

**MINISTÉRIO DA DEFESA  
EXÉRCITO BRASILEIRO  
DEPARTAMENTO DE CIÊNCIA E TECNOLOGIA  
INSTITUTO MILITAR DE ENGENHARIA  
CURSO DE MESTRADO EM SISTEMAS E COMPUTAÇÃO**

**FÁBIO LUIZ JUNIOR**

**ESTRATÉGIA LÍDER-SEGUIDOR UTILIZANDO VANTS PARA A  
NAVEGAÇÃO ATRAVÉS DO RECONHECIMENTO DE IMAGENS  
DO TERRENO**

**Rio de Janeiro  
2020**

**FÁBIO LUIZ JUNIOR**

**ESTRATÉGIA LÍDER-SEGUIDOR UTILIZANDO VANTS PARA A  
NAVEGAÇÃO ATRAVÉS DO RECONHECIMENTO DE IMAGENS  
DO TERRENO**

Dissertação de Mestrado apresentada ao Programa de Pós-graduação em Sistemas e Computação do Instituto Militar de Engenharia, como requisito parcial para a obtenção do título de Mestre em Ciências em Sistemas e Computação.

Orientador: Paulo Fernando Ferreira Rosa - Ph.D.

Rio de Janeiro

2020

©2020

INSTITUTO MILITAR DE ENGENHARIA

Praça General Tibúrcio, 80 – Praia Vermelha

Rio de Janeiro – RJ CEP: 22290-270

Este exemplar é de propriedade do Instituto Militar de Engenharia, que poderá incluí-lo em base de dados, armazenar em computador, microfilmар ou adotar qualquer forma de arquivamento.

É permitida a menção, reprodução parcial ou integral e a transmissão entre bibliotecas deste trabalho, sem modificação de seu texto, em qualquer meio que esteja ou venha a ser fixado, para pesquisa acadêmica, comentários e citações, desde que sem finalidade comercial e que seja feita a referência bibliográfica completa.

Os conceitos expressos neste trabalho são de responsabilidade do autor e do orientador.

Luiz Junior, Fábio

Estratégia líder-seguidor utilizando VANTs para a navegação através do reconhecimento de imagens do terreno / Fábio Luiz Junior. – Rio de Janeiro, 2020.

120 f.

Orientador: Paulo Fernando Ferreira Rosa.

Dissertação (mestrado) – Instituto Militar de Engenharia, Sistemas e Computação, 2020.

1. Navegação Visual. 2. Drone. 3. VANT. 4. Reconhecimento de imagem.  
5. OpenCV. I. Rosa, Paulo Fernando Ferreira, orient. II. Título

**FÁBIO LUIZ JUNIOR**

**Estratégia líder-seguidor utilizando VANTs para a navegação através do reconhecimento de imagens do terreno**

Dissertação de Mestrado apresentada ao Programa de Pós-graduação em Sistemas e Computação do Instituto Militar de Engenharia, como requisito parcial para a obtenção do título de Mestre em Ciências em Sistemas e Computação.

Orientador: Paulo Fernando Ferreira Rosa

Aprovado em Rio de Janeiro, 24 de Janeiro de 2020, pela seguinte banca examinadora:



---

Prof. Paulo Fernando Ferreira Rosa - Ph.D. do IME - Presidente



---

Prof. Luiz Carlos Pacheco Rodrigues Velho - Ph.D. do IMPA



---

Prof. Esteban Walter Gonzalez Clua - D.Sc. da UFF



---

Profª. Claudia Marcela Justel - D.Sc. do IME

Rio de Janeiro

2020

*Ao Instituto Militar de Engenharia,  
alicerce da minha formação e aperfeiçoamento.*

# AGRADECIMENTOS

Agradeço primeiramente a Deus, por me fornecer saúde, paz e motivação para o estudo continuado e o trabalho. Agradeço à minha esposa e aos meus pais, por estarem sempre ao meu lado e me apoiarem nos momentos mais difíceis dessa jornada. Agradeço ao professor Paulo Rosa, pela paciência, ensinamento, dedicação e disponibilidade para me orientar na execução do mestrado. Agradeço aos meus amigos do Programa de Pós-Graduação, Leandro Moreira, Erick Menezes, Ricardo Maroquio, Luís Silva e Thiago Baldivieso, pelo companheirismo nas inúmeras horas de laboratório. Particularmente, agradeço ao Leandro Moreira por me fornecer preciosas sugestões de melhoria sobre o meu trabalho, e ao Ricardo Maroquio, por compartilhar comigo o código fonte de um simulador de quadricóptero, por ele desenvolvido na plataforma Unity. Por fim, agradeço aos professores do Instituto Militar de Engenharia, por dividirem comigo seus preciosos conhecimentos sobre engenharia.

*“O importante é não parar de questionar.  
A curiosidade tem a sua própria razão  
para existir.”  
(Albert Einstein)*

# RESUMO

Os Veículos Aéreos Não Tripulados (VANTs) são aeronaves que não necessitam de um piloto humano a bordo, podendo ser controlados por um computador embarcado ou por um operador à distância. Nos últimos anos, os VANTs vêm ganhando cada vez mais importância no cenário mundial devido às suas inúmeras possibilidades de aplicação. Neste contexto, as pesquisas sobre VANTs têm buscado introduzir um nível cada vez maior de inteligência embarcada, possibilitando que a máquina tome algumas decisões automaticamente, como o controle do seu trajeto de voo e a resposta a estímulos externos. Uma das linhas de pesquisa que está sendo empregada na automatização de voos é a utilização da visão computacional, a qual está em grande ascensão recentemente.

Este trabalho de dissertação realiza um estudo e implementa um software de navegação de um VANT utilizando o reconhecimento de imagens do terreno. O conceito principal envolvido é navegar um VANT de maneira autônoma, capturando e utilizando as imagens do terreno como *waypoints* visuais de uma trajetória. Para realizar o pretendido, utilizam-se mecanismos de visão computacional e um VANT com uma câmera embarcada. Divide-se o funcionamento do software de navegação em duas etapas principais, as quais serão descritas a seguir.

Na primeira etapa, é realizado um voo de reconhecimento e aquisição das imagens do terreno. Neste voo, um VANT líder fará uma trajetória qualquer, controlado manualmente por um operador. Durante este voo, realizam-se a aquisição das imagens do terreno, o seu processamento e o armazenamento das informações resultantes em um banco de dados de trajetórias. Tais informações irão compor o percurso que se deseja realizar no voo autônomo.

Na segunda etapa, são realizados voos autônomos utilizando as imagens colhidas no primeiro voo. Os dados da trajetória são carregados no software de navegação. O controle da trajetória é realizado através do processamento em tempo real das informações obtidas pela câmera embarcada. A partir disso, são enviados os comandos necessários ao piloto automático da aeronave para reconstruir a trajetória original coletada na primeira etapa.

Por fim, destaca-se, como contribuição obtida por este trabalho, o desenvolvimento de uma estação de solo capaz de realizar a navegação de VANTs conforme explicitado. Para tanto, foi utilizado o SURF (*Speeded-Up Robust Features*) como mecanismo de visão computacional para realizar o reconhecimento das imagens do terreno. Além da navegação, o sistema desenvolvido possui também a capacidade para fazer a aquisição, persistência e compartilhamento, entre os VANTs, da trajetória executada. Nos experimentos executados,



foram obtidos resultados satisfatórios, tanto para as simulações, quanto para os voos com a plataforma robótica utilizada. Finalmente, frisa-se que o sistema implementado é independente dos mecanismos tradicionais de georreferenciamento por satélites.

**Palavras-chave:** Navegação Visual. Drone. Reconhecimento de imagem. OpenCV. VANT.

# ABSTRACT

Unmanned Aerial Vehicles (UAVs) are aircrafts that do not require a human pilot on board and can be controlled by an on-board computer or a remote operator. In recent years, UAVs have become increasingly important on the world stage due to their numerous application possibilities. In this context, UAV research seeks to introduce an ever-increasing level of embedded intelligence, enabling the machine to make certain decisions automatically, such as controlling its flight path and responding to external stimuli. One of the lines of research employed in flight automation is the use of computer vision.

This work conducts a study and implements a UAV navigation software using terrain image recognition. The main concept involved is navigating a UAV autonomously, capturing and using images of the ground as visual waypoints of a trajectory. To accomplish the desired, computational vision mechanisms and a UAV with an embedded camera are used. The operation of the navigation software is divided into 2 main steps described below.

In the first stage, a flight of recognition and acquisition of terrain images is performed. In this flight, a leader UAV makes any trajectory manually controlled by an operator. During this flight, the terrain images are acquired, processed and stored in a trajectory database. Thus, this information will compose the route you wish to take on the autonomous flight.

In the second stage, autonomous flights are performed using the images taken on the first flight. The trajectory data is loaded into the navigation software. Trajectory control is performed through real-time processing of information obtained by the onboard camera. From this, the necessary commands are sent to the autopilot of the aircraft to remake the original trajectory collected in the first stage.

Finally, I highlight, as a contribution obtained by this work, the development of a ground station capable of performing UAV navigation as explained. For that, the SURF (*Speeded-Up Robust Features*) was used as a computer vision mechanism to perform the recognition of the terrain images. In addition to navigation, the developed system also has the capacity to make the acquisition, persistence and sharing, between UAVs, of the executed trajectory. In the experiments performed, satisfactory results were obtained, both for simulations and for flights with the robotic platform used. Lastly, it is emphasized that the system implemented is independent of the traditional mechanisms of georeferencing by satellites.

**Keywords:** Visual Navigation. Drone. Image Recognition. OpenCV. UAV.

# LISTA DE ILUSTRAÇÕES

Figura 1 – Ilustração da navegação através do reconhecimento de imagens do terreno.	23
Figura 2 – Exemplo de VANT de asa fixa.	29
Figura 3 – Exemplo de VANT de asa rotativa.	29
Figura 4 – Exemplo de VANT dirigível.	29
Figura 5 – Exemplo de VANT de asa móvel. Fonte [1]	29
Figura 6 – Rotação do VANT nos três eixos ( <i>roll</i> , <i>pitch</i> e <i>yaw</i> ).	31
Figura 7 – Relação entre a configuração dos motores de um quadricóptero e os movimentos resultantes da aeronave.	31
Figura 8 – Exemplos de <i>frames</i> para aeronaves de asa rotativa no Ardupilot. Figura adaptada de [2].	32
Figura 9 – Princípio básico de funcionamento de um acelerômetro MEMS. A parte externa é fixa e a parte interna é móvel.	34
Figura 10 – Princípio de funcionamento de um giroscópio MEMS por efeito Coriolis.	35
Figura 11 – Exemplo de um sensor giroscópio comercial utilizado em VANTs.	35
Figura 12 – Princípio básico de funcionamento de um barômetro MEMS.	36
Figura 13 – Exemplo de um sensor barômetro MEMS comercial utilizado em VANTs.	36
Figura 14 – Princípio de funcionamento de um magnetômetro por efeito Hall.	37
Figura 15 – Exemplo de um sensor magnetômetro comercial utilizado em VANTs.	37
Figura 16 – Funcionamento de um sensor ultrassônico na medição de distâncias.	37
Figura 17 – Exemplo de um sensor ultrassônico comercial utilizado em VANTs.	37
Figura 18 – Constelação de satélites do sistema GPS. O conjunto mínimo é formado por 24 satélites divididos em seis órbitas. Fonte da imagem [3].	38
Figura 19 – Transmissão dos parâmetros posição e tempo dos 4 satélites para o receptor GPS.	39
Figura 20 – Exemplo de um sensor GPS comercial utilizado em VANTs.	39
Figura 21 – Princípio básico de funcionamento de uma câmera baseado no modelo da câmara escura. Imagem adaptada de [4].	39
Figura 22 – Exemplo de câmera comercial utilizada em VANTs.	40
Figura 23 – Exemplo de funcionamento de um controlador On-Off.	41
Figura 24 – Malha fechada de controle utilizando um controlador PID.	42
Figura 25 – Exemplos de keypoints, destacados em azul, de uma imagem aérea obtida por VANT.	44
Figura 26 – Ilustração da homografia. Mapeamento do ponto $Q$ de uma superfície plana no ponto $q$ de outra superfície também plana. Imagem retirada e adaptada de [4].	45

Figura 27 – Exemplo de comando visual enviado ao VANT para execução e controle do voo. Figura retirada do artigo [5]. . . . .	46
Figura 28 – Demonstração do VANT em voo estabilizado utilizando visão computacional. Figura retirada de [6] . . . . .	47
Figura 29 – Rastreamento de <i>features</i> para estimar o ângulo de <i>heading</i> e a velocidade vertical da aeronave. Figura retirada do artigo [7]. . . . .	47
Figura 30 – VANT capaz de seguir formas lineares extraídas do ambiente. Figura adaptada do artigo [8]. . . . .	48
Figura 31 – Quadricóptero capaz de realizar o pouso em uma plataforma móvel através da identificação de um marcador vermelho. Figura retirada de [9]	49
Figura 32 – Bloqueador GPS. Fonte [10]. . . . .	53
Figura 33 – Diagrama com as etapas do voo de reconhecimento e aquisição de dados do trajeto. . . . .	57
Figura 34 – Voo de reconhecimento e aquisição das imagens do terreno. Neste voo, o VANT líder fará a aquisição das imagens do terreno e a transmissão para a estação de solo. . . . .	57
Figura 35 – Diagrama com as etapas do voo de navegação por imagem. . . . .	59
Figura 36 – Voo de navegação por imagem. Neste voo, os dados do percurso são carregados no VANT seguidor e, utilizando reconhecimento de imagens do terreno, a aeronave é capaz de refazer o trajeto carregado. . . . .	59
Figura 37 – Estratégia de navegação através da perseguição do <i>waypoint</i> visível de maior índice. . . . .	60
Figura 38 – Estratégia de navegação com visitação de todos os <i>waypoints</i> . . . . .	60
Figura 39 – VANT seguidor visualizando um <i>waypoint</i> no terreno. . . . .	61
Figura 40 – Ângulos de abertura da câmera vertical embarcada no VANT. Na direita, tem-se a abertura horizontal. Na esquerda, tem-se a abertura vertical. . . . .	61
Figura 41 – Distância do VANT até o centro do <i>waypoint</i> quando a aeronave não está paralela ao terreno. . . . .	63
Figura 42 – Rotação e translação da aeronave para se alcançar o <i>waypoint</i> visual. . . . .	64
Figura 43 – Nesta figura, estão representados a aeronave, a estação de solo e o <i>joystick</i> para controle manual. . . . .	65
Figura 44 – Arquitetura de comunicação do sistema. . . . .	66
Figura 45 – Diagrama com os principais componentes do software da estação de solo. . . . .	69
Figura 46 – Diagrama de sequência representando uma iteração no ciclo de processamento e controle durante o voo do VANT seguidor. . . . .	70
Figura 47 – Interface gráfica do software da estação de solo. . . . .	71
Figura 48 – Imagem de satélite do Parque Olímpico obtida em março de 2018. Fonte: <i>Google Earth</i> [11]. . . . .	75

Figura 49 – Imagem de satélite do Parque Olímpico obtida em maio de 2018. Fonte: <i>Google Earth</i> [11]. . . . .	75
Figura 50 – Recorte de um trecho da Figura 48, de março de 2018, e extração dos seus <i>keypoints</i> . . . . .	76
Figura 51 – Reconhecimento do trecho na Figura 50, obtida em março de 2018, na Figura 49 obtida em maio de 2018. . . . .	76
Figura 52 – Recorte de um trecho da Figura 48, de março de 2018, e extração dos seus <i>keypoints</i> . . . . .	77
Figura 53 – Reconhecimento do trecho na Figura 52, obtida em março de 2018, na Figura 49 obtida em maio de 2018. . . . .	77
Figura 54 – Mosaico construído através da concatenação de todas as imagens obtidas durante o segundo voo do VANT sobre o terreno. Fonte: [12] . . . . .	78
Figura 55 – Subconjunto das imagens obtidas durante o primeiro voo da aeronave sobre o terreno. Fonte: [12] . . . . .	79
Figura 56 – Subconjunto das imagens obtidas durante o segundo voo da aeronave sobre o terreno. Fonte: [12] . . . . .	79
Figura 57 – Gráfico com o número de <i>keypoints</i> semelhantes ou compatíveis da imagem <i>DJI_0008.jpg</i> , capturada no primeiro voo, com outras imagens do segundo voo. . . . .	80
Figura 58 – Destaque em azul de alguns <i>keypoints</i> extraídos da imagem <i>DJI_0008.jpg</i>	81
Figura 59 – Equivalência dos <i>keypoints</i> das imagens <i>DJI_0008.jpg</i> e <i>DJI_0380.jpg</i> , comprovando a grande semelhança entre elas. . . . .	81
Figura 60 – Gráfico com o número de <i>keypoints</i> semelhantes ou compatíveis da imagem <i>DJI_0013.jpg</i> , capturada no primeiro voo, com outras imagens do segundo voo. . . . .	82
Figura 61 – Destaque em azul de alguns <i>keypoints</i> extraídos da imagens <i>DJI_0013.jpg</i>	82
Figura 62 – Equivalência dos <i>keypoints</i> das imagens <i>DJI_0013.jpg</i> e <i>DJI_0385.jpg</i> , comprovando a grande semelhança entre elas. . . . .	83
Figura 63 – Experimento de bancada 1: Identificação de múltiplos <i>waypoints</i> na imagem fornecida pela câmera. . . . .	85
Figura 64 – Experimento de bancada 2: Identificação e navegação utilizando <i>waypoints</i> visuais. Os quadros estão dispostos temporalmente, à medida que a câmera se desloca pelo mosaico. . . . .	86
Figura 65 – Captura de tela de uma simulação em execução. Na esquerda, tem-se o software de navegação por imagem. Na direita, tem-se o simulador em Unity. . . . .	88
Figura 66 – Curva de resposta ao deslocamento em $x$ utilizando o controlador On-Off.	88
Figura 67 – Curva de resposta ao deslocamento em $x$ utilizando o controlador PID.	88
Figura 68 – Curva de resposta ao deslocamento em $y$ utilizando o controlador On-Off.	89

Figura 69 – Curva de resposta ao deslocamento em $y$ utilizando o controlador PID.	89
Figura 70 – Curva de resposta à rotação em $z$ utilizando o controlador On-Off.	89
Figura 71 – Curva de resposta à rotação em $z$ utilizando o controlador PID.	89
Figura 72 – Trajeto salvo 1.	90
Figura 73 – Trajeto salvo 2.	90
Figura 74 – Trajeto salvo 3.	90
Figura 75 – Trajeto salvo 4.	90
Figura 76 – Trajeto 1 realizado utilizando a configuração 1.	92
Figura 77 – Trajeto 2 realizado utilizando a configuração 1.	93
Figura 78 – Trajeto 3 realizado utilizando a configuração 1.	93
Figura 79 – Trajeto 4 realizado utilizando a configuração 1.	94
Figura 80 – Trajeto 1 realizado utilizando a configuração 2.	95
Figura 81 – Trajeto 2 realizado utilizando a configuração 2.	96
Figura 82 – Trajeto 3 realizado utilizando a configuração 2.	96
Figura 83 – Trajeto 4 realizado utilizando a configuração 2.	97
Figura 84 – Trajeto 1 realizado utilizando a configuração 3.	98
Figura 85 – Trajeto 2 realizado utilizando a configuração 3.	99
Figura 86 – Trajeto 3 realizado utilizando a configuração 3.	99
Figura 87 – Trajeto 4 realizado utilizando a configuração 3.	100
Figura 88 – Trajeto 1 realizado utilizando a configuração 4.	101
Figura 89 – Trajeto 2 realizado utilizando a configuração 4.	101
Figura 90 – Trajeto 3 realizado utilizando a configuração 4.	102
Figura 91 – Trajeto 4 realizado utilizando a configuração 4.	102
Figura 92 – <i>Setup</i> do experimento de alinhamento e centralização sobre o <i>waypoint</i> .	105
Figura 93 – Aeronave pousada sobre o <i>waypoint</i> visual carregado no sistema.	105
Figura 94 – Execução do software de navegação por reconhecimento de imagem.	105
Figura 95 – Posição relativa do VANT sobre o <i>waypoint</i> (Teste 1).	106
Figura 96 – Posição relativa do VANT sobre o <i>waypoint</i> (Teste 2).	106
Figura 97 – Posição relativa do VANT sobre o <i>waypoint</i> (Teste 3).	106
Figura 98 – Posição relativa do VANT sobre o <i>waypoint</i> (Teste 4).	106
Figura 99 – Alinhamento do VANT com o <i>waypoint</i> visual (Teste 5).	107
Figura 100 – Alinhamento do VANT com o <i>waypoint</i> visual (Teste 6).	107
Figura 101 – Alinhamento do VANT com o <i>waypoint</i> visual (Teste 7).	107
Figura 102 – Alinhamento do VANT com o <i>waypoint</i> visual (Teste 8).	107
Figura 103 – VANT seguidor no início do trajeto a ser realizado.	108
Figura 104 – Ilustração de 16 dos 26 <i>waypoints</i> do trajeto executado.	109
Figura 105 – Foto do percurso com alguns <i>waypoints</i> destacados.	109
Figura 106 – Execução da trajetória pelo VANT seguidor.	110
Figura 107 – Trajetória realizada pelo VANT líder.	110

Figura 108–Alinhamento do VANT com o <i>waypoint</i> visual (Teste 9). . . . .	111
Figura 109–Deslocamento do VANT sobre o percurso mapeado (Teste 9). . . . .	111
Figura 110–Alinhamento do VANT com o <i>waypoint</i> visual (Teste 10). . . . .	111
Figura 111–Deslocamento do VANT sobre o percurso mapeado (Teste 10). . . . .	111
Figura 112–Alinhamento do VANT com o <i>waypoint</i> visual (Teste 11). . . . .	111
Figura 113–Deslocamento do VANT sobre o percurso mapeado (Teste 11). . . . .	111
Figura 114–Alinhamento do VANT com o <i>waypoint</i> visual (Teste 12). . . . .	112
Figura 115–Deslocamento do VANT sobre o percurso mapeado (Teste 12). . . . .	112
Figura 116–Gráfico com as durações dos processos de transmissão das imagens, seu processamento e envio dos comandos de controle ao VANT. . . . .	113

# LISTA DE TABELAS

Tabela 1 – Classificação dos VANTs segundo <i>European Association of Unmanned Vehicles Systems</i> (EUROUVS), conforme expresso em [13]. . . . .	30
Tabela 2 – Compilação das tarefas desenvolvidas por cada trabalho relacionado. . . . .	49
Tabela 3 – Compilação das características de cada trabalho relacionado. . . . .	50
Tabela 4 – Alguns métodos de localização. Adaptada de [14]. . . . .	52
Tabela 5 – Taxa de <i>matching</i> variando-se a intensidade e composição de cores das imagens. Extraído de [15]. . . . .	72
Tabela 6 – Desempenho para <i>matching</i> com 45°. Extraído de [15]. . . . .	72
Tabela 7 – Taxa de <i>matching</i> com diferentes ângulos de rotação. Extraído de [15].	72
Tabela 8 – Taxa de <i>matching</i> com adição de 30% de <i>salt and pepper noise</i> . Extraído de [15]. . . . .	73
Tabela 9 – Tabela de desempenho na busca dos <i>waypoints</i> para diferentes resoluções da imagem. . . . .	86
Tabela 10 – Dados dos trajetos 1, 2, 3 e 4 adquiridos pelo VANT líder. . . . .	91
Tabela 11 – Informações dos trajetos 1, 2, 3 e 4 realizados utilizando a configuração 1.	94
Tabela 12 – Informações dos trajetos 1, 2, 3 e 4 realizados utilizando a configuração 2.	95
Tabela 13 – Informações dos trajetos 1, 2, 3 e 4 realizados a configuração 3. . . . .	99
Tabela 14 – Informações dos trajetos 1, 2, 3 e 4 realizados a configuração 4. . . . .	103
Tabela 15 – Dados dos testes de centralização do VANT no <i>waypoint</i> visual. . . . .	105
Tabela 16 – Alinhamento do VANT com o <i>waypoint</i> visual. . . . .	106
Tabela 17 – Informações do trajeto executado. . . . .	108
Tabela 18 – Dados dos testes de execução da trajetória pelo VANT seguidor. . . . .	112
Tabela 19 – Tabela com as latências médias do sistema. . . . .	113



# LISTA DE ABREVIATURAS E SIGLAS

API	Application Programming Interface
CI	Circuito Integrado
CR	Close Range
DDP	Diferença de Potencial
DOF	Degrees of Freedom
ERB	Estação Rádio Base
EUROUVS	European Association of Unmanned Vehicles Systems
FPGA	Field Programmable Gate Array
FPS	Frames per Second
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
GPU	Graphics Processing Unit
HALE	High Altitude, Long Endurance
HD	High Definition
IMU	Inertial Measurement Unit
LR	Long Range
MEMS	Micro-Electro-Mechanical Systems
MR	Medium Range
OpenCV	Open Source Computer Vision Library
ORB	Oriented FAST and Rotated BRIEF
PID	Proporcional, Integral e Derivativo
RFID	Radio-Frequency Identification
RGB	Red, Green and Blue

RMSE	Root Mean Square Error
RPM	Rotações por Minuto
RTL	Return To Launch
SDK	Software Development Kit
SIFT	Scale-Invariant Feature Transform
SNI	Sistema de Navegação Inercial
SR	Short Range
SURF	Speeded-Up Robust Features
TCP	Transmission Control Protocol
UAV	Unmanned Aerial Vehicle
UDP	User Datagram Protocol
VANT	Veículo Aéreo Não Tripulado
VGA	Video Graphics Array
WLAN	Wireless Local Area Network

# LISTA DE SÍMBOLOS

$R_x$	Resolução da câmera no eixo $x$
$R_y$	Resolução da câmera no eixo $y$
$X_{waypoint}$	Coordenada $x$ , em pixel, do <i>waypoint</i> identificado
$Y_{waypoint}$	Coordenada $y$ , em pixel, do <i>waypoint</i> identificado
$E_1$	Distância entre a aeronave e o <i>waypoint</i>
$E_2$	Ângulo entre a aeronave e o <i>waypoint</i>
$\alpha$	Ângulo formado entre o eixo óptico da câmera e a reta que liga a câmera ao ponto $(X_{waypoint}, R_y/2)$
$\beta$	Ângulo formado entre o eixo óptico da câmera e a reta que liga a câmera ao ponto $(Y_{waypoint}, R_x/2)$
$\theta_1$	Ângulo de abertura da câmera no eixo $x$
$\theta_2$	Ângulo de abertura da câmera no eixo $y$
$\varphi$	Ângulo entre os eixos longitudinais da aeronave e do <i>waypoint</i>
$m$	Elemento da matriz homográfica
$K_p$	Ganho proporcional
$K_i$	Ganho integral
$K_d$	Ganho derivativo

# SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>22</b>
<b>1.1</b>	<b>Motivação</b>	<b>23</b>
<b>1.2</b>	<b>Objetivos</b>	<b>25</b>
1.2.1	Objetivo geral	25
1.2.2	Objetivos específicos	25
<b>1.3</b>	<b>Contribuições</b>	<b>26</b>
<b>1.4</b>	<b>Estrutura da dissertação</b>	<b>26</b>
<b>2</b>	<b>FUNDAMENTAÇÃO TEÓRICA</b>	<b>28</b>
<b>2.1</b>	<b>Veículos aéreos não tripulados</b>	<b>28</b>
2.1.1	Classificação por sustentação	28
2.1.2	Classificação por dimensão	29
2.1.3	Dinâmica dos VANTs quadricópteros	30
<b>2.2</b>	<b>Piloto Automático</b>	<b>32</b>
2.2.1	Modos de voo	33
<b>2.3</b>	<b>Sensores</b>	<b>34</b>
<b>2.4</b>	<b>Controladores automáticos</b>	<b>41</b>
2.4.1	Controlador On-Off	41
2.4.2	Controlador PID	42
<b>2.5</b>	<b>Biblioteca OpenCV</b>	<b>43</b>
2.5.1	<i>Keypoints e Descriptors</i>	43
2.5.2	Homografia	44
<b>3</b>	<b>REVISÃO DA LITERATURA</b>	<b>46</b>
<b>4</b>	<b>CARACTERIZAÇÃO DO PROBLEMA</b>	<b>51</b>
<b>4.1</b>	<b>Técnicas de localização</b>	<b>51</b>
<b>4.2</b>	<b>Vulnerabilidades do GPS</b>	<b>52</b>
<b>4.3</b>	<b>Definição do problema</b>	<b>54</b>
<b>5</b>	<b>SOLUÇÃO PROPOSTA</b>	<b>56</b>
<b>5.1</b>	<b>Etapa 1: Voo de aquisição das imagens do terreno</b>	<b>56</b>
<b>5.2</b>	<b>Etapa 2: Voo autônomo guiado por imagens</b>	<b>58</b>
5.2.1	Estratégias para execução da trajetória armazenada	59
5.2.2	Cálculo da posição do VANT em relação ao <i>waypoint</i> visual	61
5.2.3	Cálculo dos movimentos da aeronave	63

5.3	<b>Considerações finais</b>	64
6	<b>IMPLEMENTAÇÃO DA SOLUÇÃO</b>	65
6.1	<b>Aeronave utilizada nos experimentos</b>	65
6.2	<b>Biblioteca de comunicação com o VANT</b>	67
6.3	<b>Estação de solo</b>	68
6.3.1	Aspectos da implementação	68
6.3.2	Interface gráfica	70
6.4	<b>Escolha do algoritmo para <i>image matching</i></b>	72
7	<b>EXPERIMENTOS E RESULTADOS</b>	74
7.1	<b>Experimentos iniciais com imagens aéreas</b>	74
7.1.1	Reconhecimento de imagens aéreas obtidas por satélites	74
7.1.2	Reconhecimento de imagens aéreas obtidas por VANTs	78
7.2	<b>Experimentos de bancada</b>	84
7.2.1	Experimento de bancada 1	84
7.2.2	Experimento de bancada 2	84
7.3	<b>Simulação de navegação por imagem</b>	87
7.3.1	Curvas de resposta do VANT simulado	88
7.3.2	Aquisição das trajetórias pelo VANT líder	90
7.3.3	Execução das Trajetórias pelo VANT seguidor	91
7.3.3.1	Simulação da configuração 1: Piloto automático com controlador On-Off e estratégia de perseguição ao <i>waypoint</i> visível de maior índice.	92
7.3.3.2	Simulação da configuração 2: Piloto automático com controlador On-Off e estratégia de visitação de todos os <i>waypoints</i> .	95
7.3.3.3	Simulação da configuração 3: Piloto automático com controlador PID e estratégia de perseguição ao <i>waypoint</i> visível de maior índice.	98
7.3.3.4	Simulação da configuração 4: Piloto automático com controlador PID e a estratégia de visitação de todos os <i>waypoints</i> .	101
7.4	<b>Experimentos de voo real</b>	104
7.4.1	Experimento de voo para alinhamento e centralização da aeronave sobre o <i>waypoint</i> .	104
7.4.1.1	Resultados	104
7.4.2	Experimento de voo real na execução de um trajeto	108
7.4.2.1	Resultados	109
7.4.3	Dados de latência do sistema	113
8	<b>CONSIDERAÇÕES FINAIS</b>	114
8.1	<b>Conclusão</b>	114
8.2	<b>Trabalhos futuros</b>	115

REFERÊNCIAS	117
-------------	-----

# 1 INTRODUÇÃO

Os Veículos Aéreos Não Tripulados (VANTs) são aeronaves que não necessitam de um piloto humano a bordo, podendo ser controlados por um computador embarcado ou à distância por um operador humano. Nos últimos anos, os VANTs têm ganhado cada vez mais importância no cenário mundial, possuindo inúmeras aplicações em ambientes civis e militares. Como exemplos de utilização destas aeronaves, tem-se: operações de busca e salvamento, vigilância de grandes áreas incluindo fronteiras de países, monitoramento de plantações, inspeção de linhas de transmissão de energia elétrica e conflitos bélicos.

A pesquisa e o desenvolvimento de VANTs pelas Forças Armadas são de vital importância para se criar uma independência tecnológica do Brasil. Além disso, o domínio dessa tecnologia contribuirá, de forma expressiva, no poder dissuasório do país e se encontra alinhado com o Plano Estratégico do Exército 2016-2019 [16]. Aparentemente, há uma tendência de que as guerras do futuro irão utilizar maciçamente VANTs e robôs no campo de batalha, visto que as principais nações do mundo estão realizando grandes investimentos nesta área.

Sobre este assunto, o controle de posicionamento e de trajetória são dois componentes de grande importância nos VANTs, dado que estes normalmente precisam se locomover no terreno, desviar de obstáculos e conhecer a sua posição. Conforme destacado por [17], o sucesso de uma missão está diretamente relacionado com a qualidade da localização e da navegação utilizado pela plataforma robótica. Quando se trata de posicionamento de múltiplos veículos aéreos, a complexidade do problema cresce ainda mais. A possibilidade de colisões, a navegação em ambientes desestruturados, as perturbações causadas pelo meio (vento, chuva, névoa, etc) e a própria instabilidade dinâmica do VANT são fatores que aumentam a complexidade dos algoritmos de controle. Frequentemente, estes algoritmos realizam cálculos em tempo real e necessitam de diversos sensores e poder de processamento embarcados para realizarem o controle da aeronave.

Atualmente, o principal mecanismo de localização e navegação empregado pelos VANTs é o baseado em satélites (*Global Navigation Satellite System* - GNSS). Dentre esses sistemas, o *Global Positioning System* - GPS (Sistema de Posicionamento Global, em português) [3] é o mais comum. Porém, há também outros, como: o Beidou chinês [18], o Glonass russo [19] e o Galileo europeu [20]. Entretanto, nenhum desses sistemas é de propriedade brasileira. Assim, em casos de conflito de interesse entre os países, é possível que os proprietários dessas tecnologias desliguem o sistema seletivamente ou introduzam erros propositais para prejudicar o Brasil. Além disso, sabe-se que esses sistemas de posicionamento não são livres de erros. Eles podem falhar em determinadas situações,

como em áreas de sombra ou multi-reflexões do sinal dos satélites, em ambientes *indoor*, durante tempestades solares intensas, pela utilização de bloqueadores ou *spoofing* do sinal dos satélites, entre outras situações. Tais fatos podem expor a risco o cumprimento da missão que se deseja realizar.

Visando diminuir essa dependência na utilização do GPS pelos VANTs, este trabalho propõe a utilização de técnicas de visão computacional para realizar a localização e navegação sobre trajetórias visualmente mapeadas. Para isso, o sistema proposto é dividido em dois passos principais, os quais envolvem a utilização de um VANT líder e de um VANT seguidor. Inicialmente, o VANT líder realizará um percurso qualquer sobre uma área. O controle desse percurso inicial poderá ser realizado manualmente por um operador humano. Durante esse voo, o VANT líder realizará a captura de diversas imagens do terreno abaixo da aeronave. Tais imagens serão processadas e armazenadas em um banco de dados de trajetórias. Cada uma dessas imagens adquiridas será utilizada posteriormente como um ponto de passagem do percurso original. Tal conceito é conhecido como marco visual ou *waypoint* visual. Após a realização do voo inicial pelo VANT líder, o VANT seguidor entrará em cena. Para isso, os dados do percurso serão carregados pelo sistema de navegação e o VANT seguidor levantará voo partindo do ponto inicial da trajetória. Utilizando técnicas de visão computacional e a câmera embarcada disponível, o VANT seguidor irá refazer o percurso inicial do VANT líder percorrendo, em sequência, todos os *waypoints* visuais adquiridos pelo líder. A Figura 1 ilustra, de maneira simplificada, a navegação utilizando o reconhecimento de imagens do terreno.

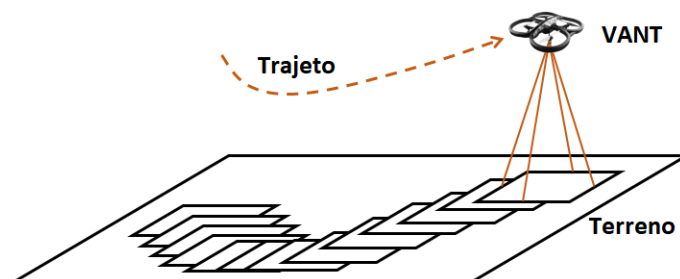


Figura 1 – Ilustração da navegação através do reconhecimento de imagens do terreno.

Por fim, durante a elaboração deste trabalho, foi publicado por este autor um artigo sobre o tema. Intitulado "*Navegação de VANT por Reconhecimento de Imagem do Solo*" [21], esse artigo descreve de maneira sumária alguns dos experimentos descritos nesta dissertação.

## 1.1 Motivação

Como já destacado no início deste capítulo, atualmente os VANTs são muito dependentes dos sistemas de posicionamento por satélites para a realização das mais



diversas tarefas. Como consequência disso, caso o sistema de localização por satélite fique indisponível, mesmo por um pequeno período de tempo, a missão a ser cumprida pela aeronave poderá ser comprometida. Tendo isso em mente, este trabalho possui como motivação utilizar mecanismos de visão computacional para possibilitar a localização e a navegação de VANTs em percursos já conhecidos e mapeados. Assim, a utilização desse mecanismo de navegação proposto permite refazer trajetórias já mapeadas, mesmo em ambientes *GPS-denied*. Abaixo, seguem alguns pontos importantes que reforçam e destacam as vantagens do mecanismo proposto.

- **Possibilidade de armazenamento e compartilhamento da trajetória entre múltiplos VANTs.** No caso da navegação por visão, os dados do percurso são essencialmente imagens de diversos pontos da trajetória. Essas imagens podem ser armazenadas e compartilhadas entre as diversas aeronaves líderes para compor um trajeto maior. Além disso, o trajeto a ser feito pode ser enviado para diversos VANTs seguidores, os quais possuem a missão de refazer aquela rota múltiplas vezes.
- **Não necessita da utilização de marcos artificiais colocados no terreno.** Diferentemente de outros sistemas de navegação por imagem, o mecanismo aqui proposto não necessita da utilização de marcos artificiais colocados no terreno. Os marcos visuais utilizados para a navegação são as próprias imagens do terreno. Assim, os operadores do sistema não precisam de acesso físico por solo para realizar a navegação pela área desejada.
- **O sistema proposto pode ser utilizado para implementar a função *Return-to-Home* baseado em visão.** Atualmente, diversos VANTs comerciais implementam a função *Return-to-Home* baseado em GPS. Esta função permite que a aeronave retorne ao ponto de partida em casos críticos, como bateria baixa e perda do sinal de controle ou, ainda, pelo simples acionamento de um botão pelo operador. Porém, caso a aeronave esteja sem sinal de GPS, essa função poderá ser comprometida, não funcionando corretamente. O *Return-to-Home* baseado em visão computacional pode ser uma alternativa para esses casos.
- **O sistema de localização por imagem não requer grande capacidade de processamento.** Assim, pode-se realizar o embarque do sistema diretamente na aeronave. Isso fará com que o software de navegação por imagem não dependa de nenhum sistema externo ao VANT.
- **Esse sistema de localização pode ser utilizado como mais um mecanismo de localização existente na aeronave.** Através de fusão de dados com outros mecanismos de localização da aeronave, pode-se obter uma maior precisão e confiabilidade no seu posicionamento.

## 1.2 Objetivos

### 1.2.1 Objetivo geral

O objetivo geral deste trabalho é pesquisar, implementar e testar um sistema para navegação de VANTs capaz de adquirir, armazenar e executar trajetórias utilizando mecanismos de visão computacional e imagens aéreas do terreno.

### 1.2.2 Objetivos específicos

Para alcançar o objetivo geral, foram elencados os seguintes objetivos específicos, como pontos intermediários, a serem cumpridos durante o desenvolvimento deste trabalho.

- **Configurar uma aeronave para voo:** Configurar uma aeronave quadrirrotor visando realizar voos de teste do algoritmo proposto. A aeronave utilizada deve possuir uma câmera embarcada para a aquisição das imagens durante o voo.
- **Criar um algoritmo utilizando técnicas de visão computacional para processar as imagens do terreno:** Estudar, implementar e testar um algoritmo, utilizando visão computacional, capaz de reconhecer imagens do terreno capturadas pela câmera embarcada no VANT.
- **Estimar a posição do VANT através das imagens reconhecidas do terreno:** Este objetivo específico compreende estudar e desenvolver um algoritmo que possibilite estimar a posição do VANT, em voo, utilizando as imagens reconhecidas do terreno.
- **Desenvolver uma estação de solo:** Desenvolver uma estação de solo para processar as imagens e controlar o VANT. Através de um canal de comunicação com a aeronave, a estação receberá da aeronave as imagens em tempo real, irá processá-las utilizando visão computacional e, então, enviará os comandos de controle para a execução da trajetória.
- **Realizar a transmissão das imagens:** Realizar a transmissão, em tempo real, das imagens capturadas pela câmera embarcada para a estação de solo durante a realização do voo.
- **Ajustar os parâmetros do algoritmo de reconhecimento:** Refinar o ajuste de alguns parâmetros dos algoritmos de visão computacional para que se encontre a melhor eficiência, correção e confiabilidade no reconhecimento das imagens e estimativa de movimentação do VANT.
- **Realizar experimentos de voo utilizando navegação por imagens:** Com o sistema integrado e funcionando, este objetivo específico compreende a realização

de experimentos de voo, real e simulado, utilizando navegação por reconhecimento de imagens do terreno. Nestes voos, não será utilizado qualquer mecanismo de localização por satélite.

### 1.3 Contribuições

Uma das contribuições obtidas por este trabalho é o desenvolvimento de uma estação de solo capaz de realizar a navegação de VANTs baseado em reconhecimento de imagens do terreno utilizando visão computacional. Através desse sistema, é possível refazer trajetórias anteriormente realizadas por outros VANTs que adquiriram e armazenaram as imagens do terreno em um banco de dados. Como diferencial deste trabalho em relação a outros existentes, destaca-se que a grande maioria dos trabalhos pesquisados necessitam que marcos visuais artificiais sejam fisicamente colocados no terreno para a localização e orientação da aeronave. Alguns exemplos desses marcos artificiais comumente utilizados são: figuras geométricas com formas definidas (triângulos, círculos, etc), figuras com cores contrastantes em relação ao terreno (figura vermelha em um terreno predominantemente verde), espelhos refletores, *QR-Codes*, entre outros. O sistema desenvolvido neste trabalho utiliza como marcos (*waypoints*) as próprias imagens do terreno. Para isso, técnicas de visão computacional serão empregadas para reconhecer as imagens aéreas como *waypoints* de uma trajetória. Além disso, esses *waypoints* visuais podem ser salvos em um banco de dados e compartilhados entre diversas aeronaves componentes de uma frota.

### 1.4 Estrutura da dissertação

Este trabalho, com exceção desta introdução, possui a seguinte estrutura:

- **Capítulo 2 - Fundamentação Teórica:** Neste capítulo, são abordados alguns conceitos básicos para o entendimento deste trabalho. Tópicos sobre VANTs, sensores utilizados nas aeronaves, reconhecimento de imagens utilizando visão computacional, entre outros conteúdos são apresentados ao leitor.
- **Capítulo 3 - Revisão da Literatura:** Neste capítulo, é realizada uma revisão da literatura existente. São expostos de maneira resumida alguns trabalhos relacionados com o tema desta dissertação.
- **Capítulo 4 - Caracterização do Problema:** Neste capítulo, é discutido o problema de navegação de VANTs de maneira mais detalhada.
- **Capítulo 5 - Solução Proposta:** Neste capítulo, é descrita a solução proposta para se realizar a navegação utilizando o reconhecimento de imagens do terreno.

- **Capítulo 6 - Implementação da Solução:** Neste capítulo, é descrita a implementação da solução proposta utilizando uma plataforma robótica comercial.
- **Capítulo 7 - Experimentos e Resultados:** Neste capítulo, são descritos os experimentos realizados e analisados os resultados encontrados.
- **Capítulo 8 - Considerações Finais:** Neste capítulo, são expostas a conclusão e as sugestões de trabalhos futuros sobre o tema.

## 2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo, serão apresentados os principais referenciais teóricos sobre o tema abordado nesta dissertação. Estes referenciais compõem a base para o entendimento desta dissertação e do funcionamento da solução proposta.

### 2.1 Veículos aéreos não tripulados

Os VANTs são aeronaves que não necessitam de um piloto humano a bordo, podendo ser controlados por um computador embarcado ou à distância por um operador humano utilizando um enlace de dados para comunicação. Recentemente, essas aeronaves vêm ganhando muita atenção, principalmente por suas inúmeras possibilidades de aplicação, melhorias no poder de processamento embarcado, aumento da capacidade das baterias, entre outros avanços. Após destacada a importância dos VANTs, serão aqui tratados alguns aspectos sobre sua classificação de acordo com o mecanismo de sustentação utilizado, sobre uma categorização utilizando parâmetros como peso, autonomia, altitude máxima e distância do enlace de dados e, por último, aspectos sobre a dinâmica dos quadricópteros.

#### 2.1.1 Classificação por sustentação

Segundo [22], as aeronaves podem ser classificadas em quatro grandes grupos, de acordo com o seu mecanismo de sustentação. O primeiro grupo é composto pelos VANTs de asa fixa. Estes possuem como características a capacidade de voos de longo alcance, rápidas velocidades e consumo moderado de energia. A asa-delta, Figura 2, se enquadra nessa categoria. O segundo grupo é composto pelos de asa rotativa. Estes possuem como características a alta manobrabilidade e a possibilidade de pouso e decolagem verticais. Porém, como desvantagem, possuem alto consumo de energia e velocidade de deslocamento moderada. Os quadricópteros, Figura 3, estão incluídos nessa categoria. O terceiro grupo é composto pelos dirigíveis. Estes veículos possuem uma estrutura completa com densidade mais leve que o ar atmosférico. Isso faz com que a força de empuxo seja maior que o peso da aeronave, resultando na sua sustentação. Possuem como características a baixa velocidade de locomoção e baixo consumo de energia. A Figura 4 ilustra um dirigível. Por último, tem-se o quarto grupo composto pelas aeronaves de asa móvel. Possuem como características a grande manobrabilidade e alta eficiência energética. A Figura 5 ilustra um exemplo dessa aeronave.

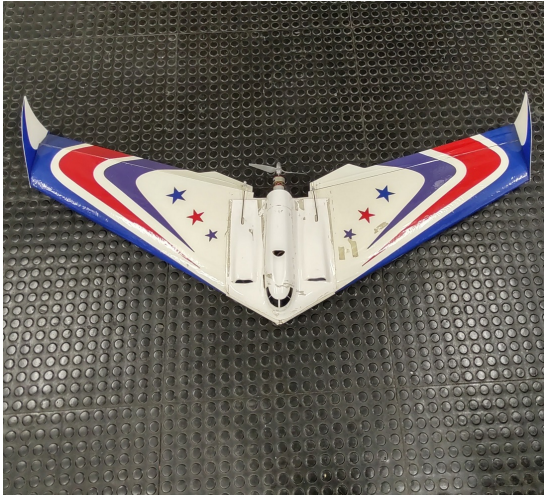


Figura 2 – Exemplo de VANT de asa fixa.



Figura 3 – Exemplo de VANT de asa rotativa.



Figura 4 – Exemplo de VANT dirigível.



Figura 5 – Exemplo de VANT de asa móvel. Fonte [1]

### 2.1.2 Classificação por dimensão

Os VANTs podem ainda ser classificados de acordo com algumas características físicas e operacionais da aeronave. A *European Association of Unmanned Vehicles Systems* (EUROUVS) classifica os VANTs de acordo com os seguintes critérios: peso máximo de decolagem, altitude máxima alcançada em voo, autonomia de voo e distância máxima do enlace de dados. Essa classificação pode ser utilizada para a escolha correta do VANT de acordo com a missão que se deseja realizar, pois cada categoria de aeronave possui características e aplicações diferentes. Conforme destacado em [13], os Micro e Mini VANTs são pequenas aeronaves que voam abaixo de 300 metros. São desenhadas para operar em ambientes urbanos, podendo ainda funcionar dentro de construções. Podem carregar microfones e pequenas câmeras para transmissão de vídeo em tempo real. Os VANTs

táticos são aeronaves mais pesadas, com altitude de voo variando entre 3000 a 8000 metros, e possuem aplicação primariamente militar. Já os VANTs estratégicos são as aeronaves de maiores pesos, alta resistência e longo alcance. São também capazes de carregar grandes quantidades de carga (*payload*). A Tabela 1 expressa a classificação EUROUVS.

VANT	Categoria	Peso Máximo (kg)	Altitude máxima (m)	Autonomia (horas)	Enlace de dados (km)
Micro/Mini	Micro	0,1	250	1	<10
	Mini	<30	150-300	<2	<10
Tático	Close Range (CR)	150	3.000	2-4	10-30
	Short Range (SR)	200	3.000	3-6	30-70
	Medium Range (MR)	150-500	3.000-5.000	6-10	70-200
	Long Range (LR)	-	5.000	6-13	200-500
	Endurance (EN)	500-1.500	5.000-8.000	12-24	>500
Estratégico	High Altitude, Long Endurance (HALE)	2.500-12.500	15.000-20.000	24-48	>2.000

Tabela 1 – Classificação dos VANTs segundo *European Association of Unmanned Vehicles Systems* (EUROUVS), conforme expresso em [13].

### 2.1.3 Dinâmica dos VANTs quadricópteros

Os quadricópteros possuem seis graus de liberdade, sendo três de rotação e três de translação. Os três ângulos de rotação definem a atitude da aeronave em determinado instante. A rotação sobre o eixo  $x$ , longitudinal à aeronave, é chamada de rolagem ou *roll*. A rotação sobre o eixo  $y$ , transversal a aeronave, é chamado de arfagem ou *pitch*. A rotação sobre o eixo  $z$ , perpendicular ao plano da aeronave, é conhecida como guinada ou *yaw*. Na Figura 6, estão ilustrados os eixos citados e seus respectivos movimentos de rotação.

Conforme destacado por [23], os movimentos da aeronave são resultados da configuração de empuxo, *thrust*, dos seus quatro motores ilustrada na Figura 7. A seta circular vermelha indica uma maior rotação por minuto (RPM) do motor, enquanto a seta circular azul indica uma menor RPM. Assim, para a aeronave realizar um movimento de subida perpendicular ao solo, deve-se aumentar o RPM dos seus quatro motores em igual proporção, como mostrado na Figura 7.e. Para o movimento de descida, deve-se diminuir igualmente o RPM dos quatro motores, como ilustrado na Figura 7.f. Os movimentos de translação para as laterais podem ser realizados aumentando-se o RPM dos motores do lado oposto ao que se deseja movimentar, como ilustrados nas Figuras 7.c e 7.d. Isso modificará a atitude da aeronave, gerando uma rolagem e, como consequência, um movimento lateral. De maneira semelhante, tem-se os movimentos no eixo longitudinal ilustrados nas Figuras 7.a e 7.b. Os

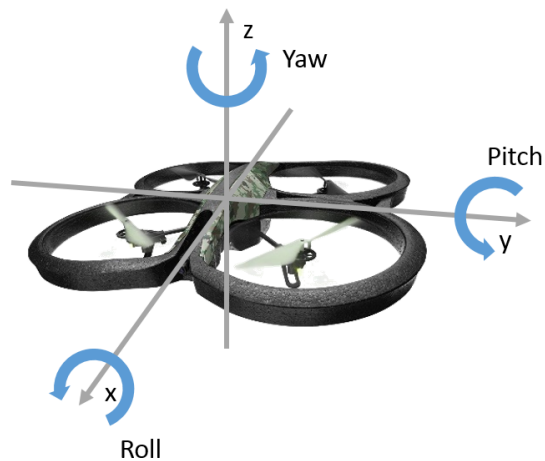


Figura 6 – Rotação do VANT nos três eixos (*roll*, *pitch* e *yaw*).

movimentos de rotação em torno do eixo perpendicular à aeronave (*yaw*), são resultantes da lei de conservação da quantidade de movimento angular. Assim, como expostos nas Figuras 7.g e 7.h, quando se aumenta o RPM dos motores opostos, ou seja, a quantidade de movimento angular em um sentido, a aeronave é rotacionada automaticamente no outro sentido com a finalidade de preservar essa propriedade física.

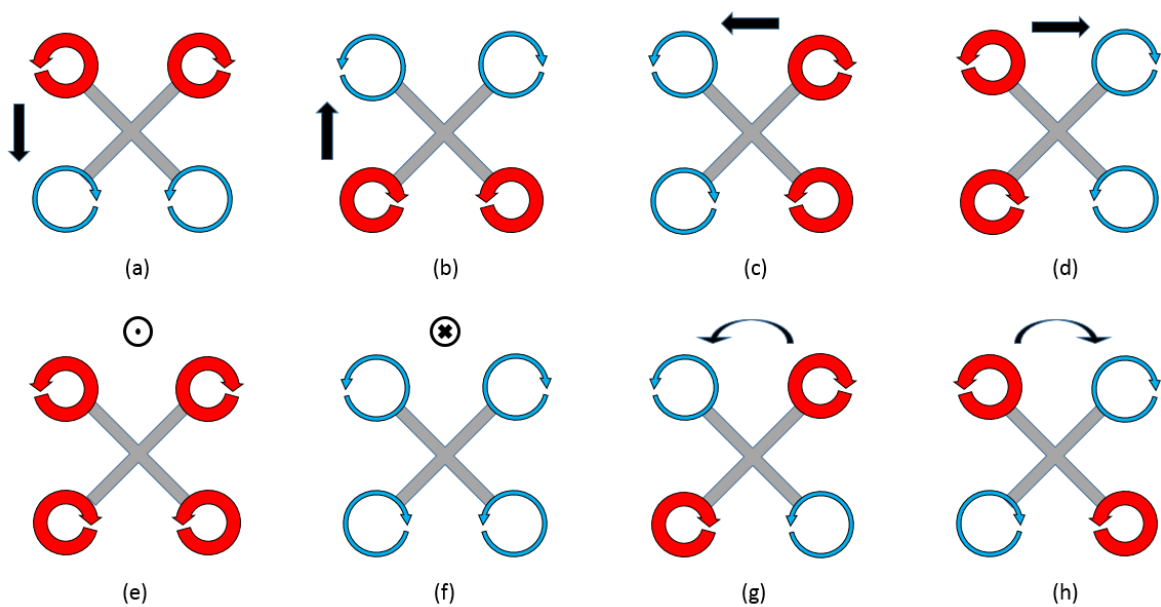


Figura 7 – Relação entre a configuração dos motores de um quadricóptero e os movimentos resultantes da aeronave.



## 2.2 Piloto Automático

O piloto automático é o software que efetua o controle da aeronave. Ele realiza a leitura dos sensores embarcados, faz os cálculos necessários para a estabilização e movimentação da aeronave e, então, opera sobre os mecanismos de controle do VANT. Esses mecanismos podem ser a atuação direta nos motores, como é o caso dos quadricópteros, ou nos atuadores, como é o caso dos aviões.

Um dos pilotos automáticos mais conhecidos e utilizados é o Ardupilot [2]. Ele é um software de código aberto e mantido por uma grande comunidade de desenvolvedores. Pode ser utilizado em diversos tipos de veículos, como em aeronaves de asa rotativa e fixa, em veículos terrestres e em submarinos. Esse piloto automático possui diversas características, dentre as quais se destacam:

- Ser compatível com várias controladoras de voo (*hardware*) existentes no mercado, como a Navio2 [24], Pixhawk [25] e a Kakute [26].
- Possui mecanismo de *failsafe*, onde o software monitora o estado do sistema e dispara o mecanismo de *Return-to-Home* em caso de perda de sinal, bateria baixa ou quando a aeronave sai de uma região de segurança previamente definida (*geofence*).
- Possui a capacidade de realizar missões autônomas, seguindo diversas coordenadas de GPS cadastradas (GPS *waypoints*).
- É muito flexível e estável, permitindo diversas configurações de aeronaves. No caso dos VANTs de asa rotativa, o software permite diversos *frames*, conforme ilustrado na Figura 8.

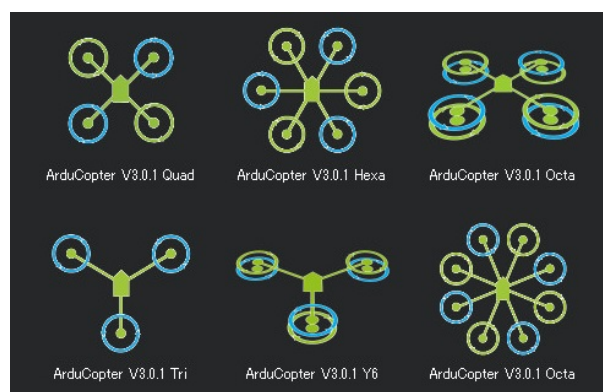


Figura 8 – Exemplos de *frames* para aeronaves de asa rotativa no Ardupilot. Figura adaptada de [2].

### 2.2.1 Modos de voo

O modo de voo informa ao piloto automático qual a forma de controle da aeronave, ou seja, expressa como deve ser o comportamento da plataforma robótica durante o voo. Cada modo de voo possui características específicas e adequadas para cada tipo de situação. Assim, o operador da aeronave deve escolher o modo mais adequado para cada circunstância. A mudança entre os modos de voo pode ocorrer através de um simples acionamento de uma chave no *joystick*, ou enviando comandos de controle, através do protocolo Mavlink [27], para o piloto automático. No caso dos quadricópteros, o Ardupilot implementa aproximadamente 20 modos de voo. Dentre esses, se destacam alguns principais descritos a seguir.

**AltHold:** O usuário controla a *roll* e a *pitch* da aeronave e o software faz as correções necessárias para manter a altitude constante.

**Auto:** O piloto automático executa uma trajetória pré configurada seguindo coordenadas de GPS cadastradas.

**Loiter:** O piloto automático controla a altitude e a posição quando as alavancas de controle (*sticks*) estão em posição neutra. Este modo utiliza o GPS para manter a posição da aeronave fixa.

**Return-to-Launch (RTL):** Também conhecido como *Return-to-Home*. Esse modo faz a aeronave retornar para o ponto de decolagem, podendo ou não incluir o pouso da plataforma robótica. Necessita de GPS para seu funcionamento.

**Stabilize:** O usuário controla manualmente a atitude da aeronave. Quando os controles estão em posição neutra, o piloto automático estabiliza a aeronave. Neste modo, quando há forças externas sobre a aeronave, o usuário deve atuar constantemente nos controles do VANT.

**FlowHold:** O piloto automático mantém a posição da aeronave fixa utilizando fluxo ótico.

**Land:** Reduz incrementalmente a altitude da aeronave até ocorrer o pouso (*landing*).

## 2.3 Sensores

Os sensores são elementos essenciais no controle do VANT. Eles fornecem as informações necessárias para que o piloto automático possa realizar o controle da aeronave. Através das leituras desses dados, o piloto automático realiza os cálculos necessários para estabilizar ou mover a aeronave. Esse controle ocorre através de um *loop* fechado, onde os sensores são os componentes indispensáveis para a leitura do estado da aeronave. Nos VANTs comerciais de baixo custo, muitos dos sensores empregados são do tipo *Micro-Electro-Mechanical Systems* (MEMS). Estes são sensores com sistemas micro eletro-mecânicos encapsulados em um circuito integrado (CI). Dentro do MEMS, há partes mecânicas e eletrônicas que, em conjunto, são capazes de fazer o sensoriamento de acelerações, rotações e altitude da aeronave. Há também sensores ópticos, sonoros e elétricos. Abaixo, serão descritos brevemente alguns sensores utilizados em VANTs. Os três primeiros são MEMS.

### Acelerômetros

O acelerômetro mede a aceleração que atua sobre um corpo e pode possuir um ou mais eixos de leitura. No caso dos sensores de três eixos, estes normalmente estão posicionados de maneira perpendicular entre si. Desta maneira, é possível ler a aceleração em todas as direções de deslocamento. Nos MEMS, o princípio básico de funcionamento do acelerômetro é baseado em um sistema massa-mola e está ilustrado simplificadaamente na Figura 9. Dentro do CI, há uma parte fixa, destacado na Figura em azul escuro, e uma parte móvel, destacado em azul claro. A parte móvel está apoiada sobre "molas" ou sobre materiais elásticos. Quando o sensor sofre uma aceleração ao longo do eixo de leitura, a parte fixa se move junto com o sensor. Entretanto, a parte móvel, devido à inércia, sente uma força ( $F$ ) contrária ao movimento. Isso resulta no deslocamento da parte móvel dentro do sensor. Esse movimento, por sua vez, causa a mudança na capacitância ( $C$ ) entre as placas das partes fixas e móveis. Através da medição na variação da capacitância, pode-se realizar o cálculo da aceleração sofrida pelo corpo.

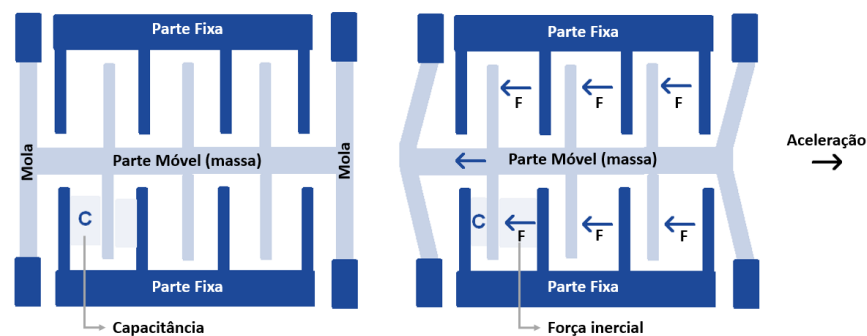


Figura 9 – Princípio básico de funcionamento de um acelerômetro MEMS. A parte externa é fixa e a parte interna é móvel.

## Giroscópios

O giroscópio é um sensor que mede a velocidade angular de um corpo. Semelhantemente aos acelerômetros, os giroscópios possuem um ou mais eixos de leitura. No caso de possuírem três eixos, estes normalmente também estão posicionados de maneira perpendicular entre si. Isso possibilita a leitura em todas as direções. Nas aeronaves, este sensor é utilizado para medir rotações. Um dos princípios de funcionamento dos giroscópios MEMS é com base no efeito Coriolis. Neste caso, um corpo de prova é posto para vibrar sobre o eixo  $x$ . Quando a estrutura vibrante sofre uma rotação sobre o eixo  $z$ , surge a aceleração de Coriolis que, então, induz uma vibração no eixo  $y$ . Através de transdutores instalados no eixo  $y$ , pode-se realizar leituras e calcular a rotação sofrida pelo sensor. A Figura 10 ilustra simplificada o funcionamento do giroscópio. Já a Figura 11 mostra um giroscópio MEMS comercial.

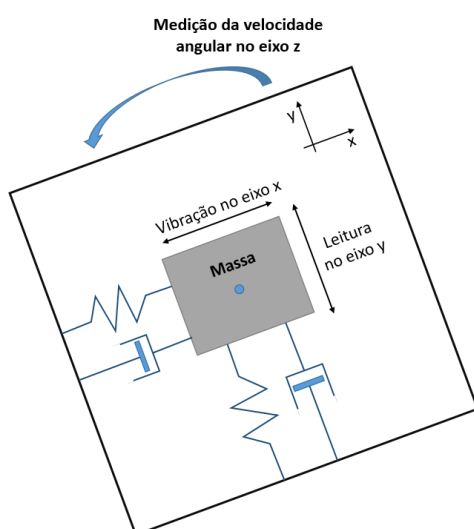


Figura 10 – Princípio de funcionamento de um giroscópio MEMS por efeito Coriolis.

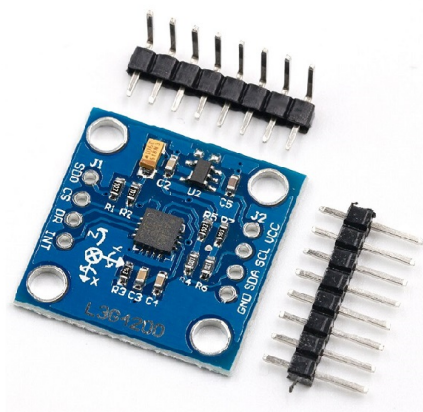


Figura 11 – Exemplo de um sensor giroscópio comercial utilizando em VANTs.

## Barômetros

O barômetro é um sensor utilizado para medir a variação da pressão atmosférica. Nos VANTs, ele é utilizado para estimar a altitude da aeronave. O princípio mais comum de funcionamento deste sensor é baseado no deslocamento de um diafragma. Neste sensor MEMS, há uma câmara fechada com uma determinada pressão interna. Lacrando-se essa câmara, há uma película diafragma que possui a capacidade de ser flexionada de acordo com a diferença de pressão entre as suas faces. Quando a pressão externa ao sensor está igual à pressão interna dentro da câmara, o diafragma fica reto, sem flexão. Quando a pressão externa está maior que a interna, o diafragma é flexionado para dentro. Na periferia desse diafragma, são instalados transdutores para ler a força aplicada sobre a película.

Através da leitura da força aplicada, é possível estimar a variação da altitude sofrida pela aeronave. A Figura 12 ilustra simplificada o funcionamento do barômetro. Já a Figura 13 mostra um barômetro MEMS comercial.

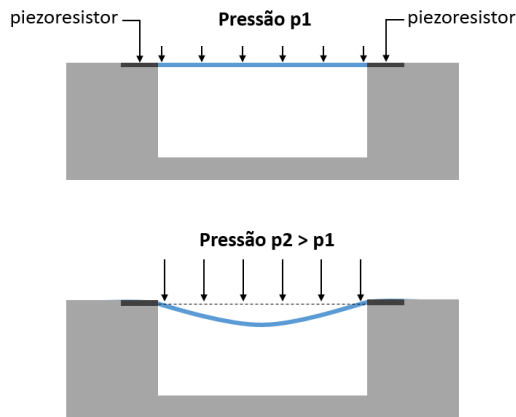


Figura 12 – Princípio básico de funcionamento de um barômetro MEMS.

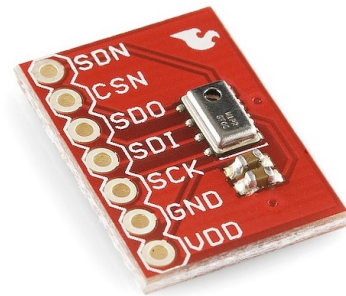


Figura 13 – Exemplo de um sensor barômetro MEMS comercial utilizando em VANTs.

### Magnetômetros

O magnetômetro é um sensor utilizado para medir a direção e intensidade de um campo magnético. Igualmente aos acelerômetros e giroscópios, os magnetômetros possuem um ou mais eixos de leitura. No caso de possuírem três eixos, estes normalmente também estão posicionados de maneira perpendicular entre si. Isso possibilita a leitura do campo magnético em todas as direções. Nos VANTs, este sensor é utilizado para medir o campo magnético terrestre com a finalidade de fazer o controle da orientação (*yaw*) da aeronave. Há várias técnicas para medir o campo magnético, e uma das mais conhecidas é o efeito Hall. Nesse efeito, há o surgimento de uma diferença de potencial (DDP) perpendicularmente a um condutor, quando este se encontra inserido em um campo magnético e percorrido por uma corrente. Através da leitura da DDP gerada, pode-se calcular a intensidade do campo magnético que corta o condutor. A Figura 14 ilustra simplificada o funcionamento do magnetômetro. Já a Figura 15 mostra um magnetômetro comercial.

### Ultrassônico

O sensor ultrassônico é utilizado para medir pequenas distâncias. Seu princípio de funcionamento é baseado no cálculo do tempo necessário para um pulso sonoro emitido pelo sensor refletir em um obstáculo e voltar novamente até o sensor. Através do intervalo de tempo encontrado e conhecendo-se a velocidade de propagação do som no ar, pode-se estimar a distância até o objeto pretendido. No entanto, este sensor possui algumas limitações. Uma delas é o alcance que, dependendo do hardware utilizado, vai até poucos metros. Outra limitação é a precisão, pois mudanças na temperatura e na pressão

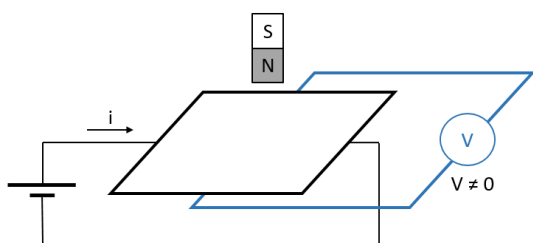


Figura 14 – Princípio de funcionamento de um magnetômetro por efeito Hall.

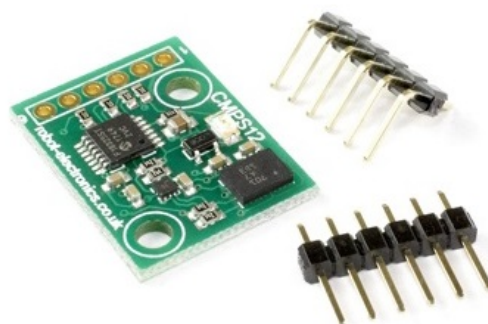


Figura 15 – Exemplo de um sensor magnetômetro comercial utilizado em VANTs.

atmosférica modificam a componente velocidade do som no ar, necessária no cálculo da distância. Nos VANTs, o sensor ultrassônico é utilizado para medir pequenas altitudes, principalmente quando a aeronave está realizando o pouso (*landing*) ou a decolagem (*takeoff*). Sua utilização possibilita uma aproximação mais suave ao solo. A Figura 16 ilustra simplificada o funcionamento do sensor ultrassônico. Já a Figura 17 mostra um sensor ultrassônico comercial.

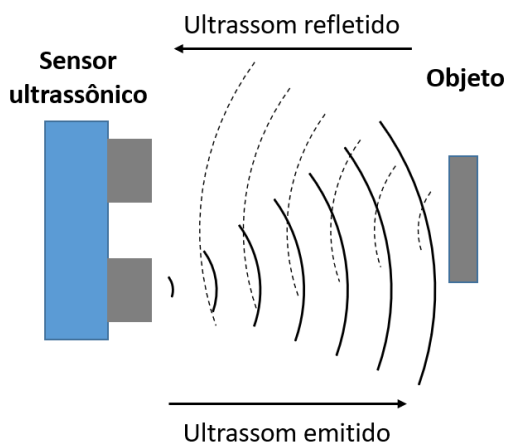


Figura 16 – Funcionamento de um sensor ultrassônico na medição de distâncias.



Figura 17 – Exemplo de um sensor ultrassônico comercial utilizado em VANTs.

### ***Global Positioning System (GPS)***

O GPS é um sistema de localização onde um dispositivo receptor é capaz de calcular as suas coordenadas a partir de sinais enviados por satélites. Este sistema é de propriedade dos Estados Unidos da América (EUA) e mantido pela Força Aérea daquele país. Segundo o site oficial do sistema [3], para que o GPS funcione, é necessário haver uma constelação mínima de 24 satélites, distribuídos em 6 órbitas de altitude média (aproximadamente 20.200 km de altura). Cada órbita possui o número mínimo de quatro satélites e cada um circula o planeta Terra duas vezes por dia. A Figura 18, retirada do site oficial, ilustra essa constelação. Com o mínimo de 24 satélites operacionais, é possível receber o sinal de pelo menos quatro satélites ao mesmo tempo, em qualquer ponto da Terra e em qualquer momento desejado. Para o funcionamento do sistema, os satélites da constelação possuem relógios atômicos precisamente sincronizados com uma estação de controle em terra. Além disso, também são conhecidas as posições de cada satélite com grande grau de acerto. Essas informações de posição e tempo, entre outras, são transmitidas continuamente pelos satélites para os receptores GPS. Recebendo os sinais dos satélites, um receptor pode calcular a diferença de tempo necessário para o sinal sair do satélite e chegar até o receptor. Com esse intervalo de tempo calculado e sabendo-se a velocidade de propagação da luz no ar atmosférico, pode-se encontrar a distância até cada um dos satélites cujo sinal foi recebido. A partir de quatro distâncias calculadas, faz-se uma triangulação e estima-se a posição do receptor. Somando-se a isso, os dispositivos GPS também podem fornecer informações de velocidade, aceleração e direção de deslocamento através do cálculo sucessivo de posições. A Figura 19 ilustra simplificada a recepção do sinal de quatro satélites, onde  $x_i$ ,  $y_i$  e  $z_i$  representam as coordenadas conhecidas de cada satélite e  $t_i$  o intervalo de tempo necessário para o sinal chegar até o receptor. A Figura 20 ilustra um receptor GPS comercial utilizando em VANTs.

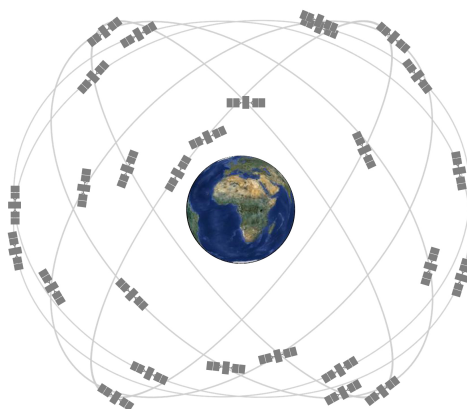


Figura 18 – Constelação de satélites do sistema GPS. O conjunto mínimo é formado por 24 satélites divididos em seis órbitas. Fonte da imagem [3].

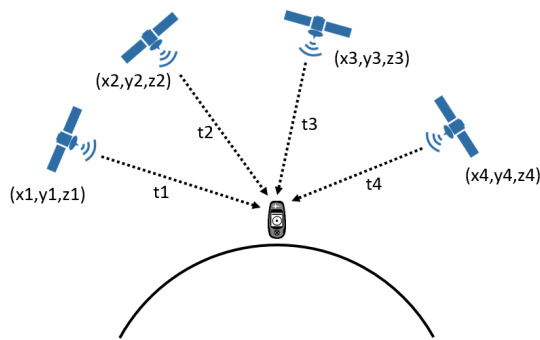


Figura 19 – Transmissão dos parâmetros posição e tempo dos 4 satélites para o receptor GPS.



Figura 20 – Exemplo de um sensor GPS comercial utilizado em VANTs.

### Câmera digital

A câmera digital é um sensor óptico que capta a luz do ambiente e constrói uma imagem representada por um mapa de bits. Nos últimos anos, este sensor tem sido cada vez mais utilizado nas plataformas robóticas devido ao seu baixo preço de aquisição e aos grandes avanços das técnicas de visão computacional. Um modelo simplificado do funcionamento de uma câmera é baseado na câmara escura ou, em inglês, *pinhole model*. Neste modelo, uma câmara escura é feita de material opaco e, em uma das faces, faz-se um pequeno orifício para a entrada de luz. Os raios luminosos que saem de um objeto, passam pelo orifício da câmara escura e iluminam sua face oposta. Nesta face, há um sensor luminoso dividido em diversas unidades de sensoriamento denominadas pixels. A imagem do objeto visualizado pela câmera se projeta sobre este sensor que lê cada pixel e os transformam em um mapa de bits. A Figura 21, adaptada de [4], ilustra simplificada o funcionamento da câmara escura. A Figura 22 ilustra uma câmera comercial utilizada em VANTs.

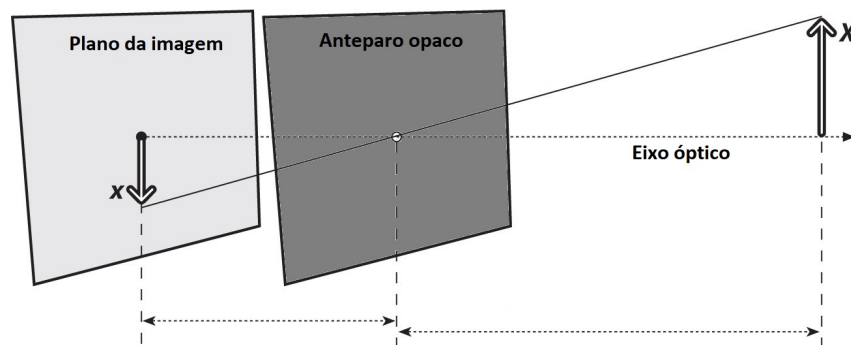


Figura 21 – Princípio básico de funcionamento de uma câmera baseado no modelo da câmara escura. Imagem adaptada de [4].

As câmeras digitais possuem algumas características importantes. Entre elas, se



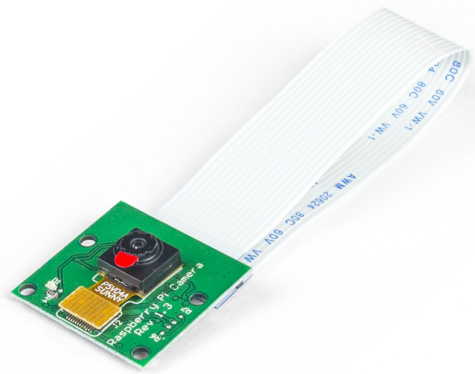


Figura 22 – Exemplo de câmera comercial utilizada em VANTs.

destacam:

- **Resolução da imagem:** Representa a quantidade de pixels que a câmera possui na captura da imagem. A resolução é expressa na quantidade de pixels na horizontal e na vertical. Alguns exemplos de resolução são: 640x480 (VGA), 1280x720 (HD) e 1920x1080 (Full-HD).
- **Quadros por segundo:** Representa a quantidade de quadros que a câmera é capaz de capturar em um segundo (*frames per second* - fps). Por exemplo, se a câmera possui 30 fps, ela é capaz de capturar 30 quadros em um segundo.
- **Espectro visível:** Representa a faixa de frequência da luz que excita o sensor da câmera. Um exemplo são as câmeras RGB (*Red, Green and Blue*). Elas trabalham no espectro visível e captam apenas as cores vermelho, verde e azul. Através da composição dessas três cores, é possível formar todas as outras. Uma segunda possibilidade são as câmeras que trabalham na faixa do infravermelho como, por exemplo, as câmeras térmicas e de visão noturna.
- **Ângulo de abertura:** Representa a abertura angular máxima na qual um objeto, dentro desse espaço, pode ser visualizado pela câmera. A câmera possui dois ângulos de abertura, um na vertical e outro na horizontal. Tais ângulos podem variar de acordo com a resolução utilizada.

## 2.4 Controladores automáticos

Segundo [28], um controlador automático é um mecanismo que compara o valor real de saída de um sistema com um valor de entrada desejado (denominado *setpoint*). Após isso, determina o desvio entre esses valores, produzindo um sinal de controle necessário que conduzirá o valor de saída ao valor de entrada desejado. Tal ação é chamada tecnicamente como *ação de controle*. Quando empregado em um sistema, o controlador atua sobre ele em um circuito de malha fechada, a fim de regular sua saída. Há diversos tipos de controladores automáticos, dentre eles se destacam: o controlador On-Off (ou de duas posições) e o controlador PID (Proporcional, Integral e Derivativo). Em seguida, serão descritos brevemente o funcionamento de cada um deles.

### 2.4.1 Controlador On-Off

O controlador On-Off é bem simples. Ele atua sobre um sistema com apenas duas posições possíveis, ligado ou desligado. Assim, o projetista do controlador regula dois valores de saída ( $U_1$  e  $U_2$ ), de tal maneira que: para saídas com valores acima de  $U_2$ , o controlador desliga; para saídas com valores abaixo de  $U_1$ , o controlador liga; ou vice-versa, ligando e desligando respectivamente. A Figura 23 representa o gráfico de saída de um sistema controlado por um controlador On-Off.

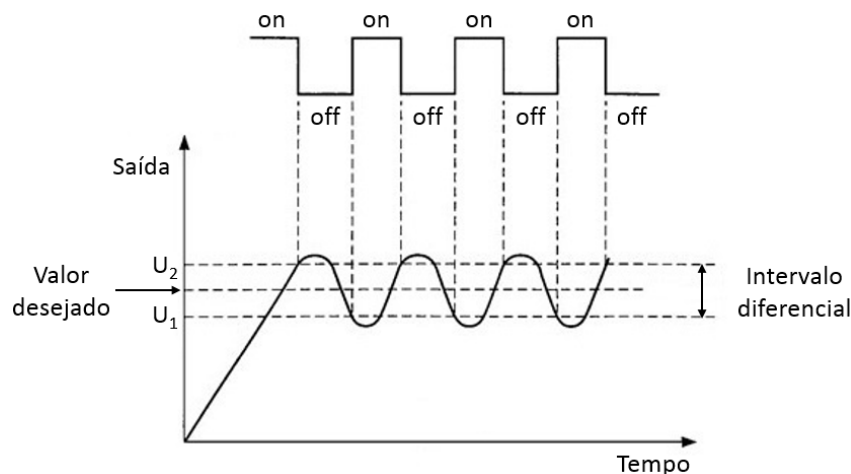


Figura 23 – Exemplo de funcionamento de um controlador On-Off.

Neste tipo de controlador, o intervalo onde o sinal de erro pode variar antes de ocorrer a comutação é denominado *intervalo diferencial*. Para valores dentro desse intervalo, entre  $U_1$  e  $U_2$ , o controlador não atua sobre o sistema controlado. Destaca-se que a diminuição do intervalo diferencial melhora a precisão do sistema, porém aumenta o número de comutações On-Off necessárias. Por outro lado, o aumento do intervalo diferencial diminui o número de comutações, mas piora a precisão do sistema.

## 2.4.2 Controlador PID

Mais complexo do que o controlador On-Off, o controlador PID conjuga as ações proporcional (P), integral (I) e derivativa (D) visando diminuir o erro resultante do sistema controlado. Segundo [28], esta técnica é amplamente utilizada em sistemas industriais onde se deseja controlar uma variável do sistema. Cada uma das três ações do controlador possui uma função diferente no controle. Tais funções estão descritas abaixo:

- **Ação proporcional:** Produz uma saída proporcional ao erro. Ou seja, quanto maior o erro, maior será o sinal de saída do controlador.
- **Ação integral:** Produz uma saída proporcional ao erro e à sua duração no tempo. Ou seja, erros pequenos que ocorrem durante muito tempo necessitam de uma correção mais intensa. Esta ação auxilia o sistema a chegar no seu valor desejado (*setpoint*), levando o erro próximo a zero no estado de regime permanente.
- **Ação derivativa:** Produz uma saída proporcional à velocidade de variação do erro. Ou seja, se o erro variar muito rápido, o controlador aumentará sua ação de controle.

A Figura 24 ilustra a planta de controle de um processo:  $r(t)$  representa o valor desejado de saída do sistema (ou *setpoint*);  $y(t)$  representa a saída do sistema;  $e(t)$  representa o erro entre o valor desejado e a efetiva saída do sistema; e  $u(t)$  representa a saída do controlador PID. A Equação 2.1 representa matematicamente o controlador PID, onde  $K_p$  é o ganho proporcional,  $K_i$  é o ganho integral e  $K_d$  é o ganho derivativo.

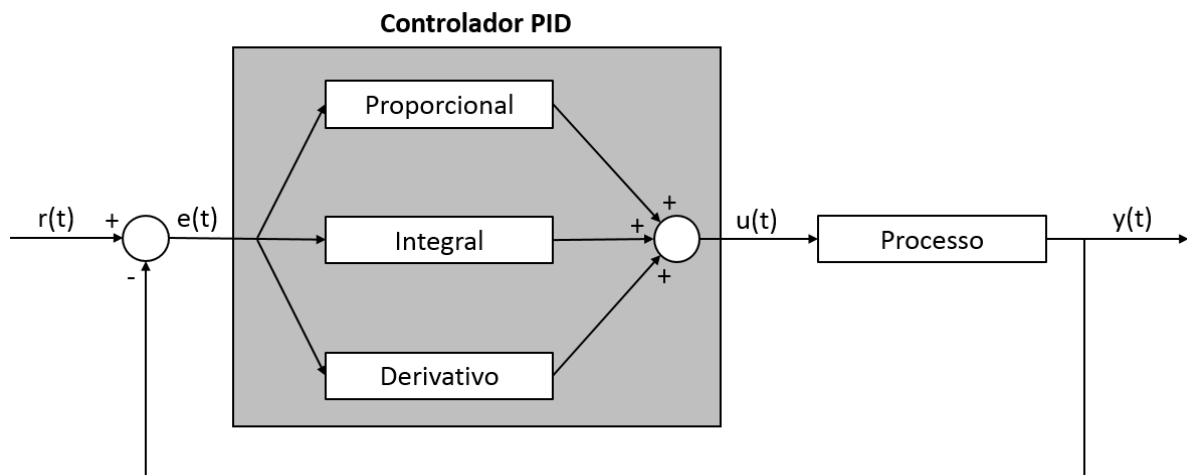


Figura 24 – Malha fechada de controle utilizando um controlador PID.

$$\mathbf{u}(t) = K_p \times e(t) + K_i \times \int_0^t e(\tau) d\tau + K_d \times \frac{de(t)}{dt} \quad (2.1)$$

## 2.5 Biblioteca OpenCV

O OpenCV (*Open Source Computer Vision Library*) é uma biblioteca de código aberto para aplicação em problemas de processamento de imagens, visão computacional e aprendizado de máquina. É desenvolvida nativamente em C/C++, porém também possui interface com as linguagens Python, Java e MATLAB. Possui suporte a *multi-threads* e, recentemente, a processamento utilizando GPU (*Graphics Processing Unit*). Esse suporte adicional forneceu à biblioteca uma melhor capacidade para solucionar problemas em tempo real e que necessitam de intenso poder computacional. Além disso, o OpenCV pode ser executado nos principais sistemas operacionais do mercado (Windows, Linux, Android e Mac OS), conforme exposto no site da equipe de desenvolvimento [29].

Esta biblioteca é relativamente grande, possuindo mais de 2500 algoritmos para os mais diversos fins, como para detectar e reconhecer faces humanas; identificar e rastrear (*tracking*) objetos em movimento; criar nuvem 3D de pontos a partir de múltiplas imagens ou câmeras estéreo; unir várias imagens em uma única; redimensionar, cortar e comparar imagens; entre outras finalidades. Por motivo de especificidade, neste trabalho, somente serão abordados as funcionalidades e pontos da biblioteca que possuem relação com o tema exposto. As funcionalidades de interesse deste trabalho são as relativas à detecção e ao reconhecimento de imagens. Tal técnica é conhecida como *image matching*.

Conforme descrito em [4], o entendimento dos conceitos *keypoints* (pontos de interesse, em Português) e *descriptors* (descritores, em Português) são de vital importância para se compreender como ocorre a detecção e o reconhecimento em imagens ou vídeos. Vários algoritmos de visão computacional utilizam esses dois conceitos-chave, os quais serão descritos a seguir.

### 2.5.1 *Keypoints* e *Descriptors*

De acordo com a referência citada, *keypoint* ou ponto de interesse é, em um alto nível de abstração, uma pequena porção da imagem que, por alguma razão, é usualmente distinta das demais e que se acredita ser possível localizá-la em outra imagem obtida da mesma cena ou objeto. Ou seja, o *keypoint* é uma pequena região da imagem com informações distintas quando comparada a outras áreas da mesma imagem. Normalmente, são áreas da imagem ricas em informação com bordas e textura. A título ilustrativo, a Figura 25 mostra os *keypoints*, em azul, extraídos de uma imagem aérea obtida por VANT.

O *descriptor* é uma construção matemática, usualmente representada por um vetor de pontos flutuantes, que descreve e especifica um *keypoint*. Pode ser utilizado para determinar se dois *keypoints* extraídos de imagens diferentes são iguais ou parecidos. Quanto mais próximos dois *descriptors* são, mais semelhantes são seus *keypoints* associados.

A utilização conjunta dos *keypoints* e *descriptors* permite criar uma assinatura da

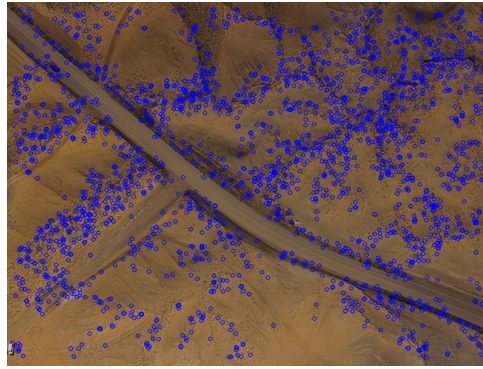


Figura 25 – Exemplos de keypoints, destacados em azul, de uma imagem aérea obtida por VANT.

imagem analisada. Ou seja, cria, a partir da imagem fornecida, uma sequência finita de números que a representa. Tal sequência pode possuir tamanho final menor que a imagem de entrada e ser utilizada para comparar similaridades entre imagens ou a detecção de pedaços de uma em outra. Tal técnica é conhecida como *image matching*.

O OpenCV possui vários métodos para manipular diversos tipos de *keypoints* e *descriptors*. Os algoritmos ou métodos mais conhecidos são: SIFT (*Scale Invariant Feature Transform*), SURF (*Speeded-Up Robust Features*), ORB (*Oriented FAST and Rotated BRIEF*), entre outros [30]. Cada um desses algoritmos possui características peculiares de desempenho, confiabilidade e robustez à rotação, orientação, escala e iluminação da imagem observada. Neste trabalho, utilizou-se satisfatoriamente o algoritmo SURF, apresentando bons resultados. Na Seção 6.4, será apresentada uma justificativa da utilização deste algoritmo, comparando-o com os outros citados.

## 2.5.2 Homografia

Segundo [4], no contexto de visão computacional, a homografia planar é uma função que realiza o mapeamento de um plano em outro plano. Isto é, mapeia os pontos de uma superfície plana em pontos de outra superfície também plana. Como exemplo desse processo, tem-se o mecanismo de geração da imagem de um objeto plano pela câmera no modelo *pinhole*, como descrito na Seção 2.3. Neste caso, o objeto plano pode ser, por exemplo, um tabuleiro de xadrez. A Figura 26 mostra um exemplo de homografia, onde o ponto  $Q$  é mapeado no ponto  $q$ .

Ainda segundo a referência, é possível demonstrar que a homografia pode ser representada por uma multiplicação de matrizes. A Equação 2.2 expressa essa multiplicação. A matriz  $H$  representa a transformação homográfica e possui dimensão  $3 \times 3$ .  $Q$  e  $q$  representam pontos em planos diferentes.

$$q = H \cdot Q \quad (2.2)$$

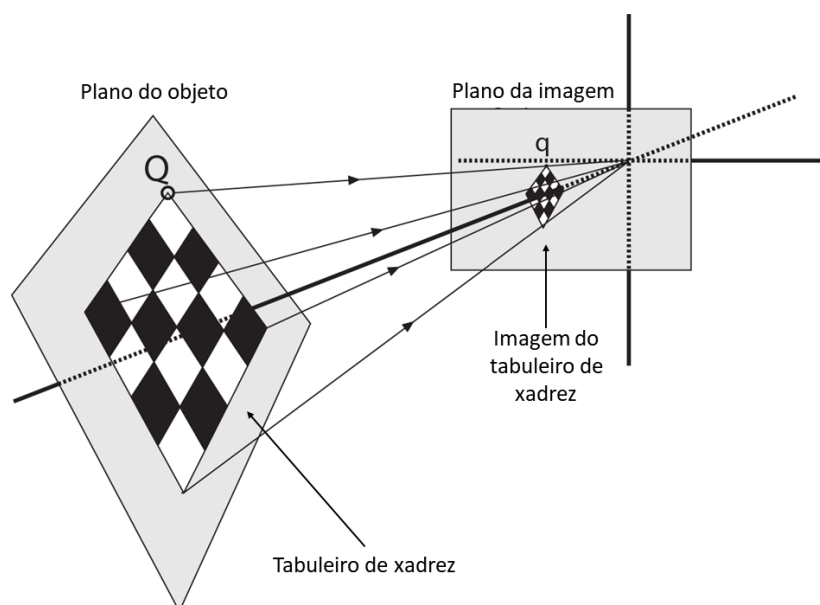


Figura 26 – Ilustração da homografia. Mapeamento do ponto  $Q$  de uma superfície plana no ponto  $q$  de outra superfície também plana. Imagem retirada e adaptada de [4].

Por fim, a homografia é bastante utilizada em visão computacional. Através dela, pode-se realizar projeções e encontrar os ângulos de rotação entre imagens de dois planos. Para facilitar a resolução de problemas envolvendo visão computacional, o OpenCV possui o método *findHomography()* para calcular a matriz  $H$  homográfica.

### 3 REVISÃO DA LITERATURA

Durante a elaboração deste trabalho, foi realizada uma pesquisa bibliográfica sobre a utilização de visão computacional em VANTs. Neste capítulo, será feito um resumo dos principais artigos relacionados a esse estudo.

No trabalho [31], foi desenvolvido um quadricóptero que empregava visão computacional para seguir uma bola amarela durante o seu voo. Este veículo utilizava um Raspberry Pi para executar o código de reconhecimento através da biblioteca OpenCV e um Arduino CRIUS AIO Pro v2 para a leitura dos sensores embarcados. Embora no referido artigo os autores afirmem ser possível controlar um VANT através do emprego de visão computacional, não ficou claro durante o texto que eles alcançaram este objetivo. Posteriormente, em outra publicação desses mesmos autores [5], eles utilizaram imagens denominadas *templates* para enviar comandos visuais ao VANT como, por exemplo, levantar voo, seguir para frente ou pousar. Um exemplo desses comandos pode ser visto na Figura 27, localizada abaixo e retirada do referido artigo.



Figura 27 – Exemplo de comando visual enviado ao VANT para execução e controle do voo. Figura retirada do artigo [5].

No trabalho [6], os autores desenvolveram um algoritmo para estabilização e localização da aeronave em ambientes *indoor*, sem sinal de GPS. Para tanto, utilizaram duas câmeras monoculares embarcadas, uma orientada verticalmente e a outra horizontalmente. A aeronave utilizada foi o AR.Drone, um equipamento comercial de baixo custo. As imagens capturadas, com resoluções de 320x240 (câmera horizontal) e 176x144 (câmera vertical), são transmitidas via WiFi a um computador externo, o qual realiza todo o processamento da informação e, então, envia os comandos de controle para a aeronave. Na Figura 28, tem-se a demonstração da aeronave em voo estabilizado.

No trabalho [7], os autores construíram um algoritmo para estimar e controlar o ângulo de *heading* da aeronave e sua velocidade vertical, ambos utilizando visão computacional.

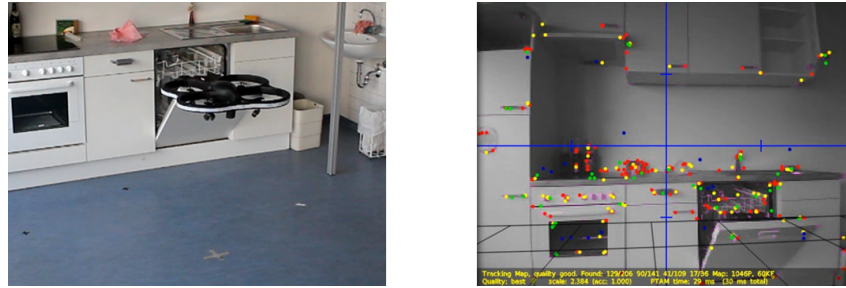


Figura 28 – Demonstração do VANT em voo estabilizado utilizando visão computacional. Figura retirada de [6]

cional. Para isso, embarcaram uma câmera monocular em um micro VANT e utilizaram o método SURF (*Speeded-Up Robust Features*) para o cálculo dos *keypoints* e seus respectivos *descriptors* nas imagens obtidas. Como utilizaram um micro VANT com baixa capacidade de payload e baixo poder de processamento embarcado, empregaram uma pequena placa FPGA (*Field Programmable Gate Array*) para realizar os cálculos relativos ao algoritmo SURF, já que este representa a parte mais custosa computacionalmente no cálculo da trajetória [32]. Conforme relatam os autores, eles conseguiram realizar o processamento a uma taxa de 10 imagens (1024x768 pixels) por segundo. A Figura 29, retirada do referido artigo, mostra o processamento de uma imagem capturada. Por fim, eles sugeriram a possibilidade de realizar uma navegação completa por imagem utilizando mecanismos de construção de mapas mais sofisticados.

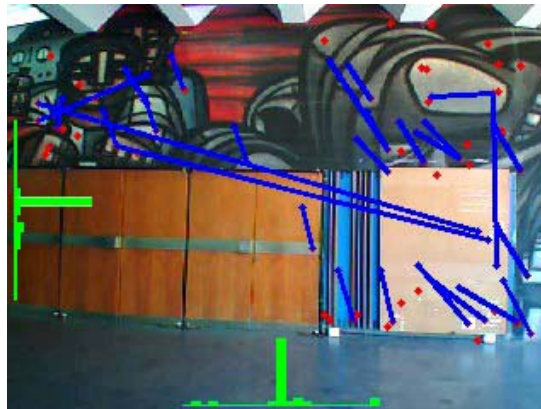


Figura 29 – Rastreamento de *features* para estimar o ângulo de *heading* e a velocidade vertical da aeronave. Figura retirada do artigo [7].

No trabalho [8], os autores desenvolveram um algoritmo capaz de controlar um VANT para seguir uma trajetória linear com base em linhas extraídas da imagem capturada pela câmera embarcada. O algoritmo de visão computacional extrai as principais formas lineares constantes na imagem, calcula uma média da posição e orientação delas e, então, envia os comandos para o piloto automático do VANT. Segundos os autores, a estratégia implementada é capaz de seguir margens de rios, calçadas ou outras formas lineares



existentes no ambiente. A Figura 30, adaptada de [8], mostra, à esquerda, um exemplo do cálculo da linha média a ser seguida e, à direita, o VANT em voo seguindo as margens de um córrego.

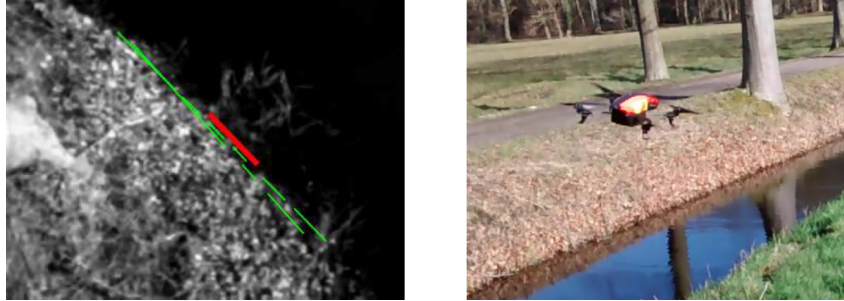


Figura 30 – VANT capaz de seguir formas lineares extraídas do ambiente. Figura adaptada do artigo [8].

No trabalho [33], os autores destacam que a navegação baseada em visão provou ser uma promissora fonte de navegação devido ao rápido desenvolvimento da visão computacional nos últimos anos. Além disso, os sensores visuais, quando comparados a outros sensores como o GPS e o ultrassônico, fornecem uma grande quantidade de informação do ambiente como cor, textura e profundidade que podem ser explorados pelos algoritmos utilizados. Ainda neste trabalho, os autores destacam que o método de rastreamento (*tracking*) de *keypoints* nas imagens tem se tornado um robusto meio de localização e mapeamento, pois é capaz de rastrear características (*features*) invariantes de elementos nas imagens em movimento. Essa extração das *features* nas imagens é normalmente invariante a rotação, perspectiva, bem como resistente a ruídos. Através disso, consegue-se determinar o movimento destes elementos detectando o deslocamento relativo dos *keypoints* nas sequências de imagens [34]. Destacam como desvantagens desse método que ele possui baixa performance em ambientes sem ou com pouca textura e que a ofuscação das imagens causadas pelo rápido movimento ou rotação do veículo durante o voo pode causar falhas na localização. Finalizando o artigo, os autores destacam que a fusão de dados de múltiplos sensores pode prover grande melhora na performance da navegação [35] e que a utilização conjunta de sensores inerciais (*Inertial Measurement Unit* - IMU) com a visão computacional tem ganhado muita atenção atualmente [36].

No trabalho [9], os autores desenvolveram um sistema capaz de pousar um quadricóptero sobre uma plataforma em movimento. Para isso, construíram um VANT com uma placa de processamento Jetson TK1 embarcada e uma câmera com lente *olho de peixe* para a aquisição das imagens. Na plataforma móvel de pouso, foi colocado um marcador quadrado vermelho para sua identificação. Assim, é extraída a posição do marcador em relação à aeronave utilizando técnicas de segmentação de cor disponíveis na biblioteca OpenCV. Após realizada a localização do marcador vermelho, comandos de deslocamento são enviados para a aeronave com a finalidade de realizar o pouso na plataforma. A Figura

31, retirada de [9], ilustra o funcionamento básico do sistema implementado.

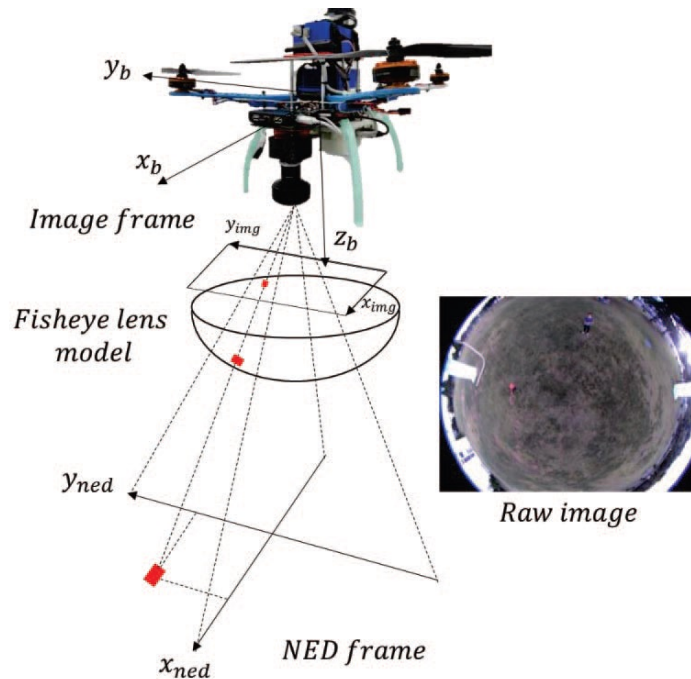


Figura 31 – Quadricóptero capaz de realizar o pouso em uma plataforma móvel através da identificação de um marcador vermelho. Figura retirada de [9]

Para compilar as características de cada trabalho aqui relacionado, foram plotadas duas tabelas comparativas. A Tabela 2 resume as tarefas, de alto nível, desenvolvidas por cada trabalho. Já a Tabela 3 sintetiza algumas das características, de nível mais baixo, descritas por eles. Por fim, após uma análise da pesquisa bibliográfica realizada, não foram encontrados trabalhos sobre execução autônoma de trajetória por *waypoints* visuais obtidos através de imagens capturadas do terreno. Assim, acredita-se ser este um dos diferenciais deste mestrado.

Tabela 2 – Compilação das tarefas desenvolvidas por cada trabalho relacionado.

Tarefa desenvolvida	Trabalhos						
	[5]	[6]	[7]	[8]	[31]	[9]	Este trabalho
Comandos por <i>templates</i> visuais	✓						
Estabilização e localização <i>indoor</i> da aeronave por visão		✓					
Controle do <i>heading</i> e da velocidade vertical por visão			✓				
VANT seguidor de linha				✓			
VANT seguidor de objeto					✓		
Pouso em plataforma móvel						✓	
Execução autônoma de trajetória por <i>waypoints</i> visuais adquiridos do terreno							✓

Tabela 3 – Compilação das características de cada trabalho relacionado.

Característica analisada	Trabalhos						
	[5]	[6]	[7]	[8]	[31]	[9]	Este trabalho
Utilização/implementação de estação solo		✓		✓		✓	✓
Utilização de VANT comercial		✓		✓			✓
Desenvolvimento do VANT utilizado	✓		✓		✓	✓	
Processamento embarcado no VANT	✓		✓		✓		
Utilização da plataforma Raspberry Pi	✓				✓		
Utilização da plataforma Jetson TK1						✓	
Utilização da plataforma PowerPC440			✓				
Utilização de FPGA			✓				
Utilização do OpenCV	✓	✓	✓	✓	✓	✓	✓
Utilização de marcadores artificiais	✓				✓	✓	
Deteção através de segmentação por cor					✓	✓	
Deteção utilizando o SURF			✓				✓
Deteção por bordas e contornos	✓			✓			
Utilização de múltiplos VANTs							✓
Possibilidade de salvar a trajetória							✓
Possibilidade de compartilhar a trajetória							✓

## 4 CARACTERIZAÇÃO DO PROBLEMA

O controle de localização e navegação são dois componentes de grande importância nos VANTs, pois possibilitam o conhecimento da posição da aeronave em determinado instante e a locomoção entre dois pontos desejados. Em diversos tipos de missão, seja de exploração e reconhecimento ou de execução de trajetos fixos e conhecidos, esses mecanismos são críticos para o alcance do objetivo pretendido. Neste contexto, será exposta a caracterização do problema de localização dos VANTs no terreno. Inicialmente, será apresentado um *overview* contendo algumas técnicas de localização, destacando-se suas deficiências. Posteriormente, serão analisadas algumas vulnerabilidades oficialmente reconhecidas do GPS, sistema muito utilizado atualmente nessas aeronaves. Em seguida, será exposta a definição do problema que se pretende resolver neste trabalho. Por fim, serão elencados as condições de contorno para a utilização do sistema proposto.

### 4.1 Técnicas de localização

Nesta Seção, será apresentado um *overview* das diferentes técnicas de localização. Conforme expresso em [14], os sistemas de navegação devem dispor de uma forma básica para fornecer sua localização e/ou orientação. Tais sistemas podem ser classificados em quatro grandes grupos, de acordo com a técnica utilizada. São elas: triangulação, reconhecimento de padrões, sensoriamento direto e localização estimada (*Dead reckoning*). Em seguida, serão descritas brevemente cada uma.

1. **Triangulação.** Os sistemas baseados em triangulação utilizam a localização de pelo menos três pontos conhecidos para, então, determinar a sua localização. As informações obtidas a partir desses três pontos conhecidos podem ser relativas a distâncias ou a ângulos. O sistema GPS é o mais conhecido que implementa essa técnica. Existem, também, as implementações baseadas na intensidade do sinal emitido pelas ERBs (Estação Rádio Base) e pelos pontos de acesso WLAN (*Wireless Local Area Network*) [37]. Neste caso, faz-se uma associação entre a intensidade do sinal e a distância até a fonte emissora. No entanto, conforme destacado pelos autores referenciados, esses dois últimos métodos possuem precisão mais baixa que o GPS. Uma desvantagem desses métodos é que eles dependem de sistemas eletrônicos externos. Assim, caso estes falhem, o sistema de localização ficará comprometido.
2. **Reconhecimento de padrões.** Nesta técnica, o sistema realiza a coleta de informações através de um conjunto de sensores embarcados e os compara com dados

armazenados em um banco de dados anteriormente adquirido. Uma possível implementação dessa técnica é baseada em visão computacional. Neste caso, imagens adquiridas por uma câmera embarcada podem ser utilizadas para realizar a comparação com imagens já conhecidas do ambiente. Uma desvantagem dessa técnica é que mudanças no ambiente podem afetar o sistema de localização.

3. **Sensoriamento direto.** Nesta técnica, a localização é determinada através da leitura direta de um identificador instalado no ambiente. A localização do identificador é bem conhecida. Duas possíveis implementações dessa técnica são através da utilização de etiquetas RFID (*Radio-Frequency Identification*) ou de *QR-Codes* no local conhecido. Uma desvantagem de sua utilização é a necessidade de acesso físico ao local para instalar o identificador.
4. **Localização estimada** (*Dead reckoning*). Neste caso, o sistema estima a localização baseada em outra anterior conhecida ou também estimada. Para isso, esta técnica utiliza dados proveniente de sensores como acelerômetros, giroscópios, magnetômetros, odômetros visuais, entre outros para realizar tais cálculos. Os Sistemas de Navegação Inercial (SNI) são exemplos de utilização desta técnica. Uma desvantagem conhecida é a ocorrência de grandes erros com o passar do tempo, devido à imprecisão nos sensores e à recursividade no funcionamento do algoritmo.

A Tabela 4 ilustra as técnicas acima descritas e mostra alguns exemplos de implementação.

Técnica	Exemplos de implementação
Triangulação	GPS Estação Rádio-Base (ERB) WLAN
Reconhecimento de padrões	Visão Computacional
Sensoriamento direto	RFID QR-Code
Localização estimada ( <i>Dead reckoning</i> )	Sistema de Navegação Inercial (SNI)

Tabela 4 – Alguns métodos de localização. Adaptada de [14].

## 4.2 Vulnerabilidades do GPS

O GPS é um sistema de localização de propriedade dos Estados Unidos da América e mantido pela Força aérea daquele país. Algumas informações sobre o seu funcionamento já foram expostas na Seção 2.3. No site oficial do sistema [3], há diversas considerações

sobre seu funcionamento. Uma delas alerta que o GPS não é imune a ataques e possui algumas vulnerabilidades oficialmente reconhecidas. Ainda no site oficial, o proprietário do sistema recomenda fortemente que, em aplicações críticas, se utilizem outros mecanismos de localização redundantes ao GPS, pois o sistema pode falhar sem aviso prévio. Neste contexto, pode-se enquadrar a utilização em VANTs, principalmente os de emprego militar, como um caso de aplicação crítica. Sobre as principais vulnerabilidades elencadas do GPS, tem-se: o bloqueio do sinal GPS, a falsificação do sinal GPS e a interferência causada por erupções solares. Esta última não será aqui analisada.

A primeira vulnerabilidade do GPS é o bloqueio do sinal, também conhecido como *jamming*. Isso é possível, pois a potência de transmissão dos satélites emissores do sinal é limitada e varia em torno de 100 a 1000 W [38]. Essa limitação decorre da restrição na fonte de energia disponível gerada pelos painéis solares. Além disso, cada satélite cobre uma grande área, o que resulta em uma pequena fração da intensidade de sinal que chega ao receptor GPS final. Assim, um bloqueador de GPS, mesmo com pouca potência de transmissão, é capaz de mascarar a recepção do sinal original no aparelho receptor. Ou seja, faz que o receptor não receba corretamente as informações do sistema GPS. Esses bloqueadores são conhecidos como *jammers* e, embora seja proibida a sua comercialização no Brasil, podem ser adquiridos pela internet de maneira relativamente fácil. Um simples *jammer* com potência de 35 W é capaz de bloquear receptores GPS em um raio de até 500 m [10]. A Figura 32 ilustra um desses bloqueadores.



Figura 32 – Bloqueador GPS. Fonte [10].

Outra vulnerabilidade conhecida é a falsificação de sinal GPS. Tal técnica é conhecida como *spoofing*. Segundo [39], o *spoofing* pode ser mais ameaçador que o *jamming*. Porque aquele falsifica o sinal GPS e introduz uma localização errada no sistema. Já este apenas nega o acesso à localização. De acordo com o artigo referenciado, a falsificação do sinal GPS pode ser feita através da junção de um software de simulação do sinal GPS e um emissor de rádio frequência. Para ilustrar a lesividade causada pelo *spoofing*, algumas fontes [40] [41] afirmam que, em 5 de dezembro de 2011, um VANT autônomo Americano

(*Lockheed Martin RQ-170 Sentinel*) foi capturado por forças adversas iraquianas que utilizaram esse ataque.

### 4.3 Definição do problema

Nesta Seção, será definido o problema de navegação nos VANT. Inicialmente, será feita uma análise das quatro técnicas de localização apresentadas no início deste capítulo. Assim, pode-se chegar a algumas conclusões. Primeira, a técnica de localização estimada necessita de sensores inerciais de alta precisão, do contrário, acumulará grandes erros com o passar do tempo, sendo imprecisa para longos percursos. Segunda, a técnica de sensoriamento direto necessita de acesso físico ao local para a instalação de um identificador. Tal ação nem sempre será possível executar. Terceira, a técnica de triangulação depende de sistemas externos para o seu funcionamento, podendo falhar caso esses sistemas não funcionem, tenham seu acesso negado ou manipulado por terceiros. Como exemplo de falhas em sistemas de triangulação, tem-se as que afetam o GPS exemplificadas na Seção anterior. Embora hoje bastante utilizado em VANTs, ele apresenta vulnerabilidades reconhecidas oficialmente por seu proprietário que adverte sobre a necessidade de haver outro sistema de localização redundante quando utilizado em aplicações críticas. Incluem-se, nesses casos, aplicações em dispositivos militares e na aviação.

Após essa análise preliminar, vislumbra-se que a técnica baseada em reconhecimento de padrões, utilizando visão computacional através de uma câmera embarcada, pode ser bastante promissora no emprego em VANTs. Por esse motivo, este trabalho se propõe a estudar, implementar e testar um sistema de localização e navegação de VANTs baseado em reconhecimento de imagens do terreno. Como expresso na introdução e depois detalhado ao longo deste trabalho, o sistema proposto é dividido em dois passos principais. Estes envolvem a utilização de um VANT líder e de um VANT seguidor. Inicialmente, o VANT líder realizará um percurso qualquer sobre uma área, capturando imagens do terreno abaixo da aeronave. Tais imagens serão processadas e armazenadas no banco de dados de trajetórias. Após a realização do voo inicial pelo VANT líder, o VANT seguidor carregará os dados da trajetória e iniciará seu voo autônomo. Este será capaz de refazer a trajetória inicial utilizando, para isso, mecanismos de visão computacional no reconhecimento de imagens do terreno.

A utilização do mecanismo de navegação proposto apresenta algumas vantagens e desvantagens conhecidas. Dentre as vantagens, destacam-se:

1. Possibilidade de armazenamento da trajetória realizada;
2. Possibilidade de compartilhamento da trajetória entre diversas aeronaves;

3. Possibilidade de execução da trajetória múltiplas vezes até a modificação das características do terreno; e
4. Possibilidade de aquisição e fusão de trajetórias por muitos VANTs líderes.

Após as vantagens, passa-se para a análise das restrições e condições de contorno necessárias ao funcionamento do sistema.

1. O VANT deve possuir visibilidade do terreno abaixo sobre qual sobrevoa;
2. A superfície abaixo do VANT deverá possuir características (*features*) suficientes para que ocorra o reconhecimento da imagem; e
3. As características do terreno devem permanecer constantes durante o intervalo de tempo entre a aquisição da trajetória pelo VANT líder e a sua execução pelo VANT seguidor.



## 5 SOLUÇÃO PROPOSTA

Neste capítulo, será apresentada a solução proposta para a localização e navegação de VANTs utilizando reconhecimento de imagens do terreno. Serão descritos os princípios de funcionamento do sistema estudado e implementado neste trabalho. Como já explicitado anteriormente, o objetivo é possibilitar a realização de voos pelo VANT seguindo um trajeto estabelecido, utilizando-se uma câmera embarcada e o emprego de visão computacional. Assim, para que se possa alcançar este objetivo, o funcionamento do sistema foi dividido em duas grandes etapas. A primeira etapa consiste na realização de um voo manual para reconhecimento e aquisição de imagens do terreno. Esta etapa é realizada por um *VANT líder*. A segunda etapa consiste em um ou mais voos autônomos, utilizando a navegação através do reconhecimento das imagens adquiridas na etapa anterior. Estes voos guiados por imagem poderão ser realizados por um ou mais *VANTs seguidores*. A seguir, serão descritas estas duas etapas com maiores detalhes.

### 5.1 Etapa 1: Voo de aquisição das imagens do terreno

Esta é a primeira etapa do funcionamento do sistema de navegação por imagem e será executado pelo VANT líder. Nesta etapa, a aeronave irá sobrevoar, de maneira manual, uma trajetória qualquer realizando, através da câmera embarcada, a aquisição de imagens do terreno imediatamente abaixo. Cada imagem capturada do caminho sobrevoado representará um *waypoint* visual que posteriormente será utilizado na etapa seguinte para voos autônomos. Após a captura das imagens, estas são transmitidas em sequência para a estação de solo, onde serão extraídos os seus *keypoints* e *descriptors*. Sabe-se que um conjunto de *keypoints* e *descriptors* podem formar uma espécie de "assinatura" da imagem analisada. Estas informações serão úteis para realizar o seu reconhecimento posterior dentro de um banco de dados com diversas outras imagens (*image matching*). Juntamente com essas informações, são salvos também os respectivos índices das imagens. Estes especificam a posição em que a imagem se encontra dentro de uma sequência de outras imagens do trajeto. Assim, esse conjunto de *waypoints* visuais adquirido durante o voo irá compor o percurso executado pelo VANT líder. A Figura 33 representa o fluxograma das atividades realizadas nesta primeira etapa descrita.

Com relação aos aspectos operacionais da navegação, a Figura 34 ilustra a movimentação do voo do VANT líder no terreno. Primeiramente, a aeronave fará uma decolagem vertical no ponto de partida da trajetória. Após isso, por motivo de simplificação do sistema, a aeronave manterá a sua altitude constante, movendo-se sempre paralelamente ao solo. Tal condição de contorno não prejudica a utilidade do sistema de navegação por

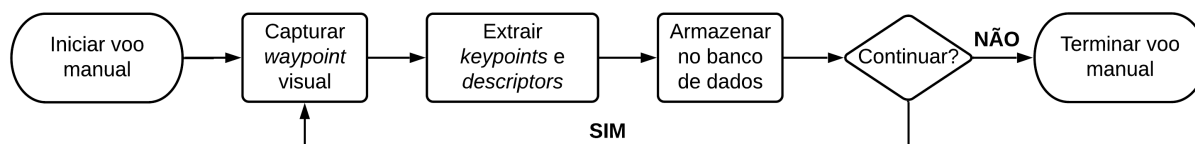


Figura 33 – Diagrama com as etapas do voo de reconhecimento e aquisição de dados do trajeto.

imagem, pois há outros sensores embarcados de fácil utilização (ultrassônico, barômetro e altímetro a LASER) que permitem o voo em altitude constante. Pode-se também, juntamente com cada *waypoint* visual capturado, salvar a altitude em que a aeronave se encontra. A partir do ponto inicial de decolagem, serão, então, adquiridas as imagens do terreno imediatamente abaixo do VANT. Caso um *gimbal* esteja disponível no VANT, a câmera embarcada deverá ficar sempre orientada verticalmente, apontando para baixo. Se não houver um *gimbal*, a aquisição das imagens será feita com os menores ângulos possíveis de atitude da aeronave. Como uma solução possível, pode-se fazer pequenos voos pairados para, então, realizar a captura. O objetivo disso é que as imagens possam refletir a posição imediatamente abaixo do VANT.

Voltando à Figura 34, o quadrado cinza representa a área de visada da câmera e o retângulo mais escuro representa a área já capturada e processada do trajeto. A seta para a direita indica a direção e o sentido do movimento do VANT. Embora o percurso ilustrado na figura seja uma linha reta, é possível que o VANT realize outros trajetos mais complexos no terreno. Por fim, após percorrer todo o trajeto desejado e ter capturado os *waypoints* visuais, a aeronave pousará verticalmente no destino final. As imagens adquiridas em voo poderão ser enviadas para a estação de solo durante o voo, se houver um link constante de dados, ou após a sua conclusão.

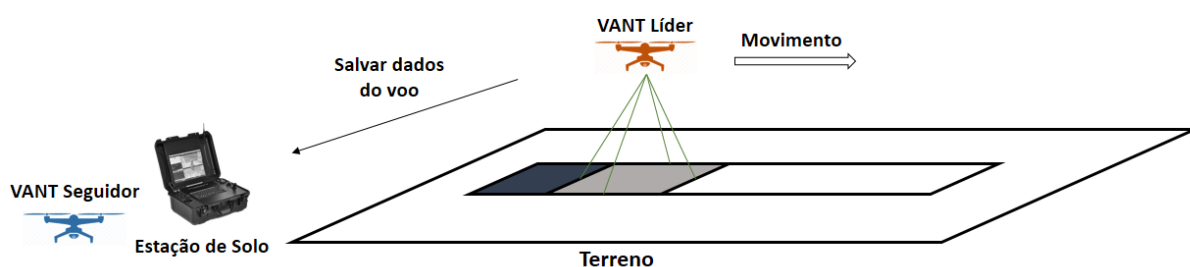


Figura 34 – Voo de reconhecimento e aquisição das imagens do terreno. Neste voo, o VANT líder fará a aquisição das imagens do terreno e a transmissão para a estação de solo.

## 5.2 Etapa 2: Voo autônomo guiado por imagens

Esta é a segunda etapa do funcionamento do sistema de navegação por imagem e representará os voos do VANT seguidor, após os dados da trajetória já estarem processados e armazenados na estação de solo. Neste voo, a aeronave seguidora irá primeiramente carregar os dados da trajetória armazenada provenientes da estação de solo. Novamente, tais dados são compostos pelos *keypoints*, *descriptors* e os respectivos índices. Em conjunto, esses dados representam os sucessivos *waypoints* visuais da trajetória original realizada pelo VANT líder. Após isso, o VANT seguidor realizará a sua decolagem de maneira vertical, partindo do ponto de origem até alcançar a altitude do voo de reconhecimento. Para isso, serão utilizados apenas os sensores embarcados na aeronave, como o barômetro, ultrassônico, acelerômetro e giroscópio. Com o veículo no ar, será então feita a aquisição da imagem do terreno pela câmera embarcada. Nesta imagem adquirida, serão extraídos os *keypoints* e calculados os seus respectivos *descriptors*. Com tais informações disponíveis, estes serão comparados, através de *image matching*, com os dados constantes no banco de dados da trajetória carregada da estação de solo. Consultando-se o índice da imagem que ocorreu o *matching*, pode-se conhecer o ponto da trajetória em que o VANT seguidor se encontra. Então, extrai-se o centro do *waypoint* visual que está presente na imagem adquirida. Caso a aeronave, por condições externas, se desvie do percurso e não encontre nenhum *waypoint* visual associado ao trajeto (não houve o *matching*), ela sairá do modo autônomo e aguardará comandos do operador. Uma outra solução possível é aumentar a sua altitude de voo para possibilitar a câmera capturar uma área maior do terreno.

Após reconhecido um *waypoint*, o algoritmo centralizará a aeronave sobre ele e, então, voltará à altitude original de voo. Com o *waypoint* identificado e seu centro calculado, o sistema de navegação irá executar comandos para centralizar a aeronave sobre esse ponto. Aqui, pode-se empregar duas estratégias de perseguição dos *waypoints*. Uma é relativa à perseguição do *waypoint* com o maior índice visível na imagem capturada e a outra é a visita obrigatória de todos os *waypoints* do trajeto. Tais estratégias serão descritas com mais detalhes na Subseção 5.2.1. Após os cálculos dos movimentos necessários para centralizar a aeronave sobre um *waypoint*, são, então, enviados comandos para o seu piloto automático. Finalizada a perseguição/visitação de todos os *waypoints* visuais do trajeto, o VANT seguidor realizará o pouso vertical. Destaca-se que não há, neste voo, a leitura da posição através do sistema de posicionamento por satélite. Estas etapas aqui descritas estão sequenciadas e esquematizadas na Figura 35.

A Figura 36 possui os mesmos elementos que a Figura 34 exposta na Seção anterior, dispensando-se, assim, explicações repetidas. A diferença, no entanto, consiste que este voo é realizado pelo VANT seguidor, enquanto que o VANT líder fica pousado em terra. Na figura, o trajeto pontilhado representa a parte que já foi executada pelo VANT seguidor. Uma observação importante é que o VANT seguidor estará apto a refazer o trajeto até

que ocorra a modificação física do solo, ou seja, até o ponto em que o algoritmo de *image matching* não consiga mais identificar os *waypoint* visuais.

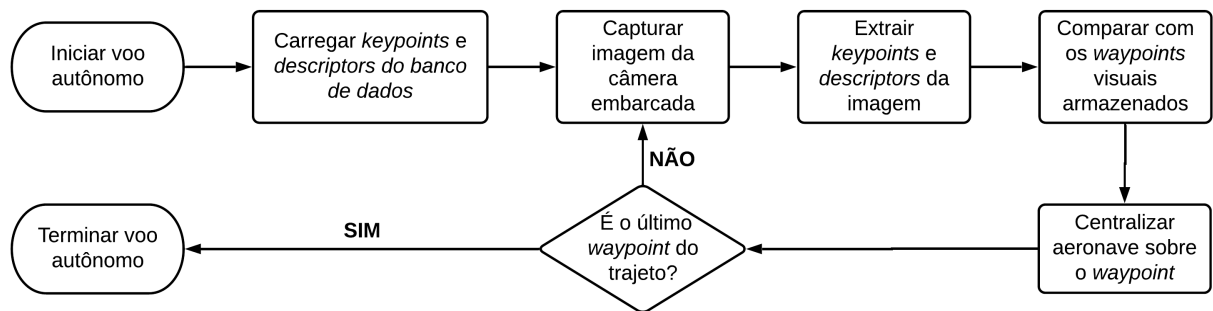


Figura 35 – Diagrama com as etapas do voo de navegação por imagem.

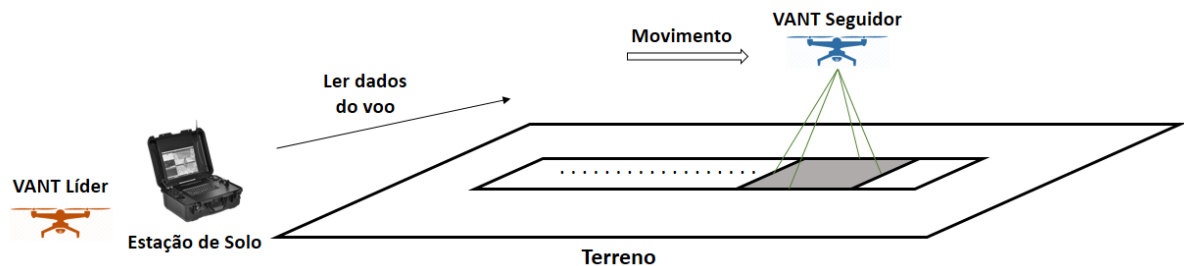


Figura 36 – Voo de navegação por imagem. Neste voo, os dados do percurso são carregados no VANT seguidor e, utilizando reconhecimento de imagens do terreno, a aeronave é capaz de refazer o trajeto carregado.

### 5.2.1 Estratégias para execução da trajetória armazenada

Em relação às estratégias de perseguição dos *waypoints*, há duas possíveis implementações. A primeira baseia-se na perseguição do *waypoint* visível com maior índice. Ou seja, caso a câmera embarcada no VANT seguidor visualize simultaneamente os *waypoints* de índices  $n$ ,  $(n + 1)$ , ...,  $(n + i)$ , o VANT se movimentará em direção ao *waypoint*  $(n + i)$ . Isso fará com que os *waypoints*  $n$ ,  $(n + 1)$ , ...,  $(n + i - 1)$  não sejam perseguidos. Isso pode ocorrer, pois, como há sobreposição (*overlap*) entre os *waypoints*, uma mesma imagem capturada pode conter vários *waypoints* do trajeto. Essa estratégia está esquematizada na Figura 37.

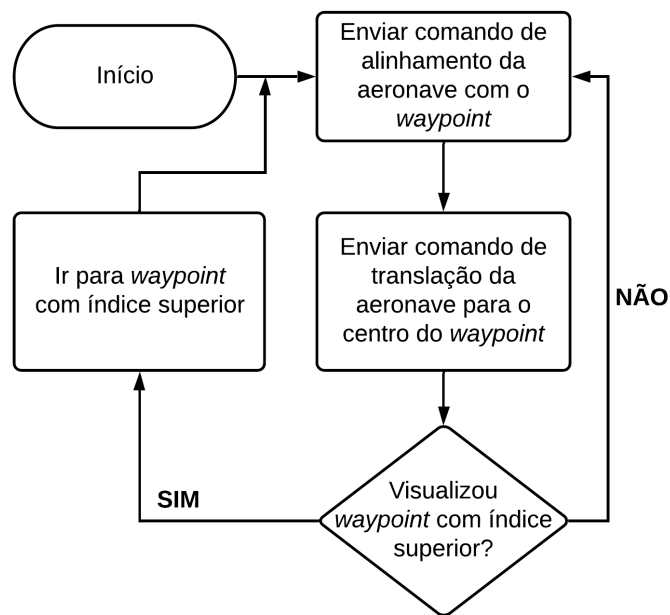


Figura 37 – Estratégia de navegação através da perseguição do *waypoint* visível de maior índice.

A segunda estratégia possível é a perseguição obrigatória de todos os *waypoints* em sequência. Ou seja, o VANT seguidor deverá visitar obrigatoriamente todos os *waypoints* pertencentes à trajetória armazenada. Nesse caso, um *waypoint* será considerado visitado quando os parâmetros de distância da aeronave até o centro do *waypoint* e de ângulo de alinhamento estiverem abaixo de limiares previamente definidos. A Figura 38 representa essa estratégia de perseguição. Na figura, os valores  $E1$  e  $E2$  representam, respectivamente, a distância e o ângulo entre a aeronave e o *waypoint*.

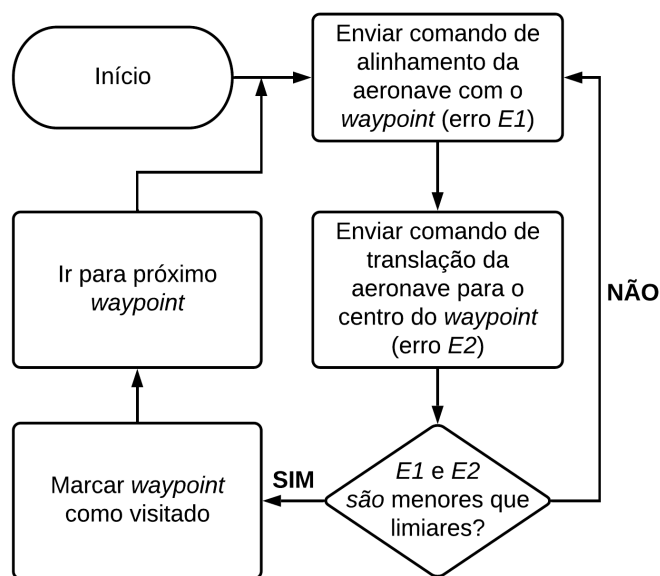


Figura 38 – Estratégia de navegação com visitaçõ de todos os *waypoints*.

### 5.2.2 Cálculo da posição do VANT em relação ao *waypoint* visual

Nesta Subseção, serão discutidos aspectos sobre o posicionamento da aeronave em relação ao *waypoint* visualizado. Para facilitar o entendimento, a Figura 39 ilustra um VANT em voo e mostra a parte do terreno capturado pela câmera embarcada. As linhas vermelhas indicam o limite do campo visual da câmera, e a cruz tracejada indica o centro de um *waypoint* visual reconhecido. Utilizando os mecanismos de visão computacional disponíveis no OpenCV, pode-se encontrar, na imagem processada, o centro do *waypoint* visual sobre o qual se deseja sobrevoar.

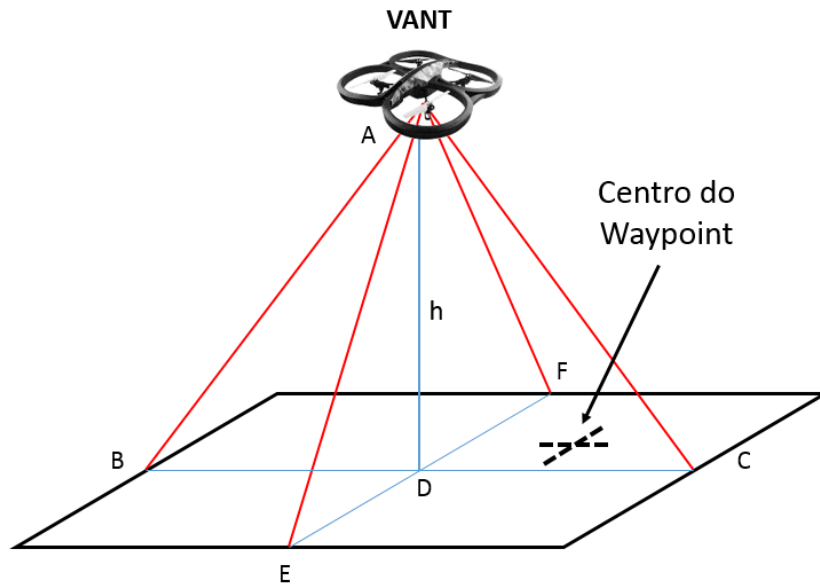


Figura 39 – VANT seguidor visualizando um *waypoint* no terreno.

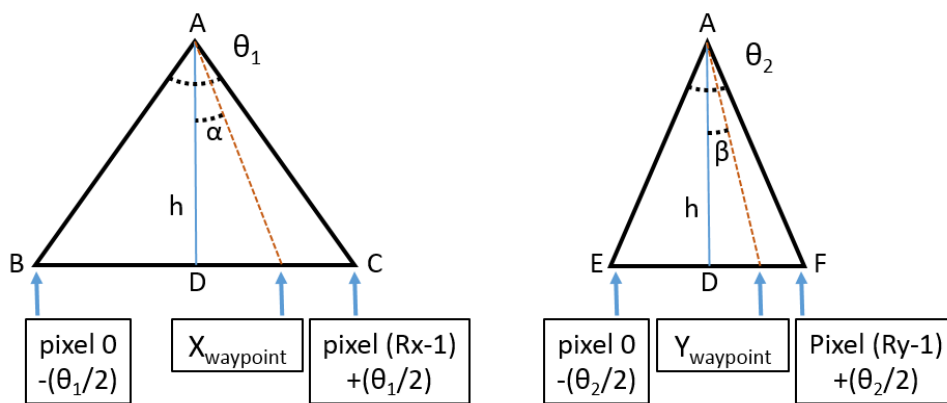


Figura 40 – Ângulos de abertura da câmera vertical embarcada no VANT. Na direita, tem-se a abertura horizontal. Na esquerda, tem-se a abertura vertical.

No resultado do *imagem matching*, o centro do *waypoint* é expresso em pixels, tendo como coordenadas  $X_{waypoint}$  e  $Y_{waypoint}$  para os eixos  $x$  e  $y$  respectivamente. No entanto, pode-se necessitar da informação do centro do *waypoint* em unidades métricas e não em

pixels. Para isso, serão necessárias algumas informações adicionais, como a resolução da câmera, os seus ângulos de abertura e a altitude da aeronave. Conforme representados na Figura 40,  $\theta_1$  e  $\theta_2$  são os ângulos de abertura da câmera nos eixos horizontal ( $\overline{BC}$ ) e vertical ( $\overline{EF}$ ) respectivamente.  $h$  ( $\overline{AD}$ ) representa a altura de voo.  $R_x$  e  $R_y$  são as resoluções, em pixels, da câmera na horizontal e vertical. Assim, utilizando-se a reta  $\overline{AD}$  como referência para a medição dos ângulos, o pixel  $(R_x - 1)$  representa o ângulo  $+(\theta_1/2)$  e o pixel 0 representa o ângulo  $-(\theta_1/2)$ . Similarmente, é válido para o eixo  $y$ . O pixel  $(R_y - 1)$  representa o ângulo  $+(\theta_2/2)$  e o pixel 0 representa o ângulo  $-(\theta_2/2)$ .

Visto isso, ligando-se a coordenada  $X_{waypoint}$ , do centro do *waypoint*, à câmera (ponto A), tem-se uma reta que forma um ângulo  $\alpha$  com a reta  $\overline{AD}$ . Similarmente, ligando-se  $Y_{waypoint}$  à câmera, tem-se outra reta que forma um ângulo  $\beta$ , também, com  $\overline{AD}$ . Esses dois ângulos são importantes pois, conhecendo-os, pode-se calcular a distância métrica do centro da imagem fornecida pela câmera até o centro do *waypoint* visual identificado. As equações para o cálculo de  $\alpha$  e  $\beta$  estão representadas em 5.1 e 5.2 respectivamente.

$$\alpha = \arctan \left( \frac{(X_{waypoint} - R_x/2) \cdot \tan(\theta_1/2)}{R_x/2} \right) \quad (5.1)$$

$$\beta = \arctan \left( \frac{(Y_{waypoint} - R_y/2) \cdot \tan(\theta_2/2)}{R_y/2} \right) \quad (5.2)$$

Até o momento, os cálculos mostrados foram para encontrar a distância até o centro do *waypoint* quando a aeronave está voando paralelamente ao solo. Caso a imagem seja capturada quando o VANT possui alguma atitude, os ângulos de *roll* e *pitch* devem ser levados em conta nos cálculos. A Figura 41 ilustra esse caso. O ângulo de *roll* deve ser somando ao ângulo  $\alpha$  e o *pitch*, ao  $\beta$ . A leitura da atitude pode ser realizada através de uma consulta aos sensores inerciais da aeronave. Com todas as informações disponíveis, pode-se, então, utilizar as Equações 5.3 e 5.4 para calcular a distância métrica  $X$  e  $Y$  do VANT ao centro do *waypoint* no plano do terreno.

$$Distância X = h \cdot \tan(roll + \alpha) \quad (5.3)$$

$$Distância Y = h \cdot \tan(pitch + \beta) \quad (5.4)$$

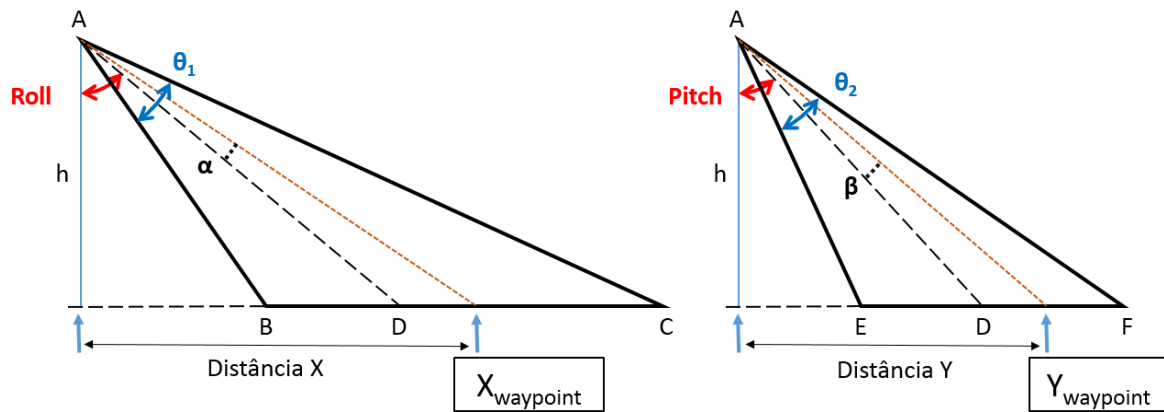


Figura 41 – Distância do VANT até o centro do *waypoint* quando a aeronave não está paralela ao terreno.

### 5.2.3 Cálculo dos movimentos da aeronave

Nesta Subseção, serão descritos os métodos utilizados para o cálculo dos movimentos de rotação e de translação necessários para posicionar o VANT sobre o centro do *waypoint* visual. Os movimentos da aeronave serão decompostos em movimentos de rotação e de translação, realizados sucessivamente até se alcançar o *waypoint*. Como a aeronave se move paralelamente ao solo, os movimentos de translação também são paralelos ao solo e não possuem componentes no eixo  $z$ . Em relação às rotações, estas são sempre sobre o eixo  $z$ , visando alinhar a aeronave ao *waypoint* visual.

**Rotação da aeronave:** O ângulo de rotação necessário a ser aplicado à aeronave é calculado através da imagem obtida em tempo real da câmera embarcada e da imagem processada do *waypoint* visual. Para isso, utiliza-se a função *findHomography()*, disponível no OpenCV. Esta função calcula a matriz homográfica que contém a relação do mapeamento entre as duas imagens, a da câmera e a do *waypoint* visual. Através dela, podem-se obter várias informações interessantes. Uma delas é o ângulo relativo de rotação entre as duas imagens. Este ângulo pode ser encontrado utilizando a Equação 5.5, onde  $\varphi$  representa o ângulo desejado e  $m_{1,1}$  e  $m_{1,2}$  elementos da matriz homográfica. A Figura 42 ilustra, em azul, o ângulo  $\varphi$  de rotação.

$$\varphi = \arctan\left(\frac{m_{1,2}}{m_{1,1}}\right) \quad (5.5)$$

**Translação da aeronave:** A translação da aeronave é calculada utilizando-se, como origem do vetor deslocamento, o centro da imagem obtida (posição atual do VANT) e, como destino, o centro do *waypoint* visual identificado (posição desejada do VANT). A Figura 42 ilustra, em vermelho, esse vetor deslocamento. Para o cálculo do centro em pixels do *waypoint* identificado, basta conhecer a resolução da imagem utilizada como *waypoint* visual. Assim, calculam-se as coordenadas do centro dessa imagem e aplica-se a



matriz homográfica. O resultado será a localização em pixels do centro do *waypoint* visual na imagem atual vista pelo VANT.

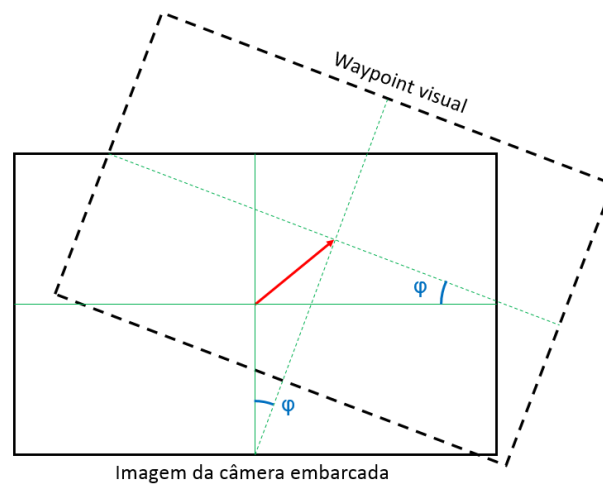


Figura 42 – Rotação e translação da aeronave para se alcançar o *waypoint* visual.

### 5.3 Considerações finais

O sistema proposto permite a localização e a navegação de VANTs utilizando o reconhecimento de imagens do terreno, onde cada imagem adquirida funciona como um *waypoint* da trajetória realizada pelo VANT líder. Embora estas imagens armazenadas registrem o *waypoint* como uma cena estática, elas podem conter pequenos elementos móveis como, por exemplo, pessoas e veículos. Tais elementos, a partir de determinadas altitudes de voo, não influenciarão no reconhecimento pretendido do *waypoint*, pois representarão uma pequena porção da imagem. Destaca-se também que, para ocorrer o *image matching*, não é necessária a compatibilidade de todos os *keypoints* da imagem, mas apenas de uma porcentagem predefinida como limite. Além disso, podem ser utilizados diferentes limiares de aceitabilidade para o reconhecimento dos *waypoints* nas diversas altitudes de voo da aeronave. Por fim, no funcionamento do sistema, foram utilizados apenas mecanismos de visão computacional, poder de processamento e uma câmera embarcada na aeronave. Cabe, ainda, destacar algumas das possíveis vantagens da utilização desse sistema: a) Possibilidade de armazenamento e compartilhamento da trajetória entre múltiplos VANTs; b) Não é necessário a colocação de marcos artificiais no terreno; c) O sistema de localização por imagem não requer grande capacidade de processamento; e d) O sistema proposto pode ser utilizado como um mecanismo redundante de localização existente na aeronave.

## 6 IMPLEMENTAÇÃO DA SOLUÇÃO

Neste capítulo, serão descritos aspectos sobre a implementação da solução proposta. Serão abordados pontos sobre a aeronave utilizada, incluindo suas principais características e seu protocolo de comunicação. Logo após, será descrita a biblioteca de controle empregada. Em seguida, serão abordados aspectos sobre a estação de solo, como seus principais componentes e funcionamento. Por fim, será exposto um estudo comparativo de alguns algoritmos de *image matching* para auxiliar na escolha do melhor método a ser utilizado.

### 6.1 Aeronave utilizada nos experimentos

A aeronave utilizada nos experimentos foi a *AR.Drone 2.0 Elite Edition* do fabricante *Parrot*, representada na Figura 43. Esta aeronave possui tanto a capacidade de voo *indoor*, utilizando o protetor de hélices, como de voo *outdoor*. Possui como hardware um processador ARM Cortex A8 de 32 bits e 1GHz de *clock*, giroscópio, acelerômetro e magnetômetro, todos de 3 eixos, barômetro e sensor ultrassônico, estes dois últimos para a medição de altitude. Soma-se ao hardware da aeronave duas câmeras embarcadas, uma orientada horizontalmente e outra verticalmente. A câmera vertical possui resolução máxima de 640x360 pixels e a horizontal de 1280x720 pixels.

Os dados de controle, leitura dos parâmetros (*status*) e transmissão de vídeo do VANT ocorrem através de um enlace WiFi 802.11 padrões b, g ou n. Esses dados trafegam



Figura 43 – Nesta figura, estão representados a aeronave, a estação de solo e o *joystick* para controle manual.

através dos protocolos de camada de transporte UDP (*User Datagram Protocol*) e TCP (*Transmission Control Protocol*). As portas utilizadas são: comandos de controle via UDP porta 5556, leitura dos parâmetros (*status*) via UDP porta 5554 e *streaming* de vídeo via TCP porta 5555. A Figura 44 ilustra essa comunicação. A seguir, serão descritos cada um desses mecanismos com mais detalhes.

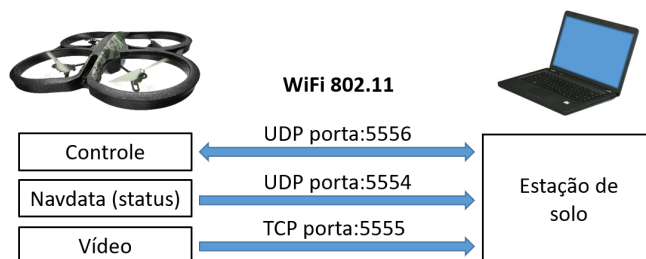


Figura 44 – Arquitetura de comunicação do sistema.

**Controle:** A configuração e o controle dos movimentos da aeronave são feitos através de comandos AT (*Attention commands*) enviados via UDP para a porta 5556. Tais comandos são strings de texto contendo principalmente cabeçalho, número de sequência e parâmetros de configuração da aeronave. No caso dos comandos de movimentos, estes devem ser enviados a cada 30 ms para resultar em deslocamentos mais suaves, conforme orientação do fabricante. Dentre os comandos existentes, os principais são:

**AT\*REF:** Utilizado para realizar a decolagem (*takeoff*), o pouso (*landing*) e a parada de emergência (*emergency stop*). Este último comando é utilizado em casos de colisão, o qual desliga rapidamente todos os motores da aeronave;

**AT\*FTRIM:** Utilizado para configurar o plano horizontal de referência da aeronave. Tal comando deve ser enviado antes do VANT levantar voo, ainda sobre o solo. A configuração do plano horizontal é importante para o correto funcionamento dos sensores inerciais; e

**AT\*PCMD:** Utilizado para controlar a atitude e os movimentos da aeronave. Controla os ângulos de *roll*, *pitch*, *yaw*, além de sua velocidade vertical. Alterando-se os ângulos de atitude, a aeronave se move em voo.

**NavData:** O status da aeronave pode ser obtido a partir da leitura dos pacotes *NavData* enviados via UDP para a porta 5554. Informações de velocidade, atitude, altitude, velocidade dos motores, tensão da bateria, entre outras podem ser acessadas através da leitura desse pacote. O VANT envia essas informações juntas em um único pacote de dados, havendo duas possíveis taxas de envio. Uma a 15 vezes por segundo, no modo DEMO, e outra a 200 vezes por segundo, no modo FULL.

**Vídeo:** O AR.Drone 2.0 utiliza dois mecanismos de compressão de vídeo, o H.264

via hardware e o MPEG4.2 via software. Na configuração da aeronave, um desses algoritmos é escolhido para a compressão do vídeo. Além disso, pode-se escolher entre a câmera vertical e a horizontal como origem das imagens. A transmissão ocorre via TCP na porta 5555.

A documentação completa da aeronave pode ser encontrada no documento *ARDrone Developer Guide.pdf*, disponível em [42].

## 6.2 Biblioteca de comunicação com o VANT

Com o intuito de facilitar e diminuir o tempo de desenvolvimento do sistema de navegação por imagem, foi utilizada a biblioteca *YADrone* [43], implementada na linguagem de programação Java. Esta biblioteca faz a interface com a aeronave utilizada, fornecendo uma API (*Application Programming Interface*) para controle e programação mais facilitada ao programador. A API possui diversas funções que permitem enviar comandos de movimento à aeronave, ler os parâmetros (status) de voo e gerenciar o *streaming* de vídeo. Algumas dessas funções serão brevemente expostas e descritas abaixo.

- **flatTrim():** Esta função configura o plano de referência da aeronave. Deve ser executada com a aeronave no solo, antes do início do voo;
- **landing():** Esta função realiza o pouso da aeronave;
- **takeOff():** Esta função realiza a decolagem da aeronave;
- **move3D(speedX, speedY, speedZ, speedSpin):** Esta função realiza a movimentação da aeronave. Os parâmetros são ponto flutuante que variam de -1 a +1;
- **isConnected():** Esta função verifica se o programa está conectado à aeronave;
- **getNavDataManager():** Este objeto é utilizado para o gerenciamento da leitura dos parâmetros (status) da aeronave;
- **getVideoManager():** Este objeto é utilizado para o gerenciamento do controle e recepção do *streaming* de vídeo saindo da aeronave; e
- **setMaxAltitude(altitude)** e **setMinAltitude(altitude):** Funções utilizadas para configurar a altitude máxima e mínima de voo da aeronave.

## 6.3 Estação de solo

Nesta Seção, serão discutidos alguns aspectos sobre a estação de solo implementada e sua utilização pelo operador do sistema. Na concepção inicial da solução, a estação de solo executaria apenas as tarefas de processar, salvar e compartilhar os dados da trajetória executada, inicialmente, pelo VANT líder e, posteriormente, replicada pelo VANT seguidor. Porém, na efetiva implementação do sistema, necessitou-se fazer uma adaptação. Isso ocorreu porque a aeronave utilizada não possuía um sistema de processamento embarcado acessível e de fácil utilização pelo desenvolvedor. Ou seja, o acesso ao hardware interno e ao sistema operacional da aeronave está parcialmente bloqueado pela empresa fabricante. Assim, o código responsável pelo controle de voo, que estaria anteriormente embarcado no VANT, teve que ser implementado na estação de solo. No entanto, essa mudança não prejudicou a demonstração e validação do funcionamento do sistema. A estação de solo, no caso do VANT líder, ainda é capaz de salvar, processar e armazenar os *waypoints* da trajetória executada. Já no caso do VANT seguidor, a estação de solo carrega os dados da trajetória salva, recebe o *streaming* de vídeo em tempo real da aeronave, realiza o processamento necessário utilizando visão computacional e envia os comandos de movimento para o VANT.

### 6.3.1 Aspectos da implementação

Sobre os componentes de software da estação de solo, estes estão ilustrados na Figura 45. Cada caixa representa um componente do sistema e as linhas representam os relacionamentos entre eles. A seguir, serão descritos brevemente cada componente e o seu papel no sistema.

**Principal:** Este componente é o núcleo do software da estação de solo. Ele controla e coordena todos os outros componentes. Também implementa a interface gráfica do usuário.

**Thread Captura *Waypoints*:** Este componente é responsável por realizar a captura dos *waypoints* visuais durante o voo de aquisição das imagens no trajeto realizado pelo VANT líder. A captura pode ocorrer de forma manual, através de cliques em botão específico constante na interface gráfica, ou de forma automática, a uma taxa fixa selecionada pelo operador do sistema.

**Thread *Joystick*:** Este componente realiza a leitura do *joystick* e envia os dados para o componente *Comandos VANT*. O componente *Thread Joystick* assume o controle do VANT sempre que qualquer botão do *joystick* é pressionado pelo operador. Assim, o piloto humano do VANT sempre terá precedência sobre o modo autônomo de voo.

**Thread Processamento de Imagem:** Este componente é responsável pelo processamento do *streaming* de vídeo recebido do VANT. Ou seja, ele realiza a busca da

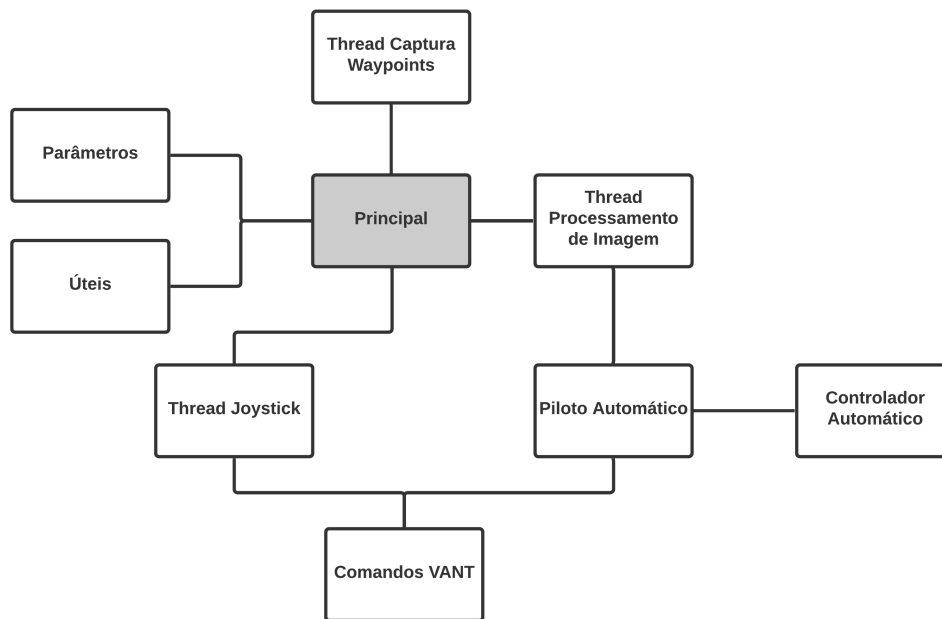


Figura 45 – Diagrama com os principais componentes do software da estação de solo.

imagem recebida (*image matching*) dentro do conjunto dos *waypoints* visuais da trajetória carregada. Além disso, envia para o componente *Piloto Automático* as informações sobre os *waypoints* encontrados na imagem analisada.

**Piloto Automático:** Este componente implementa as duas estratégias de perseguição dos *waypoints* visuais, descritas na Subseção 5.2.1 deste trabalho. Ele também envia para o componente *Comandos VANT* os movimentos necessários para se alcançar o *waypoint* desejado.

**Comandos VANT:** Este componente é responsável pela comunicação e envio dos comandos de movimento à aeronave. Para isso, ele faz uso da biblioteca YADrone para interfacear com o VANT. Os comandos de movimento podem ter como origem o componente *Thread Joystick* ou o componente *Piloto Automático*. Os movimentos oriundos do operador humano atuando no *joystick* sempre possuem precedência sobre os comandos do piloto automático. Assim, o pressionamento de qualquer botão do *joystick* desabilita o piloto automático.

**Controlador Automático:** Este componente implementa o controlador On-Off e o PID, os quais são utilizados para regular os movimentos da aeronave.

**Parâmetros:** Este componente é responsável por armazenar alguns parâmetros de configuração da estação de solo e do VANT.

**Úteis:** Este componente possui algumas funções recorrentemente utilizadas pelo sistema. São implementadas funções para salvar o *log* de funcionamento do sistema, para escrever e ler arquivos, para realizar temporizações de *delay*, entre outras funções.

Após a apresentação dos principais componentes do sistema, será descrita a sequência de atividades realizadas pela estação de solo quando esta recebe uma imagem durante o voo do VANT seguidor. Visando facilitar o entendimento, foi utilizado simplificada-mente um diagrama de sequência, ilustrado na Figura 46. Quando conectado, o VANT seguidor fica constantemente enviando o *streaming* de vídeo para a estação de solo, que pode aceitar e processar um quadro recebido ou rejeita-lo. Aceitando o quadro, inicia-se um ciclo de processamento com as seguintes etapas: 1) São extraídos os *keypoints* e *descriptors* desse quadro. 2) Realiza-se a busca, por *image matching*, do quadro no banco de dados de *waypoints* visuais. 3) Verificam-se quais os índices dos *waypoints* visíveis no quadro analisado. 4) Aplica-se a estratégia de perseguição dos *waypoints* selecionada. 5) Calculam-se os parâmetros necessários para deslocamento do VANT para o *waypoint*. 6) Aplica-se o controlador automático nos parâmetros encontrados. 7) Envia-se os comandos de movimento para o VANT.

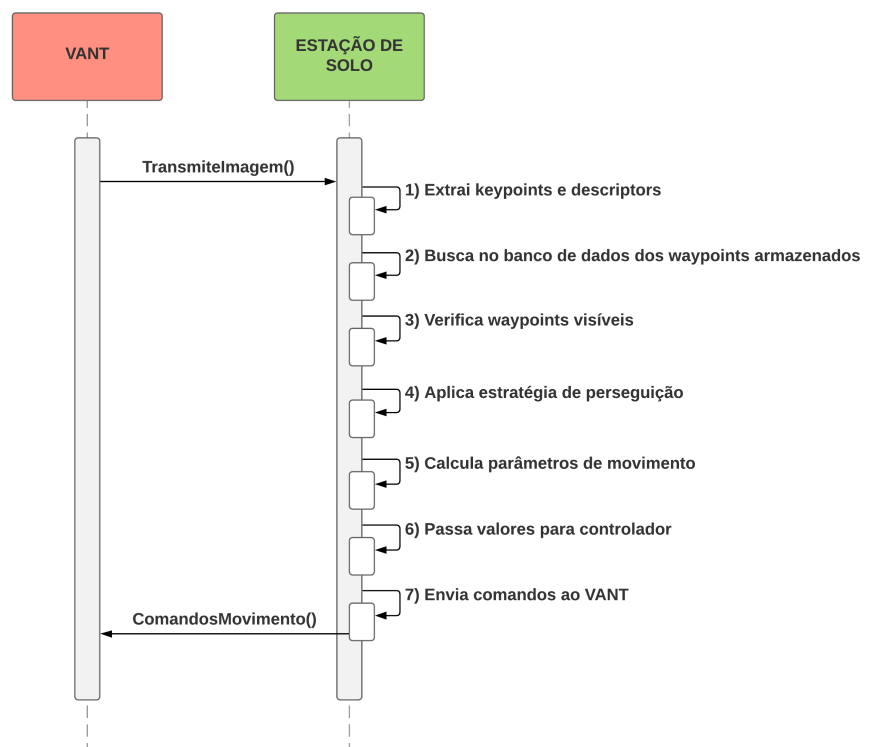


Figura 46 – Diagrama de sequência representando uma iteração no ciclo de processamento e controle durante o voo do VANT seguidor.

### 6.3.2 Interface gráfica

Nesta Subseção, será apresentada a interface gráfica do software da estação de solo desenvolvida. Serão apresentadas as diferentes áreas de controle no software, desde a configuração do *streaming* de vídeo, passando pela aquisição dos *waypoints*, até a

habilitação do piloto automático. A Figura 47 mostra a janela principal do software da estação de solo.

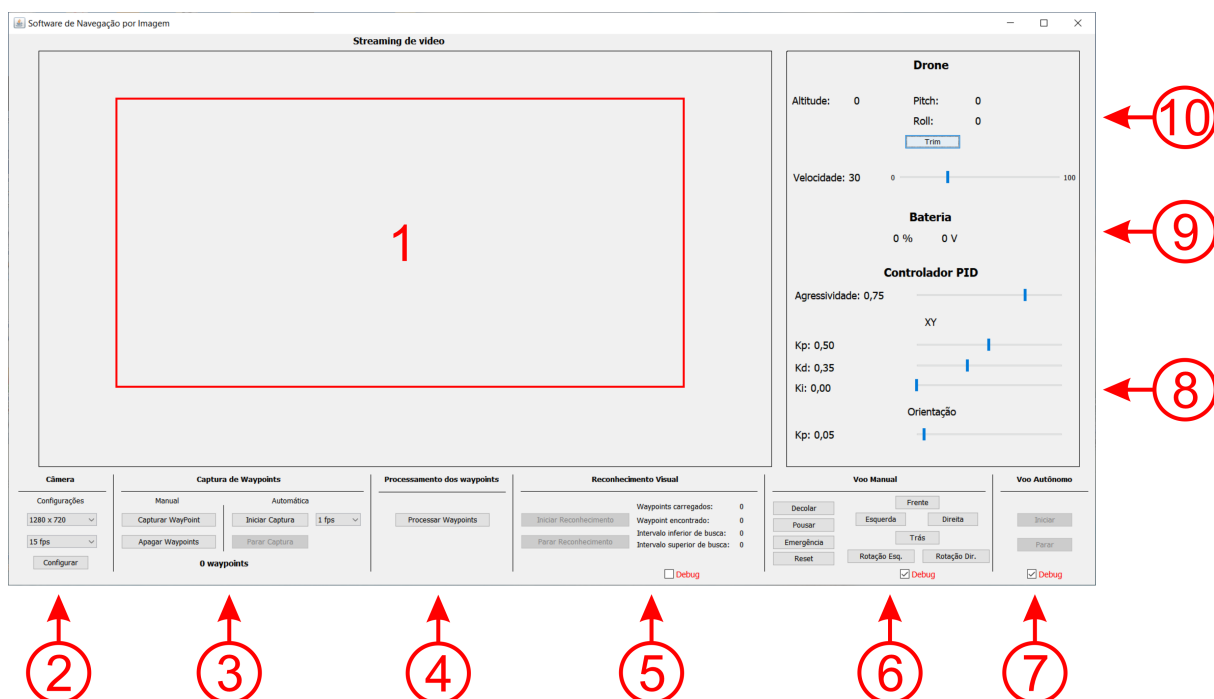


Figura 47 – Interface gráfica do software da estação de solo.

1. Área destinada à visualização do *streaming* de vídeo enviado pelo VANT.
2. Configuração da resolução e taxa de quadros por segundo do *streaming* de vídeo.
3. Área destinada à captura dos *waypoints* visuais. Esta pode ser feita de maneira manual ou automática, a uma taxa selecionada.
4. Botão para realizar o processamento dos *waypoints* do trajeto e, posteriormente, o carregamento em memória desses dados processados.
5. Área destinada à inicialização da busca e do reconhecimento dos quadros enviados pelo VANT. Tal busca ocorre dentro do conjunto dos *waypoints* carregados em memória.
6. Painel para controle dos movimentos do VANT (auxiliar ao *joystick*).
7. Área de controle do voo autônomo. Pode-se habilitar ou desabilitar o voo autônomo.
8. Área de controle dos parâmetros do controlador PID.
9. Informação da bateria do VANT.
10. Informações de altitude, *pitch*, *roll*, configuração de velocidade e *trim* da aeronave.



## 6.4 Escolha do algoritmo para *image matching*

O algoritmo de *image matching* é parte integrante do sistema de navegação por imagem do VANT. Na literatura, há diversos algoritmos para a extração dos *keypoints* e cálculo dos seus *descriptors* disponíveis. Alguns dos mais conhecidos são o SURF, SIFT e ORB, todos disponíveis na biblioteca OpenCV. Visando realizar uma boa escolha do algoritmo utilizado neste trabalho, fez-se uma pesquisa bibliográfica sobre o tema. Deseja-se escolher um algoritmo que seja rápido computacionalmente, invariante à rotação e escala, robusto a diferentes intensidades luminosas, composição das cores e ruídos. Durante a pesquisa, encontrou-se um artigo, referenciado em [15], que já realizou esse estudo. Aqui, então, serão replicados alguns dos dados publicados no artigo. Assim, para a escolha do algoritmo, foram analisadas as seguintes características:

1. Taxa de *matching* variando-se a intensidade e composição de cores das imagens (Tabela 5).
2. Taxa de *matching* com diferentes ângulos de rotação (Tabelas 6 e 7).
3. Taxa de *matching* com adição de ruído nas imagens (Tabela 8).

Tabela 5 – Taxa de *matching* variando-se a intensidade e composição de cores das imagens. Extraído de [15].

	<b>Tempo de processamento</b>	<b>Taxa de acerto</b>
SIFT	0,13 s	76,7 %
SURF	0,04 s	72,6 %
ORB	0,03 s	63,6 %

Tabela 6 – Desempenho para *matching* com 45°. Extraído de [15].

	<b>Tempo de processamento</b>	<b>Taxa de acerto</b>
SIFT	0,16 s	65,4 %
SURF	0,03 s	50,8 %
ORB	0,03 s	46,2 %

Tabela 7 – Taxa de *matching* com diferentes ângulos de rotação. Extraído de [15].

	<b>0°</b>	<b>45°</b>	<b>90°</b>	<b>135°</b>	<b>180°</b>	<b>225°</b>	<b>270°</b>	<b>Média</b>
SIFT	100 %	65 %	93 %	67 %	92 %	65 %	93 %	82,1 %
SURF	99 %	51 %	99 %	52 %	96 %	51 %	95 %	77,6 %
ORB	100 %	46 %	97 %	46 %	100 %	46 %	97 %	76,0 %

Analisando-se o artigo citado e as tabelas aqui replicadas, pode-se concluir os seguintes pontos:

Tabela 8 – Taxa de *matching* com adição de 30% de *salt and pepper noise*. Extraído de [15].

	Tempo de processamento	Taxa de acerto
SIFT	0,115 s	53,8 %
SURF	0,059 s	39,48 %
ORB	0,027 s	54,48 %

- Da análise das Tabelas 5, 6 e 8, pode-se concluir que o ORB e SURF possuem tempos de processamento comparáveis e ambos são mais rápidos que o SIFT. Assim, por motivo de desempenho, foi descartada a utilização do SIFT;
- Da análise das Tabela 5, 7 e 8, o SURF desempenha melhor que o ORB em duas das três análises realizadas; e
- No artigo citado, os autores concluem que os *keypoints* extraídos pelo ORB ficam concentrados em torno do objeto na cena. Enquanto que no SURF, os *keypoints* estão melhor distribuídos pela imagem. Essa melhor distribuição ajuda na identificação do *waypoint* visual, quando apenas uma parte dele está visível pela câmera.

Assim, a partir dos pontos analisados acima, concluiu-se pela utilização do método SURF neste trabalho. Porém, não há impedimento sobre a utilização também do método ORB.

# 7 EXPERIMENTOS E RESULTADOS

Durante a execução deste trabalho, foram executados diversos experimentos com níveis crescentes de complexidade a fim de se verificar a viabilidade do objetivo proposto. Primeiramente, foram realizados alguns experimentos de reconhecimento de imagens aéreas utilizando a biblioteca OpenCV. Tais experimentos validaram a ideia de que é possível utilizar o reconhecimento de imagens do solo para se localizar. Posteriormente, realizou-se um conjunto de experimentos em bancada utilizando uma *webcam* conectada ao computador e fotos aéreas impressas. Em seguida, utilizou-se um simulador em Unity para se realizar testes de voo guiado por imagens. Por fim, após a validação e o teste de todos os experimentos anteriores, foram realizados voos reais com um VANT comercial.

## 7.1 Experimentos iniciais com imagens aéreas

Nesta Seção, serão mostrados alguns experimentos iniciais utilizando a biblioteca OpenCV para a validação da hipótese de que é possível realizar a localização através de imagens aéreas do terreno. Para isso, foram utilizadas, primeiramente, imagens aéreas obtidas de satélites e, posteriormente, de VANTs em voo.

### 7.1.1 Reconhecimento de imagens aéreas obtidas por satélites

Neste experimento, objetivou-se testar o algoritmo de reconhecimento através de imagens de satélite obtidas através do software *Google Earth* [11]. Neste programa, foram capturadas duas imagens do Parque Olímpico Rio 2016 separadas temporalmente de três meses. A primeira imagem, representada na Figura 48, foi obtida em março de 2018 e a segunda, representada na Figura 49, em maio de 2018. O objetivo deste experimento foi verificar se é possível, utilizando mecanismos de visão computacional, reconhecer e comparar imagens do terreno mesmo que obtidas em datas diferentes, ou seja, com um lapso temporal entre elas.

Assim, para testar essa hipótese, foram destacados dois trechos da imagem mais antiga do Parque Olímpico e utilizou-se o código criado para buscar esses trechos na imagem mais recente. Nas Figuras 50 e 52, podem-se ver os trechos recortados da primeira imagem com os respectivos *keypoints* marcados em azul e extraídos através do SURF. Nas Figuras 51 e 53, podem-se ver a associação e o reconhecimento desses trechos na imagem mais recente do Parque Olímpico.

Em uma rápida análise, o algoritmo de reconhecimento funcionou e o resultado encontrado foi satisfatório. No entanto, para que esse método possa reconhecer as imagens,

é necessário que as características do terreno não mudem drasticamente, possuindo certo grau de similaridade.

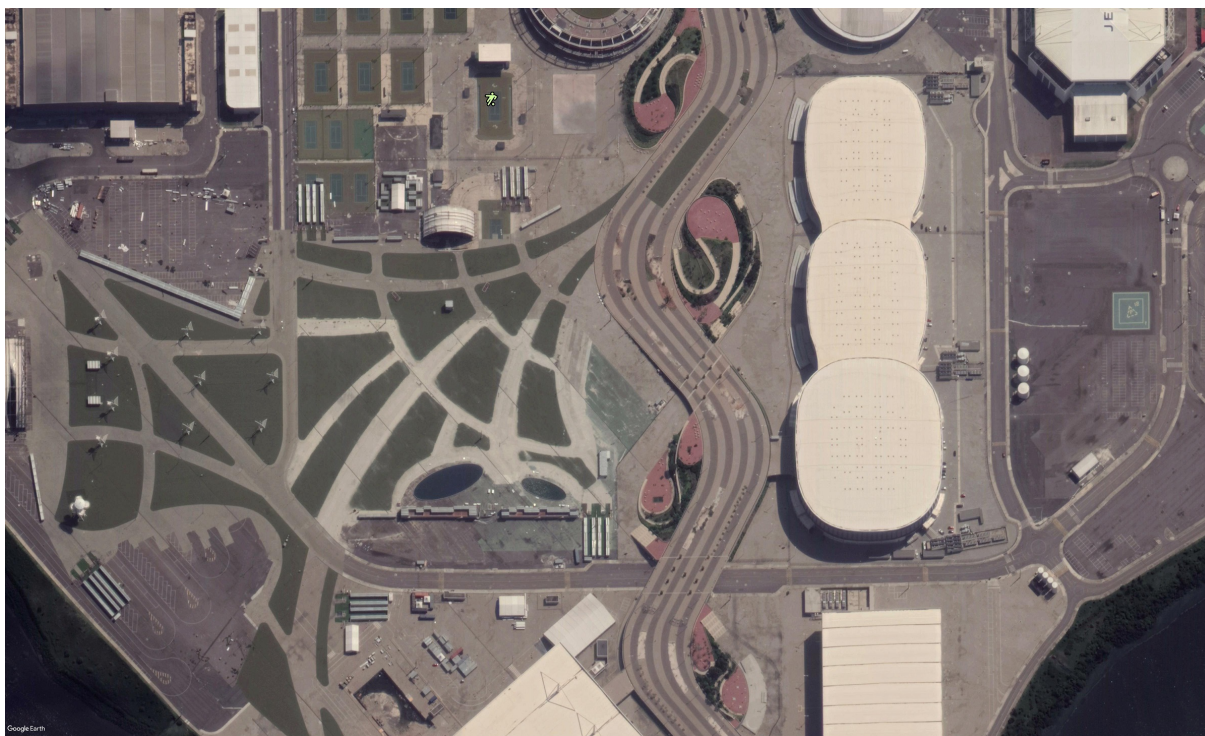


Figura 48 – Imagem de satélite do Parque Olímpico obtida em março de 2018. Fonte: *Google Earth* [11].



Figura 49 – Imagem de satélite do Parque Olímpico obtida em maio de 2018. Fonte: *Google Earth* [11].

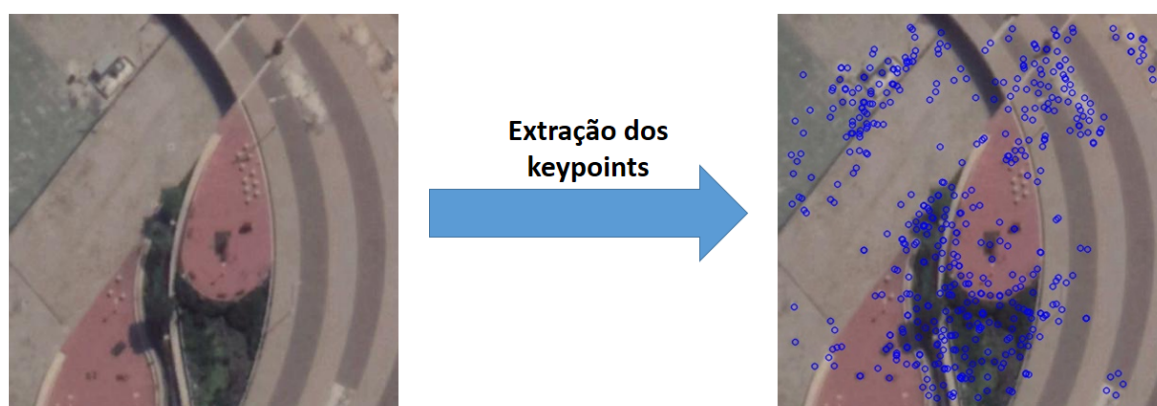


Figura 50 – Recorte de um trecho da Figura 48, de março de 2018, e extração dos seus keypoints.

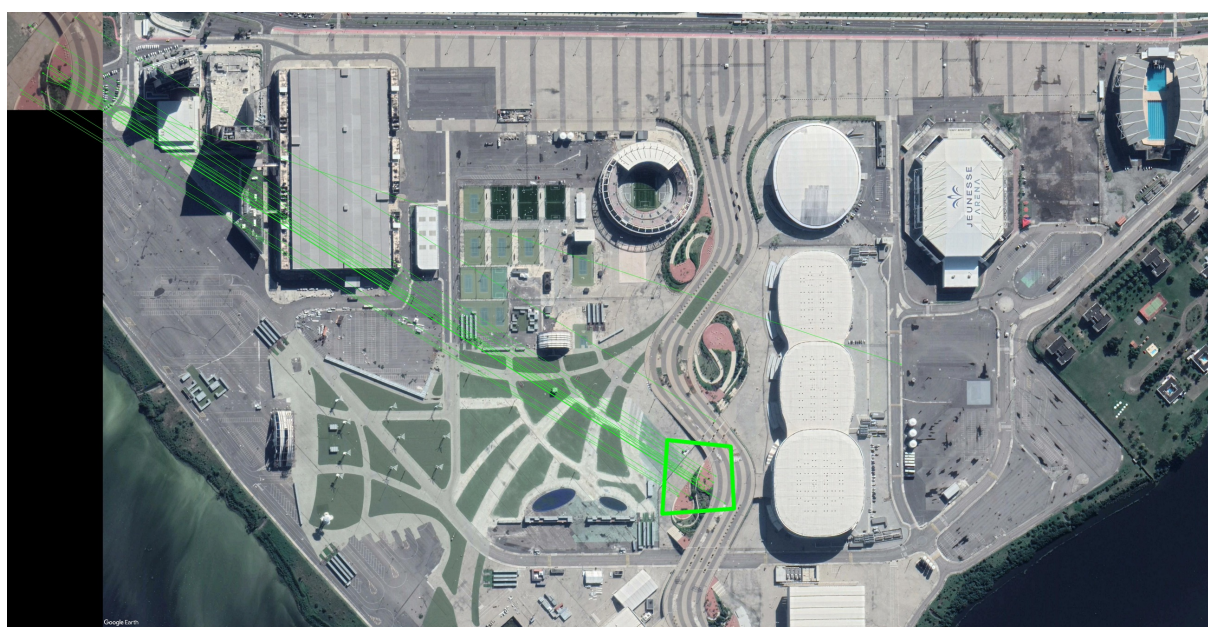


Figura 51 – Reconhecimento do trecho na Figura 50, obtida em março de 2018, na Figura 49 obtida em maio de 2018.

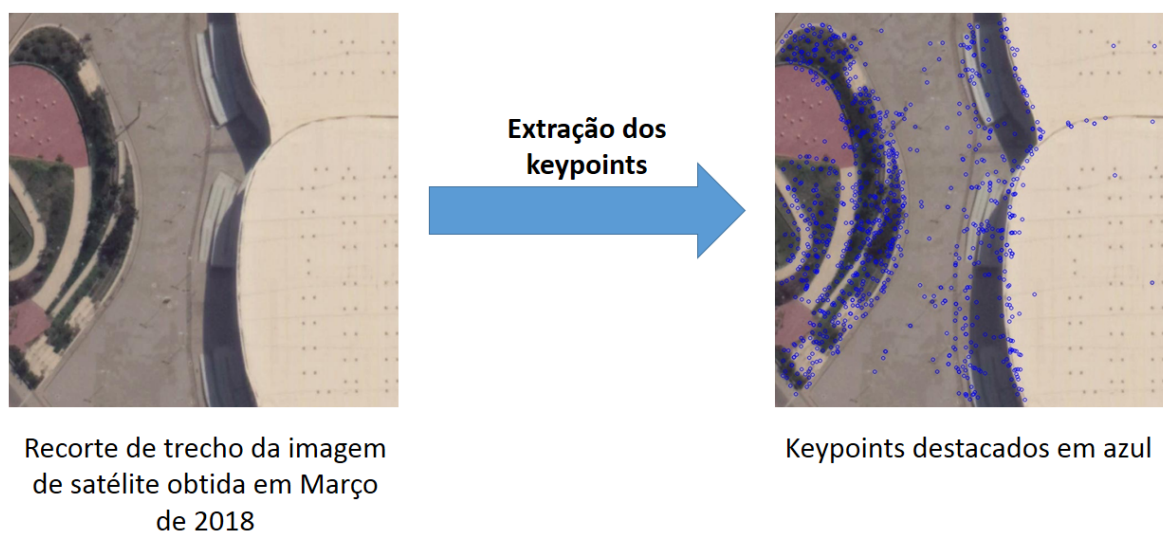


Figura 52 – Recorte de um trecho da Figura 48, de março de 2018, e extração dos seus *keypoints*.

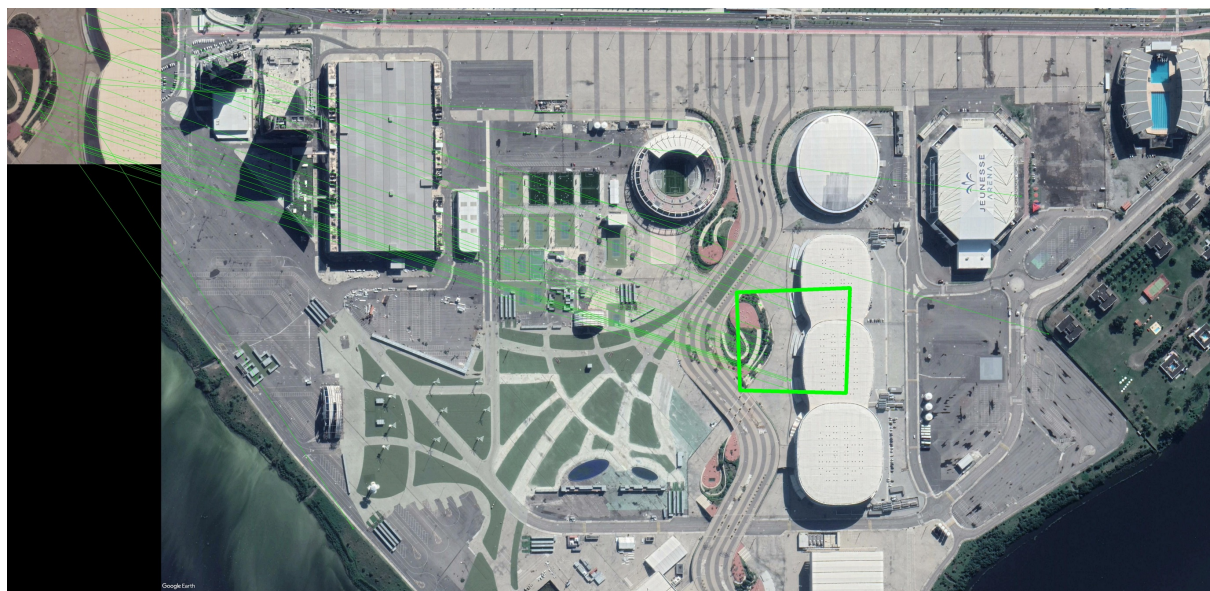


Figura 53 – Reconhecimento do trecho na Figura 52, obtida em março de 2018, na Figura 49 obtida em maio de 2018.

### 7.1.2 Reconhecimento de imagens aéreas obtidas por VANTs

Neste experimento, objetivou-se testar o algoritmo de reconhecimento do OpenCV com imagens aéreas obtidas por VANTs. Buscou-se, então, uma base de imagens na internet com as seguintes características desejadas: a) imagens capturadas durante dois voos diferentes sobre uma mesma área; b) terreno fotografado com textura ou características (*features*) que permitem a sua distinção; e c) pequeno lapso temporal entre as imagens dos dois voos realizados.

Assim, encontrou-se uma base de imagens com tais características no site [12]. Tal base é formada por dois conjuntos de imagens obtidas em voos diferentes. No primeiro voo, foram obtidas 160 imagens com resolução de 4000x3000 pixels e cada uma com aproximadamente 5MB de tamanho, totalizando 857 MB de dados. No segundo voo, foram obtidas 371 imagens com a mesma resolução e tamanho cada uma, totalizando 2,01 GB de dados. Na Figura 54, pode-se ver o mosaico construído através da concatenação de todas as imagens obtidas do segundo voo. A Figura 55 representa um subconjunto das imagens obtidas durante o primeiro voo da aeronave. A Figura 56 representa um subconjunto das obtidas no segundo voo.

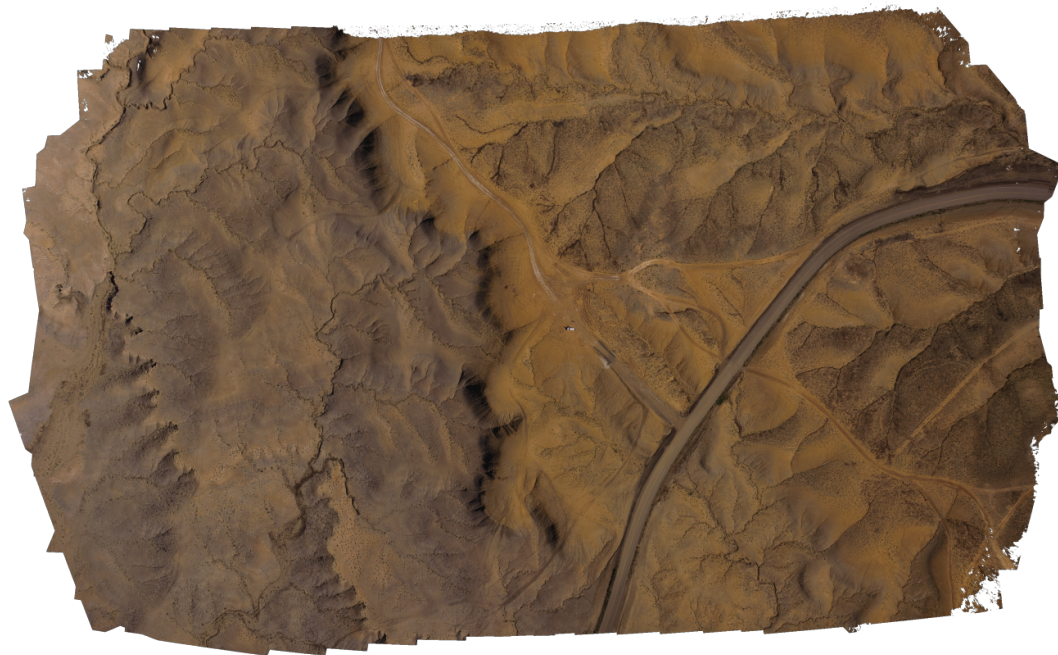


Figura 54 – Mosaico construído através da concatenação de todas as imagens obtidas durante o segundo voo do VANT sobre o terreno. Fonte: [12]

Foram realizadas diversas execuções do algoritmo, porém, nesta dissertação, serão ilustradas apenas duas. Em uma das execuções, forneceu-se, como entrada, a imagem *DJI\_0008.jpg* e configurou-se o programa para identificar, no banco de imagens do segundo voo, qual seria a mais semelhante a esta fornecida.

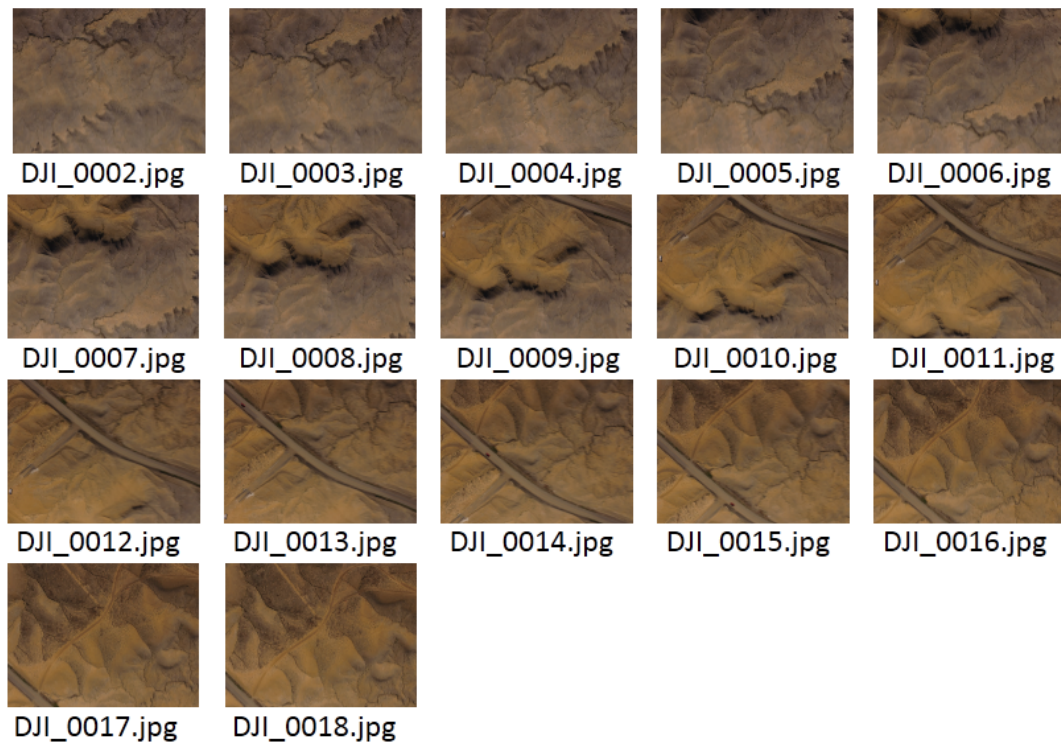


Figura 55 – Subconjunto das imagens obtidas durante o primeiro voo da aeronave sobre o terreno. Fonte: [12]

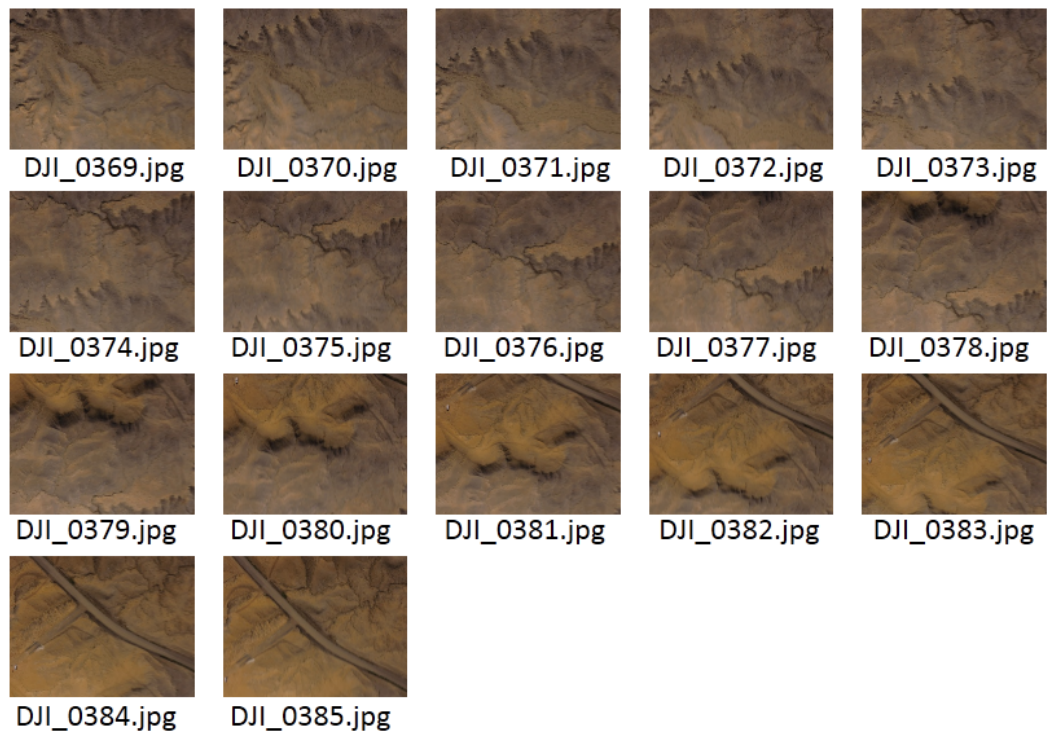


Figura 56 – Subconjunto das imagens obtidas durante o segundo voo da aeronave sobre o terreno. Fonte: [12]



Como já descrito durante este trabalho, a similaridade entre duas imagens distintas pode ser realizada comparando-se os *descriptors* dos *keypoints* extraídos da imagem. Quanto mais próximos são dois *descriptors*, mais semelhantes serão os seus respectivos *keypoints* extraídos da imagem. Assim, ainda nesse raciocínio, quanto maior o número de *keypoints* semelhantes ou próximos, maior será a semelhança das imagens analisadas.

A Figura 57 representa essa comparação. O eixo vertical expressa o número de *keypoints* semelhantes ou compatíveis. O eixo horizontal mostra as imagens utilizadas como base para a comparação com a *DJI\_0008.jpg*. Analisando-se o valor de pico do gráfico, é possível verificar que a imagem com maior semelhança a *DJI\_0008.jpg*, do primeiro voo, é a imagem *DJI\_0380.jpg*, do segundo voo. Entretanto, verifica-se que as imagens *DJI\_0379.jpg* e *DJI\_0381.jpg* também possuem um grande grau de semelhança com a *DJI\_0008.jpg*. Isso acontece pois, como há sobreposição das imagens (*overlap*), elas compartilham uma certa quantidade de *keypoints* com a *DJI\_0008.jpg*. A título de comprovação do resultado fornecido pelo algoritmo, pode-se, visualmente, verificar essa equivalência.

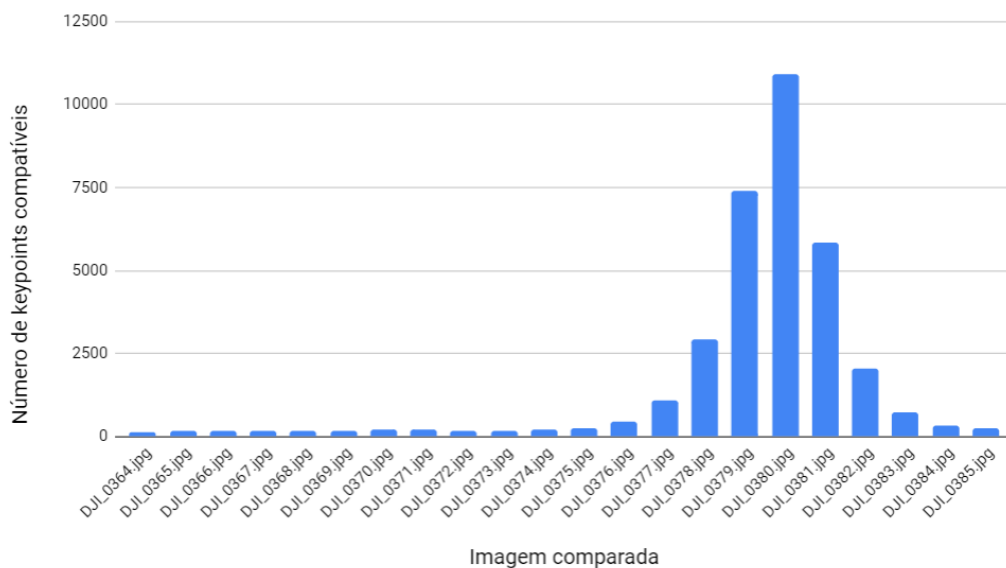


Figura 57 – Gráfico com o número de *keypoints* semelhantes ou compatíveis da imagem *DJI\_0008.jpg*, capturada no primeiro voo, com outras imagens do segundo voo.

Na Figura 58, foram destacados em azul alguns *keypoints* da imagem *DJI\_0008.jpg*. É possível verificar, visualmente, que os *keypoints* se concentram em áreas não uniformes da imagem. Na Figura 59, foram plotadas algumas das equivalências dos *keypoints* das imagens *DJI\_0008.jpg* e *DJI\_0380.jpg*.

A segunda execução do algoritmo aqui ilustrada realizou-se igualmente à primeira. Porém, neste caso, outra imagem de entrada no algoritmo foi utilizada. Nesta execução, forneceu-se como entrada a imagem *DJI\_0013.jpg*, do primeiro voo, para, então, localizar

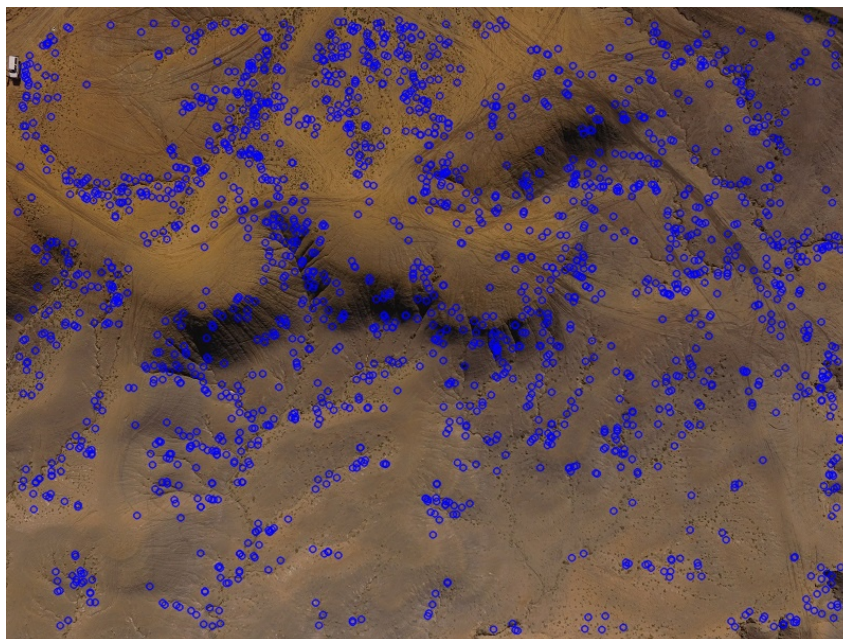


Figura 58 – Destaque em azul de alguns *keypoints* extraídos da imagem *DJI\_0008.jpg*

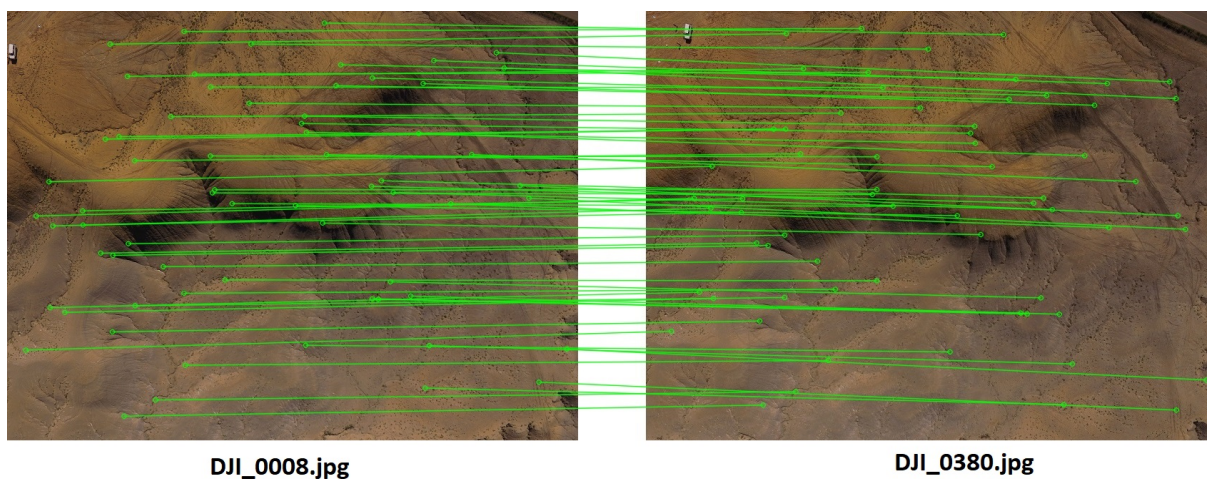


Figura 59 – Equivalência dos *keypoints* das imagens *DJI\_0008.jpg* e *DJI\_0380.jpg*, comprovando a grande semelhança entre elas.

uma semelhante no banco de imagens do segundo voo. Analisando a Figura 60, pode-se verificar que a imagem com maior semelhança a *DJI\_0013.jpg* é a imagem *DJI\_0385.jpg* do segundo voo. Novamente, verifica-se também que a imagem *DJI\_0384.jpg* possui grande semelhança com a *DJI\_0013.jpg* devido à sobreposição. Examinando a Figura 56, vê-se que *DJI\_0384.jpg* e *DJI\_0385.jpg* representam imagens de regiões muito próximas geograficamente.

Por fim, a título de ilustração, foram destacados em azul na Figura 61 alguns *keypoints* da imagem *DJI\_0013.jpg*. Inspeccionando esses pontos, é possível verificar visualmente que as bordas da estrada possuem vários *keypoints* e que, na estrada em si, há poucos pontos azuis. Isso ocorre pois, conforme já destacado, superfícies uniformes

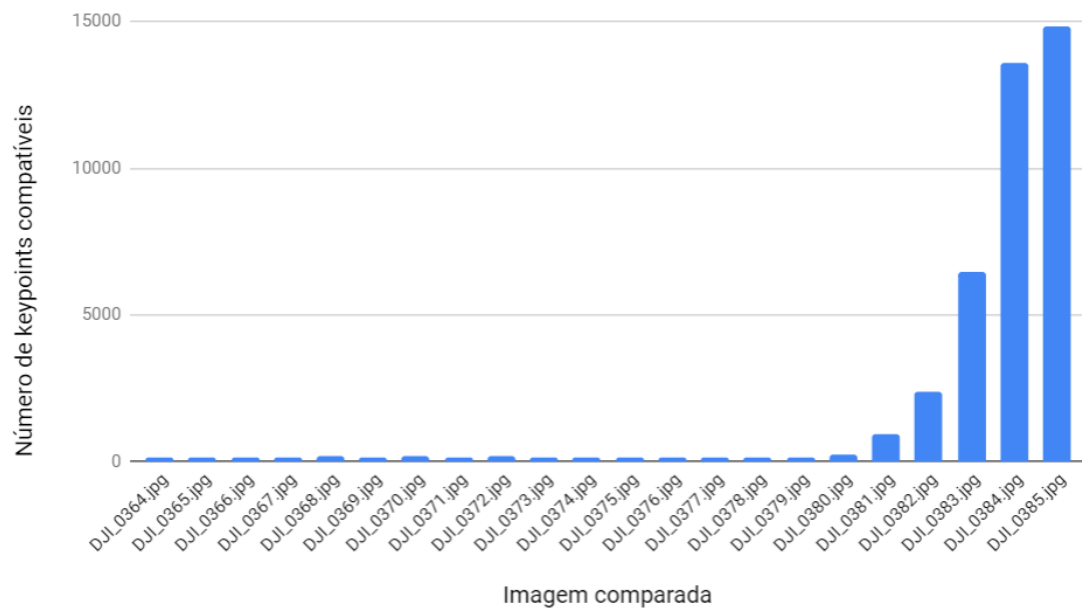


Figura 60 – Gráfico com o número de *keypoints* semelhantes ou compatíveis da imagem *DJI\_0013.jpg*, capturada no primeiro voo, com outras imagens do segundo voo.

possuem poucas características a serem extraídas. Terminando a análise, na Figura 62, foram plotadas algumas das equivalências dos *keypoints* das imagens *DJI\_0013.jpg* e *DJI\_0385.jpg*.

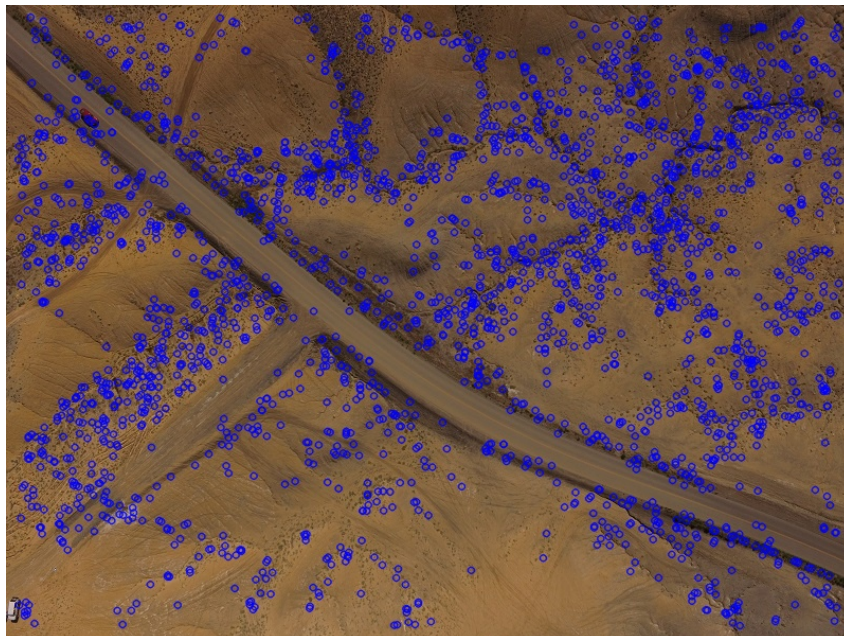


Figura 61 – Destaque em azul de alguns *keypoints* extraídos da imagens *DJI\_0013.jpg*

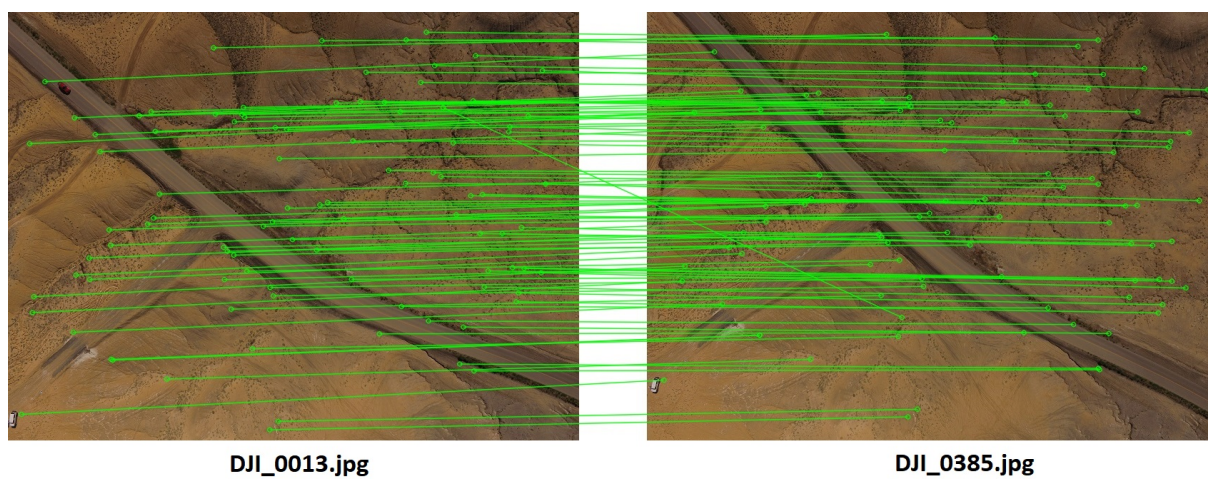


Figura 62 – Equivalência dos *keypoints* das imagens *DJI\_0013.jpg* e *DJI\_0385.jpg*, comprovando a grande semelhança entre elas.

## 7.2 Experimentos de bancada

Foram realizados dois experimentos de bancada relativos ao código de reconhecimento de imagens aéreas. Estes experimentos tiveram como objetivos os seguintes pontos: 1) Testar o código na identificação de múltiplos *waypoints* simultaneamente; 2) Verificar o desempenho na identificação, em tempo real, dos *waypoints* cadastrados; e 3) Calcular o centro e o ângulo de rotação de cada *waypoint* identificado. Para isso, foram utilizados, no *setup* dos experimentos, uma *webcam* comercial, um notebook e um mosaico de imagens aéreas impressas em escala de cinza. Para iniciar a execução dos testes, os *waypoints* visuais do trajeto foram carregados no software de navegação, o mosaico foi disposto sobre a bancada e a câmera posicionada sobre as imagens, apontando para baixo. As descrições de cada experimento serão detalhas a seguir.

### 7.2.1 Experimento de bancada 1

O primeiro experimento está ilustrado na Figura 63. Ele testou a identificação simultânea de vários *waypoints* na imagem fornecida pela câmera. Essa identificação simultânea é importante, pois será necessária para a implementação das estratégias de perseguição dos *waypoints* no mecanismo de navegação. Neste experimento em particular, não se implementou nenhum mecanismo de otimização na busca. Ou seja, o software de navegação procurava, na imagem fornecida, se havia a presença de cada um dos *waypoints* cadastrados no banco de dados. Na prática, esse mecanismo de busca linear é ineficiente e lento. Um processo de busca mais eficiente será testado no segundo experimento desta Seção.

Ainda na Figura 63, cada quadrado colorido representa um *waypoint* visual identificado. Neste experimento, foram utilizados 20 *waypoints* com resolução de 600x600 pixels e um *overlap* médio de 87 % entre eles. O *overlap* existente justifica a sobreposição dos quadrados coloridos na figura ilustrada. Por fim, o vídeo deste experimento pode ser acessado pelo link abaixo:

Vídeo 1: <<https://www.youtube.com/watch?v=iskeYtj0hcQ>>.

### 7.2.2 Experimento de bancada 2

O segundo experimento está ilustrado na Figura 64. Ele objetivou calcular o centro de cada *waypoint* e o ângulo de rotação em relação à câmera. Essas informações serão úteis para o cálculo do vetor deslocamento e do ângulo de giro necessários para o VANT alcançar e se alinhar com o *waypoint* identificado. Na Figura 64, tem-se a identificação consecutiva dos *waypoints* 4, 6, 11 e 19, destacados em vermelho, em cada uma das imagens. Estas estão dispostas em ordem temporal, conforme a câmera se desloca pelo mosaico na direção de perseguição de cada *waypoint* durante o experimento. A seta vermelha

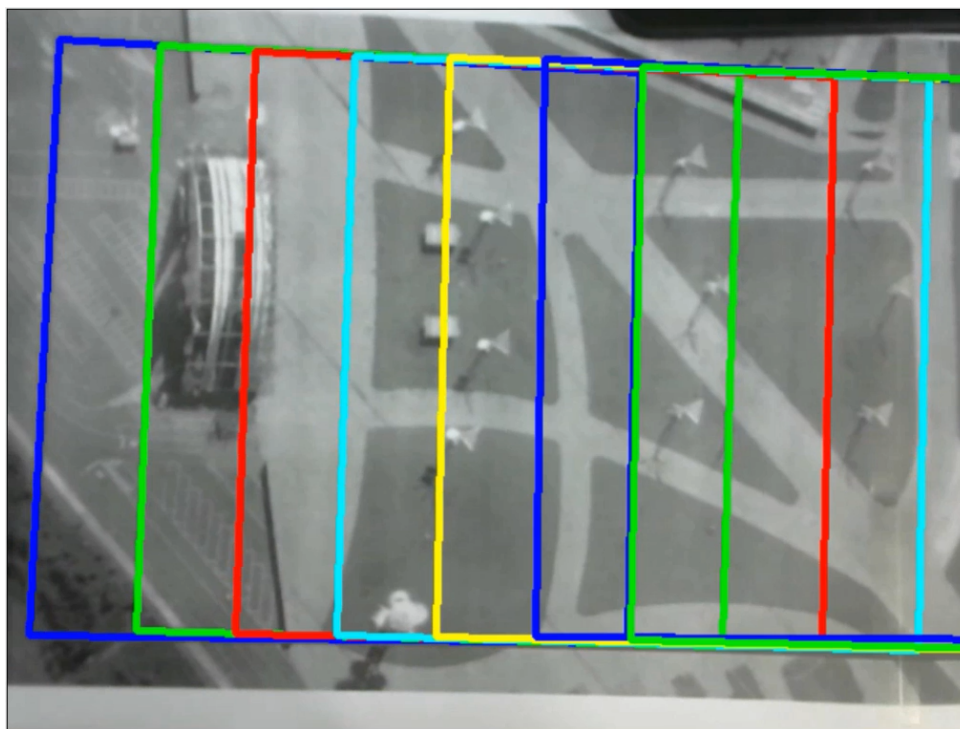


Figura 63 – Experimento de bancada 1: Identificação de múltiplos *waypoints* na imagem fornecida pela câmera.

representa o vetor deslocamento instantâneo. Ela possui como origem o centro da imagem adquirida pela câmera e, como destino, o centro do *waypoint* identificado. O valor em verde, no centro da imagem, representa o ângulo, em graus, de rotação do *waypoint* em relação à imagem capturada pela câmera. Esse ângulo será utilizado para alinhar o VANT ao *waypoint*. A seta verde representa também o vetor deslocamento, porém com algum tratamento estatístico sobre ele. São utilizados mecanismos de média móvel para suavizar os movimentos do VANT e descartados valores com grandes variações em relação ao atual. A natureza dessa filtragem decorre das próprias restrições físicas da aeronave, uma vez que, em condições normais de utilização, não há grandes alterações no deslocamento e no ângulo de *yaw* em pequenos intervalos de tempo.

Como melhoramento em relação ao primeiro experimento, neste foi implementada uma otimização na busca dos *waypoints* no banco de dados da trajetória. Utilizou-se o mecanismo de janela deslizante a fim de que não seja necessária a busca em todo o conjunto de *waypoints* cadastrados. O centro da janela de busca é o índice do último *waypoint* visual reconhecido com sucesso. O tamanho da janela é obtido utilizando-se um *range*,  $j$ , para cada lado. Ou seja, se o último *waypoint* identificado for o de índice  $i$ , o mecanismo de busca vai procurar os *waypoints* no intervalo de  $(i - j)$  a  $(i + j)$ . Esse processo deixou a busca dos *waypoints* mais rápida. Neste processo de busca, foram ainda testadas algumas resoluções de imagem da câmera, utilizando uma janela deslizante de tamanho 5 ( $j = 2$ ).

Os resultados estão expressos na Tabela 9. Por fim, o vídeo deste experimento pode ser acessado pelo link abaixo:

Vídeo 2: <<https://www.youtube.com/watch?v=HINGbXxHG7g>>.

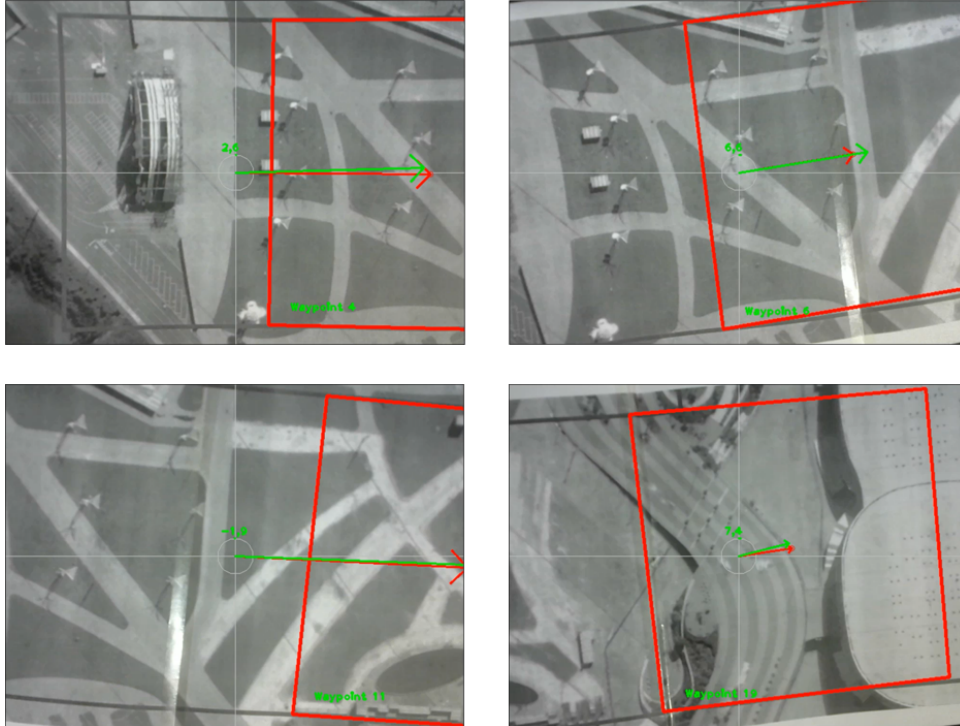


Figura 64 – Experimento de bancada 2: Identificação e navegação utilizando *waypoints* visuais. Os quadros estão dispostos temporalmente, à medida que a câmera se desloca pelo mosaico.

Tabela 9 – Tabela de desempenho na busca dos *waypoints* para diferentes resoluções da imagem.

Método	Resolução	Taxa obtida
SURF	320x240	14,6 fps
SURF	640x360	8,3 fps
SURF	1280x720	2,6 fps

### 7.3 Simulação de navegação por imagem

Nesta Seção, foi realizado um conjunto de simulações de navegação por reconhecimento de imagem do terreno visando testar e validar o software criado neste trabalho. Para isso, utilizou-se o código fonte de um simulador de VANT implementado em Unity [44]. O software Unity é uma plataforma completa que permite a criação de jogos, simulações e ambientes em três dimensões para diversos fins. Nele, é possível incorporar características físicas aos elementos da cena, como inércia, massa, gravidade, capacidade de colisão entre objetos, sombra, entre outras propriedades. O simulador replica as condições encontradas em um quadricóptero comum, onde são permitidos movimentos de translação nos 3 eixos e de rotação no eixo perpendicular ao da aeronave. Os comandos de movimento são enviados através de uma conexão TCP, semelhante ao caso do VANT real. Nesta mesma conexão, pode-se também ler dados de telemetria da aeronave, como a altitude e o ângulo de *yaw*. Além disso, o simulador também fornece a posição real da aeronave no ambiente simulado. Isso possibilita que seja feita uma comparação entre as trajetórias executadas de maneira visual, pelo VANT seguidor, e a trajetória original, feita pelo VANT líder. Como saída de vídeo, o simulador é capaz de fornecer imagens em primeira pessoa (câmera embarcada na aeronave) ou em terceira pessoa (observador externo). Um componente adicional de software, desenvolvido neste trabalho, captura essa saída de vídeo e a envia ao software de navegação implementado. Assim, fica transparente ao software de navegação se ele está conectado ao VANT real ou ao simulador.

A Figura 65 mostra uma captura de tela (*screenshot*) de uma simulação em execução. Do lado esquerdo, tem-se o software de navegação por imagem. Do lado direito, tem-se a saída de vídeo, em primeira pessoa, do simulador. A partir dessa estrutura de software descrita, foram realizadas diversas simulações utilizando as imagens do terreno, obtidas por VANTs, descritas na Subseção 7.1.2. O vídeo ilustrativo de uma simulação realizada pode ser acessado através do link abaixo:

Vídeo 3: <<https://www.youtube.com/watch?v=7ARSqkT8Ag8>>.



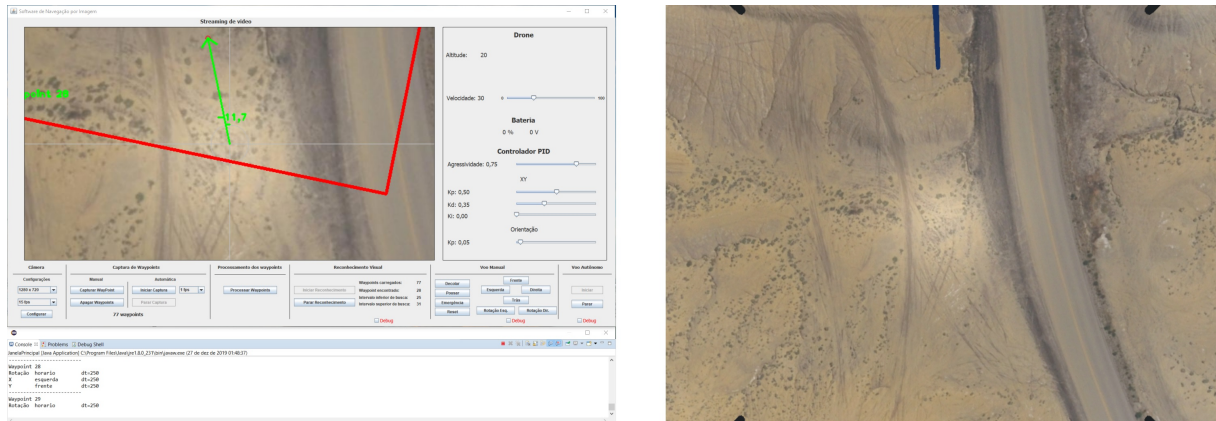


Figura 65 – Captura de tela de uma simulação em execução. Na esquerda, tem-se o software de navegação por imagem. Na direita, tem-se o simulador em Unity.

### 7.3.1 Curvas de resposta do VANT simulado

Inicialmente, para se verificar o comportamento do VANT no simulador, foram plotadas as curvas de resposta à rotação no eixo  $z$  e ao deslocamento nos eixos  $x$  e  $y$ . As rotações em  $z$  representam o controle de  $yaw$  da aeronave. Já os deslocamentos em  $x$  e  $y$  representam os movimentos de translação no plano paralelo ao solo. Analisando essas curvas geradas, podem-se conhecer as características dinâmicas da aeronave, como velocidade de deslocamento e velocidade rotacional. As curvas foram levantadas para os controladores On-Off e PID implementados. As Figuras 66, 68 e 70 representam, respectivamente, as respostas em  $x$ ,  $y$  e  $z$  utilizando-se o controlador On-Off. As Figuras 67, 69 e 71 representam também essas respostas, porém utilizando-se o controlador PID.

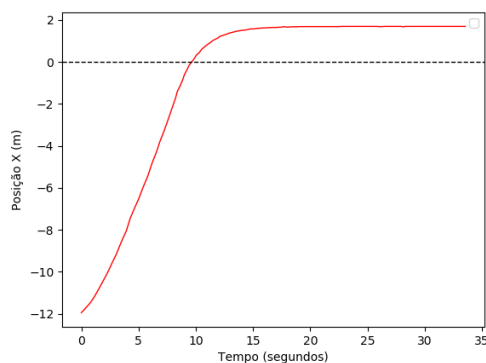


Figura 66 – Curva de resposta ao deslocamento em  $x$  utilizando o controlador On-Off.

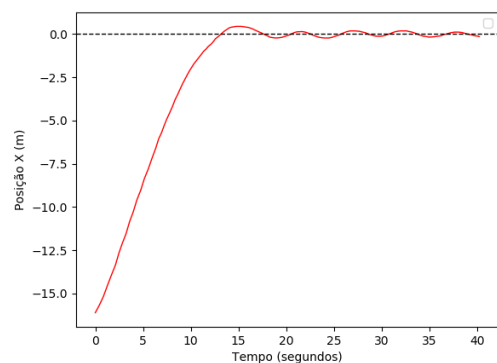


Figura 67 – Curva de resposta ao deslocamento em  $x$  utilizando o controlador PID.

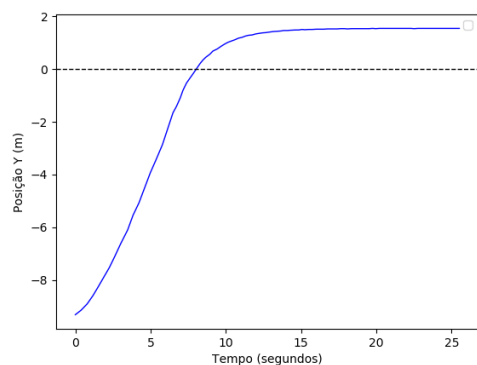


Figura 68 – Curva de resposta ao deslocamento em  $y$  utilizando o controlador On-Off.

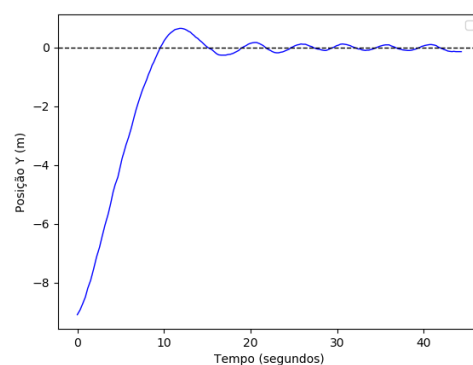


Figura 69 – Curva de resposta ao deslocamento em  $y$  utilizando o controlador PID.

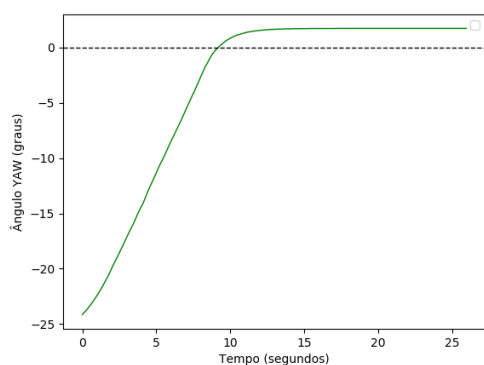


Figura 70 – Curva de resposta à rotação em  $z$  utilizando o controlador On-Off.

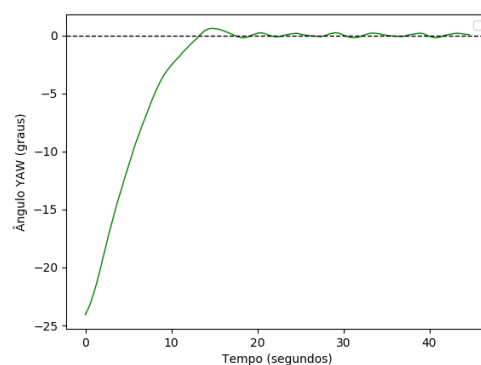


Figura 71 – Curva de resposta à rotação em  $z$  utilizando o controlador PID.

### 7.3.2 Aquisição das trajetórias pelo VANT líder

Para testar o sistema de navegação por imagem, foram adquiridas, inicialmente, quatro trajetórias pelo VANT líder. Elas foram obtidas controlando o VANT simulado por intermédio de um *joystick*. Essas trajetórias serão tomadas como referência para as execuções da navegação por imagem. Após as aquisições dos trajetos, estes foram salvos e compartilhados com o VANT seguidor. As Figuras 72, 73, 74 e 75 representam os quatro percursos salvos. Cada ponto em verde, no traçado do gráfico, representa um *waypoint* visual adquirido no caminho percorrido. Os *waypoints* são obtidos como imagens em escala cinza na resolução de 640x360 pixels. Essa resolução foi escolhida, porque é a da câmera vertical do VANT que será utilizado nos experimentos de voo real. Os eixos  $x$  e  $y$  do gráfico representam a posição do VANT, em metros, no ambiente simulado. Esses trajetos foram escolhidos, pois possuem retas, curvas e mudança de rumo com ângulos reto e agudo. Assim, visa-se medir o comportamento e o desempenho das estratégias de navegação implementadas.

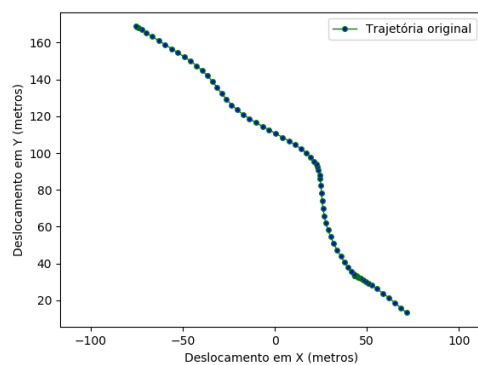


Figura 72 – Trajeto salvo 1.

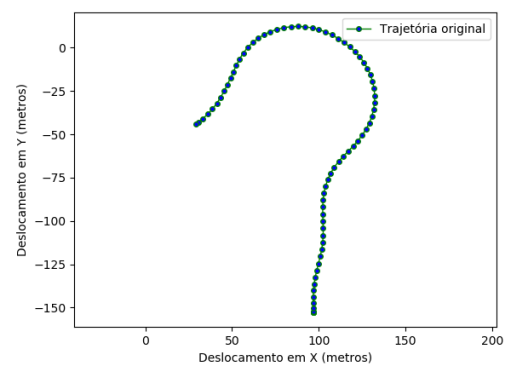


Figura 73 – Trajeto salvo 2.

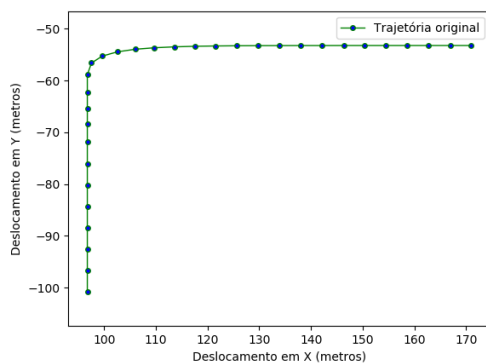


Figura 74 – Trajeto salvo 3.

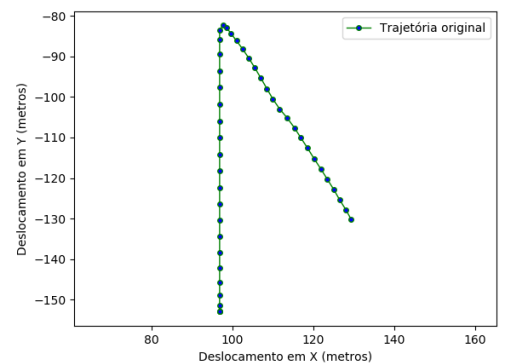


Figura 75 – Trajeto salvo 4.

As informações de cada trajeto podem ser encontradas na Tabela 10. Por exemplo, o trajeto 1 possui 74 *waypoints* visuais, 224,2 metros de extensão, uma média de 87,9 %

de *overlap* entre os *waypoints* e necessita de 1206,2 Kb para seu armazenamento.

Tabela 10 – Dados dos trajetos 1, 2, 3 e 4 adquiridos pelo VANT líder.

	Trajeto 1	Trajeto 2	Trajeto 3	Trajeto 4
Quantidade de <i>waypoints</i>	74	77	32	44
Comprimento do trajeto	224,2 m	286,6 m	118,2 m	128,5 m
Armazenamento do trajeto	1206,2 Kb	1255,1 Kb	521,6 Kb	717,2 Kb
<i>Overlap</i> entre <i>waypoints</i>	87,9 %	85,2 %	85,3 %	88,4 %

### 7.3.3 Execução das Trajetórias pelo VANT seguidor

Após a aquisição e o processamento das quatro trajetórias pelo VANT líder, os dados resultantes foram carregados na aeronave seguidora simulada que, então, realizará novamente os percursos. Para cada uma das trajetórias, foram simulados quatro diferentes configurações de piloto automático. O que resulta na combinação de um tipo de controlador, On-Off ou PID, com uma estratégia de perseguição dos *waypoints*, maior índice visível pela câmera ou visitação de todos os *waypoints*. Assim, as configurações de piloto automático testados estão expressas abaixo.

- **Configuração 1:** Piloto automático com controlador On-Off e estratégia de perseguição ao *waypoint* visível de maior índice.
- **Configuração 2:** Piloto automático com controlador On-Off e estratégia de visitação de todos os *waypoints*.
- **Configuração 3:** Piloto automático com controlador PID e estratégia de perseguição ao *waypoint* visível de maior índice.
- **Configuração 4:** Piloto automático com controlador PID e estratégia de visitação de todos os *waypoints*.

Da combinação das quatro trajetórias salvas com as quatro configurações de piloto automático, resultaram 16 trajetos realizados pelo VANT seguidor. Os resultados dessas execuções serão exibidos e analisados a seguir.

### 7.3.3.1 Simulação da configuração 1: Piloto automático com controlador On-Off e estratégia de perseguição ao *waypoint* visível de maior índice.

Nesta simulação, foi utilizado o piloto automático com o controlador On-Off e a estratégia de perseguição ao *waypoint* visível de maior índice. Além disso, foram utilizados, na configuração do controlador, os valores limites de acionamento de 2 metros de erro linear para os eixos  $x$  e  $y$  e de 3 graus de erro angular para o  $yaw$ . Ou seja, caso as medições na posição ou na orientação do VANT em relação ao *waypoint* estejam acima desses limites, comandos de movimento com duração de 250 ms são enviados à aeronave. Caso estejam dentro desse limite, o piloto automático considera que a aeronave está voando sobre o *waypoint*. As Figuras 76, 77, 78 e 79 traçam as trajetórias. Em verde, está representada a trajetória original adquirida pelo VANT líder. Em vermelho, está representada a trajetória executada por navegação visual pelo VANT seguidor.

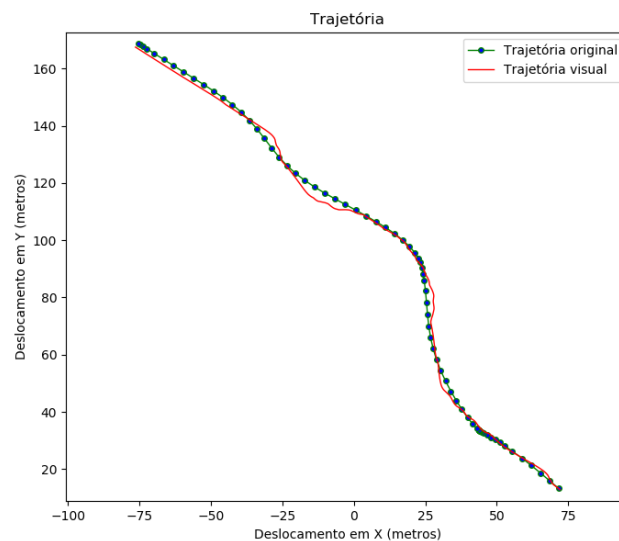


Figura 76 – Trajeto 1 realizado utilizando a configuração 1.

Com o objetivo de medir o grau de exatidão na execução da trajetória visual, foi calculada a raiz do erro médio quadrático (*Root Mean Square Error* - RMSE) entre a trajetória original e a refeita por visão computacional. Como de conhecimento, números baixos de RMSE indicam grande grau de semelhança entre os trajetos. Já números altos de RMSE indicam grande discrepância. Além disso, mediu-se, também, o número de comandos de movimento enviados ao VANT para a execução da trajetória utilizando a navegação visual. Essas informações estão apresentadas na Tabela 11.

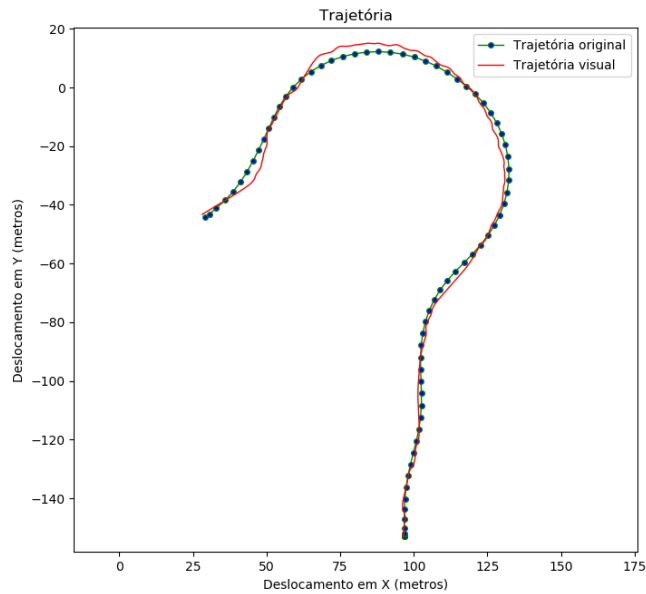


Figura 77 – Trajeto 2 realizado utilizando a configuração 1.

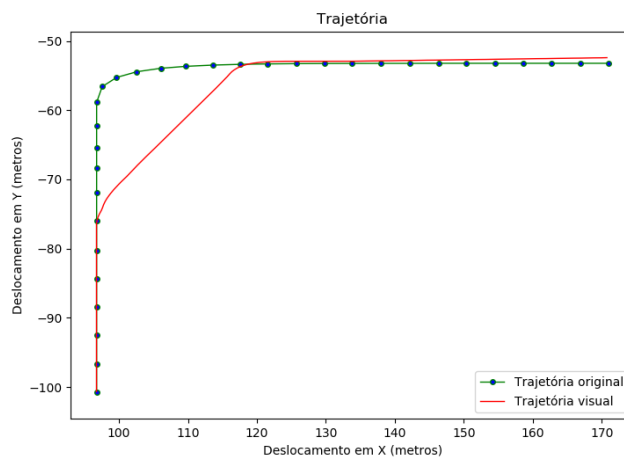


Figura 78 – Trajeto 3 realizado utilizando a configuração 1.

Por fim, após uma análise dos trajetos realizados nas Figuras 78 e 79, percebe-se que os *waypoints* presentes nas mudanças abruptas de rumo não são alcançados pelo VANT. Isso ocorre, porque a aeronave começa a visualizar *waypoints* de índices superiores, rejeitando os de índices inferiores, quando ela se aproxima dessa parte da trajetória. Este fenômeno afeta a estratégia de perseguição ao *waypoint* visível de maior índice.

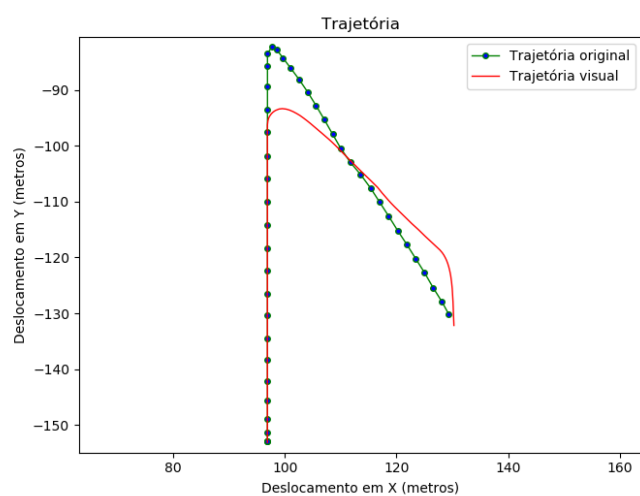


Figura 79 – Trajeto 4 realizado utilizando a configuração 1.

Tabela 11 – Informações dos trajetos 1, 2, 3 e 4 realizados utilizando a configuração 1.

	<b>Trajeto 1</b>	<b>Trajeto 2</b>	<b>Trajeto 3</b>	<b>Trajeto 4</b>
RMSE	1,3135 m	1,4655 m	4,6385 m	4,0064 m
Comandos enviados ao VANT	574	730	241	351

### 7.3.3.2 Simulação da configuração 2: Piloto automático com controlador On-Off e estratégia de visitação de todos os waypoints.

Nesta simulação, foi utilizado o piloto automático com o controlador On-Off e a estratégia de visitação de todos os waypoints. Foram configurados, como critérios de visitação do *waypoint* e limite para acionamento do controlador On-Off, os valores de 2 metros de erro linear, para os eixos  $x$  e  $y$ , e 3 graus de erro angular, para o  $yaw$ . Novamente, caso o VANT esteja a menos de 2 metros e com diferença de alinhamento menor que 3 graus, o *waypoint* será considerado visitado. Caso os valores lidos estejam acima desses limites, comandos de movimento com duração de 250 ms são enviados à aeronave. As Figuras 80, 81, 82 e 83 traçam as trajetórias. Em verde, está representada a trajetória original adquirida pelo VANT líder. Em vermelho, está representada a trajetória executada por navegação visual pelo VANT seguidor.

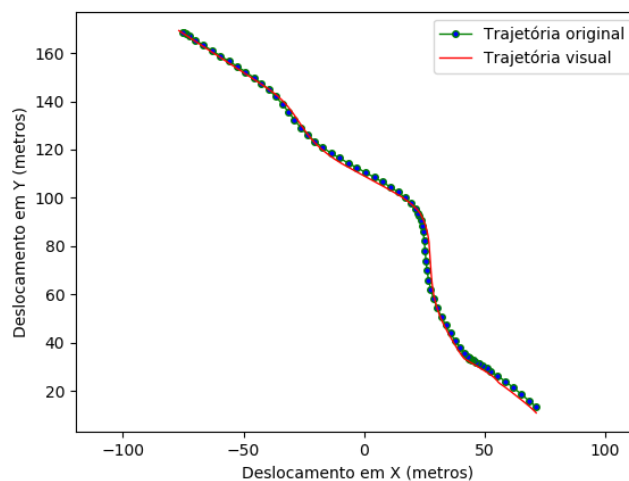


Figura 80 – Trajeto 1 realizado utilizando a configuração 2.

Por fim, a Tabela 12 fornece alguns dados sobre as execuções das trajetórias. Após uma análise da tabela, verificam-se que os trajetos sinuosos resultam em RMSE maior que os trajetos com retas.

Tabela 12 – Informações dos trajetos 1, 2, 3 e 4 realizados utilizando a configuração 2.

	<b>Trajeto 1</b>	<b>Trajeto 2</b>	<b>Trajeto 3</b>	<b>Trajeto 4</b>
RMSE	1,1476 m	1,0899 m	0,4366 m	0,3263 m
Comandos enviados ao VANT	582	526	189	236



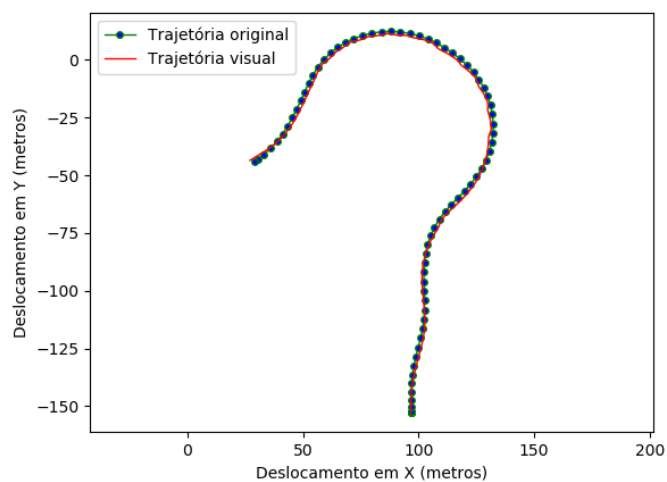


Figura 81 – Trajeto 2 realizado utilizando a configuração 2.

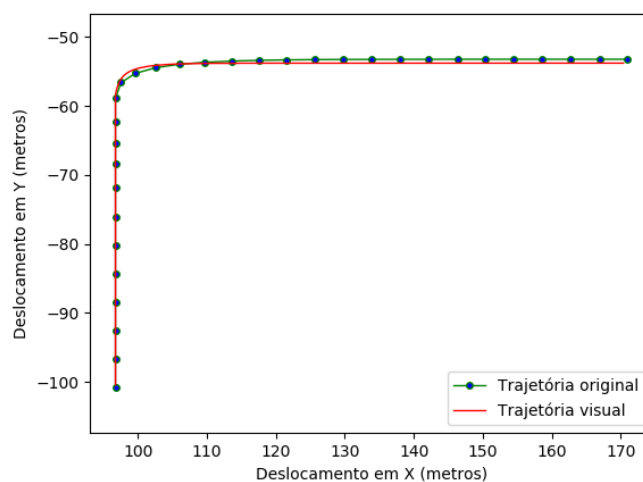


Figura 82 – Trajeto 3 realizado utilizando a configuração 2.

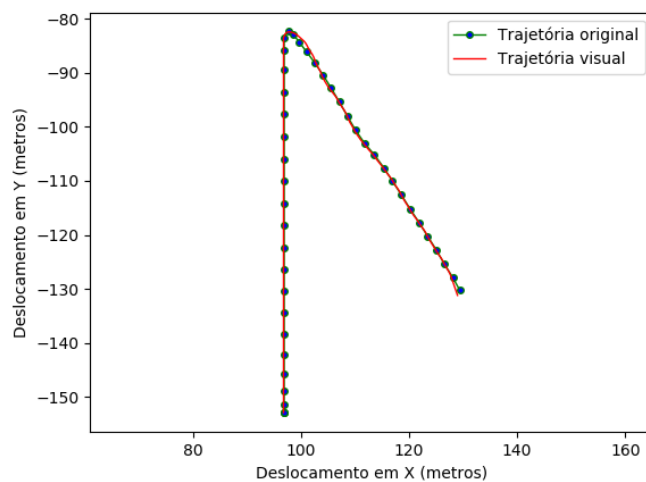


Figura 83 – Trajeto 4 realizado utilizando a configuração 2.

### 7.3.3.3 Simulação da configuração 3: Piloto automático com controlador PID e estratégia de perseguição ao *waypoint* visível de maior índice.

Nesta simulação, foi utilizado o piloto automático com controlador PID e a estratégia de perseguição ao *waypoint* visível de maior índice. Diferentemente das duas primeiras simulações, esta utilizou um controlador PID para regular o envio dos comandos ao VANT. Sabendo-se que o comportamento da aeronave simulada é igual nos eixos  $x$  e  $y$ , os respectivos ganhos  $Kp$ ,  $Ki$  e  $Kd$  também são iguais para os dois eixos, ou seja,  $Kpx = Kpy$ ,  $Kix = Kiy$  e  $Kdx = Kdy$ . Assim, de maneira experimental, foram encontrados os valores 0,07 para  $Kpx$  e  $Kpy$ , 0 para  $Kix$  e  $Kiy$  e 0,035 para  $Kdx$  e  $Kdy$ . Já no controle da rotação da aeronave, foram utilizados os valores de 10 para  $Kpr$  e 0 para  $Kir$  e  $Kdr$ , também experimentais. A estratégia utilizada foi a de perseguição ao *waypoint* visível de maior índice, já explicada anteriormente. As Figuras 84, 85, 86 e 87 traçam as trajetórias. Em verde, está representada a trajetória original adquirida pelo VANT líder. Em vermelho, está representada a trajetória executado por navegação visual pelo VANT seguidor.

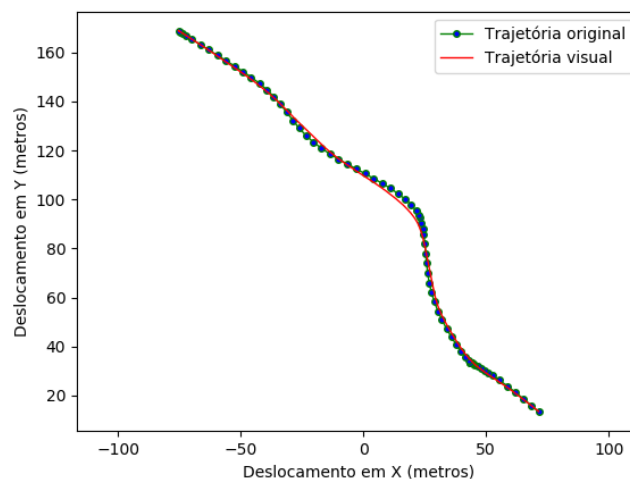


Figura 84 – Trajeto 1 realizado utilizando a configuração 3.

Novamente, conforme ocorreu nas duas últimas simulações da configuração 1, o VANT não alcança os *waypoints* próximos às mudanças abruptas de direção (*heading*). Nessa estratégia, a aeronave já começa a guinar antes de chegar na borda da trajetória. Porém, neste caso, como o controlador utilizado é do tipo PID, a mudança na direção do VANT ocorre de maneira mais suave. Tais fenômenos podem ser verificados nas Figuras 86 e 87. Por fim, foram calculados a raiz do erro médio quadrático e o número de comandos enviados ao VANT para sua execução. Essas informações podem ser vistas na Tabela 13. Confrontando os dados da Tabela 13 com os da Tabela 11 (resultados das simulações com a mesma estratégia de perseguição dos *waypoints*, porém com diferentes controladores),

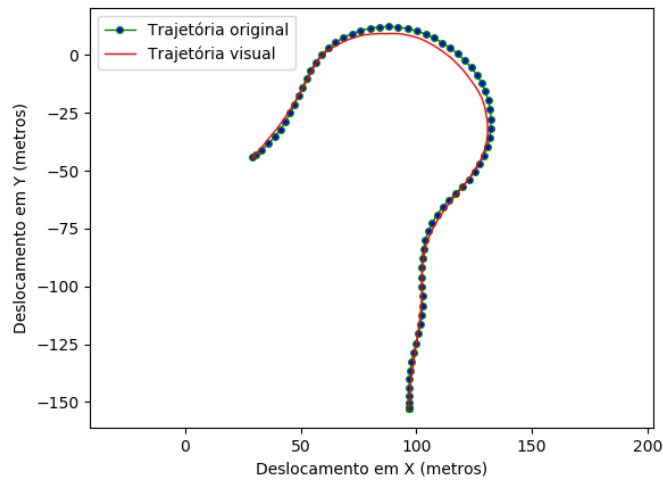


Figura 85 – Trajeto 2 realizado utilizando a configuração 3.

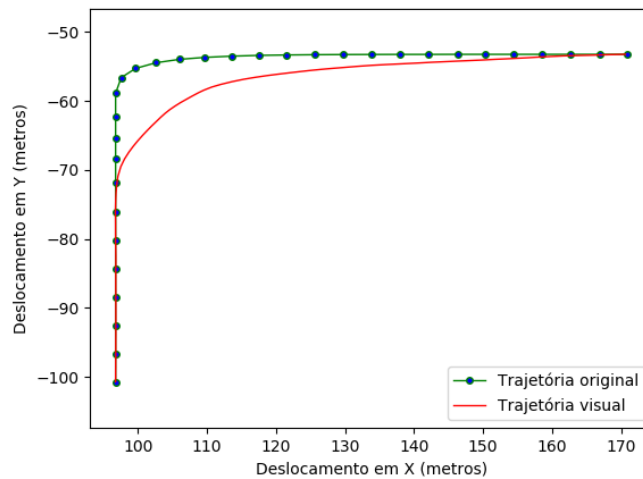


Figura 86 – Trajeto 3 realizado utilizando a configuração 3.

percebe-se que o controlador PID possui menor RMSE, resultando em maior precisão na execução do trajeto.

Tabela 13 – Informações dos trajetos 1, 2, 3 e 4 realizados a configuração 3.

	Trajeto 1	Trajeto 2	Trajeto 3	Trajeto 4
RMSE	1,0257m	1,6193m	3,3287m	3,2134m
Comandos enviados ao VANT	286	436	84	225

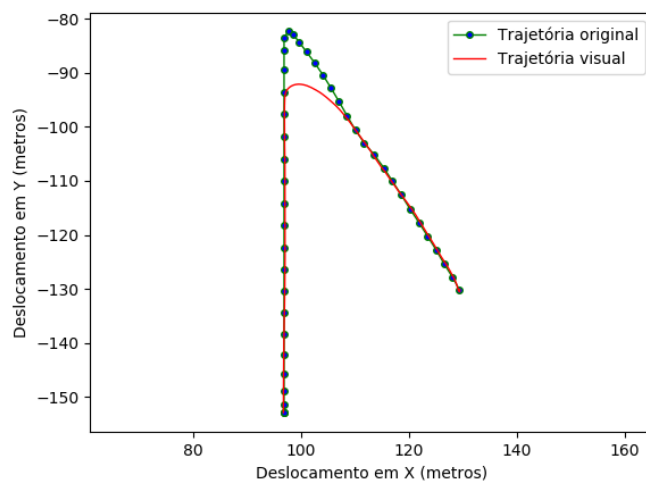


Figura 87 – Trajeto 4 realizado utilizando a configuração 3.

### 7.3.3.4 Simulação da configuração 4: Piloto automático com controlador PID e a estratégia de visitação de todos os *waypoints*.

Nesta última simulação, foi utilizado o piloto automático com o controlador PID e a estratégia de visitação de todos os *waypoints*. Os valores dos ganhos e limiares de visitação dos *waypoints* são os mesmos descritos nas simulações anteriores. As Figuras 88, 89, 90 e 91 ilustram as trajetórias executadas.

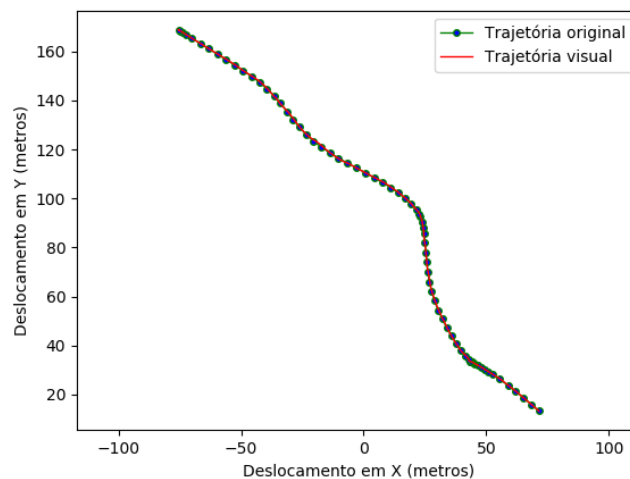


Figura 88 – Trajeto 1 realizado utilizando a configuração 4.

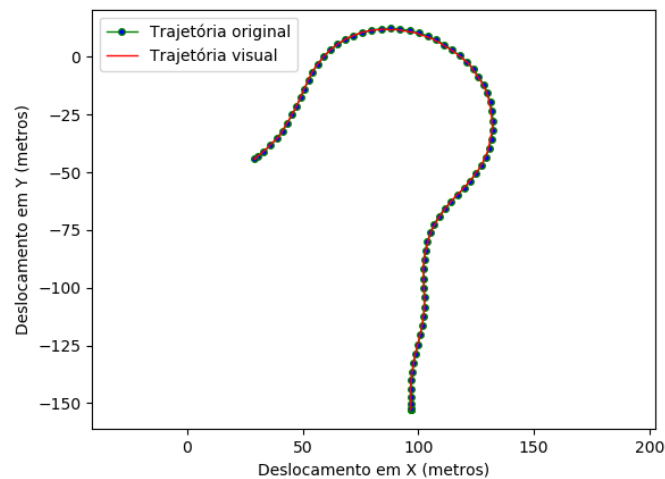


Figura 89 – Trajeto 2 realizado utilizando a configuração 4.

Por fim, a Tabela 14 mostra o RMSE e o número de comandos enviados ao VANT para a execução dos trajetos. Com base nas simulações realizadas e nos respectivos parâmetros utilizados, pode-se concluir que: 1) Os controladores PID apresentam um

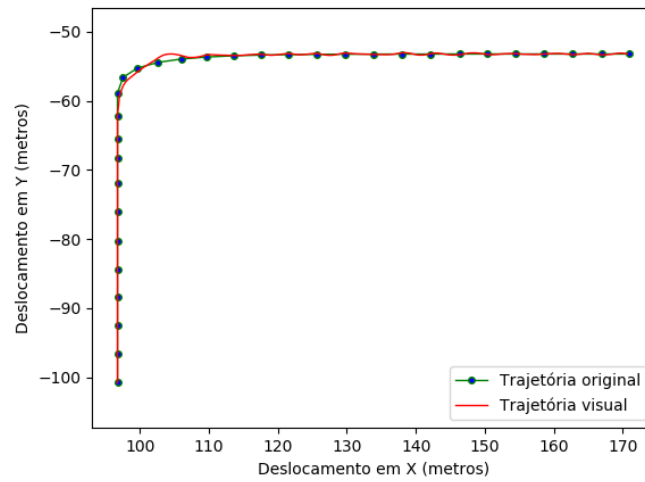


Figura 90 – Trajeto 3 realizado utilizando a configuração 4.

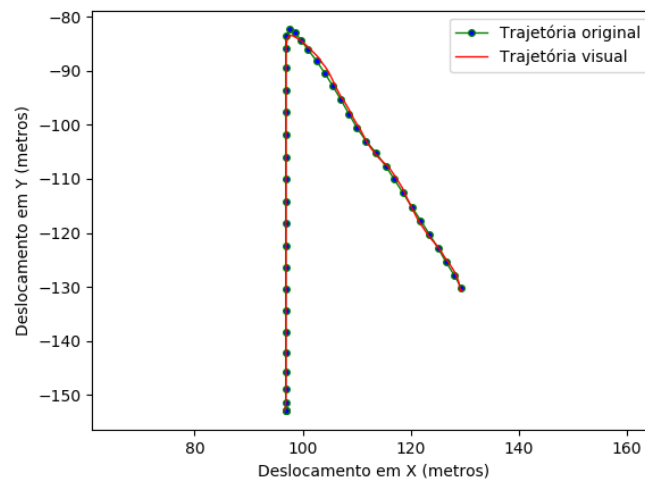


Figura 91 – Trajeto 4 realizado utilizando a configuração 4.

RMSE menor que os controladores On-Off; e 2) A estratégia de visitação de todos os *waypoints* apresenta um RMSE menor que a de perseguição do *waypoint* de maior índice visível. Desta maneira, a conjugação do controlador PID e da estratégia de visitação de todos os *waypoints* (configuração 4) apresentou os menores RMSE, como se pode constatar na Tabela 14. Tal configuração resultou em uma melhor precisão na execução da trajetória pretendida.

Tabela 14 – Informações dos trajetos 1, 2, 3 e 4 realizados a configuração 4.

	<b>Trajeto 1</b>	<b>Trajeto 2</b>	<b>Trajeto 3</b>	<b>Trajeto 4</b>
RMSE	0,1619m	0,2289m	0,2472m	0,3352m
Comandos enviados ao VANT	1164	1080	424	523



## 7.4 Experimentos de voo real

Nesta Seção, serão descritos os experimentos de voo real realizados. Para isso, foram utilizados a aeronave AR.Drone 2.0 do fabricante Parrot e o software implementado conforme descritos no Capítulo 6. Os experimentos de voo real foram divididos em duas partes. A primeira parte refere-se à capacidade do sistema de perseguir um único *waypoint* visual, alinhando e centralizando a aeronave sobre ele. A segunda parte refere-se à execução de um pequeno trajeto utilizando navegação visual. Tais experimentos serão detalhados a seguir.

### 7.4.1 Experimento de voo para alinhamento e centralização da aeronave sobre o *waypoint*.

Este experimento de voo real teve como objetivo perseguir um único *waypoint* visual, alinhando e centralizando a aeronave. Tal experimento é importante, pois estas atividades podem ser consideradas etapas básicas para a execução do trajeto por imagem. Isso porque a execução de um trajeto completo, por imagem, poderá ser realizada através do encadeamento de sucessivas perseguições dos *waypoints* do trajeto. A Figura 92 mostra o *setup* do experimento realizado. A Figura 93 mostra a aeronave pousada sobre o *waypoint* visual carregado no sistema. A Figura 94 mostra a execução do software de navegação por imagem. Os vídeos deste experimento podem ser acessados a partir dos seguintes links:

Vídeo 4: <<https://www.youtube.com/watch?v=Bpx9OB-pwE0>>;

Vídeo 5: <[https://www.youtube.com/watch?v=-wb06Epj\\_j0](https://www.youtube.com/watch?v=-wb06Epj_j0)>; e

Vídeo 6: <<https://www.youtube.com/watch?v=37IBC2ETMcM>>.

#### 7.4.1.1 Resultados

O software de navegação implementado obteve bons resultados na tarefa pretendida. Para analisar a resposta do sistema, foram plotados quatro gráficos de alinhamento e quatro de posicionamento da aeronave sobre o *waypoint*, totalizando 8 experimentos realizados. Iniciando pelo posicionamento, utilizou-se o controlador On-Off com um limiar de acionamento de 0,07 m para os eixos  $x$  e  $y$ , sendo 0,9 m a altura de voo da aeronave. Os gráficos de posicionamento relativo do VANT estão representados nas Figuras 95, 96, 97 e 98. Cada ponto verde indica uma posição em um dado instante do experimento. O centro do *waypoint* visual encontra-se na posição  $(0, 0)$ , identificado pela interseção das retas tracejadas. A Tabela 15 mostra alguns dados coletados, como os valores de erro médio e raiz do erro quadrático médio para os eixos  $x$  e  $y$ .

Para os testes de alinhamento, a aeronave foi rotacionada manualmente, por meio do *joystick*, a 90 graus em relação ao *waypoint* visual. Então, habilitou-se o piloto automático



Figura 92 – *Setup* do experimento de alinhamento e centralização sobre o *waypoint*.



Figura 93 – Aeronave pousada sobre o *waypoint* visual carregado no sistema.

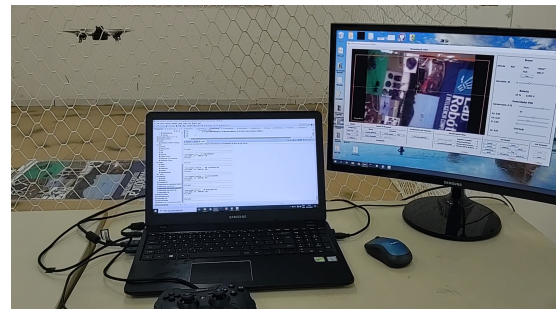


Figura 94 – Execução do software de navegação por reconhecimento de imagem.

Tabela 15 – Dados dos testes de centralização do VANT no *waypoint* visual.

	Teste 1	Teste 2	Teste 3	Teste 4
Tamanho da amostra	216	106	136	107
Erro médio em $x$	-0,014 m	-0,031 m	-0,027 m	0,018 m
Erro médio em $y$	0,025 m	-0,030 m	-0,043 m	-0,06 1m
RMSE em $x$	0,108 m	0,163 m	0,114 m	0,168m
RMSE em $y$	0,072 m	0,098 m	0,095 m	0,127m

no software de navegação visual. Com o seu funcionamento, o sistema foi gradualmente corrigindo o ângulo de *yaw* para alinhar a aeronave ao *waypoint* carregado. As Figuras 99, 100, 101 e 102 representam esse ângulo no tempo, considerando zero grau o alinhamento

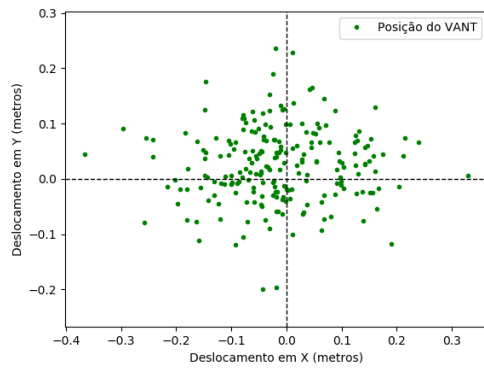


Figura 95 – Posição relativa do VANT sobre o *waypoint* (Teste 1).

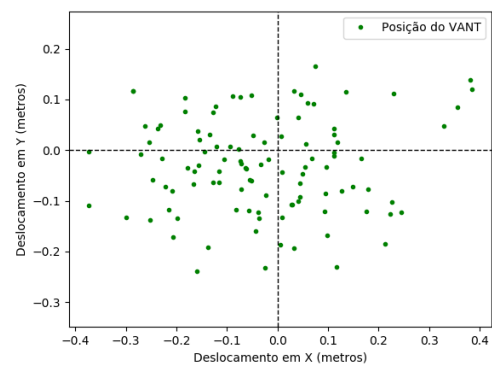


Figura 96 – Posição relativa do VANT sobre o *waypoint* (Teste 2).

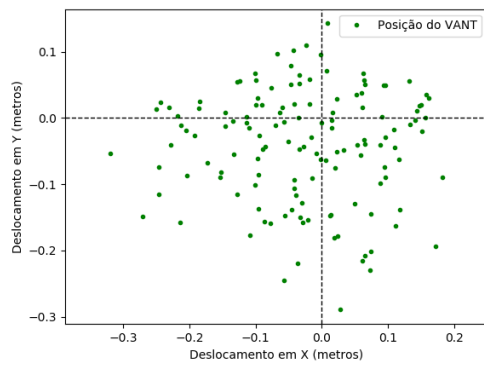


Figura 97 – Posição relativa do VANT sobre o *waypoint* (Teste 3).

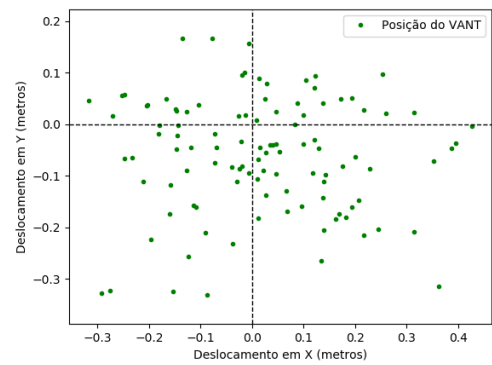


Figura 98 – Posição relativa do VANT sobre o *waypoint* (Teste 4).

perfeito. Neste experimento, utilizou-se o mesmo controlador On-Off com um limiar de acionamento de 8 graus para o *yaw*, valor no qual, quando alcançado, se considera a aeronave alinhada com o *waypoint*. A Tabela 16 apresenta alguns dados resultantes desse processo.

Tabela 16 – Alinhamento do VANT com o *waypoint* visual.

	<b>Teste 5</b>	<b>Teste 6</b>	<b>Teste 7</b>	<b>Teste 8</b>
Tempo para alinhamento	33,95 s	25,84 s	32,85 s	31,24 s
Comandos de rotação enviados	25	17	23	21
Comandos de translação enviados	25	17	23	18
Velocidade angular média	2,47 °/s	3,29 °/s	2,75 °/s	2,76 °/s

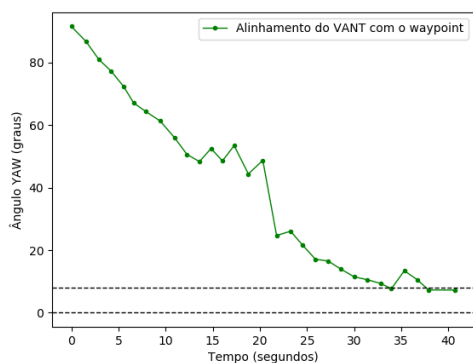


Figura 99 – Alinhamento do VANT com o *waypoint* visual (Teste 5).

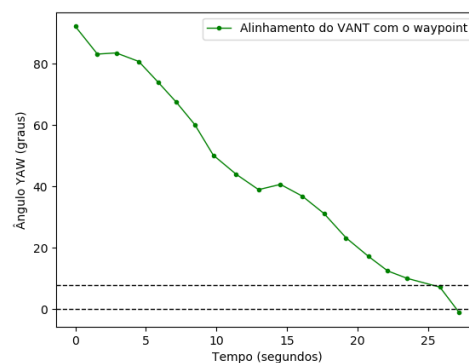


Figura 100 – Alinhamento do VANT com o *waypoint* visual (Teste 6).

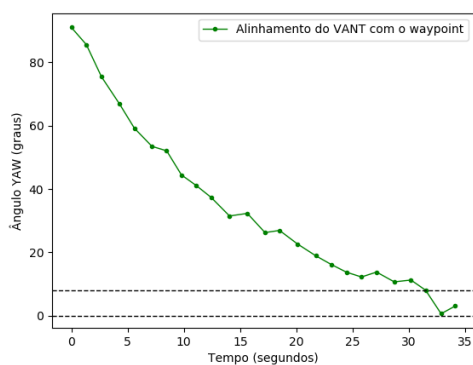


Figura 101 – Alinhamento do VANT com o *waypoint* visual (Teste 7).

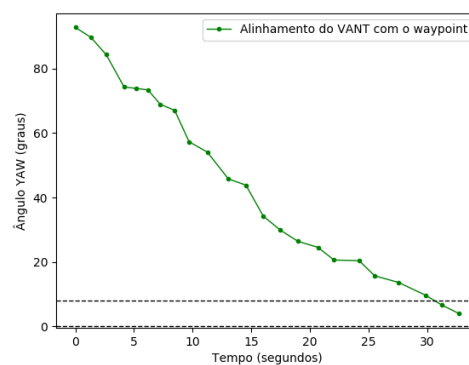


Figura 102 – Alinhamento do VANT com o *waypoint* visual (Teste 8).

## 7.4.2 Experimento de voo real na execução de um trajeto

Este experimento de voo real teve como objetivo realizar um trajeto utilizando o software de navegação por imagem desenvolvido. Para isso, inicialmente, foi construído um trajeto através da junção de materiais com imagens impressas a fim de fornecerem informações para serem utilizadas como *waypoints*. Após isso, foi realizado um voo pelo VANT líder, em modo manual, para a coleta dos marcos do trajeto. Em seguida, posicionou-se o VANT seguidor no início da trajetória, os dados foram carregados no software de navegação e o modo autônomo de navegação foi iniciado, dando início à execução dos testes. A Figura 103 ilustra o VANT seguidor no início do trajeto utilizado. A Figura 104 ilustra alguns dos *waypoints* capturados pelo VANT líder. A Figura 105 ilustra o reconhecimento desses *waypoints* sobre uma foto tirada do trajeto. Por fim, a Figura 106 mostra um instante na execução dos testes. Os vídeos deste experimento podem ser acessados através dos links abaixo:

Vídeo 7: <<https://www.youtube.com/watch?v=M0uIdKfUk3c>>; e

Vídeo 8: <<https://www.youtube.com/watch?v=Kjy94fe4u1U>>.



Figura 103 – VANT seguidor no início do trajeto a ser realizado.

Para ser utilizado como referência, a trajetória original foi plotada na Figura 107. Nela, foram capturados 26 *waypoints* ao longo dos 2,5 metros de comprimento do percurso, para isso necessitou-se de 629 Kb de armazenamento. Esses dados estão expressos na Tabela 17.

Tabela 17 – Informações do trajeto executado.

	<b>Trajeto</b>
Quantidade de <i>waypoints</i>	26
Comprimento do trajeto	2,5 m
Armazenamento do trajeto	629 Kb



Figura 104 – Ilustração de 16 dos 26 *waypoints* do trajeto executado.

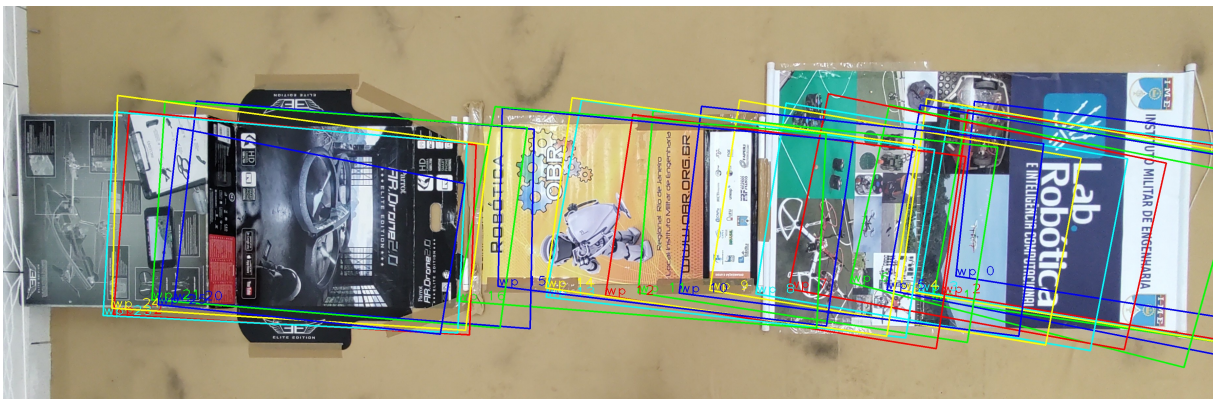


Figura 105 – Foto do percurso com alguns *waypoints* destacados.

#### 7.4.2.1 Resultados

O sistema de navegação implementado obteve êxito na execução da trajetória pretendida. Nos quatro testes realizados, o VANT seguidor percorreu os 26 *waypoints* capturados do percurso de maneira autônoma. No piloto automático, foi utilizada a estratégia de perseguição do *waypoint* de maior índice visível juntamente com o controlador On-Off. Algumas informações de desempenho estão expressas na Tabela 18. Por exemplo, no teste 9 realizado, foram enviados 9 comandos de rotação e 29 de translação. O VANT executou o trajeto em 62,2 segundos, voando a uma altitude de 0,75 m. Neste tempo de execução, não estão contabilizados a decolagem e pouso da aeronave. Na execução, o

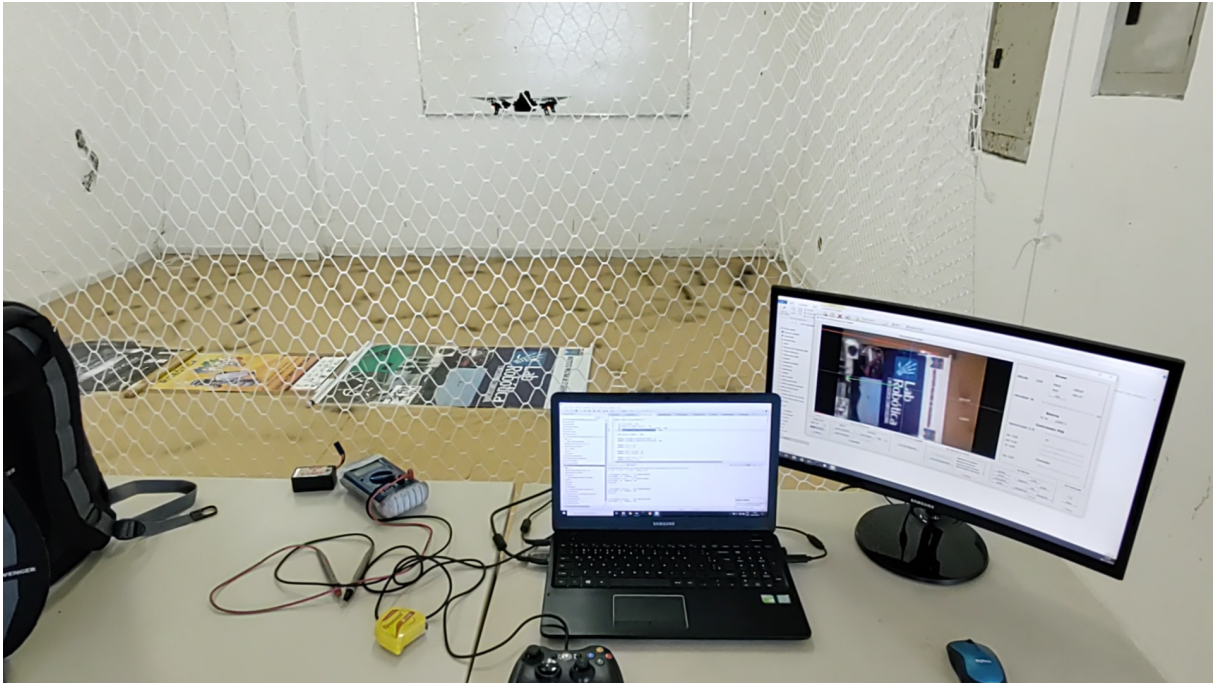


Figura 106 – Execução da trajetória pelo VANT seguidor.

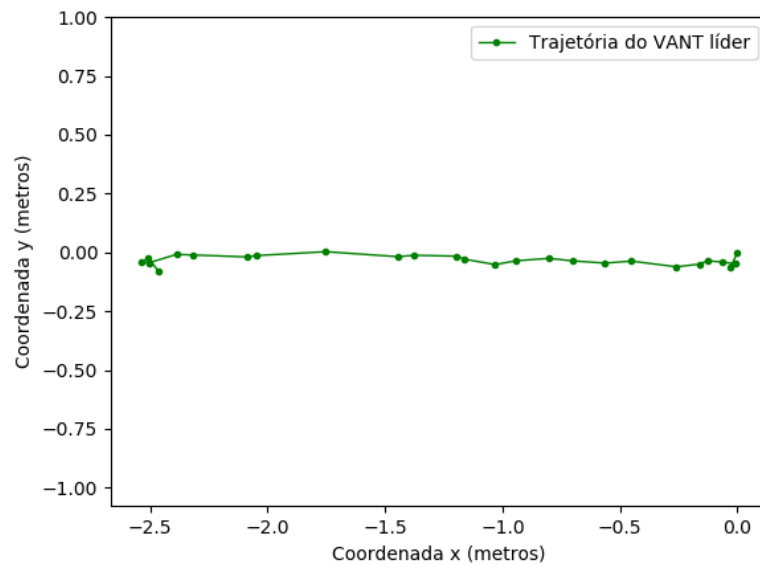


Figura 107 – Trajetória realizada pelo VANT líder.

RMSE encontrado para o eixo  $y$  foi de 0,089 m. Adicionalmente, as Figuras 108, 110, 112 e 114 mostram o comportamento do ângulo de  $yaw$  durante a execução dos testes. Já as Figuras 109, 111, 113 e 115 mostram, em verde, a trajetória original executada pelo VANT líder. Em vermelho, estão as trajetórias executadas pelo VANT seguidor utilizando o software de navegação implementado.

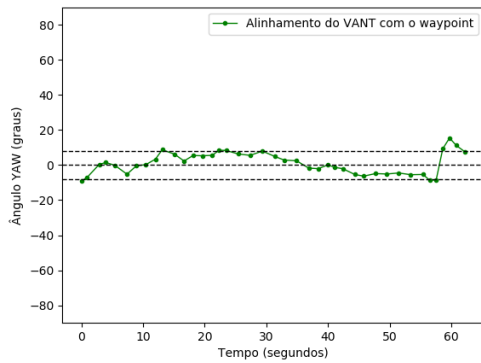


Figura 108 – Alinhamento do VANT com o *waypoint* visual (Teste 9).

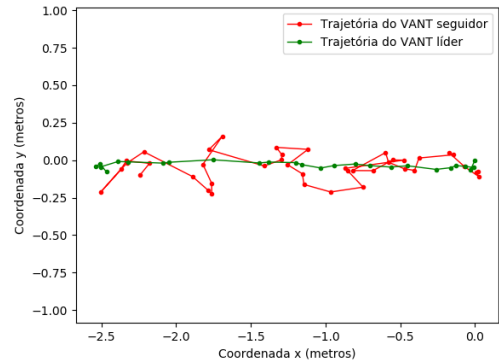


Figura 109 – Deslocamento do VANT sobre o percurso mapeado (Teste 9).

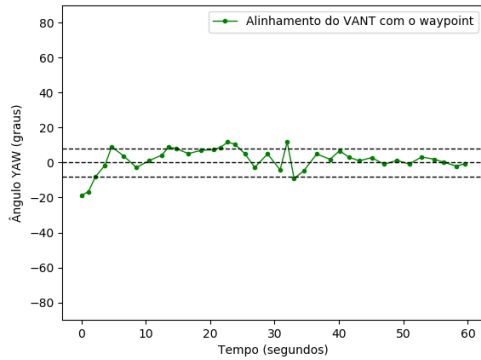


Figura 110 – Alinhamento do VANT com o *waypoint* visual (Teste 10).

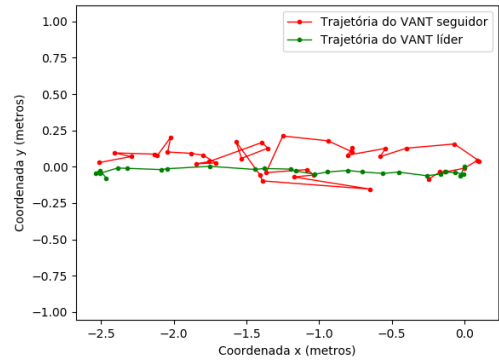


Figura 111 – Deslocamento do VANT sobre o percurso mapeado (Teste 10).

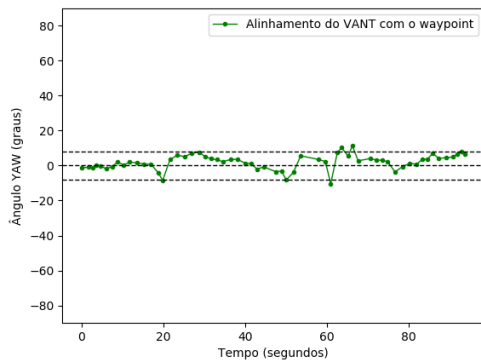


Figura 112 – Alinhamento do VANT com o *waypoint* visual (Teste 11).

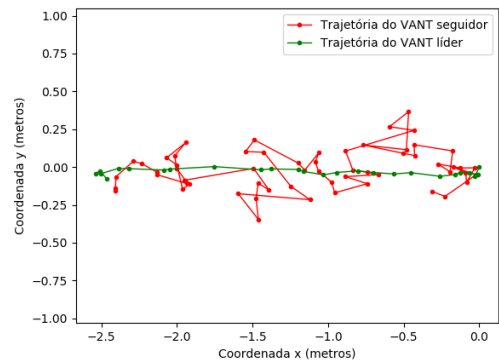


Figura 113 – Deslocamento do VANT sobre o percurso mapeado (Teste 11).



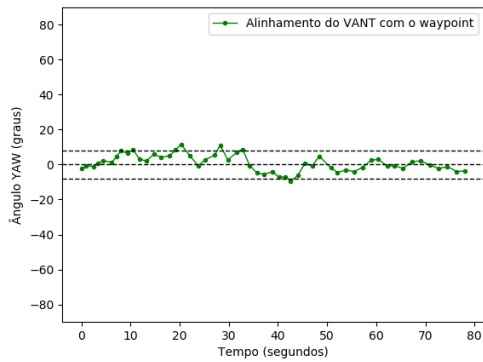


Figura 114 – Alinhamento do VANT com o *waypoint* visual (Teste 12).

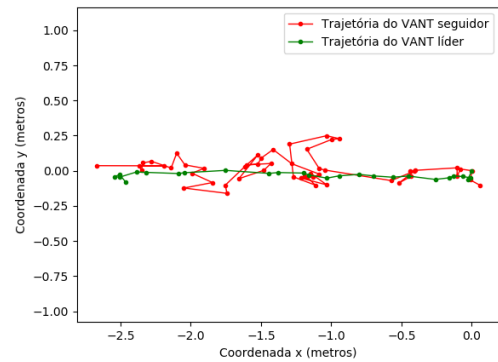


Figura 115 – Deslocamento do VANT sobre o percurso mapeado (Teste 12).

Tabela 18 – Dados dos testes de execução da trajetória pelo VANT seguidor.

	<b>Teste 9</b>	<b>Teste 10</b>	<b>Teste 11</b>	<b>Teste 12</b>
Comandos de rotação enviados	9	11	6	6
Comandos de translação enviados	29	27	49	42
Altitude de voo	0,75 m	1,00 m	1,00 m	0,75 m
RMSE em $y$	0,089 m	0,125 m	0,129 m	0,097 m
Tempo total de execução	62,2 s	59,6 s	93,9 s	78,1 s

### 7.4.3 Dados de latência do sistema

Foram coletados dados de latência do sistema. Esta inclui o tempo necessário para receber a imagem transmitida pelo VANT, processá-la, pesquisar no banco de dados os *waypoints* armazenados e enviar os comandos de movimento à aeronave. Entre essas atividades, a maior latência encontrada se deve ao tempo de transmissão da imagem do VANT para o software de navegação visual. Na Figura 116, a linha vermelha representa latência de transmissão em milissegundos. A resolução das imagens utilizadas foi 640x360 pixels. A latência média na transmissão foi de 408 ms. A linha azul representa o tempo necessário para realizar o processamento da imagem recebida, incluindo a extração dos *keypoints*, cálculo dos descritores e a busca no banco de dados de *waypoints*. Nesse caso, o tempo médio de processamento foi de 152 ms. A linha verde representa o tempo necessário para enviar os comandos de movimento na aeronave. A média desse tempo foi de 4 ms. A latência total encontrada no sistema foi de 564 ms. A Tabela 19 apresenta esses dados.

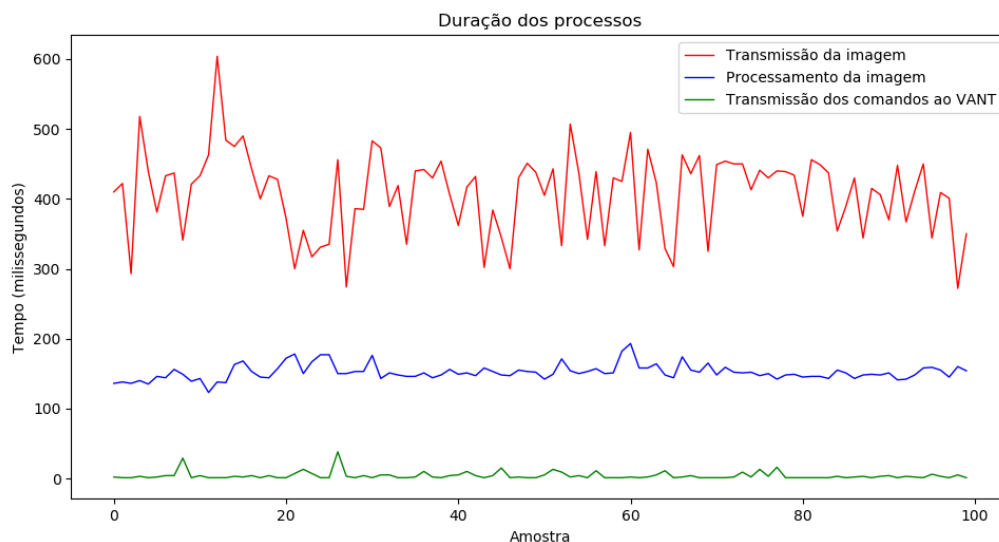


Figura 116 – Gráfico com as durações dos processos de transmissão das imagens, seu processamento e envio dos comandos de controle ao VANT.

Tabela 19 – Tabela com as latências médias do sistema.

Processo	Latência média
Transmissão e recebimento das imagens	408ms
Processamento das imagens	152ms
Envio de comandos ao VANT	4ms
Latência total	564ms

## 8 CONSIDERAÇÕES FINAIS

Neste capítulo, serão apresentadas as conclusões sobre os resultados encontrados nos experimentos realizados. Em seguida, serão expostos alguns trabalhos futuros que se deseja realizar após a finalização deste mestrado.

### 8.1 Conclusão

Na introdução, destacou-se a crescente importância que os VANTs vêm ganhando no cenário mundial, onde estão sendo empregados em inúmeras situações civis e militares. Frisou-se, também, que a pesquisa e o desenvolvimento de VANTs pelas Forças Armadas são de vital importância para se criar independência tecnológica do Brasil nesse assunto, contribuindo para o poder dissuasório do país. Como já expresso, este tema encontra-se em linha com o Plano Estratégico do Exército 2016-2019 [16] e foi aprovado nas Necessidades de Conhecimentos Específicos (NCE) de 2018 [45].

Nesse contexto de VANTs, evidencia-se uma grande dependência por parte das aeronaves em relação aos sistemas de posicionamento por satélites, como GPS, Glonass e Beidou, para sua navegação e localização. Visando encontrar meios alternativos para essa questão, este trabalho propôs utilizar uma câmera embarcada para realizar a localização e a navegação da aeronave através da aquisição de imagens do terreno. Segundo o levantamento bibliográfico realizado, nos últimos anos, a comunidade acadêmica vem aumentando as pesquisas nesse sentido, pois a câmera se tornou um sensor barato, com amplo alcance e grande quantidade de informação disponibilizada. Assim, este trabalho realizou o desenvolvimento de um sistema de navegação por reconhecimento de imagens do terreno. O sistema é dividido em dois passos principais que envolvem a utilização de um VANT líder e de um VANT seguidor. O VANT líder possui o papel de capturar diversas imagens do terreno ao longo da trajetória na qual se almeja realizar voos autônomos. O VANT seguidor, com os dados dessa trajetória, irá refazer tal percurso, percorrendo os *waypoints* visuais adquiridos pelo VANT líder.

Para alcançar este objetivo, inicialmente, demonstrou-se a viabilidade técnica e operacional da utilização de imagens aéreas do terreno para a tarefa de localização. Para isso, foram realizados os experimentos descritos na Seções 7.1. Estes experimentos demonstraram que, através de imagens aéreas do terreno e sob determinadas condições de contorno, é possível realizar associações entre imagens e, então, extrair a provável localização do VANT. Posteriormente, por intermédio dos experimentos descritos na Seção 7.2, demonstrou-se que, em bancada, é possível utilizar uma câmera para a aquisição de

imagens e realizar o reconhecimento de múltiplos *waypoints* visuais simultâneos em tempo real. Demonstrou-se, também, que é possível extrair dados, como o centro e o ângulos de rotação dos *waypoints*. No último experimento desta Seção, através do cálculo do vetor deslocamento e do ângulo de rotação, realizou-se um ensaio de navegação sobre um mapa impresso.

Em seguida, na Seção 7.3, foi realizado um conjunto de simulações em software demonstrando que um VANT quadricóptero, em condições ideais, é capaz de seguir rotas utilizando visão computacional. Para isso, foram testadas duas possíveis estratégias de perseguição dos *waypoints* conjugadas com dois tipos de controladores automáticos diferentes. Das quatro combinações resultantes, aliadas aos quatro trajetos básicos sugeridos, resultaram 16 simulações realizadas. Em todos os 16 casos testados, o software de navegação implementado foi capaz de refazer os percursos armazenados de maneira autônoma.

Após as simulações, foram realizados os testes de voo real. Para isso, foi utilizada uma aeronave comercial, conforme expresso na Seção 7.4. Os primeiros testes de voo possuíam o objetivo de centralizar a aeronave sobre um *waypoint* visual cadastrado. Foram extraídos indicadores de desempenho dessa tarefa realizada. Já os últimos testes de voo possuíam o objetivo de fazer o VANT seguidor percorrer um pequeno trajeto realizado inicialmente pelo VANT líder. Tais experimentos também foram um sucesso.

Por fim, apesar das dificuldades logísticas e técnicas encontradas ao longo do trabalho, os experimentos realizados apontam para a possibilidade da utilização de imagens aéreas como meio de localização e navegação para VANTs. Fato que demonstra a tese proposta neste trabalho e valida o funcionamento do sistema implementado. Como resultado, espera-se que os produtos resultantes deste trabalho contribuam para a pesquisa e o desenvolvimento de meios alternativos para a navegação de VANTs, diminuindo a dependência existente dos sistemas de localização por satélites e aumentando o capital intelectual das Forças Armadas relativo a esse tema em crescente ascensão mundial.

## 8.2 Trabalhos futuros

Nesta Seção, serão listados alguns trabalhos futuros que não foram executados durante este mestrado por limitação de tempo. Tais trabalhos podem complementar e agregar os resultados aqui já encontrados e validados.

- **Realização de um voo completo, em ambiente externo, utilizando o software de navegação por reconhecimento de imagem do terreno.** Devido à grande complexidade do sistema desenvolvido e às restrições logísticas e legais referentes a VANTs experimentais, não se conseguiu, em tempo hábil, realizar um voo completo por imagem em ambientes abertos. Porém, com a infraestrutura de

software já criada, pretende-se realizá-lo no futuro.

- **Criação de um protótipo operacional da função *Return-To-Launch* utilizando navegação por reconhecimento de imagens.** O princípio de funcionamento atual desta função, disponível nos principais pilotos automáticos existentes, é baseado em GPS. Assim, uma implementação do *Return-To-Launch*, baseada em visão computacional, seria um trabalho diferencial (ainda não disponível na comunidade) e bastante agregador.
- **Embarque no VANT do sistema desenvolvido.** O núcleo do código gerado, contendo as principais funcionalidades de navegação por imagem e aquisição da trajetória, está escrito em Java e C++, sendo possível o embarque em outras plataformas de hardware.
- **Aquisição da trajetória utilizando imagens de satélites.** Uma possibilidade existente, que necessita ser testada e validada, é a construção da trajetória utilizando imagens de satélites. Neste caso, o voo do VANT líder será substituído por um software que, através das imagens de satélite da área pretendida, irá gerar os *waypoints* do trajeto.
- **Utilização de *Deep Learning* para o reconhecimento dos *waypoints*.** Teste e utilização de modernos mecanismos de *Deep Learning* para a detecção e reconhecimento dos *waypoints* visuais do terreno. Uma das abordagens que pode ser utilizada é através de Redes Neurais Convolucionais com múltiplas camadas.
- **Incorporação de mecanismos de controle mais robustos como *autotuning* dos parâmetros do controlador PID.** O mecanismo de *autotuning* permite otimizar os valores dos ganhos utilizados nos controladores PID. Isso permitiria ao VANT obter seu melhor desempenho nos deslocamentos e manobras aéreas.

# REFERÊNCIAS

- 1 BIONICBIRD. *Birds - Bionic bird - YOUR NEXT GEN FLYING PET - WORLD*. 2019. 20 out. de 2019. Disponível em: <<https://bionicbird.com>>. 10, 29
- 2 ARDUPILOT. *Copter Home — Copter documentation*. 2019. 01 nov. de 2019. Disponível em: <<http://ardupilot.org/copter/index.html>>. 10, 32
- 3 POSITIONING, N. National Coordination Office for S.-B.; TIMING. *GPS: The Global Positioning System*. 2019. 2 nov. de 2019. Disponível em: <<https://www.gps.gov/>>. 10, 22, 38, 52
- 4 KAEHLER, A.; BRADSKI, G. *Learning OpenCV 3, Computer Vision in C++ with the OpenCV Library*. 1. ed. United States of America: O'Reilly Media Inc., 2017. 1018 p. 10, 39, 43, 44, 45
- 5 ALBIERO, V.; BIASI, H. H. de. Drone autônomo guiado através de templates utilizando visão computacional. *Anais SULCOMP*, v. 8, p. 20–31, 2017. 11, 46, 49, 50
- 6 ENGEL, J.; STURM, J.; CREMERS, D. Camera-based navigation of a low-cost quadcopter. In: IEEE. *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*. [S.l.], 2012. p. 2815–2821. 11, 46, 47, 49, 50
- 7 KRAJNÍK, T. et al. A simple visual navigation system for an uav. In: IEEE. *Systems, Signals and Devices (SSD), 2012 9th International Multi-Conference on*. [S.l.], 2012. p. 1–6. 11, 46, 47, 49, 50
- 8 BRANDAO, A. S.; MARTINS, F. N.; SONEGUETTI, H. B. A vision-based line following strategy for an autonomous uav. In: IEEE. *2015 12th International Conference on Informatics in Control, Automation and Robotics (ICINCO)*. [S.l.], 2015. v. 2, p. 314–319. 11, 47, 48, 49, 50
- 9 LEE, H.; JUNG, S.; SHIM, D. H. Vision-based uav landing on the moving vehicle. In: IEEE. *2016 International conference on unmanned aircraft systems (ICUAS)*. [S.l.], 2016. p. 1–7. 11, 48, 49, 50
- 10 STORE, J. *GPS JAMMERS Lojack blockers anti tracking devices*. 2019. 5 set. de 2019. Disponível em: <<https://www.jammer-store.com/altron-4-handheld-drone-jammer.html>>. 11, 53
- 11 GOOGLE. *Google Earth*. 2018. 10 out. de 2018. Disponível em: <<https://www.google.com/earth/>>. 11, 12, 74, 75
- 12 DRONEMAPPER. *Drone Mapping Aerial Imagery Example Data - DroneMapper*. 2018. 10 out. de 2018. Disponível em: <[https://dronemapper.com/sample\\_data/](https://dronemapper.com/sample_data/)>. 12, 78, 79
- 13 BENTO, M. d. F. Unmanned aerial vehicles: an overview. *Inside GNSS*, v. 3, n. 1, p. 54–61, 2008. 15, 29, 30

- 14 FALLAH, N. et al. Indoor human navigation systems: A survey. *Interacting with Computers*, Oxford University Press, v. 25, n. 1, p. 21–33, 2013. 15, 51, 52
- 15 KARAMI, E.; PRASAD, S.; SHEHATA, M. Image matching using sift, surf, brief and orb: performance comparison for distorted images. *arXiv preprint arXiv:1710.02726*, 2017. 15, 72, 73
- 16 BRASILEIRO, C. do E. *Portaria no 1.042, de 18 de agosto de 2017 - Plano Estratégico do Exército 2016-2019 (3a edição)*. 2018. 20 out. de 2018. Disponível em: <<http://www.sgex.eb.mil.br/sistemas/be/copiar.php?codarquivo=1541&act=bre>>. 22, 114
- 17 RODRIGUES, A. C. de C. *Fusão de sensores para um VANT via suavização incremental baseada em grafos-fatores*. 133 p. — Instituto Militar de Engenharia, Rio de Janeiro, 2015. 15 out. de 2018. Disponível em: <<http://www.comp.ime.eb.br/pos/arquivos/publicacoes/dissertacoes/2015/2015-Anderson.pdf>>. 22
- 18 BEIDOU. *BeiDou Navigation Satellite System*. 2018. 1 nov. de 2018. Disponível em: <<http://en.beidou.gov.cn/>>. 22
- 19 GLONASS. *Information analytical centre of GLONASS and GPS controlling*. 2018. 1 nov. de 2018. Disponível em: <<https://www.glonass-iac.ru/en/index.php>>. 22
- 20 AGENCY, E. G. N. S. S. *Galileo is the European global satellite-based navigation system - European Global Navigation Satellite Systems Agency*. 2018. 1 nov. de 2018. Disponível em: <<https://www.gsa.europa.eu/european-gnss/galileo/galileo-european-global-satellite-based-navigation-system>>. 22
- 21 JUNIOR, F. L. et al. Navegação de vant por reconhecimento de imagem do solo. *ANAIS DO 14º SIMPÓSIO BRASILEIRO DE AUTOMAÇÃO INTELIGENTE*, v. 1, 2019. 23
- 22 CARRILLO, L. R. G. et al. *Quad rotorcraft control: vision-based hovering and navigation*. [S.l.]: Springer Science & Business Media, 2012. 28
- 23 SILVA, L. C. B. da. *Sistema de aeronaves remotamente pilotadas com alocação dinâmica por controle baseado na tarefa para cobertura de áreas com diferentes prioridades de interesse*. 154 p. Tese (Doutorado) — Instituto Militar de Engenharia, Rio de Janeiro, 2018. 30
- 24 BRASIL, E. *Navio2 - Emlid Brasil*. 2019. 2 nov. de 2019. Disponível em: <<https://emlid.com/br/navio2/>>. 32
- 25 PIXHAWK. *Home Page - Pixhawk*. 2019. 2 nov. de 2019. Disponível em: <<https://pixhawk.org/>>. 32
- 26 HOLYBRO. *Holybro – Flight Controller, GPS, ESC, Radio, and Accessories for Drones and FPV*. 2019. 2 nov. de 2019. Disponível em: <<http://www.holybro.com/>>. 32
- 27 MAVLINK. *Introduction - MAVLink Developer Guide*. 2018. 5 nov. de 2018. Disponível em: <<https://mavlink.io/en/>>. 33
- 28 OGATA, K. *Engenharia de Controle Moderno*. 5. ed. [S.l.]: Pearson Prentice Hall, 2018. 809 p. 41, 42

- 29 OPENCV. *OpenCV Library*. 2018. 20 out. de 2018. Disponível em: <<https://opencv.org/>>. 43
- 30 KAEHLER, A.; BRADSKI, G. *Learning OpenCV 3, Computer Vision in C++ with the OpenCV Library*. 1. ed. United States of America: O'Reilly Media Inc., 2017. 493–585 p. 44
- 31 ALBIERO, V.; BIASI, H. H. D. Desenvolvimento de um drone autônomo guiado por meio de técnicas de visão computacional. *Unoesc & Ciência-ACET*, v. 7, n. 1, p. 35–44, 2016. 46, 49, 50
- 32 BAY, H. et al. Speeded-up robust features (surf). *Computer vision and image understanding*, Elsevier, v. 110, n. 3, p. 346–359, 2008. 47
- 33 LU, Y. et al. A survey on vision-based uav navigation. *Geo-spatial information science*, Taylor & Francis, v. 21, n. 1, p. 21–32, 2018. 48
- 34 CHO, D.-M. et al. Robust feature detection, acquisition and tracking for relative navigation in space with a known target. In: *AIAA Guidance, Navigation, and Control (GNC) Conference*. [S.l.: s.n.], 2013. p. 5197. 48
- 35 SHEN, S. et al. Multi-sensor fusion for robust autonomous flight in indoor and outdoor environments with a rotorcraft mav. In: *IEEE. Robotics and Automation (ICRA), 2014 IEEE International Conference on*. [S.l.], 2014. p. 4974–4981. 48
- 36 LEUTENEGGER, S. et al. Keyframe-based visual–inertial odometry using nonlinear optimization. *The International Journal of Robotics Research*, SAGE Publications Sage UK: London, England, v. 34, n. 3, p. 314–334, 2015. 48
- 37 DING, G. et al. Overview of received signal strength based fingerprinting localization in indoor wireless lan environments. In: *IEEE. 2013 5th IEEE International Symposium on Microwave, Antenna, Propagation and EMC Technologies for Wireless Communications*. [S.l.], 2013. p. 160–164. 51
- 38 POSITIONING, N. National Coordination Office for S.-B.; TIMING. *GPS.gov: Space Segment*. 2019. 2 nov. de 2019. Disponível em: <<https://www.gps.gov/systems/gps/space/>>. 53
- 39 JAFARNIA-JAHROMI, A. et al. Gps vulnerability to spoofing threats and a review of antispooing techniques. *International Journal of Navigation and Observation*, Hindawi, v. 2012, 2012. 53
- 40 NEWS, B. <https://www.bbc.com/news/world-middle-east-16098562>. 2019. 5 set. de 2019. Disponível em: <<https://www.csmonitor.com/World/Middle-East/2011/1215/Exclusive-Iran-hijacked-US-drone-says-Iranian-engineer>>. 53
- 41 MONITOR, T. C. S. *Exclusive: Iran hijacked US drone, says Iranian engineer*. 2019. 5 set. de 2019. Disponível em: <<https://www.csmonitor.com/World/Middle-East/2011/1215/Exclusive-Iran-hijacked-US-drone-says-Iranian-engineer>>. 53
- 42 PARROT. *Parrot Software Development Kit AR.Drone*. 2019. Disponível em: <[http://developer.parrot.com/docs/SDK2/ARDrone\\_SDK\\_2\\_0\\_1.zip](http://developer.parrot.com/docs/SDK2/ARDrone_SDK_2_0_1.zip)>. 67



- 43 YADRONE. *YADrone - Yet another AR.Drone framework*. 2019. Disponível em: <view-source:https://vsis-www.informatik.uni-hamburg.de/oldServer/teaching/projects/yadrone/index.html>. 67
- 44 TECHNOLOGIES, U. *Plataforma de desenvolvimento em tempo real do Unity | Visualizações 3D, 2D VR e AR*. 2019. 10 nov. de 2019. Disponível em: <https://unity.com/pt>. 87
- 45 TECNOLOGIA, D. de Ciência e. *Portaria Nº 014-DCT, de 14 de março de 2017 - Necessidades de Conhecimentos Específicos*. 2018. 25 out. de 2018. Disponível em: <http://www.ime.eb.mil.br/arquivos/Admissao/Pos-graduacao/2017/militares/NCE\_2018\_Portaria\_014-DCT.pdf>. 114