2.

Apesar da imagem obtida pelo VANT, Figura 3.2-(a), e a imagem extraída do mapa global, Figura 3.2-(b), mostrarem o mesmo lugar, elas sofrem dos efeitos já mencionados. Ainda, as leituras observadas,  $\mathbf{z}_t$ , e estimadas,  $\mathbf{z}_t^*$ , consideram dois pares de pixels que são diferentes por apenas um dos quatro valores do vetor binário. Para cada par em ambas imagens estes valores foram calculados considerando os canais  $a^*$  e  $b^*$ , que posteriormente são adicionados ao vetor.

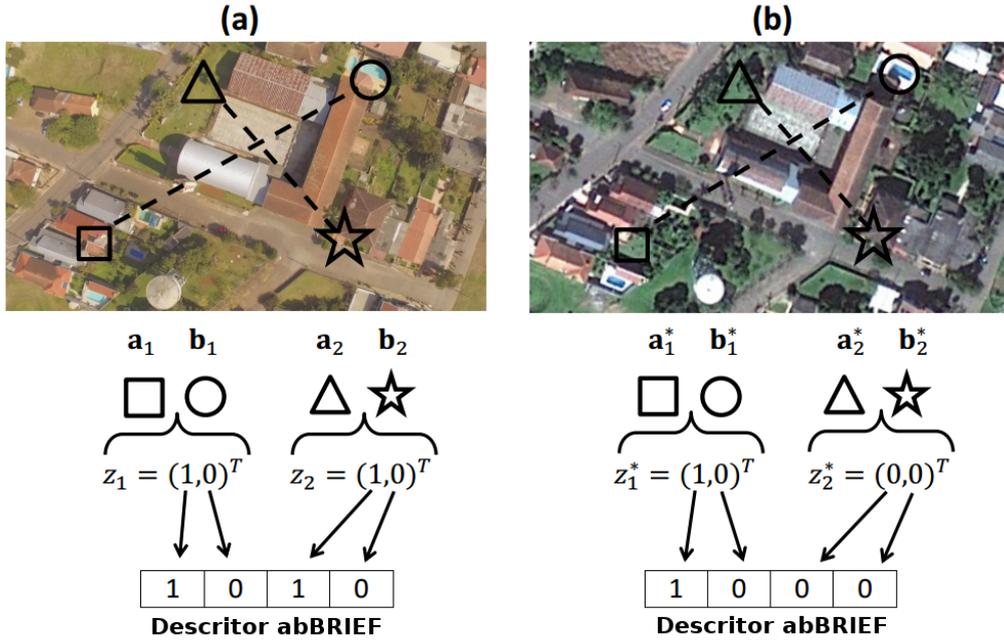
Esse erro gerado por mudanças em imagens que correspondem ao mesmo local foi modelado por uma distribuição normal com média 1 e desvio padrão<sup>4</sup>  $\sigma$ , pois o valor máximo de similaridade dado pelo modelo de observação é 1 e quando isso acontecer, a Gaussiana deverá retornar o maior valor. Então, a probabilidade

---

<sup>3</sup>O operador lógico XNOR é o inverso do operador XOR. Dado dois sinais binários, ele retornará 1 se os dois sinais de entrada forem iguais e 0 se os sinais forem diferentes.

<sup>4</sup>Empiricamente definido como  $\sigma = 0.15$ .

Figura 3.2: Exemplo de como são computadas as medidas observadas, sendo (a)  $\mathbf{z}_t$  as medidas do VANT e (b)  $\mathbf{z}_t^*$  as medidas da partícula. Em cada imagem, apenas dois pares de pontos são considerados para demonstração, mas na prática esse número é bem maior. Realiza-se a comparação binária entre os valores de intensidade dos pixels e constrói-se o vetor binário. A diferença entre ambas imagens está ilustrada pela diferença entre os descritores calculados.



da leitura é definida como

$$\begin{aligned}
 p(\mathbf{z}_t | \mathbf{x}_t, \mathbf{m}) &= \mathcal{N}(1.0, \sigma) \\
 &= \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-1)^2}{2\sigma^2}\right)
 \end{aligned}
 \tag{3.7}$$

sendo  $x = \text{sim}(\mathbf{z}_t, \mathbf{z}_t^*)$ .

### 3.3 Aplicação no MCL

A abordagem proposta para estimar a postura do VANT é incorporar o modelo de observação proposto neste trabalho ao MCL. Dado um mapa 2D, estima-se a localização de um VANT com 4 DoF: latitude, longitude, altura e ângulo *yaw*, ou seja, orientação em torno do eixo vertical. As outras orientações, *roll* e *pitch*, são consideradas próximas a zero, que pode ser alcançado através do uso de um gimbal fixado à camera.

O MCL é implementado considerando três passos: amostragem, importância

(pesagem), e reamostragem. O foco está no passo de importância, que atribui para cada partícula um peso proporcional à similaridade entre as leituras da partícula e do VANT. Este peso,  $w(\mathbf{p}_t)$ , da partícula  $\mathbf{p}$  é dado pela probabilidade computada pelo modelo descrito na Seção 3.2, e considera a leitura,  $\mathbf{z}_t$ , tomada pelo VANT no instante  $t$ , e  $\mathbf{z}^*(\mathbf{p}_t)$ , a leitura estimada associada à postura da partícula,  $\mathbf{x}(\mathbf{p}_t)$ :

$$w(\mathbf{p}_t) = p(\mathbf{z}_t | \mathbf{x}(\mathbf{p}_t), \mathbf{m}). \quad (3.8)$$

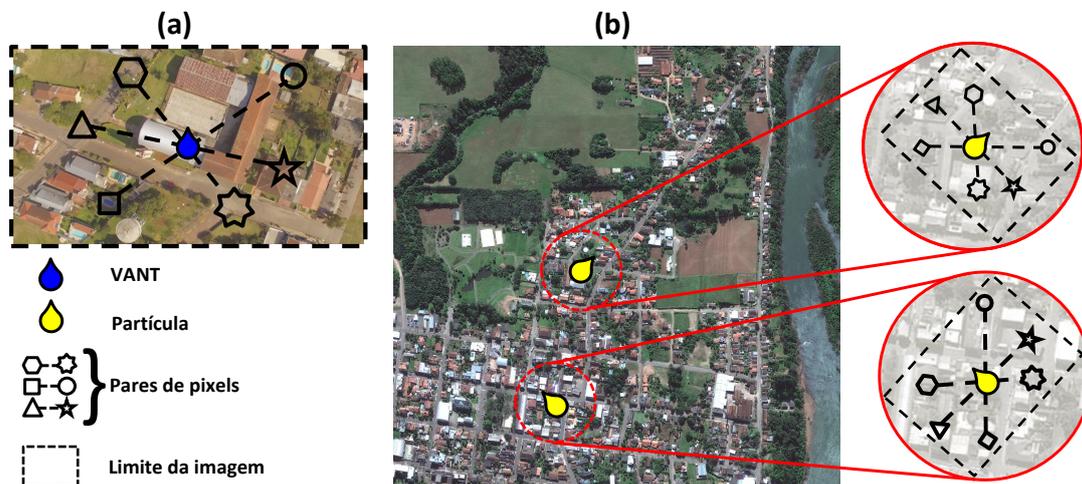
É importante destacar algumas das características do descritor abBRIEF que são bem aproveitadas pelo MCL e que foram herdadas do BRIEF original. Ele é variante a rotações e escalas, e ao usá-lo no modelo de observação do MCL, este descritor funciona muito bem para pesar as partículas de maneira que apenas aquelas que tenham postura semelhante ao VANT tenham peso alto. Por exemplo, considere que duas partículas estão na posição correta do mapa, mas uma delas está com a orientação correta enquanto que a outra está com uma orientação oposta. Apesar de estarem no mesmo lugar, aquela que está com orientação oposta irá receber um peso menor do que aquela com orientação correta. Isso não aconteceria se o descritor fosse invariante a rotação, já que nesse caso ele seria capaz de dizer que apesar da orientação oposta, ambas partículas continuam cobrindo a mesma imagem do mapa. Uma vez que apenas partículas na orientação correta são mantidas pelo MCL, o modelo de movimento pode ser simplificado a uma translação e uma variação na orientação. De forma mais detalhada, o modelo de movimento recebe as estimativas de um algoritmo de odometria visual. As informações que compõem esta estimativa da odometria são de deslocamento em três dimensões ( $x$ ,  $y$  e  $z$ ), que posteriormente serão incorporados a um erro, já que a estimativa não é perfeita. Portanto, o movimento realizado pelo robô que é aplicado às partículas é o resultado da estimativa da odometria visual somado a um erro, tudo isso calculado através do modelo de movimento.

O uso de pares de pixels por parte do descritor é outra característica muito aproveitada. Na Seção 2.4 foi apresentada a forma com que eles são amostrados, e fica claro que não é necessário nenhum processamento para encontrá-los já que eles são escolhidos por uma distribuição Gaussiana. Diferente disso, se *features* fossem utilizadas, seria necessário percorrer toda a imagem em busca deste tipo de informação, gastando muito tempo e recursos computacionais.

Além disso, outra vantagem do uso dos pares de pixels está ilustrada na

Figura 3.3. O modelo de observação proposto compara o descritor da imagem capturada pelo VANT, Figura 3.3-(a), com o descritor de cada partícula, Figura 3.3-(b). Computar o descritor da imagem capturada pelo VANT é muito simples, pois ela nunca está rotacionada ou escalada, basta distribuir os pares e calcular o descritor. Porém, em relação a coordenada global da imagem de satélite, as partículas podem estar transladadas, rotacionadas ou escaladas. Então, para computar o descritor de

Figura 3.3: Exemplo das transformações dos pares de pixels do descritor abBRIEF entre a imagem do VANT e as partículas. (a) imagem capturada pelo VANT, sua postura em relação a ela e como alguns pares de pixels do abBRIEF estão amostrados. (b) imagem de satélite em que as partículas são distribuídas para estimar a postura do robô, assim como os pares do abBRIEF aplicado em relação às suas respectivas posturas. Apenas duas estão ilustradas para facilitar a compreensão.



cada uma é necessário extrair o que ela observa. No caso do abBRIEF, é necessário apenas transladar, rotacionar e escalar o conjunto de pares de pixels considerando a postura de cada partícula, de forma que eles fiquem orientados na mesma direção do VANT. Estas transformações são mais leves do que literalmente recortar um pedaço da imagem de satélite que corresponde a partícula, para então poder computar o descritor, como é o caso do MI.

Finalmente, assume-se que os dados de odometria visual podem ser obtidos a partir de sobreposições subsequentes de imagens tiradas pela câmera do VANT. No caso deste trabalho, a transformação entre  $\mathbf{I}_{t-1}$  e  $\mathbf{I}_t$  é estimada baseado em *features*. Foi utilizado o SIFT para extrair e descrever as *features* das imagens em ambos instantes. Em seguida, foi realizada a correspondência entre as *features* das duas imagens e calculou-se a homografia para encontrar a transformação de uma para outra, sendo esta a informação da odometria visual. É importante esclarecer

que as informações de odometria são necessárias apenas para o modelo de movimento utilizado pelo MCL e que nenhuma informação proveniente disso é utilizada pelo modelo de observação proposto neste trabalho. É possível substituir o uso de odometria visual por, por exemplo, um filtro de Kalman e informações de IMU ou outros sensores disponíveis nos VANTs. Porém, para este caso, as imagens já estavam disponíveis, os cálculos de odometria poderiam ser feitos previamente e não importaria como as informações de odometria seriam obtidos, por isso a abordagem baseada em *features* foi utilizada.

## 4 EXPERIMENTOS E DISCUSSÃO

Neste Capítulo serão discutidos os testes que foram feitos durante o desenvolvimento do modelo de observação baseado no abBRIEF e os resultados obtidos. Os testes apresentados a seguir seguem uma linha de pensamento, de modo que uma seção completa a outra.

Inicialmente será discutida a necessidade de se usar o MCL para este problema. De uma forma simplificada e didática, o problema abordado neste trabalho é localizar, em imagens de satélite, a imagem capturada pelo VANT. Para isso, utiliza-se partículas como hipóteses da localização do robô, extraindo-se pequenas imagens da imagem de satélite, relativas a cada partícula, comparando-as então com a imagem do VANT. Ao distribuir as partículas por toda a imagem de satélite, o MCL consegue fazer uma busca global, comparando várias partes da imagem de satélite com a imagem capturada pelo robô. Outra possível maneira de fazer esta busca é utilizar a correspondência entre *features* para determinar a transformação entre as imagens. Foram feitos alguns testes para avaliar o desempenho desta abordagem de correspondência, para então justificar o uso do MCL.

O próximo passo é mostrar os motivos pelos quais o descritor abBRIEF é utilizado no modelo de observação do MCL. Existem outros métodos na literatura que são utilizados para determinar a similaridade entre imagens e que poderiam substituir o uso do descritor proposto no modelo de observação. Um teste comparativo entre ele e outros métodos será apresentado com o objetivo de justificar a escolha do abBRIEF para ser usado no modelo de observação.

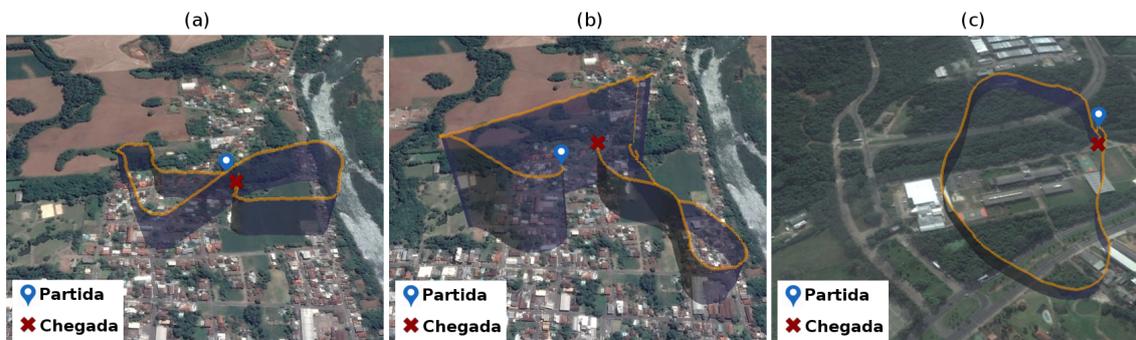
Em seguida são avaliados quais os valores ideais para dois parâmetros importantes do sistema. Algumas hipóteses foram levantadas para o número de partículas do MCL e a quantidade de pares de pixels do abBRIEF. Estes valores foram combinados e testados a fim de obter os valores que proporcionam o melhor resultado.

Por fim, os últimos testes têm o objetivo de mostrar que as modificações propostas pelo descritor abBRIEF em relação ao BRIEF original são necessárias para que o sistema seja robusto em diversos mapas. Ambos descritores foram usados como modelo de observação para o MCL e foram testados nos mesmos cenários.

#### 4.1 Especificação do ambiente de experimentação

O modelo de observação proposto foi avaliado utilizando as imagens capturadas por dois VANTs, DJI Phantom 3 Standard<sup>1</sup> e DJI Phantom 3 Advanced<sup>2</sup>. Os dois modelos são semelhantes, possuem quatro hélices rotativas (quadricópteros), um sensor de GPS, câmera embarcada e um gimbal. Pesam pouco mais de 1200g e o tempo máximo de voo é de aproximadamente 25 minutos. Para os testes, ambos foram controlados remotamente por uma pessoa. Os vôos realizados por estes robôs foram registrados em vídeos com 30fps, amostrados em 1fps para os experimentos. O primeiro modelo realizou dois vôos sobre a cidade de Arroio do Meio, Rio Grande do Sul, identificados aqui por *Flight 1* e o *Flight 2*. Durante o vôo *Flight 1*, Figura 4.1-(a), percorreu-se uma trajetória de 1.800m, capturando 358 imagens e variando a altitude entre 35 e 130m. No vôo *Flight 2*, Figura 4.1-(b), foi percorrida uma trajetória de 2.400m, capturando 500 imagens e com variação de altitude entre 40 e 300m. O segundo modelo de VANT realizou um vôo sobre a Universidade Federal do Rio Grande do Sul (UFRGS), Campus do Vale, Porto Alegre, Figura 4.1-(c), identificado aqui por *Flight 3*. Neste vôo, o VANT percorreu uma trajetória de 1.090m, capturando 147 imagens e variando a altura entre 45 e 70m. Ambos modelos de VANTs possuem gimbal, uma ferramenta que mantém a câmera apontada para uma direção especificada, neste caso mantendo a câmera perpendicular ao solo.

Figura 4.1: Trajetórias dos vôos utilizados durante os testes deste trabalho. (a) *Flight 1*. (b) *Flight 2*. Ambos foram registradas em Arroio do Meio. (c) *Flight 3* foi registrado em Porto Alegre, UFRGS - Campus do Vale. Os dois locais pertencem ao estado do Rio Grande do Sul, Brasil.



<sup>1</sup><http://www.dji.com/phantom-3-standard/info>

<sup>2</sup><http://www.dji.com/phantom-3-adv/info>

Para testar a robustez do modelo proposto em relação a variação dos mapas, foram extraídos cinco diferentes imagens de satélites de ambas áreas de vôo. São identificados aqui por *AdM-10-2010*, *AdM-08-2013*, *AdM-01-2014*, *AdM-03-2014* e *AdM-06-2015* para a região de Arroio do Meio-RS, e *UFRGS-02-2013*, *UFRGS-09-2013*, *UFRGS-03-2015*, *UFRGS-05-2015* e *UFRGS-01-2017* para a região da UFRGS. Os 10 mapas foram obtidos através do Google Earth™ e seus nomes correspondem a respectiva região (*AdM* ou *UFRGS*), seguido pelo mês e ano que estão registrados no Google™ Earth. Cada mapa de Arroio do Meio engloba uma área de aproximadamente  $1.16km^2$ , enquanto que os mapas da UFRGS equivalem a aproximadamente  $0.7km^2$ . Estes mapas apresentam grande variação entre si e em relação as imagens capturadas pelos robôs. A Figura 4.2 ilustra alguns exemplos destas mudanças nos mapas de Arroio do Meio, como a tonalidade do rio e das áreas verdes, os efeitos de perspectiva, mudanças de luminosidade e na vegetação e até mesmo o número de carros nas ruas. Por questões de organização, os cinco mapas de Arroio do Meio estão disponíveis no Apêndice B.1, e os da UFRGS no Apêndice B.2.

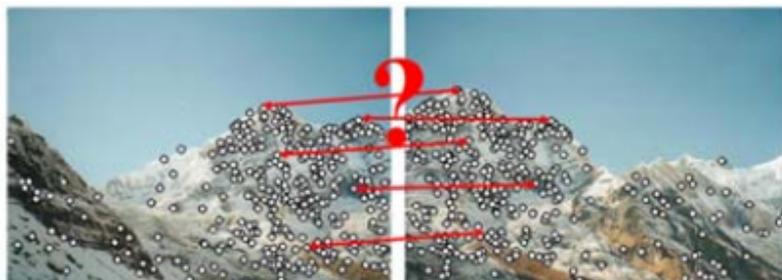
Figura 4.2: Exemplos de variações entre os mapas e a imagem capturada pelo VANT. A primeira linha mostra as imagens capturadas pelo VANT, enquanto que as outras linhas correspondem a mesma posição do robô em cada mapa.



## 4.2 Correspondência entre *Features*

Como já levantado anteriormente, é possível utilizar *features* para comparar imagens. Depois que elas são extraídas, o próximo passo é determinar a correspondência entre elas, ou seja, determinar quais *features* correspondem ao mesmo local em imagens diferentes. Imagine duas imagens capturadas em sequência com grande sobreposição entre ambas, como é ilustrado pela Figura 4.3. É bem provável que grande parte das *features* extraídas da primeira estarão presentes na próxima, então basta realizar uma comparação entre os descritores destas *features* e encontrar a correspondência.

Figura 4.3: Exemplo de correspondências entre *features* de diferentes imagens. Cada ponto branco nas imagens representam uma *feature* e as linhas vermelhas são as correspondências encontradas entre elas. Figura extraída de (SZELISKI, 2010).



Para o contexto de localização global de VANTs, ao realizar esta operação o MCL se tornaria desnecessário, já que seria possível extrair *features* da imagem capturada pelo robô e da imagem de satélite e encontrar as correspondências entre elas, obtendo então a postura do VANT.

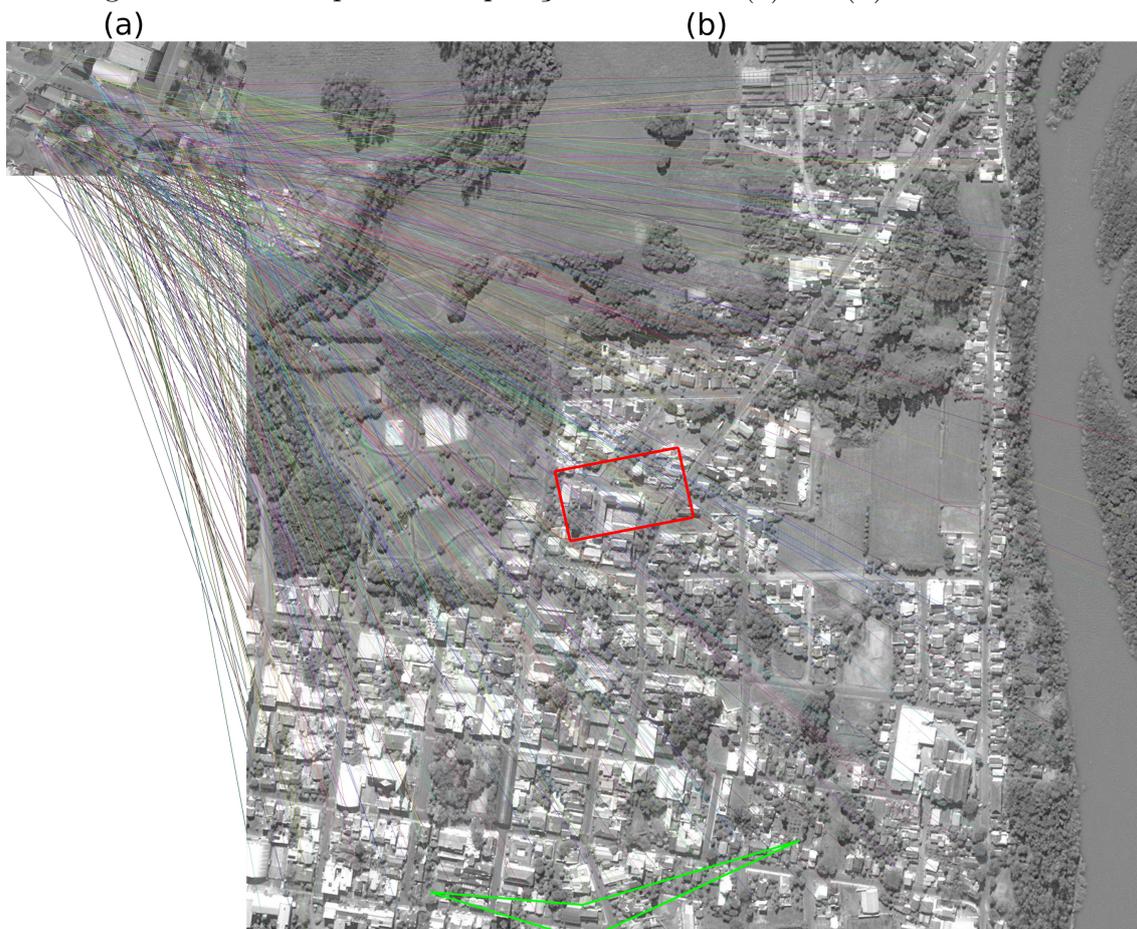
Esta abordagem parece razoável, porém existem alguns problemas tais como correspondências erradas, *features* de uma imagem inexistentes na outra, baixa qualidade da imagem de satélite e até mesmo descritores não tão robustos a variações de luminosidade ou outros aspectos. Como consequência, o resultado pode ser muito ruim.

Para testar esta abordagem no problema deste trabalho, várias imagens da trajetória *Flight 1* foram comparadas com um dos mapas disponíveis. O resultado de uma comparação é apresentado na Figura 4.4. A extração e a descrição das *features* foram realizadas utilizando ORB (RUBLEE et al., 2011). A correspondência de *features* foi feita através do método FLANN (MUJA; LOWE, 2009). Por fim, calculou-se a homografia<sup>3</sup> para posicionar a Figura 4.4-(a) na Figura 4.4-(b). Este

<sup>3</sup>Em Visão Computacional, para qualquer duas imagens da mesma superfície planar, a homo-

cálculo foi feito utilizando RANSAC (FISCHLER; BOLLES, 1981) com 200.000 iterações, e o resultado é o conjunto de linhas verdes na parte inferior da Figura 4.4-(b). Todas as implementações deste teste utilizaram as bibliotecas do OpenCV.

Figura 4.4: Teste de correspondência de *features* usando uma imagem capturada pelo VANT, (a), e uma de satélite, (b). O propósito do teste é encontrar (a) em (b) utilizando *features*. Cada linha que parte de (a) e vai até (b) é uma correspondência. As linhas verdes na parte inferior de (b) representam a transformação das bordas da imagem (a) para a imagem (b) através da homografia encontrada, enquanto que o retângulo vermelho representa a posição correta de (a) em (b).



Neste teste foram extraídas 500 *features* da Figura 4.4-(a) e 310.202 da Figura 4.4-(b). Nota-se que o resultado obtido não foi o esperado, uma vez que existem correspondências da Figura 4.4-(a) com praticamente toda Figura 4.4-(b), e não apenas na região correta, identificada pelo retângulo vermelho. Conclui-se, então, que utilizar apenas a correspondência de *features* para encontrar uma imagem capturada pelo VANT em uma imagem de satélite não é suficiente, sendo necessária

---

grafia é a relação entre ambas. Pode ser utilizada, por exemplo, para retificação de imagens ou estimativa de movimento de câmeras.

outra abordagem, como o MCL, que também pode fazer essa comparação global do mapa.

### 4.3 Métodos de comparação de imagens

Uma das vantagens do MCL é que os modelos de observação e de movimento podem ser alterados para se adequar ao robô, à aplicação e aos sensores disponíveis. Por exemplo, um robô humanoide exige um modelo de movimento diferente de um robô com esteiras, mas se ambos utilizam uma câmera como principal sensor para se localizar, o mesmo modelo de observação pode ser usado para ambos.

O modelo de observação que está sendo proposto, abBRIEF, é baseado em uma variação do descritor BRIEF, mas ele pode ser substituído por outros métodos de comparação de imagens. Para justificar o seu uso para o problema de localização utilizando MCL, serão feitas algumas análises de tempo de processamento gasto e desempenho considerando alguns métodos: MI, *Template Matching*, Histograma, ORB, SURF e SIFT. O objetivo é mostrar o comportamento de cada um como se estivessem substituindo o descritor proposto no modelo de observação, mostrando seu desempenho ao comparar a imagem do VANT com as imagens das partículas e o custo necessário para estas comparações. Mas, é importante destacar que nenhum destes métodos foi aplicado ao MCL para os testes desta seção, eles foram executados isoladamente. No caso da tabela de análise de tempo, Tabela 4.2, as colunas de "extração" se referem ao custo caso o respectivo método fosse utilizado no modelo de observação do MCL, mas isso não significa que realmente isso aconteceu para os testes. O computador utilizado nos testes possui 8 GB de memória RAM e um processador Intel Core i7-4790 3.60GHz.

Antes de explicar como os testes foram feitos, é importante explicar rapidamente como cada método foi utilizado para determinar a similaridade. O abBRIEF é o descritor proposto por este trabalho e já foi explicado no Capítulo 3. Para este teste foram utilizados 256 pares de pixels para o abBRIEF. O MI foi utilizado pelo trabalho de Yol et al. (2014) e sua explicação já foi dada na Seção 2.5. No método *Template Matching*, a imagem de busca é deslizada sobre a imagem de referência e calcula-se, para cada possível posição da janela, a similaridade através do *Normalized Cross Correlation*, NCC (BRIECHLE; HANEBECK, 2001). Então, a similaridade máxima é considerada a das imagens. Para o método de Histograma,

computa-se o histograma das duas imagens em escala de cinza e a similaridade é obtida comparando os dois histogramas, novamente utilizando como medida de distância o NCC. Por fim, os três últimos métodos, ORB (RUBLEE et al., 2011), SURF (BAY; TUYTELAARS; GOOL, 2006) e SIFT (LOWE, 2004), seguiram o mesmo procedimento para determinar a similaridade. Primeiro, faz-se a extração das *features* das duas imagens, seguido pela correspondência entre elas realizado pelo método FLANN (MUJA; LOWE, 2009). Calcula-se a homografia entre ambas imagens e realiza-se a contagem do número de correspondências de *features* que corroboram a homografia, ou *inliers*. Estes são utilizados como medida de similaridade, de forma que quanto maior o número de *inliers*, mais semelhantes são as imagens. Todos os métodos tiveram como base as implementações disponíveis na biblioteca de processamento de imagens OpenCV 3.1<sup>4</sup>.

Para este teste foram utilizadas 12 imagens capturadas pelo VANT que fazem parte do vôo *Flight 1*. Foram extraídas, manualmente, de um mapa 12 imagens que correspondem a mesma posição destas 12 imagens do *Flight 1*, ou seja, imagens alinhadas com a posição correta em que o VANT estava quando as capturou. Ainda, suponha que em cada uma destas exista uma partícula cuja postura também está correta em relação a postura do VANT, simulando a execução do MCL e a comparação entre a imagem do VANT com a imagem da partícula. Então, mediu-se a similaridade de cada uma das 12 imagens capturadas pelo VANT com as 12 extraídas do mapa, buscando encontrar a imagem que mais se assemelha com a capturada por ele. Quando o método de comparação atribui a maior similaridade para a imagem da partícula que mais se assemelha a imagem do VANT, considera-se um acerto. Este procedimento foi repetido para os cinco mapas de Arroio do Meio-RS.

A Tabela 4.1 apresenta o desempenho de cada método ao medir a similaridade entre cada imagem do VANT com as 12 imagens das partículas e considerando os cinco mapas. Olhando para a tabela, quando um método obtém um número de acertos igual a cinco para alguma imagem, significa que, para todos os mapas, ele foi capaz de atribuir o maior valor de similaridade para a imagem da partícula que mais se assemelha a imagem correta do VANT (por exemplo, a imagem da partícula, em cada mapa, que mais se assemelha a imagem I1 do VANT é a primeira. Se um método atribuiu o maior valor de semelhança para este par, ele acertou.). De maneira geral, é possível observar que o descritor proposto neste trabalho, abBRIEF, possui

---

<sup>4</sup><http://opencv.org/opencv-3-1.html>

um resultado positivo, obtendo o valor máximo de acertos em seis dos doze testes (I1, I2, I5, I9, I10 e I11). O único método que o supera é o MI, que obteve valor máximo em oito dos doze testes (I1, I2, I3, I4, I5, I9, I10 e I12). Ainda, nos casos em que o MI foi superior ao abBRIEF, houve apenas uma pequena diferença, como é o caso das imagens I3, I4 e I12, mostrando que ambos são bons candidatos para realizar a comparação entre as imagens. Os piores resultados foram gerados pelos métodos baseados em *features*, como ORB, SURF e SIFT. Na Seção 4.2, já foi demonstrado que o uso de *features* para este tipo de aplicação não é muito adequado. Aqui, apesar de utilizar apenas um pedaço da imagem de satélite, os resultados não melhoraram e estes métodos não foram capazes de informar corretamente que as imagens eram similares. Novamente, um dos problemas é a baixa definição de bordas e detalhes na imagem do mapa, prejudicando a extração de *features*.

Tabela 4.1: Número de acertos de cada método considerando cada uma das 12 imagens capturadas pelo VANT, I1 até I12. O valor de acerto representa o número de mapas em que um método foi capaz de identificar corretamente, dentre as 12 imagens extraídas de cada mapa, a que mais se assemelhava com a imagem capturada pelo VANT. Como foram utilizados cinco mapas, o valor máximo de acertos é 5.

Métodos	Acertos (Máximo 5)											
	I1	I2	I3	I4	I5	I6	I7	I8	I9	I10	I11	I12
<b>abBRIEF</b>	5	5	3	4	5	0	0	0	5	5	5	4
<b>MI</b>	5	5	5	5	5	1	0	0	5	5	4	5
<b>Template</b>	5	5	1	3	2	0	0	0	3	5	5	5
<b>Histograma</b>	1	0	0	2	1	1	0	0	0	0	0	0
<b>ORB</b>	0	1	1	1	1	1	0	0	0	0	0	0
<b>SURF</b>	2	0	0	0	0	0	0	0	0	0	0	0
<b>SIFT</b>	1	0	0	0	0	0	0	0	0	0	0	1

Além do desempenho em relação a comparação de imagens, o tempo gasto por cada método também deve ser analisado antes de usá-lo no modelo de observação para o MCL. Isso se justifica pela quantidade de partículas necessárias para realizar a localização, ou seja, se o tempo para comparar uma imagem capturada pelo VANT com uma imagem de uma partícula é alto, o tempo para comparar com todas as partículas será proibitivo. A Tabela 4.2 apresenta os tempos médios obtidos por cada método nos testes, separando em execução, que corresponde ao tempo gasto por cada método para realizar a comparação, e em extração, que corresponde ao tempo de extrair a imagem da partícula da imagem de satélite.

Os menores tempos pertencem ao descritor proposto, abBRIEF, já que para sua execução basta construir o descritor binário analisando uma quantidade de pares

de cada imagem e compará-los. É importante destacar que não é necessário extrair, do mapa, a imagem da partícula para se obter os descritores abBRIEF, é possível apenas transladar, rotacionar e escalar os pontos que compõem os pares de pixels para as posições corretas no mapa, com base na postura da partícula. Por outro lado, isso não acontece com os demais métodos, uma vez que suas formas de comparação exigem este recorte, que é uma operação computacionalmente custosa. Por fim, é importante destacar novamente que estes valores representam o tempo gasto para comparar a imagem do VANT com apenas uma partícula. Se fosse comparada com todas as partículas, o tempo cresceria proporcionalmente ao número de partículas.

Por exemplo, imagine que 50.000 partículas serão utilizadas pelo MCL e que o modelo de observação fosse baseado no MI. Considerando o seu tempo médio total de 61 milisegundos, menos o seu desvio padrão de 22 milisegundos (para considerar o melhor caso), tem-se o custo de 39 milisegundos para comparar a imagem do VANT com a imagem de uma partícula. Multiplicando este valor pelo número de partículas e fazendo a conversão de milisegundos para minutos, o custo total de tempo para o MCL, utilizando o MI, comparar uma imagem capturada pelo VANT com todas as partículas é de aproximadamente 32.5 minutos. Fazendo o mesmo para o abBRIEF, o custo total de tempo é de 0.23 minutos. Na execução do MCL, utilizam-se oito *threads* para paralelizar a pesagem das partículas e reduzir o tempo total. O efeito da *cache* do sistema também ajuda na redução do tempo total, já que quando o filtro converge, apenas parte da imagem de satélite é utilizada e esta já fica armazenada na *cache*. Na prática, utilizando 50.000 partículas todo o processamento ocorre em menos de 1 segundo.

Tabela 4.2: Tempo gasto, em milisegundos, por cada método para comparar duas imagens, analisado individualmente a execução e a extração. A coluna de execução corresponde ao tempo gasto para o respectivo método comparar duas imagens de mesmo tamanho, independente do custo para mover os pares ou extrair as imagens de algum lugar. Já a coluna de extração corresponde ao tempo gasto para, considerando a postura da partícula, extrair a sua imagem para então fazer a comparação. O Desvio Padrão foi abreviado para D.P. por questões de organização.

Métodos	Tempo em milisegundos					
	Execução		Extração		Total	
	Média	D. P.	Média	D. P.	Média	D.P.
<b>abBRIEF</b>	0,3	0,029	0,00002	0,000003	0,3	0,029
<b>MI</b>	28	1	32	21	61	22
<b>Template</b>	50	1	32	21	83	22
<b>Histograma</b>	2	1	32	21	34	22
<b>ORB</b>	77	3	32	21	110	24
<b>SURF</b>	665	70	32	21	697	91
<b>SIFT</b>	1336	104	32	21	1368	125

Por fim, apenas dois métodos apresentaram bons resultados no quesito de comparação de imagens, MI e abBRIEF. Apesar do MI ter uma certa vantagem nisso em relação ao abBRIEF, seu tempo de processamento é duas ordens de grandeza maior. Isso o torna inadequado para ser utilizado no modelo de observação para o MCL. Além do alto tempo necessário para comparar as imagens, este método também exige que a imagem das partículas seja literalmente recortada da imagem de satélite, o que agrega mais custo computacional e de tempo. Então, de maneira geral, o MI é levemente mais eficaz que o abBRIEF, mas é menos eficiente, o que justifica o uso do abBRIEF.

#### 4.4 Número de partículas e quantidade de pares de pixels

O número de partículas utilizadas no MCL é muito importante para uma boa estimativa da localização. Antes de defini-lo, é necessário analisar o tamanho do espaço de busca da aplicação, que relaciona-se ao número de DoF do robô, a quantidade de recursos computacionais disponíveis e os modelos de observação e de movimento. Isso é necessário pois quanto maior o número de partículas, maior será a cobertura do espaço de busca e existirão mais chances de algumas partículas estarem próximo à postura correta no início. Porém, usar um número elevado exige mais poder de processamento. Então, é fundamental um estudo da relação entre cobertura do espaço de busca e custo computacional.

O modelo de observação utilizando o abBRIEF também apresenta uma relação semelhante entre o número de pares e o custo computacional. Quanto maior o número de pares utilizados, mais singular será o descritor. Porém, com mais pares aumenta-se o custo para transladar, rotacionar, escalar e comparar os pares de cada partícula.

Então, no contexto deste trabalho, é importante descobrir qual é o número de partículas e a quantidade de pares de pixels necessários para atingir um bom resultado na localização do VANT. Para o número de partículas foram levantadas três hipóteses: 10.000, 20.000 e 50.000. Considerando que o espaço de busca é muito grande, com 4 DoF e um mapa extenso, a intenção é analisar se um valor baixo, neste contexto, como 10.000 partículas, é suficiente ou se é necessário um valor maior, como 50.000, para obter um resultado satisfatório. As quantidades de pares de pixels testadas no descritor abBRIEF foram 128, 256 e 512. A proposta é analisar se um descritor menos singular atinge os resultados esperados, ou se é necessário utilizar um número maior de pares, gerando assim um descritor mais robusto.

No total foram geradas nove combinações, testando todas as combinações de valores de partículas com as quantidades de pares, utilizando a trajetória *Flight 2* e 5 mapas diferentes. As posturas estimadas pelo MCL foram comparadas com os dados do GPS embarcado no VANT. É importante destacar que estes dados de GPS só foram utilizados para computar os resultados, sendo considerados como *ground-truth*.

Inicialmente, a postura das partículas  $\mathbf{x}_0^{[i]}$  são uniformemente distribuídas considerando  $\mathbf{x} \in \mathbf{m}$ ,  $\theta \in [0, 2\pi)$  e  $z(\mathbf{x}) \in [80m, 1000m]$ . Em cada instante  $t$  da trajetória, o cálculo da estimativa da localização do VANT é dado pela postura média,  $\bar{\mathbf{x}}_t = (\bar{x}_t, \bar{y}_t, \bar{z}_t)$ , sendo

$$\bar{\mathbf{x}}_t = \frac{\sum_{i=1}^{\mathbf{P}} w_t^{[i]} \mathbf{x}_t^{[i]}}{\sum_{i=1}^{\mathbf{P}} w_t^{[i]}}, \quad (4.1)$$

em que  $\mathbf{P}$  é o número de partículas,  $w_t^{[i]}$  é o peso e  $\mathbf{x}_t^{[i]}$  é a postura da  $i$ -ésima partícula.

Pela aleatoriedade do MCL e para obter um resultado estatisticamente significativo, cada combinação foi testada 30 vezes. Então, para obter o erro de estimativa de localização,  $err_t$ , ilustrado pela Figura 4.5, calculou-se a média da diferença entre

as posturas médias,  $\bar{\mathbf{x}}_t$ , e a postura dada pelo GPS,  $\mathbf{x}_t^r = (x_t^r, y_t^r, z_t^r)$ , sendo

$$err_t = \frac{\sum_{j=1}^{30} \sqrt{(x(\bar{\mathbf{x}}_{t,j}) - x(\mathbf{x}_t^r))^2 + (y(\bar{\mathbf{x}}_{t,j}) - y(\mathbf{x}_t^r))^2 + (z(\bar{\mathbf{x}}_{t,j}) - z(\mathbf{x}_t^r))^2}}{30}. \quad (4.2)$$

A Figura 4.5 apresenta o resultado com 50.000 partículas, combinado com 128, 256 e 512 pares de pixels, Figuras 4.5(a), (b) e (c), respectivamente. Analisando-a, percebe-se que a Figura 4.5-(a) apresenta o pior resultado, com um erro maior no mapa *AdM-10-2010* a partir do instante 280. Além disso, com esta quantidade de pares o filtro demora um pouco mais para convergir em relação as outras quantidades de pares de pixels. Quanto as configurações das Figuras 4.5-(b) e (c), constata-se que a diferença entre elas não é tão significativa, uma vez que o filtro convergiu aproximadamente no mesmo instante em ambos os casos, e o erro de estimativa da localização também se mostrou muito semelhante. Estas duas configurações se mostraram eficientes para o problema de localização, não havendo ganho suficiente na qualidade da localização que justificasse o uso de mais pares de pixels do que 256.

Os resultados das demais configurações, 10.000 e 20.000 partículas com as combinações das três quantidades de pares de pixels, são apresentados na Figura 4.6. Observa-se que 10.000 partículas é um valor baixo dado o espaço de busca neste cenário. As chances das partículas serem distribuídas próximas a postura correta diminui e o filtro acaba não convergindo, semelhante ao que acontece com 20.000 partículas. A respeito do número de pares de pixels, utilizar 128 acaba gerando um descritor pouco singular, incapacitando o modelo de observação de diferenciar as imagens adequadamente. Por outro lado, percebeu-se uma melhora para 256 e 512 pares, sendo que, de maneira geral, o melhor resultado considerando custo e benefício é quando se utiliza a combinação de 50.000 partículas e 256 pares de pixels.

Figura 4.5: Comparação entre a estimativa de localização com 50.000 partículas e quantidade de pares de pixels variando em: (a) 128 pares de pixels, (b) 256 pares de pixels e (c) 512 pares de pixels. O eixo vertical representa o erro, em metros, da estimativa da localização e o eixo horizontal representa o instante de tempo no decorrer da trajetória. Cada linha corresponde a um mapa.

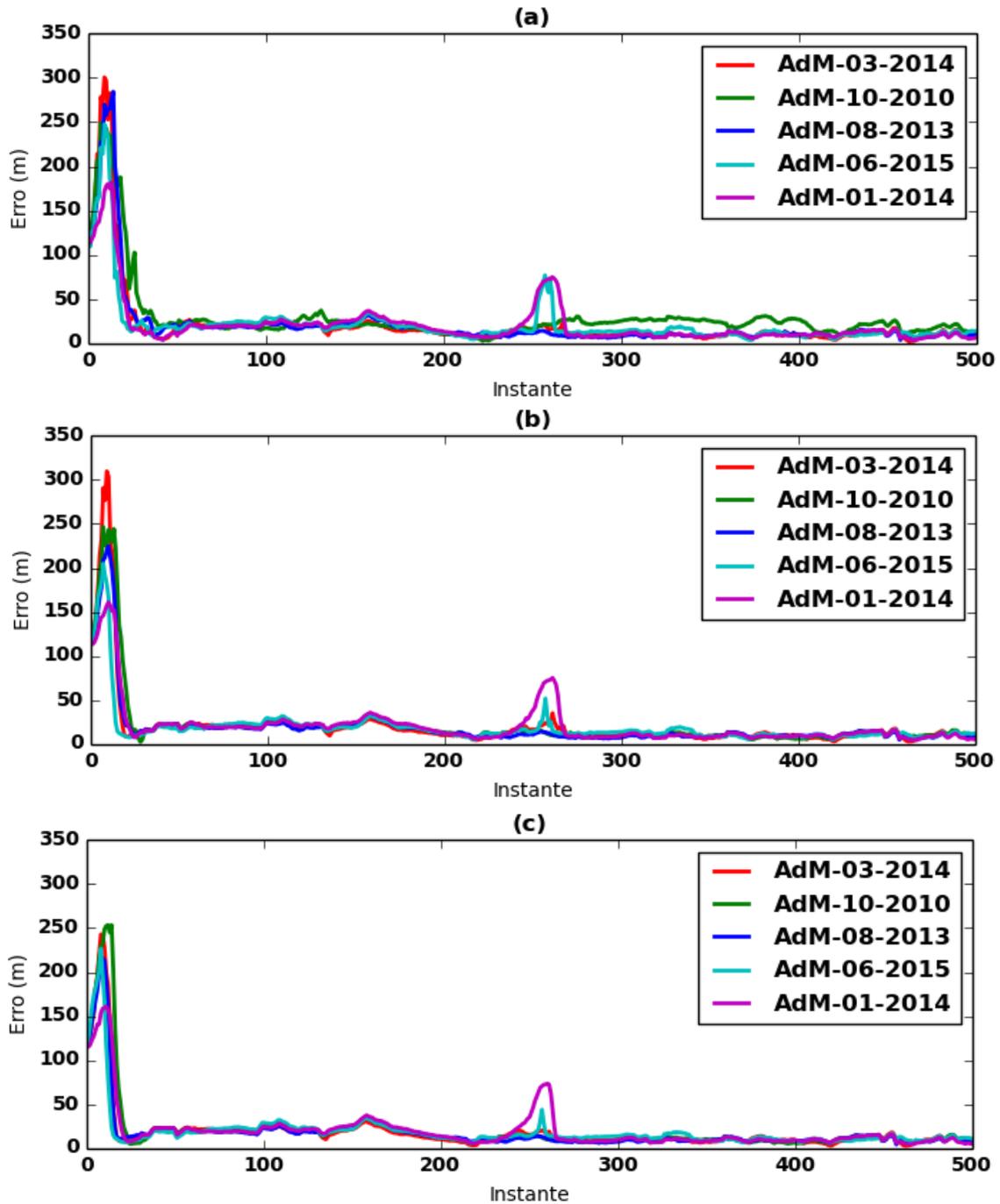
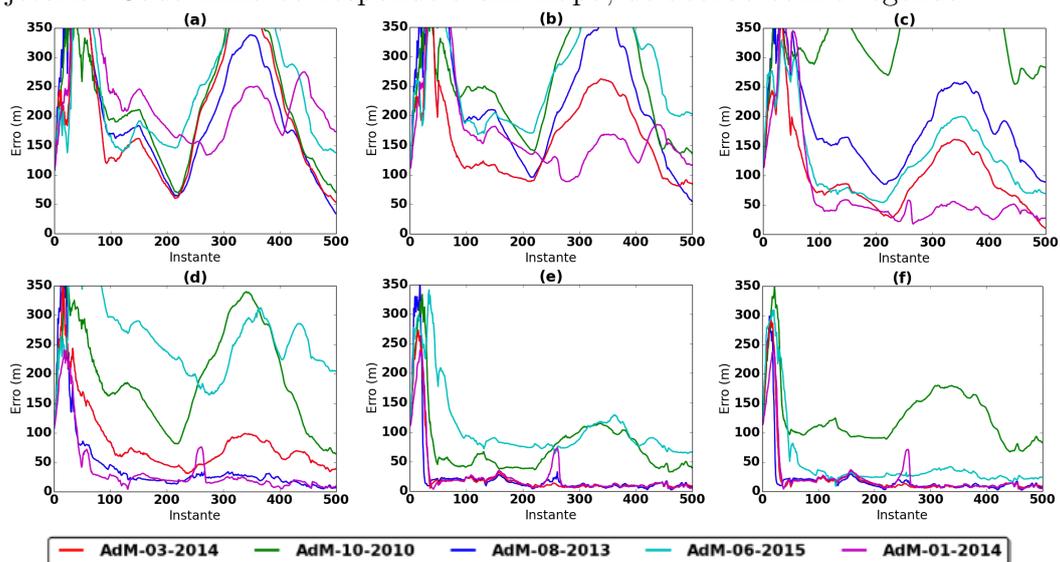


Figura 4.6: Resultados obtidos com 10.000 e 20.000 partículas, sendo: (a) 10.000 partículas e 128 pares; (b) 10.000 partículas e 256 pares; (c) 10.000 partículas e 512 pares; (d) 20.000 partículas e 128 pares; (e) 20.000 partículas e 256 pares; e (f) 20.000 partículas e 512 pares. O eixo vertical representa o erro, em metros, da estimativa da localização e o eixo horizontal representa o instante de tempo no decorrer da trajetória. Cada linha corresponde a um mapa, de acordo com a legenda.



#### 4.5 Comparação com o BRIEF original

As mudanças feitas no descritor BRIEF, que culminaram na criação do abBRIEF, buscaram tornar o descritor mais robusto, incluindo a informação de cor. Nesta sessão, compara-se o desempenho do descritor BRIEF e do abBRIEF utilizados no modelo de observação do MCL. Aqui estas mudanças serão demonstradas através dos testes que englobam as três trajetórias, os cinco mapas de cada região e os parâmetros já justificados na Seção 4.4. Estas mesmas configurações de testes serão executadas para o modelo de observação proposto utilizando o descritor abBRIEF discutido na Seção 3.1 e utilizando o descritor BRIEF original, apresentado na Seção 2.4. Para o abBRIEF foram utilizadas imagens no espaço de cor CIE  $L^*a^*b^*$  e para o BRIEF foram utilizadas imagens em escala de cinza.

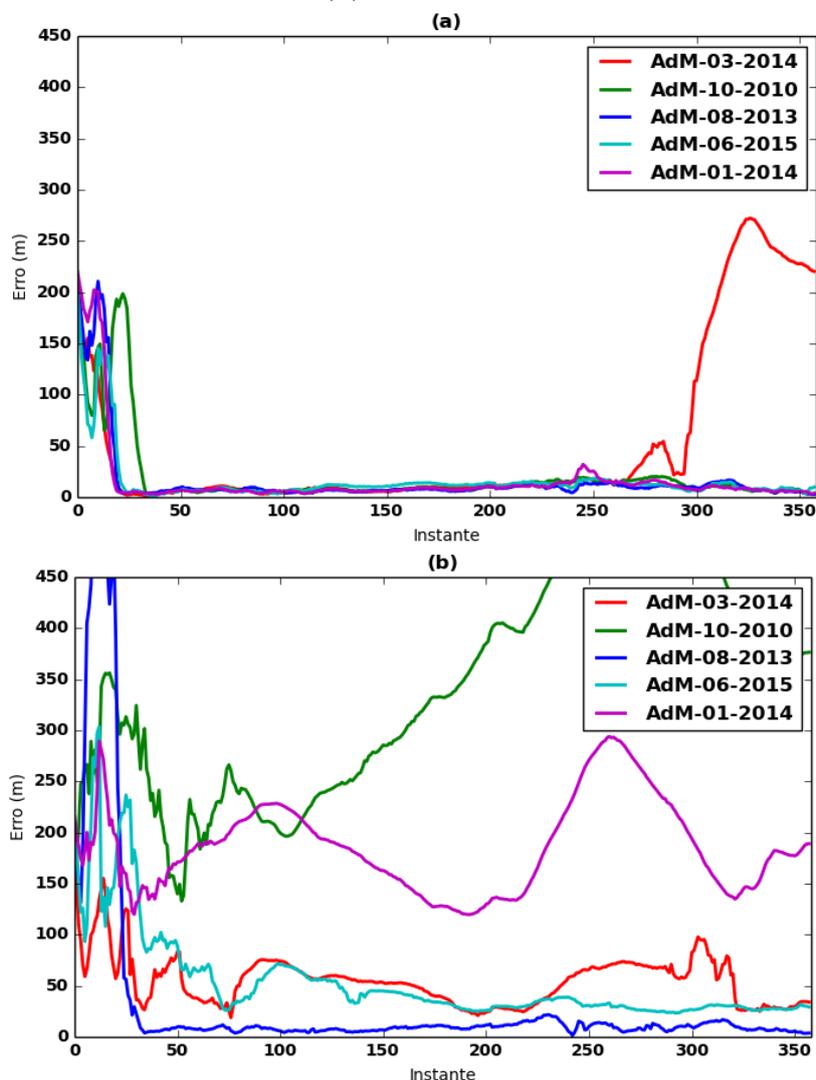
A fim de obter uma avaliação estatisticamente significativa, todos estes testes também foram repetidos 30 vezes por conta da aleatoriedade do método MCL. O procedimento para o cálculo do erro da estimativa é o mesmo apresentado na Seção 4.4, nas Equações 4.3 e 4.2. Os resultados serão apresentados por vô e em duplas. Por exemplo, para o *Flight 1* serão apresentados dois gráficos, um referente ao uso do descritor abBRIEF e outro referente ao uso do descritor BRIEF original. Dessa forma será possível fazer considerações sobre o desempenho de cada descritor considerando cada vô individualmente.

A Figura 4.7 apresenta os resultados obtidos na trajetória *Flight 1* pelo abBRIEF, Figura 4.7-(a), e pelo BRIEF, Figura 4.7-(b). Neste caso, fica claro a melhora na estimativa de localização alcançado pelas mudanças propostas no descritor BRIEF. Além de auxiliar o MCL a convergir mais rápido, a Figura 4.7-(a) também mostra que o abBRIEF auxilia o modelo de observação proposto a manter o erro de estimativa baixo em praticamente todos os mapas, falhando apenas no final do mapa *AdM-03-2014*. Este mapa em especial possui uma grande região que pode ser observada, em partes, pela Figura 4.8, em que o solo não tem vegetação e predominam cores marrons, enquanto que as imagens capturadas pelo VANT registram predominância de cores verdes. Além desta diferença, outro fator que contribui para o mau desempenho do MCL é o rio sobre qual o VANT sobrevoa, já que sua tonalidade muda de uma imagem para outra. Então, o MCL acaba convergindo para a postura incorreta e o erro da estimativa da localização aumenta.

Como os resultados anteriores da Seção 4.4 demonstraram, um desempenho

similar do uso de 256 e 512 pares combinados com 50.000 partículas, levantou-se a hipótese de que o uso de um descritor mais robusto poderia solucionar os problemas na trajetória *Flight 1* e mapa *AdM-03-2014*. Dessa forma, realizou-se um teste com 50.000 partículas e 512 pares no abBRIEF. O resultado deste teste está ilustrado na Figura 4.9 e demonstra que mesmo com o aumento no números de pares de pixels, as regiões com cores homogêneas continuam sendo um desafio.

Figura 4.7: Comparação dos resultados do descritor abBRIEF e do descritor BRIEF sendo usados no modelo de observação. A estimativa de localização do vôo *Flight 1* com 50.000 partículas e 256 pares de pixels em cada descritor. (a) representa o resultado utilizando o abBRIEF e (b) o resultado utilizando o descritor BRIEF.

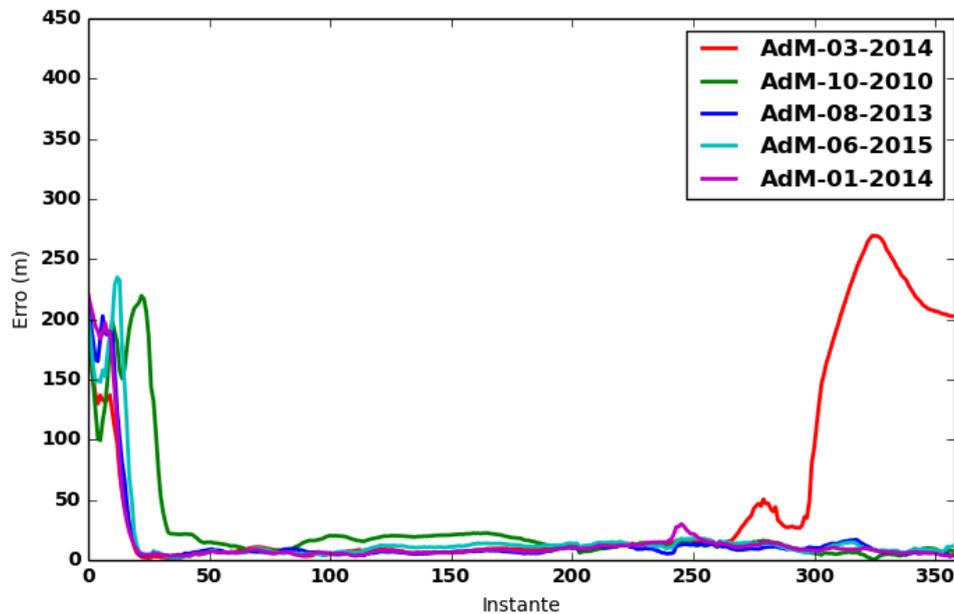


Como mostra a Figura 4.10-(b), o modelo de observação utilizando o descritor BRIEF original apresenta uma melhora quando aplicado no vôo *Flight 2*. Apesar desta melhora, em dois mapas este modelo acaba falhando e o erro da estimativa da localização aumenta muito. Por outro lado, neste vôo o modelo de observação que utiliza o abBRIEF apresenta, novamente, um resultado positivo e semelhante

Figura 4.8: Imagem da trajetória *Flight 1* (esquerda) com sua respectiva imagem do mapa *AdM-03-2014* (direita). A imagem do mapa possui um campo marrom no lugar do campo verde visto pelo VANT, assim como um rio escuro no lugar do rio mais claro no caso do VANT. Esta imagem ilustra o problema que prejudica a estimativa de localização na Figura 4.7-(a), por volta do instante 320.



Figura 4.9: Resultado do descritor abBRIEF sendo usado no modelo de observação. A estimativa de localização é feita no vôo *Flight 1* com 50.000 partículas e 512 pares de pixels no descritor. O erro na parte final da trajetória quando utilizado o mapa *AdM-03-2014* continua alto.



ao vôo *Flight 1*, com baixo erro de estimativa em praticamente toda a trajetória e rápida convergência por parte do MCL. Por volta do instante 220 os dois testes apresentam um aumento no erro de estimativa da localização, e enquanto o modelo de observação com o BRIEF original, Figura 4.10-(b), só reduz o erro no final do vôo, o outro é capaz de fazer isso logo em seguida, mostrando que é mais robusto. A Figura 4.11 mostra uma das imagens que foi capturada pelo VANT próxima ao instante do pico do erro da estimativa e percebe-se que há uma predominância da cor verde em ambas imagens. Neste instante o VANT sobrevoava uma grande região em que a cor verde predomina e isso acaba prejudicando o MCL.

Por fim, o *Flight 3* apresenta um cenário desafiador para ambos os modelos

Figura 4.10: Comparação dos resultados do descritor abBRIEF e do BRIEF sendo usados no modelo de observação. A estimativa de localização do voo *Flight 2* com 50.000 partículas e 256 pares de pixels em cada descritor. (a) representa o resultado utilizando o descritor abBRIEF e (b) o resultado utilizando o BRIEF.

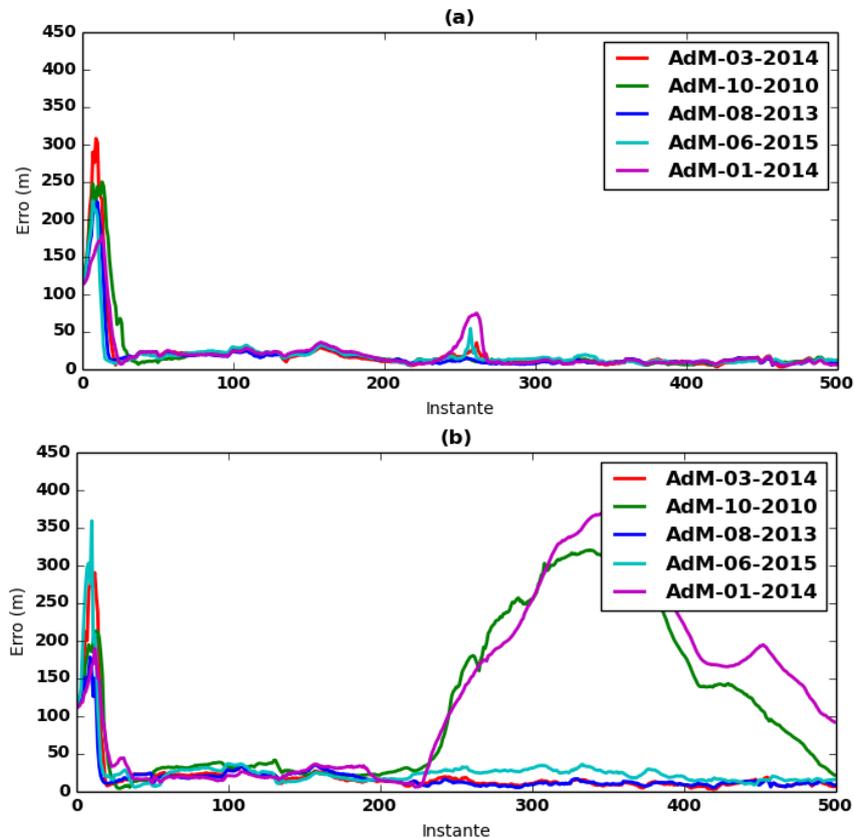


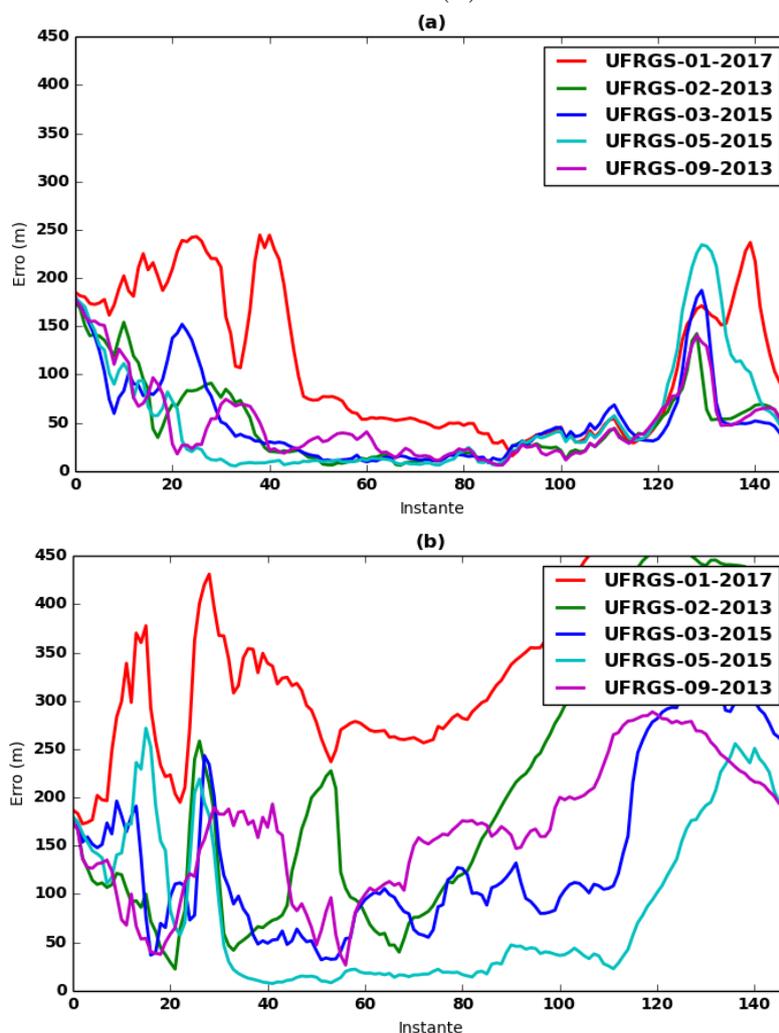
Figura 4.11: Imagem da trajetória *Flight 2* (esquerda) com sua respectiva imagem do mapa *AdM-01-2014* (direita). Ambas imagens mostram que estão em uma região uniforme, com predominância da cor verde. Esta região causa um pico de aumento no erro ilustrado pela Figura 4.10-(a) e (b), próximo ao instante 260.



de observação testados nesta seção. Por conta das diversas diferenças entre as imagens e dos problemas já discutidos nos capítulos anteriores, não é apropriado utilizar um descritor que se baseie em detalhes das imagens. Os dois descritores utilizados nestes testes não se baseiam nisso, descrevendo a imagem sem levar em conta este tipo de informação. A desvantagem disso é que em regiões homogêneas, com predominância de uma única cor, estes descritores são incapazes de proporcionar ao MCL

meios de reduzir o erro da estimativa, como é o caso dos cinco mapas da região da UFRGS. Estes mapas possuem grandes regiões com predominância da cor verde e quando o VANT sobrevoa estas regiões, o erro da estimativa aumenta. Uma imagem da trajetória *Flight 3* que ilustra este problema é mostrada na Figura 4.13. Contudo, novamente, os resultados do uso do BRIEF, Figura 4.12-(b), são piores que os gerados pelo abBRIEF, Figura 4.12-(a). O descritor abBRIEF se mostra mais robusto às dificuldades do que o BRIEF, conseguindo reduzir o erro da estimativa de localização conforme o VANT se desloca para regiões não homogêneas, como é ilustrado pelos instantes finais, de 130 até 147, da Figura 4.12-(a).

Figura 4.12: Comparação dos resultados do descritor abBRIEF e do BRIEF sendo usados no modelo de observação. A estimativa de localização do vôo *Flight 3* com 50.000 partículas e 256 pares de pixels em cada descritor. (a) representa o resultado utilizando o descritor abBRIEF e (b) o resultado utilizando o BRIEF.



A Figura 4.14 ilustra o desvio padrão das 30 execuções de testes das três trajetórias utilizando o descritor abBRIEF no modelo de observação. Para estes

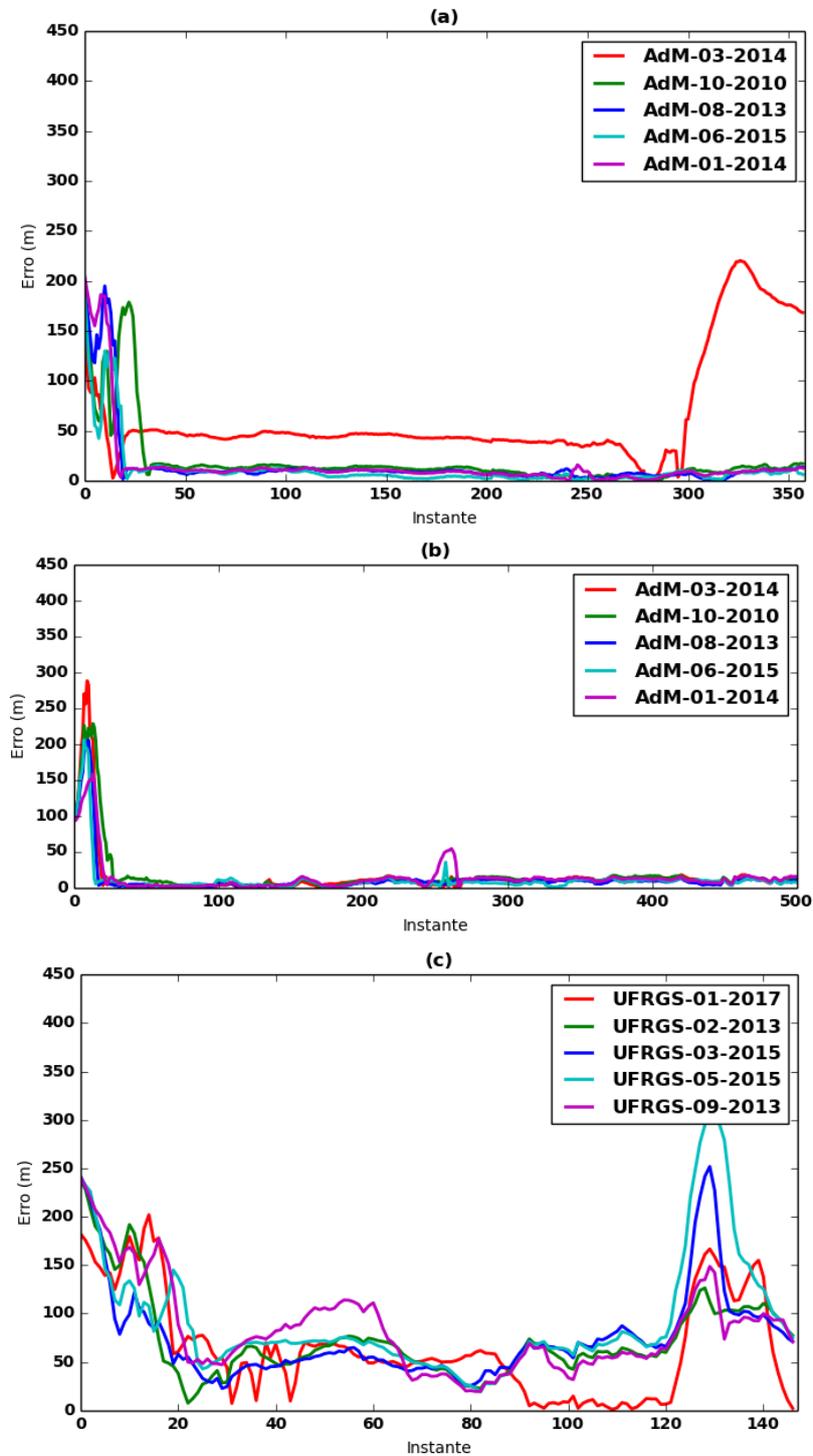
Figura 4.13: Imagem da trajetória *Flight 3* (esquerda) com sua respectiva imagem do mapa *UFRGS-05-2015* (direita). Ambas imagens mostram uma região uniforme com predominância da cor verde. Esta região causa um pico de aumento do erro ilustrado pela Figura 4.12-(a), próximo ao instante 130.



resultados, foram utilizadas 50.000 partículas e 256 pares de pixels no descritor. Para a Figura 4.14-(a) é possível observar que o mapa *AdM-03-2014* possui um erro maior que os demais devido ao erro da estimativa da localização no final da trajetória, como é ilustrado na Figura 4.7-(a). Esse erro aumenta a média geral da estimativa e por consequência, também aumenta o desvio padrão. A Figura 4.14-(b) apresenta um desvio padrão mais uniforme para todos os mapas, com um erro muito pequeno, enquanto que na Figura 4.14-(c) o desvio padrão é bem maior, já que o erro da estimativa de localização neste caso, Figura 4.12-(a), é alto.

Percebe-se que as modificações propostas no descritor BRIEF original permitiram obter resultados consideravelmente melhores quando utilizadas no modelo de observação para o MCL. O novo descritor, denominado abBRIEF, ao ser utilizado neste modelo, se mostrou robusto em relação as dificuldades do problema de localização e a variedade dos mapas, garantindo pequenos erros de estimativa de localização.

Figura 4.14: Comparação dos resultados de desvio padrão das três trajetórias utilizando o descritor abBRIEF no modelo de observação. Foram utilizadas 50.000 partículas e 256 pares de pixels no descritor. (a) representa o desvio padrão da trajetória *Flight 1*; (b) o desvio padrão para a *Flight 2*; e (c) o desvio padrão para a *Flight 3*.



## 4.6 Estimativa do ângulo

Nesta seção serão apresentadas análises de alguns outros pontos que não foram abordados até o momento em relação ao modelo de observação. Especificamente, serão analisados os resultados do ângulo estimado pelo MCL e a comparação entre a odometria visual e o *ground-truth*.

Todos os cálculos de erro dos ângulos seguiram um mesmo procedimento. As partículas foram distribuídas uniformemente considerando  $\mathbf{x} \in \mathbf{m}$ ,  $\theta \in [0, 2\pi)$  e  $z(\mathbf{x}) \in [80m, 1000m]$  para os mapas de Arroio do Meio e  $z(\mathbf{x}) \in [50m, 600m]$  para os mapas da UFRGS. Em cada instante  $t$  da trajetória, o cálculo da estimativa do ângulo do VANT é dado pelo ângulo médio,  $\bar{\theta}_t$ , sendo

$$\bar{\theta}_t = \text{atan2}\left(\frac{\sum_{i=1}^{\mathbf{P}} \sin(\theta(\mathbf{x}_t^{[i]})) * w(\mathbf{x}_t^{[i]})}{\sum_{i=1}^{\mathbf{P}} w(\mathbf{x}_t^{[i]})}, \frac{\sum_{i=1}^{\mathbf{P}} \cos(\theta(\mathbf{x}_t^{[i]})) * w(\mathbf{x}_t^{[i]})}{\sum_{i=1}^{\mathbf{P}} w(\mathbf{x}_t^{[i]})}\right), \quad (4.3)$$

em que  $\mathbf{P}$  é o número de partículas,  $w(\mathbf{x}_t^{[i]})$  é o peso e  $\theta(\mathbf{x}_t^{[i]})$  é o ângulo da  $i$ -ésima partícula no instante  $t$ .

Pela aleatoriedade do MCL e para obter um resultado estatisticamente significativo, cada trajetória foi testada 30 vezes. Então, para obter o erro do ângulo estimado,  $errYaw_t$ , ilustrado pela Figura 4.15, calculou-se a média da diferença entre os ângulos médios,  $\bar{\theta}_t$ , e o ângulo dado pelo sensor IMU,  $\theta_t^r$ , sendo

$$errYaw_t = \text{atan2}\left(\frac{\sum_{i=1}^{30} \sin(\theta(\mathbf{x}_t^{[i]}))}{30}, \frac{\sum_{i=1}^{30} \cos(\theta(\mathbf{x}_t^{[i]}))}{30}\right) - \theta_t^r. \quad (4.4)$$

A Figura 4.15 ilustra o erro do ângulo estimado pelo MCL durante cada trajetória. Na *Flight 1*, Figura 4.15-(a), percebe-se que o erro que acontece na estimativa da localização no mapa *AdM-03-2014* também acontece no ângulo. A justificativa para isso é a mesma que foi dada anteriormente, as regiões homogêneas do mapa acabam deslocando as partículas para posturas erradas e isso acaba aumentando o erro tanto de localização, quanto de orientação. A *Flight 2*, Figura 4.15-(b), apresenta picos de erros durante alguns momentos, isso acontece quando o VANT muda sua orientação e as partículas, por não estarem todas na postura exata, acabam tendo uma diferença de orientação momentânea. Por fim, a trajetória *Flight 3*, Figura 4.15-(c), apresenta os piores resultados, assim como aconteceu na estimativa de localização, Figura 4.12-(a). Como as partículas estavam em localizações erradas ou

distantes da correta, o MCL teve dificuldades para estimar uma orientação correta para as partículas, o que justifica o erro tão grande.

Por fim, a última análise que é feita aqui é a respeito da comparação entre a odometria visual, postura média e *ground-truth*. A Figura 4.16 ilustra estas comparações para cada uma das trajetórias. Importante observar que o resultado estimado é apresentado em tons de azul escuro após o filtro convergir, e em tons de azul claro antes disso, evitando que o processo inicial de convergência atrapalhe a compreensão das imagens. De maneira geral, percebe-se que a odometria visual possui um erro que aumenta conforme as trajetórias avançam. A odometria não é muito precisa, já que existe um erro em cada informação estimada, principalmente em rotações, e esse erro se acumula. Como o MCL pesa e descarta as partículas com pesos baixos, os pequenos erros de odometria abacam não acumulando, sendo compensados pelo processo de análise das partículas. Porém, nos momentos em que o método de odometria visual não consegue estimar a odometria, o filtro acaba espalhando as partículas para cobrir um espaço maior, aumentando a variedade das posturas e o erro da estimação da postura. Sobre a postura média das partículas, como já mostrado na Seção 4.5, o erro não é muito grande e, em grande parte das trajetórias, fica bem próxima do *ground-truth*.

A fim de demonstrar melhor o comportamento do MCL utilizando o modelo de observação proposto, foi gravado um vídeo da execução de cada trajetória, *Flight 1*<sup>5</sup>, *Flight 2*<sup>6</sup> e *Flight 3*<sup>7</sup>. Nestes vídeos, o MCL utilizou 50.000 partículas e o descritor abBRIEF foi construído usando 256 pares de pixels. Para as trajetórias *Flight 1* e *Flight 2* foi utilizado o mapa *AdM-08-2013*, enquanto que para a *Flight 3* foi utilizado o *UFRGS-01-2017*.

---

<sup>5</sup><https://goo.gl/pYHVh3>

<sup>6</sup><https://goo.gl/CCXYlf>

<sup>7</sup><https://goo.gl/Cf7nJx>

Figura 4.15: Erros dos ângulos estimados pelo MCL utilizando o modelo de observação com o descritor abBRIEF, 50.000 partículas e 256 pares de pixels no descritor. (a) *Flight 1*; (b) *Flight 2*; e (c) *Flight 3*.

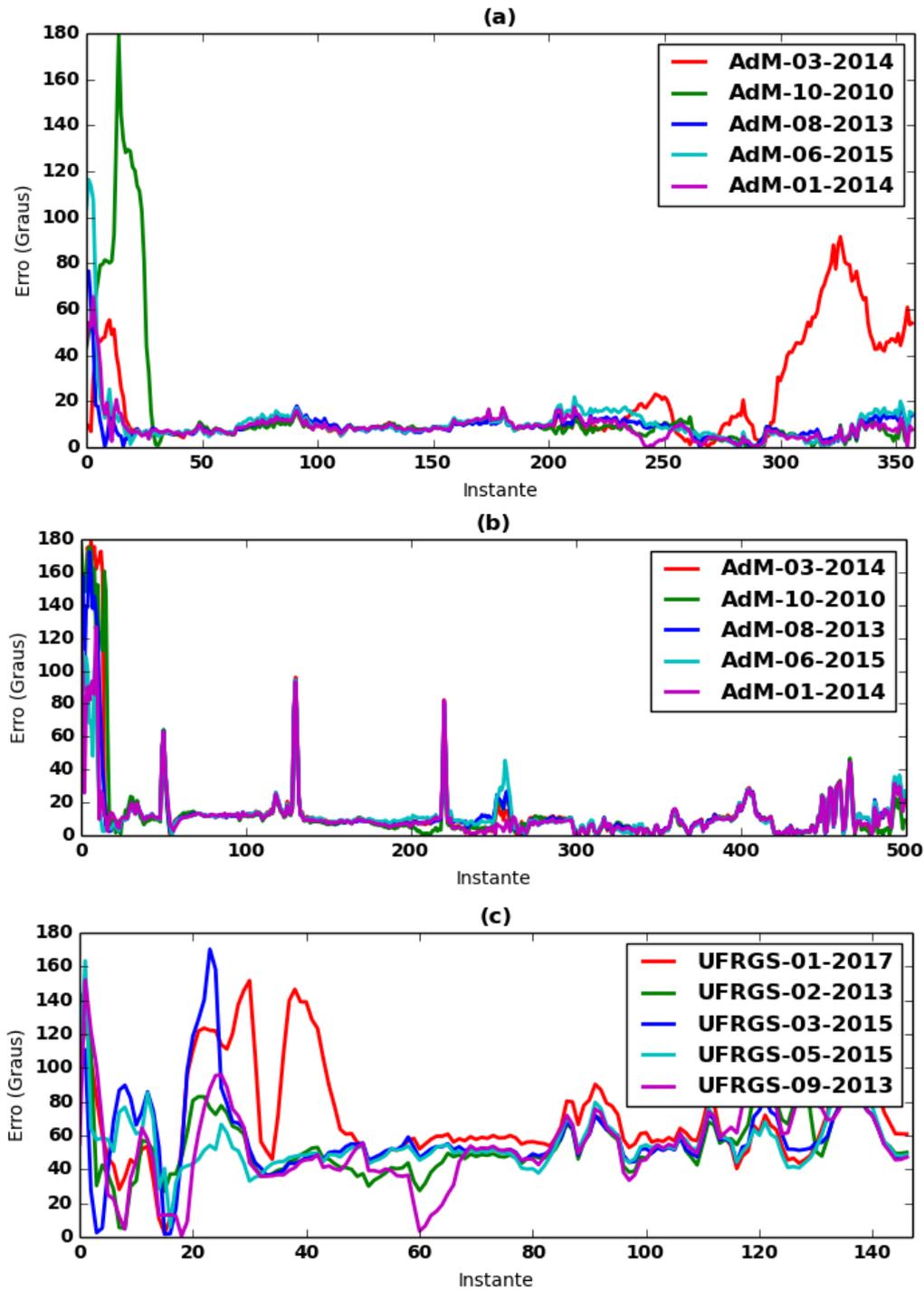
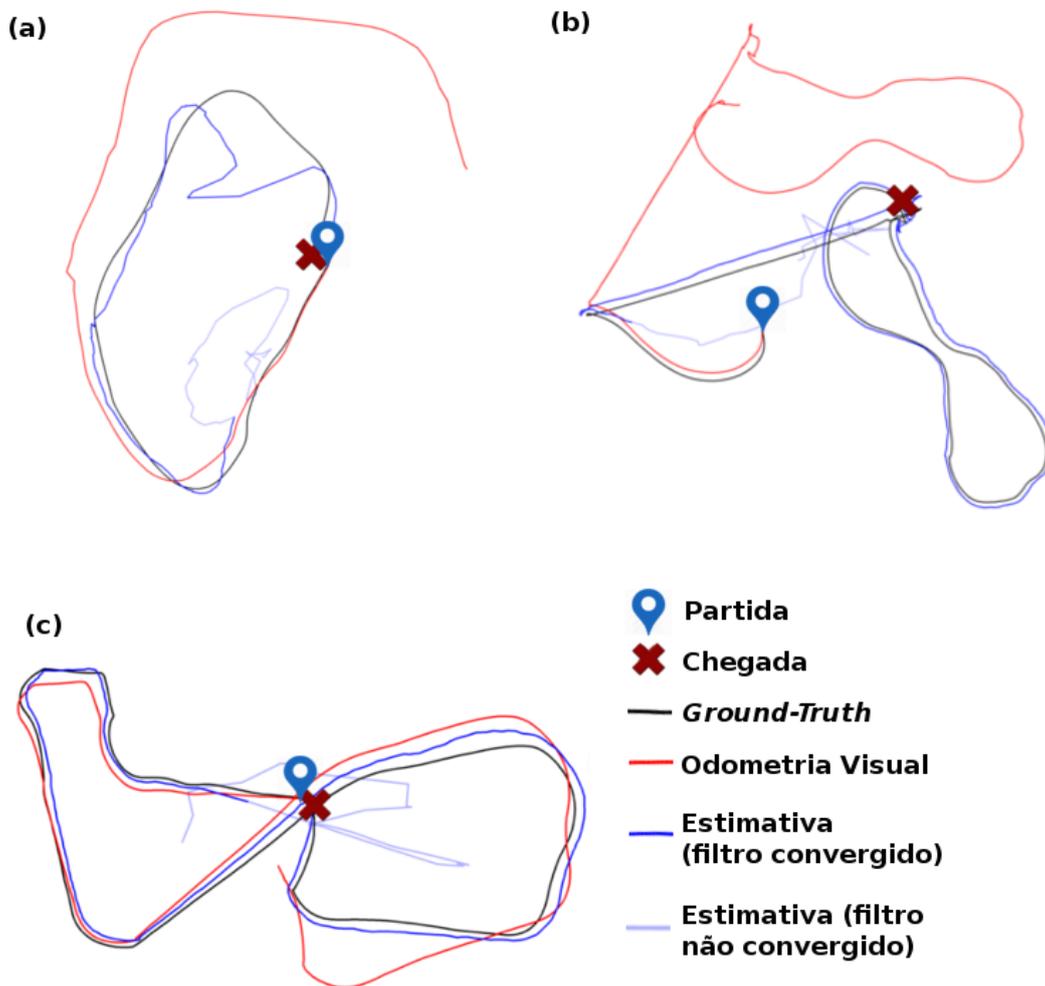


Figura 4.16: Demonstração qualitativa da diferença entre o *Ground-Truth*, odometria visual e localização estimada das trajetórias: (a) *Flight 3*; (b) *Flight 2*; e (c) *Flight 1*



## 5 CONCLUSÃO

Neste trabalho, foi proposto um modelo de observação que utiliza o, também proposto, descritor abBRIEF. Este modelo é usado pelo MCL para realizar a localização global em 4 DoF de um VANT em ambientes externos utilizando imagens de satélite. Este novo modelo de observação compara a imagem capturada pelo robô com as imagens das partículas considerando diferentes rotações e escalas. A proposta foi avaliada através de três trajetórias realizadas em dois locais diferentes e que possuem variação de altura. Para cada local, foram extraídos cinco mapas.

A criação do abBRIEF através da modificação do descritor BRIEF foi necessária para torná-lo mais robusto na avaliação da similaridade entre imagens, especialmente neste problema de localização. Seu uso no modelo de observação proporcionou bons resultados por conta das características que são muito adequadas, as quais podemos destacar:

- (i) A independência das *features* permite que o VANT sobrevoe qualquer região e utilize mapas que não possuem muitos detalhes, como os disponíveis no Google<sup>TM</sup> Earth;
- (ii) O uso de pares de pixels para gerar os descritores proporciona um baixo custo computacional e de tempo de processamento em comparação com métodos que necessitam extrair a imagem da partícula da imagem de satélite, permitindo a execução de um MCL visual neste cenário;
- (iii) Utilizar a informação de cor, através dos canais  $a^*$  e  $b^*$  do espaço de cor CIE  $L^*a^*b^*$  torna o modelo mais robusto, como foi observado na Seção 4.5 quando foi comparado ao BRIEF que utiliza apenas imagens em escala de cinza;
- (iv) Pontos aparentemente fracos do descritor, como ser variante a rotação e escala, são na verdade adequados para este contexto, que permitem ao modelo pesar corretamente apenas as partículas que encontram-se na orientação e escala corretas.

Inicialmente, testou-se uma abordagem clássica para este problema, tentando encontrar a correspondência entre a imagem do VANT e a imagem de satélite através do uso de *features* ORB. Se esta abordagem funcionasse, o MCL poderia ser dispensado, uma vez que seria possível determinar a postura do VANT no mapa. Porém, o método não obteve êxito em nenhuma das imagens do VANT testadas per-

tencentas a trajetória *Flight 1*, demonstrando não ser capaz de localizar o VANT. Isso se justifica pela grande diferença entre as imagens e pelo baixo nível de detalhes da imagem de satélite, que prejudica a extração e descrição das *features*.

Em seguida, diversos métodos foram comparados com o abBRIEF buscando determinar a capacidade de cada um em calcular a similaridade entre imagens. Observou-se que apesar do MI ter uma leve vantagem sobre o abBRIEF com relação ao número de acertos, seu custo computacional somado ao custo de extração da imagem da partícula da imagem de satélite o torna inadequado para o modelo de observação do MCL, justificando, então, o uso do abBRIEF para tal.

Também foi feito um estudo a respeito do comportamento do sistema de localização ao variar o número de partículas do MCL e a quantidade de pares de pixels para construir o descritor abBRIEF. Os resultados mostram que a combinação de 50.000 partículas e 256 pares é o suficiente para localizar o VANT. Apesar de 50.000 parecer um valor muito grande, é importante lembrar que o espaço de busca deste problema é muito extenso, sendo: as dimensões do mapa, as possíveis orientações do robô e suas possíveis alturas, ou seja,  $1050m \times 1050m \times 360^\circ \times 920m$  (com resolução de  $20cm$  e  $1^\circ$ , que totaliza, aproximadamente,  $45 \times 10^{12}$ ). O que viabiliza o MCL com 50.000 partículas é o baixo custo computacional do abBRIEF aliado com a possibilidade de computá-lo sem a necessidade de extrair a imagem da partícula, tendo um custo de tempo de menos de um segundo para processar as todas as partículas com uma imagem capturada pelo VANT, considerando que haviam 8 *threads* em paralelo para os passos de amostragem e pesagem das partículas no MCL. Este tempo foi alcançado em um computador de 8 GB de memória com um processador Intel Core i7, 3.60GHz.

Finalmente, considerando que o descritor abBRIEF foi proposto através de modificações no BRIEF, é importante justificá-las através de testes. As mudanças feitas foram: o abBRIEF usa imagens no espaço de cor CIE  $L^*a^*b^*$ , enquanto que o BRIEF considera em escala de cinza; o processo de suavização feito no BRIEF foi substituído por uma quantização das imagens no abBRIEF. Os resultados mostram que estas mudanças tornam o abBRIEF mais robusto do que o BRIEF principalmente pelo uso das informações de cores e ausência do canal de luminosidade, gerando resultados melhores no processo de localização.

De maneira geral, os resultados apresentados foram satisfatórios, sendo que o comportamento do ângulo estimado foi parecido com a posição, aumentando o

erro nas mesmas regiões. Isso significa que quando o método é capaz de determinar onde o VANT se encontra, ele também determina a orientação correta dele. Porém, quando falha em determinar a localização, acaba estimado orientações erradas. Observa-se também que a localização estimada mostrou-se muito superior ao uso de apenas odometria visual, que acumula erros ao longo da trajetória, principalmente em rotações, e acaba divergindo muito da posição correta do robô com o passar do tempo.

Em alguns poucos casos o erro da estimativa de localização aumentou devido a uma fraqueza do modelo de observação. O descritor abBRIEF baseia-se no fato de que não é possível observar os detalhes da imagem do VANT e do mapa para determinar sua similaridade, devido as diversas mudanças apresentadas entre elas, fazendo com que os detalhes de uma provavelmente não sejam encontradas na outra. O resultado disso, e a sua desvantagem, é que quando o robô sobrevoa regiões com cores homogêneas, como por exemplo florestas ou grandes campos de pastos, o modelo não consegue pesar as partículas de maneira que o filtro convirja, mantendo diversas hipóteses ao longo da região homogênea. Isso foi mostrado pela trajetória *Flight 3*, em que o VANT sobrevoa regiões com estas características. O robô parte de uma região homogênea e impede que o MCL convirja rapidamente, conforme ele segue sua trajetória, o erro sobre a estimativa da localização diminui, voltando a aumentar quando ele sobrevoa uma floresta homogênea. Ainda, é importante destacar que apesar das partículas se afastarem da postura correta do robô (o que aumenta o erro da estimativa de localização) quando ele sobrevoa a região de mata, assim que ele segue para fora desta, o filtro é capaz de convergir novamente, mostrando sua robustês.

Outra desvantagem é causada pelo uso apenas de imagens como fonte de informação, em que a altura do VANT estimada pelo método é relativa ao solo e não ao nível do mar. Assim, se o robô manter a altura, mas passar a sobrevoar uma região em que o solo se aproxime ou se distancie dele, o método estimará que ele desceu ou subiu, respectivamente. Isso não compromete o funcionamento do sistema, mas torna a altura uma informação não confiável para determinadas aplicações. Uma possível solução seria utilizar mapas que contenham informações sobre a altura do solo em relação ao nível do mar, podendo assim compensar a diferença.

Ao analisar os trabalhos que tratam o mesmo problema, e que já foram dis-

cutidos na Seção 2.5, as vantagens desta proposta são: não é necessário um processamento de aprendizado de máquina para que o sistema de localização funcione, como o trabalho de Masselli, Hanten e Zell (2016), o uso de pontos de referência colocados previamente no ambiente é dispensado, diferente do trabalho de Jayatilleke e Zhang (2013), o uso de imagens de satélite proporciona realizar a localização até mesmo em cidades pequenas que ainda não foram mapeadas por novas ferramentas como o Google<sup>TM</sup> Street View, o que não é possível na abordagem de Majdik et al. (2015) e, por fim, o uso de um mapa com dimensões de aproximadamente  $1.000m^2$  permite uma grande liberdade de vôo para o VANT, o que não acontece no caso dos mapas utilizados por Yol et al. (2014), que é de aproximadamente  $350m \times 400m$  e por Masselli, Hanten e Zell (2016), que é de  $150m \times 90m$ .

Como produto final, tem-se um sistema de localização rápido, que é capaz de localizar o VANT em diversos mapas diferentes do mesmo ambiente, é robusto a grande parte dos problemas e dificuldades existentes neste tipo de tarefa e exige apenas uma imagem de satélite do ambiente antes de sua operação.

## 5.1 Trabalhos Futuros

Apesar dos bons resultados apresentados e das vantagens em relação aos demais trabalhos semelhantes encontrados na literatura, ainda há possibilidades de melhora e expansão. Por enquanto são estimados 4 DoF, com os ângulos *roll* e *pitch* sendo ignorados. Uma nova versão deste sistema poderia incorporar estes ângulos, dispensando a necessidade do gimbal.

Vôos em baixas altitudes sofrem com o efeito de perspectiva na captura das imagens, tornando a diferença entre a imagem do VANT e a imagem de satélite ainda maior, como é o caso da trajetória *Flight 3* que inicia com aproximadamente  $50m$  de altura. Por isso, é válida uma investigação a respeito do comportamento do sistema em altitudes menores, para as aplicações que necessitem de tal configuração de vôo.

Este sistema localiza um VANT utilizando apenas uma imagem de satélite e as imagens que ele captura com sua câmera, sendo totalmente adequado para ser embarcado no robô. Por isso, pretende-se investigar o seu desempenho na prática ao ser embarcado em alguma plataforma como Jetson<sup>1</sup> ou similares.

Por fim, também é importante o estudo sobre localização de múltiplos robôs

---

<sup>1</sup><http://www.nvidia.com/object/jetson-tk1-embedded-dev-kit.html>

e como esta abordagem de localização baseado em visão computacional poderia ser ampliada para aplicações com exames de VANTs.

## REFERÊNCIAS

- BAY, H.; TUYTELAARS, T.; GOOL, L. V. Surf: Speeded up robust features. In: \_\_\_\_\_. **Computer Vision – ECCV 2006: 9th European Conference on Computer Vision, Graz, Austria, May 7-13, 2006. Proceedings, Part I.** Berlin, Heidelberg: Springer Berlin Heidelberg, 2006. p. 404–417. ISBN 978-3-540-33833-8. Available from Internet: <[http://dx.doi.org/10.1007/11744023\\_32](http://dx.doi.org/10.1007/11744023_32)>.
- BRIECHLE, K.; HANEBECK, U. D. Template matching using fast normalized cross correlation. In: . [s.n.], 2001. v. 4387, p. 95–102. Available from Internet: <<http://dx.doi.org/10.1117/12.421129>>.
- CABALLERO, F. et al. Improving vision-based planar motion estimation for unmanned aerial vehicles through online mosaicing. In: **Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.** [S.l.: s.n.], 2006. p. 2860–2865. ISSN 1050-4729.
- CALONDER, M. et al. Brief: Binary robust independent elementary features. In: \_\_\_\_\_. **Computer Vision – ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part IV.** Berlin, Heidelberg: Springer Berlin Heidelberg, 2010. p. 778–792. ISBN 978-3-642-15561-1. Available from Internet: <[http://dx.doi.org/10.1007/978-3-642-15561-1\\_56](http://dx.doi.org/10.1007/978-3-642-15561-1_56)>.
- CARROLL, J. V. Vulnerability assessment of the u.s. transportation infrastructure that relies on the global positioning system. **Journal of Navigation**, Cambridge University Press, Cambridge, UK, v. 56, n. 2, p. 185–193, 05 2003.
- CHEN, Y. et al. Hybrid aerial and aquatic locomotion in an at-scale robotic insect. In: **2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).** [S.l.: s.n.], 2015. p. 331–338.
- CHOSSET, H. et al. **Principles of Robot Motion: Theory, Algorithms, and Implementations.** Cambridge, MA: The MIT Press, 2005.
- CONTE, G.; DOHERTY, P. An integrated uav navigation system based on aerial image matching. In: **2008 IEEE Aerospace Conference.** [S.l.: s.n.], 2008. p. 1–10. ISSN 1095-323X.
- DAME, A.; MARCHAND, E. Second-order optimization of mutual information for real-time image registration. **IEEE Transactions on Image Processing**, v. 21, n. 9, p. 4190–4203, Sept 2012. ISSN 1057-7149.
- DELLAERT, F. et al. Monte carlo localization for mobile robots. In: **Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No.99CH36288C).** [S.l.: s.n.], 1999. v. 2, p. 1322–1328 vol.2. ISSN 1050-4729.
- FISCHLER, M. A.; BOLLES, R. C. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. **Commun. ACM**, ACM, New York, NY, USA, v. 24, n. 6, p. 381–395, jun. 1981. ISSN 0001-0782. Available from Internet: <<http://doi.acm.org/10.1145/358669.358692>>.

FORSTER, C.; PIZZOLI, M.; SCARAMUZZA, D. Svo: Fast semi-direct monocular visual odometry. In: **2014 IEEE International Conference on Robotics and Automation (ICRA)**. [S.l.: s.n.], 2014. p. 15–22. ISSN 1050-4729.

HAMMING, R. W. Error detecting and error correcting codes. **Bell System Technical Journal**, Blackwell Publishing Ltd, v. 29, n. 2, p. 147–160, 1950. ISSN 1538-7305. Available from Internet: <<http://dx.doi.org/10.1002/j.1538-7305.1950.tb00463.x>>.

HOWARD, A. Real-time stereo visual odometry for autonomous ground vehicles. In: **2008 IEEE/RSJ International Conference on Intelligent Robots and Systems**. [S.l.: s.n.], 2008. p. 3946–3952. ISSN 2153-0858.

HUA, G.; BROWN, M.; WINDER, S. Discriminant embedding for local image descriptors. In: **2007 IEEE 11th International Conference on Computer Vision**. [S.l.: s.n.], 2007. p. 1–8. ISSN 1550-5499.

JAYATILLEKE, L.; ZHANG, N. Landmark-based localization for unmanned aerial vehicles. In: **2013 IEEE International Systems Conference (SysCon)**. [S.l.: s.n.], 2013. p. 448–451.

KONOMURA, R.; HORI, K. Visual 3d self localization with 8 gram circuit board for very compact and fully autonomous unmanned aerial vehicles. In: **2014 IEEE International Conference on Robotics and Automation (ICRA)**. [S.l.: s.n.], 2014. p. 5215–5220. ISSN 1050-4729.

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.

LOWE, D. G. Distinctive image features from scale-invariant keypoints. **International Journal of Computer Vision**, v. 60, n. 2, p. 91–110, 2004. ISSN 1573-1405. Available from Internet: <<http://dx.doi.org/10.1023/B:VISI.0000029664.99615.94>>.

LOWRY, S.; MILFORD, M. J. Building beliefs: Unsupervised generation of observation likelihoods for probabilistic localization in changing environments. In: **2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)**. [S.l.: s.n.], 2015. p. 3071–3078.

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. ISSN 1556-4967. Available from Internet: <<http://dx.doi.org/10.1002/rob.21585>>.

MAKARENKO, A. A. et al. An experiment in integrated exploration. In: **IEEE/RSJ International Conference on Intelligent Robots and Systems**. [S.l.: s.n.], 2002. v. 1, p. 534–539 vol.1.

MASSELLI, A.; HANTEN, R.; ZELL, A. Localization of unmanned aerial vehicles using terrain classification from aerial images. In: \_\_\_\_\_. **Intelligent Autonomous Systems 13: Proceedings of the 13th International Conference IAS-13**.

Cham: Springer International Publishing, 2016. p. 831–842. ISBN 978-3-319-08338-4. Available from Internet: <[http://dx.doi.org/10.1007/978-3-319-08338-4\\_60](http://dx.doi.org/10.1007/978-3-319-08338-4_60)>.

MEHTRE, B. M. et al. Color matching for image retrieval. **Pattern Recognition Letters**, v. 16, n. 3, p. 325 – 331, 1995. ISSN 0167-8655. Available from Internet: <<http://www.sciencedirect.com/science/article/pii/016786559400096L>>.

MERINO, L. et al. Vision-based multi-uav position estimation. **IEEE Robotics & Automation Magazine**, v. 13, n. 3, p. 53 – 62, 2006.

MIKOLAJCZYK, K.; SCHMID, C. A performance evaluation of local descriptors. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 27, n. 10, p. 1615–1630, Oct 2005. ISSN 0162-8828.

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

OLLERO, A.; MERINO, L. Control and perception techniques for aerial robotics. **Annual Reviews in Control**, v. 28, n. 2, p. 167 – 178, 2004. ISSN 1367-5788. Available from Internet: <<http://www.sciencedirect.com/science/article/pii/S1367578804000367>>.

RANGO, A. et al. Research article: Using unmanned aerial vehicles for rangelands: Current applications and future potentials. **Environmental Practice**, Cambridge University Press, New York, USA, v. 8, n. 3, p. 159–168, 09 2006.

ROMERO, R. et al. **Robótica Móvel**. 1st. ed. Rio de Janeiro, RJ: LTC, 2014.

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

SCARAMUZZA, D.; SIEGWART, R. Appearance-guided monocular omnidirectional visual odometry for outdoor ground vehicles. **IEEE Transactions on Robotics**, v. 24, n. 5, p. 1015–1026, Oct 2008. ISSN 1552-3098.

SENLET, T.; ELGAMMAL, A. Satellite image based precise robot localization on sidewalks. In: **2012 IEEE International Conference on Robotics and Automation**. [S.l.: s.n.], 2012. p. 2647–2653. ISSN 1050-4729.

SIDDALL, R.; KOVAC, M. Launching the aquamav: bioinspired design for aerial–aquatic robotic platforms. **Bioinspiration & Biomimetics**, v. 9, n. 3, p. 031001, 2014.

SIM, D.-G. et al. Integrated position estimation using aerial image sequences. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 24, n. 1, p. 1–18, Jan 2002. ISSN 0162-8828.

STACHNISS, C. **Exploration and mapping with mobile robots**. Thesis (PhD) — University of Freiburg, 2006.

SZELISKI, R. **Computer vision: algorithms and applications**. [S.l.]: Springer Science & Business Media, 2010.

TAYLOR, C. et al. **Robotic vacuum with localized cleaning algorithm**. Google Patents, 2004. US Patent App. 10/798,231. Available from Internet: <<https://www.google.com/patents/US20040204792>>.

TEAM, Q. D. **QGIS Geographic Information System**. [S.l.], 2009. Available from Internet: <<http://qgis.osgeo.org>>.

THRUN, S.; BURGARD, W.; FOX, D. **Probabilistic Robotics**. 1st. ed. Cambridge, MA: The MIT Press, 2005.

VIOLA, P.; III, W. M. W. Alignment by maximization of mutual information. **International Journal of Computer Vision**, v. 24, n. 2, p. 137–154, 1997. ISSN 1573-1405. Available from Internet: <<http://dx.doi.org/10.1023/A:1007958904918>>.

VISWANATHAN, A.; PIRES, B. R.; HUBER, D. Vision-based robot localization across seasons and in remote locations. In: **2016 IEEE International Conference on Robotics and Automation (ICRA)**. [S.l.: s.n.], 2016. p. 4815–4821.

WHELAN, T. et al. Robust real-time visual odometry for dense rgb-d mapping. In: **2013 IEEE International Conference on Robotics and Automation**. [S.l.: s.n.], 2013. p. 5724–5731. ISSN 1050-4729.

YANG, X. et al. Survey on the novel hybrid aquatic–aerial amphibious aircraft: Aquatic unmanned aerial vehicle (aquauav). **Progress in Aerospace Sciences**, v. 74, p. 131 – 151, 2015. ISSN 0376-0421. Available from Internet: <[//www.sciencedirect.com/science/article/pii/S0376042114001122](http://www.sciencedirect.com/science/article/pii/S0376042114001122)>.

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. ISSN 2153-0858.

ZHENG, M. et al. Rotation and affine-invariant sift descriptor for matching uav images with satellite images. In: **Proceedings of 2014 IEEE Chinese Guidance, Navigation and Control Conference**. [S.l.: s.n.], 2014. p. 2624–2628.

ZHOU, X.; BOULANGER, P. Image matting based on mutual information. In: **2011 Visual Communications and Image Processing (VCIP)**. [S.l.: s.n.], 2011. p. 1–4.

ZITOVÁ, B.; FLUSSER, J. Image registration methods: a survey. **Image and Vision Computing**, v. 21, n. 11, p. 977 – 1000, 2003. ISSN 0262-8856. Available from Internet: <<http://www.sciencedirect.com/science/article/pii/S0262885603001379>>.

## APÊNDICE A — TEORIA DA PROBABILIDADE

### A.1 Probabilidade de uma variável aleatória

Seja  $X$  uma variável aleatória e  $x$  corresponde a um valor específico que  $X$  pode assumir. No campo da robótica probabilística, o modelo de variável aleatória que quantifica o dado, ou seja, a leitura de um sensor, a pose de um robô ou os controles de movimento. Se um conjunto de valores que  $X$  pode assumir é discreto, então

$$p(X = x) \tag{A.1}$$

expressa a probabilidade que a variável aleatória  $X$  tem um valor  $x$ . Para simplificar a notação, nesta dissertação será usado a abreviação  $p(x)$ .

### A.2 Regra do Produto

A Regra do Produto pode ser descrita como

$$p(x, y) = p(x | y) \cdot p(y), \tag{A.2}$$

$$p(y, x) = p(y | x) \cdot p(x). \tag{A.3}$$

### A.3 Probabilidade Condicional

A probabilidade de  $x$ , conhecendo o resultado de  $y$ , é conhecido como a probabilidade condicional. É dado por

$$p(x | y) = \frac{p(x, y)}{p(y)}, \tag{A.4}$$

$$p(y | x) = \frac{p(x, y)}{p(x)}, \tag{A.5}$$

com  $p(x) > 0$  e  $p(y) > 0$ .

## A.4 Independência

Se a ocorrência de  $x$  não influencia a probabilidade de  $y$ , então eles são ditos independentes, e

$$p(x, y) = p(x | y) \cdot p(y) = p(x) \cdot p(y). \quad (\text{A.6})$$

Se  $x$  e  $y$  são independentes em uma probabilidade condicionada, então

$$p(x | y) = p(x), \quad (\text{A.7})$$

$$p(y | x) = p(y). \quad (\text{A.8})$$

## A.5 Regra de Bayes

A Regra de Bayes caracteriza a relação entre uma probabilidade condicional,  $p(x | y)$ , para com sua inversa,  $p(y | x)$ . Sua fórmula é dada por

$$p(x | y) = \frac{p(x, y)}{p(y)} = \frac{p(y | x) \cdot p(x)}{p(y)}. \quad (\text{A.9})$$

$p(y)$  é independente de  $x$  na Regra de Bayes, e isso implica que  $p(y)^{-1}$ , para qualquer valor que  $x$  assumira, será o mesmo. Por isso,  $p(y)^{-1}$  é frequente se referir a ele como uma variável de normalização da Regra de Bayes, e é denotado como  $\eta$ . Então,

$$p(x | y) = \eta \cdot p(y | x) \cdot p(x). \quad (\text{A.10})$$

Esta notação é usada para simplificar e expressar que o resultado final deve ser normalizado.

### A.5.1 Probabilidade *a priori*

É a distribuição de probabilidade que expressa uma crença sem nenhuma evidência sobre um evento. É uma maneira de expressar um resultado em termos de incerteza ao invés da aleatoriedade. Para o exemplo da Regra de Bayes, se  $x$  é a quantidade que deve ser inferida a partir de  $y$ , a probabilidade  $p(x)$  é dita distribuição de probabilidade *a priori*, e  $y$  é o dado utilizado (por exemplo, a leitura

de um sensor). Então, a probabilidade  $p(x)$  resume o conhecimento a respeito de  $X$  antes de incorporar o dado  $y$ .

### A.5.2 Probabilidade *a posteriori*

É a probabilidade condicional que expressa uma crença levando em conta uma evidência relevante sobre um evento. Considerando a Regra de Bayes, a probabilidade  $p(x | y)$  é chamada de distribuição de probabilidade *a posteriori* sobre  $X$ . Esta regra permite computar a distribuição *a posteriori*  $p(x | y)$  usando a probabilidade condicional inversa,  $p(y | x)$ , junto com a probabilidade *a priori*,  $p(x)$ .

## A.6 Lei da Probabilidade Total

A Lei de Probabilidade Total expressa a relação entre uma probabilidade marginal e uma probabilidade condicional. Como sugere seu nome, expressa a probabilidade total do resultado desta relação

$$p(x) = \int_y p(x | y) \cdot p(y) dy, \quad (\text{A.11})$$

e a correspondente para o caso discreto

$$p(x) = \sum_y p(x | y) \cdot p(y). \quad (\text{A.12})$$

## A.7 Suposição de Markov

A Suposição de Markov (também conhecida como propriedade de Markov) corresponde a característica de falta de memória em processos estocásticos. Em outras palavras, se uma variável aleatória  $x_t$  de um processo estocástico depende apenas de seu pré-decessor direto,  $x_{t-1}$ , ao invés de todo histórico de eventos, então este processo tem a suposição de Markov,

$$p(x_t | x_{1:t-1}) = p(x_t | x_{t-1}). \quad (\text{A.13})$$

## APÊNDICE B — IMAGENS

## B.1 Arroio do Meio-RS

Figura B.1: Mapas de Arroio do Meio-RS.

(a) *AdM-08-2013.*(b) *AdM-10-2010.*(c) *AdM-03-2014.*(d) *AdM-06-2015.*(e) *AdM-01-2014.*

## B.2 Universidade Federal do Rio Grande do Sul, Porto Alegre-RS

Figura B.2: Mapas da Universidade Federal do Rio Grande do Sul, Porto Alegre-RS.



(a) *UFRGS-01-2017.*



(b) *UFRGS-02-2013.*



(c) *UFRGS-03-2015.*



(d) *UFRGS-05-2015.*



(e) *UFRGS-09-2013.*