Deteção e desvio de obstáculos para veículos aéreos não tripulados usando visão monocular

Rodolfo Barros Chiaramonte

Tese de Doutorado do Programa de Pós-Graduação em Ciências de Computação e Matemática Computacional (PPG-CCMC)

SERVIÇO DE PÓS-GRADUAÇÃO DO ICMC-USP

Data de Depósito:

Assinatura: _____

Rodolfo Barros Chiaramonte

Detecção e desvio de obstáculos para veículos aéreos não tripulados usando visão monocular

Tese apresentada ao Instituto de Ciências Matemáticas e de Computação – ICMC-USP, como parte dos requisitos para obtenção do título de Doutor em Ciências – Ciências de Computação e Matemática Computacional. *VERSÃO REVISADA*

Área de Concentração: Ciências de Computação e Matemática Computacional

Orientadora: Profa. Dra. Kalinka Regina Lucas Jaquie Castelo Branco

Coorientador: Prof. Dr. Fernando Santos Osório

USP – São Carlos
Janeiro de 2019

Ficha catalográfica elaborada pela Biblioteca Prof. Achille Bassi
e Seção Técnica de Informática, ICMC/USP,
com os dados inseridos pelo(a) autor(a)

C532d Chiaramonte, Rodolfo Barros
 Detecção e desvio de obstáculos para veículos
 aéreos não tripulados usando visão monocular /
 Rodolfo Barros Chiaramonte; orientadora Kalinka
 Regina Lucas Jaquie Castelo Branco; coorientador
 Fernando Santos Osório. -- São Carlos, 2019.
 136 p.

 Tese (Doutorado - Programa de Pós-Graduação em
 Ciências de Computação e Matemática Computacional) --
 Instituto de Ciências Matemáticas e de Computação,
 Universidade de São Paulo, 2019.

 1. MOSAIC. 2. Detecção e desvio de obstáculos. 3.
 Visão monocular. 4. Posição tridimensional. 5. ORCA.
 I. Castelo Branco, Kalinka Regina Lucas Jaquie,
 orient. II. Osório, Fernando Santos, coorient. III.
 Título.

Rodolfo Barros Chiaramonte

Obstacle avoidance for UAVs using monocular vision

Doctoral dissertation submitted to the Institute of Mathematics and Computer Sciences – ICMC-USP, in partial fulfillment of the requirements for the degree of the Doctorate Program in Computer Science and Computational Mathematics. *FINAL VERSION*

Concentration Area: Computer Science and Computational Mathematics

Advisor: Profa. Dra. Kalinka Regina Lucas Jaquie Castelo Branco

Co-advisor: Prof. Dr. Fernando Santos Osório

USP – São Carlos
January 2019

Dedico este trabalho à minha avó Olga Trevisan Barros

(in memoriam, ★ 19/02/1919 - † 30/07/2017).

Por todo seu carinho, sua atenção, por seus ensinamentos e memórias compartilhadas.

*Com você compreendi que amor e compreensão são os maiores presentes, e sua sabedoria a
melhor herança que poderia existir.*

Para você todo meu amor, hoje e sempre.

AGRADECIMENTOS

Agradeço a Deus pois sem Ele nada seria possível.

Agradeço à Profa. Dra. Kalinka por ter me apoiado e incentivado por todos estes anos.

Agradeço a meus pais, Alcides e Marta, e à minha irmã, Josiane, pelo estímulo, paciência e compreensão em todos os momentos. Agradeço também a todos os meus familiares.

Agradeço aos amigos e colegas, em especial: Daniel, Emerson, Mariana, Natássya e Rayner pelas sugestões e apoio.

Enfim, agradeço a todos que direta ou indiretamente contribuíram para a conclusão deste trabalho.

*“Por vezes sentimos que aquilo que fazemos não é senão uma gota de água no mar. Mas o mar
seria menor se lhe faltasse uma gota.”
(Madre Teresa de Calcutá)*

RESUMO

CHIARAMONTE, R. B. **Deteção e desvio de obstáculos para veículos aéreos não tripulados usando visão monocular**. 2019. 136 p. Tese (Doutorado em Ciências – Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, 2019.

Veículos autônomos são importantes para a execução de missões dos mais variados tipos, reduzindo riscos aos seres humanos e executando as missões de uma maneira mais eficiente. Neste contexto existem os veículos aéreos não tripulados que são cada vez mais utilizados em missões de vigilância, reconhecimento, resgate, entre outras. Uma das características destes veículos é realizar as missões de maneira autônoma, sem a intervenção de operadores humanos. Desta forma, é necessário que existam formas de detectar aproximações perigosas com outras aeronaves e objetos que possam causar risco de colisão e, consequentemente a perda de ativos de alto valor ou até mesmo vidas humanas e, posteriormente realizar o desvio necessário. Neste cenário foi proposto o MOSAIC, um sistema de detecção e desvio de obstáculos utilizando visão monocular para veículos aéreos de pequeno porte. Para isto, foi desenvolvido um método de estimativa da posição tridimensional dos obstáculos a partir de imagens monoculares e propostas melhorias em algoritmos de detecção. A validação do sistema foi obtida por meio de experimentos simulados e reais sobre cada módulo e os resultados obtidos foram promissores, apresentando um erro de apenas 9,75% em ambientes sem restrições e distâncias de até 20 metros. Com isto, os resultados se mostram melhores que os demais algoritmos encontrados no estado da arte em que o erro é menor que 10% apenas em ambientes controlados e distâncias de até 5 metros.

Palavras-chave: MOSAIC, Detecção e desvio de obstáculos, Visão monocular, Posição tridimensional, ORCA.

ABSTRACT

CHIARAMONTE, R. B. **Obstacle avoidance for UAVs using monocular vision**. 2019. 136 p. Tese (Doutorado em Ciências – Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, 2019.

Autonomous vehicles can be used for different kinds of missions reducing risks to human life and being more efficient. In this context, unmanned aerial vehicles play an important role on surveillance, recognition and rescue missions, among others. Due to the mission nature, these vehicles need to perform actions without human intervention, which requires that dangerous approximations to others aerial vehicles or objects to be detected and properly avoided. This leads to the creation of MOSAIC, an obstacle avoidance system based on monocular vision designed to meet the requirements of miniature air vehicles. A novel approach to estimate obstacle three-dimensional position based on monocular vision was developed and some improvements in the detection algorithm were proposed. The system validation was obtained through simulated and real experiments in which each module could be validated. Promising results were obtained showing an error under 9.75% in unconstrained environments and distance up to 20 meters. This results were better than the algorithms and approaches described in the state of the art where errors are under 10% only on constrained environments and distance up to 5 meters.

Keywords: MOSAIC, Obstacle avoidance, Monocular vision, Three-dimensional position, ORCA.

LISTA DE ILUSTRAÇÕES

Figura 1 – Ilustração do Espaço de Cores HSL e HSV.	43
Figura 2 – Modelo de Cor HSV.	43
Figura 3 – Exemplo da utilização do CMO	46
Figura 4 – Modelo de estimativa de posição para VANTs	50
Figura 5 – Zona de segurança tridimensional	51
Figura 6 – Orientações possíveis de um ciclo-limite circular.	53
Figura 7 – Representação dos semiplanos de $ORCA_{A B}^{\tau}$	55
Figura 8 – Representação de oito agentes e os semi-planos de velocidades permitidas ao agente A	57
Figura 9 – Arquitetura do Sistema MOSAIC	72
Figura 10 – Módulo de <i>Pose Estimation</i>	76
Figura 11 – Módulo de Processamento de Dados do VANT	77
Figura 12 – Módulo de Posição do Obstáculo	79
Figura 13 – Representação considerada para o modelo matemático	79
Figura 14 – Representação do sistema de coordenadas considerado para o VANT	80
Figura 15 – Simulação sendo executada no OASIs	92
Figura 16 – Integração do OASIS ao MOSAIC	93
Figura 17 – VANT 3DR Solo.	96
Figura 18 – Movimentação do VANT para a simulação com Dronekit para validar os métodos de estimativa de posição do obstáculo.	97
Figura 19 – Erro dos algoritmos considerando sensores perfeitos	98
Figura 20 – Erro dos algoritmos considerando valores inteiros para os pontos na imagem	99
Figura 21 – Ambiente de testes de intensidade de sinal (RSS)	101
Figura 22 – Sinal x Distância a 40 Km/h	102
Figura 23 – Sinal x Distância a 80 Km/h	103
Figura 24 – Correlação entre a distância real e a estimada	103
Figura 25 – Ilustração do alerta de colisão em cenário 2D	104
Figura 26 – Resultado da detecção utilizando CMO e Programação dinâmica	104
Figura 27 – VANT detectado com apenas 2x3 <i>pixels</i> (destaque da Figura 26)	105
Figura 28 – Erro dos algoritmos em simulação utilizando o OASIs	107
Figura 29 – Erros obtidos com os métodos a) <i>Yaw</i> , b) <i>Yaw e Pitch</i> , c) <i>Yaw, Pitch e Roll</i> e d) MMQ_N para o experimento realizado com a variação do ângulo de <i>pitch</i> durante a trajetória de voo	108

Figura 30 – Erros obtidos para o experimento realizado com a variação do ângulo de <i>pitch</i> durante a trajetória de voo	109
Figura 31 – Erros obtidos para o experimento realizado com a variação do ângulo de <i>roll</i> durante a trajetória de voo	110
Figura 32 – Experimento 2 - Variação de ângulo de <i>roll</i> (combinado)	111
Figura 33 – Simulação do MOSAIC utilizando o OASIs	112
Figura 34 – Controle do VANT utilizando o ORCA (obstáculo simulado)	113

LISTA DE ALGORITMOS

Algoritmo 1 – Controle de Missão e Navegação com Desvio de Obstáculos - MOSAIC 90

LISTA DE CÓDIGOS-FONTE

Código-fonte 1 – Consulta IEEEExplore e Springer Link	136
Código-fonte 2 – Consulta Web-Of-Science	136
Código-fonte 3 – Consulta Scopus	136

LISTA DE TABELAS

Tabela 1 – PLE em diversos ambientes	50
Tabela 2 – Comparação entre os métodos que utilizam visão monocular	67
Tabela 3 – Média e desvio padrão da intensidade do sinal nas duas velocidades testadas	101
Tabela 4 – Tempo de execução dos algoritmos baseados em CMO por <i>frame</i>	106

LISTA DE ABREVIATURAS E SIGLAS

ANAC	Agência Nacional de Aviação Civil
CAA	<i>Civil Aviation Authority</i>
CMO	<i>Close Minus Open</i>
CMY(K)	<i>Cian, Magenta, Yellow, blacK</i>
DARPA	<i>Defense Advanced Research Projects Agency</i>
DoD	<i>Department of Defense</i>
EASA	<i>European Aviation Safety Agency</i>
EKF	<i>Extended Kalman Filter</i>
ERSG	<i>European RPAS Steering Group</i>
EUA	Estados Unidos da América
FAA	Federal Aviation Administration
fps	<i>frames por segundo</i>
GPS	<i>Global Position System</i>
GPUs	<i>Graphics Processing Units</i>
HALE	<i>High altitude long endurance</i>
HRVO	<i>Hybrid Reciprocal Velocity Obstacle</i>
HSL	<i>Hue, Saturation, Lightness</i>
HSV	<i>Hue, Saturation, Value</i>
IMU	<i>Inertial Measurement Unit</i>
MALE	<i>Medium altitude long endurance</i>
MAV	<i>Miniature Air Vehicle</i>
MOPS	<i>Multi-scale Oriented Patches</i>
MOSAIC	<i>Monocular ObStacle AvoIdanCe</i>
MUAV	<i>Mini UAV</i>
NAV	<i>Nano Air Vehicles</i>
OASIS	<i>Obstacle Avoidance Simulator</i>
ORB	<i>Oriented FAST and Rotated BRIEF</i>
ORCA	<i>Optimal Reciprocal Collision Avoidance</i>
PCA-SIFT	<i>Principal Components Analysis - Scale Invariant Feature Transform</i>
PLE	<i>Path Loss Exponent</i>
PRC	Prevenção Recíproca de Colisões

PTAM	<i>Parallel Tracking And Mapping</i>
RDD	<i>RSS difference detection</i>
RGB	<i>Red, Green, Blue</i>
RLS	<i>Recursive Least Square</i>
RPV	<i>Remotely Piloted Vehicles</i>
RSS	<i>Received Signal Strength</i>
RVO	<i>Reciprocal Velocity Obstacle</i>
SANT	Sistema de Aeronaves Não Tripuladas
SfM	<i>Structure from Motion</i>
SIFT	<i>Scale Invariant Feature Transform</i>
SLAM	<i>Simultaneous Localization And Mapping</i>
SURF	<i>Speeded-Up Robust Features</i>
TUAV	<i>Medium Range or Tactical UAV</i>
UGV	<i>Unamanned Ground Vehicle</i>
UKF	<i>Unscented Kalman Filter</i>
USArmy	<i>United States Army</i>
V2I	<i>Vehicle to Infraestructure</i>
V2V	<i>Vehicle to Vehicle</i>
VANTs	Veículos Aéreos Não Tripulados
VO	<i>Velocity Obstacle</i>
VTOL	<i>Vertical Take-Off and Landing</i>

SUMÁRIO

1	INTRODUÇÃO	29
1.1	Motivação e Caracterização do Problema	32
1.2	Objetivos	35
1.3	Contribuições	35
1.4	Organização do Texto	36
2	FUNDAMENTAÇÃO TEÓRICA E CONCEITUAL	37
2.1	Veículos Aéreos Não Tripulados	37
2.1.1	<i>Composição</i>	38
2.1.2	<i>Classificações dos VANTs</i>	38
2.1.3	<i>Aplicações de VANTs</i>	40
2.2	Processamento Digital de Imagens	41
2.2.1	<i>Modelos de cores</i>	42
2.2.2	<i>Processamento Morfológico</i>	44
2.2.3	<i>Segmentação</i>	45
2.2.4	<i>Detectores/extratores de característica e Técnicas de Correspon-</i> <i>dência de Imagens</i>	47
2.3	Técnicas e Algoritmos para Detecção e Desvio de Obstáculos . . .	48
2.3.1	<i>Distância baseada na intensidade do sinal sem fio</i>	48
2.3.2	<i>Optimal Reciprocal Collision Avoidance - ORCA</i>	51
2.3.2.1	<i>Visão Geral do Algoritmo</i>	55
2.4	Considerações Finais	58
3	DETECÇÃO E DESVIO DE OBSTÁCULOS EM VANTS - ESTADO DA ARTE	59
3.1	Principais Abordagens	59
3.2	Métodos Baseados em SLAM	60
3.3	Métodos Baseados em <i>Monocular Cues</i>	62
3.4	Métodos Baseados em Aprendizado de Máquina	63
3.5	Outros Estudos Focados em Visão Monocular	64
3.6	Considerações Finais	66
4	MOSAIC - <i>MONOCULAR OBSTACLE AVOIDANCE</i>	71
4.1	MOSAIC	71

4.2	Detecção	73
4.2.1	<i>Baseado no Espaço de Cor</i>	74
4.2.2	<i>Close Minus Open (CMO)</i>	75
4.2.3	<i>Oriented FAST and Rotated BRIEF - ORB</i>	75
4.3	<i>Pose Estimation</i>	76
4.4	Processamento de Dados do VANT	77
4.5	Módulo de Posição do Obstáculo	78
4.6	Estimativa da Posição do Obstáculo	78
4.6.1	<i>Inclusão do Ângulo de Pitch</i>	80
4.6.2	<i>Inclusão do Ângulo de roll</i>	84
4.6.3	<i>Melhorias no cálculo da posição do objeto</i>	88
4.7	Controle de Missão e Navegação - MOSAIC	90
4.8	Módulo de Desvio	91
4.9	OASIs - <i>Obstacle Avoidance Simulator</i>	91
4.10	Considerações finais	94
5	RESULTADOS EXPERIMENTAIS	95
5.1	Estimativa de Posição do Obstáculo	96
5.1.1	<i>Simulação com Dronekit</i>	96
5.2	Cálculo de Distância pela Intensidade de Sinal WiFi	100
5.3	Detecção Utilizando CMO e Filtragem Temporal	103
5.3.1	<i>Simulação com OASIs</i>	106
5.3.2	<i>Experimentos Reais</i>	106
5.4	Desvio de Obstáculos com ORCA	111
5.4.1	<i>Desvio de Obstáculo Virtual</i>	111
5.4.2	<i>Considerações finais</i>	112
6	CONCLUSÕES E TRABALHOS FUTUROS	115
6.1	Contribuições	116
6.2	Limitações do MOSAIC	116
6.3	Dificuldades Encontradas	117
6.4	Trabalhos Futuros	118
6.5	Declaração de autoria	118
6.6	Lista de publicações	119
6.6.1	<i>Artigos publicados</i>	119
6.6.2	<i>Artigos submetidos para publicação</i>	119
	REFERÊNCIAS	121

APÊNDICE A	DETECÇÃO E DESVIO DE OBSTÁCULOS USANDO VISÃO MONOCULAR - REVISÃO SISTEMÁTICA . .	133
A.1	Planejamento	133
A.1.1	<i>Objetivos</i>	<i>133</i>
A.1.2	<i>Questão Principal</i>	<i>133</i>
A.1.3	<i>Seleção das Fontes e Idiomas</i>	<i>134</i>
A.1.4	<i>CrITÉrios de Seleção dos Estudos</i>	<i>135</i>

INTRODUÇÃO

Os Veículos Aéreos Não Tripulados (VANTs) são cada vez mais utilizados em aplicações civis e militares incluindo observação, vigilância, reconhecimento, agricultura de precisão, resgate em casos de acidentes, entre outras (SAHA; NATRAJ; WAHARTE, 2014; Trindade Jr.; JORGE; AGUIAR, 2004). A razão principal do interesse em VANTs é o de transferir tarefas mais críticas, que geralmente seriam feitas por humanos, para VANTs, reduzindo assim os riscos aos quais são expostos, além de permitir a execução de missões de maneira mais eficiente (MOON; PRASAD, 2011). Estes veículos são geralmente controlados, gerenciados e monitorados em tempo real por sistemas embarcados, que possuem severas restrições quanto a falhas, uma vez que isto pode acarretar a perda de vidas humanas e/ou perdas de ativos de alto valor. Com isto, tais sistemas são chamados de sistemas embarcados críticos (DUNN, 2003; YI; CAI; YUE, 2008; ARMOUSH; BECKSCHULZE; KOWALEWSKI, 2009; KUMAR; RAMAIAH; KHANAA, 2011; BRANCO, 2012).

Esses veículos possuem uma infraestrutura relacionada ao seu uso, fazendo parte do conceito, além da própria aeronave, a sua carga (*payload*), a estação de controle terrestre e os *links* de comunicação (U.S. GAO, 2008; AUSTIN, 2010). O termo Sistema de Aeronaves Não Tripuladas (SANT) é então utilizado para designar este sistema mais completo, que inclui não apenas as aeronaves, mas todos demais elementos associados.

As aeronaves possuem também diferentes tamanhos, desde micro até grande. Porém, existe uma tendência à pesquisa e ao desenvolvimento de VANTs pequenos, devido a sua grande flexibilidade, variedade de aplicações e baixo custo (VALAVANIS, 2007; BRANCO *et al.*, 2011).

Cada aplicação possui diferentes requisitos, e para supri-los, diferentes configurações de VANTs foram propostas. Uma possível classificação, de acordo com as características da aeronave, é dada por (NONAMI, 2010):

- asas-fixas - exigem uma pista para decolar e aterrissar;

- asas-rotativas - capazes de decolar e planar e aterrizar em ambientes de pequena dimensão. Cabe salientar que podem ser também chamados de *Vertical Take-Off and Landing* (VTOL);
- dirigíveis - flutuam no ar;
- batedores de asas - imitam o movimento de pássaros.

Além das características da aeronave, para o caso de VANTs autônomos, também é importante ressaltar o sistema de controle autônomo, ou piloto automático, que é crucial para o seu bom funcionamento. Sua função é manter a aeronave em condições de voo pré-definidas, executando também tarefas de navegação, como a execução de rotas. Esse controle pode ser dividido em sistema de controle e sistema de navegação, cada um executando tarefas distintas e com laços de controle executados em diferentes frequências (VALAVANIS; VACHTSEVANOS, 2015).

Outra característica presente neste tipo de veículo é que eles podem funcionar por um longo tempo sem a intervenção de um operador humano, podendo realizar a navegação baseada em *grid* ou em uma sequência de *waypoints* (TRINDADE *et al.*, 2010). Durante a operação, estas aeronaves estão sujeitas a ameaças que podem ser tanto obstáculos fixos (paredes, construções, vegetação, entre outros), quanto móveis (outras aeronaves, pássaros, entre outras) e com isto precisam ter a capacidade de detectá-los e realizar manobras evasivas.

No trabalho proposto por Saunders *et al.* (2005) os obstáculos foram divididos em duas categorias: obstáculos conhecidos *a priori* e obstáculos dinâmicos. Os autores apresentam uma solução para *Miniature Air Vehicle* (MAV) – veículos aéreos em miniatura – onde sensores tradicionais são muito pesados para serem instalados. A proposta utiliza um único feixe de *laser* de 2 watts que pode detectar obstáculos em linha reta a até 400 metros de distância. A arquitetura é composta por um *cockpit* virtual que pode ser ligado tanto a aeronave física quanto ao simulador por meio de uma interface comum. A aeronave física é um modelo em miniatura de asa fixa e utiliza o piloto automático Kestrel acoplado ao *laser*. Para realizar a detecção e o desvio de obstáculos foram utilizadas duas classes de algoritmos: uma para atender os obstáculos conhecidos *a priori* e outra para os obstáculos dinâmicos. Isso demonstra a necessidade de viabilizar elementos que permitam que aeronaves de pequeno porte possam fazer uso de sistemas de desvio de obstáculos.

Outro item a ser considerado para a realização de desvio de obstáculos é a capacidade da aeronave em realizar manobras. Moon e Prasad (2011) propõe um *framework* integrado de desvio de obstáculos e proteção de envelope¹, no qual é definido o uso de restrições para manter a proteção do envelope de voo da aeronave durante uma operação de desvio. Com isto, a

¹ Envelope de voo pode ser descrito como as capacidades relacionadas ao *design* da aeronave de realizar manobras considerando suas restrições estruturais e condições atmosféricas.

aproximação a um obstáculo foi classificada em três estados distintos: estado de alerta, estado de desvio seguro, estado de desvio inseguro. Quando uma aeronave detecta um obstáculo muito próximo, o desvio entra em um estado inseguro. Neste estado, a aeronave é desacelerada até entrar em um estado onde seja seguro realizar a manobra. Por outro lado, quando o obstáculo é detectado muito cedo, a aeronave entra em estado de alerta e pode manter a rota prevista por um determinado tempo até atingir o estado de desvio seguro.

Existe um grande número de métodos e algoritmos de visão computacional que podem ser aplicados para o problema de desvio de obstáculos. [Lan et al. \(2013\)](#) propõem um algoritmo para estimar a distância de um obstáculo utilizando fluxo óptico; [Gardeazabal, Ponomaryov e Chairez \(2011\)](#) apresentam um método utilizando *Fuzzy control*; [Souza et al. \(2011\)](#) propõem um algoritmo de navegação baseada em duas Redes Neurais Artificiais distintas e casamento de *templates*; [Linhui et al. \(2009\)](#), [Pantilie e Nedevschi \(2010\)](#) apresentam métodos que envolvem combinação de visão estéreo e fluxo óptico.

[Hrabar et al. \(2005\)](#) demonstraram que uma combinação de métodos que envolve visão estéreo e fluxo óptico apresenta melhores resultados do que o uso de um único método específico de desvio de obstáculos. No entanto, o uso de visão estéreo não é viável para aeronaves menores e com restrições de *payload* e energia.

[Cruz e Encarnação \(2011\)](#) destacam que o problema de desvio de obstáculos para aeronaves de asa fixa difere muito dos demais tipos de aeronaves em termos de plataforma e navegabilidade. Um dos problemas destacados pelos autores é que a aeronave deve se manter sempre acima da velocidade de *stall*. Os autores apresentaram uma proposta baseada em mecânica de fluidos que descrevem a movimentação do fluido na presença de obstáculos. A proposta foi testada por meio de simulação, no entanto, os próprios autores assumem que não existe prova formal de que a aeronave não colidirá com nenhum obstáculo.

Como existe uma diversidade de métodos e algoritmos para realizar a operação de desvio de obstáculos, [Srikanthakumar, Liu e Chen \(2011\)](#) destacam a necessidade de um método de avaliação destes algoritmos antes de serem integrados a aeronaves. Eles afirmam que os algoritmos devem ser verificados sobre todas as condições operacionais e variações que podem ser experimentadas pela aeronave durante seu tempo de vida. O método proposto utiliza métodos de otimização para encontrar os parâmetros de pior caso do algoritmo, onde a distância entre a aeronave e o obstáculo é mínima considerando todas as possibilidades de variação e incerteza.

[Yang, Alvarez e Bruggemann \(2012\)](#) afirmam que o problema de detecção e desvio de obstáculo (ou em inglês: *obstacle avoidance*) pode ser dividido em “*sense*” ou “*detection*” que é perceber a presença do obstáculo; e “*avoidance*” que significa realizar uma manobra evasiva de maneira a evitar o obstáculo e retomar a rota programada antes da manobra. Os autores, assim como os autores de [\(AL-KAFF et al., 2017\)](#), destacam ainda que este problema é um dos mais desafiadores na área de navegação autônoma.

Diversos são os trabalhos encontrados na literatura que tratam da detecção e desvio de obstáculos. Vários desses trabalhos fazem uso de câmeras, entre outros sensores. Entretanto, nenhum trabalho, dentro da literatura aberta e no melhor do conhecimento do autor desta tese, faz a detecção e desvio de obstáculos a partir do uso de uma câmera monocular sem modificações com a eficiência provida nesta tese.

Nesta tese é apresentado o *Monocular ObStacle AvoIdanCe* (MOSAIC). O MOSAIC é um sistema de detecção e desvio de obstáculos que utiliza como base para a detecção imagens monoculares² e dados de posicionamento e atitude³ do VANT em conjunto com uma variação do algoritmo *Optimal Reciprocal Collision Avoidance* (ORCA) 3D para efetuar o desvio. Esse sistema tem como objetivo permitir que veículos aéreos possam efetuar a detecção e o desvio de obstáculos, e com isso serem integrados no espaço aéreo não segregado⁴.

1.1 Motivação e Caracterização do Problema

Com o advento e barateamento de tecnologias e dispositivos eletrônicos e com a miniaturização dos componentes físicos necessários para a construção de aeronaves de pequeno porte, VANTs têm ganhado grande popularidade. Por não possuírem um piloto humano a bordo, essas aeronaves geralmente são menores, podendo ser controladas à distância de forma manual ou assistida (TRINDADE *et al.*, 2010). Tanto a popularização quanto a miniaturização implicam em um equipamento aéreo relativamente barato, leve, compacto, versátil (VALAVANIS; KONTITSIS, 2007), e que também não oferece riscos físicos ao controlador humano (uma vez que o mesmo não se encontra mais dentro do veículo), atraindo um maior interesse militar (SCHNEIDERMAN, 2012).

Essa popularização (ANAC, 2015a) tem levado as agências reguladoras, tais como a Federal Aviation Administration (FAA) nos Estados Unidos da América (EUA), Agência Nacional de Aviação Civil (ANAC) no Brasil e muitas outras de vários países, a regulamentar o uso destas aeronaves no espaço aéreo segregado e não-segregado. Essa regulamentação é efetuada por meio de certificações, procedimentos tecnológicos e operacionais.

Entretanto, é de conhecimento dos profissionais aeronautas, que a certificação de uma aeronave não é trivial e todos os componentes utilizados além do próprio processo de fabricação desta devem passar pelo processo de certificação (ANAC, 2010; ANAC, 2013a; ANAC, 2013b). Além disso, existem *roadmaps* publicados periodicamente por organizações militares e civis (por exemplo, *United States Army* (USArmy), *Department of Defense* (DoD), *European RPAS*

² Entende-se poder imagens monoculares imagens provenientes de um único conjunto de lente e sensor; imagens estereoscópicas são obtidas por conjuntos de duas ou mais lentes com um sensor de imagem para cada lente.

³ Neste contexto, atitude se refere aos ângulos de Euler do VANT (*yaw*, *pitch* e *roll*).

⁴ O espaço aéreo segregado é de uso exclusivo, não compartilhado com outras aeronaves; enquanto que o espaço aéreo não segregado pode ser compartilhado entre diferentes usuários e tipos de aeronaves.

Steering Group (ERSG), ou da FAA) que ilustram os avanços esperados para os SANTs (ARMY, 2010; AUTHORITY, 2012; DOD, 2013; YEARBOOK, 2011). Com base na análise detalhada destes *roadmaps*, foi possível identificar dois desafios relevantes para a integração de SANTs no espaço aéreo não-segregado: a detecção e o desvio de obstáculos. Esses desafios integram o desenvolvimento de novas regulamentações e a formulação de padrões, com destaque para duas áreas funcionais críticas: Capacidade de Detecção e Desvio e Requisitos de Desempenho de Controle e Comunicação. Parte desses desafios podem ser enfrentados por meio da abordagem apresentada nesta tese.

A dificuldade do uso de VANTs em suas diversas aplicações podem ser detalhadas de acordo com os seguintes elementos: (i) a quantidade crescente de VANTs no espaço aéreo segregado e não-segregado; (ii) o aumento do número de diferentes e simultâneas missões a serem realizadas por um ou mais VANTs; (iii) os problemas de segurança do software (*security*) e de segurança física (*safety*) relacionados com a inserção dos VANTs no espaço aéreo; e (iv) a necessidade de revisões das regulamentações e formulação de padrões e desenvolvimento tecnológico que permitam a coexistência de veículos tripulados e não tripulados no mesmo espaço aéreo.

Pesquisas nesta área estão a procura de soluções para: integrar os SANTs no espaço aéreo segregado e não-segregado; garantir *security* e *safety*; prever a auto-separação e, em última instância, a proteção contra colisão entre VANTs e outras aeronaves análogas à operação de detecção e desvio de obstáculos de aeronaves tripuladas que atenda a um nível aceitável de segurança.

Assim, a problemática de prevenção de colisões em VANTs é um tópico de estudo de longa data, mas cujas soluções historicamente tendem a ser desenvolvidas a critério *ad-hoc*, caso a caso, sendo que apenas recentemente métodos robustos de aplicação geral começaram a ser desenvolvidos. Com efeito, tanto em aplicação quanto em teoria, a área possui grande potencial para crescimento. Como o problema de detecção e desvio não é exclusivo de veículos aéreos, este tópico tem sido estudado desde os primórdios da robótica (CAMERON, 1985; ERDMANN; LOZANO-PREZ, 1987), mas geralmente com suposições específicas para veículos terrestres. Alguns desses métodos desenvolvidos para aplicações em solo são aplicáveis para veículos aéreos, mas em geral eles assumem grande controle de manobrabilidade e tempos de resposta relativamente altos (MINGUEZ; MONTANO, 2009). A maioria desses métodos também geram a ocorrência de cenários de impasse ou oscilações (ADOUANE, 2009), que em situações de grande velocidade, que é o caso dos veículos aéreos, podem ser catastróficas.

Na última década, porém, surgiram os primeiros modelos verdadeiramente apropriados para aplicações em VANTs. Em contraste com a geração anterior que procurava algoritmos de movimentação globalmente otimizados, utilizando o conhecimento total do ambiente para planejar um caminho completo (LATOMBE, 1991; RIMON; KODITSCHKEK, 1992), métodos atuais buscam minimizar o trabalho de processamento e apenas considerar a situação momentânea

e dados localizados do veículo e seus vizinhos (ADOUANE, 2009; FIORINI; SHILLER, 1998; Van Berg; LIN; MANOCHA, 2008). Embora seja fácil demonstrar que esse princípio permite desvio consistente, em geral é difícil estudá-lo em conjunto com considerações de convergência para um alvo. Além disso, esses trabalhos fazem uso de diversos dispositivos ou câmeras com características específicas.

Existe ainda a necessidade de se otimizar o código para que o tempo de execução, e consequentemente o consumo de energia, sejam reduzidos, uma vez que esses algoritmos devem ser implementados em sistemas embarcados. Tais sistemas em geral possuem restrições de memória, processamento e bateria, que podem impactar diretamente no desempenho, não fornecendo resultados rápidos o suficiente para permitir uma detecção e um desvio seguro do obstáculo detectado. Cabe ressaltar que diversos são os obstáculos que podem ser detectados, sendo eles fixos ou móveis, o que pode levar a uma necessidade de processamento mais eficaz.

Com base na crescente utilização dos VANTs em aplicações civis, aliada aos desafios e lacunas apresentadas nos *roadmaps* que demonstram uma clara necessidade de integração dos SANTs ao espaço aéreo com a necessidade de padrões de aeronavegabilidade e consequentemente uma melhor estrutura de detecção e desvio de obstáculos, é possível vislumbrar:

Problema: Como é possível aprimorar a detecção dos obstáculos para se atingir a precisão necessária para o desvio de obstáculos?

Para responder a essa questão, a seguinte hipótese foi definida:

Hipótese: A detecção e o desvio de obstáculos podem ser realizados utilizando-se uma câmera monocular para o cálculo da distância e essas estimativas podem ser aprimoradas utilizando-se a fusão de dados do posicionamento obtido pelo *Global Position System* (GPS), pela *Inertial Measurement Unit* (IMU) e das imagens monoculares.

Entretanto, tal hipótese levanta mais questões: i) quais métodos e/ou algoritmos devem ser utilizados para detectar um obstáculo com visão monocular?; ii) qual técnica deve ser utilizada para estimar a posição do obstáculo?; e iii) quais métodos devem ser utilizados para efetuar o desvio?

A partir da literatura é possível listar técnicas baseadas em visão monocular (MERRELL; LEE; BEARD, 2004; LEE *et al.*, 2010; SHAH; JOHNSON, 2009; LEE *et al.*, 2011; SAHA; NATRAJ; WAHARTE, 2014) para a detecção de obstáculos, abordagens que fazem uso de dados inerciais do VANT com dados da posição do obstáculo na imagem para a estimativa de posição do mesmo (SAHA; NATRAJ; WAHARTE, 2014; ROJAS-PEREZ; MARTINEZ-CARRANZA, 2017), e soluções para o desvio de obstáculos (FIORINI; SHILLER, 1998; Van Berg; LIN; MANOCHA, 2008; ADOUANE, 2009; BENZERROUK *et al.*, 2010; BERG *et al.*, 2011).

1.2 Objetivos

O objetivo principal desta tese de doutorado é a definição e a validação de um sistema de detecção e desvio de obstáculos que utiliza como base para a detecção imagens monoculares e dados de posicionamento e atitude do VANT.

O quesito *safety*, no contexto de aeronavegabilidade para a inserção no espaço aéreo segregado e não-segregado é contemplado pelo sistema, pois este deve garantir que a detecção e o desvio sejam realizados rápido o suficiente para que não ocorra colisão entre o VANT e o obstáculo. Esse quesito auxilia na obtenção de certificação dos SANTS para sua inserção no espaço aéreo, além de garantir a capacidade de auto-separação (manterem-se distantes) e de evitar colisões.

Como existem métodos diferentes para atender a esses requisitos, podem ser listados como objetivos secundários a avaliação e seleção das técnicas mais apropriadas para cada uma das etapas em que se divide a detecção de desvio de obstáculos.

1.3 Contribuições

Nesta tese a contribuição mais importante é a definição do sistema de detecção e desvio de obstáculos, denominado MOSAIC, que permite a detecção e o desvio de obstáculos a uma distância segura o suficiente para que não ocorram colisões.

Entretanto, para cumprir todos esses requisitos, outras contribuições foram derivadas:

- Um módulo de cálculo da posição 3D dos obstáculos baseado em dados de localização, IMU, GPS e visão monocular ([CHIARAMONTE; Castelo Branco, 2017](#));
- Método integrado de controle de navegação para evitar obstáculos fixos e outras aeronaves;
- Modificações realizadas no algoritmo ORCA para considerar a aceleração máxima possível para o VANT ([Sá De Mello et al., 2015](#));
- MOSAIC, um sistema para detecção e desvio de obstáculo que é capaz de detectar obstáculos fixos, calcular a posição 3D destes obstáculos e realizar o desvio dos mesmos;
- OASIs, um simulador para avaliação de algoritmos de detecção e desvio de obstáculos que utilizam visão monocular;
- Método de cálculo de posição do obstáculo baseado na intensidade de sinal sem fio ([CHIARAMONTE; BRANCO, 2014](#)).

1.4 Organização do Texto

- No [Capítulo 2](#) são descritos os principais conceitos relacionados a VANTs e SANTs, elucidando suas principais aplicações, bem como os conceitos relacionados a Processamento de Imagens Digitais, necessários para o entendimento desta tese;
- O [Capítulo 3](#) apresenta uma revisão bibliográfica sobre as técnicas, algoritmos e modelos que permitem a detecção de obstáculos e a percepção de sua posição, vislumbrando o estado da arte;
- A proposta do sistema de detecção e desvio de obstáculos é abordado e discutido no [Capítulo 4](#);
- O [Capítulo 5](#) descreve os resultados obtidos, tanto de experimentos simulados quanto reais, para validar o sistema proposto;
- Por fim, o [Capítulo 6](#) traz a conclusão da tese, assim como as produções bibliográficas e proposta para futuros trabalhos.

FUNDAMENTAÇÃO TEÓRICA E CONCEITUAL

Neste capítulo é descrita a fundamentação teórica necessária para o entendimento desta tese de doutorado. É possível observar que, por essa tese tratar de um tema multidisciplinar, esse capítulo apresenta uma diversidade de informações de diferentes áreas de pesquisa. Sendo assim, este capítulo aborda os conceitos fundamentais relacionados a VANTs, ao processamento de imagens, a determinação de distâncias baseadas na intensidade de um sinal sem fio e ao desvio de obstáculos.

2.1 Veículos Aéreos Não Tripulados

Veículos Aéreos Não Tripulados (VANTs) são veículos que possuem a capacidade de voar sem um piloto humano a bordo. Desta forma, eles podem ser pilotados remotamente *Remotely Piloted Vehicles* (RPV) ou autônomos o suficiente para um voo sem auxílio de um operador humano, apenas seguindo um caminho pré-programado (VALAVANIS; OH; PIEGL, 2009).

Esses veículos têm sido utilizados para a execução de diversos tipos de missões como reconhecimento, resgate, localização de alvos, entre outros.

O veículo aéreo é a unidade responsável por carregar os componentes úteis para a execução de missões (*payload*) e também os aparatos necessários para controlar e manter o veículo em funcionamento. Entre estes itens, necessários para o funcionamento e execução da missão, se encontram uma gama de sensores, atuadores, combustível, fonte de alimentação elétrica, entre outros, sendo compostos, desta forma, por diversos componentes.

2.1.1 Composição

Como descrito anteriormente, os VANTs possuem uma série de componentes que auxiliam a executar as missões. Os principais componentes de um VANT são, segundo (AUSTIN, 2011; FAHLSTROM; GLEASON, 2012):

- **A estação de controle:** pode ser definida como o centro de operações e também a interface homem-máquina para os VANTs. Ela normalmente fica localizada em solo, mas pode também estar localizada suspensa no ar e é onde as missões são pré-programadas;
- **O sistema de navegação:** é responsável por fornecer informações de localização da aeronave a qualquer momento. Ele permite que o veículo aéreo tenha uma maior autonomia na execução de missões sem o contato com a estação de controle, ou que possa realizar um retorno emergencial em caso de queda de comunicação. Em geral é no sistema de navegação que encontra-se o piloto automático, que é considerado o cérebro da aeronave, pois é ele que substitui o piloto e que realiza de forma automática as atividades realizadas pelo piloto humano;
- **A carga útil (*payload*):** é normalmente composta por sensores específicos da missão que o VANT deve realizar. Essa carga pode ser um mecanismo de vídeo simples (geralmente até aproximadamente 200g), mecanismos de vídeo com lentes de maior distância focal ou opções de zoom que podem chegar a pesar em torno de 3 a 4 kg ou radares de alta potência pesando até 1000 kg. As informações obtidas por estes sensores podem ser utilizadas pela aeronave para a execução das missões ou para o seu próprio controle, como por exemplo, para a detecção de obstáculos;
- **O sistema de lançamento, recuperação:** Alguns tipos de aeronaves não possuem sistemas tradicionais de decolagem e pouso e necessitam de mecanismos para o lançamento normalmente implementado como uma rampa a partir da qual a aeronave é acelerada até a velocidade necessária e; recuperação implementados na forma de paraquedas ou *airbags* que são utilizados para diminuir e atenuar o impacto;
- **O sistema de comunicação:** geralmente é necessário um sistema de comunicação para permitir o envio de informações da estação de controle (plano de voo, comandos para controle do *payload* entre outros) para a aeronave (*uplink*); ou informações da aeronave (posicionamento da aeronave, dados gerados pela missão, entre outros) para a estação de controle (*downlink*).

2.1.2 Classificações dos VANTs

Os veículos aéreos não tripulados podem ser classificados de acordo com seu porte, alcance e capacidade para realização de missões. Existem diversas classificações, dentre elas as

apresentadas pelo DoD (DOD, 2007; DOD, 2009) dos Estados Unidos, pelo *Safety and Airspace Regulation Group* - Grupo de Segurança e Regulamentação do Espaço Aéreo da *Civil Aviation Authority* (CAA) (CAA, 2015) do Reino Unido, pela Agência Nacional de Aviação Civil (ANAC) (ANAC, 2015b) e pela *European Aviation Safety Agency* (EASA) (EASA, 2015). A classificação utilizada neste trabalho para é a proposta por (AUSTIN, 2010), uma vez que muitos trabalhos publicados na área fazem uso desta classificação uma vez que ela inclui a missão como parâmetro para realizar a classificação. Com isto, faz-se necessário conhecê-la para entender se as propostas discutidas nos trabalhos se aplicam a aeronaves de pequeno, médio ou grande porte. Os termos descritos abaixo descrevem um grande número de sistemas compreendendo desde aeronaves de 35 metros até veículos de apenas 40 milímetros:

- **High altitude long endurance (HALE)** – Mais de 15.000 m de altitude e 24+ horas de autonomia. Esse tipo de aeronave é capaz de realizar tarefas de reconhecimento e vigilância de extremo longo alcance e estão cada vez sendo mais equipadas com armamentos. Eles geralmente são operados pelo pessoal das forças aéreas a partir de estações de controle fixas;
- **Medium altitude long endurance (MALE)** – 5000 - 15.000 m de altitude e 24 h de autonomia. Suas funções são semelhantes às dos sistemas de HALE, mas geralmente operam em áreas um pouco menores, ainda assim excedendo os 500 km e sendo controlados a partir de estações de controle fixas;
- **Medium Range or Tactical UAV (TUAV)** – Com uma faixa de operação na ordem entre 100 e 300 km. Essas aeronaves são menores e operadas por sistemas mais simples do que os usados em aeronaves do tipo HALE ou MALE e são também operadas pelas forças armadas e navais;
- **Close-Range UAV** – usados por batalhões do exército para operações militares/navais e para diversos fins civis. Eles geralmente operam em faixas de até cerca de 100 km e são, provavelmente, os mais usados nos cenários citados, incluindo diversas outras aplicações como reconhecimento, monitoramento, segurança aérea, vigilância, inspeção de linhas de transmissão, pulverização de lavouras, monitoramento de tráfego, entre outros;
- **Mini UAV (MUAV)** – Refere-se a VANTs menores que uma certa massa (ainda não definida), geralmente menor que 20 kg, mas não tão pequenos como um MAV. São aeronaves que permitem a realização de lançamento à mão e de operação a distâncias de até 30 km, aproximadamente. Estes também são usados por grupos de batalha móveis e para diversos fins civis;
- **Miniature Air Vehicle (MAV)** – O MAV foi inicialmente definido como um VANT de envergadura inferior a 150 mm. Esse conceito já foi um pouco modificado, mas o MAV é aplicado principalmente em operações em ambientes urbanos, especialmente em edifícios.

Ele costuma fazer voos a menores velocidades, executando operações tais como pairar no ar e realizar pequenos e rápidos pousos em espaços reduzidos. Geralmente espera-se que estas aeronaves sejam lançadas à mão, o que exige que elas sejam mais leves e, por consequência, elas acabam sendo mais vulneráveis a turbulências atmosféricas;

- **Nano Air Vehicles (NAV)** – Estes são projetados para terem o tamanho de uma semente e utilizados para formação de enxames de modo a serem aplicados para missões de vigilância com uma faixa de alcance extremamente pequena.

2.1.3 Aplicações de VANTs

Como descrito anteriormente, existe um grande número de possibilidades para aplicação de VANTs na execução dos mais diversos tipos de missões, não só na área militar, mas também no âmbito civil. Alguns exemplos de aplicações são listadas por ([AUSTIN, 2011](#); [BEKMEZCI; SAHINGOZ; TEMEL, 2013](#); [MERINO et al., 2006](#)):

- Aquisição de dados e imagens de alvos em áreas inacessíveis por meios terrestres;
- Localização de alvos;
- Rastreamento;
- Mapeamento de vias e construções;
- Cenários de desastres causados pelo homem ou pela natureza;
- Busca e resgate;
- Reforço na aplicação da lei;
- Vigilância e monitoramento de tráfego;
- Identificação de rodovias;
- Inspeção de linhas de transmissão de energia;
- Medições;
- Agricultura de precisão;
- Cinematografia.

Uma vez caracterizadas as missões, cabe salientar que, para que elas possam ser executadas de forma confiável, faz-se necessária a presença de um módulo que permita a aeronave detectar a presença de um obstáculo e desviar do mesmo. Independentemente do tipo de aeronave, seja ela de pequeno, médio ou grande porte, esse tipo de mecanismo torna-se indispensável

quando se optar por uma aeronave mais segura e pela inserção da mesma em um espaço aéreo segregado ou não-segregado. Entretanto, para aeronaves de grande porte é possível agregar sensores maiores e mais pesados que permitem que esses módulos sejam desenvolvidos mais facilmente. Já para aeronaves de médio e pequeno porte o desenvolvimento desse módulo se torna uma atividade mais desafiadora (DOD, 2013).

Desta forma, o foco desta tese de doutorado é desenvolver um sistema para detecção e desvio de obstáculo utilizando visão monocular e que possa atender as necessidades para VANTs do tipo MAV¹.

2.2 Processamento Digital de Imagens

Uma vez que objetiva-se fazer uso de câmera para realizar a detecção de obstáculos, faz-se necessário realizar operações sobre imagens obtidas pela câmera do VANT com a finalidade de reduzir ruídos, destacar regiões, ou extrair descrições sobre pontos específicos da imagem. O resultado destas operações pode ser combinado com outras técnicas para se chegar ao objetivo final, que é destacar possíveis obstáculos na imagem.

Uma imagem pode ser definida como uma função bidimensional na forma $f(x, y)$, onde x e y representam coordenadas no plano da imagem e f em qualquer par (x, y) é chamado de intensidade ou nível de cinza neste ponto (GONZALEZ; WOODS, 2010).

Para imagens do mundo real, $f(x, y)$ é contínua, enquanto que para imagens digitais é necessário realizar os processos de amostragem e quantização para chegar a uma função $f(x, y)$ discreta composta por um conjunto finito de elementos pictóricos, cada um com sua localização e valores específicos. Tais elementos são comumente chamados de *pixels*.

Em geral, uma imagem pode ser representada por uma matriz bidimensional de M por N em que cada *pixel* pode assumir um valor de intensidade no intervalo $[0, L - 1]$ para L níveis discretos de intensidade. Em uma imagem em tons de cinza, o nível L diz respeito a intensidade de luz acromática recebida pelo sensor.

Apesar de muitos algoritmos de processamento de imagem poderem ser aplicados a imagens em tons de cinza, a informação relacionada a cor pode ajudar na resolução de alguns tipos de problemas (GONZALEZ; WOODS, 2010).

Uma vez que as imagens digitais são representadas por meio de matrizes, as operações matriciais são válidas, assim como os operadores lógicos e os aritméticos. Várias são as operações que podem ser realizadas sobre imagens digitais, como por exemplo Histograma de Cores, Filtragem Espacial, Extração de Características, entre outras. São apresentadas nas seções 2.2.1,

¹ Como descrito por Austin (2010), o conceito relacionado ao tamanho do MAV foi um pouco modificado sendo que a literatura tem tratado veículos um pouco maiores, mas com enfoque em operações no ambiente urbano (de curto alcance) e com capacidade de pairar no ar, como sendo MAVs

2.2.2, 2.2.3 e 2.2.4 algumas das operações mais frequentes realizadas com imagens necessárias para o entendimento deste trabalho.

2.2.1 Modelos de cores

Um modelo de cor tem como objetivo padronizar a especificação das cores de uma imagem. Os principais modelos de cores são o *Red, Green, Blue* (RGB), *Cian, Magenta, Yellow, black* (CMY(K)) e *Hue, Saturation, Value* (HSV) (GONZALEZ; WOODS, 2010). Estes modelos utilizam três componentes para descrever uma imagem.

No caso do modelo RGB, inspirado no funcionamento do olho humano pela presença de cones capazes de captar luz nas frequências próximas ao vermelho, verde e azul, as componentes são estas três cores primárias. Este modelo é um modelo aditivo onde a cor a ser representada é uma combinação linear da emissão das cores em níveis correspondentes ao valor de cada componente.

O modelo CMY(K) é um modelo subtrativo e funciona baseado na absorção das cores em níveis correspondentes ao valor de cada componente. Neste modelo, as componentes são as cores primárias subtrativas ciano (C), magenta (M) e amarelo (Y). Este modelo é geralmente utilizado para representar cores para impressão e a cor preta é o resultado da absorção total de todas as cores primárias subtrativas. No entanto, a cor preta resultante desta combinação é turva e os sistemas de impressão incorporaram também um componente de cor preta (K) para resolver este problema.

Os modelos RGB e CMY(K) não são naturais para a interpretação humana. Uma forma mais natural de representação é utilizar as componentes matiz, saturação e brilho (ou luminosidade). Estas componentes formam os modelos HSV e *Hue, Saturation, Lightness* (HSL). As imagens coloridas são, atualmente, em sua maioria adquiridas por meio de um filtro de Bayer. Este consiste de uma matriz de fotossensores específicos. Mais comumente sensores de verde, azul e vermelho mas existem filtros de outras cores. De uma matriz de Bayer é adquirida uma imagem RGB representada computacionalmente como três matrizes, uma para cada componente. Para que a segmentação tenha melhor performance é possível fazer a mudança para o espaço de cor HSV. Nesse espaço a primeira componente define a cor, enquanto as outras duas representam a saturação e a luminescência (GONZALES R.; WOODS, 2002). Na Figura 1² é ilustrado como as cores ficam distribuídas no espaço HSV.

A transformação RGB-HSV é dada pelas equações 2.1, 2.2 e 2.3, para *hue*, *saturation* e *value* respectivamente. No caso do estudo foi usada a biblioteca OpenCV que além dessa conversão faz um ajuste para os valores caberem em um byte (THE..., 2014). Nesse espaço de cores basta filtrar as regiões da componente *hue* quando se está preocupado somente com a cor. Assim, espaço de cores nada mais são que modelos matemáticos utilizados para descrever

² Fonte: http://commons.wikimedia.org/wiki/File:Hsl-hsv_models.svg

Figura 1 – Ilustração do Espaço de Cores HSL e HSV.

Fonte: Adaptada de <http://commons.wikimedia.org/wiki/File:Hsl-hsv_models.svg>

cada cor a partir de fórmulas, uma vez que a cor pode ser expressa pelas características de brilho (intensidade de luz), matiz (cor dominante) e saturação (quantidade de luz branca misturada a matiz). O espaço de cor utilizado nas imagens aéreas que foram utilizadas nesta tese é o RGB adicionado de uma conversão RGB-HSV³ para que o modelo HSV possa ser utilizado permitindo detectar um objeto de uma cor pré-definida.

Figura 2 – Modelo de Cor HSV.

Fonte: Adaptada de François e Medioni (1999).

$$H = \begin{cases} 60 \cdot \frac{(G-B)}{V-\min(R,G,B)} & \text{se } V = R \\ 120 + 60 \cdot \frac{(B-R)}{V-\min(R,G,B)} & \text{se } V = G \\ 240 + 60 \cdot \frac{(R-G)}{V-\min(R,G,B)} & \text{se } V = B \end{cases} \quad (2.1)$$

$$S = \begin{cases} \frac{V-\min(R,G,B)}{V} & \text{se } V \neq 0 \\ 0 & \text{caso contrário} \end{cases} \quad (2.2)$$

³ <https://docs.opencv.org/3.1.0/de/d25/imgproc_color_conversions.html>

$$V = \max(R, G, B) \quad (2.3)$$

O modelo HSV, ilustrado na Figura 2, separa a intensidade (V) dos componentes cromáticos (H,S). No entanto, a representação da cromaticidade no subespaço UV, com base em combinações lineares de canais R, G e B, não é tão intuitiva quanto a representação do subespaço HS radial. A geometria do espaço do HSV torna mais adequado o desenvolvimento de algoritmos que dependem de medições de intensidade e de informações sobre cores, quando disponíveis e relevantes.

2.2.2 Processamento Morfológico

O processamento morfológico ou morfologia matemática é um campo de estudo que soma conhecimentos de basicamente dois subconjuntos da matemática: geometria diferencial e probabilidade geométrica junto com algumas outras áreas relacionadas como análises harmônicas, processos estocásticos e topologia algébrica [Serra \(1982\)](#). Ele pode ser usado como uma ferramenta para extrair componentes úteis na representação e descrição da forma das imagens. Para isto, a morfologia matemática utiliza como instrumento formal a teoria dos conjuntos ([GONZALEZ; WOODS, 2010](#)).

A morfologia é uma modelagem destinada à descrição da forma de um objeto digital retratando as imagens como conjuntos de pontos em um espaço, no sentido matemático.

As operações são realizadas sobre pequenos conjuntos ou sub-imagens chamados de elementos estruturantes. As operações de reflexão ([Equação 2.4](#)) e translação ([Equação 2.5](#)) sobre este conjunto são utilizadas para compor as primitivas básicas do processamento morfológico.

$$\hat{B} = \{w | w = -b, \text{ para } b \in B\} \quad (2.4)$$

$$(B)_z = \{c | c = b + z, \text{ para } b \in B\} \quad (2.5)$$

As primitivas básicas da morfologia matemática são a erosão e a dilatação. A erosão ([Equação 2.6](#)) pode ser descrita como uma operação onde detalhes menores que o elemento estruturante são removidos da imagem resultando no afinamento dos objetos em uma imagem binária. Já a operação de dilatação ([Equação 2.7](#)) realiza o efeito inverso, expandindo componentes em uma imagem binária. Nas Equações 2.6 e 2.7 A é a imagem original e B é o elemento estruturante.

$$A \ominus B = \{z | (B)_z \cap A^C = \emptyset\} \quad (2.6)$$

$$A \oplus B = \{z \mid (\hat{B})_z \cap A \neq \emptyset\} \quad (2.7)$$

Com base nestas primitivas, outras operações podem ser descritas como a abertura e o fechamento morfológico de imagens. A abertura pode ser descrita como uma erosão seguida de uma dilatação, o que faz com que os contornos dos objetos sejam suavizados e pequenas linhas são removidas (Equação 2.8). O fechamento, por sua vez, é uma dilatação seguida por uma erosão, o que faz com que os contornos dos objetos sejam suavizados e as discontinuidades estreitas sejam fundidas ao mesmo tempo que pequenas falhas são preenchidas (Equação 2.9).

$$A \circ B = (A \ominus B) \oplus B \quad (2.8)$$

$$A \bullet B = (A \oplus B) \ominus B \quad (2.9)$$

Para a tarefa de detecção de obstáculos uma operação muito utilizada é a *Close Minus Open* (CMO) que consiste em realizar a subtração entre o resultado do fechamento e o resultado da abertura como definido na Equação 2.10 (CARNIE; WALKER; CORKE, 2006; MEJIAS *et al.*, 2010; LAI; MEJIAS; FORD, 2011; MOLLOY; FORD; MEJIAS, 2014; LYU *et al.*, 2015; ZARANDY *et al.*, 2015).

$$CMO(A, B) = (A \bullet B) - (A \circ B) \quad (2.10)$$

A Figura 3 ilustra o resultado das operações morfológicas de abertura (3c) e fechamento (3d) de uma imagem contendo um obstáculo (3a) por um elemento estruturante do tipo cruz de tamanho 5 (3b). Também é possível observar o resultado da operação de CMO (3e — o *gamma* foi ajustado para 0.7 para melhorar a visualização do resultado) bem como uma limiarização destacando a detecção de um obstáculo na imagem (3f).

É importante destacar que o exemplo apresentado não contém outros artefatos que poderiam causar falsos positivos e que o valor do limiar foi escolhido empiricamente de forma a destacar apenas o obstáculo. Para tratar destes aspectos são utilizadas abordagens com algum tipo de filtragem temporal, onde os sinais que permanecem em diversos *frames* consecutivos são reforçados (CARNIE; WALKER; CORKE, 2006; LAI; MEJIAS; FORD, 2011).

2.2.3 Segmentação

A segmentação de imagens é um dos sub-tópicos dentro do processamento de imagem. A mesma consiste em conseguir extrair da imagem informações relevantes, além das inicialmente adquiridas pelo processo de captura (FORSYTH; PONCE, 2003), separando a imagem em regiões diferentes usando uma característica como critério. Esses dados retirados da imagem irão

Figura 3 – Exemplo da utilização do CMO

Fonte: Elaborada pelo autor.

compor uma nova representação da imagem, em mais alto nível, com a finalidade de conseguir melhores resultados em questões de acurácia e performance. Dentro da segmentação de imagens existem muitas técnicas conhecidas como a detecção de bordas e círculos, segmentação de cores, reconhecimento de padrões entre muitas outras. Sendo assim, esta técnica consiste na subdivisão da imagem em partes. Os algoritmos de segmentação permitem a identificação de diferenças entre objetos, discriminando-os em partes. Sendo assim, alguns operadores são definidos:

- **Limiarização (*Thresholding*)** – consiste em um operador fundamental na identificação de objetos fundamentado na similaridade de níveis de cinza, de modo a extrair objetos de interesse mediante a definição de um limiar τ que separa os agrupamentos de níveis de

cinza. Onde: $T(x,y) = \begin{cases} 0, & \text{se } f(x,y) < \tau \\ 1, & \text{se } f(x,y) \geq \tau \end{cases}$. Nesse caso, cada *pixel* é comparado com o valor limiar τ , e, em sendo menor que o limiar a imagem resultante $T(x,y)$ é rotulada como zero (preto), caso contrário com o valor um (branco). A dificuldade desta abordagem reside em encontrar um limiar adequado. Por meio da análise do histograma da imagem é possível estabelecer um valor para o limiar na região situada entre picos que caracterizam regiões de interesse na imagem.

- **Segmentação Orientada a Regiões** - Baseia-se na similaridade dos níveis de cinza da imagem. Os métodos de segmentação baseados em região visam reunir em um mesmo conjunto *pixels* adjacentes que atendem a um dado critério de heterogeneidade. Assim, regiões da imagem são agrupadas ou divididas dependendo de seus *pixels* terem ou não características semelhantes em termos de cor, textura ou forma. Dentre os métodos desta categoria destacam-se as técnicas de crescimento de região (*region growing*), divisão e união de região (*split and merge*) e divisor de águas (*watershed*). Na técnica de crescimento de região, a abordagem mais simples utilizada é iniciar com um conjunto de *pixels* denominados sementes, e a partir deles crescer as regiões anexando a cada ponto semente outros *pixels* que possuam propriedades similares. O crescimento de região agrupa um conjunto de *pixels* em regiões maiores, enquanto que a agregação de *pixels* fundamenta-se na definição de uma semente (ou um conjunto de pontos similares). É a partir desta semente que a região cresce com a agregação de cada *pixel* à semente.
- **Segmentação Baseada em Bordas** - Várias são as técnicas de segmentação que baseiam-se na detecção de bordas por meio do uso de operadores de gradiente (tais como Sobel, Roberts, Laplaciano) seguidos por um processo de limiarização. A detecção de bordas permite a análise de descontinuidade nos níveis de cinza que são úteis para segmentação.

2.2.4 Detectores/extratores de característica e Técnicas de Correspondência de Imagens

São encontradas na literatura diversas técnicas que permitem que a correspondência entre pontos de imagens possa ser realizada. As principais técnicas encontradas na literatura são: a transformada *Scale Invariant Feature Transform* (SIFT), que é a técnica mais amplamente utilizada para a extração de características de imagens (BEI; HAI ZHEN, 2009) (GENG *et al.*, 2010) apesar do *Speeded-Up Robust Features* (SURF) ter se destacado em alguns trabalhos por apresentar menor custo computacional (HONG *et al.*, 2009). Outra técnica é a *Principal Components Analysis - Scale Invariant Feature Transform* (PCA-SIFT), apresentada em (JUAN; GWUN, 2009), que apresenta uma comparação entre esta técnica e as técnicas SIFT, SURF, demonstrando que apesar de maior custo computacional, a transformada SIFT ainda apresenta melhor estabilidade quanto à invariância a escala e rotação quando comparada às demais. Por fim, outra abordagem, mais recente, apresentada por (RUBLEE *et al.*, 2011a) e denominada

Oriented FAST and Rotated BRIEF (ORB), é apresentada como alternativa mais eficiente que SIFT e SURF, sendo caracterizado como descritor com uma boa abordagem principalmente para ambientes embarcados.

O ORB (RUBLEE *et al.*, 2011a) foi desenvolvido com foco em sistemas que necessitam de descritores e detectores de baixo custo e bom desempenho. Sua base é o detector FAST (ROSTEN; DRUMMOND, 2006) e no descritor BRIEF (CALONDER *et al.*, 2010).

Famosos por sua eficiência em encontrar pontos chaves razoáveis, FAST e suas variantes tem sido utilizado em muitos sistemas de tempo real. O BRIEF, por outro lado, usa apenas um parâmetro, o limiar de intensidade entre o *pixel* central e seus *pixels* vizinhos em um anel circular. O FAST possui limitações, como por exemplo não incluir um operador de orientação e fornecer respostas grandes ao longo das bordas.

Por outro lado, o BRIEF é um descritor que faz uso de testes binários simples entre *pixels* em uma imagem plana. Possuindo desempenho similar ao SIFT, ele inclui robustez quando há luminosidade, e distorções de perspectivas.

Para lidar com o problema de rotação o ORB faz uso do centroide de intensidade para calcular uma orientação para cada canto do FAST e então adota a resposta do canto de Harris para remover os pontos ao longo das bordas. Além disso, uma vez que os cantos do FAST não produzem recursos multi-escala, são extraídos características do ORB em cinco imagens escaladas separadamente, com um fator de escala de $\sqrt{2}$. A extração das características com o ORB é mais rápida que o SURF e o SIFT, sendo que pode ser utilizado em dispositivos embarcados, como por exemplo VANTs (RUBLEE *et al.*, 2011a; ROCHA *et al.*, 2015).

2.3 Técnicas e Algoritmos para Detecção e Desvio de Obstáculos

Nas missões reais, existem muitos obstáculos que se tornam perigosos para VANTs e devem ser evitados. Em alguns casos, o operador humano pode recuperar o controle do VANT, em caso de voos assistidos, mas em voos completamente autônomos, muitas vezes essa abordagem é algo impossível. Existem algumas restrições físicas, por exemplo, distância e outros problemas, fazendo com que não haja tempo suficiente para responder aos controles. Essas restrições exigem um sistema de detecção e prevenção de colisão autônomo. Algumas técnicas e algoritmos são utilizados para isso.

2.3.1 Distância baseada na intensidade do sinal sem fio

Veículos autônomos possuem grande necessidade de comunicação com outros veículos (*Vehicle to Vehicle* (V2V)), e também com estações base (*Vehicle to Infrastructure* (V2I)). Deste modo, os veículos autônomos geralmente possuem interfaces de comunicação sem fio que podem

fornecer dados sobre a intensidade do sinal *Received Signal Strength* (RSS) captado de outros veículos. Esta informação pode ser utilizada no contexto de detecção e desvio de obstáculos por meio da inferência da posição relativa da aeronave ou da detecção de aproximações perigosas, quando a intensidade do sinal se torna muito alta.

Neste contexto, existem vários algoritmos para estimar a localização por meio da intensidade do sinal recebido (CYPRIANI; CANALDA; SPIES, 2012; CHOI *et al.*, 2012; CHENG *et al.*, 2012; YU; FEI; GENG, 2013; MASIERO *et al.*, 2015; PRAATS *et al.*, 2015). A maioria das pesquisas utilizam pontos de acesso fixos no ambiente para calcular a posição dos dispositivos móveis por meio da intensidade do sinal proveniente destes pontos de acesso de posição previamente conhecidas. Desta forma, pode ser realizada a triangulação para calcular a posição estimada do veículo e evitar que eles cheguem muito perto (LUO *et al.*, 2013).

Para calcular a distância das estações base, geralmente os sistemas de localização utilizam duas fases: (i) mapeamento dos pontos de acesso presentes no ambiente; (ii) cálculo da posição estimada utilizando o mapa previamente criado (ZÀRUBA *et al.*, 2006).

A precisão da estimativa da localização é influenciada por diversos fatores como interferência de outros sinais, número de pontos de acesso utilizados como base de localização, entre outros. Com isto, existem muitas pesquisas que visam melhorar a precisão utilizando outras informações além do valor do RSS. Uma solução apresentada por Davies *et al.* (2011) considera o uso do tempo de voo do sinal como uma forma de refinar a medida de posição.

Outra possível abordagem descrita por Zàruba *et al.* (2006) destaca o uso de uma infraestrutura de rede sem fios reduzida. Com isto seria possível realizar o rastreamento do dispositivo móvel com apenas uma estação fixa. Para atingir esse objetivo, o sinal sem fio é tratado com algoritmos de propagação de partículas, mapeando as reflexões e refrações que o sinal sofre no ambiente. Existem também algoritmos específicos para melhorar a estimativa da distância entre um nó sem fio e uma estação base, tal como proposto por Mehra e Singh (2013), onde filtros adaptativos são usados para reduzir a oscilação nos níveis de sinal que ocorre mesmo quando a unidade móvel não tem sua posição alterada.

Ishii e Sato (2013) afirmam que os pontos-chave para usar o RSS como uma medida de distância são a calibração do RSS e a detecção de diferença de RSS, o *RSS difference detection* (RDD). Eles propõem uma maneira de usar calibração RSS e RDD sem medição prévia. Seus resultados mostram a detecção da aproximação entre *hosts* não dotados de GPS ocorrem de maneira eficiente.

Mehra e Singh (2013) propõem o uso do algoritmo *Recursive Least Square* (RLS) para reduzir o erro da estimativa de distância. Eles usaram a Equação 2.11 como base da medida de distância e um filtro adaptativo para remover valores de ruído.

$$d = 10 \left[\frac{(RSSI_{dBm} - A)}{10\eta} \right] \quad (2.11)$$

Na [Equação 2.11](#), $RSSI$ é o indicador de intensidade do sinal em dBm , A é o valor do sinal medido a partir de um nó de referência a um metro de distância e η é o expoente de perda do sinal (*Path Loss Exponent* (PLE)).

[Chen et al. \(2012\)](#) afirmam que o PLE é outro ponto-chave nos algoritmos de estimativa de distância. O ambiente de propagação influencia a taxa na qual o RSS diminui com a distância e esta taxa é medida pelo expoente perda de sinal. Desta forma, este expoente deve ser considerado nas equações de estimativa de distância a partir da intensidade do sinal.

$$\eta = \frac{RSSI_{dBm} - A}{10 \log(d)} \quad (2.12)$$

O expoente de perda de sinal pode ser empiricamente calculado a partir de medidas reais por meio da [Equação 2.12](#), onde $RSSI$ é o indicador de intensidade do sinal recebido, A é o valor medido a partir do nó de referência a uma distância de um metro e d é a distância. A [Tabela 1](#) mostra alguns valores de PLE pré-calculados ([MEHRA; SINGH, 2013](#)).

Tabela 1 – PLE em diversos ambientes

Ambiente	<i>Path Loss Exponent</i> (η)
Espaço aberto	2.0
Área urbana – sinal de celular	2.7~3.5
Ambiente Interno (Linha de visão)	1.6~1.8
Ambiente Interno (obstruído)	4~6
Área urbana (sombra) – sinal de celular	3~5

Fonte: [Mehra e Singh \(2013\)](#).

Figura 4 – Modelo de estimativa de posição para VANTs

Fonte: [Luo et al. \(2013\)](#).

A pesquisa apresentada por [Luo et al. \(2013\)](#) propõe a utilização de várias estações de base com localização conhecida anteriormente (usadas na comunicação dos veículos com

infraestrutura de rede - V2I) que fornecem dados para calcular a posição estimada e a trajetória dos VANTs. Baseando no cálculo de posicionamento e trajetória, é possível prever possíveis colisões. Isto é feito por meio da triangulação da intensidade do sinal que atinge o VANT a partir das estações de base (Figura 4).

Figura 5 – Zona de segurança tridimensional

Fonte: Luo *et al.* (2013).

Uma vez que toda a transmissão sem fio é sujeita a interferências de fatores externos, Luo *et al.* (2013) propuseram modelar o ruído e utilizar o filtro de predição de Kalman (KALMAN, 1960) a fim de melhorar a precisão dos cálculos e obter um sistema mais confiável. Eles também propõem o uso de um raio de segurança para evitar a aproximação perigosa entre duas aeronaves. O raio da zona de segurança em perfeitas condições é definido pela velocidade relativa máxima e tempo de reação. A imprecisão causada por medidas de RSS também foi considerada em seu algoritmo. A zona de segurança determinada por um raio $S_{i,k}$ é ilustrada na Figura 5.

Considerando que a intensidade do sinal sem fio pode ajudar a calcular a estimativa de posição dos obstáculos, esta tese prevê um módulo de fusão desta informação com outros sensores visando o refinamento desta estimativa. Tal módulo é apresentado no Capítulo 4 e os resultados avaliando a efetividade do cálculo de distância baseado no sinal sem fio considerando diferentes velocidades é apresentado no Capítulo 5.

2.3.2 Optimal Reciprocal Collision Avoidance - ORCA

Como destacado anteriormente, o problema de detecção e desvio de obstáculos é um dos mais importantes e desafiadores na área de robótica móvel, e quando se pensa em robótica aérea esse problema se torna ainda mais desafiador. Sendo assim, ele pode ser dividido em dois subproblemas: o de *sense*, ou detecção, que é voltado ao estudo de técnicas para detectar possíveis obstáculos e; o de *avoidance*, desvio, que é relacionado a execução de manobras

evasivas para evitar possíveis colisões. Sendo assim, em geral o objetivo do desvio de obstáculos é o de evitar colisões.

As abordagens de desvio de obstáculos podem ser classificadas em cooperativas ou não cooperativas. Nas abordagens cooperativas, existe alguma forma de comunicação entre os elementos envolvidos de forma a coordenar suas posições para evitar colisões entre si. Quando associadas a veículos aéreos, estas abordagens são utilizadas geralmente em estudos relacionados ao voo em formação (LIE; GO, 2010; BAYHA; GRÜNEIS; SCHÄTZ, 2012; GONCALVES *et al.*, 2013; JESUS *et al.*, 2013).

Nas abordagens não cooperativas, o desvio de obstáculos é realizado por meio da obtenção de informações oriundas de sensores em conjunto com algoritmos para não só detectar mas também extrair informações dos obstáculos, uma vez que não há comunicação entre os elementos envolvidos. No contexto de veículos aéreos de pequeno porte existem limitações de quais os tipos de sensores que podem ser embarcados.

Para realização do desvio, existem trabalhos que abordam o controle e estabilização da aeronave durante a manobra de desvio e em condições meteorológicas severas (YANG; ALVAREZ; BRUGGEMANN, 2012). Outros abordam a geração e atualização de caminhos e trajetórias com a presença de obstáculos. Neste contexto, caminhos podem ser definidos como uma sequência de pontos por onde o veículo deve passar. Na trajetória, os pontos por onde o veículo deve passar são parametrizados pelo tempo, considerando as restrições de movimentação da aeronave (NETO; MACHARET; CAMPOS, 2009).

Em geral, estes esforços têm como objetivo permitir a integração dos VANTs no espaço aéreo, como já previamente mencionado, focando na aeronavegabilidade e na segurança desses veículos (TIRRI *et al.*, 2012).

O problema de desvio e prevenção de colisão de obstáculos não é recente, e vem sendo estudado desde o início da robótica (CAMERON, 1985; ERDMANN; LOZANO-PREZ, 1987) principalmente focado na robótica terrestre (BORENSTEIN; KOREN, 1990; BORENSTEIN; KOREN, 1991; THRUN *et al.*, 1996).

Na última década surgiram os primeiros modelos verdadeiramente apropriados para aplicações em VANTs. Em contraste com a geração anterior que procurava algoritmos de movimentação globalmente otimizados, utilizando o conhecimento total do ambiente para planejar um caminho completo (LATOMBE, 1991; RIMON; KODITSCHKE, 1992), métodos atuais buscam minimizar o trabalho de processamento e apenas considerar a situação momentânea e dados locais do veículo e seus vizinhos (FIORINI; SHILLER, 1998; Van Berg; LIN; MANOCHA, 2008; ADOUANE, 2009). Embora seja fácil demonstrar que esse princípio permite desvio consistente, em geral é difícil estudá-lo levando-se em consideração a necessidade de convergência para um alvo (ou destino).

Métodos reativos que apenas lidam com obstáculos obtidos por sensores locais não são

novidade (KHATIB, 1985), mas somente com a introdução do conceito de ciclos-limite (KIM; KIM, 2003) (Figura 6) eles se revelaram mais eficientes e robustos do que os métodos anteriores. Em (JIE *et al.*, 2006), o método foi aprimorado com o uso de campos vetoriais, mas somente em (ADOUANE, 2009) foi introduzido o conceito de movimento chaveado para a consolidação de um sistema de atração/repulsão útil para navegação formacional.

Figura 6 – Orientações possíveis de um ciclo-limite circular.

Fonte: Adouane (2009).

Esse modelo chaveado apresenta diversas limitações intrínsecas às discontinuidades e incertezas geradas no período de transição entre os controladores de atração e repulsão. Em (BENZERROUK; ADOUANE; MARTINET, 2010) foi apresentada uma adaptação que torna o sistema assintoticamente estável, mas só há garantias de zero risco de colisão caso uma função penalidade também seja introduzida (BENZERROUK; ADOUANE; MARTINET, 2012). Neste caso, é possível que o veículo seja totalmente parado a fim de evitar colisões.

Uma solução popular, frequentemente aplicada para robôs de motor diferencial ou similares, é utilizar um chaveamento de controladores de movimento (ADOUANE, 2009; BENZERROUK *et al.*, 2010), de modo que em um determinado momento ou o robô está se dirigindo para um alvo (momento de atração) ou desviando de um obstáculo (momento de repulsão). Esse chaveamento atração/repulsão é simples conceitualmente e em geral é fácil construir controladores comprovadamente estáveis isoladamente, mas não é trivial construir um sistema intrinsecamente estável nos momentos de troca entre eles – e de fato, soluções oferecidas assumem certa liberdade para instabilidade momentânea (BENZERROUK; ADOUANE; MARTINET, 2010).

Uma elaboração alternativa, evoluída do modelo *Velocity Obstacle* (VO) (FIORINI; SHILLER, 1998), desenvolvida recentemente com apoio de fundos da agência de defesa dos Estados Unidos *Defense Advanced Research Projects Agency* (DARPA), permite superar a

liberdade para instabilidades, suplantando essa limitação.

Esse modelo, assim como os anteriores, considera um ciclo frequente de estimativas e decisões; nesse ciclo, porém, a velocidade dos obstáculos em torno do veículo são estimadas, e uma velocidade próxima da desejada, mas que não coloque o veículo em uma trajetória de colisão, é dada como saída. Mas como esse processo de desvio atua diretamente sobre um fluxo prévio de velocidades idealizadas, e não de forma independente como em um movimento chaveado, não há instabilidades e descontinuidades, e se o fluxo de velocidades de entrada definir uma trajetória convergente para um ponto, o fluxo modificado também irá.

Esse modelo foi então aprimorado levando em consideração que o obstáculo pode ser reativo, ou seja, que ele pode reagir à velocidade detectada do veículo e reciprocamente iniciar um protocolo de desvio, dando origem ao modelo *Reciprocal Velocity Obstacle* (RVO) (Van Berg; LIN; MANOCHA, 2008).

Algumas melhorias subsequentes foram feitas sobre esse último, como o *Hybrid Reciprocal Velocity Obstacle* (HRVO) (SNAPE *et al.*, 2011; SNAPE *et al.*, 2009), e eventualmente foi construído o algoritmo *Optimal Reciprocal Collision Avoidance* (ORCA) (BERG *et al.*, 2011), que sob condições ideais de cooperação mútua entre os agentes, conhecimento total e perfeito do estado do sistema e para veículos holonômicos⁴ garante um cenário completamente livre de colisões, com o menor ajuste de velocidade possível (Figura 7). Devido à rapidez, à robustez e à característica de produzir mínimas alterações na trajetória, esse algoritmo se apresenta especialmente adequado a aplicações aéreas.

Embora desenvolvido especialmente para Prevenção Recíproca de Colisões (PRC) (Van Berg; LIN; MANOCHA, 2008), o algoritmo ORCA (BERG *et al.*, 2011) foi demonstrado robusto o suficiente para situações genéricas de desvio de obstáculos (CONROY; BAREISS; BEALL, 2014).

O diferencial do ORCA é que, ao invés de atuar em um estado isolado de pura repulsão, ele trabalha aplicando uma correção mínima na velocidade idealizada do veículo para que este percorra um trajeto seguro, sem colisões. Na prática, o efeito que esta elaboração tem é que o controle de prevenção de colisão é feito na forma de um filtro suave e contínuo (SNAPE *et al.*, 2010), de mínimo impacto no percurso. Em situações em que a manobrabilidade é importante, como ocorre em VANTs, essa propriedade é a ideal.

Desde a formulação do ORCA, alguns trabalhos (SNAPE; MANOCHA, 2010; CONROY; BAREISS; BEALL, 2014) exploraram o algoritmo aplicado para navegação aérea, mas até o presente momento todos os trabalhos conhecidos abordam controladores humanos, ou seja, o uso do mesmo como um meio de auxílio ao controlador humano; simulações (ZOU; ALEXANDER; MCDERMID, 2016; HO *et al.*, 2018) e trabalhos *indoor* (ALEJO *et al.*, 2014; ZEYU *et al.*, 2016).

⁴ Veículos holonômicos são aqueles que podem alterar sua velocidade instantaneamente.

Figura 7 – O conjunto $ORCA_{A|B}^\tau$ de velocidades permitidas a um veículo A para PRC ótima com um veículo B é um semiplano delimitado pela linha perpendicular a $\mathbf{u}$ pelo ponto $\mathbf{v}_A^{\text{opt}} + \frac{1}{2}\mathbf{u}$, onde $\mathbf{u}$ é o vetor de $\mathbf{v}_A^{\text{opt}} - \mathbf{v}_B^{\text{opt}}$ ao ponto mais próximo da fronteira de $VO_{A|B}^\tau$ (o conjunto de velocidades de A que levam a colisões com B).

Fonte: Berg *et al.* (2011).

O algoritmo ORCA pode ser empregado em esquemas de controle *on-line* e *offline*. O controle *on-line* precisa que o VANT adquira as posições e velocidades exatas de seus veículos vizinhos. No entanto, as informações de posição e velocidade fornecidas pelos sinais GPS são imprecisas em aplicações realizadas em campo (externas). Aplicações *outdoor* encontram maiores problemas do a execução de experimentos simulados, uma vez que distúrbios externos como vento tornam experimentos reais mais desafiadores.

O ORCA ignora a dinâmica dos veículos móveis, de modo que não considera restrições de velocidade e aceleração do veículo.

2.3.2.1 Visão Geral do Algoritmo

Considerando dois agentes a_i e a_j , localizados nas posições $\mathbf{p}_i$ e $\mathbf{p}_j$, e com raios r_i e r_j , é possível definir que os agentes estão em rota de colisão no momento τ se, assumindo que ambas as velocidades se mantenham constantes, uma colisão ocorrerá dentro do intervalo de tempo $[\tau, \tau + \Delta]$.

Define-se $D(\mathbf{p}, r)$ para um disco aberto de raio r centrado em $\mathbf{p}$:

$$D(\mathbf{p}, r) = \{\mathbf{q} \mid \|\mathbf{q} - \mathbf{p}\| < r\}. \quad (2.13)$$

Então $VO_{i|j}^\tau$, que é o *Velocity Obstacle* para i induzido por j com o horizonte de tempo

Δ no momento τ , pode ser definido como:

$$\text{VO}_{i|j}^\tau = \{\mathbf{v} \mid \exists t \in [0, \Delta] :: t\mathbf{v} \in D(\mathbf{p}_j - \mathbf{p}_i, r_j + r_i)\}. \quad (2.14)$$

Tomando $\mathbf{v}_i^{\text{new}}$ e $\mathbf{v}_j^{\text{new}}$ como as novas velocidades em τ dos agentes i e j , tem-se que a fim de se obter uma situação livre de colisões a velocidade relativa entre eles, $\mathbf{v}_j^{\text{new}} - \mathbf{v}_i^{\text{new}}$ ou equivalentemente $\mathbf{v}_i^{\text{new}} - \mathbf{v}_j^{\text{new}}$, deve estar fora de $\text{VO}_{i|j}^\tau$ ou $\text{VO}_{j|i}^\tau$, respectivamente. Existem diversos pares $(\mathbf{v}_i^{\text{new}}, \mathbf{v}_j^{\text{new}})$ que satisfazem essa condição, mas deseja-se minimizar o erro até as velocidades preferidas $(\mathbf{v}_i^{\text{pref}}, \mathbf{v}_j^{\text{pref}})$, de forma a desviar cada agente o menos possível de seu trajeto. Considera-se $\mathbf{u}$ o vetor que conecta a velocidade preferencial relativa $\mathbf{v}_i^{\text{pref}} - \mathbf{v}_j^{\text{pref}} = \mathbf{v}_{ij}^{\text{pref}}$ à fronteira de $\text{VO}_{i|j}^\tau$:

$$\mathbf{u} = \underset{\mathbf{v} \notin \text{VO}_{i|j}^\tau}{\text{argmim}} \|\mathbf{v} - \mathbf{v}_{ij}^{\text{pref}}\| - \mathbf{v}_{ij}^{\text{pref}} \quad (2.15)$$

Obtém-se que, em um sistema colaborativo, cada agente recebe metade da responsabilidade de reação, resultando em $\mathbf{v}_i^{\text{new}} = \mathbf{v}_i^{\text{pref}} + \frac{1}{2}\mathbf{u}$ e $\mathbf{v}_j^{\text{new}} = \mathbf{v}_j^{\text{pref}} - \frac{1}{2}\mathbf{u}$. Isso é suficiente quando há apenas dois agentes, mas agentes adicionais criam restrições adicionais que precisam ser consideradas no cálculo de $\mathbf{v}_*^{\text{new}}$.

Para esse fim, define-se $\text{ORCA}_{i|j}^\tau$ como o semiespaço caracterizado a partir de $\mathbf{u}$ com:

$$\text{ORCA}_{i|j}^\tau = \{\mathbf{v} \mid (\mathbf{v} - (\mathbf{v}_i^{\text{opt}} + \frac{1}{2}\mathbf{u})) \cdot \mathbf{u} \geq 0\}. \quad (2.16)$$

No caso bidimensional, $\text{ORCA}_{i|j}^\tau$ é o semiplano definido à direita da reta que passa por $\mathbf{v}_i^{\text{opt}} + \frac{1}{2}\mathbf{u}$ e tem como direção o versor $\hat{\mathbf{d}}$ perpendicular a $\mathbf{u}$ (tal que $\mathbf{u} \times \hat{\mathbf{d}} > 0$). Em três dimensões, $\hat{\mathbf{u}}$ é a normal de um plano que passa por $\mathbf{v}_i^{\text{opt}} + \frac{1}{2}\mathbf{u}$ e divide o espaço de forma semelhante. $\mathbf{v}_i^{\text{opt}}$ é escolhida como a velocidade de otimização para o problema, e pode ser igual a $\mathbf{v}_i^{\text{pref}}$, mas não necessariamente o é. Essa distinção é importante pois o conjunto $\text{ORCA}_{i|j}^\tau$ é definido para incluir todas as velocidades próximas de $\mathbf{v}_i^{\text{opt}}$ que não permitem colisões, mas não necessariamente $\mathbf{v}_i^{\text{pref}}$ ⁵. A minimização do erro é feita quando se considera a equação (2.18).

O conjunto $\text{ORCA}_{i|j}^\tau$ define todas as velocidades permitidas a a_i no momento τ para efetivar metade da reação de desvio de a_j . Desta forma, pode-se escolher velocidades neste subespaço que atendam a restrições adicionais, tais como definidas pela existência de outros agentes. Assim, o conjunto de velocidades permitidas ao agente i que atendem a todas as restrições de prevenção recíproca de colisão e não ultrapassam em módulo uma rapidez máxima

⁵ Na prática escolhe-se $\mathbf{v}_i^{\text{opt}} = \mathbf{v}_i$, ou seja, a velocidade instantânea como velocidade de otimização, conforme discutido em (BERG *et al.*, 2011).

v_{max}^i é definido por:

$$\text{ORCA}_i^\tau = D(\mathbf{0}, v_{max}^i) \cap \bigcap_{j \neq i} \text{ORCA}_{i|j}^\tau, \quad (2.17)$$

conforme ilustrado no caso 2D pela região hachurada na Figura 8. A nova velocidade escolhida para o momento τ é:

$$\mathbf{v}_i^{new} = \underset{\mathbf{v} \in \text{ORCA}_i^\tau}{\text{argmim}} \|\mathbf{v} - \mathbf{v}_i^{pref}\|, \quad (2.18)$$

Figura 8 – (a) Configuração com oito agentes no plano. As suas velocidades correntes são ilustradas com setas. (b) Os semi-planos de velocidades permitidas ao agente A induzidos por cada um dos outros agentes com horizonte de tempo $\Delta = 2$ e com $\mathbf{v}_i^{opt} = \mathbf{v}_i$ para todos os agentes i (i.e. a velocidade de otimização é igual à velocidade atual). Os semi-planos E e C coincidem. A região hachurada é ORCA_A^τ , e contém as velocidades para A que são permitidas em respeito a todos os outros robôs. A seta indica a velocidade corrente de A. Um programa quadrático pode facilmente encontrar a velocidade em ORCA_A^τ mais próxima de $\mathbf{v}_A^{opt}$, que constitui $\mathbf{v}_A^{new}$.

Fonte: [Berg et al. \(2011\)](#).

e pode ser encontrada de forma eficiente usando um programa quadrático⁶ de baixa dimensionalidade n , em tempo linear ([BERG et al., 2008](#)), pois ORCA_i^τ é uma região convexa induzida por hiperplanos que definem restrições lineares. A dimensionalidade do programa quadrático corresponde aos eixos de liberdade da navegação, então para veículos aéreos $n = 3$.

Em situações de densa ocupação do espaço, o conjunto ORCA_i^τ pode se tornar vazio, tornando impossível encontrar uma $\mathbf{v}_i^{new}$ válida. Nessa situação, as restrições induzidas pelos

⁶ Quando introduzido em ([BERG et al., 2011](#)), o algoritmo foi chamado de programa linear, porém como a distância é usada como função de minimização, o programa se torna quadrático. O único impacto que essa distinção traz na prática, entretanto, é que a solução pode se encontrar em qualquer ponto da fronteira de otimização, e não somente nos extremos.

outros agentes são relaxadas, de forma que um programa linear⁷ (de dimensionalidade $n + 1$, ou seja, 4 neste caso) é usado para encontrar a velocidade que menos penetra a região de perigo. Isso permite que uma solução razoável sempre seja alcançada. Mais detalhes sobre o algoritmo ORCA podem ser encontrados em (BERG *et al.*, 2011) e (SNAPE; MANOCHA, 2010).

2.4 Considerações Finais

Este capítulo apresentou os principais conceitos necessários para o entendimento das áreas abordadas durante o desenvolvimento deste doutorado. Foram apresentados os conceitos de VANTS, provendo informações que vão desde os conceitos básicos, definições e características, passando pela apresentação das classificações existentes em alguns países, até chegar às principais missões militares e civis nas quais os VANTS podem ser empregados.

VANTS, como mencionado anteriormente, são sistemas complexos que realizam missões complexas e precisam ser inseridos no espaço aéreo, seja segregado ou não-segregado, mas isso deve ser feito de forma segura. Para que a inserção seja feita de forma segura, um dos elementos que se fazem necessários é a detecção e desvio de obstáculos, com objetivo de evitar colisões. Isso pode ser feito com o uso de câmeras, e desta forma por meio de imagens digitais. Sendo assim, foram também apresentados os conceitos básicos sobre processamento de imagens e sobre técnicas de detecção e desvio de obstáculos.

Pode-se observar que existem diversos algoritmos e técnicas utilizados para efetuar a detecção e o desvio de obstáculos em veículos aéreos não tripulados de pequeno porte, os MAV, e que, apesar de existirem vários trabalhos com pesquisas nesse tópico, ele continua sendo um desafio a ser superado na área de robótica móvel.

Sendo assim, o próximo capítulo apresenta os trabalhos encontrados na literatura aberta que são relacionados a essa tese de doutorado, de modo a permitir delimitar e demonstrar o *gap* existente na área em que se insere esse trabalho.

⁷ Neste caso é realmente usado um programa linear, e não quadrático, pois as soluções não dependem da velocidade preferencial e são sempre encontradas nos extremos.

DETECÇÃO E DESVIO DE OBSTÁCULOS EM VANTS - ESTADO DA ARTE

Vários trabalhos tem destacado o desafio de realizar a detecção e desvio de obstáculos em veículos aéreos não tripulados ([ARMY, 2010](#); [YEARBOOK, 2011](#); [AUTHORITY, 2012](#); [DOD, 2013](#); [YANG](#); [ALVAREZ](#); [BRUGGEMANN, 2012](#); [AL-KAFF *et al.*, 2017](#)). Alguns deles afirmam a necessidade de possuir um sistema de detecção e desvio de obstáculos para permitir a inserção desses veículos no espaço aéreo, seja ele segregado ou não-segregado ([AUTHORITY, 2012](#); [DOD, 2013](#)).

Com objetivo de descrever o estado da arte os algoritmos, sistemas, modelos e abordagens de detecção e desvio de obstáculos em veículos aéreos não tripulados, foi realizada uma revisão sistemática sobre o tema. Os métodos utilizados para a revisão bem como algumas estatísticas obtidas são apresentadas no [Apêndice A](#).

3.1 Principais Abordagens

Por meio da revisão sistemática foi possível elencar quais são as principais abordagens voltadas à detecção e desvio de obstáculos em veículos aéreos não tripulados que utilizam visão monocular como base para a detecção.

Para a realização da detecção de obstáculos e percepção de sua posição podem ser utilizados métodos baseados em:

- *Simultaneous Localization And Mapping* (SLAM) - Cria um mapa do ambiente ao mesmo tempo em que se localiza. O mapa criado é utilizado para desviar dos obstáculos;
- “*Monocular cues*” - Utilizam como base “dicas” presentes em imagens para estimar a distância. Uma das informações mais consideradas é o tamanho relativo do objeto. Desta

forma, são utilizados Detectores e Descritores de características para identificar o tamanho de objetos na imagem, e com base na variação do tamanho do objeto é calculada a distância para o objeto;

- **Aprendizado de Máquina** - Emprega técnicas de aprendizado de máquina (geralmente redes neurais artificiais) para aprender a perceber a profundidade dos obstáculos e gerar uma base para realizar o desvio. Para tal, estes métodos utilizam uma fase de treinamento de acordo com um comportamento desejado pré-definido.

O método a ser empregado no desvio deve trabalhar com as informações obtidas pelo método de detecção. Para tal, a escolha do método de desvio deve considerar o aspecto de tais informações. Neste sentido, as informações obtidas pelo método de desvio podem ser: um mapa ou estrutura da cena; a direção e distância dos obstáculos; ou áreas livres onde é possível navegar.

Nas próximas seções serão apresentados os trabalhos identificados em cada uma das abordagens principais. Alguns trabalhos possuem abordagens singulares não sendo possível agrupá-los juntos com as principais. Eles estão descritos na [Seção 3.5](#)

3.2 Métodos Baseados em SLAM

Como descrito anteriormente, o SLAM permite a criação de um mapa do ambiente enquanto sua localização é calculada simultaneamente. Este mapa, bem como a localização do VANT podem ser utilizados como entrada para o sistema de desvio.

[Langelaan e Rock \(2004\)](#) descrevem a implementação de um sistema que utiliza como entrada uma imagem monocular e dados provenientes de IMU e, utilizando um estimador em conjunto com *Unscented Kalman Filter* (UKF) ([JULIER; UHLMANN, 1997](#)) criam um modelo do ambiente que pode ser utilizado em desvio de obstáculos, no entanto, a abordagem é apenas demonstrada por simulação. Em seguida ([LANGELAAN; ROCK, 2005](#)), os autores substituem a utilização do UKF por um método de SLAM e realizam experimentos utilizando um carro-robô. Ainda assim, apenas o resultado em relação a posição estimada do veículo, que é de 10% da distância percorrida, são apresentados pelos autores, que se limitam em dizer que o sistema foi usado com sucesso para navegar em um ambiente com obstáculos.

O trabalho proposto por [Weiss et al. \(2010\)](#) utiliza um método de SLAM para gerar um mapa de pontos e descreve uma forma para gerar *mesh* e textura para o mapa gerado. O posicionamento gerado pelo SLAM é utilizado como entrada para o controle e o veículo evita passar pela *mesh* gerada, o que segundo os autores é suficiente para evitar obstáculos. Porém, o maior esforço do trabalho foi concentrado na geração da *mesh* e textura, não demonstrando a efetividade do método para desvio de obstáculos.

Uma outra abordagem possível é a criação de mapas esparsos do terreno utilizando dados provenientes de um sistema inercial (IMU) e uma sequência de imagens como proposto por

Zhang, Goubran e Straznicky (2012). Neste trabalho o filtro de Kalman Estendido (*Extended Kalman Filter* (EKF)) (LEWIS, 1986) é utilizado para realizar a fusão dos parâmetros capturados da imagem com os dados da IMU. O método foi verificado utilizando vídeos gravados em um helicóptero para a geração dos mapas do terreno. Contudo, não foi proposta uma abordagem para controle de navegação e realização de desvio de obstáculos.

Similarmente, Magree, Mooney e Johnson (2013) propõem um método para criar mapas de terrenos. No entanto, eles desenvolveram um algoritmo baseado em detecção de pontos de interesse (detector de bordas de Harris) e modelo de câmera. O filtro de Kalman Estendido é utilizado para estimar a posição dos pontos detectados no ambiente e o desvio é realizado por meio da alteração de altitude do veículo. O método foi verificado por meio de simulação e demonstrou que o VANT conseguiu detectar a presença de construções, mas a falta de textura na construção afetou a detecção da altitude correta a ser seguida. Foram também realizados experimentos com gravações obtidas de um helicóptero onde foi possível calcular com sucesso a altitude das árvores presentes no cenário.

Ha e Sattigeri (2012) apresentam um método para criação de mapas densos em volta dos obstáculos utilizando técnicas de segmentação de imagem em conjunto com um algoritmo de SLAM. O método é verificado por meio de simulações e uma estratégia simples de desvio em 2D é apresentada. As simulações mostraram que o método é efetivo em detectar os limites dos obstáculos utilizando mapas esparsos obtidos por meio de um algoritmo de SLAM.

Em um outro tipo de abordagem, Williams, Konev e Coenen (2014) utilizam o método SLAM com enfoque na localização, e não no mapa gerado. No cenário proposto existem múltiplos veículos aéreos, e cada um calcula sua própria localização utilizando o método *Parallel Tracking And Mapping* (PTAM). A localização é então compartilhada entre os múltiplos agentes e as colisões são evitadas utilizando o algoritmo ORCA. A abordagem foi verificada por meio de simulações e experimentos reais utilizando um AR.Drone com processamento realizado em uma estação em terra. Nos experimentos reais, 25% dos testes resultaram em colisão, no entanto, os autores alegam que tais colisões foram resultado de falhas de comunicação.

Para incorporar informação métrica ao algoritmo ORB-SLAM2 utilizando visão monocular, Rojas-Perez e Martinez-Carranza (2017) propõem a utilização de uma câmera voltada para o solo em um ângulo de 30° e a operação da aeronave em uma altitude fixa (no caso, 1 metro). Também é utilizada a segmentação baseada no espaço de cor HSV para detectar a base dos obstáculos. Experimentos foram conduzidos utilizando o *drone* Bebop 2 com processamento realizado em estação de terra e os testes demonstraram uma taxa de falha de 14%. Os autores justificam que as falhas foram causadas pela dificuldade de manter a altitude fixa em 1 metro devido a rajadas de vento, o que corrompeu o mapa gerado.

3.3 Métodos Baseados em *Monocular Cues*

Alguns métodos utilizam informações presentes na imagem para auxiliar a estimativa da profundidade dos objetos. Algumas das informações que podem ser utilizadas para isto são linhas de perspectiva, tamanho dos obstáculos, profundidade de foco, entre outros.

Neste contexto, o método proposto por [Chavez e Gustafson \(2009\)](#) utiliza o detector e descritor de pontos de interesse SIFT para detectar e classificar pontos, em seguida estes pontos são utilizados para estimar o tamanho relativo do objeto. Por fim, a variação no tamanho dos objetos gera base para estimar o tempo para a colisão. A saída do método é a classificação de uma determinada situação como um cenário onde a colisão é iminente ou sem risco de colisão. O método foi testado utilizando um *Unmanned Ground Vehicle* (UGV) e obteve uma taxa de erro total de 18%.

[Byrne e Taylor \(2009\)](#) definem segmentação por expansão como o agrupamento e classificação de conjuntos de *pixels* como áreas de colisão e áreas de não colisão utilizando probabilidade conjunta de movimento de expansão e cor. Com isto, eles propuseram um método em que é possível calcular o tempo para colisão para todos os grupos de *pixels* de um determinado *frame*. Resultados por meio de simulação em cenário urbano mostram que, apesar de em alguns *frames* mais de 50% dos *pixels* serem classificados de maneira errada, a maioria dos *frames* possuem erro inferior a 20%. No entanto, os autores apresentam apenas um gráfico com a porcentagem de classificação errada para cada *frame* e não fornecem uma estatística mais detalhada sobre os resultados.

Uma das técnicas utilizadas para estimar a movimentação em uma cena é o fluxo óptico. Desta forma, considerando que um algoritmo de fluxo óptico gera vetores de movimentação dos *pixels* na imagem, [Gosiewski, Ciesluk e Ambroziak \(2011\)](#) propõem um método que compara o módulo dos vetores com uma média calculada com base em um subconjunto dos vetores presentes no *frame*. Com isto é possível extrair informação tridimensional sobre os obstáculos. O método foi verificado por meio de *hardware-in-the-loop* mas não há informações em relação a precisão de localização dos obstáculos. Os autores apenas afirmam que o método funciona bem ao lidar com um único obstáculo e em desfiladeiros urbanos.

[Mori e Scherer \(2013\)](#) utilizam o tamanho do obstáculo como uma informação para calcular o tempo para a colisão. Para tal, o algoritmo SURF é utilizado para detectar e descrever pontos para possibilitar a correspondência entre os *frames*. Com isto, é possível comparar o tamanho das regiões em volta do ponto e reconhecer quando a câmera se aproxima dos obstáculos. O algoritmo proposto funciona sem ajuda de um sistema inercial para estimar a movimentação da câmera, ao contrário do proposto por [Byrne e Taylor \(2009\)](#). Testes foram realizados utilizando um AR.Drone com processamento realizado em estação em terra e 13% dos experimentos realizados resultaram em colisão.

A proposta de um *framework* para atender tanto VANTs quanto UGVs foi proposta

por [Arokiasami et al. \(2016\)](#). No seu trabalho a detecção é realizada por meio de métodos de processamento de imagem simples aliados a um algoritmo de fluxo óptico. Com base nestas informações, o tamanho relativo do obstáculo é calculado e uma distância para o obstáculo é estimada. O trabalho foi testado utilizando tanto um veículo terrestre como um veículo aéreo e o maior enfoque do teste foi demonstrar a modularidade do *framework*. Alguns dados sobre a estimativa de posição utilizando o tamanho do obstáculo demonstram um erro maior que 50% quando o obstáculo estava a mais de 4 metros de distância; no entanto, quando o obstáculo estava a menos de 3,5 metros o erro cai significativamente.

Assim como os outros métodos descritos nesta seção, [Al-Kaff et al. \(2016\)](#), [Al-Kaff et al. \(2017\)](#) também utilizou da informação do tamanho do obstáculo para estimar a sua proximidade. [Al-Kaff et al. \(2016\)](#) apresenta um método baseado no algoritmo SIFT para a detecção de pontos de interesse e utiliza o tamanho dos pontos detectados, juntamente com o tamanho de uma forma convexa construída a partir destes pontos. O algoritmo foi testado em um AR.Drone com processamento realizado por uma estação em terra onde foi possível detectar obstáculos de diferentes tamanhos a uma distância de 90 a 120 centímetros com uma acurácia de 92,5%.

[Al-Kaff et al. \(2017\)](#) apresenta uma modificação do sistema de desvio em relação à [Al-Kaff et al. \(2016\)](#). Os resultados obtidos mostram uma acurácia de 97,4% a uma distância de 90 a 120 centímetros.

Outro método que utiliza o tamanho relativo do objeto com informação para a realização da detecção e desvio do obstáculo é apresentado por ([AGUILAR; CASALIGLLA; PÓLIT, 2017](#); [AGUILAR et al., 2017](#); [AGUILAR; CASALIGLLA; POLIT, 2017](#)). No entanto, em sua proposta os obstáculos devem estar previamente registrados em um banco de dados. O algoritmo utiliza o algoritmo SURF para fazer a detecção e descrição de pontos de interesse e compara com os objetos contidos no banco de dados. O principal enfoque do trabalho é em um controlador que realiza o desvio ao tentar maximizar a distância entre o centro de massa do objeto e o centro da imagem. A abordagem é testada por meio de experimentos em um *Drone Bebob* com o processamento sendo realizado em uma estação em terra. Os resultados apresentados demonstram a diferença de tempo de voo entre voos realizados com o controle proposto e voos comandados por operadores humanos (com ou sem experiência).

3.4 Métodos Baseados em Aprendizado de Máquina

Alguns trabalhos buscam resolver o problema de detecção e desvio de obstáculos por meio de visão monocular e algoritmos de aprendizado de máquina.

O trabalho realizado por [Lenz, Gemici e Saxena \(2012\)](#) propõe a utilização de um processador neuromórfico de baixo consumo de energia para classificar obstáculos. Para realizar o desvio o algoritmo utiliza uma biblioteca de movimentação e máscaras espaço-visuais, quando menos de um número determinado de *pixels* na área da máscara é rotulado como obstáculo, a

movimentação naquela direção é considerada segura. Cabe destacar que as máscaras de espaço visual são características desse trabalho especificamente.

O treinamento foi realizado com base em um *dataset* incluindo quatro tipos de obstáculo: árvores, construções (apesar de não aparecerem nos resultados apresentados pelos autores), postes e cercas; onde 80% do *dataset* foi utilizado no treinamento e os 20% restantes foram utilizados para teste. Os testes foram realizados em um *AirRobot* e o resultado demonstrou 100% de classificações corretas. No entanto, o sucesso na manobra de desvio ocorreu em 96,2% dos testes realizados.

Ross *et al.* (2013) apresenta um método para realizar o treinamento de um controle reativo a partir de imagens monoculares. Para realizar o treinamento, são computados descritores a partir da imagem. Estes descritores são entrada, juntamente com os comandos realizados por um especialista humano, para um sistema de aprendizado. Com isto, é possível treinar o VANT a realizar desvios reativos para a direita e para a esquerda. Os testes foram realizados em um AR.Drone e apresentaram uma falha (colisão) a cada 40 metros percorridos.

3.5 Outros Estudos Focados em Visão Monocular

A utilização de informações de distribuição estatística obtidas de vetores de fluxo óptico ou descritores de características podem ser utilizadas para extrair informação 3D a partir de uma sequência de imagens. Tal abordagem foi explorada por Merrell, Lee e Beard (2004), Lee *et al.* (2004) e posteriormente por Lee *et al.* (2010). Nestas abordagens os autores se inspiram em métodos de *Structure from Motion* (SfM).

Merrell, Lee e Beard (2004) propõe a utilização de distribuição vetores de fluxo óptico para melhorar dois métodos de SfM, sendo um deles linear e outro não-linear. O método foi verificado por meio de simulação e a média de erro na estimativa de profundidade foi de 10%.

Já no trabalho proposto por Lee *et al.* (2010) a preocupação foi em relação ao modelo de ruído presente no fluxo óptico. Eles propuseram um método para atender casos onde o ruído não é branco ou distribuído de maneira uniforme. No entanto, os resultados não tratam do erro em relação à estimativa de profundidade.

O modelo de movimentação da câmera em conjunto com um EKF é apresentado por Shah e Johnson (2009) como uma solução para a criação de um mapa 3D a partir de imagens monoculares. Simulações demonstram que ao utilizar dados de um voo lateral por 60 segundos é possível recuperar a informação 3D de 8 pontos com um erro de 3% e para 35 pontos são necessários 100 segundos de voo. No entanto, o voo frontal é criticamente afetado pela velocidade e o tempo de voo requerido para obter o modelo 3D é 25% superior ao voo lateral.

O trabalho realizado por Lee *et al.* (2011) constrói um mapa do ambiente utilizando como base pontos chave detectados utilizando o algoritmo SIFT e um modelo de projeção. O

algoritmo SIFT é utilizado para fazer a correspondência dos pontos em duas imagens distintas e o modelo de projeção fornece base para calcular a posição do VANT. A informação tridimensional dos pontos detectados é obtida por meio da utilização do algoritmo *Multi-scale Oriented Patches* (MOPS) juntamente com o SIFT. O método foi verificado por meio de simulação utilizando o MATLAB e foram fornecidas estatísticas em relação ao tempo de execução do algoritmo, mas não foram fornecidas estatísticas em relação à precisão do método.

De maneira similar, [Saha, Natraj e Waharte \(2014\)](#) utilizou a fórmula de projeção proposta por [Lee et al. \(2011\)](#) para gerar equações para calcular a posição 3D das *features* obtidas pelo método SURF. Para obter a localização e movimentação do VANT foram utilizados dados provenientes de uma IMU filtrados utilizando EKF. No entanto, no modelo criado os autores consideraram apenas a movimentação do VANT em relação ao seu ângulo de *yaw*. Foram realizados testes utilizando um AR.Drone pairando a uma distância de 5 metros, onde foi possível observar que o erro no cálculo de distância se manteve abaixo de 2%. Para testes realizados com o VANT em movimento a velocidade de 150 cm/s o erro observado foi de 3%.

Uma abordagem bioinspirada é apresentada por [Stowers, Hayes e Bainbridge-Smith \(2011\)](#). Em sua abordagem, informações de fluxo óptico são utilizadas para criar um controle reativo para desvio de obstáculos. O sistema foi testado utilizando um *dataset* de imagens e mostrou que o método proposto tem mérito, no entanto, a avaliação executada foi qualitativa, faltando a realização de testes quantitativos.

Outra estratégia de controle é apresentada por [Fu et al. \(2013a\)](#), [Fu et al. \(2013b\)](#) onde um método de SLAM é utilizado como base para fornecer a posição dos obstáculos. O enfoque da proposta foi em relação ao método de controle que faz o desvio ao voar paralelamente ao obstáculo (por exemplo, uma parede). Para tal, o ângulo de *yaw* é ajustado para evitar o obstáculo. O método foi verificado por simulação e experimentos reais utilizando um AR.Drone. No entanto, os resultados foram focados no desempenho do controle, não sendo fornecidos dados relacionados à precisão de identificação dos obstáculos.

Uma alternativa à realização do mapeamento do ambiente é apresentada por [Kong, Sheng e Teredesai \(2014\)](#) que procura segmentar a imagem monocular em áreas navegáveis e áreas não navegáveis utilizando apenas algoritmos simples de processamento de imagens. Em sua proposta, eles utilizam em sequência os algoritmos: *threshold*, borramento, detecção de bordas (*Canny*), detecção de formas geométricas (*Hough Transform*) e detecção de contornos. Os testes utilizando um AR.Drone e estação de terra demonstraram que o algoritmo pode executar a uma alta taxa de *frames* por segundo, no entanto, a taxa de colisões alcançou 25% em um cenário específico.

[Ma et al. \(2015\)](#) utilizam um método de detecção de saliência e utilizam a região destacada na imagem para propor um controle que varia o ângulo de *yaw* para realizar o desvio. A abordagem foi implementada em um AR.Drone com processamento em estação em terra para realizar o desvio de um único obstáculo estático, no entanto, não são apresentados resultados

quantitativos.

A proposta apresentada por [Escobar-Alvarez et al. \(2017\)](#) consiste em um sistema capaz de se mover até um objetivo visual em altas velocidades, sem colidir com obstáculos no meio do caminho. O algoritmo desenvolvido é baseado em fluxo óptico para estimar a taxa de expansão e o tempo para colisão. A proposta foi implementada em um VANT dotado de câmera monocular, IMU, sensor laser (*LiDAR-Lite V2*) e um processador NVIDIA Tegra TK1. Os testes realizados mostraram que foi possível alcançar velocidades de até 19 m/s. No entanto, os autores não fornecem informações sobre a taxa de sucesso, ou seja, quantos voos foram realizados sem colisão.

3.6 Considerações Finais

Os principais trabalhos envolvendo visão monocular para a detecção e desvio de obstáculos são sumarizados na [Tabela 2](#). Para a confecção do quadro a entrada imagens monoculares foi omitida, pois todos os métodos listados aceitam tal entrada.

É possível observar na [Tabela 2](#) que a maioria dos métodos tenta extrair informação tridimensional a partir de imagens monoculares. A maioria deles utiliza métodos baseados em SLAM para realizar esta tarefa. No entanto, a informação do erro do cálculo dos artefatos tridimensionais localizados não é informada pela maioria dos trabalhos.

Saber a posição do obstáculo com alta acurácia é importante para permitir a utilização de algoritmos de desvio mais robustos, como o ORCA ([BERG et al., 2011](#)) que possuem premissas fortes em relação aos sensores envolvidos.

Dos trabalhos listados na [Tabela 2](#) apenas um utiliza o ORCA como método para o desvio ([WILLIAMS; KONEV; COENEN, 2014](#)). Entretanto, o método apresentado por [Williams, Konev e Coenen \(2014\)](#) não utiliza um algoritmo de detecção de obstáculos, mas trabalha com VANTs independentes que inferem sua posição aplicando um método de SLAM e compartilham utilizando comunicação por rede (o que viola uma das premissas estabelecidas pelo algoritmo ORCA).

Também é possível observar que nenhum trabalho listado gera a posição tridimensional dos obstáculos a partir da fusão entre uma sequência de imagens, dados de IMU e dados de posicionamento do VANT visando obter uma melhor acurácia nos resultados.

De forma a resolver os pontos em aberto verificados no estado da arte, o [Capítulo 4](#) apresenta o MOSAIC, um sistema de detecção e desvio de obstáculos capaz de gerar a posição tridimensional do obstáculo de maneira confiável por meio da fusão de uma sequência de imagens monoculares, dados de IMU e de posicionamento (obtidos a partir de GPS); e realizar o desvio destes obstáculos utilizando um método de desvio robusto do tipo *Reciprocal n-body Collision Avoidance*, permitindo assim não só contribuições de pesquisa mas um avanço no estado da arte.

Tabela 2 – Comparação entre os métodos que utilizam visão monocular

Método	IMU	Localização	Método de Detecção	Artefato de detecção	% Erro (Posição do Obstáculo)	Método de Desvio	Método de Avaliação
Langelaan e Rock (2005)	Sim	SLAM	SLAM	Pontos 3D	N/D ¹	Campos Potenciais	UGV
Weiss <i>et al.</i> (2010)	Não	SLAM	SLAM	3D <i>Mesh</i>	N/D	N/D	MAV
Zhang, Goubran e Straznicky (2012)	Sim	N/U	Shi-Tomasi e Lucas-Kanade	Mapa do Terreno	N/D	N/R ²	UAV levado por Helicóptero
Magree, Mooney e Johnson (2013)	Não	SLAM	SLAM	Mapa do Terreno	N/D	Controle de Altitude	Simulação e Helicóptero
Ha e Sattigeri (2012)	Sim	SLAM	SLAM	Mapa Denso	N/D	Desvio em 2D	Simulação
Williams, Konev e Coenen (2014)	Sim	SLAM	N/U ³	Posição do VANT	N/D	ORCA	Ar.Drone
Rojas-Perez e Martinez-Carranza (2017)	Sim	SLAM	SLAM	Pontos 3D	N/D	Reativo (Espaço livre)	Bebop2
Chavez e Gustafson (2009)	Não	Não	Tamanho Relativo	Distância	N/D	N/D	UGV
Byrne e Taylor (2009)	Sim	GPS	Tamanho Relativo	Tempo para Colisão	N/D	N/R	Simulação

Método	IMU	Localização	Método de Detecção	Artefato de detecção	% Erro (Posição do Obstáculo)	Método de Desvio	Método de Avaliação
Gosiewski, Ciesluk e Ambroziak (2011)	Não	Não	Lucas-Kanade	Vetores no Centro da Imagem	N/D	Reativo	Simulação
Mori e Scherer (2013)	Não	Não	Tamanho Relativo	Tempo para Colisão	N/D	Reativo	AR.Drone
Arokiasami <i>et al.</i> (2016)	Sim	Não	Tamanho Relativo	Distância	27,5% à 4 metros	Reativo	Simulação e AR.Drone
Al-Kaff <i>et al.</i> (2017)	Não	Não	Tamanho Relativo	Presença de obstáculos	N/D	Reativo	AR.Drone
Aguilar, Casaliglla e Pólit (2017)	Não	Não	Tamanho Relativo	N/D	N/D	Controle Proporcional	Bebop
Lenz, Gemici e Saxena (2012)	Não	Não	MRF e RNA	Máscara de Regiões Obstruídas	N/D	RNA	AirRobot
Ross <i>et al.</i> (2013)	Não	Não	Aprendizado por Imitação	N/A ⁴	N/D	Aprendizado por Imitação	AR.Drone

Método	IMU	Localização	Método de Detecção	Artefato de detecção	% Erro (Posição do Obstáculo)	Método de Desvio	Método de Avaliação
Merrell, Lee e Beard (2004)	Não	Não	Distribuição de Probabilidade do Fluxo Óptico	Mapa de Profundidade	N/D	N/R	Imagens Aéreas
Lee <i>et al.</i> (2010)	Não	Não	Distribuição de Probabilidade do Fluxo Óptico	Mapa de Profundidade	N/D	N/R	Imagens Aéreas
Saha, Natraj e Waharte (2014)	Sim	Não	Projeção	Pontos 3D	3% à 5 metros	N/D	Ar.Drone
Lee <i>et al.</i> (2011)	Não	Não	Projeção e MOPS	Pontos 3D	N/D	N/R	Imagens
Stowers, Hayes e Bainbridge-Smith (2011)	Não	Não	Lucas-Kanade	Mapa de Profundidade	N/D	Reativo	wasp quadrotor
Fu <i>et al.</i> (2013b)	Sim	SLAM	SLAM	Pontos 3D	N/D	<i>Visual Servoing</i>	AR.Drone
Ma <i>et al.</i> (2015)	Não	Não	Detecção de Saliência	Direção do Obstáculo	N/D	Ajuste de <i>Yaw</i>	AR.Drone

Método	IMU	Localização	Método de Detecção	Artefato de detecção	% Erro (Posição do Obstáculo)	Método de Desvio	Método de Avaliação
Escobar-Alvarez <i>et al.</i> (2017)	Sim	Não	Lucas-Kanade	Taxa de Expansão	N/D	Baseado em Máquina de Estados	DJI Flamewheel F450 quadrotor

N/D¹ – Não disponível

N/R² – Não realizado

N/U³ – Não utilizado

N/A⁴ – Não se aplica

Fonte: Elaborada pelo autor.

MOSAIC - *MONOCULAR OBSTACLE AVOIDANCE*

Diversos trabalhos são encontrados na literatura da área cujo objetivo é a detecção e o desvio de obstáculos, principalmente a partir do uso de câmeras. O uso de câmeras monoculares trazem diversas vantagens, como por exemplo, o menor peso, menor consumo de energia e requer menor poder de processamento para a análise (em relação às câmeras estéreo). No entanto, dentre esses trabalhos, como pôde ser observado no capítulo anterior, poucos utilizam câmeras monoculares com foco na detecção sem alterações nas câmeras e com uso de integração dessas imagens com informações de sensores dos VANTs, além de visar o baixo custo e a segurança na detecção e desvio evitando colisão. Este capítulo tem então como objetivo a apresentação do MOSAIC, que provê a detecção e o desvio de obstáculos de VANTs do tipo MAV.

4.1 MOSAIC

Tendo em vista os vários desafios relacionados ao problema de detecção e desvio de obstáculos, foi proposto o *Monocular ObStacle AvoIdanCe* (MOSAIC), um sistema de detecção e desvio de obstáculos que utiliza como base para a detecção imagens monoculares e dados de posicionamento e atitude do VANT. O desvio é realizado utilizando uma variação do algoritmo ORCA 3D.

Os outros trabalhos que utilizam o ORCA para o desvio funcionam por meio de simulação ou necessitam de um meio de comunicação para compartilhar a localização entre os agentes, quebrando uma das premissas do algoritmo ORCA.

Apesar das fortes premissas apresentadas pelo algoritmo ORCA 3D para a realização do desvio, o método desenvolvido para o cálculo da posição do obstáculo permite a utilização de uma variação deste algoritmo para realizar o desvio do obstáculo o que apresenta grande contribuição ao estado da arte.

Na Figura 9 é ilustrada a arquitetura geral do MOSAIC. Diferentemente dos trabalhos apresentados no Capítulo 3, o MOSAIC tem como foco a detecção de pontos significativos em imagens monoculares e realiza o cálculo do posicionamento destes pontos utilizando a posição destes em uma sequência de imagens em conjunto com os dados de posicionamento do VANT.

Figura 9 – Arquitetura do Sistema MOSAIC

Fonte: Elaborada pelo autor.

Para seu funcionamento, o MOSAIC pressupõe que o VANT possui uma câmera fixa e uma fonte de posicionamento e atitude (IMU e GPS) alinhados com esta câmera. É importante notar, entretanto, que como os dados de vídeo e de posicionamento e a atitude partem de *links* diferentes pode não haver o alinhamento temporal (os dados de vídeo em geral possuem um atraso em relação aos dados de posicionamento e atitude).

O módulo de **Detecção** (Seção 4.2) é então responsável por receber os dados do *link* de vídeo proveniente do VANT e encontrar pontos na imagem que devem ser verificados como possíveis obstáculos.

Ao mesmo tempo, o módulo de **Processamento de Dados do VANT** (Seção 4.4) recebe os dados do *link* de dados de posicionamento e atitude (no MOSAIC isto ocorre especificamente por meio do protocolo MAVLINK¹) e guarda um histórico para permitir o posterior alinhamento

¹ O protocolo MAVLINK foi selecionado por ser o utilizado em VANTs com piloto automático pixhawk[®] que é o caso dos VANTs disponíveis para uso no laboratório de pesquisa. Entretanto, o módulo de **Processamento dos Dados do VANT** pode ser facilmente adaptável para uso de outros protocolos.

temporal.

Como a precisão dos dados de posicionamento e atitude do VANT são essenciais para o funcionamento do cálculo da posição dos obstáculos, está previsto o módulo de **Pose Estimation** (Seção 4.3) que pode gerar dados de posicionamento e atitude do VANT apenas a partir do *link* de vídeo. Isto permite que a fusão dos dados do VANT com dados provenientes de algoritmos baseados em imagem refinem as informações de posicionamento e atitude. Apesar de estar previsto como algo que pode melhorar o funcionamento geral do MOSAIC, este módulo de fusão não faz parte do escopo deste trabalho e pode ser explorado em trabalhos futuros.

O módulo de **Posição do Obstáculo** (Seção 4.5) recebe então as informações provenientes dos módulos de **Detecção** e **Processamento de Dados do VANT** e realiza o alinhamento temporal e interpolação para tratar dados que chegam com frequências diferentes. Os dados alinhados são armazenados em um histórico de tamanho n para serem repassados ao módulo de **Estimativa de Posição do Obstáculo** (Seção 4.6).

Assim que o módulo de **Estimativa de Posição do Obstáculo** recebe os dados provenientes de n frames é feito o cálculo da posição 3D dos pontos, que são retornados para o módulo de **Posição do Obstáculo**, responsável por repassar estas informações ao módulo de **Controle de Missão e Navegação do MOSAIC** (Seção 4.7).

Apesar do VANT possuir seu próprio módulo de missão e navegação, o módulo existente no MOSAIC acrescenta o controle do desvio dos objetos reportados pelo módulo de **Posição do Obstáculo**. Desta forma, o **Módulo de Desvio** (Seção 4.8), que compreende o algoritmo ORCA 3D, deve receber informação a respeito de um determinado objetivo e os obstáculos que possam estar no caminho entre o VANT e o objetivo. Com isto, o controle dos *waypoints* de uma missão é realizado pelo MOSAIC que registra estes objetivos no **Módulo de Desvio** juntamente com os obstáculos localizados. O **Módulo de Desvio** fornece então os vetores de velocidade que o VANT deve seguir ao módulo de **Missão e Navegação do MOSAIC** que repassa este vetor ao controle do VANT (utilizando o protocolo MAVLINK).

4.2 Detecção

Existem diversos métodos e algoritmos que podem ser empregados na tarefa de detecção de obstáculos em VANTs. Entre estas técnicas destacam-se as baseadas em SLAM (LANGELAAN; ROCK, 2004; LANGELAAN; ROCK, 2005; WEISS *et al.*, 2010; ZHANG; GOUBRAN; STRAZNICKY, 2012; MAGREE; MOONEY; JOHNSON, 2013; HA; SATTIGERI, 2012; WILLIAMS; KONEV; COENEN, 2014; ROJAS-PEREZ; MARTINEZ-CARRANZA, 2017), baseadas em *monocular cues* (CHAVEZ; GUSTAFSON, 2009; BYRNE; TAYLOR, 2009; GOSIEWSKI; CIESLUK; AMBROZIAK, 2011; MORI; SCHERER, 2013; BYRNE; TAYLOR, 2009; AROKIASAMI *et al.*, 2016; AL-KAFF *et al.*, 2017; AGUILAR; CASALIGLLA; POLIT, 2017), métodos baseados em aprendizado de máquina (LENZ; GEMICI; SAXENA, 2012; ROSS

et al., 2013) e métodos focados em visão monocular (MERRELL; LEE; BEARD, 2004; LEE *et al.*, 2010; SHAH; JOHNSON, 2009; LEE *et al.*, 2011; SAHA; NATRAJ; WAHARTE, 2014).

A abordagem adotada pelo MOSAIC divide a tarefa de detecção em duas partes. O módulo de **Detecção** do MOSAIC recebe dados do *link* de vídeo proveniente do VANT e realiza a análise dos *frames* do vídeo de forma a destacar pontos de interesse. Estes pontos de interesse podem ser obtidos por meio da execução de algoritmos detectores/descriptores como o ORB (RUBLEE *et al.*, 2011b; AL-KAFF *et al.*, 2018; AGUILAR; CASALIGLLA; PÓLIT, 2017), ou utilizando estratégias mais elaboradas para permitir a detecção de pontos referentes a objetos distantes, por exemplo, utilizando técnicas de subtração de fundo ou a técnica de CMO (MEJIAS *et al.*, 2010; CARNIE; WALKER; CORKE, 2006).

Tal abordagem permite uma maior flexibilização na escolha do método de análise de imagem que melhor se adéque ao contexto a ser aplicado.

Após a análise de imagem, pontos de interesse e seus respectivos descritores/identificadores são passados para um módulo de cálculo de posicionamento. Tal módulo é responsável por analisar o deslocamento dos pontos em uma sequência de *frames* e, considerando também a movimentação do VANT, calcular a posição no espaço tridimensional dos pontos detectados.

As próximas seções abordarão alguns dos métodos de análise de imagens que podem ser empregados no módulo de **Detecção**.

4.2.1 Baseado no Espaço de Cor

Uma das formas de obter um ponto na imagem referente a um obstáculo é assumir que o obstáculo possui uma cor pré-definida e utilizar uma filtragem baseada no espaço de cor para obter a posição do obstáculo na imagem.

Embora esta abordagem não possa ser muito aplicada em situações reais ela se mostra a ideal para permitir a validação no módulo de estimativa de posição do obstáculo. Isto ocorre pois um objeto de cor e posição conhecidos podem ser inseridos no cenário de teste e, após a execução do algoritmo de estimativa de posição, o erro do posicionamento real e do posicionamento estimado pode ser calculado.

O módulo de detecção baseado no espaço de cor funciona no espaço de cor HSV, mais especificamente sobre o valor da matiz de cor (*Hue*).

Desta forma o filtro para a detecção busca por uma faixa de valor matiz e, após obter a máscara relativa a posição dos objetos que possuem aquela determinada matiz, aplica as operações morfológicas de fechamento (para completar partes não detectadas do objeto) seguida de dilatação (para ampliar o tamanho do objeto na imagem). A operação de dilatação é realizada para melhorar a detecção de contornos e definição do ponto central do objeto. Este ponto é então repassado para o módulo de **Posição do Obstáculo**.

4.2.2 Close Minus Open (CMO)

Para detectar objetos mais distantes, alguns trabalhos seguem uma abordagem baseada nas operações morfológicas de abertura e fechamento (MEJIAS *et al.*, 2010; CARNIE; WALKER; CORKE, 2006). Com base no descrito no Capítulo 2, a operação de fechamento (*close*) funde descontinuidades, eliminando pequenos buracos e preenchendo lacunas; enquanto que a operação de abertura (*open*) rompe os istmos eliminando saliências finas.

Com isto, ao realizar o fechamento e subtrair dele o resultado da abertura (*Close Minus Open*) obtém-se algumas estruturas mais significativas presentes na imagem, eliminando um pouco do ruído ou outros elementos que não sejam preenchidos pelo elemento estruturante utilizado na operação.

Apesar de parte do ruído ser eliminado, o ruído remanescente pode ser destacado por esta técnica como possíveis objetos. Para eliminar este problema alguns trabalhos (MEJIAS *et al.*, 2010; CARNIE; WALKER; CORKE, 2006) propõem a utilização de filtragem temporal, onde, se o artefato localizado estiver presente em *frames* consecutivos ele é selecionado como um objeto, senão ele é descartado como ruído.

O trabalho realizado por Carnie, Walker e Corke (2006) considera que o objeto está se movimentando em velocidade constante e propõe a utilização de matrizes para a filtragem temporal nas diversas direções possíveis para o objeto. Se o artefato detectado permanece vários *frames* em uma mesma matriz de direção ele é selecionado como um objeto. Mejias *et al.* (2010) apresenta uma solução baseada em HMM (*Hidden Markov Model*) e realiza a implementação em *Graphics Processing Units* (GPUs) para conseguir o desempenho necessário para a detecção.

Nesta tese de doutorado foi criada uma variação de filtragem temporal baseada na dilatação do resultado do CMO utilizando um elemento estruturante quadrado com âncora no centro, isto permite que o sinal do objeto seja acompanhado independente da direção de movimentação. O resultado das dilatações são somados de maneira a permitir o cálculo da média ponderada de n *frames* consecutivos. A utilização de média ponderada permite ajustar a importância do *frame* atual para a detecção do objeto.

Os resultados obtidos com esta técnica são apresentados no Capítulo 5 e comparados com a técnica proposta por Carnie, Walker e Corke (2006).

4.2.3 Oriented FAST and Rotated BRIEF - ORB

Como descrito no Capítulo 2 o algoritmo ORB funciona como um detector e descritor de pontos de interesse. Apesar de não ser um algoritmo de detecção de obstáculos, ele é capaz de detectar pontos na imagem e calcular um vetor de características para cada um dos pontos.

Com base no vetor de características, os pontos detectados em um *frame* do *link* de vídeo proveniente do VANT podem ser identificados no *frame* seguinte. Com isto, é possível criar um

histórico de movimentação dos pontos presentes na imagem, que em conjunto com os dados do VANT, podem fornecer a posição tridimensional destes pontos.

Desta forma, o papel do algoritmo ORB no módulo de detecção é apenas gerar o conjunto de pontos e características que são repassados ao módulo de **Posição do Obstáculo** que é onde o histórico de pontos é armazenado e a correspondência dos pontos entre os *frames* ocorre.

Importante ressaltar que, como a posição tridimensional dos pontos pode ser calculada, todos os pontos detectados na imagem podem ser repassados ao módulo de **Desvio** como possíveis obstáculos.

Por se apresentar na literatura como um descritor rápido e eficiente, o ORB foi utilizado nesta tese.

4.3 Pose Estimation

Considerando que a posição e atitude do VANT é de suma importância para a estimativa da posição do obstáculo, foi idealizado um módulo de *Pose Estimation* que tem como finalidade agrupar algoritmos para prover a posição e atitude do VANT por meio de imagens de maneira que estas informações possam, posteriormente, ser fundidas com as informações dos sensores do VANT com o objetivo de aumentar a precisão.

Partindo do pressuposto que o obstáculo emite sinais de rede sem fio, uma outra maneira de refinar o posicionamento do obstáculo é fundir a informação da distância do obstáculo obtida pela intensidade do sinal sem fio com a distância calculada do obstáculo.

Figura 10 – Módulo de *Pose Estimation*

Fonte: Elaborada pelo autor.

O detalhamento proposto para este módulo é ilustrado na [Figura 10](#). No entanto, é

importante enfatizar que este módulo foi previsto de forma a prover melhorias no cálculo da posição do obstáculo e pode ser desenvolvido em trabalhos futuros.

Dentre os itens previstos para esta arquitetura, o módulo de cálculo de distância a partir de sinais sem fio (destacado em verde na [Figura 10](#)) foi implementado para verificar a viabilidade de utilizar a intensidade do sinal proveniente das redes sem fio dos VANTs como uma informação relevante para detecção de obstáculos (no caso, outras aeronaves emitindo sinais por meio de 802.11). Esta informação pode ser fundida aos dados de outros sensores para obter uma maior confiabilidade na tarefa de desvio de obstáculos.

Os resultados da aplicação deste módulo em um cenário, variando a velocidade do receptor, são apresentados no [Capítulo 5](#).

4.4 Processamento de Dados do VANT

O módulo de **Processamento de Dados do VANT** recebe informações de posicionamento e atitude do VANT utilizando o protocolo MAVLINK e os armazena em histórico para permitir o alinhamento com os dados de vídeo, que geralmente possuem um atraso.

Figura 11 – Módulo de Processamento de Dados do VANT

Fonte: Elaborada pelo autor.

Na [Figura 11](#) é ilustrado a arquitetura geral do módulo. Mais uma vez, considerando a importância da acurácia destes dados para o funcionamento do módulo de **Estimativa de Posição do Obstáculo**, está previsto um módulo de **Fusão** de dados de posicionamento e distância baseada em outros sensores (câmera e intensidade de sinal sem fio) que seriam provenientes do

módulo de **Pose Estimation**. Entretanto, este módulo de **Fusão** foge do escopo selecionado para esta tese podendo ser abordado em trabalhos futuros.

Quando o módulo de **Posição de Obstáculo** necessita dos dados de posicionamento do VANT, este módulo recebe uma solicitação contendo o *timestamp* aproximado dos dados necessários e realiza uma busca binária no histórico para retornar o valor mais próximo do *timestamp* solicitado.

4.5 Módulo de Posição do Obstáculo

Este módulo é responsável por armazenar um histórico dos pontos/descriptores detectados e realizar a correspondência (*matching*) dos pontos detectados em diferentes *frames* do vídeo, provenientes do módulo de **Deteção**.

Para realizar o alinhamento temporal dos pontos e características com os dados de posicionamento do VANT, o módulo de **Posição do Obstáculo** solicita ao módulo de **Processamento de Dados do VANT** as informações mais próximas do *timestamp* estimado do vídeo considerando o tempo de atraso de chegada do sinal do *link* de vídeo.

Como a frequência dos dados de posicionamento e atitude do VANT é inferior à frequência do vídeo (que pode ser 30 ou 60 *frames* por segundo (fps) (dependendo das configurações da câmera), os dados de posicionamento e atitude do VANT são interpolados de forma linear.

Como o módulo de **Estimativa de Posição do Obstáculo** pode trabalhar com um número n de *frames*, os dados alinhados são armazenados para serem repassados em lote para o cálculo da posição do obstáculo.

A arquitetura geral do módulo de **Posição do Obstáculo** é ilustrada na [Figura 12](#) onde é possível observar o fluxo das informações de entrada até a saída que é enviada ao módulo de **Estimativa da Posição do Obstáculo**.

4.6 Estimativa da Posição do Obstáculo

Para realizar o cálculo da posição do obstáculo é possível utilizar dados de posicionamento do VANT em conjunto com os dados de posição do ponto detectado na imagem, e isso é feito no módulo **Estimativa da Posição do Obstáculo**. Tal abordagem foi proposta por [Saha, Natraj e Waharte \(2014\)](#), no entanto, ela previa apenas a movimentação do VANT em relação ao ângulo de *yaw*.

O trabalho proposto por [Saha, Natraj e Waharte \(2014\)](#) descreve como encontrar a posição de um obstáculo utilizando um modelo de projeção da câmera apresentado na [Figura 13](#) e a fórmula de projeção (4.1) baseada em proporcionalidade do triângulo, onde: P_I é um ponto na imagem na posição (x_I, y_I) ; f é a distância focal da câmera (em pixels) obtida por meio da

Figura 12 – Módulo de Posição do Obstáculo

Fonte: Elaborada pelo autor.

calibração, $\overrightarrow{CP_o}$ é o um vetor formado entre a posição da câmera (C) e o obstáculo (P_o) e os vetores $\vec{n}_x$, $\vec{n}_y$ e $\vec{n}_z$ indicam a orientação da câmera.

$$P_I(x_I, y_I) = \frac{f}{\vec{n}_x \cdot \overrightarrow{CP_o}} (-\vec{n}_y \cdot \overrightarrow{CP_o}, \vec{n}_z \cdot \overrightarrow{CP_o}) \quad (4.1)$$

Figura 13 – Representação considerada para o modelo matemático

Fonte: Saha, Natraj e Waharte (2014).

Saha, Natraj e Waharte (2014) assumem que o eixo z da câmera ($\vec{n}_z$) está alinhado com o eixo z do mundo real e considera a rotação apenas em torno do eixo z (*yaw*). Formalmente, é definido apenas que:

$$\vec{n}_x = a\vec{x} + b\vec{y}, \quad \vec{n}_y = -b\vec{x} + a\vec{y} \quad \text{onde} \quad \sqrt{a^2 + b^2} = 1 \quad (4.2)$$

Por fim, é proposta a formulação de um sistema de equações que, baseado em pontos de interesse obtidos de duas imagens consecutivas e a posição e atitude do VANT inferidas pela integração de dados da IMU, possibilitam o cálculo da posição 3D dos pontos encontrados.

No entanto, considerando a dinâmica de movimentação dos VANTs não é razoável assumir que o eixo z da câmera ($\vec{n_z}$) está sempre alinhado com o eixo z do mundo real. Com isto, é necessário considerar também a rotação em torno do eixo y (*pitch*) e em torno do eixo x (*roll*).

Para tal, foram realizadas, como parte das contribuições desta tese de doutorado, adaptações no modelo apresentado para permitir a definição de extensões a fim de tornar o cálculo da posição tridimensional dos obstáculos mais condizente com a dinâmica de voo de um VANT real.

Desta forma, nesta tese é realizada, para efeitos de cálculo da posição do obstáculo, a inclusão dos ângulos de *pitch* e *roll*.

4.6.1 Inclusão do Ângulo de Pitch

Para realizar a inclusão do ângulo de *pitch* foi considerado o sistema de coordenadas *north, east, down (ned)* onde o eixo x aponta para o norte, o eixo y aponta para leste e o eixo z aponta para baixo. Este sistema de coordenadas foi escolhido pois é o sistema usado para observação da movimentação de VANTs, por ser inercial e fixado na Terra, do ponto de vista de um observador externo. O sistema de coordenadas utilizado é ilustrado na Figura 14.

Figura 14 – Representação do sistema de coordenadas considerado para o VANT

Fonte: Elaborada pelo autor.

Considerando o sistema de coordenadas apresentado na Figura 14, a fórmula de projeção

baseada em proporcionalidade do triângulo pode ser descrita como:

$$P_I(x_I, y_I) = \frac{f}{\vec{n}_x \cdot \vec{CP}_o} (\vec{n}_y \cdot \vec{CP}_o, \vec{n}_z \cdot \vec{CP}_o) \quad (4.3)$$

Para obter os vetores de orientação da câmera ($\vec{n}_x$, $\vec{n}_y$ e $\vec{n}_z$), considerando os ângulos $yaw = \gamma$ e $pitch = \beta$, foram utilizadas as matrizes de rotação $R_z(\gamma)$ e $R_y(\beta)$ como seguem:

$$R_y(\beta) = \begin{bmatrix} \cos(\beta) & 0 & -\sin(\beta) \\ 0 & 1 & 0 \\ \sin(\beta) & 0 & \cos(\beta) \end{bmatrix}$$

$$R_z(\gamma) = \begin{bmatrix} \cos(\gamma) & \sin(\gamma) & 0 \\ -\sin(\gamma) & \cos(\gamma) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$R_y(\beta)R_z(\gamma) = \begin{bmatrix} \cos(\beta)\cos(\gamma) & \cos(\beta)\sin(\gamma) & -\sin(\beta) \\ -\sin(\gamma) & \cos(\gamma) & 0 \\ \sin(\beta)\cos(\gamma) & \sin(\beta)\sin(\gamma) & \cos(\beta) \end{bmatrix}$$

Desta forma, $\vec{n}_x$, $\vec{n}_y$ e $\vec{n}_z$ podem ser definidos como:

$$\vec{n}_x = (\cos(\beta)\cos(\gamma), \cos(\beta)\sin(\gamma), -\sin(\beta)) \quad (4.4)$$

$$\vec{n}_y = (-\sin(\gamma), \cos(\gamma), 0) \quad (4.5)$$

$$\vec{n}_z = (\sin(\beta)\cos(\gamma), \sin(\beta)\sin(\gamma), \cos(\beta)) \quad (4.6)$$

Para simplificar, pode-se considerar:

$$a = \cos(\gamma)$$

$$b = \sin(\gamma)$$

$$c = \cos(\beta)$$

$$d = \sin(\beta)$$

Portanto, considerando que um *frame* foi capturado no instante de tempo t , a partir dos vetores de orientação da câmera (4.4,4.5,4.6) e da equação de projeção (4.3), obtém-se:

$$x_{I_t} = \frac{(-b_t)(x_O - x_t) + (a_t)(y_O - y_t)}{a_t c_t (x_O - x_t) + b_t c_t (y_O - y_t) - d_t (z_O - z_t)} f_x \quad (4.7)$$

$$y_{I_t} = \frac{(a_t d_t)(x_O - x_t) + (b_t d_t)(y_O - y_t) + (c_t)(z_O - z_t)}{a_t c_t(x_O - x_t) + b_t c_t(y_O - y_t) - d_t(z_O - z_t)} f_y \quad (4.8)$$

Para gerar o sistema de equações que permite encontrar a posição do obstáculo (x_O , y_O , z_O) pode-se dividir 4.7 por 4.8 obtendo-se:

$$\frac{x_{I_t}}{y_{I_t}} = \frac{f_x[-b_t(x_O - x_t) + a_t(y_O - y_t)]}{f_y[d_t a_t(x_O - x_t) + d_t b_t(y_O - y_t) + c_t(z_O - z_t)]}$$

Isolando x_O , y_O e z_O tem-se:

$$x_O(x_{I_t} f_y d_t a_t + y_{I_t} f_x b_t) + y_O(x_{I_t} f_y d_t b_t - y_{I_t} f_x a_t) + z_O(x_{I_t} f_y c_t) = \\ x_t(x_{I_t} f_y d_t a_t + y_{I_t} f_x b_t) + y_t(x_{I_t} f_y d_t b_t - y_{I_t} f_x a_t) + z_t(x_{I_t} f_y c_t) \quad (4.9)$$

Considerando 4.9 nos tempos $t = 1$ e $t = 2$, tem-se:

$$x_O(x_{I_1} f_y d_1 a_1 + y_{I_1} f_x b_1) + y_O(x_{I_1} f_y d_1 b_1 - y_{I_1} f_x a_1) + z_O(x_{I_1} f_y c_1) = \\ x_1(x_{I_1} f_y d_1 a_1 + y_{I_1} f_x b_1) + y_1(x_{I_1} f_y d_1 b_1 - y_{I_1} f_x a_1) + z_1(x_{I_1} f_y c_1) \quad (4.10)$$

$$x_O(x_{I_2} f_y d_2 a_2 + y_{I_2} f_x b_2) + y_O(x_{I_2} f_y d_2 b_2 - y_{I_2} f_x a_2) + z_O(x_{I_2} f_y c_2) = \\ x_2(x_{I_2} f_y d_2 a_2 + y_{I_2} f_x b_2) + y_2(x_{I_2} f_y d_2 b_2 - y_{I_2} f_x a_2) + z_2(x_{I_2} f_y c_2) \quad (4.11)$$

Como no método proposto por [Saha, Natraj e Waharte \(2014\)](#), para eliminar uma equação do sistema, similarmente os últimos termos do sistema de equações são gerados subtraindo 4.8 em $t = 2$ (4.13) de 4.8 em $t = 1$ (4.12).

$$x_O(a_1 c_1 y_{I_1} - f_y(a_1 d_1)) + y_O(b_1 c_1 y_{I_1} - f_y(b_1 d_1)) + z_O(-d_1 y_{I_1} - f_y c_1) = \\ x_1(a_1 c_1 y_{I_1} - f_y(a_1 d_1)) + y_1(b_1 c_1 y_{I_1} - f_y(b_1 d_1)) + z_1(-d_1 y_{I_1} - f_y c_1) \quad (4.12)$$

$$x_O(a_2 c_2 y_{I_2} - f_y(a_2 d_2)) + y_O(b_2 c_2 y_{I_2} - f_y(b_2 d_2)) + z_O(-d_2 y_{I_2} - f_y c_2) = \\ x_2(a_2 c_2 y_{I_2} - f_y(a_2 d_2)) + y_2(b_2 c_2 y_{I_2} - f_y(b_2 d_2)) + z_2(-d_2 y_{I_2} - f_y c_2) \quad (4.13)$$

Desta forma, realizando a subtração de 4.13 de 4.12 obtém-se:

$$\begin{aligned}
& x_O(a_1c_1y_{I_1} - f_y(a_1d_1) - a_2c_2y_{I_2} + f_y(a_2d_2)) + \\
& \quad y_O(b_1c_1y_{I_1} - f_y(b_1d_1) - b_2c_2y_{I_2} + f_y(b_2d_2)) + \\
& \quad z_O(-d_1y_{I_1} - f_yc_1 + d_2y_{I_2} + f_yc_2) = \\
& [x_1(a_1c_1y_{I_1} - f_y(a_1d_1)) + y_1(b_1c_1y_{I_1} - f_y(b_1d_1)) + z_1(-d_1y_{I_1} - f_yc_1)] - \\
& [x_2(a_2c_2y_{I_2} - f_y(a_2d_2)) + y_2(b_2c_2y_{I_2} - f_y(b_2d_2)) + z_2(-d_2y_{I_2} - f_yc_2)] \quad (4.14)
\end{aligned}$$

Conhecendo a posição e atitude (*yaw*, *pitch*) da câmera nos momentos $t = 1$ e $t = 2$, tem-se: $C_1 = (x_1, y_1, z_1)$ e $C_2 = (x_2, y_2, z_2)$; yaw_1 e yaw_2 ; e $pitch_1$ e $pitch_2$.

A partir dos ângulos de *yaw*, *pitch* em $t = 1$ e $t = 2$ tem-se $a_1, b_1, c_1, d_1, a_2, b_2, c_2, d_2$. Com isto, é possível montar um sistema de equações e encontrar os valores de posição do objeto $P_O = (x_O, y_O, z_O)$.

Para isto, a partir de 4.10 e 4.11 tem-se:

$$e_1 = x_{I_1}f_yd_1a_1 + y_{I_1}f_xb_1$$

$$e_2 = x_{I_1}f_yd_1b_1 - y_{I_1}f_xa_1$$

$$e_3 = x_{I_1}f_yc_1$$

$$e_4 = x_{I_2}f_yd_2a_2 + y_{I_2}f_xb_2$$

$$e_5 = x_{I_2}f_yd_2b_2 - y_{I_2}f_xa_2$$

$$e_6 = x_{I_2}f_yc_2$$

$$B1 = x_1(x_{I_1}f_yd_1a_1 + y_{I_1}f_xb_1) + y_1(x_{I_1}f_yd_1b_1 - y_{I_1}f_xa_1) + z_1(x_{I_1}f_yc_1)$$

$$B2 = x_2(x_{I_2}f_yd_2a_2 + y_{I_2}f_xb_2) + y_2(x_{I_2}f_yd_2b_2 - y_{I_2}f_xa_2) + z_2(x_{I_2}f_yc_2)$$

A partir de 4.14:

$$e_7 = a_1c_1y_{I_1} - f_y(a_1d_1) - a_2c_2y_{I_2} + f_y(a_2d_2)$$

$$e_8 = b_1c_1y_{I_1} - f_y(b_1d_1) - b_2c_2y_{I_2} + f_y(b_2d_2)$$

$$e_9 = -d_1y_{I_1} - f_yc_1 + d_2y_{I_2} + f_yc_2$$

$$\begin{aligned}
B3 = & [x_1(a_1c_1y_{I_1} - f_y(a_1d_1)) + y_1(b_1c_1y_{I_1} - f_y(b_1d_1)) + \\
& z_1(-d_1y_{I_1} - f_yc_1)] - [x_2(a_2c_2y_{I_2} - f_y(a_2d_2)) + y_2(b_2c_2y_{I_2} - f_y(b_2d_2)) + \\
& z_2(-d_2y_{I_2} - f_yc_2)]
\end{aligned}$$

Com isto, similarmente ao proposto por (SAHA; NATRAJ; WAHARTE, 2014), é possível encontrar a posição do objeto ao resolver o sistema de equações.

$$Ax = B$$

onde,

$$A = \begin{bmatrix} e_1 & e_2 & e_3 \\ e_4 & e_5 & e_6 \\ e_7 & e_8 & e_9 \end{bmatrix}, x = \begin{bmatrix} x_O \\ y_O \\ z_O \end{bmatrix}, B = \begin{bmatrix} B_1 \\ B_2 \\ B_3 \end{bmatrix}$$

4.6.2 Inclusão do Ângulo de roll

De forma análoga ao realizado na inclusão do ângulo de *pitch*, para a inclusão do ângulo de *roll* é necessário considerar a matriz de rotação em torno deste ângulo. Desta forma, considerando $\gamma = \text{yaw}$ e $\beta = \text{pitch}$ e $\alpha = \text{roll}$:

$$R_x(\alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\alpha) & \sin(\alpha) \\ 0 & -\sin(\alpha) & \cos(\alpha) \end{bmatrix}$$

$$R_y(\beta) = \begin{bmatrix} \cos(\beta) & 0 & -\sin(\beta) \\ 0 & 1 & 0 \\ \sin(\beta) & 0 & \cos(\beta) \end{bmatrix}$$

$$R_z(\gamma) = \begin{bmatrix} \cos(\gamma) & \sin(\gamma) & 0 \\ -\sin(\gamma) & \cos(\gamma) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$R_\alpha^x R_\beta^y R_\gamma^z = \begin{bmatrix} \cos(\beta)\cos(\gamma) & \cos(\beta)\sin(\gamma) & -\sin(\beta) \\ \sin(\alpha)\sin(\beta)\cos(\gamma) - \cos(\alpha)\sin(\gamma) & \sin(\alpha)\sin(\beta)\sin(\gamma) + \cos(\alpha)\cos(\gamma) & \sin(\alpha)\cos(\beta) \\ \cos(\alpha)\sin(\beta)\cos(\gamma) + \sin(\alpha)\sin(\gamma) & \cos(\alpha)\sin(\beta)\sin(\gamma) - \sin(\alpha)\cos(\gamma) & \cos(\alpha)\cos(\beta) \end{bmatrix}$$

Portanto, considerando o ângulo de *roll* os vetores $\vec{n}\hat{x}$, $\vec{n}\hat{y}$ e $\vec{n}\hat{z}$ passam a ser:

$$\vec{n}\hat{x} = (\cos(\beta)\cos(\gamma), \cos(\beta)\sin(\gamma), -\sin(\beta)) \quad (4.15)$$

$$\vec{n}\hat{y} = (\sin(\alpha)\sin(\beta)\cos(\gamma) - \cos(\alpha)\sin(\gamma), \sin(\alpha)\sin(\beta)\sin(\gamma) + \cos(\alpha)\cos(\gamma), \sin(\alpha)\cos(\beta)) \quad (4.16)$$

$$\vec{n}\vec{z} = (\cos(\alpha)\sin(\beta)\cos(\gamma) + \sin(\alpha)\sin(\gamma), \cos(\alpha)\sin(\beta)\sin(\gamma) - \sin(\alpha)\cos(\gamma), \cos(\alpha)\cos(\beta)) \quad (4.17)$$

Para simplificar, considera-se:

$$a = \cos(\gamma)$$

$$b = \sin(\gamma)$$

$$c = \cos(\beta)$$

$$d = \sin(\beta)$$

$$e = \cos(\alpha)$$

$$f = \sin(\alpha)$$

Da mesma forma, considerando que um *frame* seja capturado no instante t , a partir de 4.3 obtém-se:

$$x_{I_t} = \frac{(a_t d_t f_t - b_t e_t)(x_O - x_t) + (b_t d_t f_t + a_t e_t)(y_O - y_t) + c_t f_t(z_O - z_t)}{a_t c_t(x_O - x_t) + b_t c_t(y_O - y_t) - d_t(z_O - z_t)} f_x \quad (4.18)$$

$$y_{I_t} = \frac{(a_t d_t e_t + b_t f_t)(x_O - x_t) + (b_t d_t e_t - a_t f_t)(y_O - y_t) + (c_t e_t)(z_O - z_t)}{a_t c_t(x_O - x_t) + b_t c_t(y_O - y_t) - d_t(z_O - z_t)} f_y \quad (4.19)$$

Isolando x_O , y_O e z_O em 4.18 e 4.19 tem-se respectivamente:

$$\begin{aligned} x_O(a_t c_t x_{I_t} - f_x(a_t d_t f_t - b_t e_t)) + y_O(b_t c_t x_{I_t} - f_x(b_t d_t f_t + a_t e_t)) + \\ z_O(-d_t x_{I_t} - f_x c_t f_t) = x_t(a_t c_t x_{I_t} - f_x(a_t d_t f_t - b_t e_t)) + \\ y_t(b_t c_t x_{I_t} - f_x(b_t d_t f_t + a_t e_t)) + z_t(-d_t x_{I_t} - f_x c_t f_t) \end{aligned} \quad (4.20)$$

$$\begin{aligned} x_O(a_t c_t y_{I_t} - f_y(a_t d_t e_t + b_t f_t)) + y_O(b_t c_t y_{I_t} - f_y(b_t d_t e_t - a_t f_t)) + \\ z_O(-d_t y_{I_t} - f_y c_t e_t) = x_t(a_t c_t y_{I_t} - f_y(a_t d_t e_t + b_t f_t)) + \\ y_t(b_t c_t y_{I_t} - f_y(b_t d_t e_t - a_t f_t)) + z_t(-d_t y_{I_t} - f_y c_t e_t) \end{aligned} \quad (4.21)$$

Mais uma vez, dividindo x_{I_t} (4.18) por y_{I_t} (4.19) obtém-se:

$$\begin{aligned}
& x_O(x_{I_t}f_ya_td_te_t + x_{I_t}f_yb_tf_t - y_{I_t}f_xa_td_tf_t + y_{I_t}f_xb_te_t) + \\
& y_O(x_{I_t}f_yb_td_te_t - x_{I_t}f_ya_tf_t - y_{I_t}f_xb_td_tf_t - y_{I_t}f_xa_te_t) + \\
& z_O(x_{I_t}f_yc_te_t - y_{I_t}f_xc_tf_t) = \\
& x_t(x_{I_t}f_ya_td_te_t + x_{I_t}f_yb_tf_t - y_{I_t}f_xa_td_tf_t + y_{I_t}f_xb_te_t) + \\
& y_t(x_{I_t}f_yb_td_te_t - x_{I_t}f_ya_tf_t - y_{I_t}f_xb_td_tf_t - y_{I_t}f_xa_te_t) + \\
& z_t(x_{I_t}f_yc_te_t - y_{I_t}f_xc_tf_t); \quad (4.22)
\end{aligned}$$

De maneira similar a realizada na adição do ângulo de *yaw*, os últimos termos do sistema de equação podem ser obtidos subtraindo 4.19 no tempo $t = 2$ (4.24) de 4.19 no tempo $t = 1$ (4.23).

$$\begin{aligned}
& x_O(a_1c_1y_{I_1} - f_y(a_1d_1e_1 + b_1f_1)) + y_O(b_1c_1y_{I_1} - f_y(b_1d_1e_1 - a_1f_1)) + \\
& z_O(-d_1y_{I_1} - f_yc_1e_1) = x_1(a_1c_1y_{I_1} - f_y(a_1d_1e_1 + b_1f_1)) + \\
& y_1(b_1c_1y_{I_1} - f_y(b_1d_1e_1 - a_1f_1)) + z_1(-d_1y_{I_1} - f_yc_1e_1) \quad (4.23)
\end{aligned}$$

$$\begin{aligned}
& x_O(a_2c_2y_{I_2} - f_y(a_2d_2e_2 + b_2f_2)) + y_O(b_2c_2y_{I_2} - f_y(b_2d_2e_2 - a_2f_2)) + \\
& z_O(-d_2y_{I_2} - f_yc_2e_2) = x_2(a_2c_2y_{I_2} - f_y(a_2d_2e_2 + b_2f_2)) + \\
& y_2(b_2c_2y_{I_2} - f_y(b_2d_2e_2 - a_2f_2)) + z_2(-d_2y_{I_2} - f_yc_2e_2) \quad (4.24)
\end{aligned}$$

Obtém-se:

$$\begin{aligned}
& x_O(a_1c_1y_{I_1} - f_y(a_1d_1e_1 + b_1f_1) - a_2c_2y_{I_2} + f_y(a_2d_2e_2 + b_2f_2)) + \\
& y_O(b_1c_1y_{I_1} - f_y(b_1d_1e_1 - a_1f_1) - b_2c_2y_{I_2} + f_y(b_2d_2e_2 - a_2f_2)) + \\
& z_O(-d_1y_{I_1} - f_yc_1e_1 + d_2y_{I_2} + f_yc_2e_2) = \\
& [x_1(a_1c_1y_{I_1} - f_y(a_1d_1e_1 + b_1f_1)) + y_1(b_1c_1y_{I_1} - f_y(b_1d_1e_1 - a_1f_1)) + \\
& z_1(-d_1y_{I_1} - f_yc_1e_1)] - [x_2(a_2c_2y_{I_2} - f_y(a_2d_2e_2 + b_2f_2)) + \\
& y_2(b_2c_2y_{I_2} - f_y(b_2d_2e_2 - a_2f_2)) + z_2(-d_2y_{I_2} - f_yc_2e_2)] \quad (4.25)
\end{aligned}$$

Desta forma, a partir de 4.22 para $t = 1$ e $t = 2$; e 4.25 obtém-se:

$$\begin{aligned}
g_1 &= x_{I_1}f_ya_1d_1e_1 + x_{I_1}f_yb_1f_1 - y_{I_1}f_xa_1d_1f_1 + y_{I_1}f_xb_1e_1 \\
g_2 &= x_{I_1}f_yb_1d_1e_1 - x_{I_1}f_ya_1f_1 - y_{I_1}f_xb_1d_1f_1 - y_{I_1}f_xa_1e_1
\end{aligned}$$

$$g_3 = x_{I_1} f_y c_1 e_1 - y_{I_1} f_x c_1 f_1$$

$$g_4 = x_{I_2} f_y a_2 d_2 e_2 + x_{I_2} f_y b_2 f_2 - y_{I_2} f_x a_2 d_2 f_2 + y_{I_2} f_x b_2 e_2$$

$$g_5 = x_{I_2} f_y b_2 d_2 e_2 - x_{I_2} f_y a_2 f_2 - y_{I_2} f_x b_2 d_2 f_2 - y_{I_2} f_x a_2 e_2$$

$$g_6 = x_{I_2} f_y c_2 e_2 - y_{I_2} f_x c_2 f_2$$

$$g_7 = a_1 c_1 y_{I_1} - f_y (a_1 d_1 e_1 + b_1 f_1) - a_2 c_2 y_{I_2} + f_y (a_2 d_2 e_2 + b_2 f_2)$$

$$g_8 = b_1 c_1 y_{I_1} - f_y (b_1 d_1 e_1 - a_1 f_1) - b_2 c_2 y_{I_2} + f_y (b_2 d_2 e_2 - a_2 f_2)$$

$$g_9 = -d_1 y_{I_1} - f_y c_1 e_1 + d_2 y_{I_2} + f_y c_2 e_2$$

$$\begin{aligned} B_1 = & x_1 (x_{I_1} f_y a_1 d_1 e_1 + x_{I_1} f_y b_1 f_1 - y_{I_1} f_x a_1 d_1 f_1 + y_{I_1} f_x b_1 e_1) + \\ & y_1 (x_{I_1} f_y b_1 d_1 e_1 - x_{I_1} f_y a_1 f_1 - y_{I_1} f_x b_1 d_1 f_1 - y_{I_1} f_x a_1 e_1) + \\ & z_1 (x_{I_1} f_y c_1 e_1 - y_{I_1} f_x c_1 f_1) \end{aligned}$$

$$\begin{aligned} B_2 = & x_2 (x_{I_2} f_y a_2 d_2 e_2 + x_{I_2} f_y b_2 f_2 - y_{I_2} f_x a_2 d_2 f_2 + y_{I_2} f_x b_2 e_2) + \\ & y_2 (x_{I_2} f_y b_2 d_2 e_2 - x_{I_2} f_y a_2 f_2 - y_{I_2} f_x b_2 d_2 f_2 - y_{I_2} f_x a_2 e_2) + \\ & z_2 (x_{I_2} f_y c_2 e_2 - y_{I_2} f_x c_2 f_2) \end{aligned}$$

$$\begin{aligned} B_3 = & [x_1 (a_1 c_1 y_{I_1} - f_y (a_1 d_1 e_1 + b_1 f_1)) + y_1 (b_1 c_1 y_{I_1} - f_y (b_1 d_1 e_1 - a_1 f_1)) + \\ & z_1 (-d_1 y_{I_1} - f_y c_1 e_1)] - [x_2 (a_2 c_2 y_{I_2} - f_y (a_2 d_2 e_2 + b_2 f_2)) + \\ & y_2 (b_2 c_2 y_{I_2} - f_y (b_2 d_2 e_2 - a_2 f_2)) + z_2 (-d_2 y_{I_2} - f_y c_2 e_2)] \end{aligned}$$

Com isto, de maneira similar ao realizado na inclusão do ângulo de *pitch*, é possível encontrar a posição do objeto ao resolver o sistema de equações.

$$Ax = B$$

onde,

$$A = \begin{bmatrix} g_1 & g_2 & g_3 \\ g_4 & g_5 & g_6 \\ g_7 & g_8 & g_9 \end{bmatrix}, x = \begin{bmatrix} x_O \\ y_O \\ z_O \end{bmatrix}, B = \begin{bmatrix} B_1 \\ B_2 \\ B_3 \end{bmatrix}$$

4.6.3 Melhorias no cálculo da posição do objeto

No método proposto inicialmente por (SAHA; NATRAJ; WAHARTE, 2014), para evitar a formação de um sistema de equações com quatro equações e três variáveis (considerando a Equação 4.1 é possível obter as equações x_{I_t} e y_{I_t} para $t = 1$ e $t = 2$) foram realizadas divisões e subtrações entre as equações do sistema, reduzindo-o assim a três equações equivalentes.

Para a inclusão do ângulo de *pitch* e *roll* aqui realizadas utilizou-se de parte destas técnicas. No entanto, deve-se considerar que isto leva a uma perda de precisão no cálculo da posição do objeto.

Para resolver este problema, pode-se considerar todas as equações possíveis a partir de 4.20 e 4.21, em $t = 1$ e $t = 2$, gerando a seguinte situação:

$$g_1 = a_1 c_1 x_{I_1} - f_x(a_1 d_1 f_1 - b_1 e_1)$$

$$g_2 = b_1 c_1 x_{I_1} - f_x(b_1 d_1 f_1 + a_1 e_1)$$

$$g_3 = -d_1 x_{I_1} - f_x c_1 f_1$$

$$g_4 = a_1 c_1 y_{I_1} - f_y(a_1 d_1 e_1 + b_1 f_1)$$

$$g_5 = b_1 c_1 y_{I_1} - f_y(b_1 d_1 e_1 - a_1 f_1)$$

$$g_6 = -d_1 y_{I_1} - f_y c_1 e_1$$

$$g_7 = a_2 c_2 x_{I_2} - f_x(a_2 d_2 f_2 - b_2 e_2)$$

$$g_8 = b_2 c_2 x_{I_2} - f_x(b_2 d_2 f_2 + a_2 e_2)$$

$$g_9 = -d_2 x_{I_2} - f_x c_2 f_2$$

$$g_{10} = a_2 c_2 y_{I_2} - f_y(a_2 d_2 e_2 + b_2 f_2)$$

$$g_{11} = b_2 c_2 y_{I_2} - f_y(b_2 d_2 e_2 - a_2 f_2)$$

$$g_{12} = -d_2 y_{I_2} - f_y c_2 e_2$$

$$B_1 = x_1(a_1 c_1 x_{I_1} - f_x(a_1 d_1 f_1 - b_1 e_1)) + y_1(b_1 c_1 x_{I_1} - f_x(b_1 d_1 f_1 + a_1 e_1)) + z_1(-d_1 x_{I_1} - f_x c_1 f_1)$$

$$B_2 = x_1(a_1 c_1 y_{I_1} - f_y(a_1 d_1 e_1 + b_1 f_1)) + y_1(b_1 c_1 y_{I_1} - f_y(b_1 d_1 e_1 - a_1 f_1)) + z_1(-d_1 y_{I_1} - f_y c_1 e_1)$$

$$B_3 = x_2(a_2 c_2 x_{I_2} - f_x(a_2 d_2 f_2 - b_2 e_2)) + y_2(b_2 c_2 x_{I_2} - f_x(b_2 d_2 f_2 + a_2 e_2)) + z_2(-d_2 x_{I_2} - f_x c_2 f_2)$$

$$B_4 = x_2(a_2c_2y_{I_2} - f_y(a_2d_2e_2 + b_2f_2)) + y_2(b_2c_2y_{I_2} - f_y(b_2d_2e_2 - a_2f_2)) + z_2(-d_2y_{I_2} - f_yc_2e_2)$$

$$A = \begin{bmatrix} g_1 & g_2 & g_3 \\ g_4 & g_5 & g_6 \\ g_7 & g_8 & g_9 \\ g_{10} & g_{11} & g_{12} \end{bmatrix}, x = \begin{bmatrix} x_O \\ y_O \\ z_O \end{bmatrix}, B = \begin{bmatrix} B_1 \\ B_2 \\ B_3 \\ B_4 \end{bmatrix}$$

Isto pode ser resolvido utilizando a técnica de Mínimos Quadrados:

$$x = (A^T A)^{-1} A^T B$$

Esta nova abordagem permite também a utilização de n frames ($n \geq 2$) para o cálculo da posição do obstáculo, gerando ainda mais precisão e robustez em relação ao ruído dos dados de posicionamento e imagem.

Para isto, pode-se definir de maneira mais genérica:

$$x_I(t)_{x_o} = a_t c_t x_{I_t} - f_x(a_t d_t f_t - b_t e_t)$$

$$x_I(t)_{y_o} = b_t c_t x_{I_t} - f_x(b_t d_t f_t + a_t e_t)$$

$$x_I(t)_{z_o} = -d_t x_{I_t} - f_x c_t f_t$$

$$y_I(t)_{x_o} = a_t c_t y_{I_t} - f_y(a_t d_t e_t + b_t f_t)$$

$$y_I(t)_{y_o} = b_t c_t y_{I_t} - f_y(b_t d_t e_t - a_t f_t)$$

$$y_I(t)_{z_o} = -d_t y_{I_t} - f_y c_t e_t$$

$$B_{x_I}(t) = x_t(a_t c_t x_{I_t} - f_x(a_t d_t f_t - b_t e_t)) + y_t(b_t c_t x_{I_t} - f_x(b_t d_t f_t + a_t e_t)) + z_t(-d_t x_{I_t} - f_x c_t f_t)$$

$$B_{y_I}(t) = x_t(a_t c_t y_{I_t} - f_y(a_t d_t e_t + b_t f_t)) + y_t(b_t c_t y_{I_t} - f_y(b_t d_t e_t - a_t f_t)) + z_t(-d_t y_{I_t} - f_y c_t e_t)$$

E para n frames tem-se:

$$A = \begin{bmatrix} x_I(1)_{x_o} & x_I(1)_{y_o} & x_I(1)_{z_o} \\ y_I(1)_{x_o} & y_I(1)_{y_o} & y_I(1)_{z_o} \\ x_I(2)_{x_o} & x_I(2)_{y_o} & x_I(2)_{z_o} \\ y_I(2)_{x_o} & y_I(2)_{y_o} & y_I(2)_{z_o} \\ \dots & \dots & \dots \\ \dots & \dots & \dots \\ x_I(n)_{x_o} & x_I(n)_{y_o} & x_I(n)_{z_o} \\ y_I(n)_{x_o} & y_I(n)_{y_o} & y_I(n)_{z_o} \end{bmatrix}, x = \begin{bmatrix} x_O \\ y_O \\ z_O \end{bmatrix}, B = \begin{bmatrix} B_{x_I}(1) \\ B_{y_I}(1) \\ B_{x_I}(2) \\ B_{y_I}(2) \\ \dots \\ \dots \\ B_{x_I}(n) \\ B_{y_I}(n) \end{bmatrix}$$

As principais técnicas desenvolvidas foram comparadas com o estado da arte e os resultados são apresentados na seção 5.1.

4.7 Controle de Missão e Navegação - MOSAIC

O módulo de **Controle de Missão e Navegação do MOSAIC** recebe como entrada uma lista de *waypoints* que deve ser cumprida durante a execução da missão. Para permitir o desvio dos obstáculos, estes *waypoints* são registrados no módulo de **Desvio** que gera os vetores de velocidade a serem cumpridos pelo VANT a fim de chegar ao *waypoint* definido enquanto evita os obstáculos registrados. Cabe ressaltar que os obstáculos são incluídos ou atualizados a medida que são detectados (oriundos do módulo de **Posição do Obstáculo**).

Algoritmo 1 – Controle de Missão e Navegação com Desvio de Obstáculos - MOSAIC

```

1: para todo waypoint  $\in$  waypointList faça
2:   posicao  $\leftarrow$  vant.posicaoAtual()
3:   desvio.defineObjetivo(waypoint)
4:   desvio.definePosicao(posicao)
5:   enquanto posicao  $\neq$  waypoint faça
6:     obstaculos  $\leftarrow$  posicaoObstaculo.calcular()
7:     para todo o  $\in$  obstaculos faça
8:       se desvio.existe(o) então
9:         desvio.atualizar(o)
10:      senão
11:        desvio.inserir(o)
12:      fim se
13:    fim para
14:    v  $\leftarrow$  desvio.calculaVetorVelocidade()
15:    vant.defineVetorVelocidade(v)
16:    posicao  $\leftarrow$  vant.posicaoAtual()
17:    velocidade  $\leftarrow$  vant.VelocidadeAtual()
18:    desvio.definePosicao(posicao)
19:    desvio.defineVelocidade(velocidade)
20:  fim enquanto
21: fim para

```

No Algoritmo 1 são ilustrados os passos realizados pelo **controle de missão e navegação do MOSAIC**. Cada *waypoint* é tratado de forma individual e, enquanto o VANT não alcançou este *waypoint*, os obstáculos conhecidos são obtidos do módulo de **Posição do Obstáculo** e registrados no módulo de **Desvio**. Com base nestas informações, o módulo de **Desvio** retorna um vetor de velocidade a ser seguido pelo VANT e este vetor é repassado ao controle de navegação do VANT utilizando o protocolo MAVLINK. O VANT irá tentar seguir o vetor de velocidade definido, mas este pode não ser cumprido por restrições do modelo de voo da aeronave. Com isto, a posição e a velocidade que o VANT conseguiu atingir é requisitada

(também utilizando o protocolo MAVLINK) e repassada novamente ao módulo de **Desvio**, que leva isto em consideração para a definição do próximo vetor de velocidade.

Importante ressaltar que o módulo de **Controle de Missão e Navegação do MOSAIC** não substitui os respectivos módulos presentes no VANT, mas sim, trabalha em conjunto com estes para acrescentar a funcionalidade de desvio de obstáculos.

4.8 Módulo de Desvio

Como o MOSAIC calcula a posição tridimensional dos obstáculos é possível realizar o desvio utilizando uma abordagem do tipo *reciprocal n-body collision avoidance* previamente descrita por [Berg et al. \(2011\)](#) (mais detalhes no [Capítulo 2](#)). Considerando as fortes premissas estabelecidas pela abordagem de [Berry et al. \(2011\)](#), e que o MOSAIC, com base na pesquisa realizada na literatura aberta, é o único método que atende a maior parte destas premissas no cenário de VANTs de pequeno porte, permitindo a utilização conjunta dos métodos, pode-se assumir que esse sistema como um todo constitui em um grande avanço ao estado da arte.

O módulo de **Desvio** trabalha em conjunto com o módulo de **Controle de Missão e Navegação do MOSAIC**. Desta forma, ele recebe informações em relação a um objetivo imediato e todos os obstáculos detectados. Com base nestas informações, é calculado um vetor de velocidade livre de obstáculos e em direção ao objetivo para ser aplicado ao VANT durante um intervalo de tempo Δt . Este vetor é enviado ao módulo de controle de missão e navegação do veículo aéreo, para ser aplicado no mesmo.

Importante notar que o vetor de velocidade enviado ao módulo de controle do VANT deve respeitar as restrições físicas, como a aceleração máxima que o VANT pode assumir. Desta forma, a versão modificada do ORCA 3D utilizada considera também esta restrição ([Sá De Mello et al., 2015](#)).

Por fim, a velocidade real alcançada pelo VANT, bem como sua posição, é repassada ao módulo de **Desvio** que também utiliza esses dados na geração do próximo vetor de velocidade.

4.9 OASIs - *Obstacle Avoidance Simulator*

Apesar de existirem alguns simuladores para VANTs ([BU et al., 2015](#); [FILATOV; DEVYATKIN; POLYAKHOV, 2017](#)), eles geralmente são focados na dinâmica do voo e não fornecem base para simular a detecção e desvio dos obstáculos baseada em visão monocular. Desta forma foi desenvolvido o *Obstacle Avoidance Simulator* (OASIS), um simulador focado na simulação de algoritmos de detecção e desvio de obstáculos. Para realizar a simulação da parte da dinâmica do voo é utilizado o DroneKit². Com isto, o MOSAIC pode ser integrado a este simulador para validar os módulos de detecção e desvio de obstáculo.

² <http://dronekit.io/>

O DroneKit é uma API que permite a programação de VANTs que utilizam o protocolo MAVLINK para comunicação. Ele possui um simulador integrado que fornece as dinâmicas de movimentação dos principais tipos de veículos e permite testar aplicativos desenvolvidos. No entanto, para realizar a simulação de detecção e desvio de obstáculos utilizando imagens é necessário que o simulador tenha a capacidade de gerar as imagens relativas ao voo.

O OASIS recebe, através do protocolo MAVLINK, dados relativos ao posicionamento e atitude do VANT para posicionar o VANT virtual em um cenário tridimensional desenvolvido na *engine* de jogos *Unity*³. Uma câmera integrada ao VANT virtual gera um *link* de vídeo que pode ser utilizado pelo MOSAIC para realizar a detecção e o desvio de um obstáculo virtual.

Figura 15 – Simulação sendo executada no OASIS

(a) Simulador

(b) Visão do MOSAIC - Original

(c) Visão do MOSAIC - Obstáculo

Fonte: Elaborada pelo autor.

Na Figura 15a é ilustrada uma simulação utilizando o OASIS integrado ao MOSAIC. Pode-se observar o obstáculo virtual (representado por uma esfera) e o VANT virtual do tipo quadrotor em um cenário virtual. Na Figura 15b é ilustrada a visão do VANT obtida pelo *link* de

³ <https://unity3d.com/pt>

vídeo durante a simulação e na [Figura 15c](#) a posição do obstáculo detectado na imagem. Uma vez que os algoritmos, técnicas e abordagens utilizados pelo MOSAIC podem ser modificados, o OASIS passa a ser uma ferramenta importante para a verificação e validação em laboratório dos experimentos que podem ser realizados em campo *a posteriori*.

A integração do OASIS com o MOSAIC é ilustrada na [Figura 16](#) onde o Simulador do DroneKit (*DroneKit-SITL*) recebe os comandos da missão a ser executada provenientes do MOSAIC através do protocolo MAVLINK ao mesmo tempo que fornece dados de posicionamento e atitude ao MOSAIC e ao OASIS.

Figura 16 – Integração do OASIS ao MOSAIC

Fonte: Elaborada pelo autor.

O OASIS utiliza os dados recebidos para controlar um VANT virtual dentro do cenário 3D e fornece um *link* de vídeo para ser processado pelo MOSAIC. Para permitir a avaliação do erro no método criado para o cálculo da posição do obstáculo, bem como a eficácia do desvio, foi concebido um cenário que conta com um obstáculo de cor específica que pode ser detectado por meio de técnicas baseadas no espaço de cor ([Subseção 4.2.1](#)), e assim pode ser verificada e validado o OASIS.

4.10 Considerações finais

Este capítulo apresentou o MOSAIC, um sistema de detecção e desvio de obstáculos capaz de calcular a posição tridimensional dos obstáculos com acurácia ao utilizar a fusão de dados de posicionamento obtidos pelo GPS, dados de IMU e as posições dos obstáculos em uma sequência de imagens.

O cálculo da posição tridimensional do obstáculo com acurácia permite a utilização de uma variação do algoritmo ORCA 3D para a realização do desvio.

Adicionalmente foi desenvolvido o OASIs, um simulador focado na simulação de algoritmos de detecção e desvio de obstáculos que é capaz de fornecer um *link* de vídeo virtual para dados de uma simulação realizada com o DroneKit. Este *link* de vídeo pode ser utilizado pelo algoritmo de detecção para verificar o erro do cálculo de posição do obstáculo.

No [Capítulo 5](#) são descritos os resultados simulados e experimentais que investigam e validam o uso do MOSAIC.

RESULTADOS EXPERIMENTAIS

O MOSAIC, incluindo todos os seus módulos, foi validado por meio de experimentos fazendo uso de dados simulados, por meio do OASIS, e de voos reais, com o VANT de asa rotativa 3DR Solo ([3DR, 2017](#)).

Inicialmente foi realizada uma simulação utilizando o DroneKit e a fórmula de projeção para verificar o funcionamento do método de estimativa de posição do obstáculo. Posteriormente, o simulador OASIS foi empregado para o teste do método de detecção em conjunto com o método de estimativa de posição do obstáculo. Posteriormente foram realizados experimentos que averiguaram a efetividade do algoritmo de detecção e desvio por meio de um voo real, mas com presença de um obstáculo virtual. Neste caso foi verificado o funcionamento da implementação da abordagem *reciprocal n-body collision avoidance* já adaptada. Nesse sentido foram avaliados os métodos de detecção de obstáculos elaborados e implementados nesta tese fazendo uso da abordagem de desvio adaptada.

Esses resultados foram parcialmente publicados em [Chiaromonte e Castelo Branco \(2017\)](#) e [Sá De Mello et al. \(2015\)](#) e corroboram com o proposto para os módulos de **Deteção, Posição do Obstáculo e Estimativa de Posição do Obstáculo e Módulo de Desvio**.

Com intuito de permitir averiguar a efetividade da utilização de mais de uma métrica (principalmente no contexto do MOSAIC), foram efetuados testes em um cenário real do cálculo da intensidade do sinal como meio de detecção de obstáculos, validando assim o módulo *Pose Estimation*.

O MOSAIC foi implementado nas linguagens C++ e Python. Os experimentos simulados foram realizados em um computador com processador Intel Core i7-6700 de 4 GHz e 32 GB de memória RAM onde foi executado o simulador OASIS. Os experimentos reais foram realizados em um computador com processador Intel Core i5-7500U de 3,5 GHz e 16 GB de memória RAM. O veículo utilizado, com o já mencionado foi o Solo da 3DR, ilustrado na Figura [17](#).

Figura 17 – VANT 3DR Solo.

Fonte: [3DR \(2017\)](#).

5.1 Estimativa de Posição do Obstáculo

Para verificar o funcionamento do sistema de equações proposto, em comparação com o existente no estado da arte, inicialmente foram propostas simulações utilizando o Dronekit e em seguida utilizando o OASIs, um simulador próprio desenvolvido utilizando a *engine* de jogos Unity em conjunto com o Dronekit (descrito na [Seção 4.9](#)). Em sequência, foram realizados experimentos reais utilizando um VANT modelo 3DR Solo transmitindo as informações para serem processadas em uma estação em terra.

5.1.1 Simulação com Dronekit

Para realizar a simulação utilizando o Dronekit foi proposto o cenário apresentado na [Figura 18](#). Neste cenário, o VANT foi programado para se mover sobre o eixo x em uma distância equivalente a 6 metros apontando em um ângulo yaw de 30° . Como o VANT utilizado na simulação é do tipo quadrotor, esta configuração foi escolhida de maneira a exigir que o VANT gerasse uma variação nos ângulos de yaw , $pitch$ e $roll$ durante o deslocamento.

Uma simulação do voo com deslocamento descrito no cenário foi executado no Dronekit por 40 vezes e os dados de posicionamento e atitude do VANT foram armazenados. Posteriormente, foi utilizada a fórmula de projeção para calcular a posição de um obstáculo na imagem, utilizando dados de calibração de uma câmera *GOPRO Hero 4* anexada ao VANT 3DR Solo transmitindo vídeo em resolução 720p.

O obstáculo simulado foi colocado em diversas posições a fim de verificar o erro em relação à distância do obstáculo. Com isto, o obstáculo foi posicionado a 5m, 10m, 15m, 20m, 25m, 30m, 35m e 40m do ponto de origem do VANT, sempre a um ângulo yaw de 30° . Com isto,

Figura 18 – Movimentação do VANT para a simulação com Dronekit para validar os métodos de estimativa de posição do obstáculo.

Fonte: Elaborada pelo autor.

foi possível criar um conjunto de dados contendo a posição e atitude do VANT; e a posição do obstáculo na imagem para cada caso de teste.

Posteriormente, os algoritmos descritos na [Seção 4.5](#) foram aplicados para analisar os dados de posição e atitude do VANT e os respectivos pontos gerados nas imagens e obter a posição original do obstáculo. Na primeira situação os dados foram considerados exatamente como foram gerados, o que simula uma situação com sensores perfeitos. Os resultados são sumarizados na [Figura 19](#).

Observando a [Figura 19](#) pode-se perceber que a inclusão dos ângulos de *pitch* e *roll* seguindo a mesma abordagem apresentada por [Saha, Natraj e Waharte \(2014\)](#) gera uma redução na mediana do erro, ao mesmo tempo que a sua variância aumenta. Isto ocorre pois acrescentar os ângulos de *pitch* e *roll* no cálculo faz com que o erro tenha uma diminuição, mas estes métodos não trabalham com todas as equações possíveis, o que faz com que, mesmo considerando todos os ângulos, o erro não seja zerado e a variância aumente. Ao considerar os três ângulos em conjunto com o método dos mínimos quadrados para n frames (MMQ_N) com $n = 2$ (MMQ_2), considerando sensores perfeitos, o erro é sempre zero (0), uma vez que com o MMQ é possível considerar todas as equações de projeção disponíveis.

Outra observação possível a partir da [Figura 19](#) é que o erro dos demais algoritmos aumenta em relação à distância do obstáculo, sendo que os menores erros são observados com o obstáculo a 5 metros de distância ([Figura 19a](#)) e aumenta gradualmente até o último teste realizado com o obstáculo a 40 metros de distância ([Figura 19h](#)). Isto torna evidente a importância de considerar todas as equações para obter uma boa precisão no cálculo da posição do obstáculo.

Figura 19 – Erro dos algoritmos considerando sensores perfeitos

Fonte: Elaborada pelo autor.

Os mesmos dados gerados pela simulação do Dronekit foram aplicados novamente à fórmula de projeção, mas desta vez, a posição do obstáculo na imagem gerou um ponto com o valor truncado para inteiro. Esta situação foi gerada para verificar o erro observado quando a posição do obstáculo na imagem não é medida com perfeição.

Os dados também foram aplicados aos mesmos algoritmos e os resultados são sumaria-

Figura 20 – Erro dos algoritmos considerando valores inteiros para os pontos na imagem

Fonte: Elaborada pelo autor.

dos na Figura 20. Pode-se observar que o comportamento dos algoritmos foi muito similar ao ilustrado na Figura 19. No entanto, para esta simulação os resultados do algoritmo MMQ_n não foram sempre zero (0). Para a distância de 5 metros (Figura 20a) a mediana do erro observado foi 0,048 metros e para a distância de 40 metros (Figura 20h) a mediana do erro observado foi de 4,196 metros.

Com isto, pode-se observar que o aumento da distância torna evidente a influência do erro dos sensores no cálculo da posição do obstáculo. Ainda assim, o método apresentado nesta tese, ao considerar n frames, funciona como um filtro para os possíveis erros de sensor gerando um resultado mais confiável. Uma possibilidade de melhoria (trabalhos futuros) é utilizar outras fontes de informação para posição e atitude (assim como descrito no módulo de *Pose Estimation*).

5.2 Cálculo de Distância pela Intensidade de Sinal WiFi

Para verificar a possibilidade da utilização da intensidade do sinal sem fio recebido de nós de infraestrutura ou outras aeronaves como uma das formas de estimar a distância para obstáculos foi realizado um experimento inicial onde foi possível definir o coeficiente de perda de sinal a ser utilizado na equação de estimativa de distância. Com isto a distância pode ser estimada por meio da [Equação 5.1](#).

$$d = 10 \left[\frac{(RSS + 10)}{-30} \right] \quad (5.1)$$

Para o experimento inicial de definição do PLE (definido no [Capítulo 2](#)), bem como os experimentos de verificação apresentados na sequência, foi utilizado um roteador com o sistema operacional Linux (OpenWRT) utilizando o *chipset* sem fio *Ralink* modelo RT-3050F configurado em modo *adhoc* com intervalo entre *beacons* de 10 milissegundos, potência de transmissão de 20 dBm e ganho da antena de 5 dBi. O roteador foi fixado no final do caminho de teste.

Foi também empregada uma placa de cubieboard2 com processador ARM rodando sistema operacional Linux (Cubian), com *chipset* wireless *Ralink* modelo RT-5370 configurado em modo *monitor* com potência de transmissão de 20 dBm e ganho de antena de 2 dBi. O conjunto foi instalado em um automóvel.

Foi realizado um experimento para verificar a efetividade do cálculo da distância baseado na intensidade de sinal sem fio em cenários onde o receptor está se movimentando em altas velocidades.

O experimento foi realizado em uma estrada a cerca de 5 km de distância do centro da cidade, a fim de reduzir interferências de outras redes sem fio. Foi selecionado um caminho de 500 metros em uma área rural de Vera Cruz, SP (450 km distantes da capital, São Paulo), cujo terreno é composto por uma estrada cercada por plantações de café e algumas fazendas, com poucas árvores de grande porte que poderiam refletir ou interferir no sinal. A configuração de teste é ilustrada na [Figura 21](#).

Em cada teste o veículo entrou no percurso com uma velocidade constante. A fim de avaliar o comportamento da intensidade do sinal em diferentes velocidades, o automóvel percorreu o caminho 10 vezes a 40 e 80km/h, totalizando 20 testes.

Figura 21 – Ambiente de testes de intensidade de sinal (RSS)

Fonte: [Chiaromonte e Branco \(2014\)](#).

Para avaliar se o comportamento do sinal se altera de acordo com a velocidade e distância, o caminho foi dividido em 10 partes de 50 metros cada, a média e o desvio padrão da intensidade do sinal em cada uma destas partes é descrita na [Tabela 3](#). Considerando os testes realizados a 40 km/h a parte do percurso que apresentou o maior desvio padrão foi a localizada entre 0 e 50 metros, com 3,5 de desvio padrão. Nos testes realizados a 80 km/h o desvio padrão mais elevado foi observado entre 350 e 400 metros e foi de 1,6 (valores em destaque na tabela).

Tabela 3 – Média e desvio padrão da intensidade do sinal nas duas velocidades testadas

Classes de Distância (m)	Média - RSS (dBm)		Desvio Padrão - RSS (dBm)	
	40 km/h	80 km/h	40 km/h	80 km/h
450-500	-93,5	-92,9	0,4	0,5
400-450	-92,2	-90,3	0,6	1,1
350-400	-91,1	-89,4	0,5	1,6
300-350	-88,9	-86,9	0,5	1,3
250-300	-87,9	-86,2	0,6	0,7
200-250	-85,5	-86,2	0,9	1,3
150-200	-82,0	-81,2	1,5	0,9
100-150	-77,2	-75,3	1,2	1,5
50-100	-68,7	-67,2	1,8	0,9
0-50	-54,5	-53,8	3,5	1,0

Fonte: Elaborada pelo autor.

O experimento forneceu entre 3000 a 4000 *beacons* cada vez que o automóvel atravessou o percurso de 500 metros com a velocidade constante de 40 km/h, resultando em uma média de 6,5 *beacons*/metro. A intensidade do sinal correspondeu à distância entre o automóvel e o roteador e é apresentada na Figura 22. O desvio padrão (linhas verticais) é menor em distâncias maiores. Em distâncias menores do que 50 metros, o sinal é maior do que -70 dBm (CHIARAMONTE; BRANCO, 2014).

Figura 22 – Sinal x Distância a 40 Km/h

Fonte: Elaborada pelo autor.

No teste a 80km/h aproximadamente 1800-2000 *beacons* foram recebidos cada vez que o automóvel atravessou o caminho gerando uma média de 3,8 *beacons*/metro. O comportamento da intensidade do sinal, a uma velocidade de 80km/h é mostrado na Figura 23. Nesta velocidade o comportamento da intensidade média de sinal foi semelhante à observada a 40 km/h. No entanto o desvio padrão (linhas verticais) apresentado nos testes a 80km/h foi maior.

Após a coleta dos dados de intensidade de sinal foi possível aplicar a Equação 5.1 definida na Seção 4.3 a fim de se obter uma distância estimada a partir de uma dada indicação de intensidade do sinal recebido. A distância real apresentou alta correlação positiva com a distância estimada, o coeficiente de correlação de *Pearson* observado foi de $r = 0,93$, indicando 93% de similaridade entre as distâncias reais e estimadas. O coeficiente de correlação é maior em pequenas distâncias com $r = 0,96$ em distâncias inferiores a 200 metros. O gráfico de correlação é apresentado na Figura 24.

Uma vez que a distância estimada apresentou elevada correlação com a distância real, foi preparado um cenário em que um alerta é disparado no primeiro sinal (RSS) que a distância seja de 50 metros. Os resultados são apresentados na Figura 25 onde estão destacadas as distâncias em que ocorreram os alertas. O alerta mais distante do roteador ocorreu a 66,4 metros; e o mais próximo ocorreu a 50,4 metros. É importante destacar que em nenhum dos testes o alerta ocorreu

Figura 23 – Sinal x Distância a 80 Km/h

Fonte: Elaborada pelo autor.

Figura 24 – Correlação entre a distância real e a estimada

Fonte: Elaborada pelo autor.

dentro do raio de segurança de 50 metros estipulado.

5.3 Detecção Utilizando CMO e Filtragem Temporal

Para verificar a detecção de possíveis obstáculos distantes, utilizando métodos baseados em CMO e filtragem temporal, foi realizado um experimento utilizando dois VANTs modelo 3DR Solo equipados com câmeras *GOPRO Hero 4* capturando vídeo em *Full HD* (1920x1080) a 60fps e campo de visão *Wide* (aproximadamente 133,6° em diagonal).

Figura 25 – Ilustração do alerta de colisão em cenário 2D

Fonte: Elaborada pelo autor.

Os VANTs se afastaram a uma distância de 300 metros, um deles foi mantido estável a 50 metros de altura e o outro a 27 metros de altura realizou manobra de aproximação. Desta forma, foi possível aplicar os algoritmos de detecção e verificar a distância onde o VANT passou a ser detectado na imagem.

Figura 26 – Resultado da detecção utilizando CMO e Programação dinâmica

Fonte: Elaborada pelo autor.

No método proposto por [Carnie, Walker e Corke \(2006\)](#) são definidos os parâmetros α que ajusta o efeito memória, ou seja, a proporção entre a contribuição do *frame* atual e dos *frames* anteriores na detecção; e τ que ajusta o *threshold* da detecção. O ajuste destes parâmetros pode diminuir a quantidade de falsos positivos gerados por ruídos ou artefatos com o efeito colateral de atrasar a detecção de um possível obstáculo.

A [Figura 26](#) apresenta o momento da detecção utilizando o método CMO aliado à filtragem temporal baseado em programação dinâmica ([CARNIE; WALKER; CORKE, 2006](#)). Utilizando como parâmetros $\alpha = 0,5$ e $\tau = 15,0$ a primeira detecção ocorreu quando os VANTs estavam a uma distância de 192 metros se tornando consistente à 157 metros. No momento da

detecção o VANT estava com um tamanho de apenas 2x3 *pixels* na imagem, como apresentado em [Figura 27](#).

Figura 27 – VANT detectado com apenas 2x3 *pixels* (destaque da [Figura 26](#))

Fonte: Elaborada pelo autor.

Apesar da detecção utilizando o método apresentado por [Carnie, Walker e Corke \(2006\)](#) ser eficiente a longas distâncias, o tempo de execução do método é muito alto, tornando inviável para a execução em tempo real. Com isto, uma variação do método (descrita no [Capítulo 4](#)) foi elaborada e implementada com a finalidade de reduzir o tempo de execução e permitir a execução em tempo real.

Na variação implementada, os resultados referentes à detecção se mostraram idênticos aos resultados do método apresentado por [Carnie, Walker e Corke \(2006\)](#), com detecção inicial à 192 metros e detecção consistente à 157 metros. Entretanto, a grande vantagem da variação proposta é em relação ao tempo de execução.

Os dois métodos foram utilizados para processar 1000 *frames* e o tempo de processamento de cada *frame* foi medido. A média de tempo de processamento de um *frame* com o método de [Carnie, Walker e Corke \(2006\)](#) foi de 37,012 milissegundos com desvio padrão de 1,244; enquanto que o tempo para cada *frame* na variação proposta foi de 1,411 milissegundos com um desvio padrão de 0,341. Os tempos são melhores ilustrados na Tabela 4 onde os tempos são todos expressos em milissegundos.

É importante ressaltar que, para que o sistema funcione em tempo real em uma taxa de 30 *frames* por segundo, o tempo de execução de todos os processos (incluindo a detecção) sejam menores que 33,000 milissegundos. Neste sentido, a variação desenvolvida atende a este requisito podendo ser executada em tempo real.

Tabela 4 – Tempo de execução dos algoritmos baseados em CMO por *frame*

Algoritmo	Tempo mínimo	Tempo máximo	Média	Desvio padrão
CMO e Programação Dinâmica	35,281 ms	49,565 ms	37,012 ms	1,244 ms
Variação proposta	1,085 ms	5,218 ms	1,411 ms	0,341 ms

Fonte: Elaborada pelo autor.

5.3.1 Simulação com OASIs

Para obter uma simulação mais próxima ao real, o mesmo cenário apresentado na [Subseção 5.1.1](#) também foi aplicado em simulação utilizando o OASIs. Para esta simulação foi utilizado o valor $n = 60$ para o algoritmo MMQ_N. Isto significa que 60 *frames* foram considerados para o cálculo da posição do obstáculo de forma a reduzir o impacto gerado pelos eventuais erros de sensores e sincronia de tempo entre o *link* de vídeo e os dados do VANT.

Os resultados da simulação estão sumarizados na [Figura 28](#) onde pode-se perceber mais uma vez a influência da distância na precisão do cálculo da posição do obstáculo. No entanto, neste ambiente a diferença da mediana do erro entre os algoritmos que consideram as equações de maneira parcial não foram significativas, destacando ainda mais a vantagem de utilizar o método dos mínimos quadrados considerando n *frames*.

5.3.2 Experimentos Reais

Para verificar a validade do método de cálculo da posição do obstáculo em ambiente real foram realizados experimentos utilizando um VANT modelo 3DR Solo com uma câmera *GOPRO Hero 4* configurada para capturar vídeo em uma resolução de 1080p. Os dados dos sensores e o vídeo foram armazenados para processamento posterior.

Inicialmente o percurso do experimento realizado consistiu em aproximar e afastar o VANT do obstáculo mantendo uma distância mínima de 4 metros e máxima de 40 metros. Este percurso foi escolhido para favorecer a alteração do ângulo de *pitch* e foram executadas 5 repetições. Mais uma vez, para o algoritmo MMQ_N o valor de n escolhido foi 60.

A distribuição do erro em relação à distância pode ser observada na [Figura 29](#). É possível perceber em [Figura 29a](#) que no início e no final do trajeto o erro é menor que na parte intermediária. Isto ocorre pois o algoritmo avaliado não considera o ângulo de *pitch* que é mais acentuado durante o movimento constante para a frente; o que não ocorre nos extremos do percurso devido a mudança de aceleração.

Ao utilizar também o ângulo de *pitch* no cálculo ([Figura 29b](#)) o erro na parte intermediária do trajeto diminui. Especificamente neste cenário a adição do ângulo de *roll* ([Figura 29c](#)) não modificou a distribuição do erro em relação à adição do ângulo de *pitch*, o que já era esperado, uma vez que para realizar o percurso do cenário o VANT não emprega alterações em relação ao

Figura 28 – Erro dos algoritmos em simulação utilizando o OASIs

Fonte: Elaborada pelo autor.

ângulo de *roll*.

Finalmente, em [Figura 29d](#) pode ser observada uma distribuição do erro mais homogênea demonstrando a resistência do método aos eventuais erros existentes nos sensores.

Em todos os algoritmos ainda é observado que a distância entre o VANT e o obstáculo

Figura 29 – Erros obtidos com os métodos a) *Yaw*, b) *Yaw e Pitch*, c) *Yaw, Pitch e Roll* e d) *MMQ_N* para o experimento realizado com a variação do ângulo de *pitch* durante a trajetória de voo

Fonte: Elaborada pelo autor.

influencia no erro. Isto ocorre pois o impacto da precisão dos sensores fica mais evidente conforme a distância aumenta.

É importante destacar que métodos similares encontrados na literatura realizam a validação em distâncias curtas (no máximo 5 metros) e que o método desenvolvido nesta tese apresentou uma média de erro menor mesmo a distâncias de 20 metros.

Para efeitos de comparação, todos os algoritmos foram compilados na [Figura 30](#) para evidenciar as diferenças nos resultados entre eles.

O segundo experimento realizado consistiu em variar a posição do VANT ao longo do

Figura 30 – Erros obtidos para o experimento realizado com a variação do ângulo de *pitch* durante a trajetória de voo

Fonte: Elaborada pelo autor.

eixo *y*, gerando variações do ângulo de *roll*. Assim foi possível verificar o comportamento dos algoritmos que incorporam o ângulo de *roll* no cálculo e realizar uma comparação com os demais algoritmos. Neste cenário, o VANT foi posicionado inicialmente à 10 metros de distância do obstáculo e se moveu lateralmente até o obstáculo sair do campo de visão, realizando o mesmo movimento no sentido contrário e, com isto, a distância do VANT em relação ao obstáculo variou até 14 metros. Foram realizadas 5 repetições deste movimento.

O erro no cálculo de posição do obstáculo para este experimento está sumarizado na Figura 31. Uma vez, que o ângulo de *pitch* permaneceu relativamente constante durante o teste, os resultados do algoritmo que considera apenas o ângulo de *yaw* (Figura 31a) e o algoritmo que considera também o *pitch* (Figura 31b) são similares e a mediana do erro é aproximadamente 10 metros. Para o algoritmo que considera também o ângulo de *roll* (Figura 31c), mas sem utilizar o MMQ, a mediana do erro foi de 8 metros e o erro mínimo foi de 26 centímetros.

O método MMQ_N com $n = 60$ (Figura 31d) apresentou erro quase constante e muito próximo de zero (88 centímetros de mediana e erro mínimo de apenas 3 centímetros).

Para efeitos de comparação, os resultados de todos os algoritmos foram compilados na Figura 32 para evidenciar as diferenças nos resultados entre eles.

As quatro principais variações consideradas para efeito de comparação foram:

- Apenas *Yaw* (SAHA; NATRAJ; WAHARTE, 2014);
- *Yaw* e *Pitch* (CHIARAMONTE; Castelo Branco, 2017);
- *Yaw*, *Pitch* e *Roll*

Figura 31 – Erros obtidos para o experimento realizado com a variação do ângulo de *roll* durante a trajetória de voo

Fonte: Elaborada pelo autor.

- *Yaw, Pitch e Roll* utilizando método dos mínimos quadrados para n frames (*MMQ_n*).

Os resultados permitem evidenciar que a evolução da proposta de detecção de obstáculo desenvolvida o escopo desta tese de doutorado apresenta melhores resultados avançando assim no estado da arte.

Figura 32 – Experimento 2 - Variação de ângulo de *roll* (combinado)

Fonte: Elaborada pelo autor.

5.4 Desvio de Obstáculos com ORCA

Com a posição do obstáculo calculada a partir do módulo de **Estimativa de Posição do Obstáculo** é possível utilizar o algoritmo ORCA (BERG *et al.*, 2011) para realizar o desvio.

Para uma validação inicial, foi criado um cenário no simulador OASIs onde o VANT deve chegar à uma posição objetivo e, um obstáculo foi posicionado um pouco antes desta posição. Caso o VANT siga em linha reta para o objetivo ele estará em rota de colisão com o obstáculo.

A Figura 33 apresenta o percurso realizado pelo VANT (em metros) utilizando o MO-SAIC em uma das simulações realizadas. É possível visualizar que assim que o VANT chega próximo ao obstáculo o ORCA começa a fornecer vetores de velocidade que não permitem o conflito com o obstáculo, evitando assim a colisão.

O vídeo de uma das simulações mostrando o funcionamento do MOSAIC pode ser visualizado em: https://youtu.be/lBG_ckKQWJg.

5.4.1 Desvio de Obstáculo Virtual

Para verificar a integração do ORCA com o módulo de **Missão e Navegação do MO-SAIC** foi realizado um experimento real em que o VANT teve que cumprir uma missão simples partindo de uma posição original predefinida até chegar a um *waypoint*.

Um obstáculo virtual foi definido entre a origem e o *waypoint* e registrado no ORCA. Com isto, o ORCA passou a gerar vetores de velocidade com as alterações necessárias para não entrar em conflito com o obstáculo virtual.

Figura 33 – Simulação do MOSAIC utilizando o OASIs

Fonte: Elaborada pelo autor.

Durante todo o trajeto o módulo de **Missão e Navegação do MOSAIC** repassou ao VANT os vetores de velocidade indicados pelo algoritmo ORCA. O trajeto realizado pelo VANT para desviar da posição onde o obstáculo virtual foi inserido é apresentado na [Figura 34](#).

5.4.2 Considerações finais

Este capítulo apresentou experimentos e os respectivos resultados de forma a validar cada módulo do MOSAIC individualmente, validando o sistema como um todo.

A validação inicial do módulo de **Estimativa de Posição do Obstáculo** foi realizada por meio de uma simulação. Foi utilizado apenas o DroneKit para obter dados relativos ao posicionamento e atitude do VANT. Estes dados foram aplicados à formula de projeção considerando o obstáculo em diferentes distâncias e foram geradas as posições na imagem equivalentes às posições e atitudes do VANT.

Figura 34 – Controle do VANT utilizando o ORCA (obstáculo simulado)

Fonte: Elaborada pelo autor.

Posteriormente, os algoritmos de estimativa de posição de obstáculos foram aplicados considerando uma situação com sensores perfeitos. Ficou evidente neste experimento que os algoritmos que não consideravam todos os ângulos de atitude do VANT não possuem um bom rendimento enquanto que o método desenvolvido nesta tese de doutorado obteve erro zero.

Uma outra avaliação foi realizada considerando a situação onde os pontos na imagem não foram representados com perfeição. Neste caso, como era esperado, todos os algoritmos apresentaram erro, mas o método aqui desenvolvido apresentou os melhores resultados, corroborando com a hipótese desta tese e constituindo assim um avanço ao estado da arte.

Uma simulação utilizando o OASIS foi realizada de forma a validar os módulos de **Deteccção, Posição do Obstáculo, Estimativa de Posição do Obstáculo e Processamento de Dados do VANT**. Nesta simulação, o *link* de vídeo proveniente do OASIS é enviado ao módulo de **Deteccção** enquanto que os dados provenientes do MAVLINK são enviados ao módulo de **Processamento de Dados do VANT**. Ambos os sinais são recebidos e alinhados no módulo de **Posição do Obstáculo** que por sua vez, repassa as informações ao módulo de **Posição do Obstáculo**. Os resultados da simulação com o OASIS são consistentes com os resultados obtidos pela simulação realizada apenas com o DroneKit, o que demonstra que os módulos funcionaram bem em conjunto e em tempo real.

Dois experimentos reais, com distância até 40 metros do obstáculo, foram realizados para a validação dos mesmos módulos citados anteriormente. No experimento real, os resultados também são consistentes com os resultados obtidos pela simulação, destacando mais uma vez que o método MMQ_N é superior aos demais métodos. Outra observação importante é que o

experimento realizado supera a distância de teste dos demais trabalhos encontrados na literatura aberta e, ainda assim obtém melhores resultados.

Além dos módulos já citados, os módulos de **Missão e Navegação do MOSAIC** e o módulo de **Desvio** foram validados por simulação e experimentos reais onde foi possível observar que o MOSAIC realizou com êxito o desvio do obstáculo fazendo uso de todas as informações provenientes dos demais módulos e do ORCA adaptado.

Finalmente, para validar o módulo de *Pose Estimation* também foram conduzidos experimentos para verificar a possibilidade de utilizar a intensidade do sinal sem fio como base para o cálculo de distância de um possível obstáculo (considerando que este emite sinais referentes a uma rede sem fio 802.11). Os resultados mostram que a informação de distância obtida tem potencial para ser fundida com outras informações com o objetivo de melhorar a precisão da posição do obstáculo. No entanto, tal fusão não faz parte do escopo desta tese.

No próximo capítulo são apresentadas as principais contribuições desta tese, as conclusões, as limitações do sistema, as dificuldades encontradas e os trabalhos futuros.

CONCLUSÕES E TRABALHOS FUTUROS

Apesar das diversas pesquisas realizadas na área de robótica aérea focando na detecção e desvio de obstáculos e dos avanços já conquistados, essa área continua sendo desafiadora de maneira a permitir a inserção completa dos veículos aéreos não tripulados no espaço aéreo segregado ou não segregado.

A motivação inicial para a realização desta tese foi o desenvolvimento de um sistema de detecção e desvio de obstáculos modular e com arquitetura flexível que permitisse a integração dos SANTs ao espaço aéreo garantindo padrões de aeronavegabilidade.

Assim, esta pesquisa de doutorado investigou o problema de detecção e desvio de obstáculos em VANTs de pequeno porte (MAV) fazendo uso de imagens monoculares e uma abordagem do tipo *reciprocal n-body collision avoidance*, mais especificamente o *Optimal Reciprocal Collision Avoidance* (ORCA), para realizar o desvio.

Para o funcionamento de uma abordagem do tipo *reciprocal n-body collision avoidance* é necessário conhecer com acurácia a posição tridimensional dos obstáculos. Desta forma, foi proposto um novo método de estimativa de posição do obstáculo baseado em visão monocular e os resultados se mostraram superiores aos demais trabalhos encontrados na literatura aberta.

Com isto, foi proposto e implementado o MOSAIC, um sistema de detecção e desvio de obstáculo baseado em visão monocular e foram realizadas avaliações por simulação e experimental por meio de voos reais. Todos os módulos do MOSAIC foram validados e os resultados obtidos mostraram a eficácia do sistema para a detecção e realização do desvio.

Por ser um sistema modular e com arquitetura flexível, é possível integrar novos módulos de detecção e desvio, desta forma, é importante destacar que embora o MOSAIC inicialmente tenha suporte para VANTs de asa rotativa, esta limitação está ligada ao método de desvio empregado que não considera as restrições de aceleração de um veículo de asa fixa. Desta forma, o MOSAIC pode ser embarcado em qualquer tipo de aeronave (mesmo asa fixa) com a utilização

de outro algoritmo de desvio ou com a inclusão de tais restrições no algoritmo ORCA. Além disso, ao abordar essas demandas em soluções integradas, melhores resultados provavelmente serão produzidos.

Deste modo, o MOSAIC contribui para a detecção e desvio de obstáculos tendo sido validado por meio de experimentos simulados e reais, provando que o sistema é aplicável aos campos a que ele se propõe a melhorar. Conclusivamente, as contribuições fornecidas pelo MOSAIC satisfazem as demandas multidisciplinares das aplicações modernas. É um sistema aberto para novos desenvolvimentos por pesquisadores e desenvolvedores potencialmente interessados.

6.1 Contribuições

Nesta tese a contribuição mais importante é a definição do sistema de detecção e desvio de obstáculos, denominado MOSAIC, que permite a detecção e o desvio de obstáculos a uma distância segura o suficiente para que não ocorram colisões.

Entretanto, para cumprir todos esses requisitos, outras contribuições foram derivadas:

- Um módulo de cálculo da posição 3D dos obstáculos baseado em dados de localização, IMU, GPS e visão monocular ([CHIARAMONTE; Castelo Branco, 2017](#));
- Método integrado de controle de navegação para evitar obstáculos fixos e outras aeronaves;
- Modificações realizadas no algoritmo ORCA para considerar a aceleração máxima possível para o VANT ([Sá De Mello et al., 2015](#));
- MOSAIC, um sistema para detecção e desvio de obstáculo que é capaz de detectar obstáculos fixos, calcular a posição 3D destes obstáculos e realizar o desvio dos mesmos;
- OASIs, um simulador para avaliação de algoritmos de detecção e desvio de obstáculos que utilizam visão monocular;
- Método de cálculo de posição do obstáculo baseado na intensidade de sinal sem fio ([CHIARAMONTE; BRANCO, 2014](#)).

6.2 Limitações do MOSAIC

A pesquisa aqui realizada, assim como toda pesquisa, também apresenta limitações. Uma vez que o trabalho realizado possui um prazo para ser finalizado não é possível que todas as possibilidades sejam abordadas e tratadas, fazendo-se assim necessário delimitar o escopo do trabalho.

- Algoritmo de Desvio (ORCA) funciona com segurança apenas considerando algoritmos de asa rotativa;
- Desvio de obstáculos fixos com boa acurácia enquanto que obstáculos dinâmicos podem não ser corretamente localizados (o algoritmo de cálculo da posição do obstáculo não gera vetores de velocidade, considerando todos os obstáculos com velocidade zero);
- O alinhamento temporal considera apenas atraso fixo na transmissão do vídeo.

6.3 Dificuldades Encontradas

Essa pesquisa envolveu conhecimentos multidisciplinares, o que permitiu gerar resultados inovadores ao custo de lidar com diversas dificuldades, não só tecnológicas como também científicas, aqui elencadas:

- **Ausência de padrões:** Existem diversos tipos de abordagens relacionados à detecção de obstáculos, cada abordagem pode gerar um tipo de artefato diferente. No entanto, mesmo as abordagens similares apresentam uma falta de padronização para desenvolvimento e avaliação dos métodos desenvolvidos;
- **Ausência de regras:** A falta de padronização acarreta também na dificuldade de comparação dos resultados uma vez que mesmo artefatos similares são mensurados de maneiras diferentes entre os trabalhos;
- **Integração das abordagens:** a integração das diferentes áreas, vista por meio do sistema apresentado, aparenta ser trivial. No entanto, a definição da arquitetura do sistema, bem como a ordem das etapas e a real necessidade de cada uma dessas etapas demandou muito tempo, além de todo o processo de implementação e validação, trabalhoso e desafiador, que apesar de ser tecnológico foi necessário para demonstrar e apresentar a prova de conceito da validade do MOSAIC.
- **Integração Hardware e Software:** realizar voos com um VANT não é algo complicado quando se tem uma boa tecnologia em mãos. Entretanto, a regulamentação do uso desses veículos ainda é algo novo. O voo em regiões povoadas não é autorizado. A realização do voo demandou diversos esforços para que o mesmo acontecesse. Vale a pena ressaltar que diferentes algoritmos foram incluídos para obter informação e prover informações para o VANT de modo que ele realizasse as tarefas solicitadas. Essas inserções e modificações não são triviais. Diferenças quando da implementação física (real) dos algoritmos e abordagens causam modificações e acertos necessários, que só são possíveis de serem realizados quando da execução real dos mesmos no VANT. Na prática teorias foram adaptadas e adequadas para que o funcionamento fosse apropriado.

6.4 Trabalhos Futuros

O desafio no desenvolvimento de sistemas de detecção e desvio de obstáculos é algo bastante amplo e contínuo, especialmente com o aumento no número de veículos aéreos não tripulados que estão sendo inseridos diariamente no espaço aéreo, seja ele segregado ou não segregado. Na versão em que se encontra o MOSAIC permite que tanto a detecção quanto o desvio sejam efetuados de um objeto fixo com uma acurácia na detecção melhor do que o que há na literatura aberta atual. Inúmeras são as possibilidades extensão do MOSAIC, sendo assim, sugerem-se os seguintes trabalhos a serem realizados que permitirão dar continuidade do MOSAIC:

- Fusão de SLAM e GPS para aumentar a acurácia da localização e consequentemente a acurácia do cálculo do obstáculo;
- Integração de um método de *Visual Pose Estimation* e fusão com dados inerciais provenientes de IMU para prover melhor acurácia no cálculo da posição do obstáculo;
- Método de alinhamento temporal baseado na fusão dos itens mencionados anteriormente;
- Método de calibração e alinhamento da câmera em relação ao *body frame* do VANT;
- Método de integração do OASIS como o algoritmo de detecção para automatizar o cálculo do erro uma vez que os dois módulos ainda não tem integração após feito o cálculo da posição do obstáculo;
- Adicionar restrições de aceleração diferentes para cada direção de manobra do VANT no algoritmo ORCA para a extensão do MOSAIC em VANTs de asa fixa;
- Modificar o sistema de cálculo de posição do obstáculo para diminuir o erro na presença de obstáculos dinâmicos.

6.5 Declaração de autoria

Eu confirmo que esta tese de doutorado não foi apresentada como parte dos requisitos para obtenção de um outro grau nesta ou em qualquer outra instituição de ensino e pesquisa. Afirmando ainda que ela é o resultado do meu próprio trabalho e todo o material de outras fontes foi plenamente e adequadamente reconhecido/referenciado. Pesquisas realizadas em colaboração estão claramente indicadas. Trechos desta tese, quer tenham sido publicados ou submetidos à apreciação dos conselhos editoriais de revistas, conferências e *workshops*, são apresentados a seguir de acordo com a lista de publicações. Minhas contribuições a cada publicação também são listadas.

6.6 Lista de publicações

6.6.1 Artigos publicados

- **CHIARAMONTE, R. B.;** BRANCO, K. R. L. J. C. Collision detection using received signal strength in FANETs.

Evento: 2014 International Conference on Unmanned Aircraft Systems (ICUAS)

Nível de contribuição: Alto - o aluno de doutorado é o principal pesquisador e conduziu os trabalhos em conjunto com os demais colaboradores.

- **CHIARAMONTE, R. B.;** Castelo Branco, K. R. L. J. Estimativa de Posição 3D de Obstáculos para VANTs.

Evento: V Workshop de Comunicação em Sistemas Embarcados Críticos (WoCCES 2017). Belém - Pará.

Nível de contribuição: Alto - o aluno de doutorado é o principal pesquisador e conduziu os trabalhos em conjunto com os demais colaboradores.

- Sá De Mello, A.; **CHIARAMONTE, R.;** FILHO, L.; INOUE, R.; Castelo Branco, K. Simulation results of cooperative formation flying control of UAVs with collision avoidance based on ORCA.

Evento: APISAT 2015 - 7th Asia-Pacific International Symposium on Aerospace Technology.

Nível de contribuição: Alto - o pesquisador é um dos principais investigadores .

6.6.2 Artigos submetidos para publicação

- **CHIARAMONTE, R. B.;** INOUE, R.; BRANCO, K. R. L. J. C. MMQ - A new approach to estimate obstacle three-dimensional position based on monocular vision.

Journal: Journal of Control, Automation and Electrical Systems.

Nível de contribuição: Alto - o aluno de doutorado é o principal pesquisador e conduziu os trabalhos em conjunto com os demais colaboradores.

REFERÊNCIAS

3DR. *3DR Solo*. 2017. Disponível em: <<https://3dr.com/solo-drone/>>. Citado nas páginas 95 e 96.

ADOUANE, L. Orbital Obstacle Avoidance Algorithm for Reliable and On-Line Mobile. p. 79–85, 2009. Citado nas páginas 33, 34, 52 e 53.

AGUILAR, W.; CASALIGLLA, V.; PÓLIT, J. Obstacle avoidance based-visual navigation for micro aerial vehicles. **Electronics**, MDPI AG, v. 6, n. 4, p. 10, jan 2017. Citado nas páginas 63, 68 e 74.

AGUILAR, W. G.; CASALIGLLA, V. P.; POLIT, J. L. Obstacle avoidance for low-cost UAVs. In: **2017 IEEE 11th International Conference on Semantic Computing (ICSC)**. [S.l.]: IEEE, 2017. Citado nas páginas 63 e 73.

AGUILAR, W. G.; CASALIGLLA, V. P.; PÓLIT, J. L.; ABAD, V.; RUIZ, H. Obstacle avoidance for flight safety on unmanned aerial vehicles. In: **Advances in Computational Intelligence**. [S.l.]: Springer International Publishing, 2017. p. 575–584. Citado na página 63.

AL-KAFF, A.; GARCÍA, F.; MARTÍN, D.; ESCALERA, A. D. L.; ARMINGOL, J. Obstacle detection and avoidance system based on monocular camera and size expansion algorithm for UAVs. **Sensors**, MDPI AG, v. 17, n. 12, p. 1061, may 2017. Citado nas páginas 31, 59, 63, 68 e 73.

AL-KAFF, A.; MARTÍN, D.; GARCÍA, F.; ESCALERA, A. de la; ARMINGOL, J. M. Survey of computer vision algorithms and applications for unmanned aerial vehicles. **Expert Systems with Applications**, Elsevier BV, v. 92, p. 447–463, feb 2018. Citado na página 74.

AL-KAFF, A.; MENG, Q.; MARTIN, D.; ESCALERA, A. de la; ARMINGOL, J. M. Monocular vision-based obstacle detection/avoidance for unmanned aerial vehicles. In: **2016 IEEE Intelligent Vehicles Symposium (IV)**. [S.l.]: IEEE, 2016. Citado na página 63.

ALEJO, D.; COBANO, J.; HEREDIA, G.; OLLERO, A. Optimal reciprocal collision avoidance with mobile and static obstacles for multi-uav systems. In: IEEE. **Unmanned Aircraft Systems (ICUAS), 2014 International Conference on**. [S.l.], 2014. p. 1259–1266. Citado na página 54.

ANAC. **Manual de procedimentos - Certificação de projeto de tipo brasileiro - MPR-200**. [S.l.], 2010. Revisão 02. Citado na página 32.

_____. **Requisitos de Aeronavegabilidade: Aeronaves de Asas Rotativas, Categoria Normal - RBAC 27**. [S.l.], 2013. Citado na página 32.

_____. **Requisitos de Aeronavegabilidade: Aviones Categoria Normal, Utilidade, Acrobática e Transporte - RBAC 23**. [S.l.], 2013. Citado na página 32.

_____. **Proposição de um Regulamento Especial para Sistema de Aeronaves Remotamente Pilotadas – RPAS e Emenda ao RBAC 67**. [S.l.], 2015. Citado na página 32.

_____. **Requisitos Gerais para Veículos Aéreos não Tripulados e Aeromodelos**. [S.l.], 2015. Citado na página 39.

ARMOUSH, A.; BECKSCHULZE, E.; KOWALEWSKI, S. Safety assessment of design patterns for safety-critical embedded systems. In: SEAA '09: PROCEEDINGS OF THE 2009 35TH EUROMICRO CONFERENCE ON SOFTWARE ENGINEERING AND ADVANCED APPLICATIONS. Washington,DC: IEEE Computer Society, 2009. p. 523–527. Citado na página 29.

ARMY, U. S. **U.S. Army Roadmap for UAS 2010-2035**. Alabama, 2010. Citado nas páginas 33 e 59.

AROKIASAMI, W. A.; VADAKKEPAT, P.; TAN, K. C.; SRINIVASAN, D. Interoperable multi-agent framework for unmanned aerial/ground vehicles: towards robot autonomy. **Complex & Intelligent Systems**, Springer Nature, v. 2, n. 1, p. 45–59, mar 2016. Citado nas páginas 63, 68 e 73.

AUSTIN, R. **Unmanned Aircraft Systems: UAVS Design, Development and Deployment**. USA: Wiley, 2010. (Aerospace Series). ISBN 9780470664803. Citado nas páginas 29, 39 e 41.

_____. **Unmanned Aircraft Systems: UAVS Design, Development and Deployment**. [S.l.]: Wiley, 2011. ISBN 978-0-470-05819-0. Citado nas páginas 38 e 40.

AUTHORITY, U. C. A. **Unmanned Aircraft System Operations in UK Airspace–Guidance**. [S.l.]: CAP722, 2012. Citado nas páginas 33 e 59.

BAYHA, A.; GRÜNEIS, F.; SCHÄTZ, B. Model-based software in-the-loop-test of autonomous systems. Society for Computer Simulation International, p. 30, mar 2012. Citado na página 52.

BEI, L.; HAIZHEN, Z. An algorithm of fabric image mosaic based on sift feature matching. In: IEEE. **2009 International Conference on Artificial Intelligence and Computational Intelligence**. [S.l.], 2009. p. 435–438. Citado na página 47.

BEKMEZCI, İ.; SAHINGOZ, O. K.; TEMEL, Ş. Flying ad-hoc networks (FANETs): A survey. **Ad Hoc Networks**, Elsevier BV, v. 11, n. 3, p. 1254–1270, may 2013. Citado na página 40.

BENZERROUK, A.; ADOUANE, L.; LEQUIEVRE, L.; MARTINET, P. Navigation of multi-robot formation in unstructured environment using dynamical virtual structures. **IEEE/RSJ 2010 International Conference on Intelligent Robots and Systems, IROS 2010 - Conference Proceedings**, p. 5589–5594, 2010. ISSN 2153-0858. Citado nas páginas 34 e 53.

BENZERROUK, A.; ADOUANE, L.; MARTINET, P. Lyapunov global stability for a reactive mobile robot navigation in presence of obstacles. In: **ICRA'10 International Workshop on Robotics and Intelligent Transportation System, RITS10**. [S.l.: s.n.], 2010. Citado na página 53.

_____. Dynamic obstacle avoidance strategies using limit cycle for the navigation of multi-robot system. In: **2012 IEEE/RSJ IROS'12, 4th Workshop on Planning, Perception and Navigation for Intelligent Vehicles**. [S.l.: s.n.], 2012. Citado na página 53.

BERG, J.; GUY, S. J.; LIN, M.; MANOCHA, D. Reciprocal n-Body Collision Avoidance. **Springer Tracts in Advanced Robotics**, v. 70, p. 3–19, 2011. ISSN 1610-742X. Citado nas páginas 34, 54, 55, 56, 57, 58, 66, 91 e 111.

BERG, M. d.; CHEONG, O.; KREVELD, M. v.; OVERMARS, M. **Computational Geometry: Algorithms and Applications**. 3rd ed.. ed. Santa Clara, CA, USA: Springer-Verlag TELOS, 2008. ISBN 3540779736, 9783540779735. Citado na página 57.

BERRY, A. J.; HOWITT, J.; GU, D.-W.; POSTLETHWAITE, I. A continuous local motion planning framework for unmanned vehicles in complex environments. **Journal of Intelligent & Robotic Systems**, Springer Nature, v. 66, n. 4, p. 477–494, dec 2011. Citado na página 91.

BORENSTEIN, J.; KOREN, Y. Real-time obstacle avoidance for fast mobile robots in cluttered environments. In: IEEE. **Robotics and Automation, 1990. Proceedings., 1990 IEEE International Conference on**. [S.l.], 1990. p. 572–577. Citado na página 52.

_____. The vector field histogram-fast obstacle avoidance for mobile robots. **IEEE transactions on robotics and automation**, IEEE, v. 7, n. 3, p. 278–288, 1991. Citado na página 52.

BRANCO, K. R. L. J. C. **Contribuições na área de Sistemas Distribuídos e Redes de Computadores e suas aplicações em Sistemas Embarcados Críticos**. Tese (Livre Docência em Redes de computadores) — Universidade de São Paulo, São Carlos, 2012. Citado na página 29.

BRANCO, K. R. L. J. C.; PELIZZONI, J. M.; NERIS, L. O.; TRINDADE, O.; OSORIO, F. S.; WOLF, D. F. Tiriba - a new approach of UAV based on model driven development and multiprocessors. In: **2011 IEEE International Conference on Robotics and Automation**. [S.l.]: IEEE, 2011. Citado na página 29.

BU, Q.; WAN, F.; XIE, Z.; REN, Q.; ZHANG, J.; LIU, S. General simulation platform for vision based uav testing. In: **2015 IEEE International Conference on Information and Automation**. [S.l.: s.n.], 2015. p. 2512–2516. Citado na página 91.

BYRNE, J.; TAYLOR, C. J. Expansion segmentation for visual collision detection and estimation. In: **2009 IEEE International Conference on Robotics and Automation**. [S.l.]: IEEE, 2009. Citado nas páginas 62, 67 e 73.

CAA. **Unmanned aircraft system operations in UK airspace – Guidance**. [S.l.], 2015. Citado na página 39.

CALONDER, M.; LEPETIT, V.; STRECHA, C.; FUA, P. Brief: Binary robust independent elementary features. **Computer Vision–ECCV 2010**, Springer, p. 778–792, 2010. Citado na página 48.

CAMERON, S. A study of the clash detection problem in robotics. **Proceedings. 1985 IEEE International Conference on Robotics and Automation**, v. 2, 1985. Citado nas páginas 33 e 52.

CARNIE, R.; WALKER, R.; CORKE, P. Image processing algorithms for uav "sense and avoid". In: **Proceedings - IEEE International Conference on Robotics and Automation**. [S.l.: s.n.], 2006. v. 2006, p. 2848–2853. Citado nas páginas 45, 74, 75, 104 e 105.

CHAVEZ, A.; GUSTAFSON, D. Vision-based obstacle avoidance using SIFT features. In: **Advances in Visual Computing**. [S.l.]: Springer Berlin Heidelberg, 2009. p. 550–557. Citado nas páginas 62, 67 e 73.

CHEN, Q.; LIU, H.; YU, M.; GUO, H. RSSI ranging model and 3d indoor positioning with ZigBee network. In: **Proceedings of the 2012 IEEE/ION Position, Location and Navigation Symposium**. [S.l.]: IEEE, 2012. Citado na página 50.

- CHENG, L.; WU, C.-d.; ZHANG, Y.-z.; WANG, Y. An indoor localization strategy for a mini-uav in the presence of obstacles. **International Journal of Advanced Robotic Systems**, SAGE Publications Sage UK: London, England, v. 9, n. 4, p. 153, 2012. Citado na página 49.
- CHIARAMONTE, R. B.; BRANCO, K. R. L. J. C. Collision detection using received signal strength in FANETs. In: **2014 International Conference on Unmanned Aircraft Systems (ICUAS)**. [S.l.]: IEEE, 2014. Citado nas páginas 35, 101, 102 e 116.
- CHIARAMONTE, R. B.; Castelo Branco, K. R. L. J. Estimativa de Posição 3D de Obstáculos para VANTs. In: **V Workshop de Comunicação em Sistemas Embarcados Críticos (WoCCES 2017)**. Belém - Pará: [s.n.], 2017. p. 45–55. Citado nas páginas 35, 95, 109 e 116.
- CHOI, J. S.; SON, B. R.; KANG, H. K.; LEE, D. H. Indoor localization of unmanned aerial vehicle based on passive uhf rfid systems. In: **2012 9th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)**. [S.l.: s.n.], 2012. p. 188–189. Citado na página 49.
- CONROY, P.; BAREISS, D.; BEALL, M. 3-D Reciprocal Collision Avoidance on Physical Quadrotor Helicopters with On-Board Sensing for Relative Positioning. 2014. Citado na página 54.
- CRUZ, G. C. S.; ENCARNACÃO, P. M. M. Obstacle avoidance for unmanned aerial vehicles. **Journal of Intelligent & Robotic Systems**, Springer Nature, v. 65, n. 1-4, p. 203–217, oct 2011. Citado na página 31.
- CYPRIANI, M.; CANALDA, P.; SPIES, F. Performance evaluation of a self calibrated 3d wi-fi fingerprinting positioning system. In: **2012 Fourth International Conference on Communications and Electronics (ICCE)**. [S.l.: s.n.], 2012. p. 274–278. Citado na página 49.
- DAVIES, J. J.; KAMANGAR, F.; ZARUBA, G.; HUBER, M.; ATHITSOS, V. Use of RSSI and time-of-flight wireless signal characteristics for location tracking. In: **Proceedings of the 4th International Conference on Pervasive Technologies Related to Assistive Environments - PETRA 11**. [S.l.]: ACM Press, 2011. Citado na página 49.
- DOD. **Unmanned Systems Roadmap 2007-2032**. Washington,DC, 2007. 187 p. Citado na página 39.
- _____. **Unmanned Systems Integrated Roadmap FY2009-2034**. Washington,DC, 2009. 210 p. Citado na página 39.
- _____. **Unmanned systems integrated roadmap: FY2013-2038**. Washington,DC, 2013. Citado nas páginas 33, 41 e 59.
- DUNN, W. R. Designing safety-critical computer systems. **Computer**, v. 36, n. 11, p. 40–46, 2003. Citado na página 29.
- EASA. **Introduction of a regulatory framework for the operation of unmanned aircraft**. [S.l.], 2015. Citado na página 39.
- ERDMANN, M.; LOZANO-PREZ, T. On multiple moving objects. **Algorithmica**, v. 2, p. 1419–1424, 1987. Citado nas páginas 33 e 52.
- ESCOBAR-ALVAREZ, H. D.; JOHNSON, N.; HEBBLE, T.; KLINGEBIEL, K.; QUINTERO, S. A. P.; REGENSTEIN, J.; BROWNING, N. A. R-ADVANCE: Rapid adaptive prediction for vision-based autonomous navigation, control, and evasion. **Journal of Field Robotics**, Wiley-Blackwell, v. 35, n. 1, p. 91–100, sep 2017. Citado nas páginas 66 e 70.

FAHLSTROM, P.; GLEASON, T. **Introduction to UAV Systems**. 4. ed. [S.l.]: Wiley, 2012. ISBN 978-1-119-97866-4. Citado na página 38.

FILATOV, D. V.; DEVYATKIN, A. V.; POLYAKHOV, N. D. Development of software simulator of uav flight for the studies of trajectory motion systems. In: **2017 IEEE VI Forum Strategic Partnership of Universities and Enterprises of Hi-Tech Branches (Science. Education. Innovations) (SPUE)**. [S.l.: s.n.], 2017. p. 224–226. Citado na página 91.

FIORINI, P.; SHILLER, Z. **Motion Planning in Dynamic Environments Using Velocity Obstacles**. 1998. 760–772 p. Citado nas páginas 34, 52 e 53.

FORSYTH, D. A.; PONCE, J. **Computer Vision, A Modern Approach**. [S.l.]: Prentice Hall, 2003. Citado na página 45.

FRANÇOIS, A. R.; MEDIONI, G. G. Adaptive color background modeling for real-time segmentation of video streams. In: **Proceedings of the International Conference on Imaging Science, Systems, and Technology**. [S.l.: s.n.], 1999. v. 1, n. 121, p. 227–232. Citado na página 43.

FU, C.; OLIVARES-MENDEZ, M. A.; CAMPOY, P.; SUAREZ-FERNANDEZ, R. UAS see-and-avoid strategy using a fuzzy logic controller optimized by cross-entropy in scaling factors and membership functions. In: **2013 International Conference on Unmanned Aircraft Systems (ICUAS)**. [S.l.]: IEEE, 2013. Citado na página 65.

FU, C.; OLIVARES-MENDEZ, M. A.; SUAREZ-FERNANDEZ, R.; CAMPOY, P. Monocular visual-inertial SLAM-based collision avoidance strategy for fail-safe UAV using fuzzy logic controllers. **Journal of Intelligent & Robotic Systems**, Springer Nature, v. 73, n. 1-4, p. 513–533, oct 2013. Citado nas páginas 65 e 69.

GARDEAZABAL, D.; PONOMARYOV, V.; CHAIREZ, I. Fuzzy control for obstacle avoiding in mobile robots using stereo vision algorithms. In: **2011 8th International Conference on Electrical Engineering, Computing Science and Automatic Control**. [S.l.]: IEEE, 2011. Citado na página 31.

GENG, N. *et al.* Algorithm for sequence image automatic mosaic based on sift feature. In: IEEE. **Information Engineering (ICIE), 2010 WASE International Conference on**. [S.l.], 2010. v. 1, p. 203–206. Citado na página 47.

GONCALVES, V. M.; PIMENTA, L. C. A.; MAIA, C. A.; PEREIRA, G. A. S. Coordination of multiple fixed-wing UAVs traversing intersecting periodic paths. In: **2013 IEEE International Conference on Robotics and Automation**. [S.l.]: IEEE, 2013. Citado na página 52.

GONZALES R.; WOODS, R. **Processamento de imagens digitais**. [S.l.]: Prentice Hall, 2002. Citado na página 42.

GONZALEZ, R. C.; WOODS, R. E. **Processamento Digital De Imagens**. 3ª. ed. São Paulo: Addison Wesley, 2010. 624 p. ISBN 8576054019. Citado nas páginas 41, 42 e 44.

GOSIEWSKI, Z.; CIESLUK, J.; AMBROZIAK, L. Vision-based obstacle avoidance for unmanned aerial vehicles. In: **2011 4th International Congress on Image and Signal Processing**. [S.l.]: IEEE, 2011. Citado nas páginas 62, 68 e 73.

HA, J.; SATTIGERI, R. Vision-based obstacle avoidance based on monocular SLAM and image segmentation for UAVs. In: **AIAA Infotech at Aerospace Conference and Exhibit 2012**. [S.l.: s.n.], 2012. Citado nas páginas 61, 67 e 73.

HO, F.; GERALDES, R.; GONCALVES, A.; CAVAZZA, M.; PRENDINGER, H. Simulating shared airspace for service uavs with conflict resolution. In: **INTERNATIONAL FOUNDATION FOR AUTONOMOUS AGENTS AND MULTIAGENT SYSTEMS. Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems**. [S.l.], 2018. p. 2192–2194. Citado na página 54.

HONG, J.; LIN, W.; ZHANG, H.; LI, L. Image mosaic based on surf feature matching. In: **IEEE. Information Science and Engineering (ICISE), 2009 1st International Conference on**. [S.l.], 2009. p. 1287–1290. Citado na página 47.

HRABAR, S.; SUKHATME, G.; CORKE, P.; USHER, K.; ROBERTS, J. Combined optic-flow and stereo-based navigation of urban canyons for a UAV. In: **2005 IEEE/RSJ International Conference on Intelligent Robots and Systems**. [S.l.]: IEEE, 2005. Citado na página 31.

ISHII, K.; SATO, N. GPS-free host approaching in mobile ad-hoc networks. In: **2013 Seventh International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing**. [S.l.]: IEEE, 2013. Citado na página 49.

JESUS, T. A.; PIMENTA, L. C. de A.; TÔRRES, L. A. B.; MENDES, E. M. A. M. On the coordination of constrained fixed-wing unmanned aerial vehicles. **Journal of Control, Automation and Electrical Systems**, Springer Nature, v. 24, n. 5, p. 585–600, jul 2013. Citado na página 52.

JIE, M. S.; BAEK, J. H.; HONG, Y. S.; LEE, K. W. Real time obstacle avoidance for mobile robot using limit-cycle and vector field method. In: **SPRINGER. International Conference on Knowledge-Based and Intelligent Information and Engineering Systems**. [S.l.], 2006. p. 866–873. Citado na página 53.

JUAN, L.; GWUN, O. A comparison of sift, pca-sift and surf. **International Journal of Image Processing (IJIP)**, v. 3, n. 4, p. 143–152, 2009. Citado na página 47.

JULIER, S. J.; UHLMANN, J. K. A New Extension of the Kalman Filter to Nonlinear Systems. In: **Proceedings of AeroSense: The 11th International Symposium on Aerospace/Defence Sensing, Simulation and Controls**. Orlando, FL, United States: [s.n.], 1997. p. 182–193. Citado na página 60.

KALMAN, R. E. A new approach to linear filtering and prediction problems. **Journal of basic Engineering**, American Society of Mechanical Engineers, v. 82, n. 1, p. 35–45, 1960. Citado na página 51.

KHATIB, O. Real-time obstacle avoidance for manipulators and mobile robots. In: **IEEE International Conference on Robotics and Automation Proceedings**. [S.l.: s.n.], 1985. v. 2, p. 500–505. Citado na página 53.

KIM, D. H.; KIM, J. H. A real-time limit-cycle navigation method for fast mobile robots and its application to robot soccer. **Robotics and Autonomous Systems**, v. 42, n. 1, p. 17–30, 2003. ISSN 09218890. Citado na página 53.

KONG, L.-K.; SHENG, J.; TEREDESAL, A. Basic micro-aerial vehicles (MAVs) obstacles avoidance using monocular computer vision. In: **2014 13th International Conference on Control Automation Robotics & Vision (ICARCV)**. [S.l.]: IEEE, 2014. Citado na página 65.

KUMAR, S. P.; RAMAIAH, P. S.; KHANAA, V. Architectural patterns to design software safety based safety-critical systems. In: **ICCCS 11: PROCEEDINGS OF THE 2011 INTERNATIONAL CONFERENCE ON COMMUNICATION, COMPUTING & SECURITY**. New York: ACM, 2011. p. 620–623. Citado na página [29](#).

LAI, J.; MEJIAS, L.; FORD, J. J. Airborne vision-based collision-detection system. **Journal of Field Robotics**, Wiley-Blackwell, v. 28, n. 2, p. 137–157, feb 2011. Citado na página [45](#).

LAN, J.; LI, J.; HU, G.; ZENG, Y. Distance estimation using a panoramic sensor based on a novel spherical optical flow algorithm. **Optics & Laser Technology**, Elsevier BV, v. 45, p. 168–176, feb 2013. Citado na página [31](#).

LANGELAAN, J.; ROCK, S. Navigation of small UAVs operating in forests. In: **Collection of Technical Papers - AIAA Guidance, Navigation, and Control Conference**. [S.l.: s.n.], 2004. v. 3, p. 2031–2041. Citado nas páginas [60](#) e [73](#).

_____. Towards autonomous UAV flight in forests. In: **Collection of Technical Papers - AIAA Guidance, Navigation, and Control Conference**. [S.l.: s.n.], 2005. v. 1, p. 636–648. Citado nas páginas [60](#), [67](#) e [73](#).

LATOMBE, J.-C. **Robot motion planning**. [S.l.: s.n.], 1991. v. 54. 430 p. ISSN 10499660. ISBN 0792391292 9780792391296 079239206X 9780792392064. Citado nas páginas [33](#) e [52](#).

LEE, D.-J.; BEARD, R. W.; MERRELL, P. C.; ZHAN, P. See and avoidance behaviors for autonomous navigation. In: GAGE, D. W. (Ed.). **Mobile Robots XVII**. [S.l.]: SPIE, 2004. Citado na página [64](#).

LEE, D.-J.; MERRELL, P.; WEI, Z.; NELSON, B. E. Two-frame structure from motion using optical flow probability distributions for unmanned air vehicle obstacle avoidance. **Machine Vision and Applications**, Springer-Verlag, v. 21, n. 3, p. 229–240, jun 2010. Citado nas páginas [34](#), [64](#), [69](#) e [74](#).

LEE, J.-O.; LEE, K.-H.; PARK, S.-H.; IM, S.-G.; PARK, J. Obstacle avoidance for small UAVs using monocular vision. **Aircraft Engineering and Aerospace Technology**, Emerald, v. 83, n. 6, p. 397–406, oct 2011. Citado nas páginas [34](#), [64](#), [65](#), [69](#) e [74](#).

LENZ, I.; GEMICI, M.; SAXENA, A. Low-power parallel algorithms for single image based obstacle avoidance in aerial robots. In: **2012 IEEE/RSJ International Conference on Intelligent Robots and Systems**. [S.l.]: IEEE, 2012. Citado nas páginas [63](#), [68](#), [73](#) e [74](#).

LEWIS, F. **Optimal estimation: with an introduction to stochastic control theory**. USA: Wiley, 1986. (A Wiley-interscience publication). ISBN 9780471837411. Citado na página [61](#).

LIE, F. A. P.; GO, T. H. A collision-free formation reconfiguration control approach for unmanned aerial vehicles. **International Journal of Control, Automation and Systems**, Springer Nature, v. 8, n. 5, p. 1100–1107, oct 2010. Citado na página [52](#).

LINHUI, L.; MINGHENG, Z.; LIE, G.; YIBING, Z. Stereo vision based obstacle avoidance path-planning for cross-country intelligent vehicle. In: **2009 Sixth International Conference on Fuzzy Systems and Knowledge Discovery**. [S.l.]: IEEE, 2009. Citado na página [31](#).

- LUO, C.; MCCLEAN, S. I.; PARR, G.; TEACY, L.; NARDI, R. D. UAV position estimation and collision avoidance using the extended kalman filter. **IEEE Transactions on Vehicular Technology**, Institute of Electrical and Electronics Engineers (IEEE), v. 62, n. 6, p. 2749–2762, jul 2013. Citado nas páginas 49, 50 e 51.
- LYU, Y.; PAN, Q.; ZHAO, C.; ZHU, H.; TANG, T.; ZHANG, Y. A vision based sense and avoid system for small unmanned helicopter. In: **2015 International Conference on Unmanned Aircraft Systems (ICUAS)**. [S.l.]: IEEE, 2015. Citado na página 45.
- MA, Z.; HU, T.; SHEN, L.; KONG, W.; ZHAO, B. A detection and relative direction estimation method for UAV in sense-and-avoid. In: **2015 IEEE International Conference on Information and Automation**. [S.l.]: IEEE, 2015. Citado nas páginas 65 e 69.
- MAGREE, D.; MOONEY, J. G.; JOHNSON, E. N. Monocular visual mapping for obstacle avoidance on UAVs. **Journal of Intelligent & Robotic Systems**, Springer Nature, v. 74, n. 1-2, p. 17–26, oct 2013. Citado nas páginas 61, 67 e 73.
- MASIERO, A.; FISSORE, F.; GUARNIERI, A.; PIROTTI, F.; VETTORE, A. Uav positioning and collision avoidance based on rss measurements. In: . [s.n.], 2015. v. 40, n. 1W4, p. 219–225. Cited By 5. Disponível em: <<https://www.scopus.com/inward/record.uri?eid=2-s2.0-84974627298&doi=10.5194%2fispresarchives-XL-1-W4-219-2015&partnerID=40&md5=c966225aa1245e4851f918bb742b70c5>>. Citado na página 49.
- MEHRA, R.; SINGH, A. Real time RSSI error reduction in distance estimation using RLS algorithm. In: **2013 3rd IEEE International Advance Computing Conference (IACC)**. [S.l.]: IEEE, 2013. Citado nas páginas 49 e 50.
- MEJIAS, L.; MCNAMARA, S.; LAI, J.; FORD, J. Vision-based detection and tracking of aerial targets for UAV collision avoidance. In: **2010 IEEE/RSJ International Conference on Intelligent Robots and Systems**. [S.l.]: IEEE, 2010. Citado nas páginas 45, 74 e 75.
- MERINO, L.; CABALLERO, F.; DIOS, J. M. de; FERRUZ, J.; OLLERO, A. A cooperative perception system for multiple UAVs: Application to automatic detection of forest fires. **Journal of Field Robotics**, Wiley-Blackwell, v. 23, n. 3-4, p. 165–184, 2006. Citado na página 40.
- MERRELL, P. C.; LEE, D.-J.; BEARD, R. W. Obstacle avoidance for unmanned air vehicles using optical flow probability distributions. In: GAGE, D. W. (Ed.). **Mobile Robots XVII**. [S.l.]: SPIE, 2004. Citado nas páginas 34, 64, 69 e 74.
- MINGUEZ, J.; MONTANO, L. Extending collision avoidance methods to consider the vehicle shape, kinematics, and dynamics of a mobile robot. **IEEE Transactions on Robotics**, v. 25, n. 2, p. 367–381, 2009. ISSN 15523098. Citado na página 33.
- MOLLOY, T. L.; FORD, J. J.; MEJIAS, L. Looming aircraft threats: Shape-based passive ranging of aircraft from monocular vision. In: **Australasian Conference on Robotics and Automation, ACRA**. [S.l.: s.n.], 2014. v. 02-04-Dece. Citado na página 45.
- MOON, J.; PRASAD, J. Minimum-time approach to obstacle avoidance constrained by envelope protection for autonomous UAVs. **Mechatronics**, Elsevier BV, v. 21, n. 5, p. 861–875, aug 2011. Citado nas páginas 29 e 30.
- MORI, T.; SCHERER, S. First results in detecting and avoiding frontal obstacles from a monocular camera for micro unmanned aerial vehicles. In: **2013 IEEE International Conference on Robotics and Automation**. [S.l.]: IEEE, 2013. Citado nas páginas 62, 68 e 73.

NETO, A. A.; MACHARET, D. G.; CAMPOS, M. F. M. On the generation of trajectories for multiple UAVs in environments with obstacles. **Journal of Intelligent and Robotic Systems**, Springer Nature, v. 57, n. 1-4, p. 123–141, aug 2009. Citado na página 52.

NONAMI, K. Classification of uav platforms. In: **Modeling and Control of Unmanned Small Scale Rotorcraft UAVs & MAVs**. Japan: Springer Japan, 2010. cap. Chapter 7, p. 10–12. Citado na página 29.

PANTILIE, C. D.; NEDEVSCI, S. Real-time obstacle detection in complex scenarios using dense stereo vision and optical flow. In: **13th International IEEE Conference on Intelligent Transportation Systems**. [S.l.]: IEEE, 2010. Citado na página 31.

PRAATS, B.; BERKVEN, R.; ERGEERTS, G.; WEYN, M. Large scale distributed localization based on rss and mass-spring model. In: **2015 10th International Conference on P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC)**. [S.l.: s.n.], 2015. p. 629–634. Citado na página 49.

RIMON, E.; KODITSCHKEK, D. E. Exact robot navigation using artificial potential functions. **IEEE Robotics and Automation**, v. 8, n. 5, p. 501–518, 1992. ISSN 1042296X. Disponível em: <http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=163777>. Citado nas páginas 33 e 52.

ROCHA, G. M. da; BRANCO, L. H. C.; INOUE, R. S.; BRANCO, K. R. L. J. C. Mosaicism automática e em tempo real de imagens obtidas a partir de vants de asa rotativa. **Journal on Advances in Theoretical and Applied Informatics**, v. 1, n. 1, p. 72–82, 2015. Citado na página 48.

ROJAS-PEREZ, L. O.; MARTINEZ-CARRANZA, J. Metric monocular SLAM and colour segmentation for multiple obstacle avoidance in autonomous flight. In: **2017 Workshop on Research, Education and Development of Unmanned Aerial Systems (RED-UAS)**. [S.l.]: IEEE, 2017. Citado nas páginas 34, 61, 67 e 73.

ROSS, S.; MELIK-BARKHUDAROV, N.; SHANKAR, K. S.; WENDEL, A.; DEY, D.; BAGNELL, J. A.; HEBERT, M. Learning monocular reactive UAV control in cluttered natural environments. In: **2013 IEEE International Conference on Robotics and Automation**. [S.l.]: IEEE, 2013. Citado nas páginas 64, 68, 73 e 74.

ROSTEN, E.; DRUMMOND, T. Machine learning for high-speed corner detection. In: **Computer Vision–ECCV 2006**. [S.l.]: Springer, 2006. p. 430–443. Citado na página 48.

RUBLEE, E.; RABAUD, V.; KONOLIGE, K.; BRADSKI, G. Orb: an efficient alternative to sift or surf. In: IEEE. **Computer Vision (ICCV), 2011 IEEE International Conference on**. [S.l.], 2011. p. 2564–2571. Citado nas páginas 47 e 48.

_____. ORB: An efficient alternative to SIFT or SURF. **Proceedings of the IEEE International Conference on Computer Vision**, p. 2564–2571, 2011. ISSN 1550-5499. Citado na página 74.

Sá De Mello, A.; CHIARAMONTE, R.; FILHO, L.; INOUE, R.; Castelo Branco, K. Simulation results of cooperative formation flying control of UAVs with collision avoidance based on ORCA. In: **APISAT 2015 - 7th Asia-Pacific International Symposium on Aerospace Technology**. [S.l.: s.n.], 2015. Citado nas páginas 35, 91, 95 e 116.

SAHA, S.; NATRAJ, A.; WAHARTE, S. A real-time monocular vision-based frontal obstacle detection and avoidance for low cost UAVs in GPS denied environment. In: **2014 IEEE International Conference on Aerospace Electronics and Remote Sensing Technology**. [S.l.]: IEEE, 2014. Citado nas páginas 29, 34, 65, 69, 74, 78, 79, 82, 84, 88, 97 e 109.

SAUNDERS, J. B.; CALL, B.; CURTIS, A.; BEARD, R. W.; MCLAIN, T. W. Static and Dynamic Obstacle Avoidance in Miniature Air Vehicles. **AIAA InfotechAerospace**, American Institute of Aeronautics and Astronautics, n. AIAA-2005-6950, p. 1–14, 2005. Citado na página 30.

SCHNEIDERMAN, R. Unmanned drones are flying high in the military/aerospace sector [Special Report]. **IEEE Signal Processing Magazine**, v. 29, n. 1, p. 8–11, 2012. Citado na página 32.

SERRA, J. **Image Analysis and Mathematical Morphology**. [S.l.]: Academic Press, INC., 1982. Citado na página 44.

SHAH, S.; JOHNSON, E. 3-d obstacle detection using a single camera. In: **AIAA Guidance, Navigation, and Control Conference**. [S.l.]: American Institute of Aeronautics and Astronautics, 2009. Citado nas páginas 34, 64 e 74.

SNAPE, J.; BERG, J. V. D.; GUY, S. J.; MANOCHA, D. The hybrid reciprocal velocity obstacle. **IEEE Transactions on Robotics**, v. 27, n. 4, p. 696–706, 2011. ISSN 15523098. Citado na página 54.

SNAPE, J.; MANOCHA, D. Navigating multiple simple-airplanes in 3D workspace. **Proceedings - IEEE International Conference on Robotics and Automation**, p. 3974–3980, 2010. ISSN 10504729. Citado nas páginas 54 e 58.

SNAPE, J.; Van Den Berg, J.; GUY, S. J.; MANOCHA, D. Independent navigation of multiple mobile robots with hybrid reciprocal velocity obstacles. **2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2009**, p. 5917–5922, 2009. Citado na página 54.

_____. Smooth and collision-free navigation for multiple robots under differential-drive constraints. **IEEE/RSJ 2010 International Conference on Intelligent Robots and Systems, IROS 2010 - Conference Proceedings**, p. 4584–4589, 2010. ISSN 2153-0858. Citado na página 54.

SOUZA, J. R.; SALES, D. O.; SHINZATO, P. Y.; WOLF, D. F. Template-based autonomous navigation and obstacle avoidance in urban environments. **ACM Applied Computing Review**, 2011. Citado na página 31.

SRIKANTHAKUMAR, S.; LIU, C.; CHEN, W. H. Optimization-based safety analysis of obstacle avoidance systems for unmanned aerial vehicles. **Journal of Intelligent & Robotic Systems**, Springer Nature, v. 65, n. 1-4, p. 219–231, sep 2011. Citado na página 31.

STOWERS, J.; HAYES, M.; BAINBRIDGE-SMITH, A. Biologically inspired UAV obstacle avoidance and control using monocular optical flow & divergence templates. In: **The 5th International Conference on Automation, Robotics and Applications**. [S.l.]: IEEE, 2011. Citado nas páginas 65 e 69.

THE OpenCV Reference Manual. [S.l.], 2014. Citado na página 42.

THRUN, S.; BUECKEN, A.; BURGARD, W.; FOX, D.; WOLFRAM, A. B.; FOX, B. D.; FRÖHLINGHAUS, T.; HENNIG, D.; HOFMANN, T.; KRELL, M. *et al.* Map learning and high-speed navigation in rhino. Citeseer, 1996. Citado na página 52.

TIRRI, A. E.; FASANO, G.; ACCARDO, D.; MOCCIA, A.; De Lellis, E. Advanced sensing issues for UAS collision avoidance. **Proceedings of the 2nd International Conference on Application and Theory of Automation in Command and Control Systems**, IRIT Press, France, n. May, p. 12–19, may 2012. Citado na página 52.

Trindade Jr., O.; JORGE, L. A. d. C.; AGUIAR, J. G. B. Field of dreams using uavs for precision farming. **Unmanned Systems**, v. 22, n. 6, p. 35–39, 2004. Citado na página 29.

TRINDADE, O.; NERIS, L. de O.; BARBOSA, L. C. P.; BRANCO, K. R. L. J. C. A layered approach to design autopilots. In: **2010 IEEE International Conference on Industrial Technology**. [S.l.]: IEEE, 2010. Citado nas páginas 30 e 32.

UNMANNED aircraft systems: federal actions needed to ensure safety and expand their potential uses within the national airspace system. [S.l.]: U.S. Government Accountability Office, 2008. Citado na página 29.

VALAVANIS, K.; KONTITSIS, M. A Historical Perspective on Unmanned Aerial Vehicles. In: VALAVANIS, K. P. (Ed.). **Advances in Unmanned Aerial Vehicles**. [S.l.]: Springer Netherlands, 2007, (Intelligent Systems, Control and Automation: Science and Engineering, v. 33). p. 15–46. ISBN 978-1-4020-6113-4. Citado na página 32.

VALAVANIS, K. P. **Advances in Unmanned Aerial Vehicles: State of the Art and the Road to Autonomy**. 1st. ed. USA: Springer Publishing Company, Incorporated, 2007. ISBN 1402061137, 9781402061134. Citado na página 29.

VALAVANIS, K. P.; OH, P.; PIEGL, L. A. (Ed.). **Unmanned Aircraft Systems**. [S.l.]: Springer Netherlands, 2009. Citado na página 37.

VALAVANIS, K. P.; VACHTSEVANOS, G. J. **Handbook of Unmanned Aerial Vehicles**. USA: Springer Publishing Company, Incorporated, 2015. ISBN 9048197082, 9789048197088. Citado na página 30.

Van Berg, J. D.; LIN, M.; MANOCHA, D. Reciprocal velocity obstacles for real-time multi-agent navigation. **Proceedings - IEEE International Conference on Robotics and Automation**, p. 1928–1935, 2008. ISSN 10504729. Citado nas páginas 34, 52 e 54.

WEISS, S.; ACHELTIK, M.; KNEIP, L.; SCARAMUZZA, D.; SIEGWART, R. Intuitive 3d maps for MAV terrain exploration and obstacle avoidance. **Journal of Intelligent & Robotic Systems**, Springer Nature, v. 61, n. 1-4, p. 473–493, nov 2010. Citado nas páginas 60, 67 e 73.

WILLIAMS, R.; KONEV, B.; COENEN, F. Multi-agent environment exploration with AR.drones. In: **Advances in Autonomous Robotics Systems**. [S.l.]: Springer International Publishing, 2014. p. 60–71. Citado nas páginas 61, 66, 67 e 73.

YANG, X.; ALVAREZ, L. M.; BRUGGEMANN, T. A 3d collision avoidance strategy for UAVs in a non-cooperative environment. **Journal of Intelligent & Robotic Systems**, Springer Nature, v. 70, n. 1-4, p. 315–327, aug 2012. Citado nas páginas 31, 52 e 59.

YEARBOOK, U. Unmanned aircraft systems—the global perspective 2011/2012. **Blyenburg & Co, June**, p. 1967–1709, 2011. Citado nas páginas 33 e 59.

- YI, Z.; CAI, W.; YUE, W. Adaptive safety critical middleware for distributed and embedded safety critical system. In: NCM 08: PROCEEDINGS OF THE 2008 FOURTH INTERNATIONAL CONFERENCE ON NETWORKED COMPUTING AND ADVANCED INFORMATION MANAGEMENT. Washington, DC: IEEE Computer Society, 2008. v. 1, p. 162–166. Citado na página [29](#).
- YU, L.; FEI, Q.; GENG, Q. Combining zigbee and inertial sensors for quadrotor uav indoor localization. In: **2013 10th IEEE International Conference on Control and Automation (ICCA)**. [S.l.: s.n.], 2013. p. 1912–1916. ISSN 1948-3449. Citado na página [49](#).
- ZARANDY, A.; ZSEDROVITS, T.; PENCZ, B.; NAMETH, M.; VANEK, B. A novel algorithm for distant aircraft detection. In: **2015 International Conference on Unmanned Aircraft Systems (ICUAS)**. [S.l.]: IEEE, 2015. Citado na página [45](#).
- ZÀRUBA, G. V.; HUBER, M.; KAMANGAR, F. A.; CHLAMTAC, I. Indoor location tracking using RSSI readings from a single wi-fi access point. **Wireless Networks**, Springer Nature, v. 13, n. 2, p. 221–235, jun 2006. Citado na página [49](#).
- ZEYU, J.; HUI, C.; ZEFENG, Z.; XIAOWU, Z.; XINYAO, N.; WEIBIN, L.; YUXIANG, Z.; SHING, W. W. Autonomous formation flight of uavs: Control algorithms and field experiments. In: IEEE. **Control Conference (CCC), 2016 35th Chinese**. [S.l.], 2016. p. 7585–7591. Citado na página [54](#).
- ZHANG, F.; GOUBRAN, R.; STRAZNICKY, P. Obstacle detection for low flying UAS using monocular camera. In: **2012 IEEE International Instrumentation and Measurement Technology Conference Proceedings**. [S.l.]: IEEE, 2012. Citado nas páginas [61](#), [67](#) e [73](#).
- ZOU, X.; ALEXANDER, R.; MCDERMID, J. Testing method for multi-uav conflict resolution using agent-based simulation and multi-objective search. **Journal of Aerospace Information Systems**, American Institute of Aeronautics and Astronautics, p. 191–203, 2016. Citado na página [54](#).

DETECÇÃO E DESVIO DE OBSTÁCULOS USANDO VISÃO MONOCULAR - REVISÃO SISTEMÁTICA

A.1 Planejamento

O processo de revisão sistemática deve seguir um protocolo pré-estabelecido. Este protocolo deve especificar quais os objetivos da revisão, a questão principal que deve ser respondida por meio da revisão, as palavras-chave utilizadas nas buscas, critérios para definição de bases de busca e critérios para inclusão e exclusão de artigos. Esta seção apresenta o planejamento referente à revisão sistemática.

A.1.1 *Objetivos*

O principal objetivo motivador desta revisão sistemática é identificar e classificar os trabalhos voltados a *obstacle avoidance* especificamente em VANTs e que utilizam de métodos de visão monocular para a fase de detecção.

A.1.2 *Questão Principal*

Uma das etapas mais importantes da revisão sistemática é a formulação da questão principal. É ela que norteia o processo de revisão e conduz todas as suas etapas. Os critérios de busca, critérios de inclusão e critérios de exclusão são confeccionados com base nesta questão principal.

Como o objetivo principal da revisão sistemática é realizar levantamento, análise e classificação de todos os trabalhos que envolvam esforços para *obstacle avoidance* em veículos

aéreos não tripulados e que utilizam métodos de visão computacional para a fase de detecção, a questão principal pode ser definida como:

Questão: Quais são os principais algoritmos e métodos utilizados para a detecção e desvio de obstáculos em veículos aéreos não tripulados que utilizam visão monocular como base para detecção?

Esta questão considera as seguintes especificidades:

População: Serão observados projetos relacionados a veículos aéreos não tripulados com enfoque principal em *obstacle avoidance*;

Intervenção: As principais abordagens de *obstacle avoidance* aplicados a veículos aéreos não tripulados baseadas em visão monocular serão analisadas e classificadas;

Comparação: Classificação dos trabalhos quanto ao enfoque de pesquisa dentro da área de *obstacle avoidance*;

Resultado: Identificação das principais abordagens utilizadas para *obstacle avoidance* em veículos aéreos não tripulados e uma classificação dos trabalhos com enfoque em detecção utilizando visão computacional.

A.1.3 Seleção das Fontes e Idiomas

As fontes de busca e os idiomas de trabalho são de vital importância para a revisão sistemática. Nesta seção são apresentados os critérios que motivaram as escolhas pelos mecanismos de busca e idiomas considerados nesta revisão sistemática.

Critério de definição das fontes

A definição das fontes de busca utilizada nesta revisão sistemática considerou critérios como cobertura, conteúdo atualizado e disponibilidade de trabalhos. Foram consideradas fontes que possuem grande importância e abrangência dentro das áreas de Ciência da Computação e Engenharias, apresentando trabalhos atualizados sobre o tema de busca e que permitam o acesso na íntegra dos trabalhos.

Com base nestes critérios foram escolhidas as seguintes bases de busca eletrônica:

- IEEEExplore Digital Library;
- Scopus;
- Springer;
- Web Of Science.

Idiomas dos trabalhos

Os idiomas considerados nesta revisão sistemática são o Inglês e o Português. Uma vez que o Inglês é o idioma internacionalmente aceito para redação de trabalhos científicos e o Português é o idioma oficial brasileiro.

Palavras-chave

As consultas referentes à revisão sistemática foram compostas utilizando as seguintes palavras-chave e seus respectivos sinônimos:

- Veículos aéreos não tripulados: *unmanned aerial vehicles*, UAV, *unmanned aerial systems*, UAS;
- Detecção e desvio de obstáculos: *obstacle avoidance*, *obstacle detection*, *collision avoidance*;
- Baseados em visão computacional: *vision-based*, *computer vision*, *stereo*, *monocular*.

A.1.4 Critérios de Seleção dos Estudos

Em uma revisão sistemática, os trabalhos resultantes das buscas são selecionados de acordo com critérios bem definidos. Isto contribui para que o foco da revisão sistemática seja mantido durante todo o processo garantindo uma maior credibilidade. A seguir são apresentados os critérios de inclusão e de exclusão de trabalhos na revisão, bem como o procedimento de seleção e análise dos trabalhos.

Critérios de inclusão

Para atender os objetivos definidos na revisão sistemática foram especificados quatro critérios de inclusão. Foram definidos critérios diferenciando trabalhos com enfoque específico na detecção e no desvio de obstáculos. Outros critérios foram utilizados para considerar trabalhos com esforços para detecção em tempo real ou que utilizem mais do que um tipo de sensor para a detecção. Os critérios de inclusão são apresentados a seguir:

- I-1: Estudos que apresentem algoritmos de detecção de obstáculos para veículos aéreos não tripulados utilizando visão monocular;
- I-2: Estudos que apresentem algoritmos de desvio de obstáculos para veículos aéreos não tripulados;
- I-3: Estudos que envolvam esforços para detecção em tempo real.

Critérios de exclusão

Os critérios de exclusão definem que trabalhos que não tenham como foco principal *obstacle avoidance* em veículos aéreos não tripulados devem ser desconsiderados da revisão.

- E-1: Estudos em que os algoritmos não sejam voltados a veículos aéreos não tripulados;
- E-2: Estudos que não focam em algoritmos de detecção ou desvio de obstáculo;
- E-3: Estudos em que não seja possível o acesso na íntegra do artigo;
- E-4: Estudos que sejam em idiomas diferentes do inglês ou do português;
- E-5: Estudos que não sejam completos (resumos).

Código-fonte 1 – Consulta IEEEExplore e Springer Link

```

1: (UAS OR "Unmanned aerial System" OR UAV OR "Unmanned aerial
   Vehicle")
2: AND
3: ("Obstacle Avoidance" OR "Obstacle Detection" OR "Collision
   Avoidance" OR "Sense and Avoidance" OR "Sense and Avoid")
4: AND
5: ("monocular")

```

Código-fonte 2 – Consulta Web-Of-Science

```

1: (TS=((UAS OR "Unmanned aerial System" OR UAV OR "Unmanned
   aerial Vehicle") AND ("Obstacle Avoidance" OR "Obstacle
   Detection" OR "Collision Avoidance" OR "Sense and Avoidance"
   OR "Sense and Avoid") AND ("monocular"))))

```

Código-fonte 3 – Consulta Scopus

```

1: TITLE-ABS-KEY(
2: (UAS OR "Unmanned aerial System" OR UAV OR "Unmanned aerial
   Vehicle")
3: AND
4: ("Obstacle Avoidance" OR "Obstacle Detection" OR "Collision
   Avoidance" OR "Sense and Avoidance" OR "Sense and Avoid")
5: AND
6: ("monocular")
7: )

```