MINISTÉRIO DA CIÊNCIA, TECNOLOGIA, INOVAÇÕES E COMUNICAÇÕES
INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS

sid.inpe.br/mtc-m21c/2018/05.18.16.04-TDI

NAVEGAÇÃO AUTÔNOMA DE VANT POR IMAGENS LIDAR

José Renato Garcia Braga

Tese de Doutorado do Curso de Pós-Graduação em Computação Aplicada, orientada pelos Drs. Haroldo Fraga de Campos Velho, e Elcio Hideiti Shiguemori, aprovada em 15 de maio de 2018.

URL do documento original:
<http://urlib.net/8JMKD3MGP3W34R/3R5MGG8>

INPE
São José dos Campos
2018

PUBLICADO POR:

Instituto Nacional de Pesquisas Espaciais - INPE
Gabinete do Diretor (GBDIR)
Serviço de Informação e Documentação (SESID)
Caixa Postal 515 - CEP 12.245-970
São José dos Campos - SP - Brasil
Tel.:(012) 3208-6923/6921
E-mail: pubtc@inpe.br

COMISSÃO DO CONSELHO DE EDITORAÇÃO E PRESERVAÇÃO DA PRODUÇÃO INTELECTUAL DO INPE (DE/DIR-544):
Presidente:
Maria do Carmo de Andrade Nono - Conselho de Pós-Graduação (CPG)
Membros:
Dr. Plínio Carlos Alvalá - Centro de Ciência do Sistema Terrestre (COCST)
Dr. André de Castro Milone - Coordenação-Geral de Ciências Espaciais e Atmosféricas (CGCEA)
Dra. Carina de Barros Melo - Coordenação de Laboratórios Associados (COCTE)
Dr. Evandro Marconi Rocco - Coordenação-Geral de Engenharia e Tecnologia Espacial (CGETE)
Dr. Hermann Johann Heinrich Kux - Coordenação-Geral de Observação da Terra (CGOBT)
Dr. Marley Cavalcante de Lima Moscati - Centro de Previsão de Tempo e Estudos Climáticos (CGCPT)
Silvia Castro Marcelino - Serviço de Informação e Documentação (SESID)
BIBLIOTECA DIGITAL:
Dr. Gerald Jean Francis Banon
Clayton Martins Pereira - Serviço de Informação e Documentação (SESID)
REVISÃO E NORMALIZAÇÃO DOCUMENTÁRIA:
Simone Angélica Del Ducca Barbedo - Serviço de Informação e Documentação (SESID)
Yolanda Ribeiro da Silva Souza - Serviço de Informação e Documentação (SESID)
EDITORAÇÃO ELETRÔNICA:
Marcelo de Castro Pazos - Serviço de Informação e Documentação (SESID)
André Luis Dias Fernandes - Serviço de Informação e Documentação (SESID)

MINISTÉRIO DA CIÊNCIA, TECNOLOGIA, INOVAÇÕES E COMUNICAÇÕES
INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS

sid.inpe.br/mtc-m21c/2018/05.18.16.04-TDI

NAVEGAÇÃO AUTÔNOMA DE VANT POR IMAGENS LIDAR

José Renato Garcia Braga

Tese de Doutorado do Curso de Pós-Graduação em Computação Aplicada, orientada pelos Drs. Haroldo Fraga de Campos Velho, e Elcio Hideiti Shiguemori, aprovada em 15 de maio de 2018.

URL do documento original:
<http://urlib.net/8JMKD3MGP3W34R/3R5MGG8>

INPE
São José dos Campos
2018

Dados Internacionais de Catalogação na Publicação (CIP)

Braga, José Renato Garcia.

Br73n Navegação Autônoma de VANT por Imagens LiDAR / José Renato Garcia Braga. – São José dos Campos : INPE, 2018.
xxx + 274 p. ; (sid.inpe.br/mtc-m21c/2018/05.18.16.04-TDI)

Tese (Doutorado em Computação Aplicada) – Instituto Nacional de Pesquisas Espaciais, São José dos Campos, 2018.
Orientadores : Drs. Haroldo Fraga de Campos Velho, e Elcio Hideiti Shiguemori.

1. VANT. 2. Navegação autônoma. 3. Visão computacional. 4. Sensores LiDAR. 5. Fusão de dados. I.Título.

CDU 004.455.1:629.7

Esta obra foi licenciada sob uma Licença Creative Commons Atribuição-NãoComercial 3.0 Não Adaptada.

This work is licensed under a Creative Commons Attribution-NonCommercial 3.0 Unported License.

Aluno (a): *José Renato Garcia Braga*

Título: "NAVEGAÇÃO AUTÔNOMA DE VANT POR IMAGENS LiDAR"

Aprovado (a) pela Banca Examinadora em cumprimento ao requisito exigido para obtenção do Título de *Doutor(a)* em *Computação Aplicada*

Dr. Nandamudi Lankalapalli Vijaykumar

Presidente / INPE / SJCampos - SP

( ) Participação por Video - Conferência

Dr. Haroldo Fraga de Campos Velho

Orientador(a) / INPE / São José dos Campos - SP

( ) Participação por Video - Conferência

Dr. Elcio Hideiti Shiguemori

Orientador(a) / IEAv / São José dos Campos - SP

( ) Participação por Video - Conferência

Dr. Valdivino Alexandre de Santiago Júnior

Membro da Banca / INPE / São José dos Campos - SP

( ) Participação por Video - Conferência

Este trabalho foi aprovado por:

( ) maioria simples

(X) unanimidade

São José dos Campos, 15 de maio de 2018

Aprovado (a) pela Banca Examinadora em cumprimento ao requisito exigido para obtenção do Título de *Doutor(a)* em

Computação Aplicada

Dr. Lamartine Nogueira Frutuoso Guimarães

Membro da Banca / INPE / SJCampos - SP

( ) Participação por Video - Conferência

Dr. Alexandre Carlos Brandão Ramos

Convidado(a) / UNIFEI / Itajubá - MG

( ) Participação por Video - Conferência

Dr. Antonio de Padua Braga

Convidado(a) / UFMG / Belo Horizonte - MG

( ) Participação por Video - Conferência

Este trabalho foi aprovado por:

( ) maioria simples

(X) unanimidade

São José dos Campos, 15 de maio de 2018

"Quem nunca errou nunca experimentou nada novo".

Albert Einstein

A minha mãe Rita que sempre primou pela minha educação, exemplo de mãe e profissional e uma grande incentivadora do meu trabalho.

A minha noiva e futura esposa Wanessa por todo amor, apoio, companheirismo e incentivo na realização dos meus sonhos.

AGRADECIMENTOS

Primeiramente agradeço a Deus por mais essa conquista alcançada e por me dar inspiração e sabedoria para todos os momentos da minha vida pessoal, acadêmica e profissional.

Agradeço aos meus orientadores Dr. Haroldo Fraga de Campos Velho e Dr. Elcio Hideiti Shiguemori, pela amizade, pela disponibilidade, pela ajuda e pelas observações pertinentes e recomendações valiosas, que muito ajudaram no processo de desenvolvimento desta tese.

Aos professores Dr. Patrick Doherty e Dr. Gianpaolo Conte, pela disposição e orientação no meu doutorado sanduíche na Linköping University (LIU), na Suécia. Ambos Dr. Doherty e Dr. Conte, contribuíram muito nesta pesquisa e o tempo que passei na LIU foi de fundamental importância para este trabalho.

Ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) e a Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), pelos auxílios financeiros concedidos, os quais contribuíram para a viabilização desta pesquisa.

Agradeço pela participação de todos os membros da banca examinadora da defesa final dessa tese.

Aos meus familiares em especial a minha mãe Rita Garcia Gomes, pela dedicação, por entender minhas ausências e por estar sempre torcendo pelas minhas conquistas e ao meu irmão Pedro Luiz Garcia Braga, pela amizade e companheirismo de sempre.

A minha noiva Wanessa por todo carinho, ajuda e incentivo incondicional. Aos meus sogros, Beatriz Maria Silva e Pedro Afonso da Silva, por todo apoio sincero, torcida e confiança que depositaram em mim.

Por fim, agradeço a todos aqueles que colaboraram direta ou indiretamente para a realização deste trabalho.

RESUMO

Nas últimas décadas, observou-se um aumento gradativo na utilização de Veículo Aéreo Não Tripulado (VANT) em diversas áreas, entre elas, monitoramento do uso da terra, operações de busca e resgate, acompanhamento de impactos ambientais, vigilância de fronteiras, entre outras. Há expectativas no aumento do emprego de VANT em aplicações civis e militares devido ao baixo custo de desenvolvimento e operacional, quando comparados com aeronaves tripuladas. A ausência de tripulação embarcada durante o emprego de VANT, principalmente em missões de grande periculosidade, também é uma vantagem, pois diminui os riscos à vida humana. Com o aumento do emprego de VANT o desenvolvimento de sistemas para a sua navegação autônoma tem sido tema de diversas pesquisas. Uma das tarefas dos sistemas de navegação autônoma de VANT é estimar sua posição. A grande maioria dos sistemas de navegação autônoma de VANT utilizam a fusão de dados do Sistema de Navegação Inercial *(INS, do inglês, Inertial Navigation System)* com os dados do Sistema de Navegação Global por Satélite *(Global Navigation Satellite System)* (principalmente o Sistema de Posicionamento Global *(GPS, do inglês, Global Positioning System)*) para estimar a posição do VANT. Mas o GNSS pode sofrer interferência em seu sinal devido a ataques maliciosos e a fenômenos naturais. O desenvolvimento de um sistema de visão computacional, que utiliza imagens capturadas e processadas em tempo de voo para determinar a localização do VANT, pode substituir a informação do GNSS. Entretanto, ainda é um desafio o desenvolvimento de um sistema de estimação da posição do VANT por processamento de imagens para operar durante a navegação da aeronave sobre regiões cobertas por água (por exemplo, o oceano) e em condições de baixa ou sem luminosidade. Neste contexto, esta tese de doutorado apresenta uma metodologia que estima a posição geográfica de um VANT quando este sobrevoa regiões cobertas por água e regiões com baixa ou sem luminosidade utilizando imagens provenientes de um sensor ativo denominado *Light Detection And Ranging* (LiDAR). A metodologia proposta utiliza a fusão de dados, por um Filtro de Partículas Não-Extensível (FPNE), de duas técnicas de visão computacional, odometria visual e registro automático de imagens, para estimar a posição da aeronave. O sistema de estimação da posição requer que o processamento seja feito em sistemas embarcados e em tempo de voo, por este motivo, a metodologia proposta é desenvolvida em computadores portáteis e de baixo consumo e que permitam a computação de alto desempenho. Para a validação da metodologia proposta são realizados testes com dois conjunto de dados diferentes: o primeiro com dados reais do voo de VANT, mas capturado por uma câmera de vídeo; o segundo, uma simulação utilizando imagens de um sensor LiDAR. Os resultados obtidos com as técnicas empregadas mostram-se promissores para emprego na navegação autônoma de VANT em regiões cobertas por água ou e em regiões com baixa ou sem luminosidade.

Palavras-chave: VANT. Navegação autônoma. Visão computacional. Sensores LiDAR. Fusão de dados.

UAV AUTONOMOUS NAVIGATION BY LIDAR IMAGES PROCESSING.

ABSTRACT

In the last years, there was a gradual increase in the employment of Unmanned Aerial Vehicles (UAV) in several areas, among them, land-use monitoring, search and rescue operations, monitoring of environmental impacts, border surveillance, and others. There is an expectation that the use of UAVs will increase in the future, due to the low costs of development and low operational costs when compared to manned aircraft. The main advantage of the UAV employment is the absence of onboard crew, which reduces the risks to human life. With the increase of UAV employment, the development of systems for its autonomous navigation has been subject of several investigations. One of the tasks of autonomous navigation systems for UAV is to estimate its position. The main strategy applied for estimating the UAV position is the use of information from Inertial Navigation System (INS) combined with information from Global Navigation Satellite System (GNSS).The signal of GNSS can suffer an outage due to malicious attacks and natural phenomena. Thus, the development of a computer vision system, which uses images captured and processed in flight time, can be used to determine the UAV location and replace the information from GNSS. However, the development of a computer vision system for estimating the UAV position in a situation of flight over water-covered areas (eg the ocean) and flight in low light conditions is a challenge. In this context, this doctoral thesis presents an approach for estimating the geographical position of a UAV, when it flies over regions covered by water and regions with low light conditions. The approach uses images from an active sensor called Light Detection And Ranging (LiDAR) to allow the flight in those conditions. The proposed approach estimates the aircraft position by employing data fusion of two techniques of position estimation by computer vision: visual odometry and the image matching. The data fusion algorithm is performed by an Non-Extensible Particle Filter (NEPF). The approach requires that the processing be done in embedded systems and in flight time, thus, the proposed method is developed in portable and low-power high-performance computer. For the validation of the proposed method, two different data sets are applied: the first one composed by real flight data from an UAV, with images captured by its video camera; and the second one, a simulation using images from a LiDAR sensor. The approach results are promising for the UAV position estimation in regions covered by water or in regions with low light conditions.

Keywords: UAV. Autonomous Navigation. Computer Vision. LiDAR Sensors. Data Fusion.

LISTA DE FIGURAS

Pág.

3.1 Abordagem proposta para a estimação da posição de VANT por fusão de dados de técnicas de visão computacional. 23
3.2 Algoritmo de odometria visual monocular aplicado para estimar a posição do VANT. 25
3.3 Registro automático de entre uma imagem capturada por VANT e uma imagem georreferenciada. 27
3.4 Algoritmo aplicado para realizar a estimação da posição do VANT por registro automático de imagens. 28
3.5 Computador Raspberry PI Model B-1. 31
3.6 Conjunto de pinos GPIO para a conexão de periféricos. 32
3.7 Placa LOGI PI, que possui a FPGA Spartan 6 LX9. 33
3.8 Comunicação entre CPU e FPGA no computador Raspberry PI Model B-1. 33
3.9 Placa Zybo Zynq 7000. SoC com processador ARM Cortex-A9 e FPGA Artix-7. 34
3.10 Comunicação entre o processador ARM e a FPGA no computador Zybo por meio do barramento AXI 36

4.1 Esboço da câmera estenopeica 38
4.2 Modelo de projeção perspectiva da câmera estenopeica. 38
4.3 Transformações necessárias para mapear um ponto do objeto, no sistema de coordenadas do mundo real, para um ponto na imagem, no sistema de coordenadas em pixel. 39

5.1 Funcionamento de um sensor LiDAR aerotransportado. 50
5.2 Mapeamento batimétrico realizado por um sensor LiDAR. 53

6.1 Método de registro automático de imagens aplicado para estimar o posicionamento geográfico do VANT. 57
6.2 Etapas da aplicação do filtro da Mediana. 64
6.3 (A) - Imagem de intensidade LiDAR sem aplicação do filtro da Mediana. (B) - Imagem de intensidade LiDAR com aplicação do filtro da Mediana. 64
6.4 Etapas da aplicação do filtro Gaussiano. 65
6.5 (A) - Imagem de intensidade LiDAR sem aplicação do filtro Gaussiano. (B) - Imagem de intensidade LiDAR com aplicação do filtro Gaussiano. . 66

6.6 (A) - Imagem formada por uma transição de uma região escura para clara e retornando para escura novamente. (B) - Imagem formada por uma transição de uma região clara para escura e retornando para escura novamente. 68
6.7 Neurônio artificial. 73
6.8 Funções de ativação. 74
6.9 Arquitetura de uma rede neural artificial MLP. 76
6.10 Padrões de borda e não-borda utilizados para o treinamento da MLP . . 82
6.11 Transformação de um padrão de treinamento (matriz) em vetor. 83
6.12 Ativação da MLP. 83
6.13 Consulta à LUT implementada na FPGA do computador Raspberry PI . 86
6.14 Consulta à LUT implementada na FPGA do computador Zybo ZINQ 7000 87
6.15 Matriz resultado da correlação cruzada entre duas imagens. O ponto marcado pelo círculo em vermelho é o maior valor da matriz de correlação, a posição ocupada pelo VANT 89
6.16 Correlação cruzada entre imagem aérea, capturada pelo sensor imageador do VANT, e uma imagem georreferenciada da região sobrevoada. Ambas as imagens estão com as bordas detectadas. 90
6.17 Imagem georreferenciada do conjunto de dados do primeiro experimento. 95
6.18 Helicóptero autônomo RMAX. 96
6.19 Imagem capturada pela câmera do helicóptero autônomo RMAX. 97
6.20 Trajetória do VANT. 97
6.21 Combinações de algoritmos aplicadas para a produção dos resultados. . . 98
6.22 Erro da estimação da posição, obtido pelo algoritmo de registro automático de imagens, em cada um dos 1443 pontos de avaliação. Os algoritmos Canny e filtro gaussiano foram utilizados no algoritmo de registro automático de imagens. 99
6.23 Desvio padrão do erro da estimação da posição obtido pelo algoritmo de registro automático de imagens configurado com o Canny e o filtro Gaussiano. 100
6.24 Trajetória original do VANT, em vermelho, e a trajetória estimada, em azul. Apenas pontos classificados como EGM são exibidos na trajetória estimada. 100
6.25 Erro da estimação da posição obtido pelo algoritmo de registro automático de imagens configurado com o Sobel e o filtro Gaussiano. 102
6.26 Desvio padrão do erro das posições estimadas, obtidas pelo algoritmo de registro automático de imagens configurado com o Sobel e o filtro Gaussiano. 102

6.27 Trajetória original do VANT, em vermelho, e a trajetória estimada em azul. A trajetória estimada é formada por pontos classificados como EGM, e foi obtida pelo algoritmo de registro automático de imagens configurado com o Sobel e o filtro Gaussiano. 103
6.28 Gráfico do erro da estimação da posição, obtido pelo algoritmo de registro automático de imagens configurado com a fase de ativação MLP e o filtro gaussiano. 104
6.29 Gráfico do erro da estimação da posição, obtido pelo algoritmo de registro automático de imagens, em cada um dos 1443 pontos de avaliação. A MLP em LUT e o filtro gaussiano foram utilizados no algoritmo de registro de imagens. Pontos abaixo da linha vermelha tracejada são classificados como EGM. 105
6.30 Gráfico do desvio padrão do erro, obtido pelo algoritmo de registro de imagens configurado com a MLP de arquitetura determinada pelo MPCA, em azul, e a MLP em LUT, em verde. O filtro Gaussiano foi aplicado para a remoção de ruídos. 106
6.31 Trajetória original do VANT, em vermelho, e trajetória estimada pelo algoritmo de registro automático de imagens cofigurado com a MLP em LUT e o filtro Gaussiano, em azul. 107
6.32 Gráfico do erro da estimação da posição, obtido pelo algoritmo de registro automático de imagens configurado com o Canny e o filtro da Mediana. . 108
6.33 Gráfico do desvio padrão do erro da posição estimada pelo algoritmo de registro automático de imagens configurado com o Canny e o filtro da Mediana. 109
6.34 Trajetória original do VANT, em vermelho, e a trajetória estimada pelo algoritmo de registro automático de imagens configurado com o Canny e o filtro da Mediana, em azul 109
6.35 Gráfico do erro da estimação da posição, obtido pelo algoritmo de registro automático de imagens configurado com o Sobel e o filtro da Mediana. . 111
6.36 Gráfico do desvio padrão do erro das posições estimadas pelo algoritmo de registro automático de imagens configurado com o Sobel e o filtro da Mediana. 112
6.37 Trajetória estimada pelo algoritmo de registro automático de imagens, em azul, e trajetória real do VANT em vermelho. Somente os pontos classificados como EGM são exibidos na trajetória estimada. 112
6.38 Gráfico do erro da estimação da posição, obtido pelo algoritmo de registro automático de imagens configurado com a MLP em LUT e o filtro da Mediana. 114

6.39 Gráfico do desvio padrão do erro das posições estimadas pelo algoritmo de registro automático de imagens configurado com a MLP em LUT e o filtro da Mediana. 114
6.40 Trajetória estimada pelo algoritmo de registro automático de imagens, em azul, e a trajetória real do VANT, em vermelho. Somente os pontos classificados como EGM são exibidos na trajetória estimada. 115
6.41 Imagem de intensidade LiDAR da cidade de Baltimore - EUA 121
6.42 Exemplo de imagem aérea de intensidade LiDAR da trajetória do VANT. 122
6.43 Imagem de satélite georreferenciada da cidade de Baltimore, capturada na banda pancromática do sensor *Quickbird* 122
6.44 Imagem de intensidade LiDAR georreferenciada com a trajetória do VANT em vermelho. 123
6.45 Imagem de satélite georreferenciada com a trajetória do VANT em vermelho. 123
6.46 Combinações de algoritmos utilizadas para a produção dos resultados do segundo experimento. 124
6.47 Gráfico do erro da estimação da posição obtido pelo algoritmo de registro automático de imagens para cada um dos 172 pontos de avaliação. 126
6.48 Gráfico do desvio padrão do erro da estimação da posição do algoritmo de registro automático de imagens. O gráfico exibe o desvio padrão do erro de cada configuração do algoritmo de registro automático de imagens. 127
6.49 Trajetória estimada pelo algoritmo de registro automático de imagens configurado com a MLP em LUT e o filtro gaussiano, em azul. A trajetória original do VANT é exibida em vermelho 128
6.50 Gráfico do erro da estimação da posição obtido pelo algoritmo de registro automático de imagens configurado com cada algoritmo para a detecção de bordas (Canny, Sobel e MLP) e com o filtro da Mediana para a remoção de ruídos das imagens. 130
6.51 Gráfico do desvio padrão do erro da estimação da posição obtido pelo algoritmo de registro automático configurado com cada operador de detecção de bordas (Canny, Sobel e MLP) e com o filtro da Mediana. . . . 131
6.52 Em azul, a trajetória estimada pelo algoritmo de registro automático de imagens, configurado com a MLP em LUT e o filtro da Mediana. A trajetória simulada do VANT é exibida em vermelho. 132
6.53 Gráfico do erro da estimação da posição obtido pelo algoritmo de registro automático de imagens configurado com a MLP em LUT e o filtro da Mediana 134

6.54 Gráfico do desvio padrão do erro da estimação da posição, obtido pelo algoritmo de registro automático de imagens. A configuração do algoritmo de registro automático de imagens utilizou a MLP em LUT e o filtro da Mediana. 135
6.55 Trajetória estimada pelo algoritmo de registro automático de imagens, configurado com a MLP em LUT e o filtro da mediana, em azul. A trajetória simulada do VANT é exibida em vermelho. Somente os pontos classificados como EGM são exibidos na trajetória estimada. 136
6.56 Mapa Digital de Superfície do fundo do oceano com a trajetória do VANT em vermelho. 137
6.57 Subimagem do MDS georreferenciado do fundo do oceano, que representa uma imagem capturada pelo VANT. 139
6.58 Combinação empregadas para configurar o algoritmo de registro automático de imagens no terceiro experimento. 140
6.59 Gráfico do erro da estimação da posição, obtido pelo algoritmo de registro automático de imagem configurado com cada detector de borda e o filtro Gaussiano 142
6.60 Gráfico do desvio padrão do erro da estimação da posição, obtido pelo algoritmo de registro automático utilizando imagens de um MDS da superfície do fundo do oceano. 143
6.61 Trajetória estimada pelo algoritmo de registro automático de imagens configurado com a MLP em LUT e o filtro Gaussiano. 144
6.62 Gráfico do erro da estimação da posição, obtido pelo algoritmo de registro automático de imagens, configurado com os três operadores para a detecção de borda e com o filtro da Mediana. 146
6.63 Gráfico do desvio padrão do erro da estimação da posição obtido pelo algoritmo de registro automático configurado com cada um dos detectores de borda. 147
6.64 Trajetória estimada pelo algoritmo de registro automático de imagens configurado com a MLP em LUT e o filtro da Mediana. 148

7.1 Sequência de operações do algoritmo de odometria visual monocular. . . 150
7.2 Algoritmo de odometria visual monocular aplicado para estimar a trajetória do VANT. 153
7.3 Aplicação do algoritmo RANSAC para determinar modelo para ajuste dos dados. O modelo obtido é representado pela linha azul. Pontos em verde são os *inliers*. Pontos em vermelho são os *outliers*. Em (A) é exibido o resultado obtido após 5 iterações, com 14 *inliers*. Em (B) é exibido o resultado obtido após 11 iterações, com 60 *inliers*. 166

7.4 Correspondência de pontos de interesse de duas imagens do VANT. Em A a imagem capturada em um instante t_1. Em B a imagem capturada em um instante t_2. Em C a correspondência com a presença de *outliers*. Em D a correspondência sem a presença de *outliers*. 168
7.5 Imagens capturadas pela câmera de vídeo do helicóptero autônomo RMAX durante sua trajetória 173
7.6 Trajetória realizada pelo helicóptero autônomo RMAX. A trajetória possui aproximadamente 1 km de extensão. 174
7.7 Região sobrevoada pelo helicóptero autônomo RMAX e sua trajetória marcada em vermelho. O círculo preto marca o ponto de partida e o branco o ponto de chegada da aeronave. 175
7.8 Trajetória real do VANT em vermelho. Trajetória do VANT obtida por odometria visual monocular em azul. 175
7.9 Imagem de satélite com a região sobrevoada pelo helicóptero autônomo RMAX e sua trajetória marcada em vermelho. A trajetória estimada pelo algoritmo de odometria visual monocular está representada na cor azul. . 176
7.10 Erro de estimação da trajetória obtida pelo algoritmo de odometria visual monocular proposto. 177
7.11 Imagens de intensidade que representam as imagens de capturadas pela câmera LiDAR embarcada na aeronave durante sua trajetória. 179
7.12 Trajetória, em duas dimensões, realizada pelo VANT na simulação utilizando imagens de intensidade LiDAR. 180
7.13 Trajetória realizada pelo VANT exibida na imagem de intensidade LiDAR georreferenciada. 180
7.14 Casamento entre pontos de interesse em imagens de intensidade LiDAR . 181
7.15 Gráfico de posição da trajetória simulada do VANT, em vermelho, e o gráfico de posição da trajetória estimada por odometria visual monocular, em azul. 182
7.16 Trajetória simulada do VANT, em vermelho, e a trajetória estimada por odometria visual monocular, em azul. 183
7.17 Gráfico do erro da trajetória estimada do VANT por odometria visual monocular. A linha tracejada vermelha representa um erro de 5 metros. . 183
7.18 Exemplo de subimagens do MDS do fundo do oceano que representam a trajetória do VANT 185
7.19 Trajetória, em duas dimensões, realizada pelo VANT na simulação de navegação sobre uma região coberta por água. 186
7.20 Trajetória realizada pelo VANT exibida no MDS georreferenciado do fundo do oceano obtido por sensor LiDAR. 186

7.21 Correspondência entre pontos de interesse entre dois MDS subsequentes do fundo do mar. 187
7.22 Gráfico de posição da trajetória simulada do VANT, em vermelho, e o gráfico de posição da trajetória estimada por odometria visual monocular, em azul. 188
7.23 Trajetória simulada do VANT, em vermelho, e a trajetória estimada por odometria visual monocular, em azul. 188
7.24 Trajetória simulada do VANT, em vermelho, e a trajetória estimada por odometria visual monocular, em azul. 189

8.1 Modelo de espaço de estados. O modelo é dinâmico e em tempo discreto. 194
8.2 Distribuição de Tsallis para diferentes valores do parâmetro livre q. . . . 205
8.3 Metodologia aplicada para a obter a estimação da posição do VANT por fusão de dados utilizando o Filtro de Partículas Não-Extensivo. 208
8.4 Gráfico da correlação cruzada entre a imagem capturada pelo VANT e a área de correlação 210
8.5 Gráfico do erro da estimação da posição do VANT obtido por fusão de dados pelo FPNE, odometria visual e registro automático de imagens. . . 216
8.6 Gráfico do erro da estimação da posição do VANT obtido por fusão de dados pelo FPNE com $q = 2,57$ e com $q = 1$. 218
8.7 Gráfico do desvio padrão do erro da estimação da posição do VANT obtido por fusão de dados pelo FPNE configurado com $q = 2,57$ e com $q = 1$. 219
8.8 Trajetória estimada por fusão dos dados pelo FPNE configurado com $q = 2,57$, na cor azul, e a trajetória original do VANT, em vermelho . . 220
8.9 Gráfico do erro da estimação da posição do VANT obtido por fusão de dados pelo FPNE, odometria visual e registro automático de imagens. . . 222
8.10 Gráfico do erro da estimação da posição do VANT obtido por fusão de dados pelo Filtro de Partículas Não-Extensivo com $q = 2,57$ e com $q = 1$. 224
8.11 Gráfico do desvio padrão do erro da estimação da posição do VANT obtido por fusão de dados pelo FPNE configurado com $q = 2,57$ e com $q = 1$. 225
8.12 Trajetória estimada por fusão dos dados por FPNE na cor azul e a trajetória original do VANT em vermelho 225
8.13 Gráfico do erro da estimação da posição do VANT obtido por fusão de dados pelo FPNE, odometria visual e registro automático de imagens. . . 228
8.14 Gráfico do erro da estimação da posição do VANT obtido por fusão de dados pelo FPNE com $q = 2.57$ e com $q = 1$. 229

8.15 Gráfico do desvio padrão do erro da estimação da posição do VANT obtido por fusão de dados pelo FPNE com $q = 2,57$ e com $q = 1$. 230
8.16 Trajetória estimada por fusão dos dados por FPNE na cor azul, e a trajetória original do VANT em vermelho 230

B.1 Conjunto de 3 pontos correspondentes. R e t são as matrizes de rotação e o vetor de translação, respectivamente. 265
B.2 Translação dos centroides de cada conjunto de pontos para a origem. . . 267

C.1 FDP Gaussiana com $\mu = 0$ e diferentes valores de σ 272
C.2 FDP de Cauchy com $x_0 = 0$ e diferentes valores de γ 273
C.3 FDP de Levy com $a = 0$ e diferentes valores de c 274

LISTA DE TABELAS

Pág.

2.1 Modelos, números de rotores e pesquisas onde VANTs de asa rotativa foram aplicados 15

6.1 Arquitetura da MLP para a detecção de bordas em imagens, determinada pelo MPCA 84
6.2 Exemplos de padrões presentes na LUT 85
6.3 Resultados das quantidades de posições estimadas classificadas como EGM, SDGM e GM 101
6.4 Resultados das quantidades de posições estimadas classificadas como EGM, SDGM e GM 103
6.5 Resultados das quantidades de posições estimadas classificadas como EGM, SDGM e GM 107
6.6 Resultados das quantidades de posições estimadas classificadas como EGM, SDGM e GM 110
6.7 Resultados das quantidades de posições estimadas classificadas como EGM, SDGM e GM. 113
6.8 Resultados das quantidades de posições estimadas classificadas como EGM, SDGM e GM. 115
6.9 Tempo de execução na Raspberry PI 116
6.10 Tempo de execução dos algoritmos para a detecção de bordas na Raspberry PI 117
6.11 Tempo de execução na Zybo ZINQ 7000 117
6.12 Tempo de execução dos algoritmos para a detecção de bordas na Zybo ZINQ 7000 118
6.13 Número de estimações por segundo, obtido pelo algoritmo de registro automático de imagens configurado com cada detector de borda. Este resultado considera a execução realizada na Raspberry Pi. 118
6.14 Número de estimações por segundo, obtido pelo algoritmo de registro automático de imagens configurado com cada detector de borda. Este resultado considera a execução realizada na ZYBO ZINQ 7000. 119
6.15 Tempo de execução dos algoritmos para a detecção de bordas na Zybo ZINQ 7000 120
6.16 Resultado da média das métricas EGM, SDGM e GM do algoritmo de registro automático de imagens configurado com cada detector de borda. 125

6.17 Resultado da média das métricas EGM, SDGM e GM considerando as 4 combinações possíveis do ruído de escala e rotação. O algoritmo de registro automático de imagens foi configurado com os três operadores de detecção de borda e com o filtro da mediana. 129
6.18 Resultado das métricas EGM, SDGM e GM para o algoritmo de registro automático de imagens configurado com a MLP em LUT e o filtro da Mediana. 133
6.19 Resultado da média das métricas EGM, SDGM e GM. O algoritmo de registro automático de imagens foi configurado com cada operador de detecção de borda e com o filtro Gaussiano. 141
6.20 Resultado da média das métricas EGM, SDGM e GM para o algoritmo de registro automático de imagem configurado com cada detector de borda. 145

7.1 Avaliação dos descritores segundo as seguintes propriedades: invariante a rotação (Rotação), invariante a mudança de escala (Escala), invariante a transformação afim (Afim), repetibilidade, diferenciabilidade, quantidade, eficiência 157
7.2 Tempo médio de para obtenção da trajetória entre duas imagens subsequentes 177

8.1 Algoritmo de Filtro de Partícula 203
8.2 Algoritmo de Filtro de Partículas Não-Extensivo 206
8.3 Algoritmo para determinar o melhor valor de q 207
8.4 Algoritmo de FP para estimação da posição do VANT 209
8.5 Algoritmo de Filtro de Partículas Não-Extensivo para estimação da posição do VANT 211
8.6 Valores das métricas EGM, SDGM e GM para as três técnicas de estimação da Posição do VANT. 217
8.7 Valores das métricas EGM, SDGM e GM para o Filtro de Partículas Não-Extensivo com $q = 1$ e $q = 2,57$ 218
8.8 Valores das métricas EGM, SDGM e GM para as três técnicas de estimação da Posição do VANT. 223
8.9 Valores das métricas EGM, SDGM e GM para o FPNE com $q = 1$ e $q = 2,75$ 224
8.10 Valores das métricas EGM, SDGM e GM para as três técnicas de estimação da Posição do VANT. 228
8.11 Valores das métricas EGM, SDGM e GM para o FPNE com $q = 1$ e $q = 2,57$ 229

LISTA DE ABREVIATURAS E SIGLAS

AMAS	–	Anomalia Magnética do Atlântico Sul
ANAC	–	Agência Nacional de Aviação Civil
ANATEL	–	Agência Nacional de Telecomunicações
API	–	Application Programming Interface
ARP	–	Aeronave Remotamente Pilotada
AXI	–	Advanced eXtensive Interface
CASA	–	Civil Aviation Safety Agency
CLB	–	Configurable Logic Blocks
CPU	–	Central Processing Unit
DECEA	–	Departamento de Controle do Espaço Aéreo
EASA	–	European Aviation Safety Agency
EGM	–	Error Good Matching
EUA	–	Estados Unidos da América
FAA	–	Federal Aviation Administration
FAB	–	Força Aérea Brasileira
FAST	–	Features from Accelerated Segment Test
FDP	–	Função Densidade de Probabilidade
FDPs	–	Funções de Densidade de Probabilidade
FP	–	Filtro de Partículas
FPA	–	Filtro de Partículas Adaptativo
FPGA	–	Field Programmble Gate Array
FPNE	–	Filtro de Partículas Não-Extensivo
FREAK	–	Fast Retina Keypoint
GM	–	Good Matching
GNSS	–	Global Navigation Satellite System
GPS	–	Global Positioning System
GPU	–	Graphics Processing Unit
GSI	–	Geological Survey of Ireland
HDMI	–	High-Definition Multimedia Interface
IEAv	–	Instituto de Estudos Avançados
INFOMAR	–	Integrated Mapping For Sustainable Development of Ireland Marine Resources
INPE	–	Instituto Nacional de Pesquisas Espaciais
INS	–	Inertial Navigation System
ITA	–	Instituto Tecnológico de Aeronáutica
JIRA	–	Japanese Industrial Robot Association
KB	–	Kilo Bytes
LiDAR	–	Light Detection and Ranging
LUT	–	Look-Up Table
LXDE	–	Lightweight X11 Desktop Environment
MB	–	Mega Bytes

MDE	–	Modelos Digitais de Elevação
MDS	–	Modelos Digitais de Superfícies
MLP	–	Multilayer Perceptron
MOS	–	Mapa Digital de Superfície
MPI	–	Message Passing Interface
MPCA	–	Multiple Particle Collision Algorithm
MSMC	–	Método Sequencial de Monte Carlo
ORB	–	Oriented Fast and Rotated Brief
PCA	–	Particle Collision Algorithm
PITER	–	Processamento de Imagens em Tempo Real
PL	–	Programmable Logic
PS	–	Processing System
RAB	–	Registro Aeronáutico Brasileiro
RAI	–	Registro Automático de Imagens
RAM	–	Random Access Memory
RANSAC	–	Random Sample Consensus
RNA	–	Rede Neural Artificial
RNAs	–	Redes Neurais Artificiais
SAR	–	Synthetic Aperture Radar
SDGM	–	Standard Deviation Good Matching
SDRAM	–	Synchronous Dynamic Random Access Memory
SIFT	–	Scale Invariant Feature Transform
SLAM	–	Simultaneous Localization and Mapping
SOC	–	System on Chip
SONAR	–	Sound Navigation and Ranging
SPI	–	Serial Peripheral Interface
SURF	–	Speeded Up Robust Features
SVD	–	Singular Value Decomposition
UFMG	–	Universidade Federal de Minas Gerais
UNIFEI	–	Universidade Federal de Itajubá
URSS	–	União das Repúblicas Socialistas Soviéticas
USB	–	Universal Serial Bus
VANT	–	Veículo Aéreo Não Tripulado
VANTs	–	Veículos Aéreos Não Tripulados
VARC	–	Veículo Aéreo Radio Controlado

SUMÁRIO

Pág.

1 INTRODUÇÃO 1
1.1 MOTIVAÇÃO 7
1.2 OBJETIVO DO TRABALHO 8
1.3 CONTRIBUIÇÕES 9
1.4 ORGANIZAÇÃO DO TEXTO 10

2 VANT - ESTADO DA ARTE 13
2.1 REGULAMENTAÇÃO DA UTILIZAÇÃO DE VANTS NO BRASIL . . 20

3 MÉTODO E FERRAMENTAS 23
3.1 HARDWARE DE DESENVOLVIMENTO 30
3.1.1 RASPBERRY PI MODEL B-1 30
3.1.2 ZYBO ZYNQ 7000 34

4 FORMAÇÃO DE IMAGEM EM CÂMERA ESTENOPEICA . 37
4.1 CÂMERA ESTENOPEICA 37
4.1.1 PARÂMETROS EXTRINSECOS DA CÂMERA ESTENOPEICA . . 39
4.1.2 PARÂMETROS INTRÍNSECOS DA CÂMERA ESTENOPEICA . . . 42

5 FORMAÇÃO DE IMAGENS EM SENSORES LiDAR 47
5.1 FUNCIONAMENTO DO SENSOR LiDAR 49
5.2 MAPEAMENTO BATIMÉTRICO POR SENSORES LiDAR 52

6 ESTIMAÇÃO DA POSIÇÃO DO VANT POR REGISTRO AUTOMÁTICO DE IMAGENS 55
6.1 CONVERSÃO DE IMAGENS PARA TONS DE CINZA 60
6.2 FILTRAGEM DA IMAGEM 61
6.2.1 FILTRO DA MEDIANA 63
6.2.2 FILTRO GAUSSIANO 64
6.3 BINARIZAÇÃO DE IMAGENS MONOCROMÁTICAS 66
6.4 SEGMENTAÇÃO DE IMAGENS DIGITAIS 67
6.4.1 DETECÇÃO DE BORDAS 67
6.4.2 OPERADOR DE SOBEL 69
6.4.3 OPERADOR DE CANNY 71

6.4.4 REDES NEURAIS ARTIFICIAIS PARA DETECÇÃO DE BORDAS . 72
6.4.4.1 REDES NEURAIS ARTIFICIAIS 72
6.4.4.2 REDE NEURAL ARTIFICIAL COM PERCEPTRONS EM MÚLTIPLAS CAMADAS 75
6.4.4.3 CONFIGURAÇÃO AUTOMÁTICA DE REDES PERCEPTRON EM MÚLTIPLAS CAMADAS 78
6.4.4.4 ALGORITMO DE COLISÃO DE MÚLTIPLAS PARTÍCULAS . . . 79
6.4.4.5 TREINAMENTO DA REDE NEURAL COM PERCEPTRONS EM MÚLTIPLAS CAMADAS PARA A DETECÇÃO DE BORDAS . . . 81
6.4.4.6 IMPLEMENTAÇÃO DA MLP PARA DETECÇÃO DE BORDA EM FPGA 84
6.5 ESTIMAÇÃO DA POSIÇÃO DO VANT POR CORRELAÇÃO DE IMAGENS 88
6.6 RESULTADOS: DISCUSSÃO E ANÁLISE 90
6.6.1 RESULTADOS: MÉTRICAS DE AVALIAÇÃO 94
6.6.2 RESULTADOS: PRIMEIRO EXPERIMENTO 95
6.6.2.1 RESULTADOS: CANNY E FILTRO GAUSSIANO 98
6.6.2.2 RESULTADOS: SOBEL E FILTRO GAUSSIANO 101
6.6.2.3 RESULTADOS: MLP E FILTRO GAUSSIANO 104
6.6.2.4 RESULTADOS: CANNY E FILTRO DA MEDIANA 107
6.6.2.5 RESULTADOS: SOBEL E FILTRO DA MEDIANA 110
6.6.2.6 RESULTADOS: MLP E FILTRO DA MEDIANA 113
6.6.2.7 RESULTADO: TEMPO DE EXECUÇÃO 116
6.6.2.8 RESULTADOS: ANÁLISE DOS RESULTADOS DO PRIMEIRO EXPERIMENTO 119
6.6.3 RESULTADOS: SEGUNDO EXPERIMENTO 120
6.6.3.1 RESULTADOS: FILTRO GAUSSIANO COM ALGORITMOS DE DETECÇÃO DE BORDAS 124
6.6.3.2 RESULTADOS: FILTRO DA MEDIANA COM ALGORITMO DE DETECÇÃO DE BORDAS 128
6.6.3.3 RESULTADOS: REGISTRO AUTOMÁTICO DE IMAGENS DE INTENSIDADE LIDAR COM IMAGEM DE SATÉLITE NA BANDA DO VISÍVEL 132
6.6.3.4 RESULTADOS: ANÁLISE DOS RESULTADOS DO SEGUNDO EXPERIMENTO 136
6.6.4 RESULTADOS: TERCEIRO EXPERIMENTO 137

6.6.4.1 RESULTADOS: FILTRO GAUSSIANO COM ALGORITMOS DE DETECÇÃO DE BORDAS . . . 140
6.6.4.2 RESULTADOS: FILTRO DA MEDIANA COM ALGORITMOS DE DETECÇÃO DE BORDAS . . . 144
6.6.4.3 RESULTADOS: ANÁLISE DOS RESULTADOS DO TERCEIRO EXPERIMENTO . . . 148

7 ESTIMAÇÃO DA POSIÇÃO DE VANT POR ODOMETRIA VISUAL MONOCULAR . . . 149
7.1 ODOMETRIA VISUAL MONOCULAR . . . 151
7.1.1 ALGORITMO DE DETECÇÃO E DESCRIÇÃO DOS PONTOS DE INTERESSE - DESCRITORES . . . 155
7.1.2 SURF: SPEEDED UP ROBUST FEATURES . . . 157
7.1.3 MOVIMENTO DA CÂMERA A PATIR DE PONTOS CORRESPONDENTES . . . 161
7.1.4 OBTENÇÃO DA MATRIZ DE ROTAÇÃO E DO VETOR DE TRANSLAÇÃO . . . 163
7.1.5 REMOÇÃO DE CORRESPONDÊNCIAS INVÁLIDAS . . . 165
7.2 RESULTADOS: DISCUSSÃO E ANÁLISE . . . 168
7.2.1 MÉTRICAS DE AVALIAÇÃO . . . 171
7.2.2 RESULTADOS: PRIMEIRO EXPERIMENTO . . . 172
7.2.2.1 RESULTADOS: TEMPO DE EXECUÇÃO . . . 177
7.2.2.2 RESULTADOS: ANÁLISE DO RESULTADO DO PRIMEIRO EXPERIMENTO . . . 178
7.2.3 RESULTADOS: SEGUNDO EXPERIMENTO . . . 178
7.2.3.1 RESULTADOS: ANÁLISE DO RESULTADO DO SEGUNDO EXPERIMENTO . . . 184
7.2.4 RESULTADOS: TERCEIRO EXPERIMENTO . . . 184
7.2.4.1 RESULTADOS: ANÁLISE DO RESULTADO DO TERCEIRO EXPERIMENTO . . . 189

8 ESTIMAÇÃO DA POSIÇÃO DE VANT POR FUSÃO DE DADOS 191
8.1 ESTIMAÇÃO BAYESIANA . . . 193
8.2 FILTRO DE PARTÍCULAS . . . 197
8.2.1 AMOSTRAGEM PERFEITA . . . 198
8.2.2 AMOSTRAGEM POR IMPORTÂNCIA . . . 199
8.2.3 ALGORITMO DE FILTRO DE PARTÍCULAS . . . 200
8.3 FILTRO DE PARTÍCULAS NÃO-EXTENSIVO . . . 203

8.4 FILTRO DE PARTÍCULAS NÃO-EXTENSIVO PARA ESTIMAÇÃO DA POSIÇÃO DO VANT . . . 207
8.5 RESULTADOS: DISCUSSÃO E ANÁLISE . . . 211
8.5.1 MÉTRICAS DE AVALIAÇÃO . . . 214
8.5.2 RESULTADOS: PRIMEIRO EXPERIMENTO . . . 215
8.5.2.1 RESULTADOS: ANÁLISE DOS RESULTADOS DO PRIMEIRO EXPERIMENTO . . . 220
8.5.3 RESULTADOS: SEGUNDO EXPERIMENTO . . . 221
8.5.3.1 RESULTADOS: ANÁLISE DOS RESULTADOS DO SEGUNDO EXPERIMENTO . . . 226
8.5.4 RESULTADOS: TERCEIRO EXPERIMENTO . . . 227
8.5.4.1 RESULTADOS: ANÁLISE DOS RESULTADOS DO TERCEIRO EXPERIMENTO . . . 231

9 CONCLUSÕES . . . 233

REFERÊNCIAS BIBLIOGRÁFICAS . . . 239

APÊNDICE A - DECOMPOSIÇÃO EM VALORES SINGULARES . . . 259
A.1 NOTAÇÕES E DEFINIÇÕES . . . 259
A.2 TEOREMA DA DECOMPOSIÇÃO EM VALORES SINGULARES . . . 261

APÊNDICE B - DETERMINAR A ROTAÇÃO E TRANSLAÇÃO ENTRE PONTOS CORRESPONDENTES . . . 265
B.1 DETERMINAR OS CENTRÓIDES DE UM CONJUNTO DE PONTOS 266
B.2 DETERMINAR A MATRIZ DE ROTAÇÃO . . . 266
B.3 DETERMINAR O VETOR DE TRANSLAÇÃO . . . 268

APÊNDICE C - CONCEITOS DE PROBABILIDADE . . . 269
C.1 VARIÁVEL ALEATÓRIA . . . 269
C.2 VALOR ESPERADO DE VARIÁVEL ALEATÓRIA CONTÍNUA . . . 270
C.3 VARIÂNCIA DE VARIÁVEL ALEATÓRIA . . . 270
C.4 PROBABILIDADE CONDICIONAL . . . 270
C.5 DISTRIBUIÇÃO DE PROBABILIDADE MARGINAL . . . 271
C.6 DISTRIBUIÇÃO NORMAL . . . 271
C.7 DISTRIBUIÇÃO DE CAUCHY . . . 272
C.8 DISTRIBUIÇÃO DE LEVY . . . 273

1 INTRODUÇÃO

Robôs estão na mente dos seres humanos desde que estes começaram a construir sistemas. A tentativa de automatizar atividades realizadas por humanos aparecem nas máquinas desenvolvidas por artesões desde o século XV, como exemplo, pode-se citar as estátuas *Os Mouros*, da torre do relógio da Praça de São Marcos em Veneza, Itália, que tocam o sino na hora certa (PETER, 2011).

A origem da palavra robô vem do idioma tcheco. Em 1922 o escritor Karel Capeck, em sua obra *Robôs Universais de Rossum* (do inglês, *Rossums Universal Robots*) introduziu a palavra *rabota*, que significa trabalhador. O termo robótica foi introduzido pelo autor americano Isaac Asimov, em sua coletânea de contos intitulada *Eu, Robô* (do inglês, *I, Robot*), publicada em 1950 (NIKU, 2014).

A robótica é definida como a ciência que estuda o desenvolvimento de máquinas controladas por computador para interação com o ambiente. O desenvolvimento da robótica ocorre principalmente após a segunda metade do século XX e, atualmente, os robôs estão presentes em diversas áreas, como: exploração planetária, produção de automóveis em linha de montagem, limpeza doméstica, etc (PETER, 2011).

A Associação Japonesa de Robôs Industriais (JIRA, do inglês, *Japanese Industrial Robot Association*) estabelece a seguinte classificação para os robôs (NIKU, 2014):

- Classe 1: Dispositivo de Movimentação Manual: dispositivo com vários graus de liberdade mas controlado por um operador;
- Classe 2: Robô de Sequência Fixa: dispositivo que executa etapas sucessivas de uma tarefa. Não pode ser modificado para executar outra tarefa;
- Classe 3: Robô de Sequência Variável: dispositivo que executa etapas sucessivas de uma tarefa. Pode ser modificado para executar outra tarefa;
- Classe 4: Robô de Reprodução: operador humano realiza a tarefa manualmente, o robô registra os movimentos e realiza a mesma tarefa posteriormente;
- Classe 5: Robô de Controle Numérico: operador humano fornece ao robô um programa de movimentos;
- Classe 6: Robô Inteligente: robô capaz de compreender o ambiente no qual está inserido, detectar as possíveis mudanças e concluir com êxito a tarefa para qual foi programado.

Os robôs inteligentes (Classe 6) necessitam de sistemas que os habilitem para a tomada de decisões e os dotem de certo grau de autonomia para completar a tarefa para a qual foram projetados. Neste contexto, sistemas autônomos podem ser definidos como sistemas que desenvolvem por si mesmos leis e estratégias, que regulam seu comportamento no ambiente (CRESTANI, 2001). Para ser autônomo, um sistema deve ser capaz de operar em um ambiente, agir de forma benéfica para si e ser capaz de concluir tarefas que são cruciais à sua existência futura (CRESTANI, 2001). Uma definição mais técnica para sistemas autônomos é encontrada em Fabro (1996), que define: são sistemas que executam suas tarefas sem a necessidade de interferência externa durante o processo, adequando seu comportamento, adaptando-se ao ambiente de forma a contornar situações imprevistas.

Particularmente, robôs inteligentes móveis são dotados de sistemas que possibilitam a navegação autônoma. Estes sistemas são caracterizados pela capacidade de definir uma sequência de ações tomadas pelo robô, que possui um conjunto limitado de sensores, que o auxiliam para atingir um objetivo específico. (FIGUEIREDO, 1997).

A navegação autônoma de robôs inteligentes móveis é um desafio científico e tecnológico, pois tenta imitar a capacidade de mobilidade de sistemas biológicos dotados de movimento, que não encontram dificuldade em determinar uma trajetória em um ambiente estranho. Entretanto, dotar um sistema eletrônico com tal habilidade é uma tarefa árdua (FIGUEIREDO, 1997; MAIA et al., 2017).

Atualmente, a navegação autônoma é desenvolvida em robôs aquáticos, terrestres e aéreos (PETER, 2011). De modo que, há grande interesse da comunidade científica no desenvolvimento de robôs inteligentes móveis aéreos, pois o emprego de Veículo Aéreo Não Tripulado (VANT) é cada vez mais comum em diversas áreas, como exemplo:

- Monitoramento do espaço (LEWIN, 2015);
- Inspeção Industrial (NIKOLIC et al., 2013);
- Monitoramento do uso da terra (HERWITZ et al., 2004);
- Operações de busca e resgate (WAHARTE; TRIGONI, 2010);
- Acompanhamento de impactos ambientais (HODGSON et al., 2013);
- Vigilância de fronteiras (GIRARD et al., 2004).

- Detecção de mudanças no solo (MOTA et al., 2014)

O aumento na utilização de Veículos Aéreos Não Tripulados (VANTs) ocorre do fato de que apresentam menos risco a vida humana, pois não há tripulação embarcada, e ao baixo custo de desenvolvimento e utilização, quando comparados com o uso de aeronaves tripuladas (CONTE; DOHERTY, 2008).

Atualmente, as soluções mais empregadas para a navegação autônoma de VANTs são:

- Utilização de um Sistema Inercial de Navegação (INS, do inglês, *Inertial Navigation System*) associado a um Sistema Global de Navegação por Satélite (GNSS, do inglês, *Global Navigation Satellite System*) para fornecer ao sistema de controle da aeronave as informações necessárias para a navegação. O principal GNSS em atividade, em escala global, é o Sistema de Posicionamento Global (GPS, do inglês, *Global Positioning System*), controlado pelo governo estadunidense, fato que impede que tecnologias militares sejam desenvolvidas pela aeronáutica brasileira com segurança, mesmo sendo os Estados Unidos da América uma nação aliada ao Brasil (SOUZA C. R, 2005). Além disso, o GPS pode sofrer interferência em seu sinal. Esta interferência pode ser maliciosa, causada por meio de técnicas de *spoofing* ou *jamming* (CONTE; DOHERTY, 2008), ou causada por fenômenos naturais como a Anomalia Magnética do Atlântico Sul (AMAS), que é uma dobra da ionosfera sobre a atmosfera terrestre, que pode afetar os dados oriundos de satélites, inclusive os dados provenientes da constelação que compõe o GPS, provocando uma imprecisão em suas medições (ABDU et al., 2005). A AMAS é um fenômeno que tem sua maior atuação sobre o continente sul-americano, o que torna o GPS não adequado para sistemas que irão operar no Brasil (ABDU et al., 2005). O INS, principalmente os de baixo custo utilizados em VANTs de pequeno porte, possui um acúmulo de erro devido ao uso prolongado o que impede que seja utilizado para a navegação sem a associação com outro sensor, como o GPS, para sua correção;

- Utilização de um sistema de visão computacional, que por meio de uma ou mais câmeras realiza a captura de imagens do terreno sobrevoado e por meio do processamento de tais imagens fornece informações para o sistema de controle da aeronave para a sua navegação (SHIGUEMORI et al., 2007).

A visão computacional é uma área da ciência da computação, que por meio de um conjunto de técnicas computacionais torna explícita as propriedades geométricas (forma e posição de objetos) e dinâmicas (velocidade de objetos) do mundo real em três dimensões a partir de imagens. A principal vantagem no emprego de visão computacional para a navegação autônoma é o fato desta técnica fornecer informações detalhadas do ambiente (FORSYTH; PONCE, 2011).

O sistema de visão computacional projetado para a navegação autônoma de VANTs deve ser robusto, capaz de fornecer uma visão adequada do mundo, adaptar-se as variações do ambiente, altamente eficiente e ser um sistema embarcado. Sua principal função é fornecer a estimação da posição da aeronave, ou seja, informar ao sistema de controle a posição ocupada pela aeronave no espaço, essa resposta deve ser fornecida em tempo suficiente para que o sistema de controle do VANT possa manter a segurança da aeronave durante o voo (CONTE; DOHERTY, 2008).

Para o sistema embarcado de visão computacional, empregado para a estimação da posição do VANT, ser altamente eficiente há a necessidade da utilização de tecnologias que consomem pouca energia, que não necessitem de sofisticado equipamento computacional, para apoiar sua operação, e que realizem o processamento com alto desempenho. Neste contexto, a *Field Programmable Gate Array* (FPGA) surge como uma alternativa viável para implementação de sistemas embarcados, pois seu consumo energético por operação é baixo, não necessita de um equipamento computacional sofisticado para suporte de sua operação e possui um alto paralelismo computacional, o que permite com que este hardware reprogramável gere aplicações com alto desempenho (PEDRONI, 2008).

Estimar a posição do VANT utilizando um sistema de visão computacional não é uma tarefa fácil, pois há variações no processo de captura de imagens (SHIGUEMORI et al., 2007). Estas variações são oriundas do movimento da aeronave, por exemplo, a altura da aeronave em relação ao solo, e do ambiente sobrevoado, por exemplo, condições de luminosidade devido ao horário do voo ou presença de nebulosidade (CASTRO et al., 2009). A grande maioria dos trabalhos que aplicam a visão computacional para a navegação autônoma de VANT utilizam câmeras de vídeo, que captam dados provenientes do espectro visível para realizar o imageamento da região sobrevoada, assim, a navegação pode ocorrer somente durante o dia (LENSKIY; LEE, 2010). Outro problema relacionado com a navegação autônoma de VANT por visão computacional é a existência de regiões onde as câmeras de vídeo não são capazes de imagear, como exemplo, regiões cobertas por água.

As informações de sensores, como o altímetro, bússola, acelerômetro, etc, podem ser utilizadas para tratar as variações oriundas do movimento da aeronave, e auxiliar o sistema de visão computacional a fornecer uma estimação correta de sua posição (CONTE; DOHERTY, 2008; GOLTZ et al., 2011; BRAGA et al., 2016). A solução para superar as variações provocadas pela iluminação do ambiente é a utilização de diferentes sensores, como: sensores infra-vermelho, Radar de Abertura Sintética (SAR, do inglês, Synthetic Aperture Radar) e o *Light Detection And Ranging* (LiDAR), este último pode ser empregado para o imageamento de regiões cobertas por água a uma profundidade máxima de 60 metros. Além disso, sensores LiDAR vêm sendo aplicados com sucesso na construção de sistemas de processamento de imagens para a navegação de robôs em situação de baixa ou sem luminosidade (MCMANUS et al., 2013).

As principais técnicas de visão computacional atualmente empregadas para estimar a posição do VANT são: o registro automático de imagens e a odometria visual (CLARKE, 2014; MATHE; BUSONIU, 2015).

Pequenos desvios no formato, tamanho, orientação e na luminosidade do ambiente podem afetar o resultado do registro automático de imagens. Além disso, esta técnica pode não ser estável durante o processo de estimação da posição. O emprego de imagens com as borda detectadas no registro automático de imagens surge como uma alternativa para diminuir influências de variações na luminosidade do ambiente que possam afetar o resultado (GOLTZ et al., 2011; BRAGA et al., 2016; SILVA et al., 2015).

Os algoritmos tradicionais para extração de bordas, como o Sobel e o Canny, não demonstraram resultados adequados quando aplicados para a extração de bordas e posterior registro automático das imagens para a estimação da posição de VANT (GOLTZ et al., 2011; BRAGA et al., 2016; SILVA et al., 2015). O emprego de Redes Neurais Artificiais (RNAs) surge como uma nova possibilidade para o processo de extração de bordas de imagens, anterior ao registro automático das mesmas (GOLTZ et al., 2011; BRAGA et al., 2016; SILVA et al., 2015).

A odometria visual é uma técnica bastante estável para estimar a posição do VANT, mas possui um erro de deriva, ou seja, um erro de estimação do posicionamento que aumenta com o distanciamento do ponto de partida da aeronave (NISTER et al., 2004; SCARAMUZZA; FRIEDRICH, 2011). Portanto, a informação fornecida por apenas uma técnica de visão computacional pode não ser suficiente para estimar de forma correta a posição do VANT, sendo ideal, o emprego da fusão de dados provenientes das duas

técnicas (CHUNHUI et al., 2014).

A fusão de dados é um conjunto de técnicas que combinam dados de diferentes fontes para que o resultado final seja melhor do que quando considerada a informação isoladamente (KHALEGHI et al., 2013). Recentemente, a fusão de dados tem sido aplicada para combinar a estimação da posição fornecida por odometria visual e registro automático de imagens para obtenção de uma informação melhor sobre o posicionamento da aeronave (CHUNHUI et al., 2014).

Segundo Gustafsson (2013), uma técnica aplicada para realizar a fusão de dados é o algoritmo de Filtro de Partículas (FP), que vem sendo aplicado para a combinação de dados de estimação da posição de VANT, oriundos de técnicas de visão computacional, para obtenção de uma melhor informação sobre o posicionamento da aeronave (CONTE; DOHERTY, 2008; CHUNHUI et al., 2014). Nesta pesquisa de doutorado, é aplicada uma variação do algoritmo de FP, denominada Filtro de Partículas Não-Extensivo (FPNE), para realizar a fusão dos dados (CAMPOS VELHO; FURTADO, 2011).

Verificando-se as considerações descritas, nesta tese de doutorado sugere-se o desenvolvimento de uma abordagem de visão computacional, para estimar a posição de um VANT e auxiliar sua navegação autônoma em caso de falha do sinal do GNSS. A abordagem deverá operar em três situações distintas: 1 - em ambiente urbano com boas condições de luminosidade; 2 - em ambiente urbano com baixa ou sem luminosidade; 3 - sobre regiões cobertas por água, por exemplo, o oceano.

Os métodos realizarão a estimação da posição da aeronave por meio do processamento de imagens oriundas de uma câmera de vídeo, quando a aeronave sobrevoar regiões urbanas em condições favoráveis de iluminação (durante o dia), e imagens oriundas de um sensor LiDAR, quando sobrevoar regiões urbanas com baixa ou sem luminosidade ou regiões cobertas por água.

A abordagem proposta estima a posição da aeronave por meio da fusão de dados do registro automático de imagens e da odometria visual. O algoritmo de FPNE é aplicado para a fusão dos dados das duas técnicas de visão computacional. Adicionalmente, esta tese de doutorado sugere uma metodologia para embarcar em um VANT a abordagem proposta, responsável pela estimação da posição de VANT, utilizando *hardware* de baixo consumo e que permite a programação de alto desempenho.

1.1 MOTIVAÇÃO

Atualmente, um do principais temas de pesquisas na área de VANT é a busca por um sistema que permita sua navegação autônoma em caso de falha do sinal do GNSS e em ambiente onde não seja possível o imageamento com câmera, que captura apenas a energia eletromagnética no comprimento de onda visível, como o exemplo, ambientes com baixa ou sem luminosidade ou em regiões cobertas por água. Embora muito tenha sido estudado, ainda existem questões em aberto sobre como desenvolver um sistema de visão computacional para a navegação de VANT nas condições anteriormente citadas. Neste sentido, destaca-se a importância do desenvolvimento de uma metodologia, que permita a navegação de VANT em caso de falha do sinal do GNSS, em ambiente urbano durante o dia, com baixa ou sem luminosidade e sobre regiões cobertas por água, para ser embarcada em uma aeronave.

O tema sugerido nesta tese de doutorado possui elevada relevância. VANTs têm sido fonte de estudos de diversos centros de pesquisas nacionais, principalmente Instituto Nacional de Pesquisas Espaciais *(INPE)* (GOLTZ et al., 2011; SILVA et al., 2015) e Institutos de Estudos Avançados *(IEAv)* (SHIGUEMORI et al., 2007) e internacionais, como a Universidade de Linköping - Suécia (CONTE; DOHERTY, 2008). Além disso, há uma demanda crescente por automatizar processos que a participação humana é fonte constante de falhas.

A principal vantagem do uso da abordagem proposta nesta tese de doutorado é a possibilidade da navegação autônoma de VANTs sobre regiões urbanas, em qualquer condição de luminosidade, e em regiões cobertas por água. A solução opera sem a participação de um piloto em uma estação terrestre, o que permite que mão-de-obra especializada não seja ocupada durante a operação da aeronave, barateando os custos de voo e, principalmente, preservando a vida humana, pois a pilotagem de aeronaves é trabalho de grande periculosidade (CHAVES, 2013). Outra vantagem do emprego da abordagem proposta é a não dependência do GPS, que é uma tecnologia controlada por uma nação estrangeira (SOUZA C. R, 2005) e que sofre influência da AMAS (ABDU et al., 2005).

O tema desta tese de doutorado está relacionado com o desenvolvimento de três projetos de pesquisas que apresentam grande relevância para a Força Aérea Brasileira *(FAB)*:

- Avaliação de técnicas de correção de trajetória de Veículos Aéreos Não Tripulados por Imagens;

- Sistema embarcado para processamento de vídeos obtidos por Veículos Aéreos Não Tripulados para aplicação em navegação autônoma;

- Processamento de Imagens em Tempo-Real *(PITER)*.

Os três projetos são desenvolvidos no IEAv, com a participação de pesquisadores de diversas universidades federais: Universidade Federal de Itajubá *(UNIFEI)*, Universidade Federal de Minas Gerais *(UFMG)*, e de dois institutos de pesquisas nacionais, INPE e Instituto de Tecnologia da Aeronáutica *(ITA)*.

1.2 OBJETIVO DO TRABALHO

Dentro deste contexto, a tese de doutorado tem como objetivo geral apresentar uma abordagem para estimar o posicionamento de um VANT, quando este sobrevoa regiões urbanas (em qualquer condição de luminosidade) e cobertas por água, sem a participação de um piloto em uma estação terrestre e que não utilize GNSS para a sua localização. Com a finalidade de desenvolver uma metodologia que permita que o VANT realize seu voo sem a iteração com um piloto e não dependente de sinal do GNSS para a localização de sua trajetória.

Para o cumprimento deste objetivo geral, propõe-se os seguintes objetivos específicos:

a) A abordagem proposta utilizará dados de estimação da posição provenientes do registro de imagens, por este motivo, propõe-se o estudo de método para estimar a posição do VANT por registro automático de imagens;

b) Para evitar influência da iluminação do ambiente e permitir a utilização de imagens de diferentes sensores no processo de estimação da posição por registro automático de imagens, propõe-se o estudo de técnicas, baseadas em RNAs para a extração de bordas em imagens. As imagens com as bordas detectadas são aplicadas no processo de registro automático de imagens;

c) A abordagem proposta utilizará dados de estimação da posição do VANT provenientes da odometria visual, por este motivo, propõe-se o estudo de método para estimar a localização do VANT por odometria visual;

d) Estudo do sensor e de imagens oriundas de um equipamento LiDAR;

e) Estudo da formação de imagens provenientes de câmera de vídeo;

f) A abordagem proposta realizará a fusão dos dados da estimação da posição por registro automático de imagens e por odometria visual para obtenção de uma melhor informação sobre o posicionamento da aeronave, por este motivo, propõe-se o estudo do algoritmo de FPNE para realizar a fusão dos dados provenientes da odometria visual e do registro automático de imagens;

g) A abordagem deverá ser desenvolvida para permitir a estimação da posição durante o voo do VANT, por este motivo, propõe-se a implementação da abordagem em hardware que possa ser embarcado;

h) Implementação do processo de detecção de bordas por RNA em FPGA, para aumentar a eficiência do algoritmo de registro automático de imagens e melhorar o desempenho do algoritmo quando comparado as técnicas tradicionais de extração de bordas, como Canny e Sobel;

i) A abordagem poderá ser embarcada em um VANT, por este motivo propõe-se a medição do tempo de processamento da técnica proposta;

j) Teste da abordagem proposta em base de dados com imagens obtidas por uma câmera de vídeo de um VANT;

k) Teste da metodologia em bases de dados que possuam imagens LiDAR de regiões urbanas e regiões cobertas por água.

1.3 CONTRIBUIÇÕES

As principais contribuições originais presentes nesta tese de doutorado, para o problema de estimação da posição do VANT para permitir sua navegação em ambiente urbano, em boas ou más condições de iluminação, ou quando a aeronave sobrevoa regiões cobertas por água, são:

- A utilização de um método de otimização, o MPCA, para determinar a arquitetura de uma RNA aplicada para a detecção de bordas;
- O desenvolvimento de um extrator de bordas baseado em RNA e implementado em uma FPGA;
- A aplicação de imagens provenientes de um sensor LiDAR, o que permite a estimação da posição do VANT, quando este sobrevoa uma região coberta

por água, ou quando a aeronave sobrevoa uma região urbana em com pouca ou sem luminosidade;

- A utilização do FPNE no processo de fusão de dados oriundos do registro de imagens e da odometria visual.

1.4 ORGANIZAÇÃO DO TEXTO

O texto aqui apresentado está organizado da seguinte forma:

- Capítulo 2 - VANT - ESTADO DA ARTE - descreve o histórico dos VANTs, também faz uma revisão bibliográfica dos trabalhos que propõem soluções para a navegação autônoma de VANT por processamento de imagens. Para finalizar, o capítulo apresenta uma discussão sobre a regulamentação da utilização de VANT no Brasil;

- Capítulo 3 - MÉTODO E FERRAMENTAS - apresenta a abordagem proposta nesta tese de doutorado para estimar a posição do VANT e assim permitir sua navegação autônoma, em caso de falha de sinal do GNSS, sobre regiões urbanas, em qualquer condição de luminosidade e sobre regiões cobertas por água. O capítulo também apresenta os computadores no qual a abordagem é implementada;

- Capítulo 4 - FORMAÇÃO DE IMAGENS EM CÂMERA ESTENOPEICA - faz a descrição do processo de formação de imagens em câmeras de vídeo (câmeras estenopeica);

- Capítulo 5 - FORMAÇÃO DE IMAGENS EM SENSORES LiDAR - descreve o funcionamento do sensor LIDAR e a obtenção de imagens pelo sensor;

- Capítulo 6 - ESTIMAÇÃO DA POSIÇÃO DE VANT POR REGISTRO AUTOMÁTICO DE IMAGENS - faz a descrição completa do algoritmo de estimação de posição por registro automático utilizando imagens oriundas da câmera de vídeo e do sensor LiDAR. O capítulo apresenta a fundamentação teórica sobre o algoritmo de registro automático de imagens e também apresenta os resultados de estimação da posição obtidos pelo algoritmo;

- Capítulo 7 - ESTIMAÇÃO DA POSIÇÃO DO VANT POR ODOMETRIA VISUAL MONOCULAR - faz a descrição do algoritmo aplicado para a

estimação da posição do VANT por odometria visual monocular, que emprega imagens oriundas da câmera de vídeo e do sensor LiDAR. O capítulo apresenta a fundamentação teórica sobre o algoritmo de odometria visual monocular e também apresenta os resultados de estimação da posição obtidos pelo algoritmo;

- Capítulo 8 - ESTIMAÇÃO DA POSIÇÃO DE VANT POR FUSÃO DE DADOS - apresenta o algoritmo de fusão dos dados obtidos pelo registro automático de imagens e pela odometria visual monocular. A fusão de dados é realizada por meio do FPNE. O capítulo apresenta a fundamentação teórica sobre o algoritmo de fusão de dados e também apresenta os resultados obtidos por esta técnica;

- Capítulo 9 - CONCLUSÕES - apresenta as considerações finais do estudo realizado, as contribuições alcançadas e sugestões de possíveis trabalhos futuros.

2 VANT - ESTADO DA ARTE

A definição para VANT, segundo o Departamento de Defesa dos Estados Unidos da América, é: toda aeronave motorizada, que não possui piloto a bordo, que pode ser remotamente pilotada ou voar autonomamente, para fins não-recreativos, recuperável após o cumprimento da missão para o qual foi projetada e que transporta uma carga, que pode ser letal ou não. São excluídos da categoria de VANT os artefatos mais leves que o ar, como pipas, balões e dirigíveis, e os misseis rádio-controlados (DEPARTMENT, 2013). Esta é a mesma definição para VANT aplicada no Brasil (ANAC, 2017)

O interesse por VANT inicia-se em 1911, apenas 5 anos após a criação do avião por Santos Dumont, quando Elmir Sperry, o inventor do giroscópio, inicia seus estudos de aplicação do rádio-controle em aeronaves. Com o início da primeira guerra mundial, em 28 de julho de 1914, e a entrada dos Estados Unidos da América (EUA) no conflito, em 6 de abril de 1917, a marinha americana financiou os estudos de Elmir Sperry para o desenvolvimento do primeiro Veículo Aéreo Radio Controlado (VARC), um torpedo aéreo para entregar bombas em um alvo determinado, que não foi utilizado durante o conflito (VAN WIEN, 1969).

Durante o período entre guerras, de 1918 a 1939, a aviação comercial obteve um grande desenvolvimento, o que possibilitou o aprimoramento de diversas tecnologias, entre elas, dos sistemas de rádio-controle. Este desenvolvimento permitiu o surgimento da primeira Aeronave Remotamente Pilotada (ARP), a F-5L, que foi testada pela marinha americana em 1924, e pela primeira vez na história uma aeronave operada por rádio-controle foi capaz de decolar, voar e pousar com sucesso. No final da década de 1930 inicia-se a segunda guerra mundial e durante este período os projetos para desenvolvimento de VANT não foram os principais alvos das forças armadas dos países do Eixo (Alemanha, Itália e Japão) e dos países Aliados (EUA, França, Reino Unido e União das Repúblicas Socialistas Soviéticas (URSS))(KEANE; CARR, 2013).

Com o fim da Segunda Guerra Mundial, tem início imediato a Guerra Fria, conflito indireto de ordem ideológico, tecnológico, econômico e militar entre EUA e URSS. Com o objetivo de conter o avanço do comunismo, as forças armadas dos EUA investem no desenvolvimento de um banco de dados com informações para assistir seu planejamento estratégico. Marinha, exército e aeronáutica americanas iniciam missões de reconhecimento em aeronaves tripuladas, que, entre Abril de 1950 e Abril de 1969, culminaram com a perda de 163 vidas. Desta forma, cresce o interesse do

emprego de VANT (NEWCOME, 2004; KEANE; CARR, 2013).

A Guerra do Vietnã, combate inserido no contexto da Guerra Fria, entre os anos de 1955 e 1975, foi o primeiro conflito armado onde VANTs foram amplamente empregados em missões de inteligência. Durante este combate 3435 missões de inteligência e reconhecimento foram realizadas com a utilização de VANTs. A maior parte dessas missões eram lançadas de aeronaves DC-130, que transportavam o VANT denominado "*Lightning Bug*", este era equipado com uma câmera de alta qualidade utilizada para capturar imagens das tropas vietnamitas. Os helicópteros H-35 realizavam a recuperação do VANT enquanto ele descia de paraquedas, após completar a missão para qual foi designado (NEWCOME, 2004; KEANE; CARR, 2013).

Na década de 1990, durante a Guerra do Golfo, foi utilizado o principal VANT de combate já produzido, o AAI RQ-2 Pioneer. Este VANT foi desenvolvido a partir de uma cooperação entre Israel e EUA e foi utilizado em mais de 300 missões entre os anos de 1990 e 1991. O AAI RQ-2 Pioneer foi empregado em missões de inteligência e em missões de ataque, pois foi modificado para carregar armamentos. Desde o fim da Guerra do Golfo, o AAI RQ-2 Pioneer continuou sendo empregado com sucesso em outros combates: como na Guerra da Bósnia, entre Abril de 1992 e Dezembro de 1995; na Guerra da Somália, entre Dezembro de 1992 a Março de 1995; e, mais recentemente, este VANT foi empregado com sucesso na Guerra do Afeganistão, entre Outubro de 2001 e Dezembro de 2014 (NEWCOME, 2004; KEANE; CARR, 2013).

A aplicação de VANT desenvolveu-se primeiramente no campo militar, devido as vantagens oferecidas por estas aeronaves em missões categorizadas como tediosas, sujas e perigosas (em inglês essas missões são denominadas "DDD", dull, dirty and dangerous), mas, a partir do final da década de 1990, observa-se uma rápida proliferação do emprego de VANT na sociedade civil, pois há uma barateamento nos custos de produção de aeronaves de pequeno porte e, também, uma miniaturização de componentes eletrônicos, permitindo embarcar tecnologias para ampliar a utilização dessas aeronaves (CLARKE, 2014).

Há duas principais categorias de VANT: asa rotativa e asa fixa. Os VANTs de asa rotativa são os mais empregados atualmente, pois apresentam a capacidade de pairar no ar, esses são divididos em dois subtipos, helicópteros e multirrotores, onde os multirrotores são os preferidos por serem menos suscetíveis a danos e de fácil reparo (BALAMURUGAN et al., 2016). A Tabela 2.1 apresenta exemplos de VANT de asa rotativa empregados pela comunidade científica recentemente.

Tabela 2.1 - Modelos, números de rotores e pesquisas onde VANTs de asa rotativa foram aplicados

VANT	$N^{\underline{o}}$ de rotores	Artigo
Helicóptero Yamaha RMAX	2	(CONTE; DOHERTY, 2008)
Helicóptero Acrobático 23cc	2	(CHEVIRON et al., 2007)
Helicóptero (modelo não especificado)	2	(WANG et al., 2012a)
Helicóptero (modelo não especificado)	2	(WANG et al., 2012b)
Quadcóptero (modelo não especificado)	4	(LI et al., 2013)
Quadcóptero (modelo não especificado)	4	(MAGREE; JOHNSON, 2014)
Quadcóptero (modelo não especificado)	4	(SA et al., 2013)
Quadcóptero AscTec Pelican	4	(MEBARKI et al., 2013)
Quadcóptero (modelo não especificado)	4	(ACHTELIK et al., 2013)
Quadcóptero (modelo não especificado)	4	(BACHRACH et al., 2010)
Hexacóptero (modelo não especificado)	6	(SCARAMUZZA et al., 2014)
Hexacóptero (modelo não especificado)	6	(WEISS et al., 2012)

Fonte: Adaptado de Balamurugan et al. (2016)

VANTs tornaram-se uma tecnologia com potencial para possibilitar a inovação em aplicações civis. Desde o início do século XXI observa-se o emprego de VANT em diversas áreas como: agricultura, monitoramento ambiental, inspeção industrial, etc. Devido ao aumento da aplicação de VANT, há grande interesse no desenvolvimento de uma técnica que permita sua navegação autônoma de forma segura (CLARKE, 2014).

O principal desafio durante a navegação autônoma de VANT é estimar e manter a posição da aeronave. Um dos sensores embarcado é o INS, que provém a medida de posicionamento, mas esta informação é afetada por um erro acumulativo de deriva. O GNSS pode ser empregado para fornecer a informação de posicionamento do VANT e corrigir o erro do INS, mas seu sinal pode ficar indisponível devido a interferência maliciosa ou a fenômenos naturais. Para superar as situações de falha do sinal GNSS, técnicas de visão computacional tem sido empregadas para estimar a posição do VANT em situações de falha do sinal GNSS e realizar a correção do INS e assim permitir a navegação autônoma. (BALAMURUGAN et al., 2016; CLARKE, 2014).

As técnicas de visão computacional mais empregadas para para a navegação autônoma de VANT são:

- registro automático de imagens - processo de atribuir coordenadas geográficas a uma imagem (imagem template) por meio do casamento de feições homólogas de uma imagem georreferenciada (imagem referência) da mesma cena (CONTE; DOHERTY, 2008; BALAMURUGAN et al., 2016; BRAGA et al., 2016);

- odometria visual - processo de estimação do movimento de um sensor imageador por meio da análise de duas imagens subsequentes que contenham grande parte da mesma cena imageada (CONTE; DOHERTY, 2008; SCARAMUZZA; FRIEDRICH, 2011; BALAMURUGAN et al., 2016).

A pesquisa de Conte e Doherty (2008) apresenta um sistema para a navegação autônoma de VANT fundamentado no registro automático de imagens. Para estimar a posição do VANT, o sistema realiza o registro automático da imagem da região sobrevoada, capturada pelo câmera do VANT, com uma imagem aérea georreferenciada da mesma região. Para evitar que varições na luminosidade do ambiente influencie no resultado, o registro automático das imagens é realizado após a detecção de bordas nas imagens.

Em Goltz et al. (2011) é descrito um sistema para a navegação autônoma de VANT a partir do registro automático de imagens. O sistema realiza o registro automático entre a imagem capturada pelo VANT e uma imagem de satélite georreferenciada. Uma RNA é aplicada para extrair as bordas das imagens e, assim, evitar as variações de luminosidade do ambiente no resultado do registro automático das imagens.

O trabalho desenvolvido por Silva et al. (2015) apresenta uma abordagem de casamento de imagens para realizar a estimativa de posição de VANT e, assim, possibilitar sua navegação de forma autônoma. A estimativa de posição da aeronave é obtida por meio cálculo da uniformização das imagens, via extração de bordas, e descoberta da posição na imagem georreferenciada, por meio da maior correlação com a imagem capturada em tempo de voo. O método é avaliado considerando diferentes tipos de terreno, como: florestas, rodovias e zona urbana.

O artigo de Santos et al. (2013) descreve a utilização do casamento de padrões em imagens com objetivo de verificar a eficiência da mesma em uma tarefa que consiste em auxiliar a navegação de um VANT.

A dissertação de mestrado de Jaimes (2005) propõem duas abordagens distintas para melhorar o processo de estimação de posição geográfica de um VANT, por meio do

casamento de padrões das imagens capturadas pela aeronave em tempo de voo. A primeira abordagem é uma técnica que utiliza os ângulos de inclinação do VANT, fornecidos pelos sensores inerciais da aeronave, para obter-se a matriz homográfica e corrigir a imagem. A segunda abordagem apresenta um método para a correção de distorção projetiva e espectral nas imagens aéreas.

A pesquisa desenvolvida por Silva et al. (2015) apresenta uma metodologia para realizar a navegação autônoma de VANT, em situação de baixa luminosidade ou em período noturno, por meio do registro automático de uma imagem capturada por uma câmera termal com uma imagem de satélite georreferenciada, obtida por um sensor óptico na banda do visível. O registro automático de imagem ocorre após a extração de bordas das imagens por uma RNA.

Em Amidi (1996) é apresentado uma das primeiras pesquisas presentes na literatura que utiliza a odometria visual para estimar a posição de um VANT. Nesta pesquisa, o autor desenvolve uma metodologia para a navegação autônoma de VANT, mas que utiliza o sinal do GPS para aumentar a precisão da posição estimada.

A pesquisa desenvolvida por Wang et al. (2012a) aplica a odometria visual para determinar a posição de um VANT e assim realizar sua navegação de forma autônoma. Em sua pesquisa, o autor apresenta a vantagem de utilizar a odometria visual monocular ao invés da estéreo. A primeira técnica estima a posição utilizando imagens provenientes de uma câmera de vídeo, enquanto a segunda estima a posição utilizando imagens de duas ou mais câmeras. Segundo o autor a odometria visual monocular é mais adequada para a navegação autônoma de VANT, pois realiza uma quantidade menor de cálculos para estimar a posição da aeronave, o que torna os algoritmos mais rápidos quando comparados a odometria visual estéreo.

A odometria visual monocular pode apenas estimar a posição correta de um VANT segundo um fator de escala, ou seja, necessita da informação de outro sensor, como o velocímetro ou altímetro, para determinar corretamente a translação da aeronave (SCARAMUZZA; FRIEDRICH, 2011). O trabalho de Huang et al. (2016) aplica a odometria visual monocular para estimar a posição de VANT utilizando uma câmera RGB-D, que além da imagem fornece a informação de distância do objeto imageado, solucionando o problema do fator de escala. Uma das etapas do algoritmo de odometria visual é a detecção e descrição dos pontos de interesse, no estudo desenvolvido por (HUANG et al., 2016) o algoritmo FAST é aplicado para esta finalidade.

A pesquisa de Krajnik et al. (2012) desenvolve um sistema de navegação autônoma

para VANT baseado em odometria visual para sobrevoar regiões com poucos marcos fixos no solo, o autor denomina essas regiões como "*poor landmarks*". Para realizar a detecção e descrição dos pontos de interesse aplica-se o algoritmo "*Scale-Invariant Feature Transform*" (SIFT).

O trabalho de Byun et al. (2017) aplica a odometria visual para estimar a posição de VANT, de médio e grande porte, e que voam a alta altitude, aproximadamente 4500 metros. Para atingir este objetivo o sistema utiliza câmeras de alta definição. O algoritmo aplicado no processo de detecção e descrição dos pontos de interesse é o "*Speed Up Robust Feature*" (SURF).

O Mapeamento e Localização Simultâneo (SLAM, do inglês, Simultaneous Localization and Mapping) é uma técnica que emprega visão computacional e que vem sendo utilizada para a navegação autônoma de robôs terrestres e aéreos (MATHE; BUSONIU, 2015). O SLAM constrói um mapa do ambiente e localiza a posição do robô neste mapa por meio do casamento de padrões de imagens capturadas no momento presente. Porém, o SLAM não se mostra adequado para ser aplicado à navegação autônoma de VANT, pois para diminuir o erro da estimação da posição, o VANT deve retornar a uma posição já visitada anteriormente, o que aumenta o consumo energético durante a operação (MATHE; BUSONIU, 2015; CLARKE, 2014). Mesmo em técnicas de SLAM denominadas "*small loop closure*" a necessidade de retorno a um ponto já visitado pela aeronave, apesar deste retorno ser pequeno, há um aumento do consumo energético para manter a localização da aeronave (CONTE; DOHERTY, 2008).

Apesar de o registro automático de imagens e a odometria visual serem as técnicas de visão computacional mais empregadas e adequadas para auxiliar a navegação autônoma de VANT, elas podem apresentar erros (MATHE; BUSONIU, 2015; CLARKE, 2014). O registro automático de imagens é sensível a variações no formato, tamanho e orientação das imagens e necessitam de ajuste de escala e rotação das imagens antes do registro dessas, e, além disso, este método pode ser instável em sua estimação (CONTE; DOHERTY, 2008; GOLTZ et al., 2011; BRAGA et al., 2016; SILVA et al., 2015). A odometria visual é um método estável de estimação da posição, mas apresenta um erro de deriva que aumenta com o passar do tempo (NISTER et al., 2004; CONTE; DOHERTY, 2008; SCARAMUZZA; FRIEDRICH, 2011; SCARAMUZZA; FRIEDRICH, 2012). Desta forma, para contornar os problemas apresentados pelas duas técnicas, pesquisas recentes apresentam como estratégia a fusão dos dados provenientes destas técnicas (CONTE; DOHERTY, 2008; CHUNHUI et al., 2014).

A fusão de dados é a combinação de dados derivados de fontes diferentes ou de dados diferentes da mesma fonte, de modo que a informação resultante tenha menos incerteza do que quando essas fontes são utilizadas individualmente. Há diversas técnicas que são empregadas para realizar a fusão de dados, entre elas pode-se citar o Filtro de Kalman e o Filtro de Partículas (KAM et al., 1997; GUSTAFSSON, 2013).

A pesquisa de Cheviron et al. (2007) apresenta um sistema para estimar a posição de um VANT que utiliza dados do acelerômetro, giroscópio e visão computacional. O sistema aplica o Filtro de Kalman Estendido para realizar a fusão dos dados dos diversos sensores da aeronave. O sistema proposto apresenta alta estabilidade e as estimativas das posições são comparáveis as do GNSS.

O trabalho de (WEISS et al., 2012) utiliza a combinação de sensores visuais e inerciais para promover a estimação da posição do VANT, e assim permitir sua navegação de forma autônoma em caso de falha do sinal do GNSS. Por meio do Filtro de Kalman Estendido o sistema realiza a fusão dos dados oriundos da odometria visual e do sensor inercial para ter uma estimação da posição da aeronave. O sistema proposto foi implementado e testado em um hexacoptero.

Em Bachrach et al. (2010) apresenta-se um sistema para a navegação autônoma de VANT por meio da fusão de dados oriundos de um sistema de visão computacional e do sensor inercial da aeronave. O sistema de fusão de dados é baseado no Filtro de Kalman Estendido, que diminuí a incerteza de estimação da posição do VANT. O sistema de visão computacional é baseado em odometria visual e as informações dos três giroscópios e dos três acelerômetros do sensor inercial. O autor destaca que o sistema apresentado é testado e validado em um quadcoptero.

Em Conte e Doherty (2008) explora-se a possibilidade de usar imagens de satélite georreferenciadas para implementar o sistema de navegação de um VANT em caso de falha do sinal do GNSS. O sistema de navegação combina sensores inerciais, odometria visual e registro automático de imagens. Para realizar a fusão dos dados o sistema aplica o Filtro de Partículas. Os resultados desta pesquisa demonstram que é possível extrair informações de posição do processamento de imagens aéreas, mesmo quando não há informação do GNSS. Neste estudo, mostra-se que informações da fusão de dados podem ser usadas para compensar o erro de deriva da estimação da posição do VANT, que ocorre quando somente os sensores inerciais e a odometria visual são aplicados. O estudo também demonstra que a fusão de dados gera uma estimação com menor instabilidade que quando utiliza-se apenas o registro de imagens.

A pesquisa desenvolvida por Chunhui et al. (2014) aplica o Filtro de Partículas para realizar a fusão dos dados de estimação provenientes do registro automático de imagens e provenientes da odometria visual. O resultado da pesquisa demonstra que a fusão dados produz resultados melhores que os algoritmos de visão computacional quando considerados isoladamente.

Antes do desenvolvimento de qualquer pesquisa em que se utilize VANT é importante o conhecimento da legislação que regulamenta sua aplicação. Por este motivo, na próxima seção apresenta-se a regulamentação para a utilização de VANT no Brasil.

2.1 REGULAMENTAÇÃO DA UTILIZAÇÃO DE VANTS NO BRASIL

Em 2 de maio de de 2017 a diretoria colegiada da Agência Nacional de Aviação Civil (ANAC) aprovou a regulamentação, Regulamento Brasileiro de Aviação Civil Especial $n^{\underline{o}}$ 94/2017 (RBACE $n^{\underline{o}}$ 94/2017) para a utilização de VANT no Brasil. O objetivo desta regulamentação é tornar viável a operação dessas aeronaves preservando a segurança das pessoas (ANAC, 2017).

A regulamentação foi elaborada levando-se em conta regras já etipuladas em normativos de outras autoridades de aviação civil como a Adimnistração de Aviação Federal (FAA, do inglês, Federal Aviation Administration), Autoridade de Segurança de Aviação Civil (CASA, Civil Aviation Safety Authority) e Agência de Segurança da Aviação Europeia (EASA, do inglês, European Aviation Safety Agency), organizações reguladoras da aviação nos Estados Unidos da América, Austrália e União Europeia, respectivamente.

Desde de 2 de Maio de 2017 as operações com VANT, sejam elas para uso comercial ou experimental, devem seguir as regras da regulamentação RBACE $n^{\underline{o}}$ 94/2017 da ANAC, que são complementares aos normativos do Departamento de Controle do Espaço Aéreo (DECEA) e da Agência Nacional de Telecomunicações (ANATEL).

A regulamentação RBACE $n^{\underline{o}}$ 94/2017 estabelece duas categorias de VANT:

- Aeromodelos: é toda aeronave não tripulada com finalidade de recreação, que deve ser pilotada em locais próprios como clubes de aeromodelismo e locais autorizados pela ANAC;
- Aeronave Remotamente Pilotada (ARP ou VANT-ARP): é o VANT pilotado a partir de uma estação remota de solo com finalidade comercial,

corporativa ou experimental.

As ARPs foram classificadas em três categorias de acordo com o peso máximo de decolagem do equipamento:

- Classe 1 - VANT-ARP acima de 150 Kg - este equipamento deve sofrer o mesmo processo de certificação para aeronaves tripuladas e registradas no Registro Aeronáutico Brasileiro (RAB). O operador necessita de licença e habilitação emitida pela ANAC. O voo desta aeronave precisa ser registrado e autorizado pela ANC;
- Classe 2 - VANT-ARP acima de 25 Kg e abaixo ou igual a 150 Kg - o projeto do equipamento deve ser aprovado pela ANAC e também registradas no RAB. O operador necessita de licença e habilitação emitida pela ANAC. O voo desta aeronave precisa ser registrado e autorizado pela ANAC;
- Classe 3 - VANT-ARP abaixo ou igual a 25 Kg - Há nesta categoria uma subdivisão, VANT-ARP com até 0.25 Kg, que não precisam ser registrados. VANT-ARP com mais de 0.25 Kg e abaixo ou igual a 25 Kg, que operarem em até 400 pés (120 metros) e com o operador mantendo contato visual com a aeronave sem auxílio de lentes e equipamentos ou outro observador, não precisam de projeto aprovado pela ANAC mas precisam ser cadastrados. VANT-ARP com mais de 0.25 Kg e abaixo ou igual a 25 Kg, que operarem acima de 400 pés (120 metros) e sem o operador mantendo contato visual com a aeronave, sendo isto possível apenas com operador utilizando lentes e equipamentos ou outro observador, necessitam do projeto ser autorizado pela ANAC. Os operadores destas aeronaves precisam de licença e habilitação emitida pela ANAC para voos acima de 400 pés.

Para a pilotagem de VANT-ARP de qualquer classe ficou estabelecida a idade mínima de 18 anos. É obrigatório a contratação de seguro com cobertura de danos a terceiros nas operações de VANT-ARP com mais de 0.25 Kg. Não podem ser transportadas pessoas, animais e cargas perigosas por VANT-ARP de qualquer classe. Artigos perigosos relacionados à atividade agrícola, como agrotóxicos, podem ser transportados e lançados por estas aeronaves com pedido e cadastramento anterior da atividade e do operador junto a ANAC. Poderão ser transportados equipamentos eletrônicos, como câmera fotográficas, filmadoras, computadores e baterias necessários ao funcionamento da aeronave.

A operação totalmente autônoma, ou seja, naquela onde o piloto remoto não é capaz e intervir, de qualquer classe de VANT-ARP é proibida no Brasil. Essas operações diferem-se das automatizadas, nas quais o piloto remoto pode intervir em qualquer momento da operação.

Irregularidades em relação ao cumprimento da RBACE $n^{\underline{o}}$ 94/2017 são passíveis de sansões previstas no Código Brasileiro de Aeronáutica (Lei $n^{\underline{o}}$ 7565/86).

3 MÉTODO E FERRAMENTAS

Este capítulo apresenta a abordagem de estimação de posição do VANT proposta nesta tese de doutorado e também as plataformas de hardware escolhidas para sua implementação e realização de testes.

Na Figura 3.1, descreve-se a abordagem utilizada para realizar a estimação da posição do VANT por meio da fusão de dados provenientes das técnicas de visão computacional. As técnicas de visão computacional empregadas são o registro automático de imagens e a odometria visual monocular. A fusão de dados é feita pelo algoritmo de FPNE. O método apresentado na Figura 3.1 é empregado para a estimação da posição por meio do processamento de imagens oriundas de uma câmera de vídeo ou de imagens oriundas do sensor LiDAR, para permitir a navegação do VANT em situações de luminosidade adequada, baixa ou nenhuma luminosidade ou navegação sobre regiões cobertas por água, por exemplo, o oceano, em uma situação de falha do sinal do GNSS.

Figura 3.1 - Abordagem proposta para a estimação da posição de VANT por fusão de dados de técnicas de visão computacional.

Fonte: Produção do autor.

Utilizando uma câmera de vídeo ou um sensor LiDAR, o sistema captura imagens da região sobrevoada, durante todo o tempo de voo da aeronave, e armazena estas imagens. A câmera de vídeo é utilizada para capturar as imagens da região sobrevoada quando a navegação do VANT ocorre durante o dia e sobre áreas urbanas. Contudo,

o sensor LiDAR é utilizado para capturar imagens quando o VANT sobrevoa uma região coberta por água, por exemplo, o oceano, ou uma região urbana em situação de baixa ou sem luminosidade (período noturno).

O método ilustrado na Figura 3.1, entra em operação no momento da falha do sinal do GNSS do VANT para fornecer um mecanismo de correção do erro de estimação da posição do INS. Os INS, principalmente os de baixo custo, possuem um erro de deriva e necessitam de uma mecanismo de correção. Na Figura 3.1, no instante T da falha do GNSS, o método de estimação da posição da aeronave por processamento de imagens é acionado para substituir a estimativa da posição fornecida pelo GNSS (CONTE; DOHERTY, 2008; BRAGA et al., 2016).

Na figura 3.1, a abordagem proposta de estimação da posição do VANT seleciona, entre as imagens armazenadas, a imagem capturada no instante da falha T (Imagem VANT (T)) e a imagem capturada no instante imediatamente anterior a falha, imagem capturada no instante $T - d_T$ (Imagem VANT ($T - d_T$)). Segundo (SCARAMUZZA; FRIEDRICH, 2011), para uma correta estimação da posição realizada pela odometria visual monocular, a diferença d_T entre esses dois instantes de tempo deve ser suficiente para que a maior parte da imagem capturada no instante $T - d_T$ esteja contida na imagem capturada no instante T. Estas duas imagens selecionadas são provenientes do mesmo sensor, câmera de vídeo ou LiDAR, dependendo das condições do ambiente e da região sobrevoada.

Na abordagem proposta, e ilustrada na Figura 3.1, as imagens capturadas no instante $T - d_T$ e no instante T são utilizadas na odometria visual monocular, que estima a posição relativa do VANT (SCARAMUZZA; FRIEDRICH, 2011; SCARAMUZZA; FRIEDRICH, 2012). A odometria visual monocular determina a posição e a orientação de um sensor por meio da análise das características presentes em duas imagens subsequentes. Desde que o sensor esteja fixado a um agente (por exemplo, robô ou aeronave), determinar o orientação e posição do sensor é determinar a orientação e posição do agente (SCARAMUZZA; FRIEDRICH, 2011; SCARAMUZZA; FRIEDRICH, 2012). O termo monocular refere-se ao fato das imagens serem provenientes de um único sensor (SCARAMUZZA; FRIEDRICH, 2011; SCARAMUZZA; FRIEDRICH, 2012).

A Figura 3.2 ilustra com mais detalhes o algoritmo de odometria visual monocular aplicado na abordagem proposta.

Figura 3.2 - Algoritmo de odometria visual monocular aplicado para estimar a posição do VANT.

Fonte: Produção do autor.

Na Figura 3.2, a primeira etapa do algoritmo de odometria visual monocular é a conversão das imagens, capturadas no instante $T - d_t$ e no instante T, para tons de cinza. Esta operação é necessária pois os algoritmos aplicados para a detecção e descrição dos pontos de interesse operam sobre imagens com uma única banda.

As duas etapas seguintes do algoritmo de odometria visual, ilustrado Na Figura 3.2, consistem na detecção e descrição dos pontos de interesse das imagens subsequentes (Imagem $(T - d_T)$ e Imagem (T)). Nestas etapas, localizam-se pontos de interesse pertencentes as duas imagens e para cada um deles constrói-se um vetor de atributos que os identifiquem de maneira única. Posteriormente, realiza-se o casamento (correspondência) entre os pontos de interesse pertencentes as duas imagens, ou

seja, determina-se na imagem capturada no instante T o mesmo ponto de interesse presente na imagem $T - d_t$. Esta correspondência é realizada por meio de uma métrica de similaridade sobre o vetor de atributos de cada ponto de interesse, por este motivo que o vetor de atributos deve identificar de maneira única cada ponto de interesse, para evitar correspondências erradas. A partir desses pares de pontos de interesse correspondentes das duas imagens que pode-se obter o movimento (matriz de rotação e o vetor de translação) realizado pelo sensor imageador entre a captura da Imagem $T - d_T$ e a captura da Imagem T . De posse do movimento realizado pelo sensor, pode-se estimar a posição do VANT. O método de estimação da posição do VANT por odometria visual monocular será retomado com mais detalhes no capítulo 7.

Na Figura 3.1, paralelamente ao processo de odometria visual monocular, ocorre a estimação da posição do VANT por registro automático de imagens. O registro de imagens é o processo de correspondência de duas ou mais imagens da mesma cena, porém, obtidas por diferentes sensores, em instantes de tempo diferentes ou sob diferentes pontos de observação. O objetivo do registro de imagens é relacionar as coordenadas de uma imagem (linha e coluna) com coordenadas geográficas (latitude e longitude) (PEDRINI; SCHARTZ, 2008). Na Figura 3.1, o registro automático de imagens é realizado entre uma imagem georreferenciada da região sobrevoada (denominada Imagem Referência) e uma imagem aérea, capturada pelo sensor do VANT em tempo de voo no instante T (Imagem VANT (T)).

O registro automático da imagem aérea sobre a imagem georreferenciada estima a posição absoluta do VANT (CONTE; DOHERTY, 2008; BRAGA et al., 2016). Devido a aeronave ocupar o pixel central da imagem aérea captura pelo VANT (Imagem VANT T), localizando-se esse pixel na imagem georreferenciada é possível obter suas coordenadas geográficas (CONTE; DOHERTY, 2008; GOLTZ et al., 2011; BRAGA et al., 2016). A Figura 3.4 ilustra o processo de registro automático de uma imagem aérea capturada por um VANT com uma imagem georreferenciada da região sobrevoada.

Figura 3.3 - Registro automático de entre uma imagem capturada por VANT e uma imagem georreferenciada.

Fonte: Produção do autor.

Nesta pesquisa, o registro automático de imagens é realizado entre a imagem capturada pelo sensor (câmera de vídeo ou LiDAR) da aeronave no instante de tempo T e uma imagem georreferenciada da cena sobrevoada. A imagem georreferenciada é proveniente de um sensor óptico ou de um sensor LiDAR. O sensor de origem das imagens está relacionada com a situação de voo da aeronave, em boas condições de iluminação do terreno (navegação durante o dia) e sobre uma região urbana utiliza-se a imagens capturadas pela câmera de vídeo e a imagem georreferenciada do sensor óptico. Em condições de baixa ou sem iluminação sobre uma região urbana ou em navegação sobre regiões cobertas por água, utiliza-se imagens do sensor LiDAR da aeronave e imagens georreferenciadas também provenientes de sensores LiDAR.

A Figura 3.4 ilustra com mais detalhes o algoritmo de registro automático de imagens aplicado neste trabalho.

Figura 3.4 - Algoritmo aplicado para realizar a estimação da posição do VANT por registro automático de imagens.

Fonte: Produção do autor.

Na Figura 3.4, a imagem aérea, capturada pelo sensor da aeronave, é convertida para tons de cinza, este processo é necessário para melhorar o desempenho do algoritmo de registro de imagens e porque as demais etapas operam sobre imagens em tons de cinza. Em seguida, para a eliminação de ruídos indesejados, a imagem aérea passa por um processo de filtragem, esta operação se faz necessária para evitar que

ruídos presentes na imagem capturada pelo sensor imageador do VANT influenciem negativamente o resultado do registro automático de imagens.

Para diminuir o erro no processo de registro automático, a imagem aérea passa por ajuste de escala e rotação em relação à imagem georreferenciada. O ajuste de escala converte a imagem aérea para a mesma escala da imagem georreferenciada, este ajuste é obtido utilizando informações do altímetro da aeronave. O ajuste de rotação alinha a imagem aérea em relação a imagem georreferenciada, este alinhamento da imagem é realizado utilizando as informações da bússola magnética da aeronave.

Na Figura 3.4, a imagem georreferenciada é convertida para tons de cinza, este processo é necessário para melhor o desempenho do algoritmo de registro de imagens e porque as etapas seguintes operam sobre imagens em tons de cinza. Para a eliminação do ruídos indesejáveis na imagem georreferenciada, esta passa por um processo de filtragem, que é necessária para evitar que estes ruídos influenciem negativamente o resultado do registro de imagens.

Na Figura 3.4, para evitar áreas de baixa probabilidade da localização do VANT na imagem georreferenciada e para diminuir a utilização de recursos computacionais, o registro automático de imagens é realizado em uma região restrita da imagem georreferenciada. Esta região restrita, denominada área de correlação, é selecionada utilizando a melhor informação disponível sobre a posição do VANT fornecida pelo INS. Como o sistema é acionado no instante da falha do sinal do GNSS, o erro de estimação da posição fornecido pelo INS é igual ao erro do GNSS (CONTE; DOHERTY, 2008; SILVA et al., 2015; BRAGA et al., 2016).

Na Figura 3.4, para diminuir a influência da variação de iluminação do ambiente entre as imagens no resultado final do registro automático e para permitir a utilização de imagens de diferentes sensores, a imagem aérea e a imagem georreferenciada tem suas bordas detectadas (CONTE; DOHERTY, 2008; BRAGA et al., 2016; SILVA et al., 2015).

O último passo do algoritmo estimação da posição do VANT por registro automático de imagens é o cálculo do índice de correlação entre área de correlação e a imagem aérea capturada pelo sensor da aeronave. O cálculo do índice de correlação é realizado pelo algoritmo de correlação cruzada após o processo de extração de bordas das imagens. O resultado da correlação cruzada entre as imagens é uma matriz, cujo o maior valor representa a estimação da posição do VANT (CONTE; DOHERTY, 2008; GOLTZ et al., 2011; SILVA et al., 2015; BRAGA et al., 2016). O algoritmo de estimação

da posição do VANT por registro automático de imagens será retomado com mais detalhes no capítulo 6.

O passo final da abordagem de estimação da posição do VANT proposta, exibida na Figura 3.1, é a fusão da resposta da estimação da posição fornecida pela odometria visual com a resposta da estimação da posição fornecida pelo registro automático de imagens. Para esta finalidade, utiliza-se a fusão de dados, que por meio da combinação de dados derivados de fontes distintas ou de dados distintos da mesma fonte produz uma informação com menor incerteza do que quando considera-se essas fontes de dados isoladamente (MOSHIRI, 2010). O termo menor incerteza refere-se ao fato de produzir um dado com mais confiabilidade, mais preciso (MOSHIRI, 2010). Há diferentes algoritmos desenvolvidos para a realizar a fusão de dados, nesta pesquisa, é aplicado o FPNE. Segundo (CAMPOS VELHO; FURTADO, 2011), o FPNE pode produzir respostas melhores de estimação que o algoritmo tradicional de FP. A resposta do FPNE é a estimação da posição do VANT, que é utilizada para substituir a informação do GNSS, quando há interrupção do seu sinal, na correção do INS da aeronave. O algoritmo de estimação da posição do VANT por fusão de dados será retomado com mais detalhes no capítulo 8.

O método de estimação da posição do VANT proposto necessita ser implementado em um sistema de computação que permita manter a segurança da aeronave durante voo (CONTE; DOHERTY, 2008; BRAGA et al., 2016). As restrições de tempo para a resposta sobre o posicionamento da aeronave necessitam de plataformas computacionais com bom desempenho computacional e que permitam ser embarcadas em um VANT. Desta forma, a próxima seção descreve as plataformas computacionais aplicadas para a implementação da abordagem de estimação de posição do VANT proposta.

3.1 HARDWARE DE DESENVOLVIMENTO

Dois computadores "*single-board*", que podem ser embarcados em VANT devido ao seu baixo consumo energético e tamanho pequeno, foram utilizados para a implementação do método proposto, eles são: Raspberry PI Model B-1 e o Zybo Zynq 7000.

3.1.1 RASPBERRY PI MODEL B-1

O computador Raspberry PI Model B-1 é exibido na Figura 3.5

Figura 3.5 - Computador Raspberry PI Model B-1.

Fonte: Obtido de Raspberry Pi (2015)

As principais características do Raspberry PI Model B-1 são (Raspberry Pi, 2015):

- Peso de 45 gramas;
- Comprimento de 8.56 cm e largura de 5.65 cm;
- Consumo energético de 3.5 wats e 700 miliampere;
- Processador ARM1176JZF-S single core com frequência de 700 MHz com arquitetura de 32-bits;
- Memória de 512 "*Mega Bytes*" (MB) "*Synchronous Dynamic Random Access Memory*" (SDRAM);
- 2 portas "*Universal Serial Bus*" (USB) 2.0 para conexão de teclado, "mouse" ou outro periférico;
- Saída de vídeo "*High-Definition Multimedia Interface*" (HDMI);
- Unidade de Processamento Gráfico (GPU, do inglês, Graphics Processing Unit) Broadcom VideoCore IV com freqüência de 250 MHz;
- "*Slot*" para cartão de memória de estado sólido;

Utilizando um cartão de memória de no mínimo 4 "*Giga Bytes*" (GB) é possível a instalação do sistema operacional Raspbian para realizar o controle do Raspberry

PI Model B-1. Este sistema operacional é baseado no Debian (Linux), possui interface gráfica amigável (ambiente "*desktop*") "*Lightweight X11 Desktop Environment*" (LXDE), nele também é possível a instalação de compiladores da linguagem de programação C e da biblioteca OpenCV. A biblioteca OpenCV é utilizada no desenvolvimento de aplicativos na área de visão computacional e possui como principais características: a facilidade na implementação de aplicações; ser multiplataforma, está disponível em diversas linguagens de programação (C, C++, Java, Pyhton); e totalmente livre ao uso acadêmico e comercial.

O Raspberry PI Model B-1 possui um conjunto de 26 pinos, exibido na Figura 3.6, para a conexão com o mundo exterior. Dos 26 pinos, 17 são "*General Purpose Input/Output*", que permitem a comunicação com outros periféricos, 5 são terra e os outro 4 pinos são para a alimentação do periférico conectado (Raspberry Pi, 2015).

Figura 3.6 - Conjunto de pinos GPIO para a conexão de periféricos.

Fonte: Obtido de Raspberry Pi (2015).

Um dos periféricos que pode ser conectado à Raspberry PI Model B-1, por meio do conjunto de 26 pinos mostrado na 3.6, é a placa LOGI PI, exibida na Figura 3.7, que possui entre seus principais componentes a FPGA Spartan 6 LX9.

Figura 3.7 - Placa LOGI PI, que possui a FPGA Spartan 6 LX9.

Fonte: Obtido de ValentFX (2014).

As principais características da FPGA Spartan 6 LX9 são (XILINX, 2015):

- Fabricante: Xilinx Inc;
- Número de elementos lógicos: 9152;
- Número de Blocos Lógico Configuráveis (CLB, do inglês, Configurable Logic Blocks): 715;
- Total de Mémória de Acesso Aleatório (RAM, do inglês, Random Access Memory): 72 KB (Kilo Bytes).

A comunicação (lado da CPU) entre a CPU da Raspberry e a FPGA Spartan 6, presente na placa LOGI PI, é realizada via "*Serial Peripheral Interface*" (SPI), que é um protocolo desenvolvido para a comunicação de um microcontrolador e periféricos (VALENTFX, 2014).

A comunicação entre CPU e FPGA é exibida na Figura 3.8.

Figura 3.8 - Comunicação entre CPU e FPGA no computador Raspberry PI Model B-1.

Fonte: Adaptado de ValentFX (2014).

O componente "*Wishbone*", exibido na Figura 3.8, é um barramento de código aberto destinado a comunicação de periféricos, na placa LOGI PI é o responsável pela a comunicação CPU-FPGA do lado FPGA. O tamanho da palavra na comunicação CPU-FPGA é de 16 bits. O clock do SPI é de 32 MHz. Testes em laboratório mostram que o "*throughput*" de comunicação CPU-FPGA é de 3.0 MB por segundo (VALENTFX, 2014).

A empresa Valent-FX, desenvolvedora da placa LOGI PI, criou uma Interface de Programação de Aplicação (API, do inglês, Application Programming Interface) desenvolvida em linguagem C e Python, para tornar transparante a comunicação entre a CPU e a FPGA.

3.1.2 ZYBO ZYNQ 7000

O computador Zybo Zynq 7000, exibido na Figura 3.9, é um hardware para sistema embarcado de nova geração, que utiliza a tecnologia "*System-on-Chip*" (SoC) em seu "*micro-chip*". Esta nova geração de computadores, que utilizam a tecnologia SoC, contém em um único encapsulamento uma parte de lógica programável (FPGA) e um ou mais núcleos dedicados com processador ARM (DIGILENT, 2016).

Figura 3.9 - Placa Zybo Zynq 7000. SoC com processador ARM Cortex-A9 e FPGA Artix-7.

Fonte: Obtido de Digilent (2016).

Os computadores SoC consomem menos energia do que os computadores "*multi-chip*" que eles substituem, pois a maior parte do consumo ocorre na comunicação entre os componentes do sistema. Em computadores SoC, como todos os compo-

nentes estão no mesmo "*chip*" e internamente conectados o consumo de energia é reduzido. Além disso, o computador SoC fornece uma execução mais rápida devido a comunicação entre os componentes ser mais veloz (FLYNN; LUK, 2014).

As principais caracteristicas do Zybo Zynq 7000 são (DIGILENT, 2016):

- Peso de 114 gramas;
- Comprimento de 12.27 cm e largura de 7.62 cm;
- Consumo energético de 12.5 wats e 2.5 ampere;
- Processador ARM Cortex 9 dual core com frequência de 650 MHz e arquitetura de 32-bits;
- FPGA Artix-7;
- Memória de 512 MB SDRAM;
- 1 porta USB 2.0 para conexão de teclado, "mouse" ou outro periférico;
- Saída de vídeo HDMI;
- "*Slot*" para cartão de memória de estado sólido;

Utilizando um cartão de memória de 4 GB é possível a instalação de um sistema operacional baseado em Linux para realizar o controle do computador Zybo Zynq 700. No sistema operacional é possível instalar um compilador de linguagem C e a biblioteca OpenCV.

O computador Zybo Zynq 7000 possui em um mesmo chip um processador ARM, que é o núcleo do Sistema de Processamento (PS, do inglês, Processing System), e uma FPGA Artix 7, que o núcleo da Lógica Programável (PL, do inglês, Programmable Logic) cuja as principais características são (DIGILENT, 2016):

- Fabricante: Xilinx Inc;
- Número de elementos lógicos: 12800;
- Número de CLB: 2000;
- Total de RAM disponível: 720 KB.

A principal característica do computador Zybo é o forte acoplamento e a rápida comunicação entre o PS e o PL (AHMADINIA, 2016). O processador ARM pode ser conectado diretamente à FPGA por meio do do barramento "*Advanced Extensible Interface*" (AXI) (AHMADINIA, 2016). A Figura 3.10 exibe como é realizada a comunicação de dados entre processador ARM e FPGA no computador Zybo.

Figura 3.10 - Comunicação entre o processador ARM e a FPGA no computador Zybo por meio do barramento AXI

Fonte: Adaptado de Ahmadinia (2016).

O barramento AXI opera em uma frequência de 450 MHz e um "*throughput*" de comunicação CPU-FPGA de 1200 MB por segundo. O tamanho da palavra na comunicação entre processador ARM e FPGA e de 32 bits.

A empresa Digilent, desenvolvedora da placa Zybo Zynq 7000, criou uma API, em linguagem C, para tornar transparante a comunicação entre o processador AMR e a FPGA (DIGILENT, 2016).

4 FORMAÇÃO DE IMAGEM EM CÂMERA ESTENOPEICA

As câmeras são dispositivos para a aquisição de imagens com aplicação em diversas áreas, entre elas, a robótica (PETER, 2011). Uma câmera representa uma relação entre o espaço tridimensional e um espaço bidimensional, denominado imagem. Esta relação consiste na construção de um modelo geométrico que mapeia pontos do objeto no plano da imagem (PEDRINI; SCHARTZ, 2008).

A câmera realiza uma transformação de perspectiva do espaço tridimensional para o plano da imagem. Transformações de perspectiva, também denominadas homografias, mapeiam cada ponto dos objetos que compõem a cena, no espaço com três dimensões, para o plano da imagem (PEDRINI; SCHARTZ, 2008).

Vários modelos de câmera tem sido empregados para aproximar sistemas ópticos, por exemplo, as câmeras esféricas e as câmeras estenopeica. O modelo de câmera mais simples é modelo de câmera estenopeica, do qual são derivados modelos mais complexos (HARTLEY; ZISSERMAN, 2004).

Nas próximas seções do capítulo é realizada a fundamentação teórica sobre câmeras estenopeica, informações sobre outros modelos de câmera, como a câmera esférica podem ser obtidas em Nayar (1997).

4.1 CÂMERA ESTENOPEICA

O modelo de projeção perspectiva utilizado nesta pesquisa, para a formação de imagens por radiação eletromagnética com comprimento de onda no visível, é o de câmera estenopeica (*"pinhole camera"*), exibido na Figura 4.1. Este modelo de câmera define o mapeamento de um ponto do mundo real, em três dimensões, para o plano da imagem, em duas dimensões. O modelo de câmera estenopeica é uma simplificação da câmera real (NISHITANI, 2015).

Figura 4.1 - Esboço da câmera estenopeica

Fonte: Adaptado de MathWorks (2017).

A imagem em uma câmera estenopeica é formada pela entrada de feixes de luz através de uma abertura e a projeção desses feixes em uma tela, denominada plano da imagem. O modelo de projeção perspectiva da câmera estenopeica é exibido na Figura 4.2.

Figura 4.2 - Modelo de projeção perspectiva da câmera estenopeica.

Fonte: Adaptado de Nishitani (2015).

Pontos de um objeto, pertencentes à cena, correspondem-se com pontos no plano da imagem por meio da composição de transformações aplicadas sobre quatro sistemas de coordenadas, estes sistemas são: sistema de coordenadas do mundo global; sistema de coordenadas da câmera estenopeica; sistema de coordenadas da imagem; sistema de coordenadas em pixel.

O sistema de coordenada do mundo real é um sistema de referência com três dimensões, que descreve objetos na cena. O ponto **A**, na Figura 4.2 é um ponto pertencente à este sistema com coordenadas denotadas por (X, Y, Z).

O sistema de de coordenadas da câmera também é um sistema com três dimensões com origem no ponto **C**, exibido na Figura 4.2. Este ponto é denominado centro de projeção, ou centro óptico, que é a origem do centro de coordenadas da câmera (PEDRINI; SCHARTZ, 2008). As coordenadas do sistema da câmera são denotadas por (X_c, Y_c, Z_c). O eixo que passa pela reta $Z_c = 0$ é denominado eixo principal ou eixo óptico.

O sistema de coordenadas da imagem é um sistema com duas dimensões, com origem no ponto **P**, denominado ponto principal. O ponto **P** é colinear ao ponto **C**. A distância entre esses dois pontos é denominada distância focal e tem valor f, como exibido na Figura 4.2. A imagem é formada no plano $Z_c = f$, denominado plano da imagem. O ponto P é a intersecção entre o plano da imagem e o eixo principal. As coordenadas no sistema da imagem são denotados por (X_i, Y_i)

O sistema de coordenadas em pixel possui duas dimensões, suas coordenadas, denotadas por (X_{pi}, Y_{pi}) e são expressas em valores de pixel da imagem. Nesta pesquisa, a origem deste sistema é o canto superior esquerdo da imagem.

4.1.1 PARÂMETROS EXTRINSECOS DA CÂMERA ESTENOPEICA

A composição de transformações necessárias para mapear um ponto do objeto, pertencente à cena, em uma imagem é exibida na Figura 4.3.

Figura 4.3 - Transformações necessárias para mapear um ponto do objeto, no sistema de coordenadas do mundo real, para um ponto na imagem, no sistema de coordenadas em pixel.

Fonte: Adaptado de Pedrini e Schartz (2008).

Na Figura 4.3, TR_1 é a transformação do sistema de coordenada do mundo real para o sistema de coordenada da câmera; Tr_2 é a transformação do sistema de coordenada da câmera para o sistema de coordenada da imagem; Tr_3 é a transformação do sistema de coordenada da imagem para o sistema de coordenadas em pixels.

As transformações de corpos rígidos, uma combinação de translações e rotações, podem estabelecer a relação entre o sistema de coordenadas do mundo real e o sistema da câmera(PEDRINI; SCHARTZ, 2008). O primeiro passo é deslocar a origem do sistema de coordenadas do mundo real para origem do sistema de coordenadas da câmera, este deslocamento é obtido pela matriz de translação T ,em coordenadas homogêneas, exibida na Equação 4.1.

$$T = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.1}$$

onde t_x, t_y e t_z são os valores do deslocamento nos eixos X, Y e Z, respectivamente.

Após a translação do sistema de coordenadas do mundo real para o sistema da câmera, ocorre a transformação de rotação entre os dois sistemas para que eles coincidam. A transformação de rotação é obtida pela combinação de rotação em cada um dos eixos do sistema: rotação em X (Equação 4.2); rotação em Y (Equação 4.3); e rotação em Z (Equação 4.4).

$$R(\phi)_x = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & cos(\phi) & -sen(\phi) & 0 \\ 0 & sen(\phi) & cos(\phi) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.2}$$

$$R(\theta)_y = \begin{bmatrix} cos(\theta) & 0 & -sen(\theta) & 0 \\ 0 & 1 & 0 & 0 \\ sen(\theta) & 0 & cos(\theta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.3}$$

$$R(\psi)_z = \begin{bmatrix} cos(\psi) & sen(\psi) & 0 & 0 \\ -sen(\psi) & cos(\psi) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.4}$$

Nas Equações (em coordenadas homogêneas) 4.2, 4.3 e 4.4, ângulos positivos são aqueles medidos no sentido anti-horário, considerando a origem vista pela parte positiva dos eixos.

A combinação das três rotações em torno de cada um dos eixos Z, Y, X, nesta ordem, é expressa pela Equação 4.5

$$R = R_x(\phi)R_y(\theta)R_z(\psi) \tag{4.5}$$

Desta forma, o valor de R é definido pela Equação 4.6, em coordenada homogênea.

$$R = \begin{bmatrix} r_1 & r_2 & r_3 & 0 \\ r_4 & r_5 & r_6 & 0 \\ r_7 & r_8 & r_9 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.6}$$

onde

$$\begin{aligned} r_1 &= cos(\theta)cos(\psi) \\ r_2 &= cos(\theta)sin(\psi) \\ r_3 &= -sen(\theta) \end{aligned} \tag{4.7}$$

$$\begin{aligned} r_4 &= -cos(\phi)sen(\psi) + sen(\phi)sen(\theta)cos(\psi) \\ r_5 &= cos(\phi)cos(\psi) + sen(\phi)sen(\theta)sen(\psi) \\ r_6 &= sen(\phi)cos(\theta) \end{aligned} \tag{4.8}$$

$$\begin{aligned} r_7 &= sen(\phi)sen(\psi) + cos(\phi)sen(\theta)cos(\psi) \\ r_8 &= -sen(\phi)cos(\psi) + cos(\phi)sen(\theta)sen(\psi) \\ r_9 &= cos(\phi)cos(\theta) \end{aligned} \tag{4.9}$$

A rotação nos eixos X, Y e Z e a translação podem ser combinadas, como mostrada na Equação 4.10.

$$Tr_1 = A_c = R_x(\phi)R_y(\theta)R_z(\psi)TA = \begin{bmatrix} & R & & T \\ 0 & 0 & 0 & 1 \end{bmatrix} A = EA \tag{4.10}$$

onde A_C é o ponto transformado para o sistema de coordenadas da câmera, A é o ponto no sistema de coordenadas do mundo real e E é a matriz de parâmetros extrínsecos da câmera, responsável pela transformação do sistema de coordenada do mundo real para o sistema de coordenada da câmera. O ponto A está em coordenadas homogêneas, sendo assim sua dimensão é igual 4×1, a matriz E possui dimensão 4×4, desta forma, o ponto A_c possui dimensão 4×1.

4.1.2 PARÂMETROS INTRÍNSECOS DA CÂMERA ESTENOPEICA

A transformação do sistemas de coordenadas da câmera estenopeica para o sistema de coordenadas da imagem é uma transformação de perspectiva (HARTLEY; ZISSERMAN, 2004). Por semelhança de triângulos (Figura 4.2), após o mapeamento do ponto A, do sistema de coordenadas do mundo real para o sistema de coordenadas da câmera, é possível obter as relações descritas nas Equações 4.11 e 4.12, que descrevem o mapeamento do sistema de coordenada tridimensional da câmera para o sistema de coordenada bidimensional da imagem (HARTLEY; ZISSERMAN, 2004).

$$X_i = (fX_c)/Z_c \tag{4.11}$$

$$Y_i = (fY_c)/Z_c \tag{4.12}$$

onde f é a distância focal da câmera estenopeica, X_c, Y_c e Z_c são as coordenadas do ponto A no sistema de coordenadas da câmera e X_i, Y_i são as coordenadas no sistema de coordenadas da imagem.

A transformação de perspectiva, exibida nas Equações 4.11 e 4.12, pode ser escrita na forma matricial (Equação 4.13).

$$Tr_2 = A_i = \begin{bmatrix} X_i \\ Y_i \\ W \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} . \begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} \tag{4.13}$$

O mapeamento de um ponto do mundo real para o plano da imagem deve considerar:

- a unidade do sistema de coordenadas da câmera, normalmente em metros, e a unidade do sistema de coordenadas da imagem, normalmente em pixels.
- a distância entre linhas e colunas da imagem pode ser diferente;
- linhas e colunas podem não estar dispostas perpendicularmente entre si;
- a origem do centro de coordenadas da imagem está no ponto principal P (Figura 4.2), contudo, a origem do sistema de coordenadas em pixel está no canto superior esquerdo da imagem.
- distorções na câmera;

A Equação 4.14 representa a transformação do sistema de coordena bidimensional da imagem para o sistema de coordenada bidimensional em pixels, considerando os fatores citados anteriormente.

$$Tr_3 = A_{pi} = \begin{bmatrix} x_{pi} \\ y_{pi} \\ w \end{bmatrix} = \begin{bmatrix} s_x & tan(\alpha) & p_x \\ 0 & s_y & p_y \\ 0 & 0 & 1 \end{bmatrix} . \begin{bmatrix} X_i \\ Y_i \\ 1 \end{bmatrix} \tag{4.14}$$

onde s_x e s_y são o número de pixels por unidade de comprimento, nas direções do eixo X_i (direção do eixo X no sistema de coordenadas da imagem) e Y_i (direção do eixo Y no sistema de coordenadas da imagem), respectivamente. Normalmente, os pixels em uma imagem são quadrados, assim, $s_x = s_y$. Os valores de p_x e p_y correspondem aos valores do ponto principal P no sistema de coordenadas em pixels, normalmente o ponto principal ocupa o centro da imagem e esses valores são iguais à metade das dimensões da imagem. O parâmetro α é o angulo formado entre as linhas e as colunas do pixel, frequentemente igual a $\frac{\pi}{2}$, quando isto ocorre o valor de $tan(\alpha)$ deve ser substituído por zero (PEDRINI; SCHARTZ, 2008; MATHWORKS, 2017).

A Equações 4.13 e 4.14 podem ser combinadas, resultando na Equação 4.15.

$$K = \begin{bmatrix} f.s_x & f.tan(\alpha) & p_x & 0 \\ 0 & f.s_y & p_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} s_x & tan(\alpha) & p_x \\ 0 & s_y & p_y \\ 0 & 0 & 1 \end{bmatrix} . \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{4.15}$$

onde K é denominada a matriz de parâmetros intrínsecos da câmera estenopeica.

O produto da matriz de parâmetros extrínsecos pela matriz de parâmetros intrínsecos define a matriz H (homografia), que determina o mapeamento de um ponto do mundo real para o plano da imagem (PEDRINI; SCHARTZ, 2008; NISHITANI, 2015).

$$H = KE \tag{4.16}$$

$$A_{pi} = HA = KEA \tag{4.17}$$

Na Equação 4.16, H é a homografia. Na Equação 4.17, A_{pi} é um ponto no sistema de coordenadas em pixels e A é um ponto no sistema de coordenadas do mundo real.

A matriz de parâmetros extrínsecos e a matriz de parâmetros intrínsecos podem ser definidas por um processo de calibração da câmera, que determina as características ópticas (matriz de parâmetros intrínsecos) e a posição tridimensional e orientação da câmera estenopeica (matriz de parâmetros extrínsecos) por meio da analise de informações disponíveis por um conjunto de imagens da mesmo sensor (PEDRONI, 2008).

Para uma correta transformação de pontos no sistemas de coordenadas do mundo real para o sistema de coordenadas em pixels, as distorções causadas pelas lentes devem ser consideradas e corrigidas. As principais distorções que devem ser consideradas são: a distorção radial, que altera o perpendicularismo das linhas e colunas da imagem; e a distorção de descentralização, que desloca o centro das lentes da câmera em relação ao eixo principal (PEDRINI; SCHARTZ, 2008; CONTE; DOHERTY, 2008). Nesta pesquisa, não há necessidade de correções de distorção radial e distorção de descentralização para as imagens da câmera estenopeica, uma vez que a câmara utilizada tem uma distância focal relativamente grande (aproximadamente 45 graus de ângulo de visão horizontal), portanto, as distorções nas imagens são consideradas pequenas para a precisão necessária do sistema de estimação da posição do VANT (CONTE; DOHERTY, 2008). Além disso, o processo de correção das distorções das

imagens pode impactar de maneira negativa o desempenho do algoritmo de estimação da posição do VANT por processamento das imagens aéreas (CHUNHUI et al., 2014).

5 FORMAÇÃO DE IMAGENS EM SENSORES LiDAR

As câmeras de vídeo, sensores imageadores com modelo de projeção da câmera estenopeica e que formam a imagem da cena a partir da radiação eletromagnética com comprimento de onda no visível, são os sensores normalmente aplicados para obtenção de imagens de uma região de interesse (HARTLEY; ZISSERMAN, 2004). Recentemente, a tecnologia de escaneamento a laser vem sendo aplicada para o mesmo objetivo (MCMANUS et al., 2013).

Apesar de ser uma tecnologia nova, o sistema de escaneamento a laser *Light Detection and Ranging* (LiDAR) vem sendo aplicado com sucesso em áreas como: mapeamento topográfico aerotransportado (WEST; WIGGINS, 2000); sensoriamento remoto (ANTONARAKIS et al., 2008) e levantamento florestais (GORGENS et al., 2014)

A tecnologia LiDAR apresenta-se com uma das mais promissoras para realizar a modelagem digital de terreno, sendo aplicada com sucesso na extração de Modelos Digitais de Superfícies (MDS) e extração de Modelos Digitais de Elevação (MDE) (JENSEN, 2009)

O LiDAR é uma tecnologia que permite a obtenção de informações tridimensionais sobre pontos na superfície do terreno com grande precisão, por este motivo, é aplicada para a criação de MDS e MDE (JENSEN, 2009). Outra característica bastante importante dos sistemas LiDAR é conseguir captar o valor da intensidade da reflectância de objetos dispostos no terreno, por essa razão, vem sendo utilizado para a classificação de padrões no solo (ANTONARAKIS et al., 2008). O sistema LiDAR é capaz de armazenar, ao mesmo tempo, informações sobre a distância do objeto de interesse bem como sua reflectância (JENSEN, 2009) (ANTONARAKIS et al., 2008).

Mais recentemente, a aplicação de sensores LiDAR surge como uma alternativa no desenvolvimento de sistema de navegação por processamento de imagens, para robôs móveis. Segundo (MCMANUS et al., 2013), os sensores LiDAR podem ser utilizados para substituir o sistema de câmeras de vídeo de robôs móveis, para sua navegação em ambiente externo, pois o ponto fraco das câmeras de vídeo é sua dependência da luz do ambiente. Pequenas variações na luminosidade do ambiente externo podem ser fonte de erro para o sistema de navegação por processamento de imagens, que utilizam a câmera de vídeo como sensor imageador (MCMANUS et al., 2013).

O sensor LiDAR apresenta-se com uma alternativa para a substituição do sistema de câmeras de vídeo para capturar imagens em ambientes com baixa luminosidade ou

sem iluminação, pois as imagens de intensidade podem substituir as imagens em tons de cinza oriundas das câmeras de vídeo. Perspectivas futuras apontam que os sensores LiDAR serão utilizados para o desenvolvimento de sistemas de processamento de imagens em veículos espaciais (MCMANUS et al., 2013).

Apesar do potencial para substituir câmeras de vídeo, os sensores LiDAR apresentam algumas limitações que podem dificultar sua aplicação. Entre essas limitações, pode-se citar: o sensor LiDAR possui uma baixa altitude de operação, a tecnologia LiDAR pode não funcionar em altitudes superiores a 2000 metros; a tecnologia pode não funcionar em áreas ou situações onde há grande reflexão; sensores LiDAR são muito caros (NAZEMZADEH et al., 2017).

Sensores LiDARs de alta taxa de quadros "*high-framerate*" são capazes de gerar imagens de intensidade ou MDS da cena com frequência semelhante a de câmeras de vídeo, estes sensores são denominados de câmera LiDAR. De modo geral, as câmeras LiDAR possuem um modelo de projeção esférico e fornecem a imagem de intensidade da região de interesse com as distorções já corrigidas (MCMANUS et al., 2013). Informações sobre o modelo de projeção esférico podem ser obtidas em (NAYAR, 1997).

Geralmente, pulsos laser no comprimento do infra-vermelho próximo (comprimento de onda entre $0.75\mu m$ a $1.4\mu m$) são aplicados para gerar as imagens de intensidade LiDAR (GORGENS et al., 2014). O MDS pode ser obtido por meio de um pulso no infra-vermelho próximo ou pela combinação de dois pulsos com comprimento de onda diferentes (WEST; WIGGINS, 2000).

Na pesquisa desenvolvida por McManus et al. (2013), o autor compara o processo de odometria visual para estimar a trajetória realizada por um robô, utilizando uma câmera estenopeica e uma câmera LiDAR para a captura de imagens dos alvos na cena. A câmera LiDAR captura imagens de intensidade dos alvos na cena. O estudo conclui que é possível realizar a odometria visual com imagens de intensidade oriundas de uma câmera LiDAR com alta taxa de quadros. Além disso, o processo de detecção e descrição dos pontos de interesse se mantém estável, mesmo quando há variação da luminosidade (ou sem luminosidade) no ambiente. Na pesquisa de (MCMANUS et al., 2013), emprega-se uma câmera LiDAR "*Autonosys*", que gera imagens de intensidade ou MDS a uma taxa de $2H_z$ com 480×360 pixels.

Outra característica importante de alguns sensores LiDAR é a capacidade de realizar o imageamento de regiões cobertas por água, por exemplo o oceano, possibilitando

a geração de imagens de intensidade ou MDS do da cobertura do fundo desta região (JENSEN, 2009).

5.1 FUNCIONAMENTO DO SENSOR LiDAR

O sensor LiDAR é formado, basicamente, por um sistema de controle, por um espelho e por um transmissor/receptor (GORGENS et al., 2014). Seu funcionamento consiste na emissão de um pulso laser, com elevada frequência de repetição (superior a cem mil pulsos por segundo), que é direcionado ao solo pelo espelho do sensor (JENSEN, 2009; GORGENS et al., 2014). O sistema de controle do sensor é responsável pelo posicionamento do espelho para que o pulso de laser atinja o alvo desejável (GORGENS et al., 2014).

O tempo de retorno do pulso à fonte (sensor) pode ser medido, o que permite realizar uma estimativa da distância do alvo ao sensor. A Equação 5.1 determina o tempo percurso de um pulso de luz (JENSEN, 2009; GORGENS et al., 2014).

$$t = 2.\frac{d}{c} \tag{5.1}$$

onde t é o tempo do percurso do pulso (ida e volta ao sensor), d é a distância entre o sensor e o objeto e c é o valor da velocidade do pulso, que é igual a velocidade de propagação da luz no vácuo, que é $3 \times 10^8 m/s$.

Rearranjando a Equação 5.1 é possível obter a distância d do sensor ao objeto, Equação 5.2.

$$d = \frac{1}{2}.t.c \tag{5.2}$$

O espelho do sistema LiDAR direciona os pulsos para que estes atinjam o terreno com um determinado diâmetro (D_c), denominado "*footprint*", que é calculado pela Equação 5.3

$$D_c = D + 2.h.tan(\frac{\gamma}{2}) \tag{5.3}$$

onde D é o diâmetro do pulso laser ao sair do sensor, h é a altura (ou distância) do sensor ao objeto e γ é a divergência do laser.

Um dado importante, que pode ser obtido pelo sensor LiDAR, é a intensidade registrada, que corresponde a energia emitida pelo sensor e que foi refletida pelo objeto e retornou ao sensor. O valor da intensidade pode mudar de um equipamento para o outro, por este motivo, para a o desenvolvimento de um sistema de navegação por processamento de imagens de intensidade, o mesmo sensor LiDAR deve ser aplicado (MCMANUS et al., 2013; GORGENS et al., 2014). A quantidade de energia que é refletida por um objeto e recebida pelo sensor LiDAR é determinada pela Equação 5.4

$$E_r = \rho.\frac{M^2.D_c}{\pi.h^2}.E_t \tag{5.4}$$

onde D_c é a medida do "*footprint*", h é a altura (ou distância) do sensor ao objeto, E_t é a energia emitida pelo equipamento LiDAR, M é o coeficiente de transmissão atmosférica, e ρ é o coeficiente de reflectância, que é uma propriedade de cada objeto e é diferente para cada comprimento de onda (GORGENS et al., 2014).

A Figura 5.1 mostra o funcionamento de um sensor LIDAR aerotransportado, onde o solo é escaneado pelo pulso de laser emitido pelo sensor.

Figura 5.1 - Funcionamento de um sensor LiDAR aerotransportado.

Fonte: Adaptado de Guenther et al. (2000).

O sensor LiDAR possui sua própria fonte de luz (pulso de laser), desta forma, é considerado um método direto de captura de dados, ou seja, um sensor ativo (JENSEN, 2009). Os sensores ativos possuem vantagem em comparação aos sensores passivos (sensores ópticos), pois não dependem do sol como fonte de iluminação, portanto, a análise dos dados oriundos do sensor LiDAR não é comprometida pela presença de sombras, presença de nuvens ou pela falta de iluminação no ambiente (JENSEN, 2009).

A medida de distância dos pontos de uma determinada região ao sensor e o valor da quantidade de energia refletida formam um conjunto de dados de elevação e intensidade do terreno. Essas medições são sistematicamente distribuídas no tempo, acompanhando a trajetória de voo, criando uma nuvem de pontos. A partir desta nuvem de pontos são criados, por mapeamento, o MDS, o MDE e as imagens de intensidade. Durante o mapeamento, valores inteiros de coordenadas da nuvem de pontos podem originar valores fracionários na imagem. Portanto, para atribuir valores de intensidade ou distâncias que não se encontram na nova grade da imagem, um método de interpolação deve ser aplicado (MCMANUS et al., 2013; GORGENS et al., 2014) [1].

Existem diversos métodos de interpolação que podem ser aplicados sobre nuvem de pontos provenientes de um sensor LiDAR , entre eles, a interpolação bicúbica. O emprego da interpolação bicúbica deve-se ao fato desta não causar bordas serrilhadas, como na interpolação pelo vizinho mais próximo, e não causar borramento, como ocorre na interpolação bilinear (PEDRINI; SCHARTZ, 2008; MCMANUS et al., 2013).

A interpolação bicúbica aplica uma vizinhança 4×4, ao redor do ponto de interesse, para calcular seu valor de distância ou intensidade (PEDRINI; SCHARTZ, 2008). A função "*B-spline cúbica*", Equação 5.5, é aplicada para determinar as intensidades ou distâncias na imagem interpolada.

$$f(x', y') = \sum_{m=-1}^{2} \sum_{n=-1}^{2} f(x+m, y+n) R(m - dx) R(dy - n) \tag{5.5}$$

onde dx e dy são os valores distância do ponto (x, y) para o ponto (x', y'), na horizontal e vertical, respectivamente, O valor de R é definido por

[1] Informações sobre métodos de interpolação sobre nuvem de pontos LiDAR podem ser obtidas em (ASHRAF et al., 2017)

$$R(s) = \frac{1}{6}.(P(s+2)^2 - 4P(s+1)^3 + 6P(s)^3 - 4P(s-1)^3) \tag{5.6}$$

o valor de P é definido por:

$$P(t) = \begin{cases} 0, t \leqslant 0 \\ t, t > 0 \end{cases} \tag{5.7}$$

O pulso de laser proveniente do sensor LiDAR tem a capacidade de penetração em pequenas aberturas do dossel de árvores, conseguindo obter dados do solo abaixo das copas de uma floresta (GORGENS et al., 2014). Outra característica do pulso de laser dos sensores LiDAR é a capacidade de penetração na água. Pulsos de lasers que utilizam a luz verde-azul, com comprimento de pulso centrado em $0,532\mu m$, têm capacidade de penetração na água, e são aplicados para o mapeamento batimétrico (MIKHAIL et al., 2001).

5.2 MAPEAMENTO BATIMÉTRICO POR SENSORES LiDAR

A batimetria consiste no processo da medição da profundidade de, lagos, rios e regiões oceânicas por meio de ecobatímetros, ou, como feito mais recentemente, por meio de sensores LiDAR. As informações sobre a profundidade de regiões de mares e água doce pode ser determinada precisamente por sensoriamento remoto por LIDAR (JENSEN, 2009).

A batimetria por sensor LiDAR obedece aos seguintes passos (JENSEN, 2009):

- O sistema envia simultaneamente dois pulso de laser diferentes;

  Um pulso de luz no infravermelho próximo, comprimento de onda de $1,064\mu m$. Grande parte deste pulso é absorvida pela superfície da água, mas uma quantidade é refletida e retorna ao receptor.

  Um pulso de luz no verde-azul, comprimento de onda de $0,532\mu m$. Uma parte também é absorvida pela água, mas uma quantidade penetra a coluna de água, atinge o fundo e é refletida ao receptor.

- A profundidade é determinada a partir da diferença de tempo entre o retorno do pulso infravermelho próximo, refletido pela superfície da água, e do retorno do pulso verde-azul, refletido pelo terreno ao fundo da coluna de água. É importante realizar também a diferença entre a velocidade de

propagação da luz no ar e na água (GUENTHER et al., 2000).

- Um sistema GPS e um sistema INS fazem o georreferenciamento dos pontos dos quais foi calculada a profundidade;

A luz no verde-azul é selecionada, pois é capaz de penetrar colunas de águas, com atenuação mínima, devido a seu comprimento de onda que varia de $0.44\mu m$ a $0.54\mu m$. A profundidade máxima de penetração é de aproximadamente 60 metros, isto ocorre devido ao espalhamento, a absorção e a refração que luz sofre ao atravessar a coluna de água (JENSEN, 2009).

A Figura 5.2 mostra o mapeamento batimétrico realizado por um sensor LiDAR.

Figura 5.2 - Mapeamento batimétrico realizado por um sensor LiDAR.

Fonte: Adaptado de Guenther et al. (2000).

A batimetria por sensores LiDAR possui as seguintes vantagens em relação ao método tradicional de mapeamento batimétrico, realizado por Navegação e Medição de Distância pelo Som (SONAR, do inglês, Sound Navigation and Ranging) (JENSEN, 2009):

- Funciona muito bem em água rasa, onde a embarcação portadora do sistema SONAR pode encalhar (GUENTHER et al., 2000);
- Pode ser usado em áreas costeiras perigosas (com formação rochosas ou ondas altas) (WEST; WIGGINS, 2000);
- É móvel e possui rápida resposta devido ao fato de poder ser aerotransportado (GUENTHER et al., 2000);

- É capaz de investigar uma área extensa em menos tempo, proporcionando vantagens econômicas (GUENTHER et al., 2000) (WEST; WIGGINS, 2000).

Devido a capacidade de penetração na água da luz no comprimento de onda do verde-azul, sensores LiDAR que emitem pulsos de laser neste comprimento de onda estão sendo aplicados para realizar o mapeamento da superfície (gerar o MDS) da cobertura do fundo de regiões oceânicas (GUENTHER et al., 2000).

6 ESTIMAÇÃO DA POSIÇÃO DO VANT POR REGISTRO AUTOMÁTICO DE IMAGENS

Registro de imagens é o processo de sobreposição de duas imagens de uma mesma cena, porém, capturadas em instantes de tempo diferentes ou sob um diferente ponto de vista, utilizando diferentes sensores (PEDRINI; SCHARTZ, 2008). O registro de imagens tem sido aplicado, com diferentes propósitos, em diversas áreas do conhecimento, entre elas (ZITOVA; FLUSSER, 2003):

- medicina - a combinação de imagens obtidas por diferentes técnicas de aquisição, como tomografia computadorizada e ressonância magnética, geram informações mais completas sobre o paciente;
- sensoriamento remoto - para detectar a ocorrência de mudanças no terreno e acompanhar o processo de utilização do solo;
- visão computacional - recuperação de informação tridimensional por meio da visão estereoscópica.

Há na literatura duas principais abordagens que podem ser utilizadas para o registro de imagens (CONTE; DOHERTY, 2008): registro baseado na correlação de imagens e o registro de imagens baseado no casamento de padrões. O registro baseado na correlação faz a convolução entre duas imagens e então é adotado um critério de similaridade para determinar qual a melhor sobreposição entre elas. O registro baseado no casamento de padrões localiza um conjunto de características representativas (por exemplo, algum marco, do inglês, "*landmark*") em ambas as imagens, e o registro é obtido pela correspondência entre os conjuntos (BROWN, 1992). Ambos os métodos apresentam vantagens e desvantagens.

De modo geral, métodos baseados no casamento de padrões não operam diretamente sobre o valor de intensidade do pixel. Os padrões são normalmente uma informação representada por um modelo geométrico. Portanto, este método é adequado para situações onde há pontos de referência distintos no solo e que não são afetados por mudanças sazonais (por exemplo, estradas e grandes construções). A principal vantagem deste método é que mesmo um pequeno ponto de referência pode tornar uma grande porção do terreno única. Esta característica é a principal diferença para os métodos baseados em correlação, onde pequenos detalhes em uma imagem podem influenciar o resultado final da similaridade. Por outo lado, o método baseado no casamento de padrões pode ser apenas aplicado em locais onde há pontos de

referência distintos e sua performance é prejudicada pelo processo de localização dos pontos de interesse, descrição e correspondência entre esses pontos, desta forma, o método não se mostra adequado para aplicações de tempo real (BROWN, 1992; ZITOVA; FLUSSER, 2003; CONTE; DOHERTY, 2008).

Métodos baseados na correlação entre imagens podem ser aplicados em áreas onde não há pontos de referência no solo e, além disso, são mais eficientes que os métodos de registro por casamento de padrões, sendo mais adequados para aplicações de tempo-real (BROWN, 1992; ZITOVA; FLUSSER, 2003; CONTE; DOHERTY, 2008).

Recentemente, métodos baseados na correlação entre imagens tem sido aplicados para a estimação da posição geográfica do VANT. Nas pesquisas desenvolvidas por Conte e Doherty (2008), Goltz et al. (2011) e Silva et al. (2015), que realizam a estimação da posição geográfica do VANT por métodos baseados na correlação entre imagens, os autores denominam esta técnica como registro automático de imagens.

No contexto desta pesquisa, um método de registro automático de imagens é aplicado para estimar o posicionamento geográfico do VANT. Neste trabalho, o algoritmo de registro automático de imagens é composto por uma sequência de transformações sobre as imagens, por exemplo, transformações de escala e rotação. Estas transformações permitem a correta sobreposição entre as imagens. A Figura 6.1 exibe o algoritmo de registro automático de imagens utilizado.

Figura 6.1 - Método de registro automático de imagens aplicado para estimar o posicionamento geográfico do VANT.

Fonte: Produção do autor.

O método de registro automático de imagens, apresentado na Figura 6.1, aplicado para estimar a posição geográfica de um VANT é composto dos seguintes passos:

- Primeiramente, uma imagem aérea, capturada pelo sensor imageador (câmera de vídeo ou sensor LiDAR) do VANT, e a imagem georreferenciada (imagem de satélite na banda do visível ou proveniente de sensor LiDAR),

da região sobrevoada, são convertidas para tons de cinza. Esta transformação permite o aumento da eficiência computacional dos algoritmos de filtragem e extração de borda, pois os cálculos dos algoritmos são realizados em apenas uma banda;

- As imagens monocromáticas são filtradas para que ruídos indesejáveis, que possam afetar o resultado do registro automático de imagens, sejam eliminados;

- A imagem monocromática aérea passa por um ajuste de escala e rotação, em relação a imagem georreferenciada, para melhorar o resultado final do registro de imagens. Durante este ajuste, a imagem aérea é transformada para a mesma escala e rotação da imagem georreferenciada. O ajuste de escala é obtido utilizando informações do altímetro da aeronave e resolução da imagem georreferenciada. Já o ajuste de rotação é obtido utilizando informações da bússola magnética presente no VANT;

- Seleção da área de correlação, que é uma subimagem da imagem georreferenciada monocromática. Esta subimagem é empregada no processo de registro automático para evitar áreas de baixa probabilidade da posição do VANT e para melhorar o desempenho computacional do método. A área de correlação é selecionada utilizando a melhor informação sobre o posicionamento do VANT, fornecida pelo INS (CONTE; DOHERTY, 2008; BRAGA et al., 2016). O tamanho final da área de correlação, subimagem da imagem georreferenciada monocromática, depende do tamanho da imagem aérea, após o ajuste de escala, e da incerteza da informação do INS sobre posicionamento do VANT. O cálculo do tamanho a área de correlação é dado por:

$$\begin{pmatrix} T_L \\ T_A \end{pmatrix} = \begin{pmatrix} I_L + inc \\ I_A + inc \end{pmatrix} \tag{6.1}$$

onde T_L e T_A são a largura e altura da área de correlação, respectivamente. I_L e I_A são, respectivamente, a largura e altura da imagem aérea (capturada pelo VANT) após o ajuste de escala. O valor de inc é a incerteza do posicionamento do VANT em pixels. O valor de inc, obtido em pixels é determinado por:

$$inc = \frac{inc_{metros}}{I_{res}} \tag{6.2}$$

onde inc_{metros} é o valor da incerteza do posicionamento do VANT, fornecido pelo INS, em metros e I_{res} é a resolução da imagem georreferenciada em $\frac{metros}{pixel}$. O valor de inc_{metros} foi determinado por Conte e Doherty (2008) e varia de 10 a 80 metros. O valor de 10 metros corresponde ao valor da incerteza no momento da falha do GNSS, este valor é igual ao erro do sistema GNSS. O valor de 80 metros é o maior valor possível para a incerteza do INS e que permite uma navegação autônoma segura da aeronave sobre áreas urbanas, este valor foi obtido após diversas medições e considera o tamanho e velocidade do VANT (CONTE; DOHERTY, 2008; BRAGA et al., 2016);

- O passo seguinte do método de registro automático de imagens é a detecção de bordas da imagem aérea e da área de correlação (sub-imagem da imagem georreferenciada). O processo de detecção de bordas é aplicado para evitar que variações na iluminação do ambiente interfiram no resultado final do registro de imagens (GOLTZ et al., 2011; SILVA et al., 2015; CONTE; DOHERTY, 2008; BRAGA et al., 2016). Além disso, o registro automático de imagens com a bordas detectadas permite que estas sejam provenientes de diferentes sensores, como na pesquisa desenvolvida por Silva et al. (2015), que realiza o registro entre imagens aéreas, capturadas por sensor infravermelho termal, e imagens de satélite, capturadas por sensor óptico na banda do visível.

  Nesta pesquisa de doutorado, a detecção de bordas é realizada por uma RNA com Perceptrons organizados em Múltiplas Camadas (MLP, do inglês, Multilayer Perceptron) programada em FPGA, utilizando uma estratégia de "*Look-up Table*". Os operadores Canny e Sobel são utilizados para avaliar o resultado de registro de automático de imagens com as bordas detectadas pela a MLP programada em FPGA. A arquitetura da MLP aplicada para a detecção de bordas é determinada pelo Algoritmo de Colisão de Múltiplas Partículas (MPCA, do inglês, Multi-Particle Collision Algorithm);

- O passo final do algoritmo de registro é a medida de similaridade entre as imagens com as bordas detectadas. Na literatura, há diversos métodos para realizar esta medida de similaridade, entre eles: correlação cruzada, correlação cruzada normalizada e a informação mútua (MONTEIRO et al., 2006). Nesta pesquisa, empregou-se a correlação cruzada por já ter apresentado resultados promissores (CONTE; DOHERTY, 2008; GOLTZ et al., 2011;

BRAGA et al., 2016; SILVA et al., 2015). Por meio da correlação cruzada é possível localizar na área de correlação a posição central da imagem aérea. Considerando que o VANT ocupa a posição central na imagem capturada pelo seu sensor (câmera de vídeo ou sensor LiDAR), obtém-se a posição do VANT. Pelo fato de área de correlação ser uma sub-imagem da imagem georreferenciada é possível obter as coordenadas geográficas (latitude e longitude) do posicionamento do VANT.

O método proposto nesta tese de doutorado para realizar o registro automático de imagens é baseado no algoritmo desenvolvido por Conte e Doherty (2008). Resultados aplicando o método proposto nesta tese são encontrados em Silva et al. (2015), Braga et al. (2016).

Nas próximas seções é realizada a fundamentação teórica sobre cada etapa do método proposto de registro automático de imagens. O capítulo é finalizado com a apresentação dos resultados de estimação da posição de VANT obtidos pela aplicação do método proposto de registro automático de imagens.

6.1 CONVERSÃO DE IMAGENS PARA TONS DE CINZA

A conversão para tons de cinza consiste na transformação de uma imagem multibandas ou multiespectral em uma imagem monocromática, mantendo as características principais presentes na imagem (HUGHES et al., 2013). Uma imagem multibandas é formada por n, $n \geq 2$ imagens monocromáticas, que também são denominadas bandas. Cada pixel na imagem monocromática representa apenas uma quantidade de luz, ou seja, contém apenas informações de intensidade, este valor é denominado nível de cinza. O processo de conversão de uma imagem multibandas para uma imagem monocromática (também conhecida por imagens em tons de cinza) é aplicado anteriormente aos processos de filtragem, extração de bordas e detecção e descrição dos pontos característicos (PEDRINI; SCHARTZ, 2008).

A conversão de uma imagem multibanda para uma imagem monocromática permite aumento no desempenho dos algoritmos de processamento de imagens, tais como, os algoritmos de filtragem e extração de bordas, pois a computação é realizada em apenas uma banda. De modo geral, esse aumento de desempenho ocorre sem prejuízo ao resultado final do algoritmo, pois a conversão de uma imagem multibanda para monocromática (tons de cinza) mantém as características principais da imagem (HUGHES et al., 2013).

O valor do pixel em uma imagem monocromática pode ser obtido por diferentes métodos, entre eles, pela média aritmética dos valores dos pixels em cada banda da imagem multibanda, como mostrado pela equação:

$$I(x,y) = \sum_{m=1}^{N} \frac{f(x,y)_m}{N} \tag{6.3}$$

onde $f(x,y)_m$ é o valor do pixel na posição (x,y) da banda m, N é quantidade de bandas na imagem multibandas e $I(x,y)$ o valor do pixel, na posição (x,y), na imagem monocromática.

Outro método para converter uma imagem colorida (imagem multibanda formada pela combinação de bandas das cores primárias vermelha (R, "*red*"), verde (G, "*green*") e azul (B, "*blue*") em tons de cinza, segue os seguintes passos (HUGHES et al., 2013):

- Obtém-se o valor da intensidade do pixel na banda R, G e B;
- Então adiciona-se, aproximadamente, 30% do valor da intensidade do pixel na banda R, mais 59% do valor da intensidade do pixel na banda G e mais 11% do valor da intensidade do pixel na banda Azul, como mostrado por:

$$I(x,y) = 0.2989.R(x,y) \ + \ 0.5870.G(x,y) \ + \ 0.1140.B(x,y) \tag{6.4}$$

onde $I(x,y)$ é o valor da intensidade do pixel, na posição x,y, na imagem em tons de cinza, $R(x,y)$, $G(x,y)$ e $B(x,y)$ são os valores da intensidade do pixel, na posição x,y, na banda vermelho, verde e azul, respectivamente. Segundo Hughes et al. (2013), as porcentagens na Equação 6.4 relacionam-se a sensibilidade visual do olho humano para as cores primárias.

6.2 FILTRAGEM DA IMAGEM

As operações de filtragem em imagens são técnicas utilizadas para melhorar a qualidade de uma imagem digital. Estas técnicas permitem diminuir a presença de ruídos, que podem ser causados na aquisição ou das condições nas quais a imagens são capturadas (HUGHES et al., 2013).

A filtragem de imagens pode ser realizada no domínio da frequência ou no domínio

espacial. A filtragem no domínio da frequência opera sobre a transformada de Fourier da imagem, por outro lado, a filtragem no domínio espacial opera diretamente sobre os valores de intensidade dos pixels (PEDRINI; SCHARTZ, 2008).

As técnicas de filtragem espacial realizam uma transformações pixel a pixel na imagem. Os filtros espaciais não consideram apenas o valor da intensidade de um determinado pixel, mas também, o valor da intensidade dos pixels vizinhos. O processo de filtragem é realizado com uso de matrizes denominadas máscaras, as quais são aplicadas sobre a imagem em um processo conhecido por convolução (PEDRINI; SCHARTZ, 2008). Por exemplo, o processo de convolução com uma máscara W, de dimensão 3×3, com a matriz A, também de dimensão 3×3, que representa os valores dos níveis de cinza de um trecho de uma imagem monocromática, é dado por:

$$W = \begin{bmatrix} w_1 & w_2 & w_3 \\ w_4 & w_5 & w_6 \\ w_7 & w_8 & w_9 \end{bmatrix}$$

$$A = \begin{bmatrix} a_1 & a_2 & a_3 \\ a_4 & a_5 & a_6 \\ a_7 & a_8 & a_9 \end{bmatrix} \tag{6.5}$$

$$R = w_1.a_1 + w_2.a_2 + w_3.a_3 + \cdots + w_9.a_9 = \sum_{i=1}^{9} w_i.a_i$$

onde $w_1, w_2, w_3, \cdots, w_9$ são os valores da máscara W. Os parâmetros $a_1, a_2, a_3, \cdots, a_9$ são os valores dos níveis de cinza da imagem monocromática A. O valor R, resposta da convolução, é o novo valor da posição central da matriz A, neste exemplo, $a_5 = R$.

Existem diversos filtros espaciais, entre eles, os filtros passa-baixa e os filtros passa-alta. Os filtros passa-alta, como o próprio nome diz, realçam as altas frequências, realçam os detalhes da imagem, isto é, as transições entre regiões diferentes tornam-se mais nítidas. Desta forma, os filtros de passa-alta são aplicados para realçar certas características como bordas, linhas-curvas ou manchas. Mas os filtros passa-alta possuem a desvantagem de enfatizar o ruído existente na imagem (PEDRINI; SCHARTZ, 2008). Os filtros de passa-baixa atenuam as altas frequências das imagens, causam um efeito de suavização, com isto, sua aplicação minimiza os ruídos presentes

na imagem (PEDRINI; SCHARTZ, 2008).

Segundo Conte e Doherty (2008) e Silva et al. (2015), para imagens aéreas, os filtros passa-baixa realizam a atenuação de ruídos presentes na imagem mas preservam a definição das bordas, melhorando o resultado final do registro automático entre imagens com as bordas detectadas. Além disso, segundo Chust et al. (2008) e Song et al. (2012) deve-se aplicar um filtro de passa-baixa em imagens de intensidade ou de MDS oriundos de uma nuvem de pontos LiDAR, pois durante sua obtenção pelo processo interpolação dos dados geram-se ruídos do tipo "sal e pimenta", que prejudicam a análise dos dados. Exemplos do filtro passa-baixa são o filtro da Mediana e o filtro Gaussiano.

6.2.1 FILTRO DA MEDIANA

Segundo Pedrini e Schartz (2008), o filtro da Mediana é um filtro passa-baixa que pode ser aplicado aplicado no domínio espacial, que possui como principal característica a suavização da imagem, porém, minimizando o efeito de borramento.

O tamanho da máscara do filtro da Mediana tem influência direta no efeito de suavização da imagem, em outras palavras, quanto maior a dimensão da máscara, maior será o efeito de suavização (PEDRINI; SCHARTZ, 2008). Por esse motivo, para evitar perda de informações da imagem, principalmente informações relacionadas as bordas da imagem, normalmente é aplicada uma máscara de dimensões 3×3 (PEDRINI; SCHARTZ, 2008).

O algoritmo do filtro da Mediana consiste nos seguintes passos: 1 - para o tamanho de máscara, ordenar os valores de intensidade dos pixels em ordem crescente e escolher como saída o valor que está no centro da sequência (valor da mediana da sequência); 2 - após esse processo, a posição central da janela sobreposta pela máscara recebe esse valor da mediana, para geração da imagem filtrada.

O filtro da Mediana é ideal para o tratamento de ruídos do tipo "sal e pimenta", no qual, pontos brancos representam o sal e os pretos a pimenta (PEDRINI; SCHARTZ, 2008). A Figura 6.2.1 exibe a aplicação do filtro da Mediana utilizando uma máscara de dimensão 3×3 e a matriz resultante desse processo.

Figura 6.2 - Etapas da aplicação do filtro da Mediana.

Fonte: Produção do autor.

Nas Figuras 6.3 **(A)** e 6.3 **(B)** são apresentados exemplos de uma imagem de intensidade LiDAR sem a aplicação do filtro da Mediana e outra com a aplicação, respectivamente.

Figura 6.3 - (A) - Imagem de intensidade LiDAR sem aplicação do filtro da Mediana. (B) - Imagem de intensidade LiDAR com aplicação do filtro da Mediana.

Fonte: Produção do autor.

6.2.2 FILTRO GAUSSIANO

O filtro Gaussiano é muito utilizado em processamento de imagens pela sua capacidade de suavização e remoção de ruídos na imagem (PEDRINI; SCHARTZ, 2008). No filtro Gaussiano, os coeficientes da máscara são determinados pela função Gaussiana em duas dimensões. A Equação 6.6 apresenta a função Gaussiana em duas dimensões.

$$G(x, y) = \frac{1}{2\pi\sigma_x\sigma_y} e^{\frac{-(x^2+y^2)}{2\sigma^2}} \tag{6.6}$$

onde σ é a largura da função Gaussiana, que determina o grau de suavização do filtro, ou seja, quanto maior o valor de σ maior será a suavização obtida . A matriz a seguir (Equação 6.7) representa uma máscara gaussiana com dimensão 3×3, obtida pela Equação 6.6, com valor de $\sigma^2 = \frac{2}{\pi}$.

$$\begin{bmatrix} \frac{1}{16} & \frac{1}{8} & \frac{1}{16} \\ \frac{1}{8} & \frac{1}{4} & \frac{1}{8} \\ \frac{1}{16} & \frac{1}{8} & \frac{1}{16} \end{bmatrix} \tag{6.7}$$

A Figura 6.4 mostra um exemplo de aplicação de uma máscara gaussiana com dimensão 3×3, os valores da máscara foram obtidos $\sigma^2 = \frac{2}{\pi}$.

Figura 6.4 - Etapas da aplicação do filtro Gaussiano.

Fonte: Produção do autor.

Nas Figuras 6.5 **(A)** e 6.5 **(B)** são apresentados exemplos de uma imagem de intensidade LiDAR sem a aplicação do filtro Gaussiano e outra com a aplicação (máscara Equação 6.7), respectivamente.

Figura 6.5 - (A) - Imagem de intensidade LiDAR sem aplicação do filtro Gaussiano. (B) - Imagem de intensidade LiDAR com aplicação do filtro Gaussiano.

Fonte: Produção do autor

6.3 BINARIZAÇÃO DE IMAGENS MONOCROMÁTICAS

Binarização de imagens é o processo pelo qual atribui-se apenas um entre dois valores possíveis de intensidade para cada pixel. O processo de binarização de imagens deve ocorrer após a conversão da imagem para tons de cinza (LIU; WANG, 2009; HUGHES et al., 2013; SILVA et al., 2015).

A conversão de uma imagem monocromática para sua representação binária (pixel com valores de intensidade preto ou branco) é importante para uma série de objetivos, por exemplo, identificar objetos e separá-los do fundo da imagem (LIU; WANG, 2009; HUGHES et al., 2013; SILVA et al., 2015).

Há na literatura diversos algoritmos para binarizar imagens monocromáticas (PEDRINI; SCHARTZ, 2008; HUGHES et al., 2013). Todos algoritmos de binarização buscam por um limiar (também chamado de "*threshold*"), que determina o valor que minimiza a variância entre os valores de intensidade dos pixel na imagem monocromática. O limiar determina qual será o valor do pixel na imagem binarizada. O valor do limiar pode ser fixo ou variar conforme a imagem (PEDRINI; SCHARTZ, 2008).

O método de Otsu é um algoritmo de binarização de imagens que determina o valor do limiar de forma adaptativa, ou seja, há uma valor de limiar diferente para cada imagem. A principal vantagem de determinar o limiar de forma adaptativa é o ajuste do algoritmo as variações de luminosidade no ambiente, determinando seu valor de forma correta (OTSU, 1979; HUGHES et al., 2013).

As etapas do método de Otsu são:

- construção do histograma da imagem monocromática. O histograma corresponde à distribuição dos níveis de cinza, ou seja, o número de pixels para cada nível de cinza. Os valores dos níveis de cinza em uma imagem monocromática variam de 0 a 255;

- utilizando o histograma da imagem, o algoritmo seleciona o valor que minimiza a variância entre os níveis de cinza, o qual é usado como limiar. Desta forma, valores abaixo do limiar são preto (valor de nível de cinza 0) e acima são branco (valor de nível de cinza 255).

6.4 SEGMENTAÇÃO DE IMAGENS DIGITAIS

A segmentação é um processo que tem como objetivo particionar a imagem em estruturas com conteúdo semântico relevante para a aplicação (PEDRINI; SCHARTZ, 2008). A aplicação de algoritmos para segmentação de imagens justifica-se, pois sua utilização reduz significativamente a quantidade de dados a serem processados e descarta informações que são consideradas menos relevantes, ainda que preservando informações essenciais sobre as formas dos objetos na cena (PEDRINI; SCHARTZ, 2008).

As técnicas aplicadas para a segmentação de imagens são, geralmente, baseadas nas propriedades dos níveis de cinza da imagem e buscam encontrar descontinuidades ou similaridades. Algoritmos de segmentação que buscam por descontinuidades particionam a imagem com base na variação abrupta dos níveis de cinza. Pontos, linhas e bordas, presentes nas imagens, são as descontinuidades normalmente detectadas por estes algoritmos (PEDRINI; SCHARTZ, 2008).

Algoritmos aplicados para a detecção de descontinuidades, de maneira geral, aplicam o processo de convolução com uma máscara para identificá-las. O processo de convolução foi explicado na Seção 6.2.

6.4.1 DETECÇÃO DE BORDAS

Borda é o limite (fronteira) entre duas regiões com propriedades relativamente distintas de nível de cinza (PEDRINI; SCHARTZ, 2008). O processo de detecção de bordas resume-se na identificação de mudanças locais significativas nos níveis de cinza de uma imagem. A Figura 6.6 exibe duas imagens: em **(A)** uma imagem formada por

uma transição de uma região escura para clara e retornando para escura novamente; em **(B)** uma imagem formada por uma transição de uma região clara para escura e retornando para clara novamente.

Figura 6.6 - (A) - Imagem formada por uma transição de uma região escura para clara e retornando para escura novamente. (B) - Imagem formada por uma transição de uma região clara para escura e retornando para escura novamente.

Fonte: Adaptado de Pedrini e Schartz (2008)

Observa-se na Figura 6.6 o perfil horizontal da variação dos níveis de cinza nas duas imagens, a derivada primeira do perfil horizontal é positiva nas transições de regiões escuras para clara (Figura 6.6 **(A)**) e negativa nas transições de regiões claras para escura Figura 6.6 **(B)**). Entretanto, a derivada segunda é positiva na região escura da transição, negativo na região clara da transição e tem valor zero na região de nível de cinza constante. Desta forma, a magnitude da derivada primeira e a variação do sinal na derivada segunda podem ser empregados para detectar a presença de uma borda em uma imagem. A discussão sobre a Figura 6.6 utilizou um perfil horizontal na variação dos níveis de cinza, mas pode ser aplicada a qualquer orientação na imagem (PEDRINI; SCHARTZ, 2008).

A derivada primeira, para qualquer ponto na imagem, é obtida pelo operador gradiente, que é um vetor cuja direção indica o local onde há maior variação. Para o caso de imagens, a direção do vetor gradiente indica o local onde há maior variação dos níveis de cinza (PEDRINI; SCHARTZ, 2008).

Na Equação 6.8 tem-se o cálculo do vetor gradiente na posição (x, y) da imagem.

$$\nabla I(x,y) = \frac{\partial I(x,y)}{\partial x} i + \frac{\partial I(x,y)}{\partial y} j \tag{6.8}$$

onde i e j são vetores unitários nas direções x e y respectivamente.

A derivada segunda, para qualquer ponto na imagem, é obtida pelo operador Laplaciano, exibido na Equação 6.9

$$\nabla^2 I(x,y) = \frac{\partial^2 I(x,y)}{\partial x^2} + \frac{\partial^2 I(x,y)}{\partial y^2} \tag{6.9}$$

Os operadores de Sobel e Canny são algoritmos para a detecção de borda que aplicam o cálculo do vetor gradiente. Ambos operadores, Sobel e Canny, são algoritmos tradicionais aplicados para a detecção de bordas em imagens e obtiveram resultados significativos no registro automático de imagens aéreas com as bordas detectadas (CONTE; DOHERTY, 2008). Portanto, no contexto deste trabalho, os operadores Sobel e Canny são empregados para avaliar os resultados do registro automático utilizando imagens com as bordas detectadas por uma MLP implementada em FPGA, utilizando a estratégia de LUT.

6.4.2 OPERADOR DE SOBEL

O operador de Sobel realiza a detecção de bordas com base no cálculo da magnitude do vetor gradiente de um pixel em relação aos seus vizinhos. Na Equação 6.10 tem-se o cálculo da magnitude do vetor gradiente $\nabla I(x, y)$ na posição (x, y).

$$G(x,y) = \sqrt{(\frac{\partial I(x,y)}{\partial x} i)^2 + (\frac{\partial I(x,y)}{\partial y} j)^2} \tag{6.10}$$

Quando o valor de magnitude for acima de um limiar ("*threshold*") então é detectada a presença de borda na imagem.

Devido ao custo computacional, o operador Sobel faz uma aproximação do cálculo da magnitude do vetor gradiente na posição (x, y) por meio da diferença ponderada dos valores dos níveis de cinza da imagem, como exibido na Equação 6.11, que é a aproximação na forma discreta para o caso de uma região de 3×3 pixels.

$$
\begin{aligned}
G(x,y) &\approx G(x) + G(y) \\
G(x) &\approx G(x)_1 - G(x)_2 \\
G(y) &\approx G(y)_1 - G(y)_2
\end{aligned}
\tag{6.11}
$$

$$
\begin{aligned}
G(x)_1 &= [f(x+1,y-1) + 2f(x+1,y) + f(x+1,y+1)] \\
G(x)_2 &= [f(x-1,y-1) + 2f(x-1,y) + f(x-1,y+1)]
\end{aligned}
\tag{6.12}
$$

$$
\begin{aligned}
G(y)_1 &= [f(x-1,y+1) + 2f(x,y+1) + f(x+1,y+1)] \\
G(y)_2 &= [f(x-1,y-1) + 2f(x,y-1) + f(x+1,y-1)]
\end{aligned}
\tag{6.13}
$$

Os cálculos de G_x e G_y podem ser obtidos pela convolução das máscaras Mg_x (Equação 6.14) e Mg_y (Equação 6.15) com a imagem.

$$
Mg_x = G(x) = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}
\tag{6.14}
$$

$$
Mg_y = G(y) = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}
\tag{6.15}
$$

onde Mg_x calcula a magnitude do vetor gradiente na direção horizontal e Mg_y calcula a magnitude do vetor gradiente na direção vertical.

O valor final do gradiente é obtido pela combinação dos resultados G_x e G_y, como mostrado na Equação 6.16.

$$
G(x,y) \approx Mg_x + Mg_y
\tag{6.16}
$$

Máscaras de tamanho 5×5, 7×7, ou até mesmo maiores, podem ser aplicadas à imagem. Estas máscaras são menos sensíveis ao ruído, entretanto, o custo computacional para o cálculo da magnitude do vetor gradiente aumenta significativamente (PEDRINI; SCHARTZ, 2008).

6.4.3 OPERADOR DE CANNY

O operador Canny é um algoritmo que busca otimizar a localização de bordas na presença de ruídos (PEDRINI; SCHARTZ, 2008). Canny é um algoritmo multi-passos para a detecção de bordas, que realiza uma convolução de um filtro Gaussiano sobre a imagem, suaviza os ruídos presentes e localiza as bordas por meio do cálculo da magnitude do gradiente.

Segundo Pedrini e Schartz (2008), o operador Canny é composto das seguintes etapas:

1 - o primeiro passo do algoritmo é a suavização da imagem por meio do filtro Gaussiano, isto consequentemente reduz o efeito de pixels ruidosos na imagem. O filtro Gaussiano foi apresentado na Seção 6.2.2;

2 - em seguida, a magnitude do vetor gradiente é calculada utilizando a aproximação em forma discreta, descrita na seção 6.4.2;

3 - o passo seguinte é a obtenção da direção da magnitude do gradiente pela Equação 6.17. Realiza-se uma aproximação do valor obtido de θ para 135°, 90°, 45° e 0°

$$\theta = arctan\left(\frac{|Ig_y|}{|Ig_x|}\right) \tag{6.17}$$

4 - Da imagem original, criam-se duas matrizes: a primeira com a magnitude dos gradientes; a segunda com a direção de cada magnitude do gradiente. A borda localiza-se nos pontos onde a magnitude é localmente máxima na direção do gradiente, esta operação é denominada supressão não-máxima e reduz a espessura das bordas;

5 - Para evitar fragmentos de borda durante a supressão não-máxima aplica-se a limiarização com histerese. Esta técnica aplica dois valores de limiares diferentes T_1 e T_2, onde $T_2 > T_1$. Pontos de borda com magnitude maior que T_2 são mantidos como borda, qualquer ponto conectado a esse será considerado pertencente a borda se a magnitude do gradiente for maior que T_1.

6.4.4 REDES NEURAIS ARTIFICIAIS PARA DETECÇÃO DE BORDAS

As RNAs vem sendo aplicadas, com grande sucesso, na detecção de bordas (TERRY; VU, 1993; LI et al., 2008; GOLTZ et al., 2011; SILVA et al., 2015; BRAGA et al., 2016). Este modelo matemático pode ser utilizado como filtro para a detecção de borda, pois as RNAs são naturalmente um filtro não-linear, devido a não-linearidade da função de ativação dos neurônios computacionais e dos pesos sinápticos (valor da conexão entre neurônios) da rede neural (TERRY; VU, 1993). Segundo Li et al. (2008), as redes neurais possuem maior acurácia na detecção de bordas, pois não realizam mudanças nos valores de tons de cinza das imagens durante o processo de convolução da máscara (fase de ativação da MLP), diferentemente dos algoritmos tradicionais de detecção de bordas.

6.4.4.1 REDES NEURAIS ARTIFICIAIS

A RNA é um modelo matemático, projetado para delinear como o cérebro humano realiza uma tarefa de interesse. A RNA normalmente é implementada por programação em software ou utilizando componentes eletrônicos em hardware (HAYKIN, 1998). A RNA assemelha-se ao cérebro humano devido as seguintes características (HAYKIN, 1998):

- A unidade básica de funcionamento é o neurônio;
- Conexões entre os neurônios, sinapses (pesos sinápticos), armazenam o conhecimento;
- O conhecimento é adquirido por meio de um processo de aprendizagem;

O neurônio da RNA, denominado neurônio artificial, foi proposto por McCulloch e Pitts (1943) e imita o funcionamento do neurônio biológico. O neurônio artificial funciona como um somador simples, que combina todos os sinais de entrada e por meio de uma função de ativação produz um único sinal de saída.

A Figura 6.7 exibe a representação esquemática de um neurônio artificial.

Figura 6.7 - Neurônio artificial.

Fonte: Adaptado de Haykin (1998).

No neurônio artificial, a soma ponderada das entradas é efetuada pela Equação 6.18 e a saída do neurônio é dada pela Equação 6.19

$$vk = \sum_{j=1}^{n} w_{kj} \times x_j \tag{6.18}$$

Na Equação 6.18, vk é o resultado do combinador linear do neurônio, produzido pela soma de todas as entradas x_j, para $j = 1, 2, \ldots, n$ multiplicadas por seu respectivo peso sináptico w_{kj}. O valor de n é igual ao número de entradas do neurônio, ou seja, quantidades de neurônios da camada imediatamente anterior a que ele pertence.

$$y_k = \varphi(vk) \tag{6.19}$$

Na Equação 6.19, uma função de ativação φ é aplicada ao termo v_k para produzir a saída y_k do neurônio artificial.

A função de ativação é a responsável por produzir a resposta (geração do sinal de saída) do neurônio artificial. Existem diversas funções matemáticas que podem ser aplicadas como função de ativação do neurônio artificial, a Figura 6.8 ilustra três exemplos mais comumente empregados (BRAGA et al., 2007).

Figura 6.8 - Funções de ativação.

Fonte: Adaptado de Braga et al. (2007).

A principal propriedade de uma RNA é conseguir aprender a partir dos dados de entrada e assim melhorar seu desempenho (HAYKIN, 1998). A RNA aprende por meio de um processo iterativo de ajustes dos seus pesos sinápticos denominado treinamento (HAYKIN, 1998). O aprendizado ocorre quando a RNA apresenta uma solução generalizada para uma classe de problemas (HAYKIN, 1998).

O treinamento de uma RNA é marcado pela modificação dos valores dos pesos sinápticos, ou simplesmente pesos. O treinamento segue um conjunto de regras bem definidas denominado de algoritmo de treinamento, ou algoritmo de aprendizagem (HAYKIN, 1998). Existem diversos algoritmos de aprendizagem, alguns específicos para determinados modelos de RNA (HAYKIN, 1998). Estes algoritmos diferem entre si pelo modo como os pesos são modificados, mas, basicamente, são classificados em duas categorias (HAYKIN, 1998):

- Supervisionado: a RNA recebe como parâmetro de entrada um conjunto de pares ordenados, cada um formado por: entrada da rede e a resposta desejada para a respectiva entrada. Os valores dos pesos são ajustados até que a diferença entre a resposta da RNA e a resposta desejada seja menor que um valor mínimo pré-determinado;

- Não-supervisionado: A RNA recebe um conjunto de entrada sem a respectiva saída desejada. A RNA busca encontrar propriedades (similaridades)

no conjunto de dados para gerar sua resposta. A RNA agrupa os dados similares, aqueles pertencentes à mesma classe, e gera como resposta um agrupamento de dados ("*cluster*").

Dentro dessas duas classes há diversos tipos de algoritmos de treinamento, específicos para determinada classe de RNA, como por exemplo, o algoritmo de Retropropagação do Erro, um algoritmo de treinamento supervisionado, que pode ser aplicado no processo de aprendizagem da MLP (HAYKIN, 1998).

6.4.4.2 REDE NEURAL ARTIFICIAL COM PERCEPTRONS EM MÚLTIPLAS CAMADAS

A MLP, amplamente utilizada pela comunidade científica, foi concebida para a resolução de problemas linearmente inseparáveis, que não poderiam ser solucionados pela rede Perceptron de Camada Única (HAYKIN, 1998). A MLP tem sido empregada com sucesso em diversas áreas do conhecimento, como (HAYKIN, 1998):

- Aproximação de funções (BRAGA et al., 2007);
- Classificação de padrões (DUDA et al., 2000)
- Controle de navegação de VANTs (KIM et al., 1993), entre outras.

As principais características da MLP são:

- Possui uma camada de neurônios não-computacionais, que formam a camada de entrada da MLP, a qual recebe os dados (estímulos) do ambiente (BRAGA et al., 2007);
- Uma ou mais camadas formadas por neurônios computacionais, denominadas camadas ocultas. Os neurônios computacionais capacitam a MLP para a resolução de problemas complexos, extraindo progressivamente as características mais significativas dos padrões a ela apresentados (BRAGA et al., 2007);
- Uma camada de saída, também formada por neurônios computacionais (BRAGA et al., 2007);
- Cada neurônio computacional inclui uma função de ativação não-linear, mas esta não-linearidade é suave, pois a função deve ser diferenciável em

qualquer ponto (BRAGA et al., 2007). Exemplos de funções de ativação utilizadas nos neurônios computacionais foram exibidas na Figura 6.8;

- Há um alto grau de conectividade entre os neurônios que formam a MLP. Os neurônios pertencentes a uma camada estão ligados a todos os neurônios da camada imediatamente anterior. As ligações entre os neurônios são denominadas pesos, ou pesos sinápticos. O conhecimento adquirido pela rede neural durante o treinamento fica armazenado nos valores dos pesos (BRAGA et al., 2007).

A arquitetura da MLP é formada pelo arranjo topológico das unidades básicas de processamentos, os perceptrons (neurônios artificiais), e pelos pesos, que são os valores das conexões entre os perceptrons da rede (HAYKIN, 1998).

A Figura 6.9 apresenta um exemplo de arquitetura de uma MLP, constituída de uma camada de entrada, uma camada escondida e uma camada de saída.

Figura 6.9 - Arquitetura de uma rede neural artificial MLP.

Fonte: Adaptado de Haykin (1998).

O algoritmo de treinamento supervisionado de Retropropagação de Erro é o mais normalmente aplicado no processo de aprendizagem de uma MLP (HAYKIN, 1998). Este algoritmo fundamenta-se no ajuste dos pesos para minimizar a diferença entre a resposta da rede, para um padrão específico, e a resposta desejada, para o mesmo padrão. O algoritmo de Retropropagação de Erro consiste de duas etapas principais

(HAYKIN, 1998):

- Propagação: nesta etapa do algoritmo, os parâmetros livres (pesos) da MLP são todos mantidos fixos. Um conjunto de saída é produzido como resposta que é subtraído da saída desejada, gerando um sinal de erro, como mostrado na Equação 6.20.

$$e_n = yr_n - yd_n \tag{6.20}$$

  onde e_n é o erro para o padrão de entrada n apresentado à MLP, yr_n é a resposta da MLP para o padrão de entrada n e yd_n é a resposta desejada para o padrão de entrada n.

- Retropropagação: nesta etapa do algoritmo, o erro e_n calculado na fase de propagação é retro-propagado pela MLP, camada após camada, para realizar a correção dos pesos (parâmetros livres) da rede. A correção dos pesos ocorre segundo uma regra de correção, por exemplo, a regra delta generalizada. Os pesos são ajustados para que a resposta da MLP se aproxime da resposta desejada a cada novo padrão de entrada apresentado à rede neural.

A regra delta generalizada é a estratégia mais aplicada para realizar a correção dos pesos no algoritmo de Retropropagação de Erro. Na regra delta generalizada os pesos são corrigidos segundo a Equação 6.21.

$$w_{ij}(n+1) = w_{ij}(n) + \Delta w_{ij}(n+1) \tag{6.21}$$

$w_{ij}(n)$ é o valor da conexão (peso) entre a $i-ésima$ entrada do $j-ésimo$ neurônio para o padrão de entrada n e $\Delta w_{ij}(n+1)$ é o valor de incremento para correção do peso. O valor de $\Delta w_{ij}(n+1)$ é obtido segundo a Equação 6.22.

$$\Delta w(n+1) = \alpha\, \Delta w(n) + \eta\, \delta(n)\, y(n) \tag{6.22}$$

onde α é o valor da constante momentum, η é o valor da taxa de aprendizado, δ é o valor do gradiente do neurônio e y_n é o valor da saída produzida pelo neurônio.

Momentum, taxa de aprendizado, quantidade de camadas ocultas e quantidade de

neurônios em cada camada oculta são parâmetros da MLP, definidos por seu usuário, em um processo de tentativa e erro, ou obtidos automaticamente por um método de otimização como o MPCA.

6.4.4.3 CONFIGURAÇÃO AUTOMÁTICA DE REDES PERCEPTRON EM MÚLTIPLAS CAMADAS

As MLPs têm sido aplicadas com sucesso em diversas áreas do conhecimento, mas ainda é um desafio a determinação de uma arquitetura da rede neural, para solucionar um determinado problema proposto (HAYKIN, 1998; LUZ, 2012; ANOCHI, 2015).

A escolha da arquitetura de uma MLP, para solucionar um problema específico, é muitas vezes baseada na seguinte sequência:

1 - Determinação dos parâmetros livres da RNA: quantidade de camadas escondidas, quantidade de neurônios em cada camada escondida, valor da constante momentun, valor da taxa de aprendizado e função de ativação dos neurônios computacionais;

2 - Treinamento da MLP com a configuração selecionada anteriormente;

3 - Realizar o teste de generalização da MLP. Este teste consiste em a MLP apresentar uma resposta adequada a padrões que não foram apresentados durante o treinamento;

4 - Voltar a etapa 1 e comparar os resultados produzidos pelas diversas arquiteturas. Selecionar a melhor arquitetura.

O processo empírico de determinação da arquitetura da MLP não garante que a melhor configuração seja selecionada para a resolução do problema. A escolha de uma arquitetura adequada é importante, pois se uma MLP é configurada com uma quantidade pequena de neurônios esta pode não ser capaz de aprender os padrões apresentados. Entretanto, uma grande quantidade de neurônios pode causar dificuldades na generalização (HAYKIN, 1998).

Uma maneira de encontrar uma arquitetura para uma MLP é a utilização do MPCA, que determina de forma automática os parâmetros livres da arquitetura da rede neural (ANOCHI, 2015).

6.4.4.4 ALGORITMO DE COLISÃO DE MÚLTIPLAS PARTÍCULAS

O MPCA é uma nova meta-heurística introduzida por Luz (2012), que busca encontrar o melhor conjunto de parâmetros para uma função objetivo, minimizando ou maximizando seu valor. O MPCA é uma extensão do Algoritmo de Colisão de Partículas *(PCA, do inglês, Particle Collision Algorithm)* proposto por Sacco e Oliveira (2005) , que foi inspirado nas reações que ocorrem em um reator nuclear, principalmente, nas reações de espalhamento e absorção. As principais etapas do PCA são:

1 - O usuário fornece ao algoritmo uma configuração inicial dos parâmetros da função objetivo;

2 - O algoritmo cria uma nova configuração de parâmetros para a função objetivo;

3 - O algoritmo compara as respostas geradas pelas configurações das etapas 1 e 2, e seleciona (**Absorve**) a configuração de melhor resposta. A configuração selecionada (**Absorvida**) será utilizada na comparação da próxima iteração.

4 - O algoritmo descarta (**Espalha**) a configuração de pior resposta para a função objetivo;

5 - Retorna ao passo 2 ou termina a execução, caso tenha atingido um limite de testes pré-determinado pelo usuário.

O MPCA realiza a busca pela configuração de uma função objetivo de forma semelhante ao PCA, mas, o MPCA introduz o conceito de exploração múltipla, ao invés de uma única exploração (ANOCHI, 2015). Para coordenar esta busca multi-explorativa, criou-se uma função de comunicação entre processos utilizando-se a Interface de Trocas de Mensagem *(MPI, do inglês, Message Passing Interface)*, que é um padrão de comunicação de dados em computação paralela. Desta forma, as principais etapas do algoritmo MPCA são:

1 - O usuário fornece ao algoritmo uma configuração inicial dos parâmetros da função objetivo, que armazena esta configuração em uma função denominada Quadro-Negro;

2 - Cada processador (partícula) verifica a configuração armazenada na função Quadro-Negro e, com base nesta, cria uma nova configuração para a função objetivo;

3 - Cada processador (partícula) grava a resposta de sua função objetivo na função Quadro-Negro, que compara os resultados de todas as funções objetivos.

4 - A função Quadro-Negro seleciona (**Absorve**) a configuração, entre todas as configurações geradas por cada processador (partícula), que gera a melhor resposta. A configuração selecionada (**Absorvida**) será utilizada na comparação da próxima iteração.

5 - O algoritmo descarta (**Espalha**) as configurações de pior resposta para a função objetivo;

6 - Retorna a etapa 2 ou termina a execução caso tenha atingido um limite de testes pré-determinado pelo usuário.

Com essas modificações, cada processador (partícula) trabalha de maneira independente na busca pelos parâmetros que otimizam a função objetivo, mas a função Quadro-Negro encarrega-se de manter a melhor configuração e passa-lá a todos processadores.

Em Anochi (2015) foram feitas modificações no MPCA, introduzindo no algoritmo a função objetivo que determina a configuração de uma MLP, esta função objetivo é exibida na Equação 6.23 e foi determinada por Carvalho et al. (2011).

$$f_{obj} = penalty \times \left(\frac{\rho_1 \times E_{train} + \rho_2 \times E_{gen}}{\rho_1 + \rho_2} \right) \tag{6.23}$$

Na Equação 6.23 $\rho_1 = 1$ e $\rho_2 = 0.1$ são os mesmos valores proposto por Carvalho et al. (2011), que são fatores de ajustes que modificam a relevância atribuídas aos erros de treinamento e generalização, respectivamente (CARVALHO et al., 2011). Os valores E_{train} e E_{gen} são definidos pela Equação 6.24 e Equação 6.25, respectivamente.

$$E_{train} = \frac{1}{N} \sum_{j=1}^{N} (d_j - \hat{y}_j)^2 \tag{6.24}$$

$$E_{gen} = \frac{1}{N}\sum_{j=1}^{N}(y_j - \hat{y_k})^2 \tag{6.25}$$

onde n é o número de padrões apresentados à MLP durante o treinamento e generalização, d_j é a saída desejada para a $j-ésima$ entrada, apresentada à MLP durante o treinamento, e $\hat{y_j}$ é a resposta da MLP para $j-ésima$ entrada apresentada à MLP. Na Equação 6.25, y_j é a resposta desejada para a $j-ésima$ entrada apresentada à MLP no teste de generalização.

Portanto, a função objetivo definida por Carvalho et al. (2011), aplicada para a determinação de uma arquitetura de uma RNA, consiste na combinação dos critérios de erro de treinamento E_{train} e de erro de generalização E_{gen} com um fator de penalização $penalty$.

O termo *penalty* (penalidade), exibido na Equação 6.26, é aplicado para evitar arquiteturas muito complexas.

$$penalty = c_1^{(número\ de\ neurônios)^2} + c_2^{(número\ de\ épocas)} + 1 \tag{6.26}$$

onde $c_1 = 5 \times 10^8$ and $c_2 = 5 \times 10^5$ são parâmetros para encontrar um equilíbrio para a complexidade da rede neural (ANOCHI, 2015).

A função objetivo, Equação 6.23, aplicada para a determinação de uma arquitetura de uma MLP, consiste na combinação dos critérios de erro de treinamento E_{train} e de erro de generalização E_{gen} com um fator de penalização $penalty$. Desta forma, a otimização da arquitetura de uma MLP é a aquisição de configurações que apresentam um desempenho tão próximo quanto possível de um ótimo global, em outras palavras, é a busca de arquiteturas com baixos erros de treinamento e generalização (ANOCHI, 2015).

6.4.4.5 TREINAMENTO DA REDE NEURAL COM PERCEPTRONS EM MÚLTIPLAS CAMADAS PARA A DETECÇÃO DE BORDAS

Nesta pesquisa de doutorado, uma etapa importante inserida no método proposto de registro automático de imagens, para estimar a posição do VANT, é a detecção de bordas nas imagens aéreas, capturadas pela aeronave, e na imagem georreferenciada. A detecção de bordas nas imagens é importante, pois esta informação tende a estar

presente em ambas as imagens, mesmo com a variação de iluminação, condição climática ou tipos de sensores (PEDRINI; SCHARTZ, 2008; SILVA et al., 2015).

Atualmente, a MLP tem sido empregada com sucesso no processo de detecção de bordas, principalmente devido à sua capacidade de generalização, tolerância a variações nas imagens e a ruídos, desde que com uma arquitetura bem definida e devidamente treinada (LI et al., 2008; BRAGA et al., 2016; SILVA et al., 2015)

No contexto desta pesquisa de doutorado, a MLP, treinada pelo algoritmo de Retropropagação de erro, é aplicada para para a detecção de bordas. A MLP aplicada classifica padrões binários de borda. Com isso, o treinamento e ativação da MLP, para o processo de detecção de bordas, são realizados da seguinte forma:

Para uma imagem binarizada, um conjunto com 26 padrões, sendo 24 padrões de borda e 2 de não-borda, é aplicado para o treinamento da MLP. Este conjunto é o mesmo descrito em Braga et al. (2016) e é baseado nas pesquisas desenvolvida por Terry e Vu (1993) e (GOLTZ et al., 2011), que também aplicaram a MLP para a detecção de bordas em imagens binarizadas. O conjunto com os 26 padrões, utilizados para o treinamento da MLP para detecção de bordas em imagens binarizadas, é exibido na Figura 6.10.

Figura 6.10 - Padrões de borda e não-borda utilizados para o treinamento da MLP

Fonte: Produção do autor.

Os padrões presentes no conjunto de treinamento são descritos por uma matriz binária com dimensão 3×3. Matrizes com dimensões maiores, como 5×5 ou 7×7, no entanto, devido aos requisitos do problema abordado nesta tese, não são utilizados (TERRY; VU, 1993; BRAGA et al., 2016)

A camada de entrada da MLP é composta por nove neurônios (não computacionais), que recebem para treinamento um vetor de dimensões 1×9. Cada vetor corresponde a uma matriz de dimensão 3×3 com valores binários, que representa um dos padrões de treinamento. A Figura 6.11 ilustra um exemplo da transformação da matriz de um padrão de treinamento em vetor.

Figura 6.11 - Transformação de um padrão de treinamento (matriz) em vetor.

Fonte: Produção do autor.

A camada de saída da MLP é formada por um neurônio computacional que produz como resposta um valor, onde 1 corresponde a um padrão de borda e o valor 0 corresponde a um padrão de não-borda.

A ativação da MLP para a detecção de bordas é feita por meio de uma janela de 3×3 pixels, que percorre uma imagem binarizada, da esquerda para a direita, pixel a pixel, formando o vetor de entrada com dimensão 1×9. Ao chegar ao final da linha a janela decai um pixel, iniciando a nova operação à esquerda da imagem. A Figura 6.12 ilustra o processo de ativação descrito. O limiar para realizar a binarização da imagem pede ser determinado pelo método de Otsu. Este método foi escolhido, pois se adapta à luminosidade da imagem (informações sobre a binarização e o método de Otsu encontram-se na Seção 6.3) (OTSU, 1979).

Figura 6.12 - Ativação da MLP.

Fonte: Adaptado de Goltz et al. (2011).

O MPCA foi aplicado para determinar a arquitetura da MLP empregada para a detecção de borda. A arquitetura da MLP, determina pelo MPCA, é exibida na Tabela 6.1,

Tabela 6.1 - Arquitetura da MLP para a detecção de bordas em imagens, determinada pelo MPCA

	MLP
Neurônios de entrada	9
Neurônios de saída	1
Nº de camadas escondidas	1
Nº de neurônios na camada escondida	18
Função de ativação	tangente hiperbólica
Taxa de momentum	0.85
Taxa de aprendizado	0.73

6.4.4.6 IMPLEMENTAÇÃO DA MLP PARA DETECÇÃO DE BORDA EM FPGA

Durante o desenvolvimento desta pesquisa de doutorado, observou-se que a detecção de borda por meio da ativação da arquitetura da MLP (determinada pelo MPCA) era a etapa do método de registro automático de imagens proposto com pior desempenho computacional. O algoritmo de registro automático de imagens, quando configurado com os operadores Canny e Sobel, realizavam 5 e 4 estimações de posição do VANT por segundo, respectivamente. Enquanto que, o método de registro automático de imagens, configurado com a MLP de arquitetura determinada pelo MPCA, realizava 1 estimação de posição a cada 6 segundos. Estes resultado foram obtidos com a execução do algoritmo de registro automático de imagens no computador RaspBerry Pi Model B-1 e podem ser verificados em Braga et al. (2016).

Devido ao baixo desempenho da ativação da MLP para a detecção de borda, nesta tese de doutorado, é proposto sua implementação em FPGA utilizando uma estratégia em LUT, com o objetivo de melhorar o desempenho e sem prejudicar o resultado da estimação da posição do VANT.

Tendo em conta os padrões binários de bordas, determinados por uma janela de dimensões 3×3, há 512 padrões possíveis para especificar bordas e não-bordas.

Após a determinação da arquitetura da MLP pelo MPCA, a rede neural foi aplicada para classificar os 512 padrões possíveis de borda e não borda, considerando uma matriz 3×3 para identificar o padrão. O resultado final da classificação desses padrões pode ser armazenado em uma Tabela de Pesquisa, também denominada de "*Look-UP Table*" (LUT). A LUT possui um índice binário (comprimento 9), com um intervalo de 0 até 511, e um valor associado a cada índice. Este valor é 0, quando o índice binário não foi classificado pela MLP como uma borda, ou 1, quando o índice binário foi classificado pela MLP como borda. A Tabela 6.2 exibe alguns exemplos presentes na LUT.

Tabela 6.2 - Exemplos de padrões presentes na LUT

matriz de entrada (3×3)									padrão	valor
0	0	0	0	0	0	0	0	0	não-borda	0
0	0	0	0	0	0	0	0	1	não-borda	0
1	0	0	0	0	0	0	0	0	não-borda	0
1	1	1	0	0	0	1	1	1	borda	1
0	0	0	1	1	1	0	0	0	borda	1
0	0	0	0	0	0	1	1	1	borda	1

Segundo Pedroni (2008), a LUT é uma tabela que substitui alguma computação realizada em tempo de execução por uma simples operação de indexação, o principal objetivo do emprego de LUT é melhorar a eficiência computacional do algoritmo que a aplica.

Nesta tese de doutorado, a LUT para a detecção de bordas em imagens é implementada em VHDL e executada pela FPGA. Dois computadores que podem ser embarcados em VANT e que possuem FPGA foram selecionados para a implementação: o Raspberry Pi Model B-1 e o Zybo ZINQ 7000. Mais detalhes sobre esses computadores estão nas Seções 3.1.1 e 3.1.2, respectivamente.

Implementação da MLP em LUT para a detecção de borda no computador Rapeberry Pi Model B-1 - Cada padrão de borda é representado por uma matriz, com valores binários, de dimensão 3×3 (vetor com valores binários de dimensão 1×9). Uma máscara de dimensão 3×3 percorre a imagem binarizada, como mostrado na Figura 6.12, e envia estes valores a FPGA da Raspberry PI. O computador Raspberry Pi permite uma comunicação de 16 bits entre a FPGA e processador ARM, assim, os 9 bits menos significativos do canal de comunicação

são selecionados para acessar a LUT. Desta forma, a cada pulso de clock um padrão pode ser enviado à FPGA para pesquisa na LUT. A Figura 6.13 exibe o processo de consulta à LUT implementada na FPGA do computador Raspberry PI.

Figura 6.13 - Consulta à LUT implementada na FPGA do computador Raspberry PI

Fonte: Produção do autor.

Implementação da MLP em LUT para a detecção de borda no Zybo Zynq 7000 - O computador Zybo ZINQ 7000 permite uma comunicação de 32 bits entre FPGA e processador ARM. Desta forma, ao invés de uma máscara percorrendo a imagem são utilizadas três máscaras de dimensão 3×3 ao mesmo tempo. A cada pulso de clock três padrões podem ser enviados à FPGA para pesquisa na LUT, pois três padrões correspondem a 27 bits. A Figura 6.14 exibe o processo de consulta à LUT implementada na FPGA do computador Zybo ZINQ 7000.

Figura 6.14 - Consulta à LUT implementada na FPGA do computador Zybo ZINQ 7000

Fonte: Produção do autor.

A LUT implementada (Figura 6.14) no computador Zybo ZINQ 7000 possui três entradas e cada uma possui uma tabela de consulta com os 512 padrões possíveis de borda. A palavra enviada à FPGA no computador Zybo possui tamanho de 32 bits, os 27 bits menos significativos são divididos em três palavras de 9 bits (tamanho do padrão que define a borda) e enviados a cada entrada da LUT. A resposta é um vetor de tamanho 3, onde cada posição é uma resposta para cada entrada da LUT.

O processo de detecção de bordas pela LUT é semelhante ao processo de ativação da MLP para a detecção de borda. A diferença encontra-se no cálculo realizado pela MLP para determinar se um padrão é ou não uma borda. Este cálculo é substituído por uma simples consulta à LUT. Na detecção de bordas aplicando a LUT, uma máscara de dimensão 3×3 percorre a imagem binarizada, da esquerda para a direita, pixel a pixel, como exibido na Figura 6.12, os valores presentes nesta janela são enviados para pesquisa na LUT, que responde da seguinte forma: **A** - se o valor de resposta da LUT é 1, então o padrão é uma borda. O valor do pixel central na imagem coberta pela máscara recebe o valor 1; **B** - se o valor de resposta da LUT é 0, então o padrão não é uma borda. O valor do pixel central na imagem coberta

pela máscara recebe o valor 0;

Na implementação da LUT no computador Raspberry PI há uma máscara de dimensão 3×3 realizando o processo de detecção de borda descrito anteriormente. Entretanto, na implementação da LUT no computador Zybo ZINQ 7000, devido o tamanho da palavra na comunicação entre FPGA e processador ARM ser de 32 bits, há três máscaras de dimensão 3×3 realizando o processo de detecção de borda descrito anteriormente.

6.5 ESTIMAÇÃO DA POSIÇÃO DO VANT POR CORRELAÇÃO DE IMAGENS

Há diferentes métodos para se realizar a estimação da posição de um VANT por meio da correlação entre duas imagens, entre estes métodos podem ser citados: cálculo da correlação por soma da diferença quadrática (HISHAM et al., 2015); cálculo da correlação cruzada normalizada (HISHAM et al., 2015) e o cálculo da correlação cruzada (CONTE; DOHERTY, 2008). Este último método foi escolhido para realizar a correlação entre imagens nesta tese de doutorado, pois apresentou resultados promissores em Conte e Doherty (2008), Goltz et al. (2011), Braga et al. (2016) e Silva et al. (2015).

A correlação cruzada, entre as imagens com as bordas detectadas, é aplicada para a obtenção da estimação da posição do VANT. A correlação cruzada é calculada entre a imagem aérea, capturada pelo VANT, e a imagem georreferenciada da região sobrevoada. O resultado da correlação cruzada de duas imagens é uma matriz cujo o seu maior valor representa a posição do VANT na imagem georreferenciada. A Figura 6.15 exibe o gráfico da matriz resultado da correlação cruzada entre duas imagens.

Figura 6.15 - Matriz resultado da correlação cruzada entre duas imagens. O ponto marcado pelo círculo em vermelho é o maior valor da matriz de correlação, a posição ocupada pelo VANT

Fonte: Produção do autor.

Na Figura 6.15, o círculo em vermelho representa o maior valor da matriz resultado da correlação entre duas imagens, que é a posição estimada do VANT.

O cálculo da correlação cruzada entre uma matriz I de dimensões $M \times N$, que representa a imagem do VANT com as bordas detectadas, e uma matriz T de dimensões $P \times Q$, que representa a imagem georreferenciada com as bordas detectadas, é determinado por:

$$C(k,l) = \sum_{m=0}^{M-1} \sum_{n}^{N-1} (I(m,n) \times \bar{T}(m-k, n-l)) \tag{6.27}$$

onde $C(k,l)$ é a matriz de correlação C de índices k e l com $-(P-1) \leq k \leq (M-1)$ e $-(Q-1) \leq l \leq (N-1)$, a matriz $\bar{T}$ é o complexo conjugado da matriz T

Para identificar a área correspondente da imagem aérea capturada pelo VANT (imagem I) na imagem georreferenciada (imagem T), realiza-se a o cálculo da correlação cruzada entre a imagem I e a imagem T. Este cálculo é realizado entre o pixel central

da imagem do VANT, com cada pixel da imagem georreferenciada, que é percorrida da esquerda para a direita e de cima para baixo.

A Figura 6.16 (A) exibe a correlação cruzada entre a matriz I da imagem do VANT com as bordas detectadas (imagem menor) e a matriz T da imagem georreferenciada com as bordas detectadas (imagem maior). A matriz resultado da correlação cruzada C é exibida na Figura 6.16 (B), o quadrado em vermelho com o ponto no centro representa a posição estimada do VANT (maior valor da matriz de correlação).

Figura 6.16 - Correlação cruzada entre imagem aérea, capturada pelo sensor imageador do VANT, e uma imagem georreferenciada da região sobrevoada. Ambas as imagens estão com as bordas detectadas.

Fonte: Produção do autor.

O maior valor na matriz C (maior valor da correlação cruzada) corresponde ao pixel central da imagem do VANT. Considerando que a aeronave ocupa o pixel central da imagem capturada por seu sensor imageador (sensor LiDAR ou câmera de vídeo), o maior valor da matriz de correlação corresponde a posição da aeronave. Devido ao fato da correlação ser realizada com uma a imagem georreferenciada, as coordenadas geográficas do ponto de maior valor de correlação cruzada podem ser obtidas utilizando as coordenadas do ponto em T, que coincide com o ponto central de I. Desta forma, o processo de estimação da posição do VANT por registro automático de imagens estará completo.

6.6 RESULTADOS: DISCUSSÃO E ANÁLISE

O método de registro automático de imagens, empregado nesta tese de doutorado, baseia-se nos trabalhos desenvolvidos por:

- Conte e Doherty (2008), que apresenta um algoritmo de registro automático de imagens com as mesmas etapas do algoritmo empregado nesta tese de doutorado. Porém, um problema observado em seu trabalho é a não avaliação de diferentes algoritmos para a extração de bordas, apenas o operador Sobel é empregado. O conjunto de dados empregado em Conte e Doherty (2008) foi obtido por sensores ópticos que captam informação na banda do visível;

- Goltz et al. (2011), que aplica um algoritmo para o registro automático de imagens com as mesmas etapas do algoritmo empregado em Conte e Doherty (2008), mas realiza testes com diferentes algoritmos para a detecção de bordas, entre eles, a MLP, que apresenta os melhores resultados. Porém, alguns problemas são observados em seu trabalho: não é utilizada uma meta-heurística para a determinação da configuração da MLP. Além disso, somente o filtro da mediana é testado para a remoção de ruídos. Por fim, o algoritmo não é testado em uma trajetória e não é feita uma análise da eficiência do método;

A dissertação de mestrado apresentada por Silva et al. (2015) utiliza o método para registro automático de imagens apresentado nesta tese de doutorado (publicado no artigo científico Braga et al. (2016)), mas aplicado para a determinação da posição geográfica do VANT utilizando imagens de dois sensores distintos: a imagem de satélite georreferenciada é obtida por sensor óptico, que captura imagens na banda do visível; as imagens capturadas pelo VANT são obtidas por um sensor infravermelho termal. Em sua dissertação de mestrado, (SILVA et al., 2015) conclui que a extração de bordas, inserida no processo de registro automático de imagens, é essencial para evitar a influência da variação da iluminação do ambiente e as diferenças entre as imagens de diferentes sensores no resultado final da estimação da posição do VANT. Na dissertação de mestrado de Silva et al. (2015), os melhores resultados do registro automático de imagens são obtidos quando aplica-se a MLP para a detecção de bordas das imagens.

A análise do método proposto para estimar a posição de um VANT, por meio do registro automático de imagens, considera três situações de voo diferentes:

1 - **Situação de voo** 1 - o VANT sobrevoa uma região urbana em condições suficientes de iluminação do ambiente para que seja aplicado a câmera de vídeo como sensor imageador

2 - **Situação de voo** 2 - o VANT sobrevoa uma região urbana com pouca ou nenhuma iluminação do ambiente, nesta situação, utiliza-se imagens de intensidades provenientes de um sensor LiDAR;

3 - **Situação de voo** 3 - o VANT sobrevoa uma região coberta por água, por exemplo o oceano, nesta situação, utiliza-se imagens do MDS da cobertura do fundo do oceano;

Para a avaliação do algoritmo, nas três situações de voo citadas, são considerados diferentes experimentos.

O primeiro experimento realizado, que compreende a situação de voo 1, utiliza um conjunto de dados de imagens obtida por uma câmera de vídeo de um VANT e uma imagem georreferenciada. Todas as imagens foram capturadas na banda do visível. Detalhes sobre o conjunto de dados são apresentados na Seção 6.6.2. O primeiro experimento tem o objetivo de testar as seguintes hipóteses:

- primeira hipótese: O método de registro automático de imagens configurado com a MLP, de arquitetura determinada pelo MPCA, produz resultados melhores, em relação à estimação da posição do VANT, que o método configurado com os algoritmos Canny e Sobel;
- segunda hipótese: o algoritmo de registro automático de imagens configurado com a MLP, implementada em FPGA e utilizando uma estratégia em LUT, produz os mesmos resultados que o algoritmo configurado com a MLP de arquitetura determinada pelo MPCA.
- terceira hipótese: o algoritmo de registro automático de imagens configurado com a MLP implementada em FPGA produz resultados melhores que o algoritmo configurado com os operadores de detecção de bordas Canny e Sobel;
- quarta hipótese: o algoritmo de registro automático de imagens configurado com a MLP implementada em FPGA, utilizando uma estratégia em LUT, tem melhor desempenho computacional que o algoritmo configurado com os operadores Canny e Sobel;

Além disso, o primeiro experimento avalia qual processo de filtragem das imagens produz o melhor resultado na estimação da posição do VANT por registro automático de imagens.

O segundo experimento realizado, relacionado com a situação de voo 2, utiliza um conjunto de dados formado por uma imagem de intensidade LiDAR georreferenciada, de uma região urbana, um conjunto de imagens de intensidade LiDAR, que representam as imagens capturadas pelo VANT durante sua trajetória, e uma imagem de satélite georreferenciada, da mesma região urbana, obtida por um sensor óptico passivo que captura imagens na banda do visível. Mais detalhes sobre o conjunto de dados são apresentados na seção 6.6.3 . O segundo experimento tem o objetivo de testar as seguintes hipóteses:

- primeira hipótese: o método de registro automático proposto é adequado para estimar a posição do VANT utilizando imagens de intensidade de regiões urbanas obtidas por um sensor LiDAR;
- segunda hipótese: considerando o conjunto de imagens de intensidade LiDAR, o algoritmo de registro automático de imagens configurado com a MLP implementada em FPGA, utilizando uma estratégia em LUT, produz resultados melhores que o algoritmo configurado com os operadores Canny e Sobel.
- terceira hipótese: é possível aplicar o método proposto de registro automático entre imagens de intensidade LiDAR e imagens de satélite na banda do visível, para se determinar a posição geográfica do VANT.

O segundo experimento também tem como objetivo avaliar qual método de filtragem é mais adequado para ser aplicado no processo de estimação da posição geográfica do VANT utilizando imagens de intensidade LiDAR.

O terceiro experimento realizado, relacionado com a situação de voo 3, utiliza um MDS da cobertura do fundo do oceano para realizar o registro automático de imagens e, com isto, obter a posição geográfica do VANT. Mais detalhes sobre o conjunto de dados do terceiro experimento são apresentados na seção 6.6.4.

O terceiro experimento tem o objetivo de avaliar as seguintes hipóteses:

- primeira hipótese: o método de registro automático de imagens proposto é adequado para ser aplicado em MDS da cobertura do fundo do oceano, obtido por um sensor LiDAR;
- segunda hipótese: considerando o conjunto de imagens do MDS do fundo do oceano, o algoritmo de registro automático de imagens configurado com a

MLP implementada em FPGA, utilizando uma estratégia em LUT, produz resultados melhores que o algoritmo configurado com os operadores Canny e Sobel.

Outro objetivo do terceiro experimento é determinar o algoritmo de filtragem mais adequado para a remoção de ruídos nas imagens do MDS do fundo do oceano, inserido no método de registro automático de imagens, para estimar a posição geográfica do VANT.

6.6.1 RESULTADOS: MÉTRICAS DE AVALIAÇÃO

Três métricas foram aplicadas para a avaliação da estimativa da posição obtida pelo algoritmo de registro automático de imagens proposto. Estas métricas foram aplicadas nas pesquisas desenvolvidas por Conte e Doherty (2008) e Braga et al. (2016).

A primeira métrica, denominada Erro de Boa Correspondência (EGM, do inglês, Error Good Matching), compara a posição estimada, obtida pelo algoritmo de registro automático de imagens, com a posição estimada, fornecida pelo sistema GNSS da aeronave. A posição estimada pelo GNSS da aeronave é utilizada como verdade sobre o posicionamento do VANT. Se o erro (distância euclidiana) entre os posicionamentos obtidos for superior a 5 *metros*, a estimativa de posição fornecida pelo algoritmo de registro de imagens não é classificada como um EGM. Entretanto, se o erro entre os posicionamentos for inferior a 5 *metros*, a estimativa fornecida pelo algoritmo é classificada como EGM. O valor 5 *metros* de limite para a classificação de uma estimativa de posição como EGM é utilizado, pois corresponde a metade do valor máximo de erro do sistema GNSS embarcado no VANT (CONTE; DOHERTY, 2008).

No segundo e terceiro experimento, a trajetória do VANT foi obtida por simulação em computador, diferentemente do primeiro experimento, em que foi obtida por meio de um voo real da aeronave. Desta forma, o erro da posição estimada pelo algoritmo de registro automático de imagens é calculado utilizando a trajetória da simulação em computador como referência, mas o valor de 5 *metros* de limite para a classificação de uma estimativa de posição como EGM também é aplicado.

A segunda métrica, denominada Desvido Padrão de Boa Correspondência (SDGM, do inglês, Standard Deviation Good Matching), avalia a estabilidade do método de estimativa de posição aplicado. Esta métrica corresponde ao valor do desvio padrão

do erro (distância euclidiana) dos 30 últimos resultados de estimação da posição do VANT. Se o valor do desvio padrão do erro for superior a $2metros$, o posicionamento fornecido pelo sistema não é estável e confiável, desta forma, a estimação não é considerada SDGM. Entretanto, em áreas onde a estimação é confiável e estável, o desvio padrão do erro das últimas 30 estimações é inferior a 2 $metros$. (CONTE; DOHERTY, 2008).

A terceira métrica é denominada Boa Correspondência (GM, do inglês, Good Matching). Quando uma estimativa de posição é classificada como EGM e SDGM esta é classificada como GM.

6.6.2 RESULTADOS: PRIMEIRO EXPERIMENTO

O conjunto de dados do primeiro experimento é formado por uma imagem georreferenciada e um conjunto de imagens obtidas por um VANT em tempo de voo. Este conjunto de dados são provenientes de um voo real de um VANT. A imagem georreferenciada é da mesma região sobrevoada pela aeronave.

A imagem georreferenciada aplicada neste trabalho representa uma área de aproximadamente 1 km^2. A região imageada é utilizada para o treinamento de bombeiros e localiza-se no sul da Suécia. Esta imagem é monocromática e sua resolução espacial é de $0,5\ \frac{metros}{pixel}$. A Figura 6.17 exibe a imagem georreferenciada do conjunto de dados do primeiro experimento.

Figura 6.17 - Imagem georreferenciada do conjunto de dados do primeiro experimento.

Fonte: Produção do autor.

As imagens aéreas foram adquiridas por um helicóptero autônomo, apresentado na Figura 6.18, chamado RMAX e fabricado pela Yamaha Motor Company. O helicóptero voou a uma velocidade de 3 m/s e a 60 $metros$ de altitude acima do solo.

Figura 6.18 - Helicóptero autônomo RMAX.

Fonte: Produção do autor.

A câmera de vídeo foi fixada no helicóptero com visada perpendicular ao solo e captura imagens da região sobrevoada a uma taxa de 25 Hz. Devido a aeronave manter a mesma altitude em toda a trajetória a resolução espacial das imagens capturada pelo VANT é igual a $0,12\ \frac{metros}{pixel}$ e têm 288 $\times$ 360 pixels, que correspondem a uma área de aproximadamente 1540 m^2. A Figura 6.19 exibe uma imagem capturada pela câmera fixada ao VANT.

Figura 6.19 - Imagem capturada pela câmera do helicóptero autônomo RMAX.

Fonte: Produção do autor.

A Figura 6.20 exibe a imagem georreferenciada com a trajetória seguida pelo helicóptero marcada com a linha vermelha, o círculo em cor preta indica o ponto de partida e o círculo em cor branca o ponto de chegada do VANT. Esta trajetória possui aproximadamente 1 km de extensão e produziu 1443 (imagens capturadas pelo VANT) pontos para avaliar a estimativa da posição do VANT obtida pelo método de registro automático de imagens.

Figura 6.20 - Trajetória do VANT.

Fonte: Produção do autor

Neste primeiro experimento, a incerteza, que é aplicada para determinar o tamanho da área de correlação, é igual a 10 $metros$. Este valor representa o valor da incerteza do INS sobre o posicionamento do VANT no instante da falha do sinal do GNSS. A incerteza do INS é igual ao erro máximo do GNSS no momento de sua falha

(CONTE; DOHERTY, 2008). Com este valor de incerteza, a área de correlação abrange 26600 m^2.

As combinações utilizadas para a produção dos resultados do primeiro experimento são apresentadas na Figura 6.21.

Figura 6.21 - Combinações de algoritmos aplicadas para a produção dos resultados.

Fonte: Produção do autor

6.6.2.1 RESULTADOS: CANNY E FILTRO GAUSSIANO

Nesta seção, são apresentado o resultados da estimação da posição do VANT, obtidos pelo método de registro automático de imagens configurado com o operador Canny, para a detecção de bordas, e o filtro Gaussiano na filtragem dos ruídos das imagens. Estes algoritmos são aplicados na imagem georreferenciada e nas imagens aéreas capturadas pela câmera de vídeo do VANT.

A Figura 6.22 apresenta o erro da estimação da posição do VANT, obtido pelo algoritmo de registro automático de imagens para cada ponto de avaliação da trajetória, ou seja, para cada um dos 1443 pontos de avaliação. A linha tracejada em vermelho representa o erro igual a 5 *metros*, pontos abaixo dessa linha são classificados como EGM.

Figura 6.22 - Erro da estimação da posição, obtido pelo algoritmo de registro automático de imagens, em cada um dos 1443 pontos de avaliação. Os algoritmos Canny e filtro gaussiano foram utilizados no algoritmo de registro automático de imagens.

Fonte: Produção do autor

O gráfico na Figura 6.23 exibe o valor do desvio padrão do erro para as 30 últimas posições estimadas. Como o desvio padrão do erro é calculado sobre as 30 últimas estimações, o 31° ponto de avaliação da trajetória é o primeiro a ter o desvio padrão do erro calculado. A linha tracejada representa um desvio padrão de 2 *metros*, posições estimadas abaixo dessa linha são classificadas como SDGM.

Figura 6.23 - Desvio padrão do erro da estimação da posição obtido pelo algoritmo de registro automático de imagens configurado com o Canny e o filtro Gaussiano.

Fonte: Produção do autor

A Figura 6.24 apresenta a trajetória real e a trajetória estimada pelo algoritmo de registro automático de imagens. Apenas os pontos classificados como EGM são exibidos na Figura 6.24. O algoritmo de registro de imagens utilizou o operador Canny para a detecção de bordas e o filtro Gaussiano para remoção dos ruídos nas imagens.

Figura 6.24 - Trajetória original do VANT, em vermelho, e a trajetória estimada, em azul. Apenas pontos classificados como EGM são exibidos na trajetória estimada.

Fonte: Produção do autor.

A Tabela 6.3 apresenta os resultados das quantidades de posições estimadas classificadas como EGM, SDGM e GM.

Tabela 6.3 - Resultados das quantidades de posições estimadas classificadas como EGM, SDGM e GM

Variáveis	Quantidades
EGM	414
SDGM	934
GM	263

Para esta configuração do algoritmo de registro automático de imagens, que utilizou o operador Canny para a detecção de bordas e o filtro Gaussiano para remoção de ruídos nas imagens, $28,6\%$ dos pontos de avaliação foram classificados como EGM, $66,7\%$ como SDGM e $18,6\%$ como GM.

6.6.2.2 RESULTADOS: SOBEL E FILTRO GAUSSIANO

A presente seção exibe o resultado da estimação da posição do VANT, obtido pelo método de registro automático de imagens configurado com operador Sobel, para a detecção de bordas, e o filtro Gaussiano na filtragem dos ruídos das imagens. Os algoritmos Sobel e filtro Gaussiano são aplicados na imagem georreferenciada e nas imagens aéreas capturadas pela câmera de vídeo do VANT.

A Figura 6.25 apresenta o erro da estimação da posição do VANT, obtido pelo algoritmo de registro automático de imagens, para cada um dos 1443 pontos de avaliação. A linha tracejada em vermelho representa o erro igual a 5 *metros*, pontos abaixo dessa linha são classificados como EGM.

Figura 6.25 - Erro da estimação da posição obtido pelo algoritmo de registro automático de imagens configurado com o Sobel e o filtro Gaussiano.

Fonte: Produção do autor.

A Figura 6.26 exibe o gráfico do desvio padrão do erro das posições estimadas pelo algoritmo de registro automático de imagens. O desvio padrão do erro é calculado para as 30 últimas posições estimadas. A linha tracejada em vermelho representa um desvio padrão de 2 *metros*, posições estimadas abaixo dessa linha são classificadas como SDGM.

Figura 6.26 - Desvio padrão do erro das posições estimadas, obtidas pelo algoritmo de registro automático de imagens configurado com o Sobel e o filtro Gaussiano.

Fonte: Produção do autor.

A Figura 6.27 apresenta a trajetória obtida pelo pontos classificados como EGM (em azul), para a configuração do algoritmo de registro automático de imagens com o Sobel e o filtro Gaussiano. Apenas pontos classificados como EGM são exibidos na trajetória estimada. Na Figura 6.27 também é apresentada a trajetória real do VANT em vermelho, o círculo em preto marca o ponto de partida e o círculo branco o ponto de chegada do VANT.

Figura 6.27 - Trajetória original do VANT, em vermelho, e a trajetória estimada em azul. A trajetória estimada é formada por pontos classificados como EGM, e foi obtida pelo algoritmo de registro automático de imagens configurado com o Sobel e o filtro Gaussiano.

Fonte: Produção do autor.

A Tabela 6.4 apresenta os resultados das quantidades de posições estimadas classificadas como EGM, SDGM e GM, considerando a utilização dos algoritmos Sobel e filtro Gaussiano no algoritmo de registro automático de imagens.

Tabela 6.4 - Resultados das quantidades de posições estimadas classificadas como EGM, SDGM e GM

Variáveis	Quantidades
EGM	716
SDGM	939
GM	470

Para esta configuração do algoritmo de registro automático de imagens, $49,6\%$ dos pontos de avaliação foram classificados como EGM, $66,4\%$ como SDGM e $33,2\%$ como GM.

6.6.2.3 RESULTADOS: MLP E FILTRO GAUSSIANO

Esta seção apresenta o resultado da estimação da posição do VANT, obtido pelo método de registro automático de imagens configurado com a MLP e o filtro gaussiano. A MLP e o filtro Gaussiano são aplicados tanto na imagem georreferenciada quanto nas imagens aéreas capturadas pela câmera de vídeo da aeronave. A arquitetura da MLP, aplicada para a detecção de bordas, foi obtida pelo algoritmo MPCA e é exibida na Tabela 6.1.

A Figura 6.28 exibe o gráfico do o erro, obtido pelo algoritmo de registro automático de imagens, para as posições estimadas do VANT. O gráfico do erro é em função de cada um dos 1443 pontos de avaliação da trajetória. A linha tracejada em vermelho representa o erro igual a 5 *metros*, pontos abaixo desta linha são classificados como EGM. O algoritmo de registro automático de imagens foi configurado com a fase de ativação da MLP, com arquitetura definida pelo MPCA, para a detecção de borda e o filtro gaussiano, para a remoção de ruídos nas imagens.

Figura 6.28 - Gráfico do erro da estimação da posição, obtido pelo algoritmo de registro automático de imagens configurado com a fase de ativação MLP e o filtro gaussiano.

Fonte: Produção do autor.

A MLP, cuja a arquitetura foi determinada pelo MPCA, foi implementada em FPGA, utilizando a linguagem VHDL, por meio da estratégia de LUT (MLP em LUT) (detalhes da implementação estão presentes na seção 6.4.4.6). Esta abordagem foi utilizada para a detecção de bordas no lugar da fase de propagação da MLP, e os resultados do erro de estimação da posição do VANT, obtidos por esta configuração do algoritmo de registro de imagens, são exibidos na Figura 6.29.

Figura 6.29 - Gráfico do erro da estimação da posição, obtido pelo algoritmo de registro automático de imagens, em cada um dos 1443 pontos de avaliação. A MLP em LUT e o filtro gaussiano foram utilizados no algoritmo de registro de imagens. Pontos abaixo da linha vermelha tracejada são classificados como EGM.

Fonte: Produção do autor.

Os resultados do algoritmo de registro automático de imagens, para obtenção da posição geográfica do VANT, são os mesmos quando se aplica a fase de propagação da MLP, com arquitetura determinada pelo MPCA, ou quando se utiliza a MLP implementada em FPGA com a estratégia de LUT (MLP em LUT).

A Figura 6.30 exibe o gráfico do desvio padrão do erro, para as 30 últimas posições estimadas, em cada um dos 1443 pontos de avaliação. No mesmo gráfico estão representados os resultados obtidos com a fase de propagação da MLP de arquitetura determinada pelo MPCA (linha azul) e com a MLP em LUT (linha verde). Os resultados para ambas as implementações são os mesmos, mas para evitar a sobreposição dos gráficos o resultado da MLP em LUT (linha verde) está deslocado no eixo y

em 4 metros. As linhas tracejadas, em vermelho, representam um desvio padrão de 2 metros e de 6 metros (desvio padrão de 2 metros deslocado 4 metros), posições estimadas abaixo dessas linhas são classificadas como SDGM.

Figura 6.30 - Gráfico do desvio padrão do erro, obtido pelo algoritmo de registro de imagens configurado com a MLP de arquitetura determinada pelo MPCA, em azul, e a MLP em LUT, em verde. O filtro Gaussiano foi aplicado para a remoção de ruídos.

Fonte: Produção do autor.

A Figura 6.31 apresenta a trajetória dos pontos classificados como EGM, para a configuração do algoritmo de registro automático de imagens com a MLP em LUT, para a detecção de bordas, e o filtro Gaussiano, para a remoção de ruídos indesejados. Os resultados de estimação da trajetória do VANT, obtidos com o algoritmo de registro de imagens configurado com a fase de ativação da MLP de arquitetura determinada pelo MPCA, ou com a MLP em LUT, são os mesmos. Por este motivo, para evitar a sobreposição na Figura 6.31, apenas o resultado do pontos classificados como EGM, do algoritmo de registro de imagens configurado com a MLP em LUT, é exibido na imagem. A trajetória real do VANT também é exibida na cor vermelha.

Figura 6.31 - Trajetória original do VANT, em vermelho, e trajetória estimada pelo algoritmo de registro automático de imagens cofigurado com a MLP em LUT e o filtro Gaussiano, em azul.

Fonte: Produção do autor.

A Tabela 6.5 apresenta o resultado das métricas EGM, SDGM e GM do algoritmo de registro automático de imagens configurado com a MLP em LUT, para a detecção de borda, e o filtro Gaussiano, para a remoção de ruídos.

Tabela 6.5 - Resultados das quantidades de posições estimadas classificadas como EGM, SDGM e GM

Variáveis	Quantidades
EGM	892
SDGM	1317
GM	808

A configuração do algoritmo de registro automático de imagens que utilizou o operador MLP em LUT, para a detecção de bordas, e o filtro Gaussiano, para remoção de ruídos nas imagens, obteve $61,8\%$ dos pontos de avaliação classificados como EGM, $93,2\%$ como SDGM e $57,2\%$ como GM.

6.6.2.4 RESULTADOS: CANNY E FILTRO DA MEDIANA

Esta seção apresenta o resultado da estimação da posição do VANT, obtido pelo método de registro automático de imagens proposto, quando configurado com o

algoritmo Canny, para a detecção de bordas, e o filtro da Mediana, para a remoção de ruídos das imagens. Os algoritmos Canny e filtro da Mediana são aplicados na imagem georreferenciada e nas imagens aéreas capturadas pela câmera de vídeo da aeronave.

A Figura 6.32 exibe o gráfico do erro, em função de cada ponto um dos 1443 pontos de avaliação, obtido pelo algoritmo de registro automático de imagens para a estimação da posição do VANT. A linha tracejada em vermelho representa o erro igual a 5 *metros*, pontos abaixo desta linha são classificados como EGM.

Figura 6.32 - Gráfico do erro da estimação da posição, obtido pelo algoritmo de registro automático de imagens configurado com o Canny e o filtro da Mediana.

Fonte: Produção do autor.

A Figura 6.26 exibe o gráfico do valor do desvio padrão do erro das posições estimadas pelo algoritmo de registro automático de imagens. O cálculo do desvio padrão do erro considera as 30 posições estimadas anteriores. A linha tracejada representa um desvio padrão de 2 *metros*, posições estimadas abaixo dessa linha são classificadas como SDGM.

Figura 6.33 - Gráfico do desvio padrão do erro da posição estimada pelo algoritmo de registro automático de imagens configurado com o Canny e o filtro da Mediana.

Fonte: Produção do autor.

A Figura 6.34 apresenta a trajetória obtida pelo algoritmo de registro automático de imagens, com a configuração abordada nesta seção (em azul). Apenas os pontos classificados como EGM são exibidos na trajetória estimada. A Figura 6.34 também exibe a trajetória real do VANT em vermelho.

Figura 6.34 - Trajetória original do VANT, em vermelho, e a trajetória estimada pelo algoritmo de registro automático de imagens configurado com o Canny e o filtro da Mediana, em azul

Fonte: Produção do autor.

A Tabela 6.6 exibe os resultados das quantidades de posições estimadas classificadas como EGM, SDGM e GM, considerando a utilização dos algoritmos Canny e do filtro da Mediana no método proposto de registro automático de imagens.

Tabela 6.6 - Resultados das quantidades de posições estimadas classificadas como EGM, SDGM e GM

Variáveis	Quantidades
EGM	608
SDGM	1219
GM	512

Para esta configuração do algoritmo de registro automático de imagens, que utilizou o operador Canny para a detecção de bordas e o filtro Mediana para remoção de ruídos nas imagens, $42,1\%$ dos pontos de avaliação foram classificados como EGM, $86,3\%$ como SDGM e $36,2\%$ como GM.

6.6.2.5 RESULTADOS: SOBEL E FILTRO DA MEDIANA

Esta seção exibe o resultado da estimação da posição do VANT, obtido pelo método de registro automático de imagens configurado com o operador Sobel, para a detecção de bordas, e o filtro da Mediana, para a remoção de ruídos nas imagens. Os algoritmos Sobel e filtro da Mediana são aplicados na imagem georreferenciada e nas imagens aéreas.

A Figura 6.35 exibe o gráfico do erro, de cada posição estimada, obtido pelo algoritmo de registro automático de imagens. A linha tracejada, em vermelho, representa o erro igual a 5 *metros*, pontos abaixo desta linha são classificados como EGM.

Figura 6.35 - Gráfico do erro da estimação da posição, obtido pelo algoritmo de registro automático de imagens configurado com o Sobel e o filtro da Mediana.

Fonte: Produção do autor.

A Figura 6.39 apresenta o gráfico do desvio padrão do erro das posições estimadas pelo algoritmo de registro automático de imagens. O desvio padrão do erro é calculado sobre as 30 posições estimadas anteriormente. A linha tracejada, em vermelho, representa um desvio padrão de 2 *metros*, posições estimadas abaixo dessa linha são classificadas como SDGM.

Figura 6.36 - Gráfico do desvio padrão do erro das posições estimadas pelo algoritmo de registro automático de imagens configurado com o Sobel e o filtro da Mediana.

Fonte: Produção do autor.

A Figura 6.37 apresenta a trajetória estimada, obtida pelo algoritmo de registro automático de imagens configurado com o Sobel e com filtro da Mediana. A trajetória estimada, em azul, é formada pelos pontos classificados como EGM. A Figura 6.34 também exibe a trajetória real do VANT, em vermelho.

Figura 6.37 - Trajetória estimada pelo algoritmo de registro automático de imagens, em azul, e trajetória real do VANT em vermelho. Somente os pontos classificados como EGM são exibidos na trajetória estimada.

Fonte: Produção do autor

A Tabela 6.7 apresenta os resultados das quantidades de posições estimadas classificadas como EGM, SDGM e GM, para o algoritmo de registro automático de imagens configurado com o Sobel e o filtro da Mediana.

Tabela 6.7 - Resultados das quantidades de posições estimadas classificadas como EGM, SDGM e GM.

Variáveis	Quantidades
EGM	879
SDGM	1267
GM	777

Para esta configuração do algoritmo de registro automático de imagens, $60,9\%$ dos pontos de avaliação foram classificados como EGM, $89,7\%$ como SDGM e $54,9\%$ como GM.

6.6.2.6 RESULTADOS: MLP E FILTRO DA MEDIANA

A seção 6.6.2.3 demonstrou que a MLP em LUT e a fase de propagação da MLP, com arquitetura determinada pelo MPCA, obtiveram os mesmos resultados . Desta forma, os resultados desta seção referem-se ao algoritmo de registro automático de imagens configurado com a MLP em LUT, para a detecção de borda, e com o filtro da Mediana, para a remoção dos ruídos nas imagens.

A Figura 6.38 exibe o gráfico do erro, de cada posição estimada do VANT, obtido pelo algoritmo de registro automático de imagens configurado com a MLP em LUT e o filtro da Mediana. A linha tracejada em vermelho representa o erro igual a 5 *metros*, pontos abaixo desta linha são classificados como EGM.

Figura 6.38 - Gráfico do erro da estimação da posição, obtido pelo algoritmo de registro automático de imagens configurado com a MLP em LUT e o filtro da Mediana.

Fonte: Produção do autor.

A Figura 6.39 apresenta o gráfico do desvio padrão do erro das posições estimadas. O desvio padrão é calculado considerando o erro das 30 posições estimadas anteriormente. A linha tracejada representa um desvio padrão de 2 *metros*, posições estimadas abaixo dessa linha são classificadas como SDGM.

Figura 6.39 - Gráfico do desvio padrão do erro das posições estimadas pelo algoritmo de registro automático de imagens configurado com a MLP em LUT e o filtro da Mediana.

Fonte: Produção do autor.

A Figura 6.40 apresenta a trajetória obtida pelas estimações de posição da aeronave classificadas como EGM, em azul. Para este resultado, o algoritmo de registro automático de imagens utilizou a MLP em LUT, para a extração de bordas, e o filtro da Mediana para remoção de ruídos indesejáveis nas imagens. A Figura 6.40 também exibe a trajetória real do VANT, em vermelho.

Figura 6.40 - Trajetória estimada pelo algoritmo de registro automático de imagens, em azul, e a trajetória real do VANT, em vermelho. Somente os pontos classificados como EGM são exibidos na trajetória estimada.

Fonte: Produção do autor.

A Tabela 6.8 exibe os resultados das quantidades de posições estimadas classificadas como EGM, SDGM e GM, considerando a aplicação da MLP em LUT e do filtro da Mediana no método proposto de registro automático de imagens.

Tabela 6.8 - Resultados das quantidades de posições estimadas classificadas como EGM, SDGM e GM.

Variáveis	Quantidades
EGM	926
SDGM	1348
GM	856

Esta configuração do algoritmo de registro automático de imagens, que utilizou o MLP em LUT para a detecção de bordas e o filtro da Mediana, atingiu $64,2\%$ dos

pontos de avaliação classificados como EGM, $95,4\%$ como SDGM e $60,6\%$ como GM.

6.6.2.7 RESULTADO: TEMPO DE EXECUÇÃO

Nesta seção apresenta-se o resultado do tempo de execução de cada etapa do método de registro automático de imagens, aplicado para estimar a posição geográfica do VANT. O algoritmo foi implementado em linguagem C com auxílio da biblioteca OpenCV para realizar a leitura, escrita e processamento das imagens. Duas plataformas computacionais, que podem ser embarcadas, foram utilizadas: a Raspberry Pi Model B-1 e a Zybo ZINQ 7000 (para mais informações sobre estes computadores, vide capítulo 3).

Os tempos medidos, apresentados nesta seção, são obtidos para uma incerteza de 10 metros (incerteza do INS sobre a posição do VANT, informações na introdução deste capítulo), o que resulta em uma área de correlação de 110×90 pixels. O tamanho da imagem obtida pela câmera do VANT, após o ajuste de escala, é igual a 60×84 pixels.

As etapas principais do algoritmo de registro automático de imagens e sobre as quais foram efetuadas as medições de tempo de execução são: ajuste de escala e rotação da imagem do VANT em relação a imagem georreferenciada; filtragem da imagem do VANT; seleção da área de correlação na imagem georreferenciada; detecção de bordas na imagem do VANT e e correlação entre as imagens. A imagem georreferenciada pode ser armazenada no computador embarcado filtrada (sem ruídos) e com as bordas detectadas, por isso, o tempo de execução dessas etapas não são medidos.

A Tabela 6.9 exibe o tempo de execução, na Raspberry Pi, das etapas citadas no parágrafo anterior. O tempo de execução de detecção de bordas nas imagens do VANT é exibido na Tabela 6.12.

Tabela 6.9 - Tempo de execução na Raspberry PI

Etapa do Algoritmo	Tempo de Execução (segundos)
Ajuste de Escala	0,0085
Ajuste de Rotação	0,0023
Filtro Gaussiano	0,0110
Filtro da Mediana	0,0110
Seleção da Área de Correlação	0,0002
Correlação entre Imagens	0,0832

Todas as etapas do algoritmo de registro automático de imagens, apresentadas na Tabela 6.9, foram executadas na CPU do computador Raspberry PI.

A Tabela 6.12 apresenta o tempo de execução dos algoritmos para a detecção de bordas nas imagens do VANT após o ajuste de escala. Os algoritmos Canny, Sobel e a MLP MPCA (fase de ativação da MLP com arquitetura determinada pelo MPCA) foram executadas na CPU do computador Raspberry Pi. Entretanto, a MLP em LUT, foi implementada em VHDL e executada na FPGA Spartan 6 do computador Raspberry PI.

Tabela 6.10 - Tempo de execução dos algoritmos para a detecção de bordas na Raspberry PI

Algoritmo	Tempo de Execução (segundos)
Canny - CPU	**0,074**
Sobel - CPU	0,083
MLP MPCA (Tabela 6.1) - CPU	1,684
MLP em LUT - FPGA	0,587

A Tabela 6.11 exibe o tempo de execução das etapas de ajuste de escala e rotação, filtragem, seleção da área de correlação e a correlação entre as imagens, executadas no processador do computador Zybo ZINQ 7000.

Tabela 6.11 - Tempo de execução na Zybo ZINQ 7000

Etapa do Algoritmo	Tempo de Execução (em segundos)
Ajuste de Escala	0,0064
Ajuste de Rotação	0,0021
Filtro Gaussiano	0,0107
Filtro da Mediana	0,0107
Seleção da Área de Correlação	0,0002
Correlação entre Imagens	0,0665

A Tabela 6.12 apresenta o tempo de execução dos algoritmos para a detecção de bordas nas imagens do VANT, após o ajuste de escala. Os algoritmos Canny, Sobel e a MLP MPCA foram executadas na CPU do computador Zynq ZINQ 7000. Porém,

a MLP em LUT, foi implementada em VHDL e executada na FPGA Artix 7 do computador Zybo ZINQ 7000.

Tabela 6.12 - Tempo de execução dos algoritmos para a detecção de bordas na Zybo ZINQ 7000

Algoritmo	Tempo de Execução (em segundos)
Canny - CPU	0,0601
Sobel - CPU	0,0793
MLP MPCA (Tabela 6.1) - CPU	1,425
MLP em LUT - FPGA	**0,016**

Os resultados apresentados nas Tabelas 6.11 e 6.12 demonstram que, no computador Zybo ZINQ 7000, após as implementação da MLP em LUT, a etapa do algoritmo de registro de imagens com pior desempenho é a correlação cruzada entre as imagens.

A Tabela 6.13 apresenta o número de estimações por segundo, obtido pelo algoritmo de registro automático de imagens configurado com os detectores de borda e executados na Raspberry Pi.

Tabela 6.13 - Número de estimações por segundo, obtido pelo algoritmo de registro automático de imagens configurado com cada detector de borda. Este resultado considera a execução realizada na Raspberry Pi.

Algoritmo	Estimações por segundo
Canny - CPU	**6**
Sobel - CPU	5
MLP MPCA (Tabela 6.1) - CPU	1 a cada 2 segundos
MLP em LUT - FPGA	1

A Tabela 6.13 apresenta o número de estimações por segundo, obtido pelo algoritmo de registro automático de imagens configurado com os detectores de borda e executados na ZYBO ZINQ 7000.

Tabela 6.14 - Número de estimações por segundo, obtido pelo algoritmo de registro automático de imagens configurado com cada detector de borda. Este resultado considera a execução realizada na ZYBO ZINQ 7000.

Algoritmo	Estimações por segundo
Canny - CPU	7
Sobel - CPU	6
MLP MPCA (Tabela 6.1) - CPU	1 a cada $1,8$ segundos
MLP em LUT - FPGA	**10**

6.6.2.8 RESULTADOS: ANÁLISE DOS RESULTADOS DO PRIMEIRO EXPERIMENTO

No primeiro experimento, considerando o conjunto de dados utilizado, a análise dos resultados da estimação da posição geográfica do VANT demonstram que a melhor configuração do método de registro automático de imagens é a que emprega a MLP, com arquitetura determinada pelo MPCA, para a detecção de borda nas imagens aéreas, capturadas pela câmera do VANT, e na imagem georreferenciada. A MLP obteve resultados melhores na estimação da posição geográfica do VANT que os operadores Canny e Sobel em todos os testes, demonstrando que é verdadeira a primeira hipótese.

Os resultados do primeiro experimento também evidenciam que a MLP implementada em FPGA, utilizando uma estratégia em LUT (MLP em LUT), reproduz os mesmos resultados que a fase de propagação da MLP com arquitetura determinada pelo MPCA, demonstrando que a segunda e a terceira hipótese estão corretas. Além disso, a análise do tempo de execução, para o computador ZYBO ZINQ 7000, demonstram que a MLP em LUT tem desempenho melhor que os operadores Canny e Sobel no processo de detecção de bordas.

Além disso, a análise dos resultados do primeiro experimento demonstram que o filtro da Mediana obtém resultados melhores que o filtro Gaussiano, quando aplicado no algoritmo de registro automático de imagens para a estimação da posição geográfica do VANT. Desta forma, o filtro da Mediana, inserido no método proposto de registro automático de imagens, é mais adequado para a remoção de ruídos nas imagens.

A Tabela 6.15 exibe os resultados das métricas EGM, SDGM e GM para os três algoritmos de detecção de bordas (Canny, Sobel e MLP em LUT). Para este resultado, o algoritmo de registro automático de imagens utilizou o filtro da mediana e

uma incerteza de 80 metros, que é a maior valor de incerteza possível do INS da aeronave (CONTE; DOHERTY, 2008).

Tabela 6.15 - Tempo de execução dos algoritmos para a detecção de bordas na Zybo ZINQ 7000

	EGM	SDGM	GM
Canny	197	64	26
Sobel	296	63	42
MLP em LUT	414	216	103

Os resultados apresentados na Tabela 6.15 evidenciam que o algoritmo de registro automático de imagens, configurado com a MLP em LUT e o filtro da mediana, obtém melhores resultados para a estimação da posição geográfica do VANT, mesmo utilizando uma incerteza de 80 metros para determinar o tamanho da área de correlação.

6.6.3 RESULTADOS: SEGUNDO EXPERIMENTO

O segundo experimento simula a estimação da posição do VANT por registro automático de imagens quando a aeronave sobrevoa uma região urbana em condições de baixa ou sem luminosidade.

O conjunto de dados do segundo experimento é formado por uma imagem de intensidade LiDAR georreferenciada, um conjunto de 172 imagens aéreas de intensidade LiDAR e uma imagem de satélite georreferenciada na banda do visível. Todas as imagens são da mesma região.

A imagem de intensidade georreferenciada foi capturada por um sensor LiDAR aerotransportado, que emite pulso de laser com comprimento de onda de 1.045 μm a 1.065 μm (infravermelho próximo). Esta imagem possui resolução espacial de 1 $\frac{metro}{pixel}$ e foi formada a partir de um processo de interpolação sobre a nuvem de pontos em três dimensões. Informações sobre métodos de interpolação para gerar imagens de intensidade LiDAR a partir de uma nuvem de pontos podem ser obtidas em Ashraf et al. (2017). A região sobrevoada corresponde a cidade de Baltimore, no estado Maryland - Estados Unidos da América (EUA).

Imagens de intensidade LiDAR são formadas pela quantidade de energia refletida

a partir da superfície reflexiva. A intensidade representa a quantidade de energia retornada ao sensor em comparação com a energia emitida. A imagem de intensidade LiDAR indica diferenças na absorção de energia devido à interação do pulso laser com a superfície dos materiais. Desta forma, a quantidade de intensidade de energia refletida por um alvo e capturada pelo sensor depende do material que forma a superfície do alvo, bem como, do comprimento de onda do pulso emitido (GORGENS et al., 2014).

A Figura 6.41 exibe a imagem de intensidade LiDAR georreferenciada, utilizada no segundo experimento.

Figura 6.41 - Imagem de intensidade LiDAR da cidade de Baltimore - EUA

Fonte: Adaptado de DCGISopendata (2017).

Cada uma das 172 imagens aéreas são subimagens da imagem LiDAR georreferenciada e representam a trajetória realizada pelo VANT. Portanto, estas imagens foram formadas pelo mesmo processo da imagem de intensidade LiDAR georreferenciada e possuem a mesma resolução espacial. Estas imagens representam os pontos de avaliação do método de registro automático de imagens proposto. As imagens têm 40×40 pixels e correspondem a uma área de 1600 m^2, estes valores para a dimensão das subimagens foram determinados baseando-se nos valores de dimensão das imagens capturadas pela aeronave no primeiro experimento, que são de um voo real de um VANT. A Figura 6.42 apresenta um exemplo da imagem de intensidade LiDAR da trajetória do VANT.

Figura 6.42 - Exemplo de imagem aérea de intensidade LiDAR da trajetória do VANT.

Fonte: Adaptado de DCGISopendata (2017).

A imagem de satélite georreferenciada foi capturada pelo sensor *Quickbird*, que na banda pancromática tem uma resolução espacial de 0.6 $\frac{metros}{pixel}$. A Figura 6.43 exibe a imagem de satélite georreferenciada da mesma região da cidade de Baltimore - EUA.

Figura 6.43 - Imagem de satélite georreferenciada da cidade de Baltimore, capturada na banda pancromática do sensor *Quickbird*

Fonte: Adaptado de DCGISopendata (2017).

A imagem de intensidade LiDAR georreferenciada e a imagem de satélite georreferenciada são de livre acesso e uso acadêmico e estão disponíveis em DCGISopendata (2017).

A Figura 6.44 exibe a trajetória do VANT, em vermelho, na imagem de intensidade LiDAR georreferenciada. Esta trajetória do VANT é proveniente de uma simulação em computador e é utilizada como verdade sobre o posicionamento da aeronave. A trajetória possui aproximadamente 1 km de extensão. O ponto de partida do VANT

é marcado pelo círculo amarelo e o ponto de chegada é marcado pelo círculo branco.

Figura 6.44 - Imagem de intensidade LiDAR georreferenciada com a trajetória do VANT em vermelho.

Fonte: Produção do autor.

A Figura 6.45 exibe a trajetória do VANT, em vermelho, na imagem georreferenciada do satélite Quickbird. Esta trajetória do VANT é uma simulação em computador e é aplicada como verdade sobre o posicionamento da aeronave. O ponto de partida do VANT é marcado pelo círculo amarelo e o ponto de é marcado pelo círculo branco.

Figura 6.45 - Imagem de satélite georreferenciada com a trajetória do VANT em vermelho.

Fonte: Produção do autor.

Neste segundo experimento, a incerteza aplicada para determinar o tamanho da área de correlação é igual a 10 *metros*. O valor é igual a incerteza do INS, sobre o posicionamento do VANT, no instante da falha do sinal do GNSS. A incerteza do

INS é igual ao erro máximo do GNSS no momento de sua falha (CONTE; DOHERTY, 2008). Aplicando este valor de incerteza a área de correlação abrange uma área 26600 m^2.

As combinações utilizadas para a produção dos resultados do segundo experimento são apresentadas na Figura 6.46.

Figura 6.46 - Combinações de algoritmos utilizadas para a produção dos resultados do segundo experimento.

Fonte: Produção do autor.

A fase de propagação da MLP, de arquitetura determinada pelo MPCA, não foi aplicada para a produção dos resultados apresentados no segundo experimento, pois nos teste do primeiro experimento a MLP em LUT obteve os mesmos resultados, mas com um tempo de execução melhor.

6.6.3.1 RESULTADOS: FILTRO GAUSSIANO COM ALGORITMOS DE DETECÇÃO DE BORDAS

Nesta seção são apresentados os resultados do algoritmo de registro automático de imagens, configurado com os algoritmos Canny, Sobel e MLP em LUT, para a detecção de borda, e com o filtro Gaussiano para a remoção de ruídos nas imagens de intensidade LiDAR.

Devido as imagens de intensidade LiDAR do VANT serem subimagens da imagem

de intensidade LiDAR georreferenciada, neste teste, foi acrescentado um ruído de $+ \; ou \; - \; 6\%$, no ajuste de escala entre as imagens, e um ruído de $+ \; ou \; - \; 6°$, no ajuste de rotação entre as imagens. Diferentemente do conjunto de dados do primeiro experimento, o conjunto de dados do segundo experimento não são dados reais de um voo de um VANT, portanto, o ruído é acrescentado ao ajuste de escala e rotação para oferecer maior realidade ao experimento.

Os valores de 6% no ruído de escala e 6° no ruído de rotação foram assim definidos, pois valores menores que este não apresentavam mudanças nas métricas de avaliação, o que não possibilitava diferenciar as configurações do algoritmo de registro automático de imagens. Este valor de ruído para a escala e rotação possibilita 4 combinações diferentes de resultados.

A Tabela 6.16 exibe a média das métricas EGM, SDGM e GM do algoritmo de registro automático de imagens, configurado com cada algoritmo de detecção de borda (Canny, Sobel, MLP em LUT) e para as 4 combinações possíveis de ruído de escala e rotação. O filtro gaussiano foi aplicado para a remoção de ruídos indesejáveis nas imagens.

Tabela 6.16 - Resultado da média das métricas EGM, SDGM e GM do algoritmo de registro automático de imagens configurado com cada detector de borda.

	EGM	SDGM	GM
Canny	158	107	103
Sobel	164	108	107
MLP em LUT	167	111	109

A porcentagem dos pontos de avaliação classificados como EGM, para a configuração utilizando filtro Gaussiano com os algoritmos de detecção de bordas Canny, Sobel e MLP em LUT foi respectivamente de $91,8\%$, $95,3\%$ e $97,1\%$. A porcentagem dos classificados como SDGM, utilizando filtro Gaussiano com o Canny, Sobel e MLP em LUT foi respectivamente de $75,3\%$, $76,1\%$ e $78,2\%$. A porcentagem de GM, utilizando filtro Gaussiano com o Canny, Sobel e MLP em LUT foi respectivamente de $72,5\%$, $75,3\%$ e $77,8\%$.

A Figura 6.47 exibe o gráfico do erro de cada umas das 172 posições estimadas do VANT, obtido por cada configuração do algoritmo de registro automático de

imagens. Em azul, o gráfico do erro do algoritmo de registro de imagens configurado com a MLP em LUT e com o filtro gaussiano. Em vermelho, o gráfico do erro do algoritmo de registro de imagens configurado com o Sobel e o filtro gaussiano. Em verde, o gráfico do erro do algoritmo de registro de imagens configurado com o Canny e o filtro gaussiano. A linha tracejada em vermelho representa um erro igual a 5 *metros*, pontos abaixo desta linha são classificados como EGM.

Figura 6.47 - Gráfico do erro da estimação da posição obtido pelo algoritmo de registro automático de imagens para cada um dos 172 pontos de avaliação.

Fonte: Produção do autor.

Na Figura 6.48 apresenta-se o gráfico do desvio padrão do erro em função de cada ponto de avaliação, obtido pelo algoritmo de registro automático de imagens configurado com cada algoritmo de detecção de borda (Canny, Sobel e MLP em LUT) e com o filtro gaussiano. O desvio padrão é calculado considerando o erro das 30 posições estimadas anteriormente. Em azul, o gráfico do desvio padrão do erro para a configuração com a MLP em LUT. Em vermelho, o gráfico do desvio padrão do erro para a configuração com o Sobel. Em verde, o gráfico do desvio padrão do erro para a configuração com o Canny. A linha tracejada em vermelho representa um desvio padrão do erro igual a 2 *metros*, posições estimadas abaixo dessa linha são classificadas como SDGM.

Figura 6.48 - Gráfico do desvio padrão do erro da estimação da posição do algoritmo de registro automático de imagens. O gráfico exibe o desvio padrão do erro de cada configuração do algoritmo de registro automático de imagens.

Fonte: Produção do autor.

A Figura 6.49 apresenta a trajetória obtida pelo algoritmo de registro de imagens das posições do VANT classificadas como EGM. Para este resultado, o algoritmo de registro automático de imagens foi configurado com a MLP em LUT, para a detecção de bordas, e o filtro gaussiano, para a remoção dos ruídos. A trajetória estimada pelo algoritmo de registro de imagens é ilustrada sobre a imagem de intensidade LiDAR georreferenciada na cor azul. A Figura 6.49 também exibe a trajetória simulada do VANT na cor vermelha, esta trajetória simulada é utilizada como verdade sobre a posição do VANT.

Figura 6.49 - Trajetória estimada pelo algoritmo de registro automático de imagens configurado com a MLP em LUT e o filtro gaussiano, em azul. A trajetória original do VANT é exibida em vermelho

Fonte: Produção do autor.

6.6.3.2 RESULTADOS: FILTRO DA MEDIANA COM ALGORITMO DE DETECÇÃO DE BORDAS

Esta seção exibe os resultados do algoritmo de registro automático de imagens, configurado com os operadores Canny, Sobel e MLP em LUT, para a detecção de borda, e com o filtro da Mediana, para a remoção de ruídos presentes nas imagens de intensidade LiDAR.

Pelo mesmo motivo apresentado na seção anterior (seção 6.6.3.1), neste teste, também foi acrescentado um ruído de $+ \; ou \; - \; 6\%$, no ajuste de escala entre as imagens, e um ruído de $+ \; ou \; - \; 6°$ no ajuste de rotação entre as imagens. Este valor de ruído para a escala e rotação possibilita 4 combinações diferentes de resultados.

Para cada configuração do algoritmo de registro automático de imagens, há 4 combinações possíveis considerando o ruído de escala e rotação. Os resultados de estimação de posição, apresentados nesta seção (seção 6.6.3.2), representam a média das 4 combinações possíveis para cada configuração do algoritmo de registro de imagens. As métricas de avaliação utilizam a trajetória simulada em computador, ilustrada na Figura 6.44, como verdade sobre o posicionamento do VANT.

A Tabela 6.17 exibe a média das métricas EGM, SDGM e GM para as 4 combinações possíveis de ruído de escala e rotação. Para o resultado da Tabela 6.17, o algoritmo de registro automático de imagens foi configurado com cada operador de detecção de borda (Canny, Sobel, MLP em LUT) e com o filtro da mediana aplicado na remoção

de ruídos

Tabela 6.17 - Resultado da média das métricas EGM, SDGM e GM considerando as 4 combinações possíveis do ruído de escala e rotação. O algoritmo de registro automático de imagens foi configurado com os três operadores de detecção de borda e com o filtro da mediana.

	EGM	SDGM	GM
Canny	159	81	77
Sobel	165	142	136
MLP em LUT	168	142	138

A porcentagem dos pontos de avaliação classificados como EGM, para a configuração utilizando filtro da Mediana com os algoritmos de detecção de bordas Canny, Sobel e MLP em LUT foi respectivamente de $92,4\%$, $95,9\%$ e $97,7\%$. A porcentagem dos classificados como SDGM, utilizando filtro Gaussiano com o Canny, Sobel e MLP em LUT foi respectivamente de $57,0\%$, 100% e 100%. A porcentagem de GM, utilizando filtro da Mediana com o Canny, Sobel e MLP em LUT foi respectivamente de $54,2\%$, $95,7\%$ e $97,2\%$.

A Figura 6.50 exibe o gráfico do erro, de cada umas das 172 posições estimadas do VANT, obtido por cada configuração do algoritmo de registro automático de imagens. O valor do erro é o valor médio do erro obtido por cada configuração, considerando as 4 combinações possíveis de ruído de escala e rotação. Em azul, o gráfico do erro para o algoritmo de registro de imagens, configurado com a MLP em LUT e o filtro da Mediana. Em vermelho, o gráfico do erro obtido pelo algoritmo de registro de imagens, configurado com o Sobel e o filtro da Mediana. Em verde, o gráfico do erro obtido pelo algoritmo de registro de imagens, configurado com o Canny e o filtro da Mediana. A linha tracejada em vermelho representa o erro igual a 5 *metros*, pontos abaixo desta linha são classificados como EGM.

Figura 6.50 - Gráfico do erro da estimação da posição obtido pelo algoritmo de registro automático de imagens configurado com cada algoritmo para a detecção de bordas (Canny, Sobel e MLP) e com o filtro da Mediana para a remoção de ruídos das imagens.

Fonte: Produção do autor.

A Figura 6.51 exibe o gráfico do desvio padrão do erro da estimativa de posição do VANT. O desvio padrão é calculado considerando o erro das 30 posições estimadas anteriormente. A linha tracejada representa um desvio padrão de 2 *metros*, posições estimadas abaixo dessa linha são classificadas como SDGM. O desvio padrão é calculado sobre o erro médio obtido por cada configuração do algoritmo de registro de imagens, considerando as 4 combinações possíveis de ruído de ajuste de escala e rotação das imagens. Em azul, o gráfico do desvio padrão do erro para o algoritmo de registro de imagens configurado com a MLP em LUT. Em vermelho, o gráfico do desvio padrão do erro para o algoritmo de registro de imagens configurado com o Sobel. Em verde, o gráfico do desvio padrão do erro do algoritmo de registro de imagens configurado com o Canny. O filtro da Mediana foi aplicado para a remoção dos ruídos das imagens em todas configurações.

Figura 6.51 - Gráfico do desvio padrão do erro da estimação da posição obtido pelo algoritmo de registro automático configurado com cada operador de detecção de bordas (Canny, Sobel e MLP) e com o filtro da Mediana.

Fonte: Produção do autor.

A Figura 6.52 apresenta, em azul, a trajetória obtida pelas estimações de posição da aeronave classificadas como EGM. Para o resultado exibido na Figura 6.52, o algoritmo de registro automático de imagens utilizou a MLP em LUT e o filtro da Mediana no processamento das imagens de intensidade LiDAR. A trajetória ilustrada é a trajetória media do algoritmo de registro de imagem para as 4 combinações possíveis, considerando o ruído introduzido no ajuste de escala e rotação. A Figura 6.52 também exibe, em vermelho, a trajetória simulada em computador do VANT e utilizada como verdade.

Figura 6.52 - Em azul, a trajetória estimada pelo algoritmo de registro automático de imagens, configurado com a MLP em LUT e o filtro da Mediana. A trajetória simulada do VANT é exibida em vermelho.

Fonte: Produção do autor.

6.6.3.3 RESULTADOS: REGISTRO AUTOMÁTICO DE IMAGENS DE INTENSIDADE LIDAR COM IMAGEM DE SATÉLITE NA BANDA DO VISÍVEL

Nesta seção, são apresentados os resultados da aplicação do método de registro automático entre as imagens de intensidade LiDAR, que representam as imagens capturadas pelo VANT durante sua trajetória, e uma imagem de satélite georreferenciada na banda do visível. Este teste é importante, pois imagens de sensores LiDAR são uma tecnologia recente e não estão disponíveis sobre todas as regiões. Desta forma, um VANT com um sensor LiDAR (câmera LiDAR) embarcado pode sobrevoar uma região que possui apenas imagens georreferenciadas de satélite na banda do visível.

O presente teste utilizou uma imagem georreferenciada, da banda pancromática do satélite *Quickbird*, e as imagens de intensidade LiDAR, que representam as imagens capturadas pelo VANT durante seu voo. Estas imagens de intensidade LiDAR são as mesmas aplicadas nos testes realizados na seção 6.6.3.1 e na seção 6.6.3.2. Desta forma, este teste possui 172 pontos de avaliação para método de registro automático de imagens. A incerteza aplicada para obtenção da área de correlação manteve-se em 10 metros

Para este teste, o algoritmo de registro automático de imagens foi configurado com a MLP em LUT e o filtro da mediana, pois esta foi a configuração que apresentou os melhores resultados, considerando os resultados apresentados na seções 6.6.3.1 e 6.6.3.2.

O ajuste de escala entre as imagens é feito utilizando o valor de resolução espacial das imagens: $0.6\frac{metros}{pixel}$, que é o valor resolução espacial da imagem da banda pancromática do satélite *Quickbird* e $1\frac{metro}{pixel}$, que é o valor da resolução de cada uma das imagens de intensidade LiDAR, que representam o voo do VANT.

As 172 imagens, que representam a trajetória realizada pela aeronave, são subimagens da imagem de intensidade LiDAR georreferenciada, que está alinhada (ou seja, com a rotação ajustada) em relação a imagem georreferenciada de satélite, portanto, não é necessário o ajuste de rotação entre as imagens.

A Tabela 6.18 exibe as métricas EGM, SDGM e GM do algoritmo de registro automático de imagens, configurado com a MLP em LUT e o filtro da Mediana.

Tabela 6.18 - Resultado das métricas EGM, SDGM e GM para o algoritmo de registro automático de imagens configurado com a MLP em LUT e o filtro da Mediana.

	EGM	SDGM	GM
MLP em LUT	114	11	10

Neste teste, que utilizou a MLP em LUT e o filtro da Mediana na configuração do algoritmo de registro automático de imagens, $66,3\%$ dos pontos de avaliação foram classificados como EGM, $7,7\%$ como SDGM e $7,0\%$ como GM.

A Figura 6.53 apresenta o erro da estimação da posição do VANT dos 172 pontos de avaliação. Este gráfico de erro foi obtido pelo algoritmo de registro automático de imagens, configurado com a MLP em LUT e o filtro da Mediana. A linha tracejada em vermelho representa um erro na estimação de posição de 5 metros, pontos abaixo dessa linha são classificados como EGM.

Figura 6.53 - Gráfico do erro da estimação da posição obtido pelo algoritmo de registro automático de imagens configurado com a MLP em LUT e o filtro da Mediana

Fonte: Produção do autor.

A Figura 6.54 exibe o gráfico do desvio padrão do erro, da estimativa de posição do VANT, obtido pelo algoritmo de registro automático de imagens, configurado com a MLP em LUT e o filtro da Mediana. O desvio padrão do erro é calculado considerando o erro das 30 posições estimadas anteriormente. A linha tracejada representa um desvio padrão de 2 *metros*, posições estimadas abaixo dessa linha são classificadas como SDGM.

Figura 6.54 - Gráfico do desvio padrão do erro da estimação da posição, obtido pelo algoritmo de registro automático de imagens. A configuração do algoritmo de registro automático de imagens utilizou a MLP em LUT e o filtro da Mediana.

Fonte: Produção do autor.

A Figura 6.55 apresenta a imagem de satélite georreferenciada com a trajetória estimada pela algoritmo de registro automático de imagens (em azul), configurado com a MLP em LUT e o filtro da Mediana. Fazem parte da trajetória estimada, exibida na Figura 6.55, apenas pontos classificados como EGM. Na Figura 6.55, também é apresentada a trajetória simulada em computador do VANT, em vermelho. O círculo amarelo marca o ponto de partida da aeronave e o círculo branco o ponto de chegada.

Figura 6.55 - Trajetória estimada pelo algoritmo de registro automático de imagens, configurado com a MLP em LUT e o filtro da mediana, em azul. A trajetória simulada do VANT é exibida em vermelho. Somente os pontos classificados como EGM são exibidos na trajetória estimada.

Fonte: Produção do autor.

6.6.3.4 RESULTADOS: ANÁLISE DOS RESULTADOS DO SEGUNDO EXPERIMENTO

Considerando o conjunto de dados do segundo experimento, o método proposto de registro automático de imagens, aplicado para estimar a posição geográfica do VANT utilizando as imagens de intensidade LiDAR mostrou-se adequado, pois em todos os testes apresentados na seções 6.6.3.1 e 6.6.3.2 (inseridos no segundo experimento), mais de 90% dos pontos de avaliação obtiveram um valor de erro de estimação inferior a 5 metros. Estes resultados permitem concluir que a primeira hipótese levantada para o segundo experimento é verdadeira.

Os resultados das métricas EGM, SDGM e GM, apresentados nas Tabelas 6.16 e 6.17, demonstram que a melhor configuração do método proposto de registro automático de imagens é a que utiliza a MLP em LUT, para a detecção de borda, e o filtro da Mediana, para a remoção dos ruídos indesejáveis. O algoritmo de registro automático de imagens, configurado com a MLP em LUT, obteve melhores resultados para a estimação da posição do VANT que o algoritmo configurado com os operadores Canny e Sobel. Portanto, por meio da análise dos resultados apresentados nas Tabelas 6.16 e 6.17, pode-se concluir que a segunda hipótese levantada para o segundo experimento é verdadeira, isto é, a MLP em LUT produz resultados melhores no processo de registro automático de imagens de intensidade LiDAR que os operadores Canny e Sobel.

Outro teste realizado, no contexto do segundo experimento (seção 6.6.3.3), foi a estimação da posição geográfica do VANT por meio do registro automático de imagens de intensidade LiDAR, que representam a trajetória da aeronave, com uma imagem de satélite georreferenciada e capturada na banda do visível. Os resultados apresentados na Subseção 6.6.3.3, considerando o conjunto de dados utilizados, demonstram que é possível obter o posicionamento do VANT pelo registro de imagens de intensidade LiDAR e imagens de satélite georreferenciadas na banda do visível, pois no teste, a maior parte das posições estimadas obtiveram um erro inferior a 5 metros, o que valida a terceira hipótese, ou seja, que é possível aplicar o método proposto de registro entre imagens de intensidade LiDAR e imagens na banda do visível para se determinar a posição geográfica do VANT.

6.6.4 RESULTADOS: TERCEIRO EXPERIMENTO

O terceiro experimento simula a estimação da posição do VANT, por registro automático de imagens, quando a aeronave sobrevoa uma região coberta por aguá, por exemplo, uma região oceânica.

O conjunto de dados do terceiro experimento é formado por um MDS da cobertura do fundo de uma região oceânica e de subimagens desse MDS, que representam as imagens capturada pela câmera LiDAR do VANT durante sua trajetória. A Figura 6.56 exibe o MDS do fundo da região oceânica, empregada nesta pesquisa, com a trajetória simulada pela aeronave marcada na cor vermelha.

Figura 6.56 - Mapa Digital de Superfície do fundo do oceano com a trajetória do VANT em vermelho.

Fonte: Produção do autor.

O MDS exibido na Figura 6.56 foi obtido junto ao projeto denominado Mapeamento Integrado para o Desenvolvimento Sustentável dos Recursos Marinho da Irlanda (INFOMAR, do inglês, Integrated Mapping For Sustainable Development of Ireland Marine Resources) mantido pelo centro de Pesquisa Geológica da Irlanda (GSI, do inglês, Geological Survey of Ireland).

O projeto INFOMAR está produzindo o mapeamento batimétrico de toda a região costeira da Irlanda para auxiliar a exploração marítima de minerais. A empresa responsável por este mapeamento batimétrico, contrata pelo projeto INFOMAR, é a Tenix LADS Corporation, que utiliza um sensor LiDAR LADS Mk II Ground System, este sensor é capaz de obter dados até uma profundidade de 60 metros, dependendo da turbidez da coluna de água.

A imagem exibida na Figura 6.56, que representa o MDS do fundo do oceano de uma região costeira, foi obtida a partir de um processo de interpolação sobre a nuvem de pontos fornecida pelo sensor LiDAR LADS Mk II Ground System. Informações sobre métodos de interpolação sobre os dados de nuvem de pontos de sensores LiDAR, para gerar o MDS, podem ser encontradas em Chen et al. (2017). A resolução da imagem que representa o MDS gerado é igual a 5 $\frac{metros}{pixel}$. A profundidade máxima imageada pelo sensor LiDAR é igual a 35 metros. As regiões em preto, exibidas na Figura 6.56, são regiões onde, devido a profundidade, não foi possível a captura de dados pelo sensor LiDAR.

A trajetória do VANT, exibida na Figura 6.56, é composta por 92 pontos para a avaliação da posição estimada pelo algoritmo de registro automático de imagens proposto. Em cada ponto de avaliação, foi feita uma subimagem do MDS do fundo do oceano, o conjunto destas 92 subimagens representam as imagens capturadas pela aeronave durante seu voo. Cada subimagem possui 32×32 pixels, o que corresponde a uma área imageada de 26000 m^2, esta dimensão para cada subimagem foi baseada no conjunto de dados do primeiro experimento, que representam dados reais de um voo de um VANT.

Para este terceiro experimento, o valor da incerteza, associada ao erro do INS, aplicado para determinar o tamanho da área de correlação é igual a 80 metros. Este valor foi assim determinado, pois devido a imagem georreferenciada do MDS ter uma resolução espacial menor, quando comparada com as imagens georreferenciadas do primeiro e segundo experimento, a área de correlação seria pequena, o que poderia prejudicar a análise dos resultados obtidos. Desta forma, foi utilizado o maior valor possível para a incerteza. Para o conjunto de dados deste experimento, aplicando

uma incerteza de 80 metros a área de correlação possui 48 $\times$ 48 pixels, o que representa uma área de 57600 m^2.

A Figura 6.57 apresenta uma das subimagens do MDS georreferenciado do fundo do oceano, que representam as imagens capturadas pela câmera LiDAR da aeronave durante sua trajetória.

Figura 6.57 - Subimagem do MDS georreferenciado do fundo do oceano, que representa uma imagem capturada pelo VANT.

Fonte: Produção do autor.

As combinações empregadas para configurar o algoritmo de registro automático de imagens, para a produção dos resultados do terceiro experimento, são apresentadas na Figura 6.58.

Figura 6.58 - Combinação empregadas para configurar o algoritmo de registro automático de imagens no terceiro experimento.

Fonte: Produção do autor.

Nos testes do terceiro experimento, devido ao fato das imagens que representam a trajetória do VANT serem subimagens do MDS do fundo do oceano, foi acrescentado um ruído no ajuste de escala de $+ \; ou \; - \; 6\%$ e um ruído de $+ \; ou \; - \; 6°$ no ajuste de rotação. Diferentemente do conjunto de dados utilizados no primeiro experimento, o conjunto de dados do terceiro experimento não são provenientes do voo de um VANT. Portanto, o ruído é acrescentado ao processo de ajuste e escala e rotação para oferecer maior realidade ao experimento. O valor de $\pm$ 6% de ruído no ajuste de escala e $\pm$ 6° no ajuste de rotação foram assim definidos, pois valores menores que estes não apresentavam mudanças nos valores das métricas de avaliação que possibilitassem diferenciar as configurações do algoritmo de registro automático de imagens. Estes valores de ruídos permitem 4 combinações diferentes para cada configuração apresentada na Figura 6.58. Os resultados apresentados nas Subseções 6.6.4.1 e 6.6.4.2 são os valores médios obtidos por cada configuração do algoritmo de registro automático de imagens para as 4 combinações possíveis de ruído.

6.6.4.1 RESULTADOS: FILTRO GAUSSIANO COM ALGORITMOS DE DETECÇÃO DE BORDAS

Nesta seção, são apresentados os resultados da estimação da posição do VANT, por meio do registro automático de imagens do MDS do fundo do oceano. O método proposto de registro automático de imagens foi configurado com o filtro Gaussiano,

para a remoção de ruídos indesejáveis nas imagens, e com os algoritmos de detecção de borda Canny, Sobel e MLP em LUT. Nos teste de todas as configurações, também são acrescentados os valores de ruído de escala e rotação.

A Tabela 6.19 exibe o valor das métricas EGM, SDGM e GM obtidas pelo algoritmo de registro automático de imagens, configurado com o filtro Gaussiano e com cada operador de detecção de borda. Os valores exibidos na Tabela 6.19 são as médias de cada configuração para as 4 combinações possíveis de ruído de escala e rotação.

Tabela 6.19 - Resultado da média das métricas EGM, SDGM e GM. O algoritmo de registro automático de imagens foi configurado com cada operador de detecção de borda e com o filtro Gaussiano.

	EGM	SDGM	GM
Canny	88	53	53
Sobel	90	62	62
MLP em LUT	91	62	62

A porcentagem dos pontos de avaliação classificados como EGM, para a configuração utilizando filtro Gaussiano com os algoritmos de detecção de bordas Canny, Sobel e MLP em LUT foi respectivamente de $95,6\%$, $97,8\%$ e $98,9\%$. A porcentagem dos classificados como SDGM, utilizando filtro Gaussiano com o Canny, Sobel e MLP em LUT foi respectivamente de $85,5\%$, 100% e 100%. A porcentagem de GM, utilizando filtro Gaussiano com o Canny, Sobel e MLP em LUT foi respectivamente de $85,5\%$, 100% e 100%.

A Figura 6.59 exibe o gráfico do erro, de cada umas das 92 posições estimadas, obtido pelas configurações do algoritmo de registro automático deste experimento. Neste gráfico são apresentados os valores médios obtidos com as 4 possibilidades possíveis, considerando o ruído de escala e rotação. O gráfico do erro do algoritmo de registro de imagens, configurado com a MLP em LUT, é apresentado na cor azul. O gráfico do erro do algoritmo de registro de imagens, configurado com o Sobel, é apresentado na cor vermelha. O gráfico do erro do algoritmo de registro de imagens, configurado com o Canny, é apresentado na cor verde. A linha tracejada em vermelho representa o erro igual a 5 metros, pontos abaixo desta linha são classificados como EGM.

Figura 6.59 - Gráfico do erro da estimação da posição, obtido pelo algoritmo de registro automático de imagem configurado com cada detector de borda e o filtro Gaussiano

Fonte: Produção do autor.

A Figura 6.60 exibe o gráfico do desvio padrão do erro da estimativa de posição geográfica do VANT, obtido pelo algoritmo de registro automático de imagens. O desvio padrão é calculado considerando o erro das 30 posições estimadas anteriormente. A linha tracejada representa um desvio padrão de 2 *metros*, posições estimadas abaixo dessa linha são classificadas como SDGM. O gráfico da Figura 6.60 foi obtido para a média do erro do algoritmo de registro automático de imagens considerando as 4 combinações possíveis, considerando o ruído de escala e rotação. O gráfico do desvio padrão do erro do algoritmo de registro de imagens, configurado com a MLP em LUT, é apresentado na cor azul. O gráfico do desvio padrão do erro do algoritmo de registro de imagens, configurado com o Sobel, é apresentado na cor vermelha. O gráfico do desvio padrão do erro do algoritmo de registro de imagens, configurado com o Canny, é apresentado na cor verde.

Figura 6.60 - Gráfico do desvio padrão do erro da estimação da posição, obtido pelo algoritmo de registro automático utilizando imagens de um MDS da superfície do fundo do oceano.

Fonte: Produção do autor.

A Figura 6.61 apresenta a trajetória obtida pelo algoritmo de registro automático de imagens das estimações de posição classificadas como EGM, esta trajetória está na cor azul. Para o resultado exibido na Figura 6.61, o algoritmo de registro automático de imagens foi configurado com a MLP em LUT e o filtro Gaussiano. A Figura 6.52 também exibe a trajetória do VANT na cor vermelha. O ponto de partida do VANT é marcado com o círculo amarelo e o ponto de chegada é marcado com o círculo branco.

Figura 6.61 - Trajetória estimada pelo algoritmo de registro automático de imagens configurado com a MLP em LUT e o filtro Gaussiano.

Fonte: Produção do autor.

6.6.4.2 RESULTADOS: FILTRO DA MEDIANA COM ALGORITMOS DE DETECÇÃO DE BORDAS

A presente seção exibe os resultados da estimação da posição do VANT por meio do registro automático de imagens do MDS do fundo do oceano. O algoritmo proposto de registro automático de imagens foi configurado com o filtro da Mediana, para a remoção de ruídos indesejáveis, e com os operadores de detecção de borda Canny, Sobel e MLP em LUT. Nos testes, para todas as configurações do algoritmo de registro automático de imagens, também são acrescentados os valores de ruído de escala e rotação.

A Tabela 6.20 exibe o valor das métricas EGM, SDGM e GM obtidas pelo algoritmo de registro automático de imagens, configurado com o filtro da Mediana e com cada operador de detecção de borda. Os valores das métricas exibidos na Tabela 6.20 são as médias de cada configuração para as 4 combinações possíveis de ruído de escala e rotação.

Tabela 6.20 - Resultado da média das métricas EGM, SDGM e GM para o algoritmo de registro automático de imagem configurado com cada detector de borda.

	EGM	SDGM	GM
Canny	89	62	62
Sobel	91	62	61
MLP em LUT	92	62	62

A porcentagem dos pontos de avaliação classificados como EGM, para a configuração utilizando filtro da Mediana com os algoritmos de detecção de bordas Canny, Sobel e MLP em LUT foi respectivamente de $96,7\%$, $98,9\%$ e 100%. A porcentagem dos classificados como SDGM, utilizando filtro Gaussiano com o Canny, Sobel e MLP em LUT foi respectivamente de 100%, 100% e 100%. A porcentagem de GM, utilizando filtro Gaussiano com o Canny, Sobel e MLP em LUT foi respectivamente de 100%, $98,4\%$ e 100%.

A Figura 6.62 exibe o gráfico do erro da estimação de de posição do VANT para cada configuração do algoritmo de registro automático de imagens. O gráfico do erro do algoritmo de registro, configurado com a MLP em LUT, é exibido na cor azul. O gráfico do erro do algoritmo de registro automático de imagens, configurado com o Sobel, é exibido na cor vermelha. O algoritmo de registro automático de imagens, configurado com o Canny, é exibido na cor verde. A linha tracejada em vermelho representa um erro igual a 5 metros, pontos abaixo desta linha são classificados como EGM. Os gráficos do erro exibidos na Figura 6.62 são as médias considerando as 4 combinações para o ruído de escala e rotação.

Figura 6.62 - Gráfico do erro da estimação da posição, obtido pelo algoritmo de registro automático de imagens, configurado com os três operadores para a detecção de borda e com o filtro da Mediana.

Fonte: Produção do autor.

A Figura 6.63 exibe o gráfico do desvio padrão do erro, da estimativa de posição geográfica do VANT, obtido por cada configuração do algoritmo de registro automático de imagens. O desvio padrão é calculado considerando o erro das 30 posições estimadas anteriormente. A linha tracejada representa um desvio padrão de 2 *metros*, posições estimadas abaixo dessa linha são classificadas como SDGM. O gráfico da Figura 6.63 foi obtido para a média do erro do algoritmo de registro automático de imagens, configurado com a MLP em LUT (azul), Canny (verde), Sobel (vermelho), e considerando as 4 combinações possíveis para o ruído de escala e rotação.

Figura 6.63 - Gráfico do desvio padrão do erro da estimação da posição obtido pelo algoritmo de registro automático configurado com cada um dos detectores de borda.

Fonte: Produção do autor.

A Figura 6.64 apresenta a trajetória obtida pelas estimações de posição classificadas como EGM, em azul. Para o resultado exibido na Figura 6.64, o algoritmo de registro automático de imagens utilizou a MLP em LUT e o filtro da Mediana no processamento das imagens do MDS do fundo do oceano. A Figura 6.64 também exibe a trajetória do VANT, em vermelho. O ponto de partida do VANT é marcado com o círculo amarelo e o ponto de chegada é marcado com o círculo branco.

Figura 6.64 - Trajetória estimada pelo algoritmo de registro automático de imagens configurado com a MLP em LUT e o filtro da Mediana.

Fonte: Produção do autor.

6.6.4.3 RESULTADOS: ANÁLISE DOS RESULTADOS DO TERCEIRO EXPERIMENTO

Para o conjunto de dados do terceiro experimento, o método proposto de registro automático de imagens, aplicado para a estimar a posição geográfica do VANT utilizando imagens do MDS da cobertura do fundo do oceano, mostrou-se adequado, pois nos testes realizados (seção 6.6.4.1 e seção 6.6.4.2) mais de 95% dos pontos de avaliação foram classificados como EGM, ou seja, obtiveram um erro inferior a 5 metros na estimação da posição do VANT. O que demonstra que a primeira hipótese para o terceiro experimento é verdadeira.

Os resultados das métricas EGM, SDGM, GM, apresentados nas Tabelas 6.19 e 6.20, demonstraram que, para o conjunto de dados do terceiro experimento, o algoritmo de registro automático de imagens, configurado com a MLP em LUT, para a detecção de bordas, e o filtro da Mediana, para a remoção dos ruídos nas imagens, obteve os melhores resultados. Isto atesta que esta configuração estima com mais precisão a posição do VANT (métricas EGM e GM). Assim, pode-se concluir que a segunda hipótese do terceiro experimento é verdadeira.

7 ESTIMAÇÃO DA POSIÇÃO DE VANT POR ODOMETRIA VISUAL MONOCULAR

Odometria visual é o processo de estimar o movimento de um agente, seja este, um veículo, robô ou aeronave, utilizando as imagens provenientes de um ou múltiplos sensores imageadores fixados a ele. O termo foi escolhido por sua similaridade com a odometria veicular, que estima o movimento do veículo baseado no número de voltas de suas rodas em um intervalo de tempo (NISTER et al., 2004; SCARAMUZZA; FRIEDRICH, 2011; SCARAMUZZA; FRIEDRICH, 2012).

A odometria visual estima a posição e orientação do sensor imageador por meio da análise da mudança, que seu movimento induz, nas imagens capturadas por ele. Como este sensor está fixado em um agente (robô ou aeronave), encontrar o movimento do agente pode ser tratado como encontrar o movimento do sensor (NISTER et al., 2004; SCARAMUZZA; FRIEDRICH, 2011).

A odometria visual pode ser realizada por meio da análise das imagens de um único sensor imageador, denominada odometria visual monocular, ou por meio da análise de imagens de dois ou mais sensores, denominada odometria visual estéreo (SCARAMUZZA; FRIEDRICH, 2011).

A odometria visual estéreo se degenera para a odometria visual monocular quando a distância para a cena é muito maior que a distância entre os sensores fixados no agente. Por este motivo, a odometria visual monocular é empregada para estimar o movimento de VANT, pois a distância para a cena sobrevoada é muito maior que as distância entre os sensores imageadores fixados no agente (SCARAMUZZA; FRIEDRICH, 2011; SCARAMUZZA; FRIEDRICH, 2012).

Segundo Nister et al. (2004) e Scaramuzza e Friedrich (2011), um problema apresentado pela odometria visual monocular é que o movimento de translação realizado pelo agente, no qual o sensor imageador está fixado, pode ser recuperado somente pela aplicação de um fator de escala. Mas os mesmos autores destacam que este problema pode ser facilmente solucionado, pois este fator de escala pode vir de outro sensor do agente, por exemplo, o velocímetro. Outro problema da odometria visual monocular, identificado por Nister et al. (2004), Scaramuzza e Friedrich (2011) e Conte e Doherty (2008), é o erro de deriva do método, este é um erro na estimação da posição que aumenta com o distanciamento do agente ao ponto de partida da trajetória. Este erro de deriva pode ser proveniente de uma quantidade insuficiente de imagens obtidas durante o voo, ou devido as atitudes da aeronave, ou mesmo

devido as características da cobertura do solo (NISTER et al., 2004).

Recentemente, a odometria visual também tem sido aplicada para estimar o movimento de agentes utilizando imagens de intensidade provenientes de câmeras LiDAR (MCMANUS et al., 2013; DONG; BARFOOT, 2014). Segundo Dong e Barfoot (2014), os resultados da odometria visual utilizando imagens de intensidade, obtidas por câmeras LiDAR, possuem uma robustez superior aos métodos que utilizam imagens obtidas por câmeras de vídeo, em situação em que o ambiente sofre mudanças na iluminação. Além disso, segundo McManus et al. (2013), a câmera LiDAR pode substituir a câmera de vídeo comum como sensor imageador, em métodos de navegação por visão computacional, como a odometria visual, em ambientes sem ou com baixa iluminação.

O princípio básico da odometria visual está em detectar pontos de interesse e extrair desses uma estrutura de dados que os descrevam, permitindo a sua identificação em imagens diferentes. A partir da correspondência dos pontos de interesse que é possível estimar o movimento do agente ao qual o sensor imageador está fixado (HUANG et al., 2016).

O algoritmo de odometria visual monocular pode ser formulado segundo a sequência de operações exibidas na Figura 7.1.

Figura 7.1 - Sequência de operações do algoritmo de odometria visual monocular.

Fonte: Adaptado de Nister et al. (2004).

Na Figura 7.1, a operação **Sequência de imagens** obtém do sensor imageador fixado ao agente duas imagens da região onde ele se localiza. A primeira imagem capturada no instante $t - \Delta_t$ e a segunda no instante t, sendo que, o valor de Δ_t deve ser suficiente para que grande parte da cena presente na imagem captura em $t - \Delta t$ também esteja presente na imagem capturada em t. A operação **Detecção de características** encontra os pontos de interesse na imagem capturada em $t - \Delta t$ e na imagem capturada em t. Para cada um desses pontos determina-se um vetor de atributos para identificá-los, este processo é realizado por meio de um algoritmo de detecção e descrição de pontos de interesse, também conhecido por descritor.

Na Figura 7.1, a operação **Casamento entre pontos de interesse** realiza a correspondência (casamento) entre os pontos de interesse utilizando o vetor de atributos. A correspondência é feita por meio de uma métrica de similaridade, por exemplo, a distância Euclideana ou a distância de Hamming (REBOUÇAS et al., 2013). Estabelecer a correspondência é determinar na imagem capturada em t o ponto de interesse correspondente na imagem $t - \Delta t$. A última etapa do algoritmo de odometria visual monocular, exibida na Figura 7.1, é a **Estimação do movimento**, que determina o movimento do sensor imageador fixado ao agente, a partir dos pares de pontos de interesse correspondentes presentes nas duas imagens.

7.1 ODOMETRIA VISUAL MONOCULAR

A odometria visual monocular tem sido aplicada em diversas pesquisas para a estimação do movimento (ou trajetória) de VANTs (CONTE; DOHERTY, 2008). Na pesquisa desenvolvida por Rebouças et al. (2013), o descritor *Oriented Fast and Rotated Brief* (ORB) é aplicado para detectar os pontos de interesse e extrair os atributos de um par de imagens capturadas por VANT em tempo de voo. A partir da correspondência entre esses pontos de interesse, obtida pela distância euclideana dos vetores de atributos, que se determina o movimento da câmera fixada à aeronave.

A pesquisa de Chunhui et al. (2014) estima o movimento de um VANT cumprindo as mesmas etapas da pesquisa desenvolvida por Rebouças et al. (2013), a principal diferença entre elas é que a primeira aplica o o descritor *Fast Retina Keypoint* (FREAK) para detectar e extrair os atributos dos pontos de interesse.

Em Huang et al. (2016), é descrito um sistema para odometria visual monocular utilizando uma câmera RGB-D (câmera que captura radiação do espectro eletromagnético no comprimento do visível e também determina a distância para o alvo). O descritor aplicado é o *Features from Accelerated Segment Test* (FAST).

Segundo Nister et al. (2004), as etapas da odometria visual monocular, e que também foram identificadas nas pesquisas anteriormente citadas, são: 1 - detecção, extração de atributos; 2 - correspondência dos pontos de interesse; 3 - obtenção do movimento do sensor imageador a partir dos pares de pontos correspondentes.

Nesta tese de doutorado, a odometria visual monocular é aplicada para estimar a trajetória realizada pelo VANT em três situações distintas: 1 - quando o VANT navega em ambientes urbanos, com condições de iluminação suficiente para que a câmera de vídeo seja aplicada como sensor imageador; 2 - em situação em que o VANT navega sobre ambientes urbanos, em condição de pouca ou sem luminosidade. Nesta situação são aplicadas imagens de intensidade provenientes de um sensor LiDAR; 3 - em situação em que o VANT navega sobre regiões cobertas por água, por exemplo, regiões oceânicas. Nesta situação, o método utiliza imagens do MDS do fundo do oceano obtidas por um sensor LiDAR.

O algoritmo de odometria visual monocular aplicado nesta tese de doutorado, para estimar a trajetória de um VANT, por meio da análise de imagens provenientes do sensor imageador (câmera de vídeo ou sensor LiDAR) da aeronave, é exibido na Figura 7.2.

Figura 7.2 - Algoritmo de odometria visual monocular aplicado para estimar a trajetória do VANT.

Fonte: Produção do autor.

Na Figura 7.2, a primeira etapa do algoritmo é a obtenção de uma imagem, capturada pelo sensor imageador da aeronave (LiDAR ou câmera de vídeo), no instante de tempo $T - d_T$, e a obtenção de outra imagem subsequente, também capturada pelo sensor da aeronave, no instante de tempo T. Segundo Scaramuzza e Friedrich (2011). O intervalo d_T deve ser suficiente para que a maior parte da cena presente na imagem capturada no instante $T - d_t$ esteja presente na imagem capturada no instante T. Posteriormente, ambas as imagens são convertidas para tons de cinza (vide 6.1), pois os algoritmos de detecção e descrição de pontos de interesse operam sobre imagens monocromáticas (NISTER et al., 2004; SCARAMUZZA; FRIEDRICH, 2011; SCARAMUZZA; FRIEDRICH, 2012). As imagens passam por um processo de eli-

minação do ruídos indesejáveis, por meio da aplicação do filtro da mediana, segundo (CONTE; DOHERTY, 2008) e (DONG; BARFOOT, 2014), ruídos indesejáveis podem atrapalhar o processo de detecção e descrição dos pontos de interesse e o filtro da mediana é o mais indicado para este objetivo (para informações sobre a filtragem de imagens, vide 6.2). A próxima etapa do algoritmo é aplicar o descritor *Speeded Up Robust Features* (SURF) para a detecção e descrição dos pontos de interesse, este algoritmo foi selecionado, pois segundo as pesquisas de Bauer et al. (2007), Benseddik et al. (2014) e Scaramuzza e Friedrich (2012) é o que obteve o melhor desempenho e eficiência quando comparado a outros algoritmos que realizam a mesma função, sendo o mais indicado para a navegação de veículos autônomos. Em seguida, realiza-se a correspondência entre os pontos de interesse, por meio de alguma métrica de similaridade, como a distância euclidiana (SCARAMUZZA; FRIEDRICH, 2011; SCARAMUZZA; FRIEDRICH, 2012). Com o objetivo de eliminar as correspondências inválidas, utiliza-se o algoritmo *Random Sample Consensus* (RANSAC), pois segundo Conte e Doherty (2008) e Rebouças et al. (2013), sua aplicação melhora o resultado final da estimação da posição do VANT por odometria visual monocular.

Na Figura 7.2, a última etapa é a aplicação do algoritmo dos oito pontos correspondentes, para estimar a matriz Fundamental entre duas imagens subsequentes. Com a obtenção da matriz Fundamental, realiza-se sua decomposição em valores singulares. O resultado desta operação é a matriz Essencial, que passa pelo mesmo processo de fatoração em seus valores singulares, esta operação gera como resposta o vetor de translação e a matriz de rotação do movimento realizado pelo sensor (HARTLEY; ZISSERMAN, 2004).

Para a obtenção da matriz de rotação e do vetor de translação do movimento realizado entre as duas imagens subsequentes, deve-se conhecer a matriz de parâmetros intrínseco do sensor, esta matriz é obtida por um processo de calibração do sensor ou é fornecida pelo fabricante do sensor (NISTER et al., 2004).

O algoritmo de estimação da trajetória por odometria visual monocular proposto foi implementado em linguagem C com auxílio da biblioteca Opencv, uma biblioteca livre, de uso acadêmico e comercial, utilizada para o desenvolvimento de aplicações de visão computacional (ITSEEZ, 2015). O algoritmo implementado foi testado em duas plataformas de hardware diferentes e que podem ser embarcadas em um VANT: o computador Raspberry Pi Model B-1 e o computador Zybo ZINQ 7000, para mais informações sobre estes computadores, vide capítulo 3

7.1.1 ALGORITMO DE DETECÇÃO E DESCRIÇÃO DOS PONTOS DE INTERESSE - DESCRITORES

A correspondência entre pontos de interesse de duas imagens da mesma cena é parte fundamental da solução do problema de odometria visual. Para realizar esta correspondência é necessário localizar os pontos de interesse em duas imagens da mesma cena, caracterizá-los, por meio de um vetor de atributos, e efetuar o casamento (correspondência) destes pontos, por meio de uma métrica de similaridade entre os vetores de atributos (BAUER et al., 2007).

Para localizar e caracterizar os pontos de interesse de uma imagem são empregados os descritores, que são algoritmos que utilizam uma função métrica para localização e caracterização dos pontos de interesse (BENSEDDIK et al., 2014).

Segundo Scaramuzza e Friedrich (2011), o algoritmo dos descritores pode ser dividido em três etapas principais:

- Detecção: encontrar cada ponto de interesse presente na imagem;
- Descrição: gerar para cada ponto de interesse um vetor de atributos que o represente de maneira única;
- Correspondência (Casamento): efetuar a correspondência entre os pontos de interesse com base em uma métrica de similaridade entre os vetores de atributos. Uma métrica de similaridade que pode ser aplicada é a distância Euclidiana.

Segundo Nishitani (2015), os descritores possuem propriedades que afetam diretamente o resultado da odometria visual, entre elas: 1 - diferenciabilidade: o descritor deve gerar um vetor de atributos para o ponto de interesse que o identifique de forma única na imagem; 2 - repetibilidade: o descritor deve encontrar a grande parte pontos de interesse que pertençam a mais de uma imagem da mesma cena, ou seja, encontrar os pontos de interesse que são iguais e se repetem em outras imagens; 3 - quantidade: a quantidade de pontos de interesse encontrados pelo descritor pode influenciar o resultado da aplicação. Em geral, o aumento do número de pontos de interesse detectados melhora o resultado final da aplicação na qual o descritor foi empregado. Além disso, há aplicações, como a odometria visual, que exigem que um número mínimo de pontos de interesse sejam detectados. A odometria visual exige no mínimo oito pares de pontos correspondentes entre duas imagens consecutivas para

estimar o movimento da câmera; 4 - eficiência: esta propriedade está relacionada com o tempo que o descritor gasta para realizar a detecção, descrição e casamento dos pontos de interesse. Em geral, a eficiência é influenciada pela diferenciabilidade, pois quanto maior for um vetor de atributos de um ponto de interesse maior será o tempo gasto pelo descritor para gerá-lo, e pela quantidade de pontos de interesse detectados. A eficiência do descritor é determinante para sua aplicação, a odometria visual aplicada à estimação da posição de VANTs necessita de descritores eficientes; 5 - invariância: o descritor deve garantir que um ponto de interesse pertencente a duas imagens diferentes seja identificado mesmo com a presença de transformações de escala, rotação ou afim.

Existem na literatura diversos descritores, que são divididos em duas classes de algoritmos: descritores de canto e descritores de região. Na teoria de visão computacional, canto é a intersecção de duas ou mais bordas (SCARAMUZZA; FRIEDRICH, 2011). Os descritores de canto mais aplicados em odometria visual são: o Harris (HARRIS; STEPHENS, 1988), Shi-Tomasi (SHI; TOMASI, 1994) e o FAST (TRAJKOVIC; HEDLEY, 1998).

Em odometria visual, região (ou *blob*) é um padrão na imagem que se difere da sua vizinhança imediata em termos da intensidade, cor e textura. Os descritores de região mais aplicados em odometria visual são o *Scale Invariant Feature Transform* (SIFT) (LOWE, 2004) e o SURF (BAY et al., 2008).

As pesquisas desenvolvidas por Scaramuzza e Friedrich (2011), Scaramuzza e Friedrich (2012) descrevem o processo para estimação da posição de um agente, por meio da odometria visual, e realizam uma análise dos descritores mais aplicados com base nas propriedades de diferenciabilidade, repetibilidade, quantidade, eficiência e invariância a escala, rotação e transformação afim. A Tabela 7.1 apresenta a avaliação dos descritores realizada por Scaramuzza e Friedrich (2011), Scaramuzza e Friedrich (2012), que demonstra que o SIFT e o SURF obtêm os melhores resultados.

Tabela 7.1 - Avaliação dos descritores segundo as seguintes propriedades: invariante a rotação (Rotação), invariante a mudança de escala (Escala), invariante a transformação afim (Afim), repetibilidade, diferenciabilidade, quantidade, eficiência

	Rotação	Escala	Afim	Repetibilidade	Diferenciabilidade	Quantidade	Eficiência
Harris (Canto)	×			+ + +	+ + +	++	++
Shi-Tomasi (Canto)	×			+ + +	+ + +	++	++
FAST (Canto)	×	×		++	++	++	+ + +
SURF (Região)	×	×	×	+ + +	++	+ + +	++
SIFT (Região)	×	×	×	+ + +	++	+ + +	+

Fonte: Adaptado de Scaramuzza e Friedrich (2011).

A pesquisa desenvolvida por Benseddik et al. (2014) realiza um comparativo entre os algoritmos SIFT e SURF, para estabelecer qual o melhor descritor a ser aplicado para a navegação de robôs móveis por meio da odometria visual. Os resultados da avaliação demonstraram que o descritor SURF estimou de forma mais precisa a trajetória do agente e foi mais eficiente durante o processo.

O estudo de Bauer et al. (2007) apresenta um comparativo entre os descritores SIFT e SURF, e conclui que o segundo tem uma performance melhor, pois apresenta mais correspondências corretas e um intervalo específico, sendo este mais indicado para aplicações de odometria visual de agentes móveis.

7.1.2 SURF: SPEEDED UP ROBUST FEATURES

O SIFT é uma abordagem de sucesso aplicada para a detecção e descrição de pontos de interesse, introduzida por Lowe (2004). O SURF é baseado nos mesmos princípios, mas utiliza um esquema diferente para fornecer resultados melhores e com melhor desempenho (BAY et al., 2008).

O SURF pode ser dividido em duas etapas principais: a detecção dos pontos de interesse e a descrição dos mesmos. Para a detecção dos pontos de interesse, o SURF utiliza o determinante da matriz Hessiana, cujo valor é a variação dos níveis de cinza no local em que está sendo calculado. Um ponto de interesse é uma posição na imagem cujo o determinante da matriz Hessiana é máximo, ou seja, tem a maior

variação dos níveis de cinza. A Equação 7.1 demonstra o cálculo da matriz Hessiana em um ponto x.

$$H(x,\sigma) = \begin{bmatrix} L_{xx}(x,\sigma) & L_{xy}(x,\sigma) \\ L_{yx}(x,\sigma) & L_{yy}(x,\sigma) \end{bmatrix} \tag{7.1}$$

onde $L_{xx}(x,\sigma) = I(x).\frac{\partial^2}{\partial x^2}g(\sigma)$, $L_{yy}(x,\sigma) = I(x).\frac{\partial^2}{\partial y^2}g(\sigma)$ e $L_{yx}(x,\sigma) = L_{xy}(x,\sigma) = I(x).\frac{\partial^2}{\partial xy}g(\sigma)$ são a convolução da imagem $I(x)$ com a segunda derivada da função gaussiana.

A convolução com a matriz Hessiana em cada ponto da imagem é um processo muito custoso, para melhorar o desempenho do processo é feita uma aproximação aplicando o conceito de imagens integrais. Uma imagem integral $I_{int}(x)$ armazena em cada posição (x,y) a soma dos valores de todos os pixel da imagem $I(x)$, da origem até a posição. A Equação 7.2 exibe o cálculo de formação de uma imagem integral.

$$I_{int}(x) = \sum_{i=0}^{i \leq x} \sum_{j=0}^{j \leq y} I(i,j) \tag{7.2}$$

Sobre a imagem integral $I_{int}(x)$ são aplicadas as máscaras de convolução exibidas nas Equações 7.3 e 7.4

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & -2 & -2 & -2 & 1 & 1 & 1 \\ 1 & 1 & 1 & -2 & -2 & -2 & 1 & 1 & 1 \\ 1 & 1 & 1 & -2 & -2 & -2 & 1 & 1 & 1 \\ 1 & 1 & 1 & -2 & -2 & -2 & 1 & 1 & 1 \\ 1 & 1 & 1 & -2 & -2 & -2 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{7.3}$$

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & -1 & -1 & -1 & 0 \\ 0 & 1 & 1 & 1 & 0 & -1 & -1 & -1 & 0 \\ 0 & 1 & 1 & 1 & 0 & -1 & -1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & -1 & -1 & 0 & 1 & 1 & 1 & 0 \\ 0 & -1 & -1 & -1 & 0 & 1 & 1 & 1 & 0 \\ 0 & -1 & -1 & -1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{7.4}$$

O resultado da aplicação da máscara de convolução, exibida na Equação 7.3, sobre a imagem integral ($I_{int}(x)$) é D_{xx}. A aplicação da máscara de convolução, exibida na Equação 7.4, sobre a imagem integral ($I_{int}(x)$) é D_{xy}. Assim, a aproximação para obter o determinante da matriz Hessiana é exibida na Equação 7.5.

$$Det(H(x,\sigma)_{aprox}) = D_{xx}D_{xy} - (wD_{xy})^2 \tag{7.5}$$

onde w é aplicado para assegurar a conservação de energia da função Gaussiana. O valor de w mais aplicado é igual a 0.9 (BAY et al., 2008). O SIFT aplica o laplaciano da função Gaussiana para determinar a variação dos níveis de cinza em cada posição da imagem, o que é uma operação mais custosa que a aproximação do determinante da matriz Hessiana proposta por Bay et al. (2008) (BAUER et al., 2007; BENSEDDIK et al., 2014).

Para conseguir a invariância à escala, durante o cálculo do determinante da matriz Hessiana, em cada posição da imagem, cria-se um espaço de escala definidos por oitavas. As oitavas são a aplicação de diferentes tamanhos das máscaras de convolução (Equações 7.3 e 7.4) sobre a imagem integral. A primeira oitava é composta pelas máscaras de dimensão 9×9, 15×15 e 27×27 e representa a variação do fator de escala de 1 a 2.5. A segunda oitava é composta pelas máscaras de dimensão 27×27, 39×39 e 51×51 e representa a variação do fator de escala de 1.5 a 4.5. A terceira oitava é composta pelas máscaras de dimensão 51×51, 75×75 e 89×89 e representa a variação do fator de escala de 4.1 a 8.5. Segundo Bauer et al. (2007) e Benseddik et al. (2014), o espaço de escala no algoritmo SIFT é obtido pela mudança de escala decrescente da imagem, o que torna a construção do espaço de escala mais custosa quando comparada ao processo realizado no algoritmo SURF.

O ponto de interesse é o valor máximo do determinante da matriz Hessiana, considerando uma vizinhança de 8 e para cada oitava. Esta operação é denominada de supressão de não máximos em $3 \times 3 \times 3$.

A descrição dos pontos de interesse, aplicada no algoritmo SURF, utiliza a transformada discreta de Harr (BAY et al., 2008). O primeiro passo no processo de descrição dos pontos de interesse consiste na aplicação por convolução das máscaras, exibidas nas Equações 7.6 e 7.7, sobre a região onde se localiza o ponto de interesse. As máscaras exibidas nas Equações 7.6 e 7.7 representam a transformada discreta de Haar na direção x e y, respectivamente, e determinam o valor da variação da intensidade dos pixels nas direções x e y (BAY et al., 2008).

$$\begin{bmatrix} -1 & -1 & 1 & 1 \\ -1 & -1 & 1 & 1 \\ -1 & -1 & 1 & 1 \\ -1 & -1 & 1 & 1 \end{bmatrix} \tag{7.6}$$

$$\begin{bmatrix} -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} \tag{7.7}$$

As máscaras exibidas nas Equações 7.6 e 7.7 são exemplos de máscaras aplicadas em uma região, para o ponto de interesse localizado na oitava de fator de escala igual a 2.

O tamanho da região do ponto de interesse e o tamanho da máscara da transformada discreta de Haar são definidos considerando a escala na qual foi encontrado o ponto de interesse. O tamanho da região é igual a $20.s$ e o tamanho da máscara é $2.s$, onde s é o valor de escala na qual foi localizado o ponto de interesse.

Após a convolução das máscaras ,exibidas nas Equações 7.6 e 7.7, verifica-se qual é a direção predominante da variação dos níveis de cinza dos pixels na região. Esta direção predominante é utilizada para a construção do vetor de atributos no algoritmo SURF, o que garante sua invariância à rotação (BAY et al., 2008).

Para a construção do vetor de atributos, divide-se a região do ponto de interesse em 16 subrregiões de mesmo tamanho, e para cada uma delas recupera-se o valor

de d_x, d_y, $|d_x|$ e $|d_y|$. Os valores de d_x e d_y são a soma dos valores obtidos após a aplicação das máscaras para a transformada de Haar nas direções x (Equação 7.6) e y (Equação 7.7), respectivamente. Os valores de $|d_x|$ e $|d_y|$ são obtidos da mesma forma, mas considerando os valores absolutos. Desta forma, cada sub-região será formada por um vetor de 4 atributos, e a região do ponto de interesse possui um vetor de atributos com 64 valores.

7.1.3 MOVIMENTO DA CÂMERA A PATIR DE PONTOS CORRESPONDENTES

Uma das principais etapas para estimar o movimento de um agente por odometria visual monocular é determinar o conjunto de transformações $T_{1:n} = T_1, T_2, \cdots, T_n$, que aplicadas a posição inicial P_1 do agente são capazes de desloca-lo até a posição final P_n, como mostrado na Equação 7.8.

$$P_n = P_1 \cdot (T_1 T_2 \cdots T_n) \tag{7.8}$$

Cada matriz de transformação presente no conjunto $T_{1:n}$ pertence ao $\mathbb{R}^4$ e possui o seguinte formato:

$$T_k = \begin{bmatrix} R_k & t_k \\ 0 & 1 \end{bmatrix} \tag{7.9}$$

onde k varia de 1 a n, R_k é a matriz de rotação pertencente ao $\mathbb{R}^3$ e t_k é o vetor de translação pertencente ao $\mathbb{R}^3$. As matrizes de transformação, presentes no conjunto $T_{1:n}$, representam o movimento realizado pelo sensor imageador fixado ao agente e podem ser obtidas pela análise conjunto de imagens $I_{1:n} = I_1, I_2, I_3, \cdots, I_n$.

Para obter a matriz de rotação R_K e o vetor de translação t_k de cada transformação do conjunto $T_{1:n}$, deve-se determinar a matriz Fundamental F de cada par de imagens subsequentes (I_k, I_{k+1}) (HARTLEY; ZISSERMAN, 2004).

A Matriz Fundamental F, de dimensão 3×3 e posto 2 [1], representa a geometria epipolar entre duas imagens de uma mesma cena, ou seja, é a geometria projetiva inerente entre duas imagens, que depende apenas dos parâmetros intrínsecos do sensor imageador e de sua posições relativas (HARTLEY; ZISSERMAN, 2004).

[1] a definição de posto de uma matriz encontra-se no Apêndice A

Segundo Hartley e Zisserman (2004), a principal propriedade da matriz Fundamental F é a condição de correspondência: para qualquer par de pontos correspondentes $x \leftrightarrow x'$ em duas imagens, F satisfaz a seguinte condição:

$$x'^T F x = 0 \tag{7.10}$$

A matriz F pode ser obtida a partir de dois fatores: 1 - condição de correspondência (Equação 7.10); 2 - um conjunto de pares correspondentes. Pela combinação desses dois fatores determina-se um sistema de equações homogêneo, cuja a solução é a matriz Fundamental F. O algoritmo de oito pontos correspondentes é aplicado para encontrar a solução do sistema de equações homogêneo, ou seja, determinar a matriz F (HARTLEY; ZISSERMAN, 2004).

Segundo (HARTLEY; ZISSERMAN, 2004), o algoritmo de oito pontos correspondentes é uma solução linear para a obtenção da Matriz Fundamental (F), e este é composto das seguintes etapas:

1 - Dado um conjunto de m pontos correspondentes $X_{1:m} = (x_1, x_1^{'}), (x_2, x_2^{'}), \cdots, (x_m, x_m^{'})$ de duas vistas diferentes (duas imagens subsequentes da mesma cena), a matriz M, pertencente ao $\mathbb{R}^{m \times 9}$, formada pelo produto de Kronecker $(x_i \bigotimes x_1')$ entre esses pares, é definida por:

$$M = \begin{bmatrix} x_1 \bigotimes x_1' \\ x_2 \bigotimes x_2' \\ \vdots \\ x_m \bigotimes x_m' \end{bmatrix} \tag{7.11}$$

2 - Agora, dado a definição de matriz Fundamental F, e o vetor $f = [f_{11}, f_{12}, f_{13}, f_{14}, f_{15}, f_{16}, f_{17}, f_{18}, f_{19}]$ (vetorização da matriz Fundamental), pode-se escrever a Equação 7.10 da seguinte forma:

$$Mf = 0 \tag{7.12}$$

3 - Se a matriz M possuir exatamente 8 graus de liberdade, o sistema homogêneo exibido na Equação 7.12 tem solução única. Mas, se a matriz M possuir 9 graus de liberdade, a solução do sistema homogêneo (Equação 7.12) é o vetor s (vetor solução) que minimiza o erro da solução do sistema

homogêneo, determinada pelo método dos mínimos quadrados.

4 - Para encontrar o vetor s utiliza-se a Decomposição em Valores Singulares (SVD, do inglês, Singular Value Decomposition) [2] para decompor a matriz M. A decomposição de M em seus valores singulares gera como resposta o seguinte resultado:

$$SVD(M) = U\Sigma V^T \tag{7.13}$$

onde U e V^t são matrizes ortogonais e Σ é uma matriz diagonal, formada pelo valores singulares de M. O vetor s, que minimiza o erro da solução do sistema homogêneo (Equação 7.12), corresponde à coluna $v \in \mathbb{R}^9$ da matriz $V \in \mathbb{R}^{9\times 9}$ que está associada ao menor valor singular de M.

5 - A matriz Fundamental F formada pelos elementos do vetor v tem o seguinte formato:

$$F = \begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{bmatrix} \tag{7.14}$$

Segundo Scaramuzza et al. (2014), o algoritmo de oito pontos correspondes, utilizado para a obtenção da matriz Fundamental, é o mesmo para câmeras esféricas e câmeras estenopeica, desde que estas estejam devidamente calibradas, ou seja, a matriz K de parâmetros intrínsecos é conhecida, e que as correções de distorção sobre as imagens da câmera esférica tenham sido aplicadas.

7.1.4 OBTENÇÃO DA MATRIZ DE ROTAÇÃO E DO VETOR DE TRANSLAÇÃO

A matriz Fundamental F é determinada para cada par de imagens subsequentes (I_k, I_{k+1}), e a partir desta matriz é possível obter a rotação (R_k) e a translação (t_k) do movimento do sensor imageador. Devido ao fato do sensor imageador estar calibrado, a matriz de parâmetros intrínsecos K do sensor é conhecida. Desta forma, é possível definir a matriz Fundamental F da seguinte maneira:

$$F = K^{-T}.R[t]_x.K^{-1} \tag{7.15}$$

[2] Informações sobre a Decomposição de Valores Singulares estão no Apêndice A

onde K^{-T} é a transposta da inversa da matriz de parâmetros intrínsecos, K^{-1} é a matriz inversa dos parâmetros intrínsecos da câmera, $[t]_x$ é a representação matricial do produto cruzado do vetor de translação t e R é a matriz de rotação.

Segundo Hartley e Zisserman (2004), uma outra forma de se escrever a matriz F é dada por

$$F = K^{-T}.E.K^{-1} \tag{7.16}$$

onde E é matriz Essencial definida pela Equação 7.17 (HARTLEY; ZISSERMAN, 2004):

$$E = R.[t]_x \tag{7.17}$$

A matriz Essencial E pode ser decomposta por SVD como:

$$SVD(E) = U\Sigma V^T \tag{7.18}$$

onde U e V^T são matrizes ortogonais e σ é a matriz diagonal $\Sigma = diag\sigma, \sigma, 0$ contendo os valores singulares de E.

Segundo Hartley e Zisserman (2004), a matriz Essencial E difere-se da matriz $[t]_x$ apenas por uma rotação, por este motivo, os valores singulares de $[t]_x$ são os mesmos de E. Desta forma, pode-se definir $[t]_x$ e R pela Equação 7.19 e Equação 7.20, respectivamente.

$$[t]_x = U.W.\Sigma.U^T \tag{7.19}$$

$$R = U.W^T.V^T \tag{7.20}$$

onde V e $U \in \mathbb{R}^{3\times3}$ são as matrizes ortogonais obtidas pela decomposição em valores singulares de E, $W = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$ é a matriz de rotação em torno do eixo z por um angulo de 90°.

7.1.5 REMOÇÃO DE CORRESPONDÊNCIAS INVÁLIDAS

Há no processo de correspondência entre os pontos de interesse alguns fatores que geram casamentos errados. Entre esses fatores pode-se citar: elementos que compõem a cena, que possuem vetor de atributos semelhantes, e elementos que podem ser identificados na imagem como pontos de interesse, devido ao seu movimento independente do movimento do sensor imageador, dando origem a casamentos que não foram criados pelo movimento do sensor. Estes dois fatores ocorrem com grande frequência em imagens de ambientes urbanos, onde há a presença de diversos elemento que se assemelham e se movimentam, como veículos e pedestres (SILVA RAMOS, 2017; NISHITANI, 2015).

As correspondências inválidas, também denominadas de "*outliers*", influenciam de forma negativa a estimação do movimento do sensor imageador e podem levar a uma estimação errada do movimento, se sua presença não for devidamente tratada. As correspondências válidas são chamadas de "*inliers*". A abordagem mais aplicada para a remoção das correspondência inválidas é a aplicação do método RANSAC (FISCHLER; BOLLES, 1987).

O RANSAC tem sido aplicado em diversas pesquisas para a remoção de "*outliers*" entre os pontos de interesse correspondentes e que interferem na estimação da trajetória por odometria visual (KITT et al., 2010). Entre estas pesquisas podem-se citar: o trabalho desenvolvido por (SCARAMUZZA; FRIEDRICH, 2011) que realiza uma completa investigação sobre odometria visual e aplica o algoritmo RANSAC para a remoção de "*outliers*" do conjunto de pontos de interesse correspondentes, a pesquisa de Conte e Doherty (2008) e Rebouças et al. (2013) que também aplicam o RANSAC para melhorar a estimação da trajetória de um VANT obtida por odometria visual monocular.

O RANSAC foi proposto por Fischler e Bolles (1987) para ajustar um modelo a um conjunto de dados contendo um grande número de dados inválidos.

O RANSAC utiliza o método dos mínimos quadrados de forma iterativa e, basicamente, é composto das seguintes etapas: 1 - para cada iteração, um subconjunto dos dados originais é selecionado e um modelo é gerado para este subconjunto; 2 - posteriormente, por meio de uma margem de erro, verifica-se a quantidade de elementos do conjunto de dados que se ajustam ao modelo; 3 - o processo de seleção de um subconjunto dos dados, geração de um modelo para o subconjunto e verificação da quantidade de dados que se ajustam ao modelo é repetido por um número fixo de

iterações; 4 - por fim, a resposta é a iteração que gerou o modelo que ajusta a maior quantidade de dados e que se mantém dentro de um limiar de erro (SILVA RAMOS, 2017).

O RANSAC é um algoritmo não-determinístico, pois produz uma resposta aceitável de acordo com uma probabilidade, e quanto maior o número de iterações do algoritmo maior será esta probabilidade. Além disso, por ser não-determinístico, duas instâncias do algoritmo sobre o mesmo conjunto de dados pode produzir respostas diferentes (FISCHLER; BOLLES, 1987).

A Figura 7.3 apresenta uma aplicação do algoritmo RANSAC para ajustar um conjunto de dados a um modelo de uma reta. Em (A) é exibido o resultado obtido após 5 iterações, com um modelo de reta que ajusta 14 pontos, ou seja, do total de pontos apenas 14 são *inliers*. Em (B) é exibido o resultado obtido após 11 iterações, com um modelo de reta que ajusta ajusta 60 pontos, ou seja, do total de pontos 60 são *inliers*.

Figura 7.3 - Aplicação do algoritmo RANSAC para determinar modelo para ajuste dos dados. O modelo obtido é representado pela linha azul. Pontos em verde são os *inliers*. Pontos em vermelho são os *outliers*. Em (A) é exibido o resultado obtido após 5 iterações, com 14 *inliers*. Em (B) é exibido o resultado obtido após 11 iterações, com 60 *inliers*.

Fonte: Adaptado de Wedge, D. (2012)

No contexto desta tese de doutorado, o RANSAC é aplicado para selecionar, dentro do conjunto de pontos de interesse correspondentes de duas imagens, o subconjunto

que obtém a matriz Fundamental F com o menor erro de projeção, ou seja, que apresenta o melhor resultado para a relação $x'^T F x = 0$. Desta forma, o algoritmo segue as seguintes etapas (as etapas baseiam-se no algoritmo desenvolvido por Kitt et al. (2010)):

1 - O conjunto de pontos de interesse correspondentes (x e x') de duas imagens subsequentes (I e I') são colocados em coordenadas homogêneas, ou seja, a cada ponto, que possui duas dimensões, é atribuída uma nova dimensão com valor 1. Após esta operação os pontos possuem o seguinte formato: $x = (x_i, y_i, 1)^T$ e $x' = (x'_i, y'_i, 1)^T$;

2 - Cada conjunto de pontos é normalizado de modo que a origem esteja no centro e a distância média de cada ponto do conjunto ao centro seja$\sqrt{2}$. Esta normalização deixa o problema de obtenção de F bem condicionado;

3 - Seleciona-se um subconjunto de pontos correspondentes e avalia-se a relação $x'^T F x = 0$. Após um número finito de iterações o conjunto que produzir a melhor resposta são os "*inliers*" utilizados para a obtenção de F

A Figura 7.4 apresenta duas imagens subsequentes, capturadas por uma câmera de vídeo de um VANT durante sua trajetória, e a correspondência dos pontos de interesse, com e sem a presença de "*outliers*". Para a detecção e descrição dos pontos de interesse foi aplicado o algoritmo SURF, e para a remoção dos "*outliers*" o algoritmo RANSAC.

Figura 7.4 - Correspondência de pontos de interesse de duas imagens do VANT. Em A a imagem capturada em um instante t_1. Em B a imagem capturada em um instante t_2. Em C a correspondência com a presença de *outliers*. Em D a correspondência sem a presença de *outliers*.

Fonte: Produção do autor.

7.2 RESULTADOS: DISCUSSÃO E ANÁLISE

O algoritmo de odometria visual monocular aplicado nesta tese de doutorado baseia-se nos trabalhos desenvolvidos por:

- Conte e Doherty (2008) que utiliza a odometria monocular para estimar a trajetória do VANT, o algoritmo aplicado possui as mesmas etapas do algoritmo utilizado nesta tese de doutorado. Porém, o algoritmo de detecção e descrição dos pontos de interesse aplicado é o FAST, que, segundo (SCARAMUZZA; FRIEDRICH, 2012) não é um algoritmo invariante a transformação afim, o que pode prejudicar o resultado da estimação da trajetória do VANT. Além disso, segundo o trabalho desenvolvido por Suaib et al. (2014), que faz um comparativo entre algoritmos de detectores e descri-

tores de ponto de interesse, o algoritmo FAST detecta menos pontos de interesse que o SURF, sendo assim, não é adequado para ser aplicado em imagens de regiões mais homogêneas.

- Rebouças et al. (2013) que também aplica a odometria visual monocular para estimar o posicionamento do VANT, o detector e descritor dos pontos de interesse aplicado é o ORB. Porém, não é feito um estudo para estimar uma trajetória completa de uma aeronave.

Para análise do algoritmo proposto, que estima a trajetória realizada pelo VANT por meio da odometria visual monocular, são considerados diferentes experimentos, que englobam três situações de voo diferentes:

1 - **Situação de voo** 1 - o VANT sobrevoa uma região urbana, em condições suficientes de iluminação do ambiente para que seja aplicado a câmera de vídeo como sensor imageador

2 - **Situação de voo** 2 - o VANT sobrevoa uma região urbana com pouca ou nenhuma iluminação do ambiente, nesta situação, utiliza-se imagens de intensidades provenientes de um sensor LiDAR;

3 - **Situação de voo** 3 - o VANT sobrevoa uma região coberta por água, por exemplo o oceano, nesta situação, utiliza-se imagens do MDS da cobertura do fundo do oceano;

O primeiro experimento realizado, relacionado à situação de voo 1, utiliza um conjunto de dados obtidos por sensor óptico passivo (a câmera de vídeo do VANT) que captura imagens na banda do visível. O conjunto de dados foi obtido durante o voo da aeronave autônoma RMAX. Este conjunto de dados é formado por imagens obtidas durante um voo real de um VANT, e também foi aplicado no experimento para avaliar o algoritmo de estimação da posição do VANT por registro automático de imagens (vide 6.6.2). Para informações adicionais sobre o conjunto de dados deste primeiro experimento, vide seção 6.6.2.

O primeiro experimento tem o objetivo de testar a seguinte hipótese: para a situação de voo 1, o algoritmo proposto de odometria visual monocular estima a trajetória do VANT de forma satisfatória, ou seja, com um erro inferior ao erro do GNSS da aeronave. Segundo Conte e Doherty (2008), o erro do GNSS presente na aeronave RMAX é igual 10 metros.

O segundo experimento realizado, relacionado à situação de voo 2, utiliza um conjunto de dados formado por um conjunto de imagens de intensidade LiDAR, que representam as imagens capturadas pelo VANT durante sua trajetória. Estas imagens, que simulam as imagens capturadas pela câmera LiDAR durante a trajetória do VANT, são subimagens (recortes) de uma imagem de intensidade LiDAR georreferenciada, e também foram aplicadas no experimento que avaliou o algoritmo de estimação da posição do VANT por registro automático utilizando imagens de intensidade LiDAR (vide seção 6.6.3). Para mais detalhes sobre o conjunto de de dados de imagens, vide seção 6.6.3. O segundo experimento é uma simulação da estimação da posição do VANT por odometria visual monocular na situação de voo 2.

O segundo experimento tem o objetivo de testar as seguintes hipóteses:

- primeira hipótese: para a situação de voo 2, com a aplicação do algoritmo proposto de odometria visual monocular, é possível realizar a estimação da trajetória de um VANT utilizando imagens de intensidade LiDAR de uma região urbana, ou seja, é possível obter a matriz Fundamental F de um par de imagens subsequentes da mesma cena. Segundo, Nister et al. (2004), Scaramuzza e Friedrich (2011) e Scaramuzza e Friedrich (2012) com a matriz F e o sensor imageador calibrado (os parâmetros intrínsecos do sensor imageador são conhecidos) é possível obter a matriz de rotação e o vetor de translação do movimento entre um par de imagens subsequentes (NISTER et al., 2004).

- segunda hipótese: para a situação de voo 2, o algoritmo proposto de odometria visual monocular, utilizando imagens de intensidade LiDAR, estima a trajetória do VANT de forma satisfatória, ou seja, com um erro inferior ao erro do GNSS da aeronave. Para testar esta hipótese o erro do GNSS da aeronave RMAX (erro de 10 metros), obtido no estudo desenvolvido por Conte e Doherty (2008), é utilizado como parâmetro.

O terceiro experimento realizado, relacionado com a situação de voo 3, utiliza um MDS da cobertura do fundo do oceano, gerado a partir da nuvem de pontos obtida por um sensor LiDAR, para estimar a trajetória do VANT com a aplicação do algoritmo de odometria visual monocular proposto. Do MDS são obtidas subimagens (recortes) que representam as imagens capturadas por uma câmera LiDAR durante a trajetória do VANT. Estas subimagens são as mesmas aplicadas no experimento

apresentado na seção 6.6.4, que avaliou a possibilidade de realizar a estimação da posição do VANT por registro automático de imagens utilizando o MDS do fundo do oceano. Mais detalhes sobre o conjunto de dados do terceiro experimento são apresentados na seção 6.6.4. O terceiro experimento é uma simulação da estimação da posição do VANT por odometria visual monocular na situação de voo 3.

O terceiro experimento tem o objetivo de avaliar as seguintes hipóteses:

- primeira hipótese: para a situação de voo 3, com a aplicação do algoritmo proposto de odometria visual monocular é possível realizar a estimação da trajetória de um VANT utilizando imagens do MDS do fundo do oceano, ou seja, é possível obter a matriz Fundamental F de um par de imagens subsequentes do MDS. Com a matriz F e o sensor imageador calibrado (os parâmetros intrínsecos do sensor imageador são conhecidos) é possível obter a matriz de rotação e o vetor de translação do movimento (NISTER et al., 2004; SCARAMUZZA; FRIEDRICH, 2011; SCARAMUZZA; FRIEDRICH, 2012).

- para a situação de voo 3, o algoritmo proposto de odometria visual monocular, utilizando imagens do MDS da cobertura do fundo do oceano, estima a trajetória do VANT de forma satisfatória, ou seja, com um erro inferior ao erro do GNSS da aeronave. Para testar esta hipótese o erro do GNSS da aeronave RMAX (erro de 10 metros), obtido no estudo desenvolvido por Conte e Doherty (2008), é utilizado como parâmetro.

7.2.1 MÉTRICAS DE AVALIAÇÃO

A métrica EGM, aplicada para a avaliação da estimação da posição por registro automático de imagens (vide 6.6.1), é também aplicada para comparar a trajetória estimada, obtida pelo algoritmo odometria visual monocular, com a posição fornecida pelo sistema GNSS da aeronave. A posição fornecida pelo GNSS da aeronave é utilizada como verdade sobre o posicionamento do VANT. Se o erro (distância Euclidiana) entre o posicionamento obtido e a verdade for superior a 5 *metros*, esta estimativa de posição fornecida pelo algoritmo não é classificada como um EGM. Entretanto, se o erro for inferior a 5 *metros*, esta estimativa é classificada como EGM. O valor 5 *metros* de limite para a classificação de uma estimativa de posição como EGM é utilizado, pois corresponde a metade do valor máximo de erro do sistema GNSS embarcado no VANT (CONTE; DOHERTY, 2008).

No segundo e terceiro experimento, a trajetória do VANT não foi obtida por meio de um voo real da aeronave, esta trajetória foi obtida por simulação em computador. Desta forma, o erro da trajetória estimada pelo algoritmo de odometria visual monocular é calculado em relação a trajetória da simulação em computador. Porém, o valor de 5 *metros* de limite para a classificação de uma estimativa de posição como EGM também é aplicado.

Segundo Nister et al. (2004) e Scaramuzza e Friedrich (2011) a estimação da trajetória por odometria visual apresenta estabilidade, mesmo assim, a métrica SDGM será utilizada para avaliação nos experimentos. A métrica SDGM avalia a estabilidade do algoritmo de estimação de posição por meio de uma janela, que considera as 30 ultimas estimações obtidas. Dos valores contidos na janela, calcula-se o desvio padrão do erro de estimação, se este valor for inferior a 2 metros, o ponto, a partir do qual foi obtida a janela, é classificado como SDGM. A métrica SDGM também foi aplicada para avaliar o método de estimação da posição do VANT por registro automático de imagens.

A métrica GM também é aplicada, pois esta é resultado da combinação das métricas EGM e SDGM. Um ponto que é classificado como EGM e SDGM é classificado como GM.

7.2.2 RESULTADOS: PRIMEIRO EXPERIMENTO

O conjunto de dados do primeiro experimento é formado por 1443 imagens obtidas por um VANT em tempo de voo, cada uma dessas imagens representa um ponto para avaliação do algoritmo proposto. Estas imagens são provenientes de um voo real da aeronave autônoma RMAX (Figura 6.18) sobre a região de Revinge, Suécia, e também foram aplicadas no experimento que investigou a estimação do posicionamento do VANT por registro automático de imagens obtidas por uma câmera de vídeo (vide seção 6.6.2).

Neste primeiro experimento, o VANT realizou um percurso de aproximadamente 1 km de extensão, e durante este percurso sua câmera de vídeo capturou 1443 imagens. Para cada imagem, com auxilio do GNSS da aeronave, foi obtido seu posicionamento geográfico. Considerando que o VANT ocupa o pixel central da imagem capturada por seu sensor imageador, este pixel possui a mesma posição geográfica do VANT no momento em que a imagem é capturada. A Figura 7.5 exibe algumas das imagens capturadas pela câmera de vídeo do helicóptero autônomo RMAX durante sua trajetória.

Figura 7.5 - Imagens capturadas pela câmera de vídeo do helicóptero autônomo RMAX durante sua trajetória

Fonte: Produção do autor.

A câmera de vídeo utilizada na captura das imagens tem modelo de projeção perspectiva da câmera estenopeica e sua matriz de parâmetros intrínsecos é

$$K = \begin{bmatrix} f_x & 0 & p_x \\ 0 & f_y & p_y \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 472 & 0 & 180 \\ 0 & 518 & 144 \\ 0 & 0 & 1 \end{bmatrix} \tag{7.21}$$

onde f_x e f_y são as distâncias focais em pixels da câmera em x e y, respectivamente. Os valores de p_x e p_y são a posição do centro óptico da câmera. A matriz de parâmetros intrínsecos da câmera pode ser obtida por informações do fabricante ou por um processo de calibração da câmera estenopeica (PEDRINI; SCHARTZ, 2008; NISTER et al., 2004). Neste experimento, as correções devido as distorções da lente não são aplicadas, pois a câmera utilizada possui uma grande distância focal, portanto as distorções na imagem são consideradas pequenas para a precisão requerida (CONTE; DOHERTY, 2008)

A Figura 7.6 exibe a trajetória em duas dimensões realizada pelo VANT. O círculo preto, posição $(0, 0)$ do sistemas de coordenadas, marca o ponto de partida da aeronave. O círculo azul, posição $(-40, -80)$ do sistemas de coordenadas, marca o ponto de chegada da aeronave.

Figura 7.6 - Trajetória realizada pelo helicóptero autônomo RMAX. A trajetória possui aproximadamente 1 km de extensão.

Fonte: Produção do autor.

A Figura 7.7 apresenta a trajetória real do VANT sobre uma imagem georreferenciada da banda pancromática do satélite Quickbird, que contém a região sobrevoada pela aeronave. A trajetória realizada pelo VANT está na cor vermelha.

Figura 7.7 - Região sobrevoada pelo helicóptero autônomo RMAX e sua trajetória marcada em vermelho. O círculo preto marca o ponto de partida e o branco o ponto de chegada da aeronave.

Fonte: Produção do autor.

A Figura 7.8 exibe a trajetória, em duas dimensões, obtida pelo algoritmo de odometria visual monocular proposto. A trajetória em vermelho representa o voo real do VANT, determinada pelo GNSS da aeronave. Em azul é apresentada a trajetória do VANT estimada pelo algoritmo de odometria visual monocular proposto. O ponto de partida do VANT é a posição $(0, 0)$ do sistemas de coordenadas, e o ponto de chegada é a posição $(-40, -80)$ do sistemas de coordenadas.

Figura 7.8 - Trajetória real do VANT em vermelho. Trajetória do VANT obtida por odometria visual monocular em azul.

Fonte: Produção do autor.

A Figura 7.9 apresenta a trajetória real do VANT, em vermelho, e a trajetória estimada pelo algoritmo de odometria visual monocular, em azul, sobre uma imagem de satélite que contém a região sobrevoada pela aeronave. O círculo na cor preta marca o ponto de partida e o círculo em cor branca representa o ponto de chegada da aeronave.

Figura 7.9 - Imagem de satélite com a região sobrevoada pelo helicóptero autônomo RMAX e sua trajetória marcada em vermelho. A trajetória estimada pelo algoritmo de odometria visual monocular está representada na cor azul.

Fonte: Produção do autor.

A Figura 7.10 exibe o gráfico do erro de estimação de trajetória em função do tempo, ou seja, em função de cada um dos 1443 pontos de avaliação. A linha tracejada em vermelho representa um erro igual a 5 metros, pontos abaixo dessa linha são classificados como EGM. Neste experimento, 667 pontos foram classificados com EGM, ou seja, $46,2\%$ dos pontos de avaliação. Devido a estabilidade da odometria visual monocular 1413 pontos foram classificados como SDGM, ou seja, 100% dos pontos de avaliação possíveis da trajetória foram classificados como SDGM, lembrando que a métrica SDGM utiliza uma janela das últimas 30 estimações, assim, o primeiro ponto da trajetória que pode ser avaliado como SDGM é o ponto 30. No experimento, o número de pontos classificados como GM é igual 637, $45,1\%$ dos pontos.

Figura 7.10 - Erro de estimação da trajetória obtida pelo algoritmo de odometria visual monocular proposto.

Fonte: Produção do autor.

7.2.2.1 RESULTADOS: TEMPO DE EXECUÇÃO

O método de odometria visual monocular foi implementado em linguagem C com auxílio da biblioteca OpenCV. A Tabela 7.2 exibe o tempo médio obtido para obtenção da trajetória (obtenção da matriz de rotação e vetor de translação) entre duas imagens subsequentes. O tempo médio foi obtido para dois computadores: o Raspberry PI Model B-1 e o Zybo ZINQ 7000, ambos podem ser embarcados em um VANT.

Tabela 7.2 - Tempo médio de para obtenção da trajetória entre duas imagens subsequentes

Computador	Tempo de Execução (s)
Raspberry PI B1	0.48
Zybo ZINQ 7000	0.33

Por meio da análise dos resultados da Tabela 7.2 é possível observar que o computador Raspberry PI B1 realiza 2 estimações da trajetória, em média, por segundo, enquanto que o computador Zybo ZINQ 7000 realiza, em média, 3 estimações por segundo. O tamanho da imagem capturada pela câmera de vídeo da aeronave é igual 288×360 pixels. O descritor SURF detectou, em média, 51 pares de pontos de interesse correspondentes em todos os pares de imagens subsequentes da trajetória.

7.2.2.2 RESULTADOS: ANÁLISE DO RESULTADO DO PRIMEIRO EXPERIMENTO

Considerando o conjunto de dados do experimento, a análise do resultados obtidos no primeiro experimento demonstram que o método proposto de odometria visual monocular pode ser aplicado para estimar a trajetória do VANT utilizando imagens capturadas por uma câmera de vídeo, durante sua trajetória. O resultado demonstra o que as pesquisas realizadas por Nister et al. (2004) e Scaramuzza e Friedrich (2011) estão corretas sobre o erro de deriva presente na odometria visual e sobre o método ser estável em sua estimação de trajetória.

Apesar do erro de deriva, $46,2\%$ dos pontos de avaliação foram classificados com EGM, ou seja, obtiveram um erro de estimação inferior a 5 metros, e 72%0 (1038 pontos) obtiveram um erro de estimação inferior ao erro do GNSS da aeronave (10 metros). Além disso, o algoritmo SURF, para todos os pares de imagens do conjunto de dados, detectou e fez a correspondência de no mínimo 8 pares de ponto de interesse, o que permite a estimação da trajetória por odometria visual em todo seu percurso. Desta forma, a hipótese (– o algoritmo proposto de odometria visual monocular estima a trajetória do VANT de forma satisfatória, ou seja, com um erro inferior ao erro do GNSS da aeronave. –) para o primeiro experimento mostra-se verdadeira, pois o resultado obtido possui um erro menor que o GNSS da aeronave na maior parte do percurso.

O erro de deriva presente neste resultado, obtido pelo algoritmo de odometria visual monocular, pode tornar o método inseguro para trajetórias muito longas. Desta forma, a utilização de outro método de estimação da trajetória juntamente com a odometria visual monocular pode ser mais adequado.

7.2.3 RESULTADOS: SEGUNDO EXPERIMENTO

O segundo experimento é a simulação da trajetória do VANT utilizando uma câmera LiDAR para capturar imagens de intensidade da região sobrevoada. O conjunto de dados do segundo experimento é formado por 172 imagens de intensidade LiDAR, que representam as imagens capturadas pela câmera LiDAR embarcada no VANT durante sua trajetória. Cada umas das 172 imagens são subimagens de uma imagem de intensidade LiDAR georreferenciada. Estas imagens de intensidade LiDAR são as mesmas aplicadas no experimento, realizado no capítulo 6, que avaliou o algoritmo de estimação da posição geográfica do VANT por registro automático de imagens (vide seção 6.6.3).

As 172 imagens de intensidade LiDAR são utilizadas como pontos de avaliação do algoritmo proposto de odometria visual monocular para estimar a trajetória de um VANT, que navega sobre uma região urbana em condições de baixa ou sem luminosidade. Além disso, estas imagens são utilizadas para avaliar se é possível aplicar um algoritmo descritor e detector de pontos de interesse, como o SURF, em imagens de intensidade LiDAR, e assim, obter a matriz Fundamental F entre duas imagens subsequentes. A Figura 7.11 exibe algumas das imagens de intensidade LiDAR que representam a trajetória da aeronave.

Figura 7.11 - Imagens de intensidade que representam as imagens de capturadas pela câmera LiDAR embarcada na aeronave durante sua trajetória.

Fonte: Produção do autor.

A Figura 7.12 exibe a trajetória, em duas dimensões, realizada pelo VANT durante a simulação de voo. A trajetória simulada possui aproximadamente 1 km de extensão e foi realizada sobre uma imagem de intensidade LiDAR georreferenciada. Da imagem georreferenciada, foram obtidas 172 subimagens, que representam as imagens capturadas pela câmera LiDAR do VANT durante sua trajetória. O círculo preto, posição $(200, 525)$ do sistemas de coordenadas, representa o ponto de partida do VANT. O círculo azul, posição $(1260, 525)$ do sistemas de coordenadas, representa o ponto de chegada do VANT.

Figura 7.12 - Trajetória, em duas dimensões, realizada pelo VANT na simulação utilizando imagens de intensidade LiDAR.

Fonte: Produção do autor.

A Figura 7.13 exibe a trajetória simulada do VANT, linha em vermelho, na imagem de intensidade LiDAR georreferenciada. O círculo amarelo representa o ponto de partida e o círculo branco o ponto de chegada do VANT.

Figura 7.13 - Trajetória realizada pelo VANT exibida na imagem de intensidade LiDAR georreferenciada.

Fonte: Produção do autor.

Durante a trajetória, a quantidade mínima de pontos de interesse correspondentes encontrados em cada par de imagens subsequentes foi igual a 17. Desta forma, em nenhum momento da trajetória houve um par de imagens subsequentes onde não

fosse possível obter a matriz Fundamental F. A Figura 7.14 exibe um exemplo de um par de imagens de intensidade LiDAR com os pontos interesse correspondentes destacados.

Figura 7.14 - Casamento entre pontos de interesse em imagens de intensidade LiDAR

Fonte: Produção do autor.

A imagem de intensidade LiDAR georreferenciada do conjunto de dados deste experimento foi obtida por meio de um processo de interpolação pelo vizinho mais próximo [3] sobre a nuvem de pontos capturada por um sensor LiDAR aerotransportado. Os sensores LiDAR aerotransportados são diferentes das câmeras LiDAR, pois as imagens geradas no sensor LiDAR aerotransportado são geradas sobre uma nuvem de pontos sem a necessidade de correção de distorção, entretanto, para uma correta aplicação das imagens de uma câmera LiDAR as correções de distorções geradas devem ser feitas no momento de obtenção da imagem (MIRAZAEI et al., 2012).

Os parâmetros intrínsecos do sensor LiDAR empregado para a obtenção da imagem intensidade LiDAR georreferenciada deste experimento não são conhecidos, pois a imagem foi obtida por um processo de interpolação sobre a núvem de pontos. Para simular a obtenção da matriz de rotação e do vetor de translação entre um par de imagens subsequentes foi aplicado o método de Besl e McKay (1992) [4], que determina a transformação (matriz de rotação e vetor de translação) sofrida por um conjunto de pontos correspondentes. Este método foi selecionado, pois segundo Hartley e Zisserman (2004), o processo de obtenção da matriz de rotação e do vetor de

[3]Informações sobre métodos de interpolação aplicados em nuvem de pontos de sensor Lidar podem ser obtidas em Ashraf et al. (2017)

[4]Informações sobre o método de Besl e McKay (1992) encontram-se no Apêndice B

translação, utilizando a decomposição em valores singulares da matriz Fundamental e a matriz de parâmetros intrínsecos, é baseado no algoritmo de Besl e McKay (1992). A diferença entre eles é que o conhecimento dos parâmetros intrínsecos permite uma estimação da posição mais precisa (HARTLEY; ZISSERMAN, 2004).

No contexto desta tese de doutorado, o algoritmo de Besl e McKay (1992) é aplicado para simular a obtenção da matriz de rotação e do vetor de translação, entre os pontos de interesses correspondentes de duas imagens, no caso em que não há o conhecimento dos parâmetros intrínsecos do sensor imageador.

O gráfico da Figura 7.15 apresenta o gráfico de posição da trajetória simulada do VANT, em vermelho, e o gráfico da posição da trajetória estimada, pelo método de odometria visual monocular, em azul. A trajetória simulada, em vermelho, é utilizada como verdade sobre a posição do VANT. A posição $(200, 525)$ do sistemas de coordenadas representa o ponto de partida do VANT. A posição $(1260, 525)$ do sistemas de coordenadas representa o ponto de chegada do VANT.

Figura 7.15 - Gráfico de posição da trajetória simulada do VANT, em vermelho, e o gráfico de posição da trajetória estimada por odometria visual monocular, em azul.

Fonte: Produção do autor.

A Figura 7.16 exibe a trajetória simulada, em vermelho, e a trajetória estimada por odometria visual monocular, em azul, na imagem de intensidade LiDAR georreferenciada. A trajetória simulada, em vermelho, é utilizada como verdade sobre a posição do VANT. O círculo amarelo marca o ponto de partida e o círculo branco o ponto de chegada do VANT.

Figura 7.16 - Trajetória simulada do VANT, em vermelho, e a trajetória estimada por odometria visual monocular, em azul.

Fonte: Produção do autor.

A Figura 7.17 exibe o gráfico do erro da trajetória estimada por odometria visual monocular em função do tempo (número do ponto de avaliação). O erro é a distância euclideana entre cada ponto de avaliação da trajetória simulada e cada ponto da trajetória estimada por odometria visual monocular. A linha tracejada em vermelho representa o erro igual a 5 metros, pontos abaixo dessa linha são classificados como EGM.

Figura 7.17 - Gráfico do erro da trajetória estimada do VANT por odometria visual monocular. A linha tracejada vermelha representa um erro de 5 metros.

Fonte: Produção do autor.

Todos os 172 pontos de avaliação, 100% dos pontos, foram classificados como EGM. Devido a estabilidade da odometria visual, 142 pontos de avaliação foram classificados como SDGM, ou seja, 100% dos pontos possíveis de avaliação para a métrica SDGM. Neste experimento, 142 pontos foram classificados como GM.

7.2.3.1 RESULTADOS: ANÁLISE DO RESULTADO DO SEGUNDO EXPERIMENTO

Considerando o conjunto de dados utilizado, a análise do resultado obtido no segundo experimento demonstram que o algoritmo proposto de odometria visual monocular pode ser aplicado para estimar a trajetória do VANT utilizando imagens de intensidade LiDAR, pois para todos os pares de imagens subsequentes foi possível determinar a matriz Fundamental F. O algoritmo SURF demonstrou ser adequado para a detecção e descrição dos pontos de interesse em imagens de intensidade LiDAR. Este resultado demonstra que a primeira hipótese (– com a aplicação do algoritmo proposto de odometria visual monocular é possível realizar a estimação da trajetória de um VANT utilizando imagens de intensidade LiDAR de uma região urbana, ou seja, é possível obter a matriz Fundamental F de um par de imagens subsequentes da mesma cena –) para o segundo experimento é verdadeira. Além disso, o resultado do segundo experimento demonstra que a partir da utilização de uma câmera LiDAR calibrada pode-se obter o movimento do VANT.

Tendo em conta o conjunto de dados aplicado, no segundo experimento,100% dos pontos de avaliação da trajetória foram classificados como EGM, o que demonstra que a segunda hipótese (– o algoritmo proposto de odometria visual monocular, utilizando imagens de intensidade LiDAR, estima a trajetória do VANT de forma satisfatória, ou seja, com um erro inferior ao erro do GNSS da aeronave –) do experimento é verdadeira. Neste experimento também é possível observar o erro de deriva que a estimação da trajetória por odometria visual possui, além disso, também é possível observar sua estabilidade. O erro de deriva presente na odometria visual pode tornar o método inseguro para trajetórias muito longas, assim, a utilização de outro método de estimação da trajetória juntamente com a odometria visual pode ser mais adequado.

7.2.4 RESULTADOS: TERCEIRO EXPERIMENTO

O terceiro experimento é a simulação da trajetória do VANT utilizando uma câmera LiDAR para gerar o MDS do fundo de uma região coberta por água, e assim permitir sua navegação sobre o regiões como o oceano. O conjunto de dados do terceiro expe-

rimento é formado por 92 subimagens do MDS do fundo do oceano, que representam os mapas gerados pela câmera LiDAR embarcada no VANT durante sua trajetória. Cada uma das 92 subimagens são recortes de um MDS georreferenciado do fundo do oceano. Estas subimagens do MDS do fundo do oceano são as mesmas aplicadas no experimento realizado no capítulo 6, que avaliou o algoritmo de estimação da posição geográfica do VANT por registro automático de imagens do MDS do fundo do oceano (vide seção 6.6.4).

As 92 subimagens do MDS do fundo do oceano são utilizadas como pontos de avaliação do algoritmo proposto de odometria visual monocular, para estimar a trajetória de um VANT que sobrevoa regiões cobertas por água. Além disso, estas subimagens são utilizadas para avaliar se é possível aplicar um algoritmo descritor e detector de pontos de interesse, como o SURF, em MDS gerado por um sensor LiDAR, e assim, obter a matriz Fundamental F entre duas subimagens subsequentes. A Figura 7.18 exibe algumas das subimagens do MDS que representam a trajetória da aeronave.

Figura 7.18 - Exemplo de subimagens do MDS do fundo do oceano que representam a trajetória do VANT

Fonte: Produção do autor.

A Figura 7.19 exibe a trajetória, em duas dimensões, realizada pelo VANT durante a simulação de voo. A trajetória simulada possui aproximadamente 1 km de extensão e foi realizada sobre um MDS georreferenciado, gerado a partir de uma nuvem de pontos de um sensor LiDAR. Do MDS georreferenciado foram obtidas 92 subimagens, que representam os MDS gerados pela câmera LiDAR embarcada na aeronave durante a sua navegação sobre uma região coberta por água, por exemplo, o oceano.

O círculo preto, posição $(70, 238)$ do sistemas de coordenadas, marca o ponto de partida do VANT. O círculo azul, posição $(640, 238)$ do sistemas de coordenadas, marca o ponto de chegada do VANT.

Figura 7.19 - Trajetória, em duas dimensões, realizada pelo VANT na simulação de navegação sobre uma região coberta por água.

Fonte: Produção do autor.

A Figura 7.20 exibe a trajetória simulada do VANT, linha em vermelho, no MDS georreferenciado do fundo do oceano. O círculo amarelo representa o ponto de partida e o círculo branco o ponto de chegada do VANT.

Figura 7.20 - Trajetória realizada pelo VANT exibida no MDS georreferenciado do fundo do oceano obtido por sensor LiDAR.

Fonte: Produção do autor.

Durante a trajetória, a quantidade mínima de pontos de interesse correspondentes, encontrados em cada par de imagens subsequentes, foi igual a 13. Desta forma, em nenhum momento da trajetória houve um par de imagens subsequentes do MDS do fundo do oceano em que não fosse possível obter a matriz Fundamental F. A Figura 7.21 exibe um exemplo de um par de subimagens do MDS do fundo do oceano com os pontos interesse correspondentes destacados.

Figura 7.21 - Correspondência entre pontos de interesse entre dois MDS subsequentes do fundo do mar.

Fonte: Produção do autor.

O MDS do conjunto de dados deste experimento foi obtido por meio de um processo de interpolação sobre a nuvem de pontos, capturada por um sensor LiDAR aerotransportado. Pelos mesmos motivos apresentados na seção 7.2.3, o método apresentado em (BESL; MCKAY, 1992) foi aplicado para obter a matriz de rotação e o vetor de translação entre duas imagens subsequentes, utilizando os pontos de interesse correspondentes.

A Figura 7.22 apresenta o gráfico de posição da trajetória simulada do VANT, em vermelho, e o gráfico da posição da trajetória estimada pelo método de odometria visual monocular, em azul. A posição $(70, 238)$ do sistemas de coordenadas marca o ponto de partida do VANT. A posição $(640, 238)$ do sistemas de coordenadas marca o ponto de chegada do VANT.

Figura 7.22 - Gráfico de posição da trajetória simulada do VANT, em vermelho, e o gráfico de posição da trajetória estimada por odometria visual monocular, em azul.

Fonte: Produção do autor.

A Figura 7.23 exibe a trajetória simulada, em vermelho, e a trajetória estimada por odometria visual monocular, em azul, no MDS georreferenciado do fundo do oceano.

Figura 7.23 - Trajetória simulada do VANT, em vermelho, e a trajetória estimada por odometria visual monocular, em azul.

Fonte: Produção do autor.

O gráfico da Figura 7.24 exibe o erro da trajetória estimada pela odometria visual monocular em função do tempo (número do ponto de avaliação). A trajetória simulada é utilizada como verdade sobre a posição do VANT. O erro é a distância

euclideana entre cada ponto de avaliação trajetória simulada e cada ponto da trajetória estimada por odometria visual monocular. A linha tracejada em vermelho representa o erro igual a 5 metros, pontos abaixo dessa linha são classificados como EGM.

Figura 7.24 - Trajetória simulada do VANT, em vermelho, e a trajetória estimada por odometria visual monocular, em azul.

Fonte: Produção do autor

Neste experimento, todos os 92 (100%) pontos de avaliação foram classificados como EGM. Devido a estabilidade da odometria visual, 62 pontos de avaliação foram classificados como SDGM e 62 pontos foram classificados como GM, ou seja, 100% dos pontos possíveis foram classificados nas duas métricas.

7.2.4.1 RESULTADOS: ANÁLISE DO RESULTADO DO TERCEIRO EXPERIMENTO

Considerando o conjunto de dados aplicado, o resultado obtido no terceiro experimento demonstra que o algoritmo proposto de odometria visual monocular pode ser aplicado para estimar a trajetória do VANT utilizando MDS de uma região coberta por água, neste caso específico, o fundo do oceano. Está conclusão é possível, pois para todos os pares de imagens subsequentes foi possível determinar a matriz Fundamental F. O algoritmo SURF demonstrou ser adequado para a detecção e descrição dos pontos de interesse em MDS do fundo do oceano.

O resultado também demonstra que a primeira hipótese (– com a aplicação do

algoritmo proposto de odometria visual monocular é possível realizar a estimação da trajetória de um VANT utilizando imagens do MDS do fundo do oceano –) para o terceiro experimento é verdadeira, pois foi possível detectar e descrever pontos de interesse em duas subimagens subsequentes e obter a matriz Fundamental de todos os pares de subimagens. Além disso, o resultado do terceiro experimento demonstra que a partir da utilização de uma câmera LiDAR calibrada, que gera imagens do MDS da região sobrevoada, pode-se obter o movimento do VANT.

Tendo em conta o conjunto de dados, no terceiro experimento, 100% dos pontos de avaliação da trajetória obtiveram um erro inferior a 5 metros e foram classificados como EGM, o que demonstra que a segunda hipótese (– o algoritmo proposto de odometria visual monocular, utilizando imagens do MDS da cobertura do fundo do oceano, estima a trajetória do VANT de forma satisfatória –) do experimento é verdadeira. No mesmo experimento também é possível observar o erro de deriva que a estimação da trajetória por odometria visual possui. Além disso, também é possível observar a estabilidade da odometria visual. Mas, o erro de deriva presente na odometria visual pode tornar o método inseguro para trajetórias muito longas, assim, a utilização de outro método de estimação da trajetória juntamente com a odometria visual pode ser mais adequado.

8 ESTIMAÇÃO DA POSIÇÃO DE VANT POR FUSÃO DE DADOS

Os animais na natureza reconhecem o ambiente ao seu redor avaliando os sinais que recebem dos seus diferentes sensores. Os seres humanos, por exemplo, combinam as informações que recebem do meio externo, por meio dos cinco sentidos (visão, audição, paladar, tato e olfato), para criar uma percepção mais precisa do ambiente ao seu redor (KLEIN, 2004)

A partir da observação que combinar diferentes tipos de dados de diferentes fontes representa uma vantagem biológica de uma espécie, os cientistas passaram a aplicar esta ideia nas mais diversas áreas do conhecimento (ELMENREICH, 2002).

Há uma grande quantidade de termos relacionados com a fusão de dados. Terminologias como fusão de sensores, fusão de informação, fusão de dados de múltiplos sensores e integração de múltiplo sensores são empregadas em diversas áreas do conhecimento (CHANG-SUN; IEE-KI, 2003).

Segundo Krout (2012), o termo fusão de informação engloba a teoria, técnicas e ferramentas concebidas para explorar a sinergia da informação adquirida por diferentes fontes, sejam essas, sensores, banco de dados ou informação coletada por um humano, para que a tomada de decisão seja melhor do que considerando a informação de uma fonte isolada.

O termo fusão de dados de múltiplo sensores é definido por Waltz e Llinias (1999) como a tecnologia desenvolvida para combinar dados de múltiplos sensores a fim de realizar inferência sobre um evento, atividade ou situação.

Segundo Elmenreich (2002), a integração de múltiplos sensores utiliza diferentes fontes para produzir uma combinação dos dados em um único formato sem se preocupar em obter uma representação melhor do ambiente ou auxiliar na tomada de decisão.

O termo fusão de dados é utilizado pela comunidade de geociência e sensoriamento remoto para definir técnicas que combinam a informação de diversas fontes para obtenção de um dado final, o qual representa melhor o sistema observado (SCHMITT; ZHU, 2016). Mais recentemente, o termo passou a ser utilizado em robótica, segundo Nazemzadeh et al. (2017), a fusão de dados corresponde a combinação de dados de diferentes sensores embarcados em um robô, para que este realize a tarefa para qual foi desenvolvido de forma mais eficiente e segura do que quando considera-se a utilização dos sensores isoladamente.

Segundo Gustafsson (2013), a fusão de sensores é a combinação de dados de diferentes sensores, ou de dados distintos de um mesmo sensor, tal que o resultado final seja melhor do que quando considera-se os dados de maneira isolada. Já para Elmenreich (2002), fusão de dados e a fusão de sensores são consideradas subáreas da fusão de informação. Para Khaleghi et al. (2013), não importa qual o termo é aplicado, fusão de dados ou fusão de sensores, pois ambos referem-se a uma representação melhor da informação que quando a fonte de dados é considerada de forma isolada.

No contexto desta tese de doutorado, é utilizada a definição de fusão de dados fornecida por Conte e Doherty (2008), que a define como o conjunto de técnicas aplicadas para combinar dados de fontes diferentes ou dados distintos de uma mesma fonte para obter uma resposta melhor do sistema, quando comparada com a resposta obtida considerando os dados de forma isolada. A palavra melhor, na definição de fusão de dados fornecida por Conte e Doherty (2008), está relacionada com uma resposta com menor incerteza da medição. A incerteza de uma informação (ou medida) aumenta quando ocorre oclusão do objeto observado ou quando uma informação é ambígua.

A fusão de dados é uma técnica que tem sido amplamente empregada para melhorar a estimação da posição de robôs móveis, principalmente de VANTs (CHANG-SUN; IEE-KI, 2003). Segundo Gustafsson (2013), duas técnicas que podem ser utilizadas para realizar a fusão de dados são o Filtro de Kalman e o Método Sequencial de Monte Carlo (MSMC), também conhecido como Filtro de Partículas (FP). Estes dois algoritmos são métodos baseados na filtragem Bayesiana, que é uma técnica probabilística aplicada para realizar a fusão de dados.

Segundo Castanedo (2013), as técnicas de fusão de dados podem ser divididas em três categorias distintas: (i) associação de dados, cujo o objetivo é estabelecer o conjunto de observações ou medições que são geradas pelo mesma fonte de dados ao longo do tempo; (ii) estimação de estado, que é o conjunto de técnicas que visa determinar o estado de um objeto em movimento (normalmente a posição) a partir das observações; (iii) fusão de decisão, é o conjunto de técnicas cujo o objetivo é realizar uma inferência sobre os eventos e atividades que são produzidos a partir de um objeto de interesse. O problema de estimação da posição do VANT por fusão de dados relaciona-se com a subdivisão de estimação de estados. Para Castanedo (2013), as técnicas mais comuns aplicadas para a estimação de estados são os algoritmos de Filtro de Kalman e FP.

Atualmente, o FP tem sido empregado para realizar a fusão de dados de técnicas de

estimação da posição por visão computacional. Como nas pesquisas desenvolvidas por Conte e Doherty (2008) e Chunhui et al. (2014), que aplicam o FP para a fusão de dados provenientes da estimação da posição por odometria visual e registro de imagens para obter um cálculo mais preciso do posicionamento do VANT.

No contexto deste trabalho, aplica-se a fusão de dados do registro automático de imagens e da odometria visual monocular para a estimação da trajetória de VANT. Uma variação do algoritmo de FP, denominado de Filtro de Partículas Não-Extensivo (FPNE), é aplicado para a fusão de dados. Nas próximas seções é realizada a fundamentação teórica relacionada com a fusão de dados por FP, com enfase na estimação Bayesiana e Filtro de Partículas, em seguida apresenta-se a metodologia utilizada para a produção dos resultados. Para finalizar o capítulo, os resultados obtidos pela aplicação da metodologia são apresentados.

8.1 ESTIMAÇÃO BAYESIANA

A estimação Bayesiana, ou filtragem Bayesiana, é uma abordagem probabilística, que estima uma Função Densidade de Probabilidade (FDP)[1] desconhecida, recursivamente ao longo do tempo. O processo de estimação utiliza como parâmetros de entrada os dados de medidas observacionais e os dados de um modelo matemático (FURTADO, 2008).

Nesta tese de doutorado, a teoria de estimação Bayesiana é baseada na representação de um sistema, denominado modelo. Segundo Monteiro (2006), o modelo é um conjunto de objetos agrupados devido a alguma interação e há relação de causa e efeito nos fenômenos que ocorrem com esses objetos. Quando algumas grandezas que caracterizam os objetos do modelo variam no tempo o modelo é dito dinâmico.

Segundo Furtado (2008), a necessidade de modelos apropriados é fundamental. O modelo é dito apropriado quando satisfaz o objetivo para o qual foi proposto, ou seja, o modelo deve descrever as propriedades essenciais do conjunto de objetos e ser simples o bastante para que possa ser escrito em forma de um algoritmo.

Em um modelo dinâmico cada valor assumido em um determinado tempo é denominado estado e o conjunto de todos os estados é denominado espaço de estado (TULSYAN et al., 2016). No contexto deste trabalho, como não se pode realizar medidas infinitesimais, as observações sobre cada estado e medida são obtidas em tempo discreto. Portanto, considera-se apenas modelos em tempo discreto.

[1] Conceitos ligados a teoria de probabilidade encontram-se no Apêndice C.

A Figura 8.1 apresenta um modelo de espaço de estado, com os estados indicados por x_t e medidas (ou observações) indicadas por y_t.

Figura 8.1 - Modelo de espaço de estados. O modelo é dinâmico e em tempo discreto.

Fonte: Adaptado de Tulsyan et al. (2016).

Uma propriedade relacionada com o modelo de espaço de estados apresentado na Figura 8.1 é a propiedade de Markov, definida como: um modelo de espaço de estados em tempo discreto possui a propriedade de Markov se para uma dada sequência de estados $x_{0:t-1} \equiv x_0, x_1, \cdots, x_{t-1}$, o estado atual x_t somente depende do estado anterior a ele x_{t-1}, ou seja,

$$p(x_{t+1}|x_0, \cdots, x_t) = p(x_{t+1}|x_t) \tag{8.1}$$

A Equação 8.1 demonstra que a ocorrência do estado x_t contém todas as informações a respeito do passado necessárias para determinar o estado futuro x_{t+1}. Segundo Jazwinski (1970) esta propriedade pode ser denominada como o *princípio da casualidade generalizada*, que define: o futuro pode ser previsto caso o presente seja conhecido.

Na Figura 8.1, cada x_k, com $k = 1, \cdots, t+1$, é um vetor que contém os valores relacionado ao estado, e y_K, com $k = 1, \cdots, t+1$, é um vetor que contém os valores das observações relacionadas ao estado x_k. Em estimação Bayesiana, tais vetores são variáveis aleatórias e são representados em termos de uma FDP $p(x_t)$ e $p(y_t)$. A probabilidade condicional $p(x_t|y_{1:t})$ representa a FDP do estado x_t após obtida as

observações $y_{1:t} \equiv y_1, y_2, \cdots, y_t$, e é a solução para a estimação Bayesiana.

Em um modelo de espaço de estados dinâmicos, o valor de $p(x_t|y_{1:t})$ deve ser calculado recursivamente a cada instante de tempo e a cada nova observação obtida. Caso a dinâmica do modelo seja conhecida é possível utilizá-la para realizar a previsão do estado x_t por $p(x_t|x_{t-1})$, denominada transição entre os estados x_{t-1} e x_t.

A estimação Bayesiana consiste em determinar a probabilidade a posteriori $p(x_t|y_{1:t})$ dado a transição de estado $p(x_t|x_{t-1})$ e $p(y_t|x_t)$, que é a probabilidade das medidas y_t dado um estado x_t, também denominada verossimilhança das medidas. Normalmente, a FDP de transição de estado ($p(x_t|x_{t-1})$) e a FDP de verossimilhança das medidas ($p(y_t|x_t)$) são obtidas do modelo de espaço de estados dinâmicos (TULSYAN et al., 2016).

A probabilidade a posteriori $p(x_t|y_{1:t})$ pode ser calculada aplicando a propriedade de Markov e o teorema de Bayes, da seguinte forma:

$$p(x_t|y_t) = \frac{p(y_t|x_t)p(x_t|y_{1:t-1})}{p(y_t|y_{1:t-1})} \tag{8.2}$$

onde $p(y_t|y_{1:t-1})$ é uma constante de normalização denominada verossimilhança marginal. Segundo Tulsyan et al. (2016), em problemas de estimação Bayesiana, a constante de normalização não é explicitamente conhecida, portanto, pode ser ignorada. Desta forma, a equação 8.2 pode ser reescrita da seguinte forma:

$$p(x_t|y_{1:t}) \propto p(y_t|x_t)p(x_t|y_{1:t-1}) \tag{8.3}$$

Além disso, aplicando marginalização [2] e considerando que x_t tenha dimensão n, a FDP $p(x_t|y_{t-1})$ é obtida por:

$$p(x_t|y_{1:t-1}) = \int_{\mathbb{R}^n} p(x_t|x_{t-1})p(x_{t-1}|y_{t-1})dx_{t-1} \tag{8.4}$$

onde $p(x_t|x_{t-1})$ e $p(x_{t-1}|y_{t-1})$ são a transição e a probabilidade a *posteriori* no instante $t-1$, respectivamente.

Conhecendo-se o valor de $p(x_t|y_{1:t-1})$, obtém-se o valor de $p(x_t|y_{1:t})$ por:

[2] O conceito de distribuição de probabilidade marginal encontra-se no Apêndice C

$$p(x_t|y_{1:t}) = \frac{p(y_t|x_t)p(x_t|y_{1:t-1})}{\int_{\mathbb{R}^n} p(y_t|x_t)p(x_t|y_{1:t-1})dx_t} \tag{8.5}$$

Segundo Doucet et al. (2001), a Equação 8.4 é a equação de previsão, e a Equação 8.5 é a equação de atualização. Estas equações constituem a base dos algoritmos sequenciais de Monte Carlo.

Quando o valor de $p(x_t|y_t)$ é obtido pode-se calcular a calcular o valor esperado de alguma função g das variáveis de estado, $I(g(x))$, de acordo com:

$$I(g(x_t)) = E(g(x_t)|y_{1:t}) = \int_{\mathbb{R}^n} g(x_t)p(x_t|y_{1:t})dx_t \tag{8.6}$$

A estimação Bayesiana pode ser aplicada para obter-se a probabilidade a *posteriori* de um modelo de espaço de estados linear e não-linear (DOUCET et al., 2001). Um modelo de espaço de estado é dito linear quando a dinâmica do modelo é linear e os ruídos são gaussianos (GUSTAFSSON, 2013). Um modelo de espaço de estado é dito não-linear quando a dinâmica do modelo é não-linear e os ruídos são não-gaussianos (GUSTAFSSON, 2013).

Um modelo de espaço de estado não-linear pode ser representado da seguinte forma:

$$\begin{aligned} x_t &= f(x_{t-1}, u_{t-1}) + v \\ y_t &= h(x_t, u_t) + e \end{aligned} \tag{8.7}$$

em que t é o tempo discreto, v e e são variáveis aleatórias independentes, definidas como ruído do processo e ruído das observações, respectivamente. A função f descreve a evolução das variáveis de estado e a função h descreve o sistema de aquisição de dados (instrumento de obtenção das observações).

Considerando que o vetor de estado x_t tenha dimensão n, a solução da estimação Bayesiana recursiva para o modelo da Equação 8.7 é:

$$p(x_t|y_{t-1}) = \int_{\mathbb{R}^n} p_v(x_t - f(x_{t-1}, u_{t-1}))p(x_{t-1}|y_{t-1})dx_{t-1} \tag{8.8}$$

$$\alpha_t = \int_{\mathbb{R}^n} p_e(y_t - h(x_t, u_t))p(x_t|y_{t-1})dx_t \tag{8.9}$$

$$p(x_t|y_t) = \alpha_t^{-1} p_e(y_t - h(x_t, u_t))p(x_t|y_{t-1}) \tag{8.10}$$

O objetivo da recursão exibida nas Equações 8.8–8.10 é estimar a probabilidade a *posteriori* $p(x_t|y_t)$. A Equação 8.8 representa a atualização do tempo (equação de previsão) e a Equação 8.10 denota o sistema de observações (equação de atualização). As funções p_v e p_e são as FDPs do ruído de processo e observações, respectivamente. A recursão deve ser inicializada com $p(x_1|y_0) = p(x_0)p(x_0)$ que é conhecida como probabilidade *a priori*.

Contudo, há um problema com a solução apresentada. Em geral, é difícil calcular analiticamente as soluções para as integrais exibidas nas Equações 8.8 – 8.10(FURTADO, 2008). A grande maioria das soluções analíticas ocorre quando o modelo dinâmico é linear e os ruídos são gaussianos. Considerando o modelo linear de espaço de estado, os ruídos v e e gaussianos e a FDP $p(x_0)$ normalmente distribuída, a solução ótima da estimação Bayesiana pode ser obtida pelo Filtro de Kalman (KALMAN, 1960).

O Filtro de Kalman representa FDPs em termos da média e da covariância e esta solução é tratada como uma solução paramétrica (KALMAN, 1960).

Em caso de não-linearidade do modelo de espaço de estado, uma solução aproximada pode ser obtida pela linearização do modelo por uma expansão de Taylor. Após a linearização, o Filtro Kalman pode ser aplicado. Esta solução é denominada *Filtro de Kalman Estendido* (WELCH; BISHOP, 1995).

Segundo Conte e Doherty (2008), o filtro de Kalman Estendido pode não ser aplicável ou não suficientemente preciso para a situação tratada, devido a não-linearidade e as incertezas envolvidas no modelo. Portanto, outros métodos podem ser aplicados para este propósito. O mais popular, pertencente a classe de soluções não-paramétricas, é o Filtro de Partículas.

8.2 FILTRO DE PARTÍCULAS

O MSMC, ou Filtro de Partículas, busca estimar recursivamente a FDP $p(x_t|y_{1:t})$. A teoria Bayesiana define que a FDP $p(x_t|y_{1:t})$ contém todas as informações estatísticas da variável de estado x_t baseado nas informações contidas nas medidas $y_{1:t}$ (FURTADO, 2008). A FDP estimada pelo FP pode ser utilizada para obter-se diversas estimativas das variáveis de estado, de acordo com:

$$I(g(x_t)) = E(g(x_t)|y_{1:t}) = \int_{\mathbb{R}^n} g(x_t)p(x_t|y_{1:t})dx_t \tag{8.11}$$

Segundo Tulsyan et al. (2016), o MSMC representa uma FDP desconhecida por um conjunto de amostras com seus pesos associados. Este conjunto de amostras é denominado partículas, por esta razão o nome Filtro de Partículas. Basicamente, o FP aproxima a FDP $p(x_t|y_{1:t})$ por uma função de densidade empírica, definida por:

$$p(x_t|y_{1:t}) \approx \sum_{i=1}^{M} \widetilde{w}^{(i)} \delta(x_t - x_{t|s}^{(i)}) \tag{8.12}$$

$$\sum_{i=1}^{M} \widetilde{w}^{(i)} = 1 \tag{8.13}$$

$$\widetilde{w}^{(i)} \leq 1, \forall i \tag{8.14}$$

onde $\delta(.)$ é a função delta de Dirac e $\widetilde{w}^{(i)}$ denota os pesos por importância associados a cada partícula $x_t^{(i)}$.

A função Delta de Dirac apresenta a seguinte propriedade:

$$F(x^{(i)}) = \int_{-\infty}^{+\infty} F(x)\delta(x - x^{(i)})dx \tag{8.15}$$

onde

$$\delta(x - x^{(i)}) = \begin{cases} 0, \text{ se } x \neq x^{(i)} \ , \\ \infty, \text{ se } x = x^{(i)} \ . \end{cases} \tag{8.16}$$

O FP deriva de algumas ideias, como Amostragem Perfeita e Amostragem por Importância, que são discutidas nas próximas seções.

8.2.1 AMOSTRAGEM PERFEITA

Amostragem Perfeita é um algoritmo que fornece amostras garantidas de serem distribuídas de acordo com uma função densidade de interesse. O objetivo da Amostragem Perfeita é obter um estimativa para $I(g(x_t))$ (Equação 8.11) baseado na hipótese que há um conjunto de amostras de uma função densidade alvo $t(x)$ (*target density*) desconhecida. O conjunto de amostra tem tamanho M e é definido por

$[x^{(i)}]_i^M$, e toda amostra é independentemente e identicamente distribuída.

Segundo Doucet et al. (2001), a Amostragem Perfeita é não realística, mas a ideia é a base do algoritmo de FP. De acordo com Doucet et al. (2001), uma aproximação empírica da FDP $t(x)$ pode ser obtida por:

$$\hat{t}(x) = \sum_{i=1}^{M} \frac{1}{M} \delta(x - x^{(i)}) \ . \tag{8.17}$$

A partir da Equação 8.17, uma aproximação de $I(g(x))$ pode ser obtida por

$$\hat{I}(g(x)) = \hat{E}\{g(x)|y\} = \int g(x)\hat{t}(x)dx \tag{8.18}$$

$$\hat{E}\{g(x)|y\} = \int g(x) \sum_{i=1}^{M} \frac{1}{M} \delta(x - x^{(i)}) \tag{8.19}$$

$$\hat{E}\{g(x)|y\} = \frac{1}{M} \sum_{i=1}^{M} \int g(x)\delta(x - x^{(i)}) \tag{8.20}$$

$$\hat{E}\{g(x)|y\} = \frac{1}{M} \sum_{i=1}^{M} g(x^i) \tag{8.21}$$

onde $\delta(x - x^{(i)})$ é a função delta de Dirac em $x^{(i)}$ e é nula para todo $x^{(i)} \neq x$.

Segundo Doucet et al. (2001), a aproximação $\hat{I}(g(x))$ é raramente válida e pode ser obtida apenas utilizando-se um grande número de amostras. Uma alternativa para a *Amostragem Perfeita* é a aplicação da *Amostragem por Importância*.

8.2.2 AMOSTRAGEM POR IMPORTÂNCIA

A *Amostragem por Importância* utiliza uma FDP alternativa, que seja conhecida, para se extrair amostras e realizar a aproximação de $I(g(x))$ (FURTADO, 2008).

Devido ao fato de que não se pode gerar amostras diretamente da FDP alvo $t(x)$, emprega-se uma FDP alternativa $s(x)$, denominada FDP proposta, para gerá-las. A partir do momento que uma amostra é gerada, a partir de $s(x)$, pode-se avaliar se esta foi gerada da FDP alvo e, assim, considerar se é ou não uma amostra de $t(x)$ (FURTADO, 2008).

Na Amostragem por Importância, a Equação 8.18 pode ser escrita da seguinte forma:

$$\hat{I}(g(x)) = \hat{E}\{g(x)|y\} = \int g(x)\frac{t(x)}{s(x)}s(x)dx_t \tag{8.22}$$

onde $s(x)$ é a FDP proposta. A Equação 8.22 pode ser reescrita como:

$$\hat{I}(g(x)) = \hat{E}\{g(x)|y\} = \int g(x)w(x)s(x)dx_t \tag{8.23}$$

Segundo Schon (2003), a aproximação para $I(g(x))$ pela geração de $M >> 1$ amostras $[x^{(i)}]_{i=1}^{M}$ de $s(x)$ é dada por:

$$\hat{I}(g(x)) = \frac{1}{M}\sum_{i=1}^{M} w(x^{(i)})g(x^{(i)}) \tag{8.24}$$

onde

$$w(x^{(i)}) = \frac{t(x^{(i)})}{s(x^{(i)})} \quad \text{(sendo: } i = 1, \cdots, M) \ . \tag{8.25}$$

O valor $w(x^{(i)})$ é chamado de peso por importância, que é a informação a respeito da possibilidade da amostra gerada, a partir de $s(x)$, ser associada a $t(x)$. Desta forma, o valor de $w(x^{(i)})$ funciona como uma probabilidade de aceitação da amostra.

Segundo Tulsyan et al. (2016), na *Amostragem por Importância* o peso por importância $w(x^{(i)})$ deve ser normalizado da seguinte forma:

$$\tilde{w}(x_t^{(i)}) = \frac{p(y_t|x_{t|t-1}^{(i)})}{\sum_{i=1}^{M} p(y_t|x_{t|t-1}^{(i)})} = \frac{w(x_t^{(i)})}{\sum_{i=1}^{M} w(x_t^{(i)})} \ . \tag{8.26}$$

8.2.3 ALGORITMO DE FILTRO DE PARTÍCULAS

O algoritmo de FP origina-se da ideia da Amostragem por Importância. Neste texto, considera-se o algoritmo introduzido por Gordon et al. (1993). Há outras variações do algoritmo, que podem ser encontradas em Doucet et al. (2001), Arulampalam et al. (2002) e Schon (2003).

Segundo Gordon et al. (1993), o algoritmo de FP é um problema de estimação

Bayesiana, cuja FDP alvo é definida por:

$$t(x_t) = p(x_t|y_{1:t}) \tag{8.27}$$

Para encontrar a FDP alvo, utiliza-se o teorema de Bayes e a regra de Markov, assim, a Equação 8.27 pode ser escrita da forma:

$$p(x_t|y_{1:t}) = \frac{p(y_t|x_t)p(x_t|y_{1:t-1})}{p(y_t|y_{1:t-1})} \tag{8.28}$$

$$p(x_t|y_{1:t}) \propto p(y_t|x_t)p(x_t|y_{1:t-1}) \tag{8.29}$$

Desta forma, são definidos o peso por importância $w(x_t)$ e a FDP proposta $s(x_t)$.

$$w(x_t) = p(y_t|x_t) \tag{8.30}$$

$$s(x_t) = p(x_t|y_{1:t-1}) \tag{8.31}$$

A primeira etapa do algoritmo é sua **Inicialização**, que consiste da extração de amostras (partículas) da FDP a *priori* $p(x_0)$. As partículas obtidas nesta etapa são denominadas *partículas estimadas*.

A etapa seguinte do algoritmo consiste da **Atualização das medidas**. Nesta etapa, obtém-se o peso por importância normalizado de cada partícula, que designa uma probabilidade. Esta probabilidade é calculada a partir de uma função de verossimilhança entre as partículas preditas e as observações, obtidas pela Equação 8.7, ou seja, o valor y_t. O valor desta probabilidade descreve o quão provável é obter uma previsão dado a observação.

As partículas preditas $[x^{(i)}_{t|t-1}]^M_{i=1}$ são geradas a partir da aplicação da dinâmica do modelo sobre as partículas estimadas no instante de tempo anterior $[x^{(i)}_{t-1|t-1}]^M_{i=1}$. Seguindo (GORDON et al., 1993), as equações abaixo implementam as ideias expostas acima:

$$s(x_t) = p(x_t|y_{1:t-1}) = \int p(x_t|x_{t-1})p(x_{t-1}|y_{1:t-1})dx_{t-1} \tag{8.32}$$

$$p(x_t|y_{1:t-1}) \approx \int p(x_t|x_{t-1}) \sum_{i=1}^{M} \frac{1}{M} \delta(x_{t-1} - x_{t-1|t-1}^{(i)})dx_{t-1} \tag{8.33}$$

$$p(x_t|y_{1:t-1}) \approx \sum_{i=1}^{M} \frac{1}{M} \int p(x_t|x_{t-1})\delta(x_{t-1} - x_{t-1|t-1}^{(i)})dx_{t-1} \tag{8.34}$$

$$p(x_t|y_{1:t-1}) \approx \sum_{i=1}^{M} \frac{1}{M} p(x_t|x_{t-1}) \tag{8.35}$$

Devido ao fato do algoritmo ser derivado da *Amostragem por Importância*, o peso por importância normalizado $\tilde{w}_t^{(i)}$ de cada partícula é calculado da seguinte forma:

$$\tilde{w}_t^{(i)} = \frac{w_t^{(i)}}{\sum\limits_{i=1}^{M} w_t^{(i)}} \tag{8.36}$$

O peso por importância representa uma medida do quão provável uma partícula pode ter sido gerada pela FDP alvo. A partícula e o seu peso por importância correspondente são uma aproximação da FDP alvo.

Posteriormente, realiza-se a **Reamostragem**, que consiste da extração de um novo conjunto de partículas $[x_{t|t}^{(i)}]_{i=1}^{M}$ para substituir as partículas estimadas $[x_{t|t-1}^{(i)}]_{i=1}^{M}$. Desta forma, as partículas obtidas da reamostragem tornam-se as partículas estimadas. Este novo conjunto de partículas é gerado de acordo com:

$$Pr\{x_{t|t}^{(i)} = x_{t|t-1}^{(i)}\} = \tilde{w}_t^{(i)} \quad (i = 1, \cdots, M) \tag{8.37}$$

Há diversos algoritmos que podem ser aplicados no processo de reamostragem, mas, segundo Arulampalam et al. (2002), o algoritmo mais apropriado é o de reamostragem sistemática.

A etapa final do algoritmo é o **Cálculo da estimativa**, que é a média das novas partículas geradas na re-amostragem.

Caso as etapas descritas anteriormente sejam recursivamente repetidas no tempo, obtém-se a seguinte aproximação

$$p(x_t|y_{1:t}) \approx \sum_{i=1}^{M} \frac{1}{M} \delta(x_t - x_{t|t}^{(i)}) \tag{8.38}$$

que representa de fato o algoritmo de Filtro de Partículas introduzido por Gordon et al. (1993). A Tabela 8.1 apresenta um resumo do algoritmo de Filtro de Partículas descrito.

Tabela 8.1 - Algoritmo de Filtro de Partícula

Algoritmo 1: Filtro de Partícula

1 - **Inicialização:** Para $i = 1, \cdots, M$ inicializa-se as partículas $[x_{0|t-1}^{(i)}]_{i=1}^{M} \sim N(0, \sigma^2)$

2 - **Atualização da medida:** Para $i = 1, \cdots, M$.

[2.1] - Obtenção das partículas preditas $[x_{t|t-1}^{(i)}]_{i=1}^{M} = f([x_{t-1|t-1}^{(i)}]_{i=1}^{M})$

[2.2] - Cálculo da diferença entre previsão e observação $z_t^{(i)} = y_t - [x_{t|t-1}^{(i)}]_{i=1}^{M}$

[2.3] - Cálculo do peso de importância $[w_t^{(i)}] = \frac{1}{\sigma_z\sqrt{2\pi}} e^{k_i}$ onde $k_i = \frac{-(z-\bar{z})^2}{2\sigma_z^2}$, $\sigma_z^2 = \frac{1}{M-1} \sum z_t^i$ e $\bar{z} = \frac{1}{M} \sum z_t^i$

[2.4] normalização do peso por importância $\tilde{w}_t^i = \frac{w_t^{(i)}}{\sum w_t^{(i)}}$

3 - **Reamostragem:** Para $i = 1, \cdots, M$ extrair M partículas $([x_{t|t}^{(i)}]_{i=1}^{M})$, com substituição, de acordo com: $Pr\{x_{t|t}^{(i)} = x_{t|t-1}^{(i)}\} = \tilde{w}_t^{(i)}, i = 1, \cdots, M$

4 - **Calcula-se a estimativa:** que é a média das partículas geradas na reamostragem $\tilde{x} = E\{x_{t|t}^{(i)}\}$

5 - Itera para o passo 2

8.3 FILTRO DE PARTÍCULAS NÃO-EXTENSIVO

O FP é uma estratégia já consolidada para a estimação de estados em um modelo de espaço de estados não-linear (SCHON, 2003). Mas há um problema intrínseco em sua formulação: a aplicação de uma função de verossimilhança na etapa de **Atualização**

da medida (CAMPOS VELHO; FURTADO, 2011). No contexto da teoria de estimação Bayesiana, a função de verossimilhança fornece uma indicação de quanto o modelo adotado descreve o sistema do mundo real, o qual é representado pelas observações (SCHON, 2003).

Uma solução proposta por CAMPOS VELHO e Furtado (2011), para a escolha da função de verossimilhança a ser aplicada no FP, é a utilização da distribuição de Tsallis. No algoritmo apresentado na Tabela 8.1, na etapa de **Atualização da medida**, foi aplicada a distribuição Gaussiana para representar a função de verossimilhança. Sua utilização justifica-se devido ao teorema central do limite, na hipótese de que trabalha-se com um processo estocástico de variáveis aleatórias estatisticamente independentes e com todos os seus momentos estatísticos bem definidos, ou seja, a integral da FDP da variável aleatória converge para um valor finito. Entretanto, utilizando-se a distribuição gaussiana, o resultado desejado da estimação da FDP $p(x_t|y_{1:t})$ pode não ser alcançado, caso esta não tenha uma variância e/ou outros momentos estatísticos definidos. Dois exemplos de distribuições que não possuem variância definida são as distribuições de Levy e a de Cauchy[3].

A solução proposta por CAMPOS VELHO e Furtado (2011), aplica a distribuição de Tsallis para permitir que o algoritmo de FP encontre uma FDP aproximada de $p(x_t|y_{1:t})$, mesmo que está não possua variância definida.

A forma não-extensiva para a entropia apresentada por Tsallis é dada por (TSALLIS, 1999):

$$S_q(p) = \frac{k}{q-1}\left[1 - \sum_{i=1}^{N}(p_i)^q\right] \tag{8.39}$$

onde p_i é a probabilidade e q é um parâmetro livre denominado parâmetro de não-extensividade. Na termodinâmica, o parâmetro k é conhecido como constante de Boltzmann. No limite de $q \to 1$, a entropia de Tsallis se reduz à formula usual de Boltzmann-Gibbs-Shannon.

Segundo (TSALLIS, 1999), as diferentes distribuições $p(d)$ para diferentes valores de q são dadas por:

[3]Informações sobre a distribuição de Levy e Cauchy encontram-se no Apêndice C.

$$\text{Para } q = 1: \quad p(d) = \left[\frac{1}{2\pi\sigma^2}\right]^{1/2} e^{-d^2/(2\sigma^2)} \ , \tag{8.40}$$

$$\text{Para } q > 1: \quad p_q(d) = \frac{1}{\sigma}\left(\frac{q-1}{\pi(3-q)}\right)^{\frac{1}{2}} \frac{\Gamma\left(\frac{1}{q-1}\right)}{\Gamma\left[\frac{3-q}{2(q-1)}\right]} \left[1 + \frac{q-1}{3-q}\left(\frac{d}{\sigma}\right)^2\right]^{\frac{1}{1-q}} \ , \tag{8.41}$$

$$\text{Para } q < 1: \quad p_q(d) = \frac{1}{\sigma}\left(\frac{1-q}{\pi(3-q)}\right)^{\frac{1}{2}} \frac{\Gamma\left(\frac{5-3q}{2-2q}\right)}{\Gamma\left(\frac{2-q}{1-q}\right)} \left[1 - \frac{1-q}{3-q}\left(\frac{d}{\sigma}\right)^2\right]^{\frac{1}{1-q}} \ . \tag{8.42}$$

A Figura 8.2 apresenta o conjunto de distribuições obtidas para a formulação proposta por Tsallis para diferentes valores de q. Quando $q = 1$, a distribuição corresponde à distribuição gaussiana e quando $q = 2$ à distribuição de Cauchy.

Figura 8.2 - Distribuição de Tsallis para diferentes valores do parâmetro livre q.

Fonte: Produção do autor.

Devido a aplicação da distribuição de Tsallis durante a etapa de **Atualização de medida**, o algoritmo de FP é denominado algoritmo de Filtro de Partículas Não-Extensivo (FPNE). Com esta atualização, o algoritmo do FPNE é exibido na Tabela 8.2.

Tabela 8.2 - Algoritmo de Filtro de Partículas Não-Extensivo

Algoritmo 2: Filtro de Partículas Não-Extensivo

1 - **Inicialização:** Para $i = 1, \cdots, M$ inicializa-se as partículas $[x_{0|t-1}^{(i)}]_{i=1}^{M} \sim N(0, \sigma^2)$

2 - **Atualização da medida** Para $i = 1, \cdots, M$.

[2.1] - Obtenção das partículas preditas $[x_{t|t-1}^{(i)}]_{i=1}^{M} = f([x_{t-1|t-1}^{(i)}]_{i=1}^{M})$

[2.2] - Cálculo da diferença entre previsão e observação $z_t^{(i)} = y_t - [x_{t|t-1}^{(i)}]_{i=1}^{M}$

[2.3] - Determinar o valor de q

[2.4] - Cálculo do peso de importância $[w_t^{(i)}]$ que depende do valor de q. Se $q = 1$, o valor de $[w_t^{(i)}]$ é dado pela Equação 8.40, neste caso o algoritmo é igual ao apresentado na Tabela 8.1. Se $q > 1$, o valor de $[w_t^{(i)}]$ é dado pela Equação 8.41. Se $q < 1$, o valor de $[w_t^{(i)}]$ é dado pela Equação 8.42

[2.5] normalização do peso por importância $\tilde{w}_t^i = \frac{w_t^{(i)}}{\sum w_t^{(i)}}$

3 - **Reamostragem:** Para $i = 1, \cdots, M$ extrair M partículas $([x_{t|t}^{(i)}]_{i=1}^{M})$, com substituição, de acordo com: $Pr\{x_{t|t}^{(i)} = x_{t|t-1}^{(i)}\} = \tilde{w}_t^{(i)}, i = 1, \cdots, M$

4 - **Calcula-se a estimativa**, que é a média das partículas geradas na reamostragem $\tilde{x} = E\{x_{t|t}^{(i)}\}$

5 - Itera para o passo 2

Para a determinação do valor de q (item 2.3 da Tabela 8.2) é aplicado o algoritmo da Tabela 8.3. Uma outra estratégia para a determinação do valor de q pode ser encontrada em (CAMPOS VELHO; FURTADO, 2011), que utiliza um segundo FP para alcançar o objetivo.

Tabela 8.3 - Algoritmo para determinar o melhor valor de q

Algoritmo 3: Melhor valor de q

1 - Gera-se cem números reais aleatórios e uniformemente distribuídos entre zero e três. $0 \leq q \geq 3$, onde $q \sim U(100)$.

2 - Seleciona um dos valores de q gerado na etapa 1

3 - Realiza-se uma variação de $q = q + 0.05q$ e de $q = q - 0.05q$, em torno do valor selecionado.

4 - Aplica-se o algoritmo de Filtro de Partícula Não-Extensivo para o três valore de q.

5 - Armazena-se a resposta do Filtro de Partícula Não-Extensivo.

6 - Itera para a etapa 2 até terminarem os valores de q

7 - Compara-se as respostas do Filtro de Partícula Não-Extensivo e seleciona-se o de melhor resposta para o problema.

8.4 FILTRO DE PARTÍCULAS NÃO-EXTENSIVO PARA ESTIMAÇÃO DA POSIÇÃO DO VANT

No contexto desta pesquisa, o algoritmo de FPNE é aplicado para realizar a fusão dos dados provenientes da estimação da posição por registro automático de imagens e da estimação da posição por odometria visual monocular, com a objetivo de se obter uma melhor estimativa da posição do VANT do que quando considera-se estas fontes de dados isoladamente. A Figura 8.3 apresenta a metodologia aplicada nesta pesquisa para realizar a fusão de dados pelo FPNE.

A Figura 8.3 apresenta as etapas realizadas até a obtenção do posicionamento da aeronave pelo FPNE. Duas imagens da região sobrevoada e capturadas em instantes diferentes, T e $T - d_t$, são aplicadas para a estimação da posição por odometria visual monocular. Paralelamente, a área de correlação é obtida da imagem georreferenciada e com a imagem da região sobrevoada, capturada no instante T, são aplicadas no processo de estimação da posição do VANT por registro automático de imagens. O posicionamento obtido pelas duas técnicas sofrem uma fusão pelo FPNE. O resultado da estimação obtido pelo FPNE é a estimação final da posição do VANT. Espera-se que este resultado seja melhor que os resultados obtidos pelas duas técnicas separadamente.

Figura 8.3 - Metodologia aplicada para a obter a estimação da posição do VANT por fusão de dados utilizando o Filtro de Partículas Não-Extensivo.

Fonte: Produção do autor

A metodologia utilizada nesta pesquisa é baseada nas pesquisas desenvolvidas por Conte e Doherty (2008) e Chunhui et al. (2014). Ambas aplicam o FP para realizar a fusão dos dados da estimação da posição por odometria visual monocular e registro automático de imagens.

A aplicação do FP para a estimação de uma FDP de uma distribuição desconhecida em modelos não-lineares necessita que este seja composto por uma função de atualização do tempo (equação de previsão, ou dinâmica do modelo) e uma função de atualização das observações (equação de atualização).

Na metodologia desenvolvida por Conte e Doherty (2008) e Chunhui et al. (2014), na aplicação do FP para estimar a posição do VANT, a odometria visual monocular fornece a dinâmica do modelo (equação de previsão), pois esta técnica fornece a matriz de rotação R e o vetor de translação t do movimento do sensor imageador fixado ao agente. E o registro automático de imagens representa a função de atualização das medidas, pois fornece informação sobre o posicionamento absoluto do sensor imageador fixado ao agente. O algoritmo de estimação de posição do VANT por FP aplicado em Conte e Doherty (2008) e Chunhui et al. (2014) é apresentado na Tabela 8.4.

Tabela 8.4 - Algoritmo de FP para estimação da posição do VANT

Algoritmo 4: FP utilizando Odometria e Registro de Imagens

1 - **Inicialização:** - inicialização das partículas, cria-se M partículas segundo a regra $[x_{0|t-1}^{(i)}]_{i=1}^{M} \sim N(0, \sigma^2)$

2 - **Atualização da medida** - Para cada partícula M, ou seja, $i = 1, \cdots, M$

[2.1] - Obtenção das partículas preditas. Com este objetivo aplica-se da dinâmica obtida pela odometria visual monocular no conjunto de partículas. $[x_{t|t-1}^{(i)}]_{i=1}^{M} = f([x_{t-1|t-1}^{(i)}]_{i=1}^{M})$ A função $f(.)$ representa a dinâmica obtida pela odometria visual.

[2.2] - Cálculo da diferença entre predição e observação. A observação y_t é a estimação de posição fornecida pelo registro automático de imagens. $z_t^{(i)} = y_t - [x_{t|t-1}^{(i)}]_{i=1}^{M}$

[2.3] - Cálculo do peso por importância de cada partícula. Este cálculo é realizado utilizando uma função de verossimilhança em função da diferença entre predição e observação. Normalmente, a função Gaussiana $g(.)$ é aplicada como função de verossimilhança $[w_t^{(i)}] = g(z_t^{(i)})$

[2.4] - Normalização do peso por importância de cada partícula $\tilde{w}_t^i = \frac{w_t^{(i)}}{\sum w_t^{(i)}}$.

3 - **Reamostragem** - extrair M partículas com substituição de acordo com $Pr\{x_{t|t}^{(i)} = x_{t|t-1}^{(i)}\} = \tilde{w}_t^{(i)}, i = 1, \cdots, M$

4 - **Calcula-se a estimativa**. Obtenção da posição estimada, que é a média das posições partículas geradas na reamostragem $\tilde{x} = E\{x_{t|t}^{(i)}\}$

5 - Itera para o passo 2

A principal justificativa para a aplicação do FP, para realizar a estimação da posição do VANT, advém da não-linearidade das técnicas de visão computacional aplicadas. Segundo Conte e Doherty (2008) e Chunhui et al. (2014), a principal fonte de não-linearidade são oriundas do sensor imageador da aeronave. Além disso, a distribuição dos dados do resultado do registro automático de imagens por correlação cruzada é não-Gaussiana, isto pode ser observado na Figura a seguir (Figura 8.4).

Figura 8.4 - Gráfico da correlação cruzada entre a imagem capturada pelo VANT e a área de correlação

Fonte: Produção do autor.

A principal diferenças entre as metodologias de fusão de dados por FP das pesquisas desenvolvidas por (CONTE; DOHERTY, 2008) e (CHUNHUI et al., 2014) e a metodologia aplicada nesta tese de doutorado é a utilização do FPNE. Espera-se que a aplicação do FPNE, para a fusão dos dados da odometria e registro de imagens, melhore os resultados da estimação, quando comparados aos resultados da estimação obtida pelo FP.

O algoritmo de FPNE aplicado nesta tese de doutorado para estimar a posição do VANT, por meio da fusão dos dados da odometria e do registro de imagens, é exibido na Tabela 8.5

Tabela 8.5 - Algoritmo de Filtro de Partículas Não-Extensivo para estimação da posição do VANT

Algoritmo 5: Filtro de Partículas Não-Extensivo utilizando odometria e registro de imagens

1 - **Inicialização:** - inicialização das partículas, cria-se M partículas segundo a regra $[x^{(i)}_{0|t-1}]^M_{i=1} \sim N(0, \sigma^2)$

2 - **Atualização da medida** - Para cada partícula M, ou seja, $i = 1, \cdots, M$

[2.1] - Obtenção das partículas preditas. Com este objetivo aplica-se da dinâmica obtida pela odometria visual monocular no conjunto de partículas. $[x^{(i)}_{t|t-1}]^M_{i=1} = f([x^{(i)}_{t-1|t-1}]^M_{i=1})$ A função $f(.)$ representa a dinâmica obtida pela odometria visual.

[2.2] - Cálculo da diferença entre predição e observação. A observação y_t é a estimação de posição fornecida pelo registro automático de imagens. $z^{(i)}_t = y_t - [x^{(i)}_{t|t-1}]^M_{i=1}$

[2.3] - Determinação do valor do q. Algoritmo exibido na Tabela 8.2.

[2.4] - Cálculo do peso por importância de cada partícula. Este cálculo é realizado utilizando uma função de verossimilhança em função da diferença entre predição e observação. A função de verossimilhança empregada depende do valor de q. $q = 1$ $[w^{(i)}_t]$ Equação 8.40. $q > 1$ $[w^{(i)}_t]$ Equação 8.41. $q < 1$ $[w^{(i)}_t]$ Equação 8.42.

[2.5] - Normalização do peso por importância de cada partícula $\tilde{w}^i_t = \frac{w^{(i)}_t}{\sum w^{(i)}_t}$.

3 - **Reamostragem** - extrair M partículas com substituição de acordo com $Pr\{x^{(i)}_{t|t} = x^{(i)}_{t|t-1}\} = \tilde{w}^{(i)}_t, i = 1, \cdots, M$

4 - **Calcula-se a estimativa**. Obtenção da posição estimada, que é a média das posições partículas geradas na reamostragem $\tilde{x} = E\{x^{(i)}_{t|t}\}$

5 - Itera para o passo 2

8.5 RESULTADOS: DISCUSSÃO E ANÁLISE

Para a análise da metodologia aplicada para a estimação da posição do VANT, por meio da fusão de dados da odometria visual monocular e do registro automático de imagens utilizando o FPNE, são considerados diferentes experimentos.

No primeiro experimento, é avaliada a possibilidade de realizar a estimação da po-

sição de um VANT por FPNE, quando a aeronave sobrevoa uma região urbana em condições de iluminação que permitam a captura de imagens, da região sobrevoada, por sua câmera de vídeo. Neste experimento, a fusão de dados é realizada a partir do registro de imagens e da odometria visual considerando o conjunto de dados obtidos pela sensor imageador que captura imagens na banda do visível (câmera de vídeo da aeronave). Este conjunto de dados é o mesmo aplicado no primeiro experimento da estimação de posição obtida por registro automático de imagens (vide seção 6.6.2) e o mesmo aplicado no primeiro experimento da estimação da posição obtida por odometria visual monocular (vide 7.2.2).

No primeiro experimento, a fusão de dados por FPNE utiliza o resultado da odometria visual, apresentado na seção 7.2.2, e o resultado obtido pela melhor configuração do algoritmo de registro automático de imagens, apresentado na seção 6.6.2.6. O melhor resultado de estimação por registro de imagens corresponde ao obtido pelo algoritmo configurado com o filtro da mediana, para a remoção dos ruídos indesejáveis, e a MLP em LUT para a detecção de bordas nas imagens (vide 6.6.2.6).

O primeiro experimento tem o objetivo de avaliar as seguintes hipóteses:

- Tendo em conta o conjunto de dados do experimento, a estimação da posição obtida pela fusão dos dados da odometria e registro de imagens pelo FPNE produz resultados satisfatórios para a estimação da posição do VANT, ou seja, a maioria dos pontos de avaliação da trajetória são classificados com um erro inferior ao erro máximo do GPS da aeronave RMAX. Segundo Conte e Doherty (2008), o erro máximo do GPS presente no helicóptero autônomo RMAX é igual a 10 metros.

- Considerando o conjunto de dados obtido pela câmera de vídeo da aeronave, a estimação da posição obtida pela fusão dos dados por meio do FPNE obtém melhores resultados que as técnicas de estimação por odometria e registro de imagens, quando consideradas isoladamente.

- Para o conjunto de dados do experimento, a estimação da posição obtida pela fusão de dados por meio do FPNE obtém melhor resultado que a estimação da posição obtida pelo FP.

No segundo experimento, verifica-se a possibilidade de se realizar a estimação da posição do VANT pelo FPNE, durante sua navegação sobre região urbana em condições de baixa ou sem luminosidade. Nesta condições, o VANT utiliza imagens de

intensidade proveniente de uma câmera LiDAR. Neste experimento, realiza-se a fusão de dados do registro de imagens e da odometria visual considerando o conjunto de dados obtidos pelo sensor imageador LiDAR, que obtém imagens de intensidade do terreno. O conjunto de dados é o mesmo aplicado no segundo experimento da estimação de posição por registro automático de imagens de intensidade LiDAR (vide seção 6.6.3), e o mesmo aplicado no segundo experimento da estimação da posição por odometria visual monocular, que também utiliza imagens de intensidade LiDAR (vide 7.2.3).

O segundo experimento realiza a fusão de dados utilizando a resposta da odometria visual, obtida na seção 7.2.3, e a resposta obtida pela melhor configuração do algoritmo de registro automático de imagens de intensidade LiDAR. O melhor resultado do registro automático de imagens de intensidade LiDAR corresponde ao obtido pelo método configurado com o filtro da mediana e a MLP em LUT (vide 6.6.2.6).

O segundo experimento tem o objetivo de avaliar as seguintes hipóteses:

- Para o conjunto de dados de imagens de intensidade LiDAR, a estimação da posição do VANT obtida pela fusão de dados da odometria e registro de imagens pelo FPNE produz resultados satisfatórios, ou seja, uma estimação da posição com um erro inferior ao obtido pelo GPS do VANT. O valor do erro de 10 metros do GPS da aeronave RMAX é utilizada como parâmetro.
- Para as imagens de intensidade LiDAR, o FPNE, quando aplicado para estimação da posição do VANT, obtém melhores resultados que as técnicas de estimação por odometria e registro automático de imagens, quando consideradas isoladamente.
- Em relação ao conjunto de imagens de intensidade LiDAR, a estimação da posição obtida pela fusão de dados por meio do FPNE obtém melhores resultados que a estimação por fusão de dados pelo FP.

No terceiro experimento, avalia-se a possibilidade da estimação da posição do VANT por fusão de dados pelo FPNE quando este navega sobre uma região coberta por água, por exemplo, o oceano. Neste experimento, o algoritmo de fusão de dados utiliza o resultado da estimação da posição do VANT, obtido pelo registro automático de imagens e da odometria visual monocular, sobre o conjunto de dados formados por imagens do MDS da cobertura do fundo do oceano. Este conjunto de dados é o mesmo aplicado nos experimentos 6.6.4 e 7.2.4, que avaliaram a estimação de

posição do VANT por registro automático de imagens e odometria visual monocular, respectivamente. Ambos experimentos utilizam imagens do MDS do fundo do oceano obtidas por sensor LiDAR.

Para o terceiro experimento, o resultado utilizado do registro automático de imagens é obtido pelo método configurado com a MLP em LUT e o filtro da mediana, este resultado pode ser verificado na seção 6.6.4, e representa a melhor estimação de posição para as imagens do MDS do fundo do oceano. O resultado utilizado da odometria visual pode ser verificado na seção 7.2.4.

O terceiro experimento tem como objetivo avaliar as seguintes hipóteses:

- Para o conjunto de dados de imagens do MDS da cobertura do fundo do oceano, a estimação da posição do VANT obtida pela fusão de dados da odometria e registro de imagens pelo FPNE produz resultados satisfatórios, ou seja, uma estimação da posição com um erro inferior ao obtido pelo GPS do VANT. O valor do erro de 10 metros do GPS da aeronave RMAX é utilizada como parâmetro de avaliação.
- Considerando o conjunto de dados formado por imagens do MDS do fundo do oceano, obtido por sensor LiDAR, a fusão de dados por FPNE produz resultados melhores que a estimação de posição do VANT por registro automático de imagens ou odometria monocular visual, quando estas técnicas são consideradas isoladamente.
- Para as imagens do MDS do fundo do oceano obtido por sensor LiDAR, a fusão de dados por FPNE produz resultados melhores de estimação da posição do VANT que a fusão de dados por FP.

O método de estimação da posição por fusão de dados pelo FPNE foi implementado em linguagem C, com auxílio da biblioteca OpenCV (ITSEEZ, 2015), e aplicado para a produção dos resultados dos três experimentos. O método foi testado em dois computadores que podem ser embarcados em um VANT: RaspBerry PI Model B e Zybo ZYNQ 7000.

8.5.1 MÉTRICAS DE AVALIAÇÃO

As métricas EGM, SDMG e GM, aplicadas anteriormente para a avaliação das técnicas de estimação da posição do VANT por registro automático de imagens e odometria visual monocular, são aplicadas para avaliar os resultados de estimação

da posição obtido pela fusão dos dados (registro de imagens + odometria) pelo FPNE. Informações adicionais sobre as métricas de avaliação podem ser obtidas nas seções 6.6.1 e 7.2.1.

8.5.2 RESULTADOS: PRIMEIRO EXPERIMENTO

O primeiro experimento aplica o FPNE para realizar a fusão do dados de estimação da posição provenientes do registro automático de imagens e da odometria visual monocular. Para este experimento estas técnicas visão computacional para estimação da posição utilizam um conjunto de dados de imagens capturadas pela câmera de vídeo do VANT. Este conjunto de dados foi obtido a partir de um voo real de um VANT sobre uma região urbana, e fornece 1443 pontos para a avaliação do método de estimação. Cada um desses pontos de avaliação são imagens capturadas pela câmera de vídeo do VANT em tempo voo. Informações adicionais sobre as imagens do conjunto de dados podem ser obtidas nas seções 6.6.2 e 7.2.2.

O algoritmo de registro automático de imagens é configurado com o filtro da mediana, para a remoção de ruídos indesejáveis presentes nas imagens, e com a MLP em LUT para a detecção de bordas. Esta configuração foi a selecionada, pois foi a que obteve os melhores resultados para a estimação da posição por registro de imagens considerando o conjunto de dados obtidos pela câmera de vídeo, esses resultados podem ser verificados na seção 6.6.2.6. Este algoritmo realiza o registro automático da imagem capturada em tempo de voo pela câmera de vídeo do VANT, com uma imagem georreferenciada de satélite da mesma região sobrevoada, o resultado é a estimação da posição do VANT. Informações sobre a estimação da posição do VANT por registro automático de imagens podem ser obtidas no capítulo 6.

A odometria visual monocular estima a trajetória do VANT pela análise de duas imagens subsequentes capturadas pelo sensor imageador da aeronave. O resultado da odometria visual monocular aplicado na fusão de dados pelo FPNE foi apresentado na seção 7.2.2.

A Figura 8.5 apresenta os gráficos do erro da estimação da posição obtido por registro automático de imagens, em verde, odometria visual monocular, em vermelho, e fusão de dados pelo FPNE, em azul. O gráfico do erro é em função dos pontos de avaliação. A linha tracejada em vermelho representa um erro igual a 5 metros, pontos abaixo desta linha são classificados como EGM.

O valor $q = 2,57$ foi aplicado no FPNE, este valor foi selecionado pois foi o que

obteve o maior número de pontos classificados como EGM entre os 100 valores de q uniformemente distribuídos entre 0 e 3. Para este valor de q a função de verossimilhança aplicada é a apresentada na Equação 8.41.

Segundo (TULSYAN et al., 2016), a quantidade de partículas aplicadas pode influenciar o resultado final obtido pelo MSMC. O número de partículas utilizadas pelo FPNE é igual a 1000, pois foi a quantidade que obteve maior quantidade de pontos classificados como EGM.

Figura 8.5 - Gráfico do erro da estimação da posição do VANT obtido por fusão de dados pelo FPNE, odometria visual e registro automático de imagens.

Fonte: Produção do autor.

Na legenda da Figura 8.5 (canto superior direito), RAI é uma abreviação para registro automático de imagens.

A Tabela 8.6 apresenta os valores das métricas EGM, SDGM e GM para as três técnicas de estimação de posição.

Tabela 8.6 - Valores das métricas EGM, SDGM e GM para as três técnicas de estimação da Posição do VANT.

	EGM	SDGM	GM
RAI	926	1348	856
Odometria	667	1413	637
FPNE	1381	1413	1351

Os resultados apresentados na Tabela 8.6 demonstram que $95,7\%$ dos pontos de avaliação são classificados como EGM, quando utiliza-se a fusão de dados (registro de imagens + odometria visual) pelo FPNE para estimar a posição do VANT. As técnicas de estimação da posição por registro automático de imagens e a odometria visual, consideradas de forma isoladas, obtém $64,2\%$ e $46,2\%$ dos pontos classificados como EGM, respectivamente. Para a métrica SDGM, a estimação da posição pela fusão de dados por FPNE alcança 100% dos pontos de avaliação com esta classificação, para a mesma métrica, o resultados do registro de imagens e da odometria visual são de $95,4\%$ e 100%, respectivamente. Considerando a métrica GM, a fusão de dados por FPNE obtém $95,6\%$ dos pontos de avaliação estimados com esta classificação, enquanto o registro de imagens e a odometria visual obtém $60,6\%$ e $45,1\%$ dos pontos de avaliação classificados com GM, respectivamente.

A Figura 8.6 apresenta o gráfico do erro da fusão de dados pelo FPNE para dois valores de q, para $q = 2,57$, em azul, e para $q = 1$, em vermelho. Quando o valor de q é igual a 1 a função de verossimilhança aplicada é a Gaussiana, desta forma, o FPNE produz os mesmos resultados do FP. O gráfico do erro é em função dos pontos de avaliação. A linha tracejada em verde representa um erro igual a 5 metros, pontos abaixo desta linha são classificados como EGM.

Figura 8.6 - Gráfico do erro da estimação da posição do VANT obtido por fusão de dados pelo FPNE com $q = 2,57$ e com $q = 1$.

Fonte: Produção do autor.

A Tabela 8.7 exibe os valores das métrica EGM, SDGM e GM obtida pela fusão de dados por FPNE com $q = 1$ e $q = 2,57$

Tabela 8.7 - Valores das métricas EGM, SDGM e GM para o Filtro de Partículas Não-Extensivo com $q = 1$ e $q = 2,57$

	EGM	SDGM	GM
FPNE $q = 1$	1303	1412	1273
FPNE $q = 2,57$	1381	1413	1351

Os resultados exibidos na Tabela 8.7 demonstram que o FPNE, configurado com $q = 2,57$ obtém $95,7\%$ dos pontos de avaliação classificados como EGM, enquanto que o FPNE, configurado com $q = 1$ obtém $90,2\%$ dos pontos de avaliação classificados como EGM. Para a métrica SDGM, o FPNE com $q = 2,57$ alcança 100% dos pontos de avaliação contra $99,9\%$ do algoritmo FPNE com $q = 1$. Par a métrica GM, os resultados são $95,6\%$ e $90,1\%$ para o FPNE configurado com $q = 2,57$ e $q = 1$, respectivamente.

A Figura 8.7 apresenta o gráfico do desvio padrão do erro das 30 últimas estimações de posições obtidas pela fusão de dados pelo FPNE para $q = 1$, em vermelho, e

$q = 2,57$, em azul. O gráfico do desvio padrão do erro é em função dos pontos de avaliação. A linha tracejada verde representa um desvio padrão do erro de 2 metros, pontos abaixo dessa linha são classificados como SDGM.

Figura 8.7 - Gráfico do desvio padrão do erro da estimação da posição do VANT obtido por fusão de dados pelo FPNE configurado com $q = 2,57$ e com $q = 1$.

Fonte: Produção do autor.

A Figura 8.8 apresenta a trajetória estimada obtida pela fusão dos dados pelo algoritmo de FPNE configurado com $q = 2,57$. Esta trajetória estimada é exibida sobre a imagem georreferenciada da região sobrevoada pela aeronave. A trajetória estimada está em azul e a trajetória real do VANT em vermelho. O círculo preto marca o ponto de partida do VANT e o círculo branco o ponto de chegada.

Figura 8.8 - Trajetória estimada por fusão dos dados pelo FPNE configurado com $q = 2,57$, na cor azul, e a trajetória original do VANT, em vermelho

Fonte: Produção do autor

O algoritmo de FPNE, configurado com $q = 2,57$, com 1000 partículas e aplicado para a estimação da posição do VANT, realizou $1,8$ estimações por segundo, quando executados no Raspberry PI Model B-1, e $2,2$ estimações por segundo, quando executado no Zybo ZINQ 7000.

8.5.2.1 RESULTADOS: ANÁLISE DOS RESULTADOS DO PRIMEIRO EXPERIMENTO

Considerando o conjunto de dados obtido pela câmera de vídeo da aeronave RMAX, a análise dos resultados do primeiro experimento evidenciam que a fusão de dados por FPNE pode ser aplicada para obter a estimação da posição do VANT, pois dos 1443 pontos de avaliação em 1381 o método obteve um erro inferior a 5 metros, ou seja, $95,7\%$ dos pontos foram estimados com um erro que é a metade do erro do GPS presente na aeronave RMAX. Este resultado demonstra que a primeira hipótese é verdadeira.

Para o conjunto de dados do primeiro experimento, os resultados das métricas de avaliação, presentes na Tabela 8.6, demonstram que a fusão de dados produz melhores resultados que o registro automático de imagens e a odometria monocular visual, quando consideradas isoladamente. A métrica EGM apresenta que a fusão de dados estima uma maior quantidade de pontos com um erro inferior a 5 metros. Além disso, a métrica SDGM e o gráfico da Figura 8.5 demonstram que a fusão de dados é

mais estável que o registro automático de imagens. O gráfico da Figura 8.5 demonstra que o FPNE pode solucionar o problema do erro de deriva da odometria visual monocular, pois mesmo com o aumento do erro na estimação da odometria, o FPNE mantém o erro de estimação próximo ou abaixo de 5 metros. Essas considerações demonstram que a segunda hipótese para o primeiro experimento é verdadeira.

Os resultados das métricas de avaliação, presentes na Tabela 8.7, apresentam uma comparação entre a estimação obtida pelo FPNE, configurado com $q = 2,57$, e pelo FPNE, configurado com $q = 1$. Quando o algoritmo de FPNE é configurado com valor de $q = 1$ produz os mesmos resultados do algoritmo de FP. Pela análise desses resultados e considerando o conjunto de dados do primeiro experimento, verifica-se que a aplicação da função 8.41, distribuição de Tsallis com $2,57$, como função de verossimilhança é mais adequada que a utilização da função gaussiana, ou seja, o algoritmo de FPNE aproxima melhor a FDP alvo que o FP. Desta forma, a terceira hipotese para o primeiro experimento está correta.

8.5.3 RESULTADOS: SEGUNDO EXPERIMENTO

O segundo experimento aplica o FPNE para realizar a fusão dos dados de estimação da posição provenientes do registro automático de imagens e da odometria visual monocular, com a finalidade de simular a obtenção da posição do VANT quando este navega uma região urbana em condições de baixa ou sem luminosidade. Este segundo experimento realiza a fusão de dados do resultado do registro automático de imagens, apresentado na seção 6.6.3, com o resultado da odometria visual monocular, apresentado na seção 7.2.3. Neste experimento, para a configuração do algoritmo de FPNE, foi utilizado o valor de $q = 2,57$, este valor é aplicado pois no primeiro experimento foi o que obteve os melhores resultados e, além disso, os dados do primeiro experimento são de um voo real de um VANT.

O conjunto de dados, utilizado para gerar os resultados das técnicas de estimação de posição por visão computacional, é proveniente de uma imagem de intensidade obtida por um sensor LiDAR. Da imagem de intensidade LiDAR são obtidas 172 subimagens, que simulam a captura de imagens por uma câmera LiDAR, fixada à aeronave, durante o voo sobre uma região urbana. Cada uma dessas subimagens representam pontos de avaliação, desta forma, são 172 pontos de avaliação do algoritmo. Informações adicionais sobre as imagens do conjunto de dados podem ser obtidas nas seções 6.6.3 e 7.2.3.

O algoritmo de registro automático de imagens é configurado com o filtro da Me-

diana, para a remoção de ruídos indesejáveis presentes nas imagens, e com a MLP em LUT para a detecção de bordas. Esta configuração foi selecionada pois obteve os melhores resultados, considerando o conjunto de dados de imagens de intensidade LiDAR. Estes resultados foram apresentados na seção 6.6.3.2. O algoritmo simula a obtenção da posição de uma VANT por meio do registro automático entre uma imagem de intensidade LiDAR, capturada em tempo de voo pela aeronave, e uma imagem georreferenciada de intensidade LiDAR da região sobrevoada.

A odometria visual monocular estima a trajetória do VANT pela análise de duas imagens subsequentes e capturadas pelo sensor imageador da aeronave. Os resultados da odometria visual monocular, utilizando imagens de intensidade LiDAR, aplicados na fusão de dados deste experimento foram apresentados na seção 7.2.3.

A Figura 8.9 apresenta os gráficos do erro da estimação da posição por registro automático de imagens, em verde, da odometria visual monocular, em vermelho, e da fusão de dados pelo FPNE, em azul. O gráfico do erro de estimação é em função de cada ponto de avaliação. A linha tracejada em vermelho representa um erro igual a 5 metros, pontos abaixo desta linha são classificados como EGM. O algoritmo de FPNE foi configurado com 1000 partículas e com o valor de $q = 2.57$, a mesma configuração do primeiro experimento.

Figura 8.9 - Gráfico do erro da estimação da posição do VANT obtido por fusão de dados pelo FPNE, odometria visual e registro automático de imagens.

Fonte: Produção do autor.

A Tabela 8.8 apresenta os valores das métricas EGM, SDGM e GM para as três técnicas de estimação de posição.

Tabela 8.8 - Valores das métricas EGM, SDGM e GM para as três técnicas de estimação da Posição do VANT.

	EGM	SDGM	GM
RAI	168	112	109
Odometria	172	142	142
FPNE	172	142	142

Os resultados apresentados na Tabela 8.8 demonstram que 100% dos pontos de avaliação são classificados como EGM, quando realiza-se a estimação da posição por fusão de dados pelo FPNE, este é o mesmo resultado obtido pela estimação por odometria visual monocular. A técnica de estimação da posição por registro automático de imagens alcança $97,7\%$ do pontos de avaliação classificados como EGM. Para a métrica SDGM, a fusão de dados pelo FPNE e a odometria visual obtém 100% dos pontos assim classificados, enquanto que o registro automático de imagens atinge um percentual de $78,9\%$. Com relação à métrica GM, o registro automático de imagens classifica $78,9\%$ dos pontos de avaliação, enquanto que a odometria visual e a fusão de dados classificam 100% dos pontos de avaliação como GM.

A Figura 8.10 apresenta o gráfico do erro da fusão de dados pelo FPNE para dois valores de q, para $q = 2,57$, em azul, e para $q = 1$, em vermelho. O gráfico do erro é em função de cada ponto de avaliação. A linha tracejada em verde representa um erro igual a 5 metros, pontos abaixo desta linha são classificados como EGM.

Figura 8.10 - Gráfico do erro da estimação da posição do VANT obtido por fusão de dados pelo Filtro de Partículas Não-Extensivo com $q = 2,57$ e com $q = 1$.

Fonte: Produção do autor.

A Tabela 8.9 exibe os valores das métrica EGM, SDGM e GM obtida pela fusão de dados por FPNE configurado com $q = 1$ e $q = 2,57$

Tabela 8.9 - Valores das métricas EGM, SDGM e GM para o FPNE com $q = 1$ e $q = 2,75$

	EGM	SDGM	GM
FPNE $q = 1$	172	142	142
FPNE $q = 2,57$	172	142	142

Para as três métricas de avaliação dos resultados, ambas as configurações do algoritmo de FPNE obtiveram 100% dos pontos de avaliação classificados como EGM, SDGM e GM

A Figura 8.11 exibe o resultado do desvio padrão do erro das 30 últimas estimações de posições obtidas pela fusão de dados pelo FPNE configurado com $q = 1$, em vermelho, e $q = 2,57$, em azul. O gráfico do desvio padrão do erro é função dos pontos de avaliação. A linha tracejada verde representa um desvio padrão de 2 metros, pontos abaixo dessa linha são classificados como SDGM.

Figura 8.11 - Gráfico do desvio padrão do erro da estimação da posição do VANT obtido por fusão de dados pelo FPNE configurado com $q = 2,57$ e com $q = 1$.

Fonte: Produção do autor.

A Figura 8.12 exibe a trajetória estimada, resultado da fusão dos dados pelo FPNE com $q = 2,57$ na imagem georreferenciada de intensidade LiDAR. A trajetória estimada está em azul e a trajetória real do VANT em vermelho. O círculo amarelo marca o ponto de partida do VANT e o círculo branco o ponto de chegada.

Figura 8.12 - Trajetória estimada por fusão dos dados por FPNE na cor azul e a trajetória original do VANT em vermelho

Fonte: Produção do autor.

8.5.3.1 RESULTADOS: ANÁLISE DOS RESULTADOS DO SEGUNDO EXPERIMENTO

Para o conjunto de dados aplicado, a análise dos resultados das métricas de avaliação no segundo experimento, apresentados na Tabela 8.8, evidenciam que a o algoritmo de fusão de dados pelo FPNE pode ser aplicado para estimar a posição do VANT, pois todos os 172 pontos de avaliação foram classificados como EGM, ou seja, com um erro inferior a 5 metros. Este resultado demonstra que a primeira hipótese para o segundo experimento está correta.

Considerando os dados provenientes de imagem de intensidade LiDAR, os resultados das métricas EGM, SDGM e GM (exibidos na Tabela 8.8) demostram que a fusão de dados pelo FPNE obtém resultados melhores que as técnicas de registro automático de imagens e odometria visual monocular. Apesar dos resultados da odometria e do FPNE serem iguais para as três métricas de avaliação, o gráfico da Figura 8.9 demonstra que o erro de estimação obtido pelo FPNE é menor na maioria dos 172 pontos de avaliação. O erro médio da estimação da posição do VANT obtido pelo FPNE é igual $0,6743$ metros, enquanto o erro médio da estimação obtido pela odometria visual monocular é igual a $2,4713$ metros. Os resultados destacados neste parágrafo demonstram que a segunda hipótese para o segundo experimento está correta.

Tendo em conta o conjunto de dados do segundo experimento, os resultados das métricas EGM, SDGM e GM exibidos na Tabela 8.9 apresentam que o FPNE com $q = 1$ (para $q = 1$, o algoritmo do FPNE é o mesmo do FP) e $q = 2,57$ obtém os mesmos resultados. Mas a análise do gráfico 8.10, demonstra que o erro de estimação obtido pelo FPNE com $q = 2,57$ é menor para a maioria dos pontos de avaliação. O erro médio do algoritmo FPNE com $q = 2,57$ é $0,6743$ metros, enquanto que para o algoritmo configurado com o valor de $q = 1$ é $1,6070$ metros. Além disso, o gráfico da Figura 8.11 exibe que o algoritmo de FPNE com $q = 2,57$ é mais estável que o algoritmo de FPNE com $q = 1$. Por esta análise, pode-se afirmar que a a aplicação função 8.41, distribuição de Tsallis com $q = 2,57$, como função de verossimilhança apresenta melhores resultados que a utilização da função gaussiana, ou seja, o algoritmo de FPNE aproxima melhor a FDP alvo que o algoritmo de FP, o que demonstra que a terceira hipótese do segundo experimento é verdadeira.

8.5.4 RESULTADOS: TERCEIRO EXPERIMENTO

o terceiro experimento realiza a fusão dos dados do registro automático de imagens e da odometria visual monocular para estimar a posição do VANT, quando este sobrevoa regiões cobertas por água. Para cumprir este objetivo, as técnicas de estimação de posição por visão computacional utilizam imagens do MDS da cobertura do fundo do oceano, que foram capturadas por um sensor LiDAR.

Neste terceiro experimento, a estimação da posição pelo FPNE é uma simulação da navegação do VANT sobre regiões cobertas por água. As imagens capturadas pela aeronave em tempo de voo são subimagens de um MDS georreferenciado, estas subimagens são utilizadas pelo registro automático de imagens e pela odometria visual monocular para estimar a posição do VANT. As subimagens simulam a captura de imagens por uma câmera LiDAR. Nesta simulação, foram geradas 92 subimagens do MDS georreferenciado do fundo do oceano, cada uma dessas subimagens representam um ponto de avaliação do algoritmo de estimação da posição do VANT. Estas imagens são as mesmas aplicadas na estimação da posição do VANT por registro automático de imagens (seção 6.6.4) e por odometria visual monocular (seção 7.2.4). Informações adicionais sobre as imagens que formam o conjunto de dados podem ser obtidas nas seções 6.6.4 e 7.2.4.

O resultado da estimação da posição por registro automático de imagens, utilizado como parâmetro de entrada do algoritmo de FPNE neste experimento, foi obtido pelo método configurado com o filtro da Mediana e a MLP em LUT. Este resultado foi apresentado na seção 6.6.4.2.

O resultado da odometria visual monocular, utilizado como parâmetro de entrada do algoritmo de fusão de dados pelo FPNE, foi apresentado na seção 7.2.4.

Para este experimento, o algoritmo de fusão de dados pelo FPNE foi configurado com $q = 2.57$ e com 1000 partículas, a mesma configuração que obteve os melhores resultados no primeiro experimento, que utiliza dados reais de um voo de um VANT.

A Figura 8.13 apresenta o gráfico do erro da estimação da posição do VANT, de cada um dos 92 pontos de avaliação, para as três técnicas apresentadas. O gráfico do erro de estimação é em função do ponto de avaliação. Em azul, o gráfico do erro da estimação pelo algoritmo de FPNE. Em vermelho, o gráfico do erro da estimação por pela odometria visual. Em verde, o gráfico do erro da estimação obtido pelo registro automático de imagens.

Figura 8.13 - Gráfico do erro da estimação da posição do VANT obtido por fusão de dados pelo FPNE, odometria visual e registro automático de imagens.

Fonte: Produção do autor.

A Tabela 8.10 apresenta os valores das métricas EGM, SDGM e GM para as três técnicas de estimação de posição.

Tabela 8.10 - Valores das métricas EGM, SDGM e GM para as três técnicas de estimação da Posição do VANT.

	EGM	SDGM	GM
RAI	92	62	62
Odometria	92	62	62
FPNE	92	62	62

Os resultados apresentados na Tabela 8.10 demostram que as três técnicas de estimação de posição do VANT obtiveram 100% dos pontos de avaliação classificados nas métricas EGM, SDGM e GM.

A Figura 8.14 apresenta o gráfico do erro da estimação da posição do VANT obtido pelo FPNE para dois valores de q, para $q = 2,57$, em azul, e para $q = 1$, em vermelho. O gráfico do erro é em função de cada ponto de avaliação. A linha tracejada em verde representa um erro igual a 5 metros, pontos abaixo desta linha são classificados como EGM.

Figura 8.14 - Gráfico do erro da estimação da posição do VANT obtido por fusão de dados pelo FPNE com $q = 2.57$ e com $q = 1$.

Fonte: Produção do autor.

A Tabela 8.11 exibe os valores das métrica EGM, SDGM e GM obtida pela fusão de dados por FPNE com $q = 1$ e $q = 2,57$.

Tabela 8.11 - Valores das métricas EGM, SDGM e GM para o FPNE com $q = 1$ e $q = 2,57$

	EGM	SDGM	GM
FPNE $q = 1$	92	62	62
FPNE $q = 2,57$	92	62	62

Os resultados apresentados na Tabela 8.11 demostram que o algoritmo de fusão de dados pelo FPNE obteve 100% dos pontos de avaliação classificados nas métricas EGM, SDGM e GM, para os dois valores do parâmetro q.

A Figura 8.15 exibe o resultado do desvio padrão do erro das 30 últimas estimações de posições obtidas pela fusão de dados pelo FPNE para $q = 1$, em vermelho, e $q = 2.57$, em azul. O gráfico do desvio padrão do erro é função dos pontos de avaliação. A linha tracejada verde representa um desvio padrão de 2 metros, pontos abaixo dessa linha são classificados como SDGM.

Figura 8.15 - Gráfico do desvio padrão do erro da estimação da posição do VANT obtido por fusão de dados pelo FPNE com $q = 2,57$ e com $q = 1$.

Fonte: Produção do autor.

A Figura 8.16 exibe a trajetória estimada, resultado da fusão dos dados pelo FPNE configurado com valor de $q = 2,57$, sobre a imagem do MDS georreferenciado do fundo do oceano. A trajetória estimada está em azul e a trajetória real do VANT em vermelho. O círculo amarelo marca o ponto de partida do VANT e o círculo branco o ponto de chegada.

Figura 8.16 - Trajetória estimada por fusão dos dados por FPNE na cor azul, e a trajetória original do VANT em vermelho

Fonte: Produção do autor.

8.5.4.1 RESULTADOS: ANÁLISE DOS RESULTADOS DO TERCEIRO EXPERIMENTO

Considerando os dados utilizados no terceiro experimento, a análise dos resultados, apresentados na Tabela 8.10, evidenciam que a técnica de fusão de dado pelo FPNE pode ser aplicada para realizar a estimação da posição do VANT quando este navega uma região coberta por água, pois todos os 92 pontos de avaliação da trajetória foram classificados como EGM. Este resultado demonstra que a primeira hipótese para o terceiro experimento está correta.

Para o conjunto de dados do MDS do fundo do oceano, os resultados das métricas EGM, SDGM e GM (exibidos na Tabela 8.10) indicam que a fusão de dados pelo FPNE obtém resultados iguais as técnicas de registro automático de imagens e odometria visual monocular. Apesar dos resultados obtidos pelo registro de imagens, odometria e fusão de dados pelo FPNE serem iguais para as três métricas de avaliação, o gráfico da Figura 8.9 demonstra que o erro de estimação obtido pelo FPNE é menor na maioria dos 92 pontos de avaliação. O erro médio da estimação da estimação da posição do VANT obtido pelo FPNE é igual $0,9857$ metros, o erro médio obtido pela odometria visual monocular é igual a $2,7572$ metros e o erro médio do registro automático de imagens é igual a $1,8315$ metros. Estes resultados demonstram que a segunda hipótese do terceiro experimento está correta.

Tendo em conta o conjunto de dados do terceiro experimento, os resultados das métricas EGM, SDGM e GM exibidos na Tabela 8.11 apresentam que o FPNE com $q = 1$ (para $q = 1$, o algoritmo do FPNE é o mesmo do FP) e $q = 2,57$ obtém os mesmos resultados. Entretanto, por meio da análise do gráfico 8.14, observa-se que o erro de estimação obtido pelo FPNE com $q = 2,57$ é menor que o erro obtido pelo FPNE com $q = 1$ na maior parte dos pontos de avaliação. O erro médio do algoritmo FPNE com $q = 2,57$ é 0.9857 metros, enquanto que, para o algoritmo com valor de $q = 1$ é 1.6885 metros. Além disso, o gráfico da Figura 8.15 ilustra que o algoritmo de FPNE com $q = 2,57$ é mais estável que o algoritmo de FPNE com $q = 1$. Por meio desta análise dos resultados, pode-se afirmar que a a aplicação da função 8.41, distribuição de Tsallis com $q = 2,57$, como função de verossimilhança apresenta melhores resultados que a utilização da função gaussiana, ou seja, o FPNE aproxima melhor a FDP alvo que o FP, o que demonstra que a terceira hipótese do terceiro experimento está correta.

9 CONCLUSÕES

Neste trabalho, foi apresentada uma abordagem para realizar a estimação da posição do VANT, em caso de falha ou ausência do sinal do GNSS, para diferentes situações de navegação: situação

Situação 1: a aeronave sobrevoa uma região urbana em condições de iluminação do ambiente suficientes para que um sensor do VANT, que captura imagens na banda do visível, por exemplo, uma câmera de vídeo, consiga capturar imagens da região sobrevoada;

Situação 2: a aeronave sobrevoa uma região urbana em condições de pouca ou sem iluminação do ambiente, nesta situação, um sensor LiDAR (câmera LiDAR) que captura imagens de intensidade da região sobrevoada é utilizado como sensor imageador;

Situação 3: o VANT sobrevoa uma região coberta por água, por exemplo, o oceano, nesta situação, um sensor LiDAR (câmera LiDAR) que gera imagens do MDS da cobertura do fundo da região é utilizado como sensor imageador.

O uso de diferentes medidas pode aprimorar os resultados da estimação da posição do VANT. Portanto, a abordagem desenvolvida é baseada na fusão de dados da estimação da posição por registro automático de imagens e da estimação da posição por odometria visual monocular. A fusão de dados é realizada por meio do algoritmo de Filtro de Partículas Não-Extensivo. Nesta pesquisa, cada um dos métodos de estimação da posição VANT, que compõem a abordagem, são avaliados por diferentes experimento que abrangem as três situações de voo citadas anteriormente. E para cada um dos métodos que compõem a abordagem de estimação da posição, são apresentados os resultados obtidos.

Para fins de avaliação de plataformas computacionais que podem ser embarcadas em um VANT, o método proposto de estimação da posição do VANT foi testado em duas arquiteturas de hardware diferentes: o Raspberry Pi Model B-1 e o Zybo ZINQ 7000. Ambos os sistemas, devido ao seu peso e consumo energético, podem ser embarcados em VANT.

O método de registro automático de imagens realiza a estimação da posição do VANT por meio do cálculo do índice de correlação entre uma imagem da região sobrevoada, capturada em tempo de voo pelo sensor embarcado da aeronave, e

uma imagem georreferenciada da mesma região sobrevoada. A extração de bordas de imagens de diferentes sensores permite o cálculo do índice de correlação sem afetar o resultado sob diferentes condições de iluminação. Três operadores para a detecção de bordas foram aplicados, sendo eles: Canny, Sobel e MLP. Na MLP, sua arquitetura foi determinada automaticamente pelo MPCA e esta foi implementada em FPGA, utilizando uma estratégia em LUT (MLP em LUT). Para evitar que ruídos indesejáveis presentes nas imagens afetassem o resultado da estimação da posição, dois filtros foram empregados: o filtro da Mediana e o filtro Gaussiano.

O método de registro automático de imagens foi testado em diferentes conjunto de dados de imagens, que abrangem as três situações de voo. Para todos os testes, o método apresentou resultados satisfatórios, mas o método configurado com a MLP em LUT e o filtro da Mediana apresentou sistematicamente, os melhores resultados na estimação da posição do VANT. É pertinente ressaltar que, além de apresentar melhores estimativas do posicionamento do VANT, a configuração do algoritmo de registro com a MLP em LUT apresentou o melhor desempenho computacional quando implementado no computador Zybo ZINQ 7000. Devido ao computador Zybo ZINQ 7000 ter uma arquitetura SOC para CPU e FPGA, ou seja, processador e FPGA no mesmo *chip*, permite que a transferência de dados entre CPU e FPGA seja acelerada. Esta característica é o principal fator para o ganho de desempenho computacional.

O posicionamento do VANT por odometria visual monocular estima a trajetória da aeronave por meio da análise de duas imagens subsequentes capturadas pelo sensor imageador da mesma. A principal etapa do algoritmo de odometria visual monocular é a detecção e descrição dos pontos de interesse. Há na literatura diversos algoritmos para esta finalidade, mas durante a revisão bibliográfica observou-se que o algoritmo mais apropriado para ser utilizado em aplicações para estimar a posição de robôs móveis é o SURF. O método de odometria visual monocular foi testado nos mesmos conjuntos de dados de imagens aplicados nos experimentos que envolveram o método de registro automático de imagens. Para todos os testes, o algoritmo de odometria visual monocular obteve resultados satisfatórios para a estimação da posição do VANT. O método de odometria visual monocular demonstrou ser mais estável na estimação da posição do VANT que o método de registro automático de imagens, mas apresentou um erro de deriva, ou seja, o erro da posição estimada aumenta conforme a aeronave se distância do ponto de partida. O método de odometria foi também testado tanto no Raspberry Pi quanto no Zybo ZINQ 7000, obtendo melhor desempenho computacional neste último computador.

A estimação da posição por fusão de dados utiliza como parâmetro de entrada as estimativas fornecidas pelo registro automático de imagens e pela odometria visual monocular para produzir uma nova estimativa de posição. Para realizar a fusão de dados, foi aplicado o algoritmo de Filtro de Partículas Não-Extensivo, que é uma nova versão do algoritmo tradicional de Filtro de Partículas. Nesta nova formulação do Filtro de Partículas aplica-se a distribuição de Tsalli (TSALLIS, 1999) como função de verossimilhança. Ao aplicar a distribuição de Tsallis como função de verossimilhança, o algoritmo de Filtro de Partículas Não-Extensivo, permite que se encontre uma função distribuição de probabilidade que se adapte melhor ao problema.

Três experimentos que abrangem as três situações de voo foram realizados. Os conjuntos de dados de imagens aplicados nos experimentos de registro de imagens e de odometria visual também foram utilizados nos experimentos da fusão de dados. Para todos experimentos, os resultados obtidos pela fusão de dados foram melhores quando comparados aos resultados obtidos pelas técnicas de estimação por visão computacional de forma isolada. A fusão de dados produziu resultados com maior estabilidade, quando comparados com os resultados do registro automático de imagens, e sem a presença do erro de deriva, quando comparados com os resultados da odometria visual monocular. Além disso, os experimentos foram realizados tanto com a fusão de dados por Filtro de Partículas quanto por Filtro de Partículas Não-Extensivo e, sistematicamente, o segundo filtro obteve melhores resultados em todos experimentos realizados. Os resultados obtidos de estimação de posição pela fusão de dados evidenciam o potencial da aplicação da abordagem para a realização da estimação da posição de VANT nas três situações de voo em caso de falha do sinal do GNSS. O algoritmo foi implementado nas duas plataformas computacionais, Raspberry Pi e Zybo ZYNQ 7000, e o algoritmo de fusão de dados executado no segundo computador obteve melhor resultado.

As principais contribuições apresentadas nesta tese são:

- A aplicação do MPCA para determinar a arquitetura de uma MLP aplicada para a detecção de bordas;
- A proposta e o desenvolvimento de um detector baseado em MLP em LUT que, no contexto do registro automático de imagens, produz resultados melhores e com melhor desempenho quando comparados com os operadores Canny e Sobel;
- A utilização de imagens de MDS provenientes de um sensor LiDAR, o que

permite a estimação da posição do VANT quando este sobrevoa uma região coberta por água, por exemplo, o oceano, por método de fusão de dados de técnicas de visão computacional;

- A utilização de imagens de intensidade provenientes de um sensor LiDAR, o que permite a estimação da posição do VANT quando este sobrevoa uma região urbana com pouca ou sem luminosidade no ambiente, por método de fusão de dados de técnicas de visão computacional;

- A aplicação do Filtro de Partículas Não-Extensivo no processo de fusão de dados oriundos do registro de imagens e da odometria visual.

Como continuidade das pesquisas realizadas até o presente momento, algumas propostas de trabalhos são:

- Na estimação da posição por registro automático de imagens, a implementação do algoritmo de correlação cruzada com técnicas de programação de alto desempenho para melhorar o tempo de processamento do método de estimação de posição. Além disso, para melhorar a estimação da posição fornecida pelo algoritmo, realizar uma pré-classificação das imagens, por exemplo, com o uso da rede Mapa Auto-Organizáveis de Kohonen (MOTA et al., 2014);

- Na estimação de posição por odometria visual monocular, realizar teste com outros algoritmo de descrição e detecção de pontos de interesse, como o CENSURE, que vem apresentando melhores resultados que o SURF;

- Na fusão de dados por FPNE, aplicar um segundo FP para estimar o valor de q, como realizado na pesquisa desenvolvida por CAMPOS VELHO e Furtado (2011). Com esta modificação é possível estimar o melhor valor de q em tempo de execução, e assim o algoritmo de FPNE é denominado Filtro de Partículas Adaptativo (FPA);

- O estudo de técnicas de programação de alto desempenho para a implementação do algoritmo de estimação por odometria visual monocular, com o objetivo de melhorar o desempenho do algoritmo;

- O estudo de técnicas de programação de alto desempenho para a implementação do algoritmo de fusão de dados pelo Filtro de Partículas Não-Extensivo, com o objetivo de melhorar o desempenho do algoritmo.;

- Realizar mais teste do método proposto com outros dados de voo de VANT, para permitir uma melhor validação das hipóteses levantadas durante os experimentos;
- Realizar testes com outras meta-heurísticas para determinar da arquitetura da MLP;
- Embarcar o método proposto e integrar com a malha de controle do piloto automático.

REFERÊNCIAS BIBLIOGRÁFICAS

ABDU, M. A.; BATISTA, I. S.; CARRASCO, A. J.; BRUM, C. G. M. South atlantic magnetic anomaly ionization: a review and a new focus on electrodynamic effects in the equatorial ionosphere. **Journal of Atmospheric and Solar-Terrestrial Physics**, v. 67, n. 17, p. 1643–1657, 2005. Disponível em: <http://www.sciencedirect.com/science/article/pii/S1364682605001240>. Acesso em: 27 out. 2017. 3, 7

ACHTELIK, M.; WEISS, S.; SIEGWART, R. Onboard imu and monocular vision based control for mavs in unknown in-and-outdoor environments. In: IEEE INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION, 2013, Karlsruhe, Alemanha. **Proceedings...** IEEE, 2013. p. 3056–3063. Disponível em: <http://ieeexplore.ieee.org/document/5980343/>. Acesso em: 5 jun. 2017. 15

AHMADINIA, M. **Integrating a neuron network application into a ZYBO Zynq-7000 development board with an AXI-Bus interface**. 73 p. Dissertação (Mestrado em Engenharia de Computação) — Universidade Técnica de Delft (TUDelft), Delft, Países Baixos, 2016. Disponível em: <https://repository.tudelft.nl/islandora/object/44403>. Acesso em: 21 mar. 2018. 36

AMIDI, O. **An autonomous vision-guided helicopter**. 152 p. Tese (Doutorado em Engenharia Elétrica e Computação) — Carnegie Mellon University, Pittsburgh, EUA, 1996. Disponível em: <https://www.ri.cmu.edu/pub_files/pub3/amidi_omead_1996_1/amidi_omead_1996_1.pdf>. Acesso em: 04 jul. 2017. 17

ANAC. **Regulamento Brasileiro da Aviação Civil Especial RBACE - 94**. Brasília, 2017. Norma regulamentadora. Disponível em: <http://www.anac.gov.br/participacao-social/audiencias-e-consultas-publicas/audiencias/2015/aud13/anexoi.pdf>. Acesso em: 22 mar. 2017. 13, 20

ANOCHI, J. A. **Previsão climática de precipitação por redes neurais autoconfiguradas**. 161 p. Tese (Doutorado em Computação Aplicada) — Instituto Nacional de Pesquisas Espaciais (INPE), São José dos Campos, 2015. 78, 79, 80, 81

ANTONARAKIS, A. S.; RICHARDS, K. S.; BRASINGTON, J. Object-based land cover classification using cirborne lidar. **Remote Sensing of Environment**, v. 112, p. 2988–2998, 2008. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0034425708000667>. Acesso em: 21 fev. 2018. 47

ARULAMPALAM, M. S.; MASKELL, S.; GORDON, N.; CLAPP, T. A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. **IEEE Transactions on Signal Processing**, v. 50, p. 174–188, 2002. Disponível em: <http://ieeexplore.ieee.org/document/978374/>. Acesso em: 16 maio 2017. 200, 202

ASHRAF, I.; HUR, S.; PARK, Y. An investigation of interpolation techniques to generate 2d intensity image from lidar data. **IEEE Access**, v. 5, p. 8250–8260, 2017. Disponível em: <http://ieeexplore.ieee.org/document/7915679/>. Acesso em: 01 fev. 2018. 51, 120, 181

BACHRACH, A.; WINTER, A.; HE, R.; HEMANN, G.; PRENTICE, S.; ROY, N. Range - robust autonomous navigation in gps-denied environments. In: IEEE INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION, 2010, Achorage, EUA. **Proceedings...** IEEE, 2010. p. 1096–1097. Disponível em: <http://ieeexplore.ieee.org/document/5980343/>. Acesso em: 19 jul. 2017. 15, 19

BALAMURUGAN, G.; VALARMATHI, J.; NAIDU, V. P. S. Survey on uav navigation in gps denied environments. In: INTERNATIONAL CONFERENCE ON SIGNAL PROCESSING, COMMUNICATION, POWER AND EMBEDDED SYSTEM (SCOPES), 2016, Paralakhemundi, India. **Proceedings...** IEEE, 2016. p. 198–204. Disponível em: <http://ieeexplore.ieee.org/document/7955787>. Acesso em: 04 jul. 2017. 14, 15, 16

BAUER, J.; SUNDERHAUF, N.; PROTZEL, P. Comparing several implementations of two recently published feature detectors. **IFAC Proceedings Volumes of 6th IFAC Symposium on Intelligent Autonomous Vehicles**, v. 40, n. 15, p. 143–148, 2007. Acesso em: 12 nov. 2017. 154, 155, 157, 159

BAY, H.; ESS, A.; TUYTELAARS, T.; GOOL, L. V. Speeded-up robust features (surf). **Computer Vision and Image Understanding**, v. 110, n. 3, p. 346–359, 2008. Disponível em: <http:

//www.sciencedirect.com/science/article/pii/S1077314207001555>. Acesso em: 10 nov. 2017. 156, 157, 159, 160

BENSEDDIK, H. E.; DJEKOUNE, Q.; BELHOCINE, M. Sift and surf performance evaluation for mobile robot-monocular visual odometry. **Journal of Image and Graphics**, v. 2, n. 1, p. 70–76, 2014. Disponível em: <http://www.joig.org/index.phpm=contentc=indexa=showcatid=35id=71>. Acesso em: 11 nov. 2017. 154, 155, 157, 159

BESL, P. J.; MCKAY, N. D. Method for registration of 3-d shapes. **Proceedings of SPIE**, v. 1611, p. 1611 – 1632, 1992. Disponível em: <https://doi.org/10.1117/12.57955>. Acesso em: 21 maio 2017. 181, 182, 187, 265, 267, 268

BOLDRINI, J. L. **Algebra linear**. 3. ed. São Paulo, Brasil: Harbra, 1986. 411 p. 259

BRAGA, A. P.; CARVALHO, A.; LUDERMIR, T. **Redes neurais artificiais: teoria e aplicações**. 2. ed. Rio de Janeiro, RJ: LTC, 2007. 238 p. 73, 74, 75, 76

BRAGA, J. R. G.; CAMPOS VELHO, H. F.; CONTE, G.; DOHERTY, P.; SHIGUEMORI, E. H. An image matching system for autonomous uav navigation based on neural network. In: IEEE INTERNATIONAL CONFERENCE ON CONTROL, AUTOMATION, ROBOTICS AND VISION (ICARCV), 14., 2016, Phuket, Tailândia. **Proceedings...** IEEE, 2016. p. 1–6. Disponível em: <http://ieeexplore.ieee.org/document/7838775/>. Acesso em: 14 maio 2017. 5, 16, 18, 24, 26, 29, 30, 58, 59, 60, 72, 82, 84, 88, 91, 94

BROWN, L. G. A survey of image registration techniques. **ACM Computing Surveys (CSUR)**, v. 24, n. 4, p. 325–376, 1992. 55, 56

BYUN, Y.; SONG, J.; HAN, D. Real-time uav trajectory generation using feature points matching between video image sequences. **Journal of Physics: Conference Series**, v. 890, n. 1, p. 012074, 2017. Disponível em: <http://stacks.iop.org/1742-6596/890/i=1/a=012074>. Acesso em: 25 out. 2017. 18

CAMPOS VELHO, H. F.; FURTADO, H. C. M. Adaptive particle filter for stable distribution. **Integral Methods in Science and Engineering: Computational and Analytic Aspects**, v. 1, p. 47–57, 2011. Disponível em: <https://link.springer.com/chapter/10.1007/978-0-8176-8238-5_6>. Acesso em: 15 maio 2017. 6, 30, 204, 206, 236

CARVALHO, A. R.; RAMOS, F. M.; CHAVES, A. A. Metaheuristics for the feedforward artificial neural network (ann) architecture optimization problem. **Neural Computing and Applications**, v. 20, n. 8, p. 1273–1284, 2011. Disponível em: <http://dx.doi.org/10.1007/s00521-010-0504-3>. Acesso em: 06 fev. 2018. 80, 81

CASTANEDO, F. A review of data fusion techniques. **The Scientific World Journal**, p. 1–20, 2013. Disponível em: <https://www.hindawi.com/journals/tswj/2013/704504/>. Acesso em: 21 dez. 2017. 192

CASTRO, A. P. A.; SILVA, J. D. S.; MEDEIROS, F. L. L.; SHIGUEMORI, E. H. Restauração de imagens e detecção automática de características aplicados à navegação aérea autônoma. In: SIMPÓSIO BRASILEIRO DE SENSORIAMENTO REMOTO, 14., 2009, Natal, Brasil. **Anais...** São José dos Campos: INPE, 2009. p. 6497. Disponível em: <http://marte.sid.inpe.br/col/dpi.inpe.br/sbsr>. Acesso em: 02 ago. 2017. 4

CHANG-SUN, Y.; IEE-KI, A. Low cost gps/ins sensor fusion system for uav navigation. In: DIGITAL AVIONICS SYSTEMS CONFERENCE, 22., 2003, Indianapolis, EUA. **Proceedings...** IEEE, 2003. p. 1–9. Disponível em: <http://ieeexplore.ieee.org/document/5731136/>. Acesso em: 22 dez. 2017. 191, 192

CHAVES, A. N. **Proposta de modelo de Veículos Aéreos Não Tripulados (VANTs) cooperativos aplicados à operação de busca**. 149 p. Dissertação (Mestrado em Ciência) — Universidade de São Paulo (USP), São Paulo, 2013. Disponível em: <http://www.teses.usp.br/teses/disponiveis/3/3141/tde-26072013-115944/publico/Dissertacao_AquilaChaves_unprotected.pdf>. Acesso em: 14 maio 2017. 7

CHEN, Z.; GAO, B.; DEVEREUX, B. State-of-the-art: Dtm generation using airborne lidar data. **Sensors**, v. 17, n. 1, p. 1–24, 2017. Disponível em: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5298723/>. Acesso em: 21 fev. 2018. 138

CHEVIRON, T.; HAMEL, T.; MAHONY, R.; BALDWIN, G. Robust nonlinear fusion of inertial and visual data for position, velocity and attitude estimation of uav. In: IEEE INTERNATIONAL CONFERENCE ON ROBOTICS AND

AUTOMATION, 2007, Roma, Itália. **Proceedings...** IEEE, 2007. p. 2010–2016. Disponível em: <http://ieeexplore.ieee.org/document/6491082/>. Acesso em: 24 jul. 2017. 15, 19

CHUNHUI, Z.; RONGZHI, W.; TIANWU, Z.; QUAN, P. Visual odometry and scene matching integrated navigation system in uav. In: IEEE INTERNATIONAL CONFERENCE ON INFORMATION FUSION (FUSION), 17., 2014, Salamanca, Espanha. **Proceedings...** IEEE, 2014. p. 1–6. Disponível em: <http://ieeexplore.ieee.org/document/6916056/>. Acesso em: 06 nov. 2017. 6, 18, 20, 45, 151, 193, 208, 209, 210

CHUST, G.; GALPARSORO, I.; BORJA, A.; FRANCO, J.; URIARTE, A. Coastal and estuarine habitat mapping, using lidar height and intensity and multi-spectral imagery. **Estuarine Coastal and Shelf Science**, p. 633–643, 2008. 63

CLARKE, R. Understanding the drone epidemic. **Computer Law and Security Review**, v. 30, n. 3, p. 230–246, 2014. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0267364914000545>. Acesso em: 14 maio 2017. 5, 14, 15, 18

CONTE, G.; DOHERTY, P. An integrated uav navigation system based on aerial image matching. In: IEEE AEROSPACE CONFERENCE, 2008, Big Sky, EUA. **Proceedings...** IEEE, 2008. p. 1–10. Disponível em: <http://ieeexplore.ieee.org/abstract/document/4526556/>. Acesso em: 23 out. 2017. 3, 4, 5, 6, 7, 15, 16, 18, 19, 24, 26, 29, 30, 44, 55, 56, 58, 59, 60, 63, 69, 88, 91, 94, 95, 98, 120, 124, 149, 151, 154, 165, 168, 169, 170, 171, 173, 192, 193, 197, 208, 209, 210, 212

CRESTANI, P. R. J. **Sistemas inteligentes de navegação autônoma: uma abordagem modular e hierárquica com novos mecanismos de memória e aprendizagem**. 214 p. Dissertação (Mestrado em Engenharia Elétrica) — Universidade Estadual de Campinas (UNICAMP), Campinas, 2001. Disponível em: <http://repositorio.unicamp.br/handle/REPOSIP/258796>. Acesso em: 23 out. 2017. 2

DCGISOPENDATA. **LiDAR - Intensity - 2009**. 2017. Disponível em: <http://opendata.dc.gov/datasets/>. Acesso em: 22 mar. 2017. 121, 122

DEPARTMENT, U. D. **Unmanned systems integrated roadmap FY2011-2036**. La Vergne, Tennessee: Createspace, 2013. 102 p. 13

DIGILENT. **Zybo FPGA board reference manual**. 2016. Disponível em: <https://reference.digilentinc.com/_media/zybo:zybo_rm.pdf>. Acesso em: 16 maio 2017. 34, 35, 36

DONG, H.; BARFOOT, T. D. Lighting-invariant visual odometry using lidar intensity imagery and pose interpolation. **Springer Tracts in Advanced Robotics**, v. 92, p. 327–342, 2014. Disponível em: <https://link.springer.com/chapter/10.1007/978-3-642-40686-7_22>. Acesso em: 02 fev. 2018. 150, 154

DOUCET, A.; FREITAS, N. d.; GORDON, N. **Sequential Monte Carlo methods in practice**. Nova York, EUA: Springer, 2001. 582 p. 196, 199, 200

DUDA, R. O.; HART, P. E.; STORK, D. G. **Pattern classification**. 2. ed. Nova York, EUA: Wiley, 2000. 680 p. 75

ELMENREICH, W. An introduction to sensor fusion. **Research Report of Vienna University of Technology**, v. 47, n. 1, p. 1–28, 2002. Disponível em: <http://www.vmars.tuwien.ac>. Acesso em: 21 dez. 2017. 191, 192

FABRO, J. A. **Grupos neurais e sistemas nebulosos: aplicação à navegação autônoma**. 75 p. Dissertação (Mestrado em Engenharia Elétrica) — Universidade Estadual de Campinas (UNICAMP), Campinas, 1996. Disponível em: <http://repositorio.unicamp.br/handle/REPOSIP/261405>. Acesso em: 23 out. 2017. 2

FIGUEIREDO, M. F. **Redes neurais nebulosas aplicadas em problemas de modelagem e controle autônomo**. 141 p. Tese (Doutorado em Engenharia Elétrica) — Universidade Estadual de Campinas (UNICAMP), Campinas, 1997. Disponível em: <http://repositorio.unicamp.br/handle/REPOSIP/260767>. Acesso em: 23 out. 2017. 2

FISCHLER, M. A.; BOLLES, R. C. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. In: **Readings in computer vision**. San Francisco, EUA: Morgan Kaufmann, 1987. p. 726 – 740. Disponível em: <https://www.sciencedirect.com/science/article/pii/B9780080515816500702>. Acesso em: 15 nov. 2017. 165, 166

FLYNN, M. J.; LUK, W. **Projeto de sistemas de computador - system-on-chip**. São Paulo: LTC, 2014. 300 p. 35

FORSYTH, D. A.; PONCE, J. **Computer vision: a modern approach**. 2. ed. Upper Saddle River, EUA: Prentice Hall, 2011. 793 p. 4

FURTADO, H. C. M. **Redes neurais e diferentes métodos de assimilação de dados em dinâmica não linear**. 125 p. Dissertação (Mestrado em Comutação Aplicada) — Instituto Nacional de Pesquisas Espaciais (INPE), São José dos Campos, 2008. Disponível em: <http://urlib.net/sid.inpe.br/mtc-m1780/2008/02.12.12.07>. Acesso em: 23 ago. 2017. 193, 197, 199

GIRARD, A. R.; HOWELL, A. S.; HEDRICK, J. K. Border patrol and surveillance missions using multiple unmanned air vehicles. In: IEEE CONFERENCE ON DECISION AND CONTROL (CDC), 43., 2004, Nassau, Bahamas. **Proceedings...** IEEE, 2004. p. 620–625. Disponível em: <http://ieeexplore.ieee.org/abstract/document/1428713/>. Acesso em: 06 set. 2017. 2

GOLTZ, G. A. M.; SHIGUEMORI, E. H.; CAMPOS VELHO, H. F. Position estimation of uav by image processing with neural networks. In: CONGRESSO BRASILEIRO DE INTELIGêNCIA COMPUTACIONAL (CBIC), 2011, Fortaleza, Brasil. **Anais...** Abricom, 2011. p. 9–17. Disponível em: <http://abricom.org.br/eventos/cbic_2011/st_03_6/>. Acesso em: 14 maio 2017. 5, 7, 16, 18, 26, 29, 56, 59, 60, 72, 82, 83, 88, 91

GORDON, N. J.; SALMOND, D. J.; SMITH, A. F. M. Novel approach to nonlinear/non-gaussian bayesian state estimation. **Radar and Signal Processing**, v. 140, p. 107–113, 1993. Disponível em: <http://ieeexplore.ieee.org/document/210672/>. Acesso em: 14 maio 2017. 200, 201, 203

GORGENS, E. B.; SILVA, A. G. P.; RODRIGUES, L. C. E. **LiDAR: aplicações florestais**. Curitiba, PR: CRV, 2014. 124 p. 47, 48, 49, 50, 51, 52, 121

GUENTHER, C. G.; CUNNINGHAM, A. G.; LAROCQUE, P. E.; REID, D. J. Meeting the accuracy challenge in airbone lidar bathymetry. In: EARSEL WORKSHOP LIDAR REMOTE SENSING OF LAND AND SEA, 4., 2000, Dresden, Alemanha. **Proceedings...** EARSEL, 2000. p. 1–27. Disponível em: <http://www.dtic.mil/dtic/tr/fulltext/u2/a488934.pdf>. Acesso em: 11 fev. 2018. 50, 53, 54

GUSTAFSSON, F. **Statistical sensor fusion**. 2. ed. Estocolmo, Suécia: Studentlitteratur, 2013. 554 p. 6, 19, 192, 196

HARRIS, C.; STEPHENS, M. A combined corner and edge detector. In: ALVEY VISION CONFERENCE, 4., 1988, Manchester, Inglaterra. **Proceedings...** BMVA, 1988. p. 147–151. Disponível em: <http://citeseer.ist.psu.edu/viewdoc/summary>. Acesso em: 07 nov. 2017. 156

HARTLEY, R.; ZISSERMAN, A. **Multiple view geometry in computer vision**. 2. ed. Cambridge, Reino Unido: Cambridge, 2004. 672 p. 37, 42, 47, 154, 161, 162, 164, 181, 182

HAYKIN, S. **Neural networks: a comprehensive foundation**. 2. ed. Upper Saddle River, EUA: Prentice Hall, 1998. 842 p. 72, 73, 74, 75, 76, 77, 78

HERWITZ, S. R.; JOHNSON, L. F.; DUNAGAN, S. E.; HIGGINS, R. G.; SULLIVAN, D. V.; ZHENG, J.; LOBITZ, B. M.; LEUNG, J. G.; GALLMEYER, B. A.; AOYAGI, M.; SLYE, R. E.; BRASS, J. A. Imaging from an unmanned aerial vehicle: agricultural surveillance and decision support. **Computers and Electronics in Agriculture**, v. 44, n. 1, p. 49–61, 2004. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0168169904000456>. Acesso em: 09 out. 2017. 2

HISHAM, M. B.; YAAKOB, S. N.; RAOF, R. A. A.; NAZREN, A. B. A.; EMBEDDED, N. M. W. Template matching using sum of squared difference and normalized cross correlation. In: IEEE STUDENT CONFERENCE ON RESEARCH AND DEVELOPMENT (SCORED), 13., 2015, Kuala Lumpur, Malásia. **Proceedings...** IEEE, 2015. p. 100–104. Disponível em: <http://ieeexplore.ieee.org/document/7449303/>. Acesso em: 21 ago. 2017. 88

HODGSON, A.; KELLY, N.; PEEL, D. Unmanned eerial vehicles (uavs) for surveying marine fauna: a dugong case study. **PLOS ONE**, v. 8, n. 11, p. 1–15, 11 2013. Disponível em: <https://doi.org/10.1371/journal.pone.0079556>. Acesso em: 02 nov. 2017. 2

HUANG, A. S.; BACHRACH, A.; HENRY, P.; KRAININ, M.; MATURANA, D.; FOX, D.; ROY, N. Visual odometry and mapping for autonomous flight using an rgb-d camera. **Robotics Research. Springer Tracts in Advanced Robotics**, v. 100, p. 235–252, 2016. Disponível em:

<https://link.springer.com/chapter/10.1007/978-3-319-29363-9_14>. Acesso em: 25 out. 2017. 17, 150, 151

HUGHES, J.; DAM, A.; MCGUIRE, M.; SKLAR, D.; FOLEY, J.; FEINER, S.; AKELEY, K. **Computer graphics: principles and practice**. 3. ed. Boston, MA, USA: Addison-Wesley Professional, 2013. 1264 p. 60, 61, 66

ITSEEZ. **Open Source Computer Vision Library (OpenCV)**. 2015. Disponível em: <https://github.com/itseez/opencv>. Acesso em: 20 set. 2017. 154, 214

JAIMES, B. R. A. **Estratégias para aumentar a robustez de estimação de posição geográfica em VANTs através de imagens**. 144 p. Dissertação (Mestrado em Engenharia Elétrica) — Universidade de Federal de Minas Gerais (UFMG), Minas Gerais, 2005. Disponível em: <http://pgee.ime.eb.br/pdf/carlos_sousa.pdf>. Acesso em: 12 out. 2017. 16

JAZWINSKI, A. H. **Stochastic Processes and Filtering Theory**. Nova York, EUA: Academic Press, 1970. 376 p. 194

JENSEN, J. R. **Sensoriamento remoto do ambiente: uma perspectiva em recursos terrestres**. São José dos Campos, SP: Parêntese, 2009. 598 p. 47, 49, 51, 52, 53

KALMAN, R. E. A new approach to linear filtering and prediction problems. **Transactions of the ASME – Journal of Basic Engineering**, n. 82, p. 35–45, 1960. Disponível em: <http://www.cs.unc.edu/welch/kalman/media/pdf/Kalman1960.pdf>. Acesso em: 12 maio 2017. 197

KAM, M.; ZHU, X.; KALATA, P. Sensor fusion for mobile robot navigation. **Proceedings of the IEEE**, v. 85, n. 1, p. 108–119, 1997. Disponível em: <http://ieeexplore.ieee.org/document/554212/>. Acesso em: 21 dez. 2017. 19

KEANE, J. F.; CARR, S. S. A brief history of early unmanned aircraft. **Johns Hopkins Apl Technical Digest**, v. 32, p. 558–571, 12 2013. Disponível em: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.686.7958>. Acesso em: 09 jul. 2017. 13, 14

KHALEGHI, B.; KHAMIS, A.; KARRAY, O. F.; RAZAVI, S. N. Multisensor data fusion: a review of the state-of-the-art. **Information Fusion**, v. 14, n. 1, p. 28–44,

2013. Disponível em: <http://www.sciencedirect.com/science/article/pii/S1566253511000558>. Acesso em: 24 ago. 2017. 6, 192

KIM, B. S.; CALISE, A. J.; KAM, M. Nonlinear flight control using neural networks and feedback linearization. In: AEROSPACE CONTROL SYSTEMS, 1., 1993, Westlake Village, EUA. **Proceedings...** IEEE, 1993. p. 176–181. Disponível em: <http://ieeexplore.ieee.org/document/720919/>. Acesso em: 01 jul. 2017. 75

KITT, B.; GEIGER, A.; LATEGAHN, H. Visual odometry based on stereo image sequences with ransac-based outlier rejection scheme. In: IEEE INTELLIGENT VEHICLES SYMPOSIUM, 4., 2010, San Diego, EUA. **Proceedings...** IEEE, 2010. p. 486–492. Disponível em: <http://ieeexplore.ieee.org/abstract/document/5548123/>. Acesso em: 17 nov. 2017. 165, 167

KLEIN, L. A. **Sensor and data fusion: a tool for information assessment and decision making**. Cardiff, Reino Unido: SPIE Pess Book, 2004. 362 p. 191

KRAJNIK, T.; NITSCHE, M.; PEDRE, S.; PREUCIL, L.; MEJAIL, M. E. A simple visual navigation system for an uav. In: INTERNATIONAL MULTI-CONFERENCE ON SYSTEMS, SYGNALS DEVICES, 9., 2012, Chemnitz, Alemanha. **Proceedings...** IEEE, 2012. p. 1–6. Disponível em: <http://ieeexplore.ieee.org/abstract/document/6198031/>. Acesso em: 24 jul. 2017. 17

KROUT, D. W. **International society of information fusion society vision and mission**. 2012. Disponível em: <http://isif.org/about/about-isif>. Acesso em: 23 set. 2017. 191

LENSKIY, A. A.; LEE, J. S. Terrain images segmentation in infra-red spectrum for autonomous robot navigation. In: IEEE INTERNATIONAL FORUM ON STRATEGIC TECHNOLOGY (IFOST), 2010, Ulsan, Coréia do Sul. **Proceedings...** IEEE, 2010. p. 33–38. Disponível em: <http://ieeexplore.ieee.org/document/5667916/>. Acesso em: 02 ago. 2017. 4

LEWIN, S. **Drones in space! NASA's wild idea to explore Mars**. 2015. Disponível em:

<http://www.space.com/30155-nasa-drones-on-mars-video.html>. Acesso em: 21 maio 2017. 2

LI, D.; LI, Q.; CHENG, N.; WU, Q.; SONG, J.; TANG, L. Combined rgbd-inertial based state estimation for mav in gps-denied indoor environments. In: IEEE ASIAN CONTROL CONFERENCE (ASCC), 9., 2013, Istanbul, Turquia. **Proceedings...** IEEE, 2013. p. 1–8. Disponível em: <http://ieeexplore.ieee.org/document/6606361/>. Acesso em: 12 jul. 2017. 15

LI, W.; WANG, C.; CHEN, G. An edge detection method based on optimized bp neural network. In: IEEE INTERNATIONAL SYMPOSIUM ON INFORMATION SCIENCE AND ENGINEERING (ISISE), 2008, Shanghai, China. **Proceedings...** IEEE, 2008. v. 2, p. 40–44. Disponível em: <http://ieeexplore.ieee.org/abstract/document/342541/>. Acesso em: 23 jul. 2017. 72, 82

LIU, J.; WANG, C. An algorithm for image binarization based on adaptive threshold. In: IEEE CHINESE CONTROL AND DECISION CONFERENCE (CCDC), 2009, Guilin, China. **Proceedings...** IEEE, 2009. p. 3958 – 3962. Disponível em: <http://ieeexplore.ieee.org/document/5191959/>. Acesso em: 09 fev. 2018. 66

LOWE, D. Distinctive image features from scale-invariant keypoints. **International Journal of Computer Vision**, v. 60, n. 2, p. 91–110, 2004. Disponível em: <https://link.springer.com/article/10.1023/B:VISI.0000029664.99615.94>. Acesso em: 11 nov. 2017. 156, 157

LUZ, E. F. P. d. **Meta-heurísticas paralelas na solução de problemas inversos**. Tese (Doutorado em Computação Aplicada) — Instituto Nacional de Pesquisas Espaciais (INPE), São José dos Campos, 2012. 78, 79

MAGREE, D.; JOHNSON, E. N. Combined laser and vision-aided inertial navigation for an indoor unmanned aerial vehicle. In: IEEE AMERICAN CONTROL CONFERENCE, 2014, Portland, EUA. **Proceedings...** IEEE, 2014. p. 1900–1905. Disponível em: <http://ieeexplore.ieee.org/abstract/document/6858995/>. Acesso em: 05 nov. 2017. 15

MAIA, J.; GALINDO, J. C. F.; DE CASTRO, C. L.; LEMOS, A. P. Algoritmos de map-matching online para processamento de trajetórias de veículos: um estudo

comparativo. In: SIMPóSIO BRASILEIRO DE AUTOMAçãO INTELIGENTE, 13., 2017, Porto Alegre, Brasil. **Anais...** SBAI, 2017. p. 864–869. Disponível em: <https://www.ufrgs.br/sbai17/papers/paper_249.pdf>. Acesso em: 14 jul. 2017. 2

MATHE, K.; BUSONIU, L. Vision and control for uavs: a survey of general methods and of inexpensive platforms for infrastructure inspection. **Sensors**, v. 15, n. 7, p. 14887–14916, 2015. Disponível em: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4541813/>. Acesso em: 04 mar. 2018. 5, 18

MATHWORKS. **What is camera calibration?** 2017. Disponível em: <https://www.mathworks.com/help/vision/ug/camera-calibration.html>. Acesso em: 21 maio 2017. 38, 43

MCCULLOCH, W. S.; PITTS, W. H. A logical calculus of the ideas immanent in nervous activity. **Bulletin of Mathematical Biophysics**, v. 5, n. 4, p. 115–133, 1943. Disponível em: <https://doi.org/10.1007/BF02478259>. Acesso em: 23 jul. 2017. 72

MCMANUS, C.; FURGALE, P.; BARFOOT, T. D. Towards lighting invariant visual navigation: an appearance-based approach using scanning laser-rangefinders. **Robotics and Autonomous Systems**, v. 61, n. 8, p. 836–852, 2013. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0921889013000705>. Acesso em: 21 fev. 2018. 5, 47, 48, 50, 51, 150

MEBARKI, R.; CACACE, J.; LIPPIELLO, V. Velocity estimation of an uav using visual and imu data in a gps-denied environment. In: IEEE INTERNATIONAL SYMPOSIUM ON SAFETY, SECURITY, AND RESCUE ROBOTICS (SSRR), 2013, Linkoping, Suécia. **Proceedings...** IEEE, 2013. p. 1–6. Disponível em: <http://ieeexplore.ieee.org/document/6719334/>. Acesso em: 24 jun. 2017. 15

MIKHAIL, E. M.; BETHEL, J. S.; MCGLONE, J. C. **Introduction to modern photogrametry**. Nova York: John Wiley, 2001. 479 p. 52

MIRAZAEI, F. M.; KOTTAS, D. G.; ROUMELIOTIS, S. I. 3d lidar–camera intrinsic and extrinsic xalibration: identifiability and analytical least-squares-based initialization. **The International Journal of Robotics Research**, v. 31, n. 4,

p. 452–467, 2012. Disponível em: <http://www.users.cs.umn.edu/stergios/papers/IJRR-2012-LidarCameraCalib.pdf>. Acesso em: 20 set. 2017. 181

MONTEIRO, L. H. A. **Sistemas dinâmicos**. São Paulo: Livraria da Física, 2006. 625 p. 193

MONTEIRO, M. V. T.; SHIGUEMORI, E. H.; MARTINS, M. P.; MEDEIROS, F. L. L.; DOMICIANO, M. A. P. Ikonos and video image registration using gradient correlation, mutual information and combined mutual and gradient information. In: BRAZILIAN SYMPOSIUM ON COMPUTER GRAPHICS AND IMAGE PROCESSING, 19., 2006, Manaus, Brasil. **Proceedings...** IEEE, 2006. p. 1–6. Disponível em: <https://ieeexplore.ieee.org/document/4027041/>. Acesso em: 02 ago. 2017. 59

MOSHIRI, B. Tutorial: a sensor data fusion, principles and applications. In: IEEE INTERNATIONAL SYMPOSIUM ON OPTOMECHATRONIC TECHNOLOGIES (ISOT), 2010, Toronto, Canada. **Proceedings...** IEEE, 2010. p. 1–2. Disponível em: <http://ieeexplore.ieee.org/document/5687380/>. Acesso em: 13 maio 2017. 30

MOTA, R. L. M. H.; SHIGUEMORI, E. H.; RAMOS, A. C. B. Application of self-organizing maps at change detection in amazon forest. In: IEEE INTERNATIONAL CONFERENCE ON INFORMATION TECHNOLOGY: NEW GENERATIONS, 11., 2014, Las Vegas, EUA. **Proceedings...** IEEE, 2014. p. 371–376. Disponível em: <https://ieeexplore.ieee.org/document/6822225/>. Acesso em: 21 out. 2017. 3, 236

NAYAR, S. K. Catadioptric omnidirectional camera. In: IEEE COMPUTER SOCIETY CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION, 1997, San Juan, Porto Rico. **Proceedings...** IEEE, 1997. p. 482–488. Disponível em: <https://ieeexplore.ieee.org/abstract/document/609369/>. Acesso em: 22 mar. 2017. 37, 48

NAZEMZADEH, P.; FONTANELLI, D.; MACII, D.; PALOPOLI, L. Indoor localization of mobile robots through qr code detection and dead reckoning data fusion. **IEEE ASME Transactions on Mechatronics**, v. 22, n. 6, p. 2588–2599, 2017. Disponível em: <http://ieeexplore.ieee.org/document/8066377/>. Acesso em: 24 dez. 2017. 48, 191

NEWCOME, L. R. **Unmanned aviation: a brief history of unmanned aerial vehicles**. Atlanta, Georgia: American Institute of Aeronautics and Astronautics, 2004. 176 p. 14

NGHIA, H. **Finding optimal rotation and translation between corresponding 3d points**. 2013. Disponível em: <http://nghiaho.com/?page_id=671>. Acesso em: 22 nov. 2017. 265, 267

NIKOLIC, J.; BURRI, M.; REHDER, J.; LEUTENEGGER, S.; HUERZELER, C.; SIEGWART, R. A uav system for inspection of industrial facilities. In: IEEE AEROSPACE CONFERENCE, 2013, Big Sky, EUA. **Proceedings...** 2013. p. 1–8. Disponível em: <http://ieeexplore.ieee.org/abstract/document/6496959/>. Acesso em: 23 out. 2017. 2

NIKU, S. B. **Introdução à robótica: análise, controle, aplicações**. 2. ed. Rio de Janeiro, RJ: LTC, 2014. 382 p. 1

NISHITANI, A. T. N. **Localização baseada em odometria visual**. 91 p. Dissertação (Mestrado em Ciências - Ciências de Computação e Matemática Computacional) — Universidade de São Paulo, São Carlos, 2015. Disponível em: <http://www.teses.usp.br/teses/disponiveis/55/55134/tde-17082016-095838/pt-br.php>. Acesso em: 14 mar. 2018. 37, 38, 44, 155, 165

NISTER, D.; NARODITSKY, O.; BERGEN, J. Visual odometry. In: IEEE COMPUTER SOCIETY CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION (CVPR), 2004, Washington, EUA. **Proceedings...** IEEE, 2004. p. 652–659. Disponível em: <http://ieeexplore.ieee.org/abstract/document/1315094/>. Acesso em: 14 set. 2017. 5, 18, 149, 150, 152, 153, 154, 170, 171, 172, 173, 178

OTSU, N. A threshold selection method from gray-level histograms. **IEEE Transactions on Systems, Man, and Cybernetics**, v. 9, n. 1, p. 62–66, 1979. Disponível em: <http://ieeexplore.ieee.org/document/4310076/>. Acesso em: 23 abr. 2017. 66, 83

PEDRINI, H.; SCHARTZ, W. R. **Análise de imagens digitais: principios, algoritmos e aplicações**. São Paulo: Thonson Learning, 2008. 508 p. 26, 37, 39, 40, 43, 44, 51, 55, 60, 62, 63, 64, 66, 67, 68, 70, 71, 82, 173

PEDRONI, V. A. **Eletrônica digital moderna e VHDL**. São Paulo: Campus, 2008. 648 p. 4, 44, 85

PETER, C. **Robotics, vision and control**. Berlin, Alemanha: Springer International, 2011. 492 p. 1, 2, 37

Portal Action. **Portal action: probabilidades**. 2015. Disponível em: <http://www.portalaction.com.br/probabilidades>. Acesso em: 21 maio 2017. 269

Raspberry Pi. **What is a Raspberry Pi**. 2015. Disponível em: <https://www.raspberrypi.org/help/what-is-a-raspberry-pi/>. Acesso em: 13 maio 2017. 31, 32

REBOUÇAS, R. A.; ELLER, Q. C.; HABERMANN, M.; SHIGUEMORI, E. H. Embedded system for visual odometry and localization of moving objects in images acquired by unmanned aerial vehicles. In: SIMPÓSIO BRASILEIRO DE ENGENHARIA DE SISTEMAS COMPUTACIONAIS, 3., 2013, Niterói, Brasil. **Anais...** 2013. p. 35–40. Disponível em: <http://ieeexplore.ieee.org/document/6825339/>. Acesso em: 04 nov. 2017. 151, 154, 165, 169

ROSS, S. **Probabilidade. um curso moderno com aplicações**. 8. ed. Porto Alegre, Brasil: Bookman, 2010. 608 p. 269

SA, I.; HE, H.; HUYNH, V.; CORKE, P. Monocular vision based autonomous navigation for a cost-effective mav in gps-denied environments. In: IEEE/ASME INTERNATIONAL CONFERENCE ON ADVANCED INTELLIGENT MECHATRONICS, 2013, Wollongong, Australia. **Proceedings...** IEEE, 2013. p. 1355–1360. Disponível em: <http://ieeexplore.ieee.org/abstract/document/6858995/>. Acesso em: 24 jul. 2017. 15

SACCO, W. F.; OLIVEIRA, C. R. E. A new stochastic optimization algorithm based on a particle collision metaheuristic. In: WORLD CONGRESS ON STRUCTURAL AND MULTIDISCIPLINARY OPTIMIZATION, 6., 2005, Rio de Janeiro, Brasil. **Proceedings...** COPPE, 2005. p. 1–6. Disponível em: <https://pdfs.semanticscholar.org/4f40/3dda49423ad834d2e3a4c8db1ea2ccde3442.pdf>. Acesso em: 01 jun. 2017. 79

SANTOS, A. L.; TORRES, L. C. B.; CAMPOS VELHOR, H. F.; SHIGUEMORI, E. H.; BRAGA, A. P. Controle e navegação de vants através de utilização de template matching. In: COMPUTER ON THE BEACH, 2004, Florianópolis, Brasil. **Proceedings...** UNIVALI, 2013. p. 70–78. Disponível em: <https:

//siaiap32.univali.br/seer/index.php/acotb/article/view/6202>. Acesso em: 06 set. 2017. 16

SCARAMUZZA, D.; ACHTELIK, M. C.; DOITSIDIS, L.; FRIEDRICH, F.; KOSMATOPOULOS, E.; MARTINELLI, A.; ACHTELIK, M. W.; CHLI, M.; CHATZICHRISTOFIS, S.; KNEIP, L.; GURDAN, D.; HENG, L.; LEE, G. H.; LYNEN, S.; POLLEFEYS, M.; RENZAGLIA, A.; SIEGWART, R.; STUMPF, J. C.; TANSKANEN, P.; TROIANI, C.; WEISS, S.; MEIER, L. Vision-controlled micro flying robots: from system design to autonomous navigation and mapping in gps-denied environments. **IEEE Robotics Automation Magazine**, v. 21, n. 3, p. 26–40, 2014. Disponível em: <http://ieeexplore.ieee.org/document/6880770/>. Acesso em: 11 jul. 2017. 15, 163

SCARAMUZZA, D.; FRIEDRICH, F. Visual odometry part i: the first 30 years and fundamentals. **IEEE Robotics and Automation Magazine**, v. 18, n. 4, p. 80–92, 2011. Disponível em: <http://ieeexplore.ieee.org/document/6096039/>. Acesso em: 12 jul. 2017. 5, 16, 17, 18, 24, 149, 153, 154, 155, 156, 157, 165, 170, 171, 172, 178

_____. Visual odometry part ii: matching, robustness, optimization, and applications. **IEEE Robotics and Automation Magazine**, v. 19, n. 2, p. 78–90, 2012. Disponível em: <http://ieeexplore.ieee.org/document/6153423/>. Acesso em: 24 mar. 2018. 18, 24, 149, 153, 154, 156, 168, 170, 171

SCHMITT, M.; ZHU, X. Data fusion and remote sensing: an ever-growing relationship. **IEEE Geoscience and Remote Sensing Magazine**, v. 4, n. 4, p. 6–23, 2016. Disponível em: <http://ieeexplore.ieee.org/document/7740215/>. Acesso em: 21 dez. 2017. 191

SCHON, T. B. **Computational methods for nonlinear estimation**. 159 p. Tese (Doutorado em Engenharia Elétrica) — Universidade de Linkoping - (LIU), Linkoping, 2003. 200, 203, 204

SHI, J.; TOMASI, C. Good features to track. In: IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION, 1994, Seattle, EUA. **Proceedings...** IEEE, 1994. p. 593–600. Disponível em: <http://citeseer.ist.psu.edu/viewdoc/summary>. Acesso em: 08 nov. 2017. 156

SHIGUEMORI, E. H.; MARTINS, M. P.; MONTEIRO, M. V. T. Landmarks recognition for autonomous aerial navigation by neural networks and gabor transform. In: PROCEEDINGS OF IMAGE PROCESSING: ALGORITHMS AND SYSTEMS, 5., 2007, San Jose, EUA. **Proceedings...** SPIE, 2007. p. 1–9. Disponível em: <http://dx.doi.org/10.1117/12.705138>. Acesso em: 02 ago. 2017. 3, 4, 7

SILVA, C. A. O.; GOLTZ, G. A. M.; SHIGUEMORI, E. H.; CASTRO, C. L.; CAMPOS VELHO, H. F.; BRAGA, A. P. Avaliação da técnica de casamento de imagens para navegação autônoma de vants. In: CONGRESSO BRASILEIRO DE INTELIGêNCIA COMPUTACIONAL, 12., 2015, Curitiba, Brasil. **Proceedings...** ABRICOM, 2015. p. 1–6. Disponível em: <http://abricom.org.br/eventos/cbic_2015/cbic2015_submission_55/>. Acesso em: 21 out. 2017. 16

SILVA RAMOS, J. **Algoritmo de casamento de imagens com filtragem adaptativa de outliers**. 102 p. Dissertação (Mestrado em Ciência da Computação e Matemática Computacional) — Instituto de Ciências Matemáticas e Computação - ICMC-USP, São Carlos, 2017. Disponível em: <http://www.teses.usp.br/teses/disponiveis/55/55134/tde-02022017-110428/en.php>. Acesso em: 11 out. 2017. 165, 166

SILVA, W.; SHIGUEMORI, E. H.; VIJAYKUMAR, N. L.; CAMPOS VELHO, H. F. Estimation of uav position with use of thermal infrared images. In: IEEE INTERNATIONAL CONFERENCE ON SENSING TECHNOLOGY (ICST), 9., 2015, Auckland, Nova Zelândia. **Proceedings...** IEEE, 2015. p. 828–833. Disponível em: <http://ieeexplore.ieee.org/abstract/document/7438511/>. Acesso em: 27 out. 2017. 5, 7, 17, 18, 29, 56, 59, 60, 63, 66, 72, 82, 88, 91

SONG, J.; HAN, S.; YU, K.; KIM, Y. Assessing the possibility of land-cover classification using lidar intensity data. **International Archives of Photogrammetry Remote Sensing and Spatial Information Sciences**, v. 34, p. 1–4, 2012. 63

SOUTO, G. **Decomposição em valores singulares**. 102 p. Trabalho de Conclusão de Curso (Bacheral em Matemática) — Centro de Ciências Físicas e Matemáticas - Universidade Federal de Santa Catarina, Florianópolis, 2000. Disponível em: <https://repositorio.ufsc.br/bitstream/handle/123456789/97051/Gilberto_Solto.PDF>. Acesso em: 02 out. 2017. 259

SOUZA C. R, M. d. **Interferidores de GPS: análise do sistema e de potenciais fontes de interferência**. 90 p. Dissertação (Mestrado em Ciências e Engenharia Elétrica) — Instituto Militar de Engenharia (IME), Rio de Janeiro, 2005. Disponível em: <http://pgee.ime.eb.br/pdf/carlos_sousa.pdf>. Acesso em: 12 out. 2017. 3, 7

SUAIB, N. M.; MARHABAN, M. H.; SARIPAN, M. I.; AHMAD, S. A. Performance evaluation of feature detection and feature matching for stereo visual odometry using sift and surf. In: IEEE REGION SYMPOSIUM, 10., 2014, Kuala Lumpur, Malásia. **Proceedings...** IEEE, 2014. p. 200–203. Disponível em: <http://ieeexplore.ieee.org/abstract/document/6863025/>. Acesso em: 12 set. 2017. 168

TERRY, P. J.; VU, D. Edge detection using neural networks. In: IEEE ASILOMAR CONFERENCE ON SIGNALS, SYSTEMS AND COMPUTERS, 27., 1993, Pacific Grove, EUA. **Proceedings...** IEEE, 1993. p. 391–395. Disponível em: <http://ieeexplore.ieee.org/abstract/document/342541/>. Acesso em: 20 fev. 2018. 72, 82

TRAJKOVIC, M.; HEDLEY, M. Fast corner detection. **Image and Vision Computing**, v. 16, n. 2, p. 75–87, 1998. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0262885697000565>. Acesso em: 09 nov. 2017. 156

TSALLIS, C. Nonextensive statistics: theoretical, experimental and computational evidences and connections. **Brazilian Journal of Physics**, v. 29, p. 1–35, 1999. Disponível em: <http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0103-97331999000100002>. Acesso em: 03 maio 2017. 204, 235

TULSYAN, A.; BHUSHAN, R. G.; KHARE, S. R. Particle filtering without tears: a primer for beginners. **Computers and Chemical Engineering**, v. 95, p. 130–145, 2016. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0098135416302769>. Acesso em: 12 ago. 2017. 193, 194, 195, 198, 200, 216

VALENTFX. **Logi-Pi user guide**. 2014. Disponível em: <https://www.wiltronics.com.au/wp-content/uploads/datasheets/raspberry_pi/LOGi-Pi.pdf>. Acesso em: 21 maio 2017. 33, 34

VAN WIEN, A. O. **Naval aviation in World War I**. Washington DC, EUA: Chief of Naval Operations, 1969. 90 p. 13

WAHARTE, S.; TRIGONI, N. Supporting search and rescue operations with uavs. In: IEEE INTERNATIONAL CONFERENCE ON EMERGING SECURITY TECHNOLOGIES, 2010, Canterbury, Inglaterra. **Proceedings...** IEEE, 2010. p. 142–147. Disponível em: <http://ieeexplore.ieee.org/abstract/document/5600072/>. Acesso em: 21 out. 2017. 2

WALTZ, E. L.; LLINIAS, J. **Multisensor data fusion**. Norwood, EUA: Artech House Publishers, 1999. 488 p. 191

WANG, C.; WANG, T.; LIANG, J.; CHEN, Y.; WU, Y. Monocular vision and imu based navigation for a small unmanned helicopter. In: IEEE CONFERENCE ON INDUSTRIAL ELECTRONICS AND APPLICATIONS (ICIEA), 7., 2012, Cidade de Singapura, Singapura. **Proceedings...** IEEE, 2012. p. 1694–1699. Disponível em: <http://ieeexplore.ieee.org/document/6360998/>. Acesso em: 24 jul. 2017. 15, 17

WANG, C.; WANG, T.; LIANG, J.; CHEN, Y.; ZHANG, Y.; WANG, C. Monocular visual slam for small uavs in gps-denied environments. In: IEEE INTERNATIONAL CONFERENCE ON ROBOTICS AND BIOMIMETICS (ROBIO), 2012, Guangzhou, China. **Proceedings...** IEEE, 2012. p. 896–901. Disponível em: <http://ieeexplore.ieee.org/document/6491082/>. Acesso em: 27 out. 2017. 15

WEDGE, D. **The RANSAC song**. 2012. Disponível em: <http://danielwedge.com/ransac/>. Acesso em: 02 nov. 2017. 166

WEISS, S.; ACHTELIK, M. W.; LYNEN, S.; CHLI, M.; SIEGWART, R. Real-time onboard visual-inertial state estimation and self-calibration of mavs in unknown environments. In: IEEE INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION (ICRA), 2012, Saint Paul, EUA. **Proceedings...** IEEE, 2012. p. 957–964. Disponível em: <http://ieeexplore.ieee.org/document/6225147/>. Acesso em: 14 mar. 2018. 15, 19

WELCH, G.; BISHOP, G. **An introduction to the Kalman filter**. Chapel Hill, EUA: University of North Carolina at Chapel Hill, 1995. 197

WEST, G. R.; WIGGINS, C. E. Airborne lidar bathymetry in the management of florida's coastal zone. **Integrated Costal Zone**, p. 69–72, 2000. Disponível em: <http:

//shoals.sam.usace.army.mil/downloads/Publications/47Irish_00.pdf>. Acesso em: 10 fev. 2018. 47, 48, 53, 54

XILINX. **Spartan-6 family overview**. 2015. Disponível em: <https://www.xilinx.com/support/documentation/data_sheets/ds160.pdf>. Acesso em: 11 maio 2017. 33

ZITOVA, B.; FLUSSER, J. Image registration methods: a survey. **Image and Vision Computing**, v. 21, n. 11, p. 977–1000, 2003. 55, 56

APÊNDICE A - DECOMPOSIÇÃO EM VALORES SINGULARES

Neste apêndice são abordados alguns conceitos e definições de Algebra Linear importantes para o entendimento da Decomposição em Valores Singulares (SVD, do inglês, Singular Value Decomposition). Não serão feitas as demonstrações e provas das definições e teoremas, pois estes podem ser obtidas em Boldrini (1986). e Souto (2000).

A.1 NOTAÇÕES E DEFINIÇÕES

Definição A.1: $\mathbb{R}^n$ é o conjunto de veotres coluna

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \tag{A.1}$$

Definição A.2: x^T é o vetor transposto de x

$$x^T = \begin{bmatrix} x_1, x_2, \cdots, x_n \end{bmatrix} \tag{A.2}$$

Definição A.3: $\langle x, y \rangle$ é o produto escalar ou produto interno usual em $\mathbb{R}^n$ entre os veotres x e y

$$\langle x, y \rangle = x^T y = x_1 y_1 + x_2 y_2 + \cdots + x_n y_n \tag{A.3}$$

Definição A.4: $\|x\|$ é a norma euclidiana do vetor x

$$\|x\| = \sqrt{x^T x} = \sqrt{x_1^2 + \cdots + x_n^2} \tag{A.4}$$

Definição A.5: Sejam os vetores x e y pertencentes ao $\mathbb{R}^n$, a distância entre eles é definida por

$$d(x, y) = \sqrt{\sum_{j=0}^{n-1} (x_j - y_j)^2} \tag{A.5}$$

Definição A.6: $\mathbb{R}^{m\times n}$ é o conjunto de matrizes reais de ordem $m \times n$

A matriz A, exibida na Equação A.6, é um exemplo de matriz $\in$ $\mathbb{R}^{m\times n}$, $a_{i,j}$ é cada componente da matriz A, com $i = 0, 1, 2, \cdots, m-1$ e $j = 0, 1, 2, \cdots, m-1$. Se $m = n$, A é denominda matriz quadrada de ordem n, os componentes $a_{i,j}$ com $i = j$ são chamados de componentes da diagonal principal da matriz A

$$A = \begin{bmatrix} a_{0,0} & a_{0,1} & \cdots & a_{0,n-1} \\ a_{1,0} & a_{1,1} & \cdots & a_{1,n-1} \\ \vdots & \vdots & & \vdots \\ a_{m-1,0} & a_{m-1,1} & \cdots & a_{m-1,n-1} \end{bmatrix} \tag{A.6}$$

Definição A.7: A_T é a matriz transposta de A

$$A^T = \begin{bmatrix} a_{0,0} & a_{1,0} & \cdots & a_{m-1,0} \\ a_{0,1} & a_{1,1} & \cdots & a_{m-1,1} \\ \vdots & \vdots & & \vdots \\ a_{0,n-1} & a_{1,n-1} & \cdots & a_{m-1,n-1} \end{bmatrix} \tag{A.7}$$

Definição A.8: Uma matriz A é denotada como simétrica se

$$A^T = A \tag{A.8}$$

Definição A.9: Uma matriz quadrada A é dita invertível quando existe outra matriz denotada por A^{-1} tal que

$$\begin{aligned} A^{-1} \cdot A &= I \\ A \cdot A^{-1} &= I \end{aligned} \tag{A.9}$$

onde I é a matriz identidade.

Definição A.10: Uma matriz quadrada A é ortogonal se sua matriz inversa coincide com sua matriz transposta.

$$A^{-1} = A^T \tag{A.10}$$

Definição A.11: Um espaço vetorial $V_n(\mathbb{R})$ em que é válida a distância entre vetores mostrada na Eqaução A.5 é denotado espaço unitário n-dimensional, denotado por

$$\mathbb{U}_{\ltimes} \tag{A.11}$$

Definição A.12: seja $\mathbb{U}_{\ltimes}$ um espaço vetorial n-dimensional. Sejam V e W espaços vetoriais. A função $T : V \to W$ é uma transformação linear se satisfazer as seguintes condições

$$\begin{gathered} cada\ vetor\ x\ \in\ \mathbb{U}_{\ltimes}\ associa\ um\ único\ vetor\ y\ \in\ \mathbb{U}_{\ltimes} \\ T(x+y) = T(x) + T(y), x\ e\ y\ \in\ \mathbb{U}_{\ltimes} \\ T(\alpha x) = \alpha T(x), \alpha\ é\ escalar\ e\ x\ \in\ \mathbb{U}_{\ltimes} \end{gathered} \tag{A.12}$$

Definição A.13: dada a transformação linear aplicada à matriz A. Se existir um vetor x que satisfaça a Equação A.13, então λ é denominado autovalor de A associado ao autovetor X

$$Ax = \lambda X \tag{A.13}$$

Definição A.14 posto de uma matriz é a quantidade de vetores, formados pelas linhas ou colunas da matriz, que são linearmente independentes.

A.2 TEOREMA DA DECOMPOSIÇÃO EM VALORES SINGULARES

Teorema: seja uma matriz A de dimensão $m \times n$ de posto r. Então existe uma fatorização da forma

$$A = U\Sigma V^T \tag{A.14}$$

onde U é uma matriz ortogonal com dimensão $m \times m$, Σ é uma matriz diagonal com dimensão $m \times n$ com números reais não-negativos na diagonal e V^T é uma matriz ortogonal com dimensão $n \times n$.

Os vetores $u_1, \cdots u_m$ coluna da matriz U são autovetores de AA^T e formam uma base ortonormal em $\mathbb{R}^{>}$. Os vetores $v_1, \cdots v_n$ coluna da matriz V são autovetores de A^TA

e formam uma base ortonormal em $\mathbb{R}^{\ltimes}$. Os valores de $\sigma_1^2, \cdots, \sigma_r^2$ são os autovalores não nulos de A^TA. Os valores $\sigma_1, \cdots, \sigma_r$ são denominados valores singulares de A e compõem a digonal principal de Σ.

Para realizar a $SVD(A)$, deve-se seguir as seguintes etapas:

1- Obter a matriz A^TA.

2- Os valores singulares de A são a raiz quadrada dos autovalores não-negativos da matriz A^TA. Os valores singulares de A compõem a diagonal principal da matriz diagonal Σ.

3- As colunas de V são os autovetores unitários de A^TA.

4- Os autovetores unitários de U são obtidos pela seguinte relação

$$u_i = \frac{1}{\sigma_i} A v_i \tag{A.15}$$

Exemplo

Seja $A = \begin{bmatrix} 5 & 5 \\ -1 & 7 \end{bmatrix}$

obtenha $SVD(A) = U\Sigma V^T$

1- O primeiro passo é calcular os valores singulares de A. Com este objetivo, determina-se a matriz $A^TA = \begin{bmatrix} 26 & 18 \\ 18 & 74 \end{bmatrix}$

2- Agora determina-se os autovetores da matriz A^TA. Com este objetivo, calcula-se $det(A^TA - \lambda I) = 0$, ou seja, são as raízes da equação de segundo grau obtida por $det \begin{bmatrix} 26-\lambda & 18 \\ 18 & 74-\lambda \end{bmatrix} = 0$. Resolvendo a equação, obtêm-se as raízes $\lambda_1 = 20$ e $\lambda_2 = 80$. Os valores singulares de A são $\sigma_1 = \sqrt{\lambda_1}$ e $\sigma_2 = \sqrt{\lambda_2}$, ou seja, $\sigma_1 = 2\sqrt{5}$ e $\sigma_2 = 4\sqrt{5}$

3- O autovetor unitário v_1 de λ_1 é $\begin{bmatrix} \frac{-3}{\sqrt{10}} \\ \frac{1}{\sqrt{10}} \end{bmatrix}$. O autovetor unitário v_2 de λ_2 é $\begin{bmatrix} \frac{1}{\sqrt{10}} \\ \frac{3}{\sqrt{10}} \end{bmatrix}$.

4- Pela relação exibida na Equação A.15 $u_1 = \begin{bmatrix} \frac{-1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix}$ e $u_2 = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix}$

Desta forma $U = \begin{bmatrix} \frac{-1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix}$, $\Sigma = \begin{bmatrix} 2\sqrt{5} & 0 \\ 0 & 4\sqrt{5} \end{bmatrix}$ e $V = \begin{bmatrix} \frac{-3}{\sqrt{10}} & \frac{1}{\sqrt{10}} \\ \frac{1}{\sqrt{10}} & \frac{3}{\sqrt{10}} \end{bmatrix}$

APÊNDICE B - DETERMINAR A ROTAÇÃO E TRANSLAÇÃO ENTRE PONTOS CORRESPONDENTES

Neste apêndice, é abordado o método de Besl e McKay (1992), que determina a matriz de rotação e o vetor de tranlação entre pontos correspondentes.

A Figura B.1 apresenta uma ilustração do problema para o caso mais simples, de se obter a matriz de rotação R (matriz de dimensão 3×3) e o vetor de translação t (matriz coluna de dimensão 3×1) entre 3 pontos correspondentes (a quantidade mínima requerida para resolver o problema).

Figura B.1 - Conjunto de 3 pontos correspondentes. R e t são as matrizes de rotação e o vetor de translação, respectivamente.

Fonte: Obtido de Nghia (2013).

Na Figura B.1, os pontos correspondentes possuem a mesma cor, R é a matriz de rotação e t o vetor de translação.

A solução apresentada por Besl e McKay (1992) determina a matriz de rotação e o vetor de translação ótimos que alinham os pontos correspondentes. Estas matrizes são denominads matrizes de transformação Euclidiana ou matrizes transformações de corpos rígidos, pois preservam a forma e o tamanho do corpo delimitado pelo conjunto de pontos. Isto contrasta com uma transformação afim, que inclui mudanças na dimensão e distorções do corpo (BESL; MCKAY, 1992).

Dada um conjunto de pontos, designado por A, e o seus pontos correspondentes, designado por B, a Equação B.1 mostra a operação para o correto alinhamento

entre os pontos pertencentes a estes conjuntos.

$$B = R \times A + t \tag{B.1}$$

onde R e t são as matrizes de transformações aplicadas ao conjunto de pontos A para alinhá-lo com o conjunto de pontos B.

O algoritmo para determinar estas matrizes de transformações pode ser dividido em três etapas:

1- Determinar os centróides dos conjuntos de pontos;

2- Determinar a matriz de rotação;

3- Determinar o vetor de translação.

B.1 DETERMINAR OS CENTRÓIDES DE UM CONJUNTO DE PONTOS

O centróide de um conjunto de pontos é dado pelo ponto médio de cada uma das componentes dos pontos pertencentes ao conjunto.

Dado o conjunto de pontos A

$$A = P_1, \cdots, P_n \tag{B.2}$$

formado por n pontos designados por P_i, onde $P_1 = \begin{bmatrix} x_1 \\ y_2 \\ z_3 \end{bmatrix}$ e $P_n = \begin{bmatrix} x_n \\ y_n \\ z_n \end{bmatrix}$

As coordenadas do centróide deste conjunto de pontos é $C_A = \begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix}$ são: $x_c = \sum_{i=1}^{n} \frac{x_i}{n}$, $y_c = \sum_{i=1}^{n} \frac{y_i}{n}$ e $z_c = \sum_{i=1}^{n} \frac{z_i}{n}$.

B.2 DETERMINAR A MATRIZ DE ROTAÇÃO

Para determinar a matriz de rotação, o primeiro passo é realizar a translação dos centroides de cada conjunto de pontos para a origem. Posteriormente, aplica-se a

mesma translação para cada ponto, de modo que os centroides mantenham-se os mesmos. A Figura B.2 ilustra a translação dos centroides e sos pontos de cada centróide para a origem.

Figura B.2 - Translação dos centroides de cada conjunto de pontos para a origem.

Fonte: Obtido de Nghia (2013).

A tranlação, ilustrada na Figura B.2, é necessária, para que a única diferença entre o conjunto de pontos, a partir do momento desta translação, seja a rotação para alinhá-los (BESL; MCKAY, 1992).

Após a translação dos centróides e dos pontos, calcula-se a matriz de covariância H do conjunto de pontos. No cálculo de H, o valor aplicado como média é o centróide de cada conjunto de pontos. A Equação B.3 apresenta o cálculo da matriz H

$$H = \sum_{i=1}^{N} (P_A^i - C_A)(P_B^i - C_B)^T \tag{B.3}$$

onde P_A^i e P_A^i são os pontos do conjunto A e B, respectivamente. C_A e C_B são os centróides de A e B, respectivamente.

De posse da matriz H, aplica-se a decomposição em valores sigulares de H, ou seja, $SVD(H)$ [1], que produz como resposta três matrizes U, Σ, V. As Equações B.4 e B.5 apresentam a decomposição em valores singulares da matriz H.

[1]Informações sobre a decomposição em valores singulares estão presentes no Apêndice A

$$[U, \Sigma, V] = SVD(H) \tag{B.4}$$

$$H = U\Sigma V \tag{B.5}$$

A matriz de rotação para alinhar os conjuntos de pontos é igual a:

$$R = UV^T \tag{B.6}$$

onde U e V são as matrizes obtidas pela decomposição em valores singulares de H.

Segundo (BESL; MCKAY, 1992), há um caso especial ao determinar a matriz de rotação, algumas vezes o a decomposição em valores singulares retorna uma matriz de rotação denominada matriz de reflexão, que é numericamente correta, mas não representa a rotação correta entre o conjunto de pontos. Para obter a matriz de rotação correta, deve-se verificar o determinante da matriz R, se o valor do determinante for negativo, multiplica-se a terceira coluna da matriz R por -1. Esta operação garante que a matriz R obtida é a correta.

B.3 DETERMINAR O VETOR DE TRANSLAÇÃO

Após a obtenção da matriz de rotação pode-se obter o vetor de translação segundo a Equação B.7.

$$t = -R \times C_A + C_B \tag{B.7}$$

onde C_A e C_B são os centróides do conjunto de pontos A e B, respectivamente.

APÊNDICE C - CONCEITOS DE PROBABILIDADE

Neste apêndice são discutidos aguns conceito de probabilidade que são importantes para a teoria de estimação Baesyana. Não serão feitas as demosntrações e provas da definições apresentadas, pois estas podem ser obtidas em Ross (2010) e Portal Action (2015)

C.1 VARIÁVEL ALEATÓRIA

Dado um experimento e Ω o espaço amostral associado a esse experimento. Uma função x, que associa a cada elemento $\omega \in \Omega$ a um número real, $X(\omega)$, é denominada variável aleatória.

Seja X uma variável aleatória. Caso o número de valores possíveis de X seja enumerável (finito ou infinito), dizemos que X é uma variável aleatória discreta.

Como X uma variável aleatória discreta, a cada possível resultado x_i é associado um número $p(x_i) = \mathbb{P}(X = x_i)$, denominado probabilidade de x_i. Os números $p(x_i)$, $i = 1, 2, \cdots$ devem satisfazer as seguintes condições:

1- $p(x_i) \geq 0 \; \forall \; i$;

2- $\sum\limits_{i=1}^{\infty} p(x_i) = 1$.

A função p é denominada função de probabilidade da variável aleatória X.

Seja X uma variável aleatória. Se o contradomínio de X é um intervalo ou uma coleção de intervalos em $\mathbb{R}_x$, então X é uma variável aleatória contínua.

X é uma variável aleatória contínua se existe uma função $\mathbb{R} \to [0, +\infty)$ denominada Função Densidade de Probabilidade (FDP), que satisfaz às seguintes propriedades:

1- $f(x) \geq 0$, para todo $x \in \mathbb{R}_x$

2- $\int_{-\infty}^{\infty} f(x)dx = 1$

Além disso, para qualquer $c, d \in \mathbb{R}_x$, com $c < d$ que

$$\mathbb{P}(c < X < d) = \int_c^d f(x)dx. \tag{C.1}$$

C.2 VALOR ESPERADO DE VARIÁVEL ALEATÓRIA CONTÍNUA

Seja X uma variável aleatória contínua com FDP $f(x)$. O valor esperado, também denominado média ou esperança matemática, de X é igual a:

$$E(X) = \int_{-\infty}^{\infty} x f(x) d_x \tag{C.2}$$

C.3 VARIÂNCIA DE VARIÁVEL ALEATÓRIA

Seja X uma variável aleatória. A variância de X, denotada por $Var(X)$ (ou por σ_X^2), é igual a:

$$Var(X) = E((X - E(X))^2) \tag{C.3}$$

A raiz quadrada da variância $Var(X)$ é o desvio-padrão de X e denotado por σ_X.

C.4 PROBABILIDADE CONDICIONAL

Se A e B são dois eventos e a probabilidade do evento B, indicado por $P(B)$. Quando $P(B) \neq 0$, pode-se definir a probabilidade condicional de A dado B como:

$$P(A|B) = \frac{P(A \cap B)}{P(B)} \tag{C.4}$$

Da mesma forma, a probabilidade condicional de B dado A é igual a:

$$P(B|A) = \frac{P(B \cap A)}{P(A)} \ \forall \ P(A) \ \neq \ 0 \tag{C.5}$$

Combinando as Equações C.4 e C.5, obtém-se:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \tag{C.6}$$

O que é denominado como regra de Bayes para probabilidade.

C.5 DISTRIBUIÇÃO DE PROBABILIDADE MARGINAL

Dada uma função densidade de probabilidade $f(x, y)$ das variáveis aleatórias contínuas X e Y. A distribuição de probabilidade de X isolado, igual a $g(x)$, e Y isolados, igual a $h(x)$ são:

$$g(x) = \int_{-\infty}^{\infty} f(x, y) d_y \tag{C.7}$$

$$h(y) = \int_{-\infty}^{\infty} f(x, y) d_x \tag{C.8}$$

C.6 DISTRIBUIÇÃO NORMAL

Uma variável aletória contínua X é chamada de gaussiana ou normal quando sua FDP é definida por:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(x-\mu)}{2\sigma^2}} \tag{C.9}$$

onde μ é a média da distribuição e σ^2 a variância da distribuição. Uma variável aleatória x, normalmente distribuída com média μ e variância σ^2, é denotada por $x \sim N(\mu, \sigma^2)$ e sua FDP é a Equação C.9.

A Figura C.1 apresenta o gráfico da função gaussiana para $\mu = 0$ e para $\sigma = 0.5$ em azul, $\sigma = 1$ em vermelho e $\sigma = 2$ em verde.

Figura C.1 - FDP Gaussiana com $\mu = 0$ e diferentes valores de σ

Fonte: Produção do autor.

C.7 DISTRIBUIÇÃO DE CAUCHY

Uma variável aleatória contínua é dita de Cauchy quando sua FDP é definida por:

$$f(x) = \frac{1}{\pi}\left(\frac{\gamma}{(x - x_0)^2 + \gamma^2}\right) \tag{C.10}$$

onde γ é o parâmetro de escala e x_0 o pârametro de localização.

Uma variável aleatória X que segue a distribuição de Cauchy é denotada por $x \sim Cauchy(\gamma, x_0)$ e sua FDP é dada pela Equação C.10. Uma das características da distribuição de Cauchy é possuir média e variância não definidas.

A Figura C.2 apresenta o gráfico da FDP de Cauchy para $x_0 = 0$ e para $\gamma = 0.5$ em azul, $\gamma = 1$ em vermelho e $\gamma = 2$ em verde.

Figura C.2 - FDP de Cauchy com $x_0 = 0$ e diferentes valores de γ

Fonte: Produção do autor.

C.8 DISTRIBUIÇÃO DE LEVY

Uma variável aleatória contínua X é dita de Levy quando sua FDP é definida por:

$$f(x) = \sqrt{\frac{c}{2\pi}} \frac{e^{\frac{-c}{2(x-a)}}}{(x-a)^{\frac{3}{2}}} \tag{C.11}$$

A FDP de Levy (Equação C.11) é definida para $0 \leq x < \infty$, onde c é o parâmetro de escala, são válidos valores de $c > 0$, e a é o parâmetro de localização.

Uma variável aleatória X que segue a distribuição de Levy é denotada por $x \sim Levy(c, a)$ e sua FDP é dada pela Equação C.11. Uma das características da distribuição de Levy é possuir média e variância não definidas.

A Figura C.3 apresenta o gráfico da FDP de Levy para $a = 0$ e para $c = 0.5$ em azul, $c = 1$ em vermelho e $c = 2$ em verde.

Figura C.3 - FDP de Levy com $a = 0$ e diferentes valores de c

Fonte: Produção do autor.

PUBLICAÇÕES TÉCNICO-CIENTÍFICAS EDITADAS PELO INPE

Teses e Dissertações (TDI)

Teses e Dissertações apresentadas nos Cursos de Pós-Graduação do INPE.

Notas Técnico-Científicas (NTC)

Incluem resultados preliminares de pesquisa, descrição de equipamentos, descrição e ou documentação de programas de computador, descrição de sistemas e experimentos, apresentação de testes, dados, atlas, e documentação de projetos de engenharia.

Propostas e Relatórios de Projetos (PRP)

São propostas de projetos técnico-científicos e relatórios de acompanhamento de projetos, atividades e convênios.

Publicações Seriadas

São os seriados técnico-científicos: boletins, periódicos, anuários e anais de eventos (simpósios e congressos). Constam destas publicações o Internacional Standard Serial Number (ISSN), que é um código único e definitivo para identificação de títulos de seriados.

Pré-publicações (PRE)

Todos os artigos publicados em periódicos, anais e como capítulos de livros.

Manuais Técnicos (MAN)

São publicações de caráter técnico que incluem normas, procedimentos, instruções e orientações.

Relatórios de Pesquisa (RPQ)

Reportam resultados ou progressos de pesquisas tanto de natureza técnica quanto científica, cujo nível seja compatível com o de uma publicação em periódico nacional ou internacional.

Publicações Didáticas (PUD)

Incluem apostilas, notas de aula e manuais didáticos.

Programas de Computador (PDC)

São a seqüência de instruções ou códigos, expressos em uma linguagem de programação compilada ou interpretada, a ser executada por um computador para alcançar um determinado objetivo. Aceitam-se tanto programas fonte quanto os executáveis.