

UNIVERSIDADE FEDERAL DE UBERLÂNDIA
FACULDADE DE ENGENHARIA MECÂNICA
GRADUAÇÃO EM ENGENHARIA MECATRÔNICA

ANDRÉ FERNANDES DE ALMEIDA

**IMPLEMENTAÇÃO DE ALGORITMO DE VISÃO COMPUTACIONAL
PARA COMUNICAÇÃO ENTRE MEMBROS DE UM ENXAME DE
ROBÔS**

Uberlândia - MG
2018

ANDRÉ FERNANDES DE ALMEIDA

IMPLEMENTAÇÃO DE ALGORITMO DE VISÃO COMPUTACIONAL PARA COMUNICAÇÃO ENTRE MEMBROS DE UM ENXAME DE ROBÔS

Monografia apresentada à Faculdade de Engenharia Mecânica da Universidade Federal de Uberlândia, como parte das exigências para a obtenção do título de bacharel em Engenharia Mecatrônica.

Área de Concentração: Processamento de Imagem

Orientador: Prof. Dr. Mauricio Cunha Escarpinati

Uberlândia - MG

2018

*“Transportai um punhado de terra
todos os dias e fareis uma montanha”*

Confúcio

Agradecimentos

Ao meu pai André e à minha mãe Diva, que são as pessoas que formaram os valores que carrego na vida e que sempre apoiaram minhas escolhas, dando o suporte necessário para que esta graduação pudesse ser concluída.

À minha esposa Marcela por todo carinho e paciência ao longo desses anos, sendo meu porto seguro em momentos de fraqueza.

Às minhas irmãs Daniellie e Bruna que estão sempre me apoiando nos momentos que necessito.

Aos meus sogros Hécio e Cláudia pela paciência que sempre tiveram comigo.

Ao Prof. Dr. Mauricio Cunha Escarpinati pela orientação desse projeto, pelos aprendizados proporcionados e principalmente pela paciência que sempre teve comigo durante a realização desse trabalho.

Resumo

Sistemas com múltiplos robôs, também chamados de enxame de robôs, estão ganhando cada vez mais espaço em pesquisas. Devido ao grande número de aplicações, os drones lideram esse crescimento, sendo utilizados para exploração de ambientes, filmagens e entregas. Um requisito vital para que sistemas de múltiplos robôs funcione de forma adequada é a comunicação. Como os meios de comunicação convencionais estão sujeitos aos mais variados tipos de falhas, surge, como alternativa, a comunicação utilizando a visão computacional. Desse modo, inspirados pela forma de comunicação visual das abelhas, o presente trabalho tem por objetivo identificar objetos de interesse no campo de visão do drone e analisar sua movimentação a fim de reconhecer qual informação tal objeto está transmitindo. Assim, foram implementados os algoritmos de Lucas-Kanade e de detecção de cantos de Harris, para a segmentação desses objetos de interesse. Por fim, técnicas de *template matching* foram utilizadas para identificação de modelos esperados, no qual se obteve resultados satisfatórios no reconhecimento da informação transmitida de forma visual, atingindo o índice de 75% de acerto.

Palavras-chave: Enxame de robôs, Drones, Computação Bioinspirada, Visão Computacional, Template Matching.

Abstract

Systems with multiple robots, also called swarm robots, are growing and getting more importance in research. Due to the large number of applications, the drones are leading this growth, being used for exploitation of environments, filming and deliveries. An essential requirement for multi-robot systems behave properly is the communication. As the conventional way of communication can be under various kind of failures, emerges as an alternative the communication using computer vision. Thus, inspired on bees visual's communication, the present work aims to identify objects of interest in the drone's field of vision and analyzes its movement, in order to recognize what information that object is sending. Hence, were implemented the Lucas-Kanade's algorithms and Harris's corners detection, with the purpose of segmentation of these objects of interest. At last, template matching techniques were used to identify expected models, acquiring satisfactory results on information's recognition transmitted visually, which achieved the level of 75% of success.

Palavras-chave: Swarm Robots, Drones, Bioinspired Computing, Computer Vision, Template Matching.

Lista de Figuras

Figura 1 - Drone militar RQ-4 Block 10 Global Hawk (GRUMMAN, 2018)	5
Figura 2 - Drone Prime Air da Amazon carregando um pacote (AMAZON, 2018).....	6
Figura 3 - Representação da implementação do algoritmo de Harris e sua variação de janela mostrando um objeto plano (a), uma borda (b) e um canto (c) (SINHA, 2012)	11
Figura 4 - Ilustração da movimentação do drone DV2. Oscilação vertical (a) e oscilação horizontal (b)	14
Figura 5 - Ilustração dos 4 movimentos realizados pelo drone DR2. Oscilação vertical (a), oscilação horizontal (b), quadrado (c) e T (d).....	15
Figura 6 - Vetores atribuídos com o algoritmo de Lucas e Kanade (COPPELIA ROBOTICS, 2018)	16
Figura 7 - Exemplo de segmentação do drone no ambiente de simulação (COPPELIA ROBOTICS, 2018).....	17
Figura 8 - Localização do drone no resultado da subtração de frames sequenciais em ambiente real.....	18
Figura 9 - Exemplo de assinatura de movimentação.	18
Figura 10 – Exemplo de um sinal original em (a) e sinal suavizado e normalizado em (b).....	19
Figura 11 - Exemplo de um sinal com períodos e amplitudes irregulares.	20
Figura 12 - Exemplo das amostras obtidas após a divisão dos dados globais por duas vezes o tamanho do modelo inferido.	21
Figura 13 - Ambiente de simulação V-REP (COPPELIA ROBOTICS, 2018).....	24
Figura 14 - Modelo Quadricopter, fornecido por Eric Rohmer (COPPELIA ROBOTICS, 2018) .	24
Figura 15 - Quadricopter no deserto (COPPELIA ROBOTICS, 2018)	25
Figura 16 - Quadricopter no labirinto (COPPELIA ROBOTICS, 2018).....	25
Figura 17 - Drone MAVIC PRO (DJI, 2018)	25
Figura 18 - Local de captura das imagens reais.....	26
Figura 19 - Vetores segmentando o drone através do algoritmo de Lucas-Kanade. Ambiente interno (esq.) e ambiente externo (dir.).....	27
Figura 20 - Ruído na segmentação utilizando a técnica de fluxo óptico de Lucas e Kanade ...	28
Figura 21 - Drone segmentado com o uso do algoritmo de Harris.....	29
Figura 22 - Drone não encontrado na segmentação com o uso do algoritmo de Harris.....	29

Figura 23 - Drone segmentado (à esq.) e a localização do centroide (à dir.) no ambiente interno	30
Figura 24 - Drone segmentado (à esq.) e a localização do centroide (à dir.) no ambiente externo	30
Figura 25 - Gráfico da posição por frame da oscilação vertical em ambiente simulado interno. (a) Deslocamento em x e (b) deslocamento em y.....	31
Figura 26 - Gráfico da posição por frame da oscilação horizontal em ambiente simulado interno. (a) Deslocamento em x e (b) deslocamento em y.....	31
Figura 27 - Gráfico da posição por frame da oscilação vertical em ambiente simulado externo. (a) Deslocamento em x e (b) deslocamento em y.....	32
Figura 28 - Gráfico da posição por frame da oscilação horizontal em ambiente simulado externo. (a) Deslocamento em x e (b) deslocamento em y.	32
Figura 29 - Gráfico da posição por frame da movimentação em T em ambiente real mostrando a suavização dos dados.	33
Figura 30 - Gráfico da posição por frame da oscilação vertical em ambiente real. (a) Deslocamento em x e (b) deslocamento em y.	33
Figura 31 - Gráfico da posição por frame da oscilação horizontal em ambiente real. (a) Deslocamento em x e (b) deslocamento em y.	34
Figura 32 - Gráfico da posição por frame da movimentação em quadrado em ambiente real. (a) Deslocamento em x e (b) deslocamento em y.....	34
Figura 33 - Gráfico da posição por frame da movimentação T em ambiente real. (a) Deslocamento em x e (b) deslocamento em y.....	34

Lista de Tabelas

Tabela 1 – Valores de erros mínimos obtidos do conjunto de treinamento	35
Tabela 2 - Média dos erros de cada modelo	36
Tabela 3 - Valores de erros mínimos obtidos do conjunto de teste	37

Sumário

1	Introdução	1
1.1.	Visão Geral	1
1.2.	Objetivos	3
1.3.	Justificativas	3
1.4.	Organização do trabalho.....	4
2	Referencial Teórico	5
2.1.	Drone - VANT	5
2.2.	Enxame de robôs	6
2.3.	Comunicação entre membros de um enxame de robôs	7
2.4.	Computação Bioinspirada	8
2.5.	Processamento Digital de Imagens.....	9
2.5.1.	Fluxo Óptico – Lucas e Kanade	9
2.5.2.	Algoritmo de Harris	10
2.6.	Reconhecimento de padrões	11
2.7.	Ferramentas	12
2.7.1.	MATLAB	12
2.7.2.	V-REP	13
3	Materiais e Métodos	14
3.1.	Captura - Movimentação do Drone	14
3.2.	Segmentação.....	15
3.3.	Processamento – Assinatura dos movimentos.....	18
3.4.	Comparação	19
3.5.	Classificação	22

4	Resultados e Discussões.....	23
4.1.	Ambientes de captura das imagens.....	23
4.1.1.	Ambiente de Simulação	23
4.1.2.	Ambiente Real	25
4.2.	Segmentação.....	27
4.2.1.	Simulação	27
4.2.2.	Real.....	27
4.3.	Análise da trajetória do drone em ambiente simulado.....	30
4.4.	Análise da trajetória do drone em ambiente real	32
4.5.	Classificação dos movimentos	35
5	Conclusão	38
	Referências Bibliográficas.....	40

1 Introdução

1.1. Visão Geral

De forma simples, vivemos na era tecnológica. Carros elétricos e robôs inteligentes já não são coisas do futuro e estão cada vez mais presentes no nosso dia-a-dia. A alta disponibilidade de recursos fazem com que os avanços tecnológicos sejam frequentes e cada vez mais impressionantes. Essa grande quantidade de recurso com a alta difusão do conhecimento dos tempos da internet cria um ambiente sem precedentes para inovações.

Na robótica, algumas pesquisas que envolvem o uso de robôs na realização de tarefas complexas, abordam a cooperatividade de robôs menos complexos como fórmula para o sucesso de tais tarefas. Com inspiração na organização social de animais que trabalham de forma cooperativa, esse sistema tem sido chamado de enxame de robôs (NAVARRO e MATÍA, 2012).

Os robôs utilizados nesses enxames possuem características semelhantes nas diversas áreas de pesquisa: são pequenos, de baixo custo, e operam de forma autônoma, sendo essa característica o principal desafio dessa organização. Para que seja possível a autonomia do grupo, o processo de comunicação é fundamental, devendo ser claro e distribuído por todo o enxame. Diversas formas de comunicação podem ser usadas dependendo de fatores como o ambiente de realização da tarefa, o tamanho dos robôs ou as limitações do projeto (JASMINE, 2015).

Como podemos notar, sistemas com múltiplos robôs oferecem complexidades na administração do grupo para que a tarefa seja executada de forma coordenada e colaborativa. Para o bom funcionamento desses sistemas as características de navegação, planejamento e comunicação devem ser executadas em harmonia. Além disso, a todo instante cada integrante do enxame deve tomar inúmeras decisões que irão refletir no comportamento do grupo (BOSQUE, 2009).

Encontramos na natureza fontes de inspirações para solução de diversos tipos de problemas. Estas soluções são observadas nos comportamentos sociais de inúmeras espécies de seres vivos. As abelhas são exemplos de insetos sociais, as quais realizam atividades

cooperativas para a manutenção e sobrevivência da espécie. A construção da colmeia, a busca e o posterior processamento de alimentos são algumas dessas atividades. Em todas elas podemos destacar a comunicação como um fator importante para o sucesso da colônia. Essa comunicação pode ser realizada por meio de sons, substâncias químicas, tato, danças ou estímulos eletromagnéticos. Através da dança elas conseguem informar a distância e a localização exata de uma fonte de alimento. Essas interações que ocorrem em sistemas biológicos, resultam em formas alternativas de solucionar problemas na computação (FRISCH†, 1973).

Existe uma área de estudo na Ciência da Computação especialmente focada em entender esses padrões da natureza, ela é denominada computação bioinspirada. Essa área, além de buscar compreender, procura formas de aplicar tais padrões na resolução de problemas do cotidiano.

Na solução de problemas que envolvem reconhecimento de padrões, alguns algoritmos bioinspirados são baseados no sistema nervoso. As análises do comportamento de colônias de seres vivos já foram utilizadas para a resolução de problemas de otimização. Essa união entre a ciência natural e a ciência da computação tem se mostrado assertiva na medida que os desenvolvimentos de soluções eficazes para problemas complexos estão sendo descobertos (MEDEIROS, MENEZES, *et al.*, 2015).

A computação bioinspirada possui suas aplicações em outros campos, como na robótica. Essas aplicações estão presentes na implementação de controladores, na otimização de movimentos e também na classificação de dados. No uso de sistemas com vários robôs que devem trabalhar em cooperação mutua, a inteligência coletiva é utilizada como base para a configuração do enxame. Por exemplo, abordagens baseadas nos enxames de abelhas ou na colônia de formigas (MEDEIROS, MENEZES, *et al.*, 2015).

Considerando a relevância da comunicação entre integrantes de um enxame de robôs e com base em uma das formas de comunicação das abelhas, o presente trabalho busca uma alternativa de comunicação entre membros desses sistemas, a comunicação visual.

Para que essa comunicação aconteça, é necessário que os robôs sejam capazes de identificar visualmente uns aos outros e posteriormente suas respectivas movimentações.

Dessa maneira, conseguirão comparar tais movimentações com um alfabeto pré-definido e entender as mensagens que o outro robô está transmitindo.

Os robôs que serão utilizados no projeto são os drones. Eles são pequenos robôs aéreos, controlados por homens a distância, que não necessitam de tripulação. Concebidos inicialmente para uso militar, é uma forma segura de ataque que preserva a vida dos soldados. Hoje em dia, esses dispositivos tem o uso bastante diverso, sendo utilizado para controlar incêndios, em filmagens e também em entregas.

1.2. Objetivos

Esse trabalho tem como objetivo identificar padrões de movimentos realizados por um drone através da computação visual e do processamento de imagens. Para isso, serão abordados a forma de captura das imagens, técnicas de segmentação e classificadores de padrões.

Para a segmentação, será desenvolvido um algoritmo capaz de localizar objetos se movendo em uma sequência de imagens. Posteriormente, outro algoritmo será desenvolvido para a classificação das curvas de deslocamento.

1.3. Justificativas

Avanços recentes na robótica e na inteligência artificial estão revolucionando os negócios, a sociedade e nossas vidas pessoais. Máquinas trabalhando na linha de montagem, carros autoguiados e robôs que limpam a casa são situações que ilustram como os robôs estão cada vez mais presentes no nosso cotidiano.

Com esses avanços, surgiu na robótica o conceito de sistema com múltiplos robôs. Esse tipo de sistema possui algumas vantagens sobre aqueles sistemas que possuem somente um robô, visto que em situações de falha o sistema com múltiplos robôs conseguirá se reorganizar por meio de uma comunicação eficiente, a fim de continuar a tarefa prevista, ao contrário do sistema de um único robô.

Devido a essas interações são necessários mecanismos de controle robusto coordenando o sistema. Tais sistemas são complexos e fundados no princípio de que cada

elemento possui a capacidade de escolha sobre suas próprias ações, fazendo com que o processo de comunicação seja crucial para o bom funcionamento do conjunto.

Dessa forma, a comunicação possui destaque nesses sistemas. Supondo casos em que a comunicação primária de tais sistemas seja comprometida, é interessante que os membros não entrem em colapso. Assim, a comunicação visual é uma alternativa interessante nesse cenário, visto que ela tem pouca dependência do meio para se propagar e é distribuída, ou seja, a informação é disponibilizada para todos os membros simultaneamente.

1.4. Organização do trabalho

O restante desse documento está estruturado da seguinte maneira: o capítulo 2 apresenta o referencial teórico para o desenvolvimento desse trabalho. O capítulo 3, descreve a metodologia utilizada no desenvolvimento e suas respectivas implementações. O capítulo 4 detalha os resultados obtidos com os experimentos. Finalmente, o capítulo 5 apresenta as considerações finais com base nos resultados.

2 Referencial Teórico

Este capítulo fundamenta todos as técnicas, métodos e análises utilizados no trabalho e fornece os fundamentos para a completa compreensão do documento.

2.1. Drone - VANT

O termo drone refere-se a todo tipo de veículo aéreo que pode ser controlado nos 3 eixos cartesianos e que não necessite de pilotos embarcados para serem guiados (DEPARTAMENTO DE CONTROLE DO ESPAÇO AÉREO, 2017). Primeiramente, foram utilizados na área militar, pois não colocavam soldados em risco em missões de alta periculosidade (TICE, 1991), ficando encarregados de exercer os trabalhos denominado “*Three Ds*” (Três Ds em português): “*dull, dirty e dangerous*” que significam tedioso, sujo e perigoso, respectivamente (LAMY, MASKER, *et al.*, 2016). A Figura 1 mostra um drone militar atual.



Figura 1 - Drone militar RQ-4 Block 10 Global Hawk (GRUMMAN, 2018)

Atualmente, os drones estão cada vez mais acessíveis economicamente, tornando-os opções de uso em diferentes áreas de negócio. Algumas dessas áreas são: filmagem e fotografia aérea, agricultura de precisão, logística, busca e resgate, além de pesquisas acadêmicas (CRAIG, 2018).



Figura 2 - Drone Prime Air da Amazon carregando um pacote (AMAZON, 2018)

Gigantes da tecnologia como Amazon e Google têm investido muito nesses dispositivos. Como exemplo, pode-se citar o projeto *Amazon Prime Air* (AMAZON, 2018), como mostra a Figura 2, e o projeto *Wing* (X DEVELOPMENT LLC, 2017), onde a intenção é revolucionar o mercado de logística mundial pelo uso de drones.

Uma pesquisa realizada pelo Esticast Research & Consulting mostrou que a expectativa de crescimento do mercado de drones comercial é atingir 3,5 bilhões de dólares até 2024 (PHILLIPS, 2017), mostrando a importância que esses veículos terão em nosso dia-a-dia.

2.2. *Enxame de robôs*

Enxame de robôs é a designação dada a um ambiente onde um conjunto de robôs operam de forma conjunta e integrada. Foram inspirados na organização de insetos sociais, como as formigas, abelhas, cupins, entre outros. Tal inspiração ocorre pela capacidade desses insetos se comunicarem com todo o enxame de forma clara e distribuída (TAN e ZHENG, 2013). Algumas características para definir um enxame de robôs são (NAVARRO e MATÍA, 2012):

- Grande número de indivíduos – de dezenas a milhares de robôs, com sua identificação individual para comunicação e cooperação.
- Controle descentralizado e autônomo – não possuem controle centralizado ou líder. Os membros interagem entre si e com o ambiente.

- Homogeneidade do grupo – os robôs têm estrutura física e tecnológica bem parecida, uma das razões para o conceito de enxame.
- Escalabilidade e robustez – adicionar novas unidades melhorará o desempenho global do sistema enquanto que perder algumas unidades não causará colapso no enxame.

2.3. *Comunicação entre membros de um enxame de robôs*

Máquinas que operam de forma colaborativa necessitam se comunicar. A forma de comunicação varia de acordo com cada tipo de enxame e tipo de tarefa proposta. Ambiente de atuação, tamanho dos robôs e limitação orçamentaria também são determinantes no tipo de comunicação. Bluetooth, wireless, estigmergia¹ e leds infravermelhos são alguns meios de comunicação que podem ser utilizados (JASMINE, 2015).

Os principais aspectos que devem ser considerados para a escolha de um meio de comunicação são (JASMINE, 2015):

- Alcance da comunicação: se o alcance for pequeno, como o bluetooth, pode ocorrer perda de comunicação. Por outro lado, se for muito grande pode causar sobrecarga na comunicação.
- Área de comunicação: idealmente seria a cobertura de 360º na comunicação para que isso não seja um limitante primário.
- Tamanho das mensagens: pequenas mensagens podem sobrecarregar a comunicação, enquanto que mensagens grandes possuem mais chances de serem corrompidas.
- Tempo de propagação: tempo gasto para a mensagem ser entregue para todos os elementos do enxame.
- Interferências: é preciso entender o ambiente onde o enxame irá operar e quais são as possíveis interferências.

¹ Mecanismo que permite comunicação indireta. O agente se comunica com os demais alterando o estado do ambiente.

A comunicação de máquinas que operam de forma colaborativa é fundamental para a eficiência do trabalho, por isso a escolha correta da topologia de comunicação determina o quão eficiente o sistema de robôs será.

2.4. *Computação Bioinspirada*

A natureza faz as coisas de uma maneira incrível. Por trás dos fenômenos visíveis, existem inúmeras causas invisíveis que são fundamentais para a coordenação do mundo natural. Filósofos e cientistas têm observado tais fenômenos há séculos e, desde então, tentam entender, explicar, adaptar e replica-los em sistemas artificiais (SIDDIQUE e HOJJAT, 2015).

Esses fenômenos podem ser considerados a dialética da natureza, conceituando a evolução do mundo natural. Essa evolução, segue uma ordem distinta de complexidade pois o processamento das informações acontece de forma distribuída, auto organizada e sem controle central (LOPEZ-RUBIO, PALOMO e DOMINGUEZ, 2014).

Todos os fenômenos conhecidos ou parcialmente conhecidos até agora estão emergindo como novos campos de ciência e tecnologia, que estudam técnicas de resolução de problemas inspiradas pela natureza, bem como tentativas de compreender os princípios subjacentes de organismos naturais, físicos, químicos e biológicos que realizam tarefas complexas de maneira adequada com recursos e capacidades limitadas. Esse campo é denominado computação bioinspirada ou computação natural (DE CASTRO, 2007).

A computação bioinspirada é de natureza interdisciplinar, combinando ciência computacional com conhecimento de diferentes ramos das ciências, como por exemplo a física, a química, a matemática, a engenharia, a biologia, entre outros. Essa interdisciplinaridade permite o desenvolvimento de novas ferramentas computacionais, como algoritmos, hardwares, síntese de padrões, comportamentos e organismos (KARI e ROZENBERG, 2008).

Desse modo, um cientista chamado Karl Von Frisch, utilizando a natureza como inspiração, estudou o comportamento das abelhas, mais especificamente suas formas de comunicação. Assim, foi constatado que as abelhas possuem diversos modos de se

comunicarem umas com as outras. Um desses modos é a dança, o qual foi comprovado experimentalmente.

Foram definidos três tipos de dança: dança em círculo, dança do requebrado e dança em foice. Cada um desses tipos de dança possui sua função no alfabeto das abelhas. A intensidade com que a dança é realizada também é determinante para a definição do tipo de mensagem que a abelha dançarina quer transmitir (FRISCH[†], 1973).

Através de seus experimentos, Frisch descobriu que as abelhas utilizam o sol como referência para se localizarem. Assim, ele constatou que o eixo da dança em relação ao favo da colmeia indica a posição da fonte de alimento em relação ao sol (FRISCH[†], 1973).

2.5. *Processamento Digital de Imagens*

Processamento digital de imagens é o nome dado a uma técnica que realiza algumas operações em uma imagem, com o objetivo de obter uma imagem melhorada ou extrair informações relevantes dela. É oriunda do processamento de sinais, onde o sinal de entrada é uma imagem e o sinal de saída pode ser uma nova imagem ou alguma característica associada àquela imagem (ANBARJAFARI, 2018).

Existem inúmeras aplicações para o processamento de imagens, entre as quais pode-se destacar: restauração e realce de imagens, imagens médicas, transmissão e codificação, visão computacional, reconhecimento de padrões, segmentação de objetos, entre outros (GONZALEZ e WOODS, 2010);

Na literatura, existem diversas técnicas de segmentação de objetos. As técnicas utilizadas neste trabalho estão explicadas nas seções 2.5.1 e 2.5.2.

2.5.1. *Fluxo Óptico – Lucas e Kanade*

O fluxo óptico é o nome dado a técnica de estimação de movimento de duas imagens consecutivas em uma sequência de imagens. De forma genérica, essa técnica consiste na atribuição de um vetor para cada pixel analisado na imagem, contendo informações de amplitude em cada direção cartesiana, representando assim, a velocidade e a direção de movimentação da informação de um pixel para outro dentre as imagens subsequentes (WEDEL e CREMERS, 2011).

Existem várias abordagens para implementação dessa técnica, entre elas a proposta por Bruce D. Lucas e Takeo Kanade que ficou conhecido como método de Lucas e Kanade. Essa técnica admite que o deslocamento de um objeto entre dois frames sequenciais é pequena e aproximadamente constante e que a intensidade do pixel analisado e sua vizinhança são semelhantes (LUCAS e KANADE, 1981).

Deste modo, esse algoritmo não considera a cor das imagens de forma explícita, mas sim sua intensidade em escala de cinza. Assim, ao analisarmos um pixel específico entre dois frames sequencias é possível justificar qual a direção de tendência que o mesmo está se deslocando, com base na variação dos gradientes de intensidade de sua vizinhança (LUCAS e KANADE, 1981).

É uma técnica recomendada para variações moderadas de velocidade, e produz resultados sujeito a ruídos.

2.5.2. *Algoritmo de Harris*

O algoritmo de Harris é um método matemático utilizado na detecção de cantos em uma imagem. Como o próprio dicionário da língua portuguesa define, canto, como substantivo masculino, pode ser um ponto, superfície ou linha de convergência (DICIONÁRIO ONLINE DE PORTUGUÊS, 2018). Em termos práticos pode ser o limite entre dois objetos.

No campo do processamento de imagem, canto pode ser definido como uma grande variação de intensidade em todas as direções de sua vizinhança (SINHA, 2012). Para o cálculo dessa variação, é escolhida uma janela de teste. Essa janela deverá ser movida em todas as direções, a fim de verificar a diferença de intensidades entre a posição atual da janela e a posição seguinte. Esse princípio está ilustrado na Figura 03.

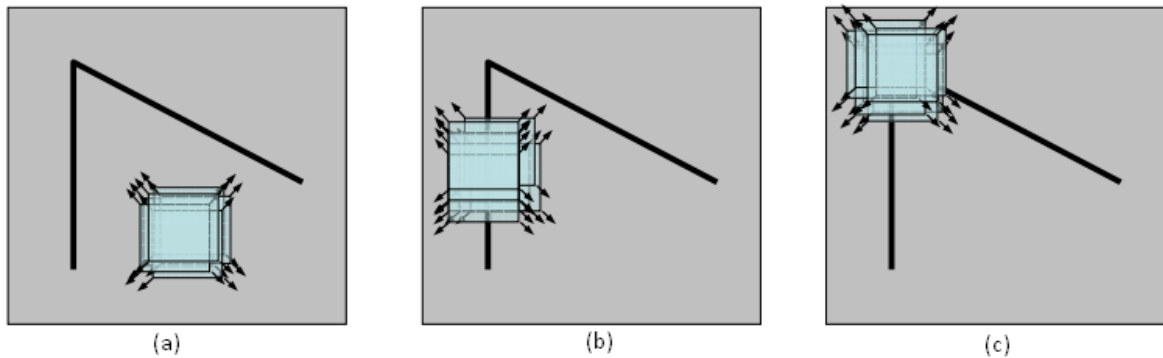


Figura 3 - Representação da implementação do algoritmo de Harris e sua variação de janela mostrando um objeto plano (a), uma borda (b) e um canto (c) (SINHA, 2012)

A localização de cantos nas imagens tem diversas aplicações, entre elas podemos destacar a reconstrução 3D de objetos, o rastreamento de movimento e o reconhecimento de objetos.

2.6. Reconhecimento de padrões

O reconhecimento de padrões tem por objetivo a classificação de objetos em um número de categorias ou classes. Algumas das principais definições em reconhecimento de padrões são as seguintes (THEODORIDIS e KOUTROUMBAS, 2008):

- **Padrão:** é uma entidade, objeto, processo ou evento, vagamente definido, que pode assumir um nome.
- **Classe:** conjunto de padrões que possuem características em comum.
- **Característica:** dado extraído de uma amostra por meio de medida ou processamento.
- **Classificação:** atribuir classes para as amostras, baseado em suas características.
- **Ruído:** distorção, falha ou imprecisão que ocorre na aquisição dos dados.
- **Classificadores:** utilizados para classificar ou descrever padrões a partir de um conjunto de características.

Há essencialmente duas maneiras de se classificar um padrão, a classificação supervisionada e a classificação não supervisionada. Na classificação supervisionada o padrão de entrada é conhecido e pertence a uma classe pré-definida pelo projetista do sistema. É também conhecido como aprendizado supervisionado. Já na classificação não supervisionada, não conhecemos o padrão, nem o número total de classes a serem encontradas durante a classificação. É conhecida como análise de agrupamentos (BIANCHI, 2006).

O projeto de um sistema de reconhecimento de padrões envolve as seguintes etapas: (i) aquisição de dados, (ii) pré-processamento dos dados, (iii) representação dos dados através das características, (iv) tomada de decisão com a escolha do classificador (BIANCHI, 2006).

A escolha do meio de aquisição dos dados, da técnica de pré-processamento, do tipo de representação e do método para tomada de decisão é dependente do tipo de problema. A escolha da abordagem para o reconhecimento de padrões não é tarefa simples e a experiência do projetista é importante.

Uma das técnicas que utiliza o reconhecimento de padrões é chamada de *Template Matching*. Essa técnica é utilizada para localizar partes de uma imagem que correspondem a um modelo dado (KADHM, 2014). Existem vários tipos de implementação que podem ser usadas para a análise de correspondência do modelo, como por exemplo a intensidade de cada pixel ou por regiões de interesse (COX, 1995).

Essa técnica é utilizada para análise de imagens médicas, na robótica, no rastreamento de veículos, na qualidade de produção, entre outros (KADHM, 2014).

O interesse na área tem crescido muito nos últimos anos, principalmente pelo uso da tecnologia em áreas como *Big Data* e *Machine Learning*.

2.7. Ferramentas

2.7.1. MATLAB

MATLAB é um software interativo usado na computação técnica. Ele integra computação, visualização e programação em uma IDE de fácil entendimento e uso. É uma linguagem matricial aproximando as expressões do mundo da matemática computacional (THE MATHWORKS INC., 2018)

Amplamente utilizado em ambientes universitários para cursos de matemática, engenharia e ciência. Na indústria é usado para pesquisa, desenvolvimento e análises de alta produtividade. Seus principais usos incluem modelagem, simulação e prototipagem de sistemas; análise, exploração e visualização de dados; processamento e digitalização de imagens; entre outras (ACKERMAN, 1997).

2.7.2. V-REP

O V-REP (*Virtual Robot Experimentation Platform*) é um simulador de robôs que possui um ambiente de desenvolvimento integrado. Baseado em uma arquitetura de controle distribuído onde cada objeto ou modelo pode ser controlado individualmente através de um script incorporado, um plugin, um cliente API, pelo ROS ou por uma solução personalizada. Isso torna o V-REP muito versátil e ideal para aplicações com vários robôs (COPPELIA ROBOTICS, 2018)

Os controladores podem ser escritos em C/C++, Python, MATLAB, Java, Lua ou Octave. É altamente utilizado para desenvolvimento de algoritmos para simulações de automação em fábricas, prototipagem, monitoramento remoto, entre outros (COPPELIA ROBOTICS, 2018);

3 Materiais e Métodos

Neste capítulo são apresentados os procedimentos utilizados a fim de identificar as movimentações do drone. Para isso, a metodologia foi separada nas seguintes etapas:

- Captura
- Segmentação
- Processamento
- Comparação
- Classificação

Portanto, o capítulo explora cada uma dessas etapas, descrevendo técnicas, métodos e conceitos utilizados na construção do trabalho.

3.1. *Captura - Movimentação do Drone*

Primeiramente, o ambiente de simulação foi utilizado com o único objetivo de validar o algoritmo de segmentação. Por isso, somente dois tipos de padrões de movimentos foram utilizados nesse ambiente.

Desse modo, dois drones foram utilizados para a realização dos experimentos e a configuração deles se deu de forma que um conseguisse visualizar o outro.

Para o ambiente virtual, o drone denominado DV1, foi programado de modo a se manter imóvel, observando e capturando a movimentação do segundo drone, denominado DV2.

Já o drone DV2 foi programado para realizar dois padrões de movimentos: oscilação vertical e horizontal. Esses padrões estão ilustrados na Figura 4.

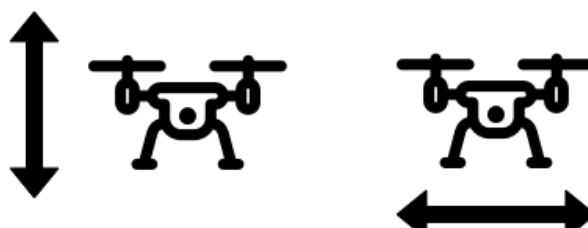


Figura 4 - Ilustração da movimentação do drone DV2. Oscilação vertical (a) e oscilação horizontal (b)

Para o ambiente real a configuração foi a mesma realizada na simulação. A única diferença se deve ao fato que o drone que se movimenta deveria realizar quatro padrões de movimentação, e não dois, como o realizado em ambiente simulado.

Assim, um drone, denominado DR1, se mantém imóvel de modo a observar e capturar a movimentação do segundo drone, denominado DR2. Enquanto que o drone DR2 realiza quatro padrões de movimentos, que são: oscilação vertical e horizontal, quadrado e T. Esses padrões estão ilustrados na Figura 5.

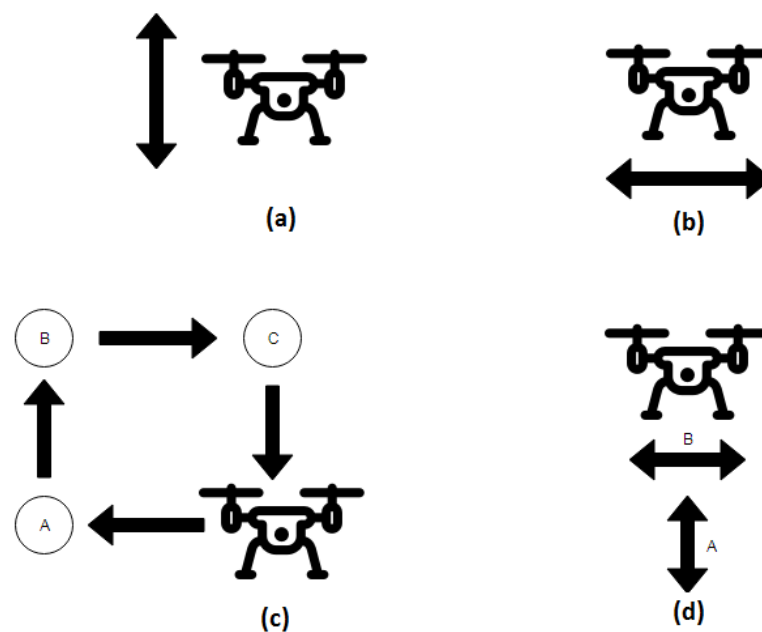


Figura 5 - Ilustração dos 4 movimentos realizados pelo drone DR2. Oscilação vertical (a), oscilação horizontal (b), quadrado (c) e T (d)

3.2. Segmentação

Com a finalidade de localizar objetos que estão em movimento, foi utilizado uma técnica de processamento de imagem denominada fluxo óptico. Essa técnica foi escolhida para selecionar os pontos (pixels) que estão se movendo em uma sequência de imagens (WEDEL e CREMERS, 2011).

O algoritmo utilizado foi o proposto por Lucas-Kanade (BOUGUET, 2000). Como visto anteriormente na seção 2, esse método compara dois *frames* de entrada, analisando os

pontos da imagem e associando um vetor em cada ponto contendo sua respectiva velocidade e direção. Esses vetores estão ilustrados na Figura 6.



Figura 6 - Vetores atribuídos com o algoritmo de Lucas e Kanade (COPPELIA ROBOTICS, 2018)

Pensando na otimização do uso do algoritmo de fluxo óptico, foi realizada uma pré-seleção de pontos antes da análise de Lucas-Kanade. Essa seleção foi implementada com o uso do Algoritmo de Harris (ROSTEN e DRUMMOND, 2006), que identifica pontos de canto na imagem sugerindo a presença de objetos.

Esse método pode apresentar ruídos na seleção do objeto em movimento. Dessa forma, afim de reduzir tais ruídos, assumiu-se os seguintes parâmetros de controle:

- Limitação do número de pontos a serem analisados pelo algoritmo Lucas-Kanade, em no máximo 200 pontos;
- O fator de representatividade daquele ponto, em que os pontos que estão abaixo do fator determinado são descartados. O valor da representatividade fixada foi de 2%.

O fator de representatividade foi definido como sendo o valor da amplitude da velocidade do ponto analisado, dividido pela soma dos valores das amplitudes de todos os pontos da amostra.

Uma vez reduzido os ruídos, ainda era necessário representar esses pontos como um único ponto material², pois o interesse estava na movimentação do objeto e não na sua dimensão. Para essa análise, os pontos foram conectados para obtenção de um polígono e a partir dele localizar o seu centroide, como pode ser observado na Figura 7.

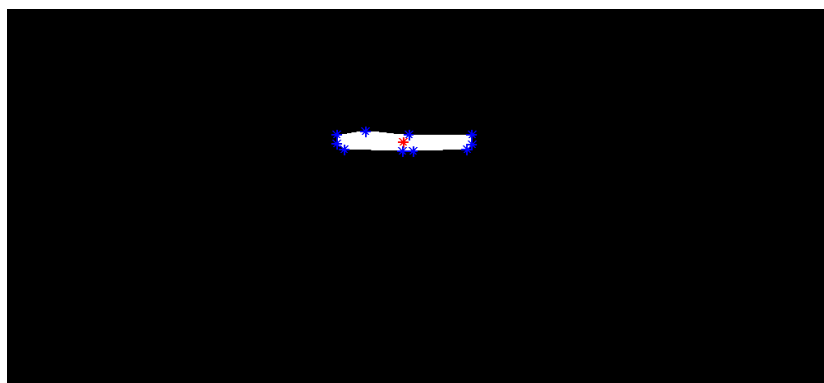


Figura 7 - Exemplo de segmentação do drone no ambiente de simulação (COPPELIA ROBOTICS, 2018)

Além da segmentação utilizando o algoritmo de Lucas-Kanade, foi desenvolvido uma segunda técnica baseada na subtração de imagens. Como será visto nos resultados apresentados na próxima seção, o algoritmo proposto por Lucas Kanade não apresentou resultados satisfatórios para imagens reais, enquanto que nas imagens simuladas o resultado foi aceitável, por isso a necessidade de se buscar uma outra solução para segmentação das imagens reais.

Dessa forma, considerando dois *frames* sequenciais de um objeto movendo-se na imagem, foi realizado a subtração desses *frames*. Admitindo que o objeto em movimento possui a maior variação de pixels do que o restante da imagem, o algoritmo de Harris foi utilizado para encontrar cantos na imagem resultante.

A função do algoritmo de Harris empregada retorna a posição do pixel relacionada ao canto encontrado. Como esse algoritmo ordena os cantos de modo decrescente em relação a sua relevância, ou seja, pixels com maiores chances de ser um canto estão nas primeiras posições, limitou-se a busca em dois cantos, os mais relevantes da imagem. Após a localização

² É um corpo com qualquer tamanho ou forma que tem por objetivo representar um objeto independente da sua dimensão.

desses, foi calculada a média da posição cartesiana de cada eixo para encontrar o ponto material que define o objeto. Um exemplo de uso desse método pode ser visualizado na Figura 8.

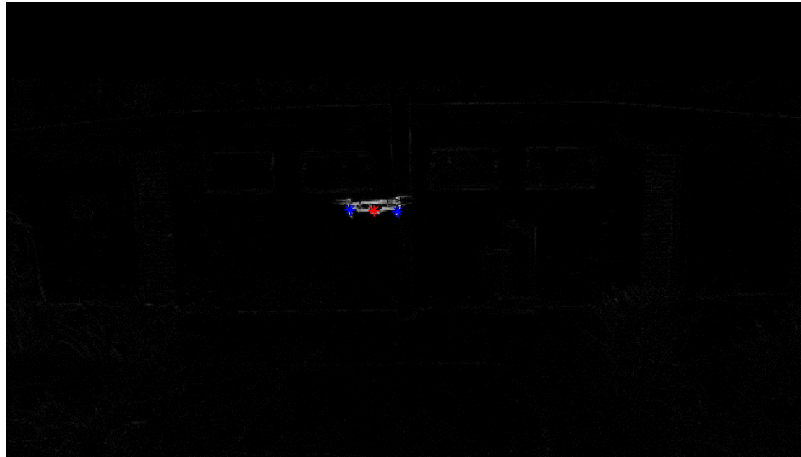


Figura 8 - Localização do drone no resultado da subtração de frames sequenciais em ambiente real

3.3. *Processamento – Assinatura dos movimentos*

Após a segmentação, foi obtido a posição do ponto material em cada *frame*, com isso, a trajetória foi definida como sendo as coordenadas x e y dessa posição ao longo do tempo. Como o movimento é espacial, foram geradas duas curvas de deslocamento para cada movimentação. Essas curvas também podem ser chamadas de assinaturas do movimento. Podemos ver um exemplo de assinatura ilustrado na Figura 9.

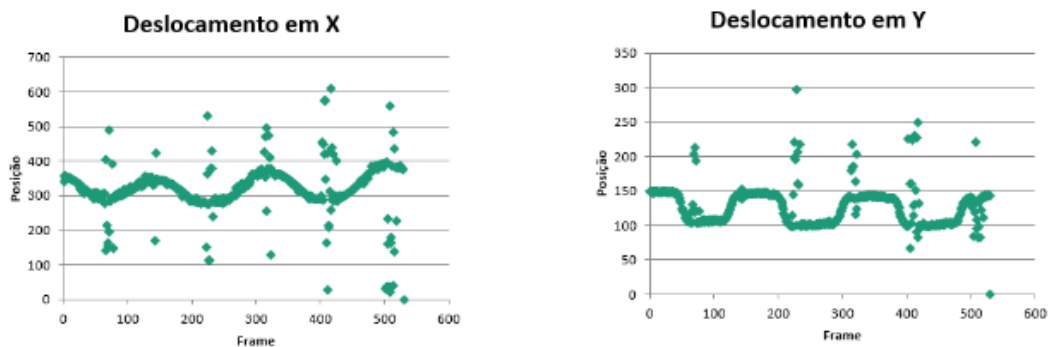


Figura 9 - Exemplo de assinatura de movimentação.

Como pode-se notar, era necessário realizar o tratamento dos dados, a fim de obter uma informação mais limpa da assinatura. Portanto, os dados foram suavizados e normalizados, para posteriormente serem classificados.

A suavização dos dados foi realizada utilizando o filtro de mediana, que é uma técnica bastante empregada na suavização de imagens. Ela consiste na atribuição de um novo valor para cada dado analisado, o qual é resultado da mediana da vizinhança desse dado. A vizinhança é escolhida conforme a necessidade da suavização, mas valores muito altos podem distorcer os dados originais.

Para que o sinal pudesse ser comparado a um modelo e a amplitude da movimentação não interferisse nessa comparação, os sinais foram normalizados pelo seu valor máximo, dessa forma a tendência do deslocamento foi preservada e a amplitude do movimento não influenciou nas análises. Um exemplo de sinal original e outro processado está ilustrado na Figura 10.

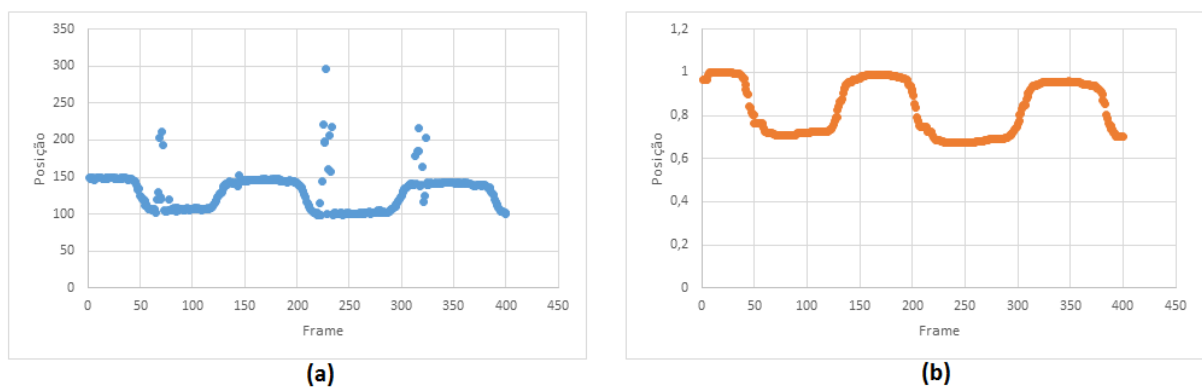


Figura 10 – Exemplo de um sinal original em (a) e sinal suavizado e normalizado em (b)

3.4. Comparação

Após a aquisição e tratamento dos dados contendo o deslocamento do drone ao longo do tempo, fez-se necessário definir qual o tipo de movimento analisado. Dessa forma, os dados foram comparados com modelos esperados de deslocamento utilizando uma técnica conhecida como *Template Matching*, como já descrito na seção 2.

Como cada movimento possui características diferentes, tais como período e amplitude, os modelos esperados de deslocamento, assim como suas respectivas análises, foram tratados de forma individual.

Devido as interferências do ambiente, como o vento, e ao fato do drone ter sido controlado de forma manual, os movimentos realizados apresentaram períodos e amplitudes irregulares, como pode ser visto na Figura 11. Assim, foi necessário inferir os modelos ideais.

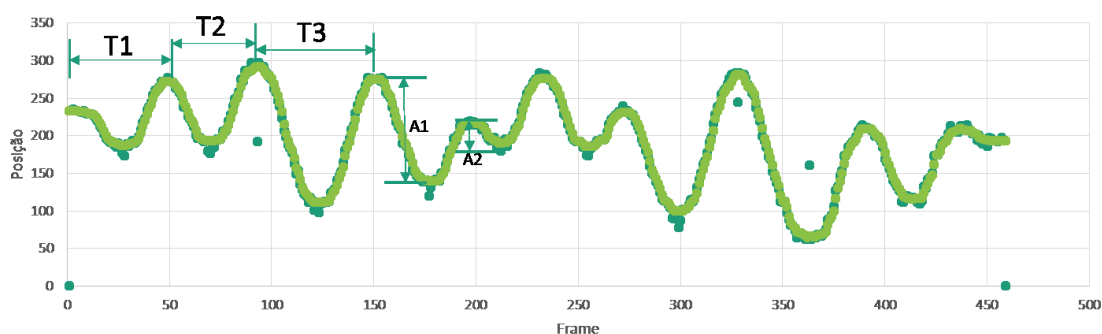


Figura 11 - Exemplo de um sinal com períodos e amplitudes irregulares.

Para a inferência, tanto o período quanto a amplitude foram calculadas como sendo a média de cada uma das respectivas grandezas analisadas. Por fim, o tamanho do modelo deveria ser o suficiente para a realização de um período completo do movimento.

De posse de todos os modelos de todos os movimentos, estabeleceu-se que o tamanho mínimo que uma amostra de comparação deveria possuir seria de duas vezes o tamanho do período do modelo, pois dessa forma seria garantido que ao menos um movimento completo estaria presente na amostra.

Após essas considerações, o conjunto de dados global de cada movimento foi dividido por duas vezes o tamanho do período e arredondado para o menor inteiro mais próximo. Assim, foram geradas três pequenas amostras, de mesmo tamanho, de cada movimento.

Essas amostras foram divididas em dois conjuntos de dados, o conjunto de dados de treinamento e o conjunto de dados de teste. O conjunto de treinamento contém as duas primeiras amostras, já o de teste a última amostra. A Figura 12 ilustra estas explicações.



Figura 12 - Exemplo das amostras obtidas após a divisão dos dados globais por duas vezes o tamanho do modelo inferido.

Para a comparação das amostras com o modelo inferido, utilizou-se o erro médio quadrático como medida de qualidade. Para tanto, a análise seguiu o seguinte algoritmo:

1. O modelo é posicionado no início da amostra analisada;
2. Subtração posicional de cada elemento da amostra por cada elemento do modelo em módulo;
3. Obtenção da média dos erros (soma de todos os valores dividido pelo tamanho do modelo);
4. Deslocamento de uma posição (frame) do modelo;
5. Repetir passos 1 a 4 até que a posição atual mais o tamanho do modelo seja maior que o tamanho da amostra ($ap + ts > ss$);
6. Obter a menor média da análise.

Como cada amostra possuía idealmente no máximo dois períodos completos, foi estabelecido que o menor erro médio obtido daquela amostra seria o valor de limiar para determinar o *matching* do modelo. Dessa forma, todos os modelos foram testados com todas as amostras de assinatura dos movimentos do conjunto de treinamento. Com isso, era esperado que onde a amostra não correspondesse com o modelo, o valor de erro seria maior.

Como o conjunto de treinamento possui duas amostras de dados de cada deslocamento, o erro limiar final foi definido como sendo a média dos erros individuais. Esses resultados estão apresentados no capítulo 4.

3.5. *Classificação*

Como cada movimento é uma combinação de deslocamentos nos eixos x e y, se faz necessário definir a ordem de comparação dos modelos. Para a análise realizada neste trabalho ficou definido a comparação primária em x e posteriormente em y, ou seja, o primeiro modelo a ser testado será sempre no deslocamento em x.

Caso a busca pelo modelo analisado em x retorne positivo, então verifica-se se o modelo em y do mesmo movimento também retorne positivo, em caso afirmativo aquele é o movimento realizado, em caso negativo a busca prossegue.

Se a amostra analisada não retornar *matching* em x com nenhum modelo, então aquele não é um movimento conhecido.

4 Resultados e Discussões

Neste capítulo são apresentados os resultados obtidos com as técnicas de segmentação utilizadas, as curvas de deslocamento obtidas após o processo de suavização e normalização dos dados e quais foram os resultados da classificação desses dados através da técnica de *template matching*.

Primeiramente foi utilizado um ambiente virtual para validar a segmentação proposta, para tanto será descrito nesse capítulo como se deu a configuração deste ambiente. Uma vez validada a segmentação através dos resultados obtidos no ambiente virtual partiu-se para verificação do desempenho da segmentação em ambiente real. As subseções a seguir detalharão todos os experimentos realizados e seus respectivos resultados.

4.1. Ambientes de captura das imagens

4.1.1. Ambiente de Simulação

Para aquisição das imagens necessárias a fim de realizar a segmentação, foi montado um ambiente de simulação. A montagem desse ambiente se deve ao fato do custo associado a validação de teorias ser menos oneroso em ambientes simulados do que em ambientes reais.

Dessa forma, para a construção desse ambiente, foi utilizado o simulador de robôs V-REP. Esse simulador possui uma biblioteca padrão de componentes, o que possibilita criar uma ampla variedade de cenários para simulação. O simulador está ilustrado na Figura 13.

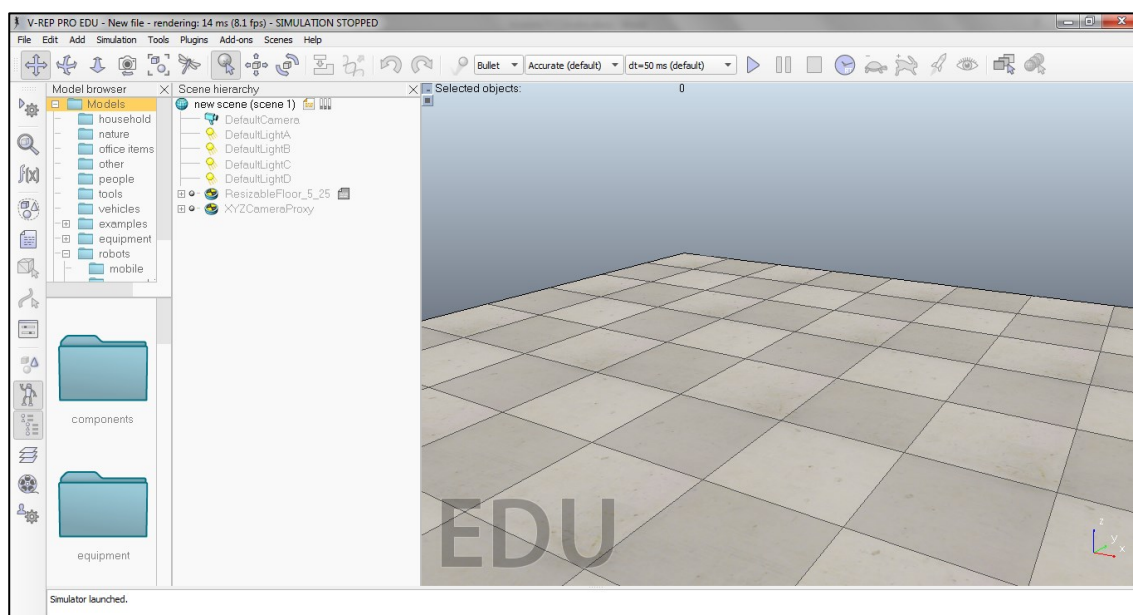


Figura 13 - Ambiente de simulação V-REP (COPPELIA ROBOTICS, 2018)

Com o auxílio da biblioteca de componentes, foi utilizado o modelo de um robô móvel chamado Quadricopter, como ilustrado na Figura 14. O Quadricopter possui um sistema de controle para movimentação que se aproxima ao de um drone real, o que é um atributo importante para o projeto proposto.

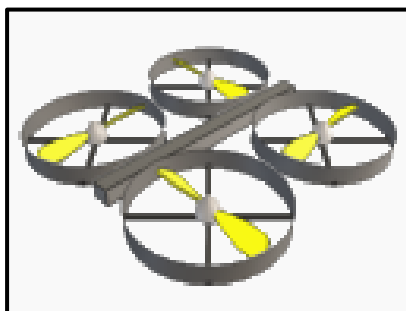


Figura 14 - Modelo Quadricopter, fornecido por Eric Rohmer (COPPELIA ROBOTICS, 2018)

Buscando aproximar a simulação a casos de usos reais, dois tipos de cenários foram criados, um em ambiente aberto e outro em ambiente fechado. O cenário em ambiente aberto representa a busca por objetos no deserto, já o de ambiente fechado, representa a busca pela saída de um labirinto. Os cenários criados estão ilustrados nas Figuras 15 e 16, respectivamente.



Figura 15 - Quadricopter no deserto (COPPELIA ROBOTICS, 2018)



Figura 16 - Quadricopter no labirinto (COPPELIA ROBOTICS, 2018)

4.1.2. *Ambiente Real*

A captura de imagens reais é interessante para validar o que foi obtido com o simulador. Ambientes reais são imprevisíveis e na maioria das vezes possuem ruídos não reproduzidos em ambiente simulados.

Como o drone utilizado no ambiente simulado era do modelo Quadricopter, a captura das imagens foi realizada com o drone MAVIC PRO, representado na Figura 17.



Figura 17 - Drone MAVIC PRO (DJI, 2018)

Esse drone foi escolhido por oferecer recursos de estabilização da imagem e também pela rápida resposta aos comandos do controle. Suas principais características técnicas são:

- Estabilizador ótico nos 3 eixos;
- Câmera de 12MP;
- Gravação em 30 frames/s.

Para validação do algoritmo de segmentação, o local para captura das imagens deveria ter vários elementos com diferentes texturas e cores. O local escolhido foi a frente de um vestiário de campo de futebol society localizado no condomínio fechado Gávea Paradiso, em Uberlândia. Na Figura 18 pode ser visualizado o local de captura.



Figura 18 - Local de captura das imagens reais

4.2. Segmentação

4.2.1. Simulação

Na simulação realizada no software V-REP, como já explicado na seção 4.1, foram montados dois tipos de ambiente para análise, o interno e o externo. Ambos os ambientes tiveram a mesma configuração, dois drones, onde um ficaria estático observando outro que se movimenta, como descrito na seção 3.1.

Como não há muitos elementos relevantes na imagem, além do próprio drone, o algoritmo foi implementado sem muitos problemas. O método de fluxo óptico utilizado para segmentar o drone foi o de Lucas-Kanade modificado. Os resultados obtidos para ambos os ambientes estão apresentados na Figura 19.



Figura 19 - Vetores segmentando o drone através do algoritmo de Lucas-Kanade. Ambiente interno (esq.) e ambiente externo (dir.)

4.2.2. Real

Para validarmos o algoritmo em situações reais de uso, utilizamos dois drones para a captura das imagens reais, como descrito no capítulo 3.

Com o uso do mesmo algoritmo utilizado na simulação, o resultado foi bastante insatisfatório. A variedade de formas e texturas, bem como a interferência do meio (em grande parte causado pelo vento) contribuíram para o mau desempenho do algoritmo. Esse resultado está ilustrado na Figura 20.



Figura 20 - Ruído na segmentação utilizando a técnica de fluxo óptico de Lucas e Kanade

De posse desse resultado, partimos para uma nova abordagem, conforme está explicada no capítulo 3. Como o objetivo do trabalho não engloba estimar a velocidade que objeto se encontra, usamos o conceito básico de segmentação, que é a subtração de imagens a fim de solucionar o problema. Dessa forma, era esperado que a diferença entre as imagens fosse exatamente a do objeto em movimento.

Com o auxílio do algoritmo de Harris para detecção de possíveis objetos, e após alguns testes, notamos que em 90% das vezes ele conseguia localizar o drone em no mínimo dois pontos. Após essa análise, limitamos a busca do algoritmo de Harris para no máximo 2 pontos. Com isso, o algoritmo retornava os pontos que possuem a melhor colocação como possíveis pontos de canto. Um exemplo de sucesso na localização do drone está ilustrado na Figura 21.

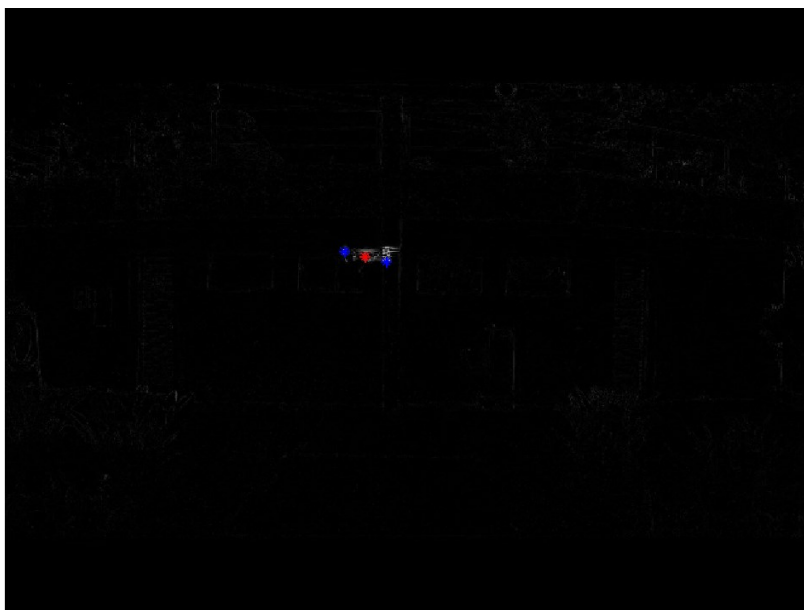


Figura 21 - Drone segmentado com o uso do algoritmo de Harris

Ficou evidenciado a melhora na segmentação utilizando esse método. O nível de ruído diminuiu bastante em relação a primeira análise e o ruído restante tem muita relação com a movimentação da câmera, ocasionada pelo vento. Na Figura 22 podemos verificar um caso em que o algoritmo não conseguiu localizar o drone.

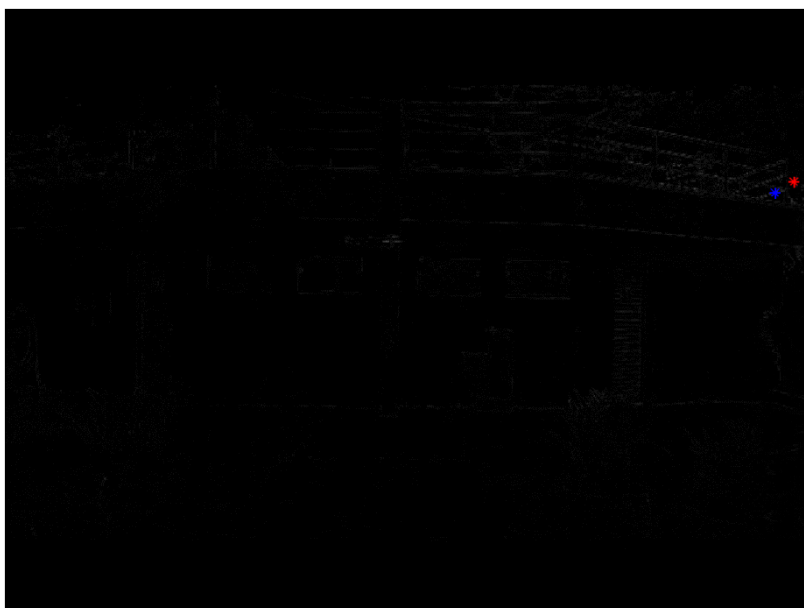


Figura 22 - Drone não encontrado na segmentação com o uso do algoritmo de Harris

4.3. *Análise da trajetória do drone em ambiente simulado*

Após a validação da segmentação, ainda era preciso representar o drone como um ponto material afim de obter as informações de posição através do tempo. Nesse momento, os pontos que obtiveram a maior velocidade na análise de Lucas-Kanade seriam considerados como pontos do drone. Esses pontos foram ligados para formação de um polígono e posterior cálculo do centroide, como pode ser visto nas Figuras 23 e 24.



Figura 23 - Drone segmentado (à esq.) e a localização do centroide (à dir.) no ambiente interno

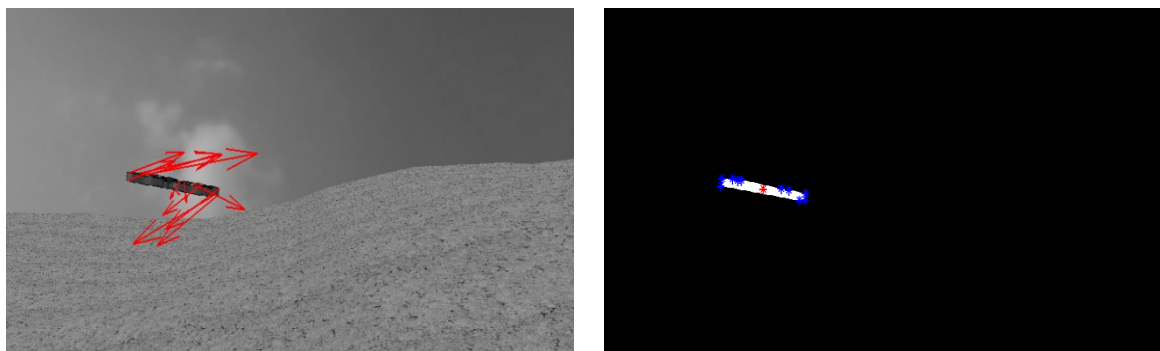


Figura 24 - Drone segmentado (à esq.) e a localização do centroide (à dir.) no ambiente externo

Nota-se alguns ruídos na seleção de pontos, principalmente em momentos de inversão de movimento. Nesses casos a velocidade de vários pontos se aproximam ou até mesmo não existe a variação do movimento do objeto entre frames, fazendo com que a condição de seleção de pontos não seja suficiente para distinguir objetos na imagem.

Os gráficos com a posição do drone ao longo do tempo (frames) evidenciam esses ruídos, que podem ser facilmente eliminados com o uso de filtro de mediana. Abaixo seguem as curvas de posição de ambos os ambientes.

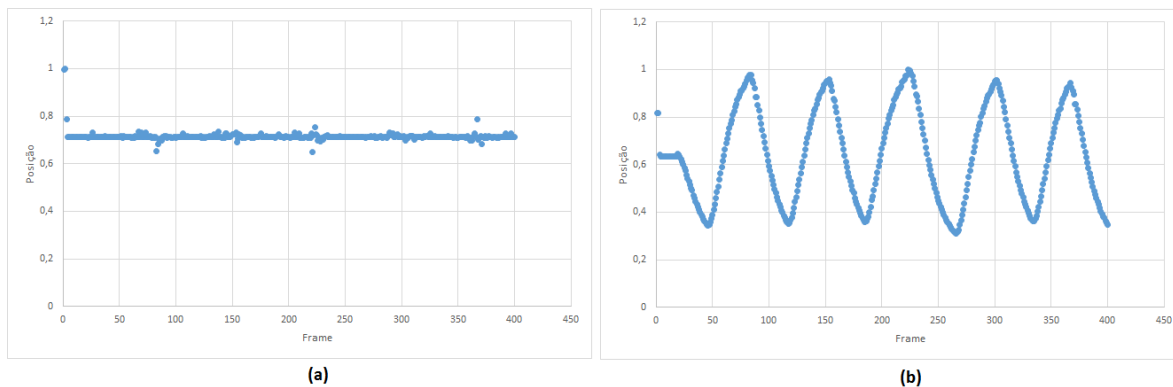


Figura 25 - Gráfico da posição por frame da oscilação vertical em ambiente simulado interno. (a) Deslocamento em x e (b) deslocamento em y.

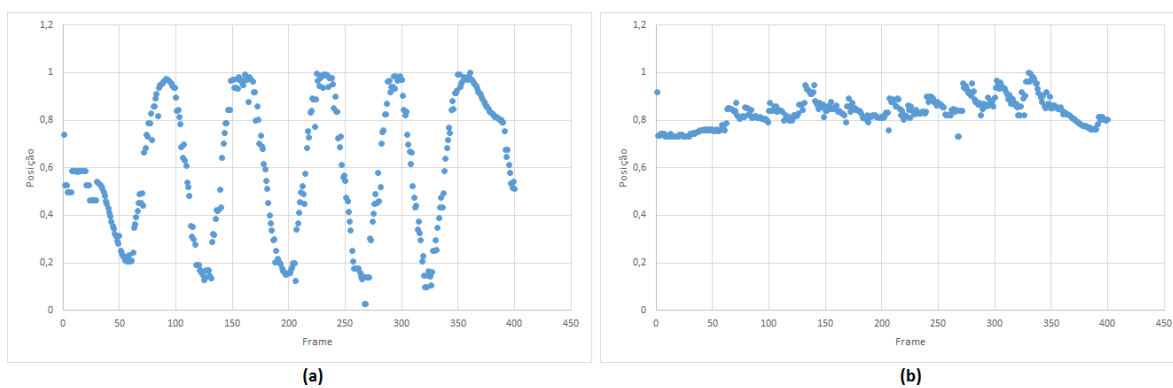


Figura 26 - Gráfico da posição por frame da oscilação horizontal em ambiente simulado interno. (a) Deslocamento em x e (b) deslocamento em y.

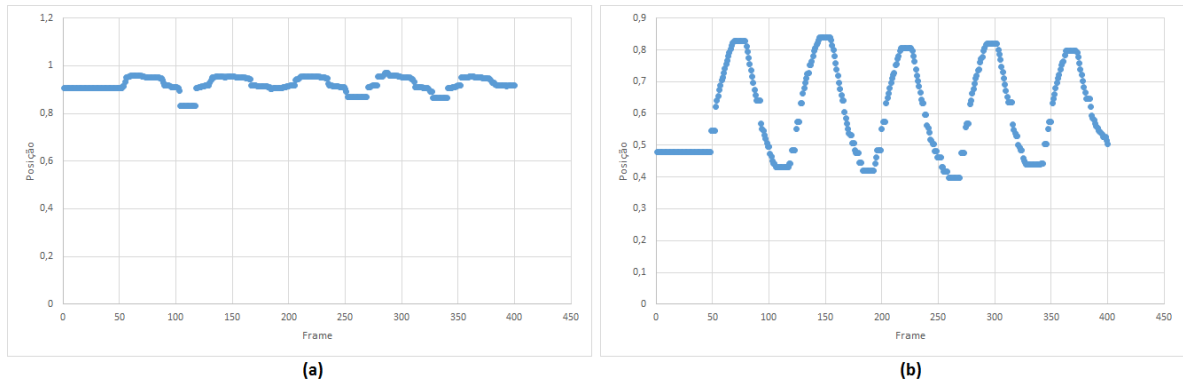


Figura 28 - Gráfico da posição por frame da oscilação vertical em ambiente simulado externo. (a) Deslocamento em x e (b) deslocamento em y.

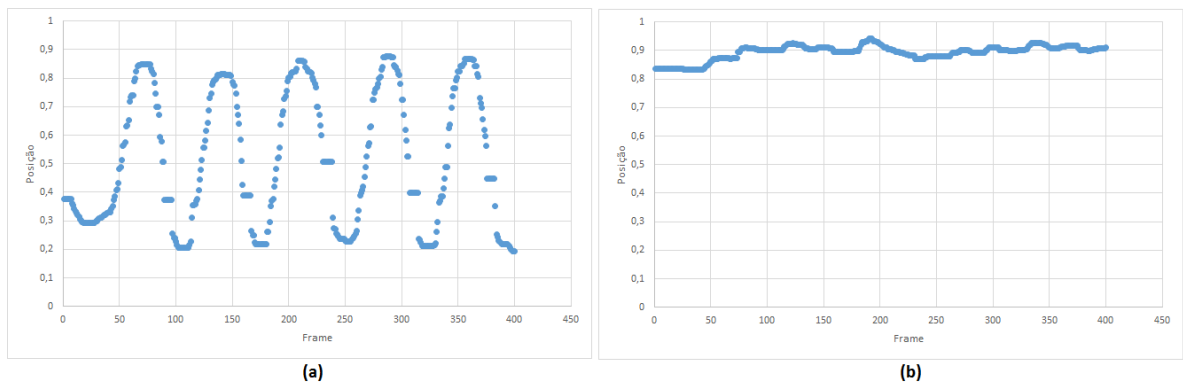


Figura 27 - Gráfico da posição por frame da oscilação horizontal em ambiente simulado externo. (a) Deslocamento em x e (b) deslocamento em y.

Com base nesses dados, foi demonstrada a validade desse algoritmo como forma de segmentação de objetos em movimento.

4.4. *Análise da trajetória do drone em ambiente real*

Com o resultado da segmentação, foi utilizado a média entre a posição absoluta dos pontos encontrados pelo algoritmo de Harris na determinação do ponto material. Dessa forma foi possível gerar dados suficientes para análise da movimentação do drone.

Para a análise real, quatro tipos de movimentos foram realizados: oscilação vertical e horizontal, movimentação em forma de quadrado e em forma de T.

Vale ressaltar que os dados originais da segmentação possuem relativa quantidade de ruídos, por isso foi realizado a suavização com uso de filtro de mediana em todas as análises. Um exemplo dessa suavização pode ser visto na Figura 26.

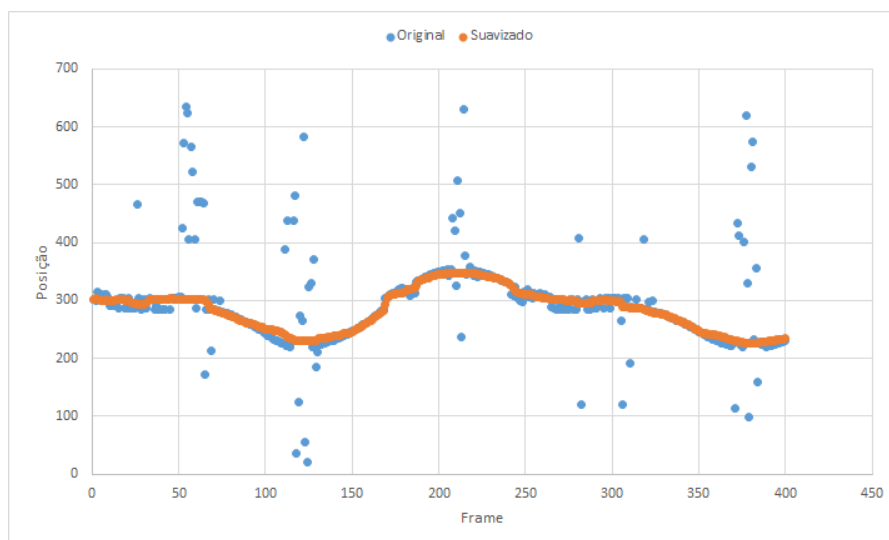


Figura 29 - Gráfico da posição por frame da movimentação em T em ambiente real mostrando a suavização dos dados.

Após a suavização, todos os movimentos foram normalizados na amplitude como descrito no capítulo 3 e os resultados estão apresentados abaixo.

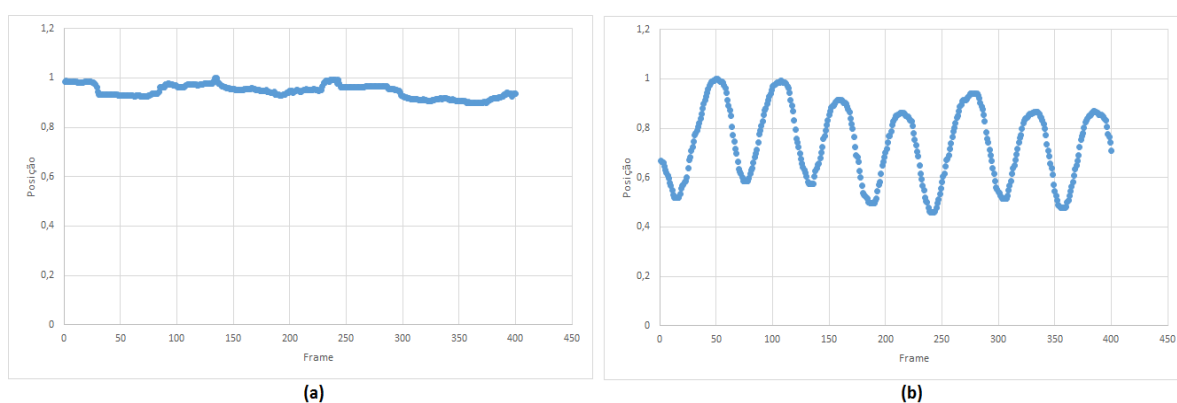


Figura 30 - Gráfico da posição por frame da oscilação vertical em ambiente real. (a) Deslocamento em x e (b) deslocamento em y.

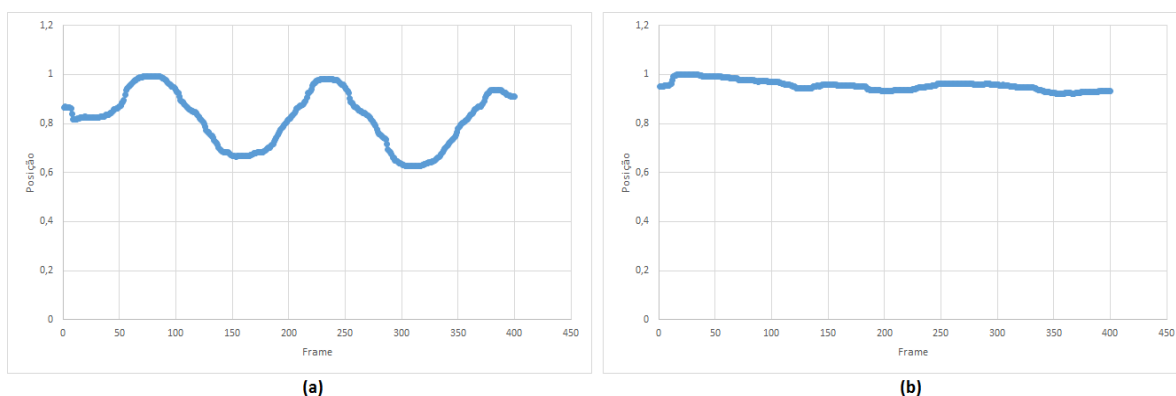


Figura 32 - Gráfico da posição por frame da oscilação horizontal em ambiente real. (a) Deslocamento em x e (b) deslocamento em y.

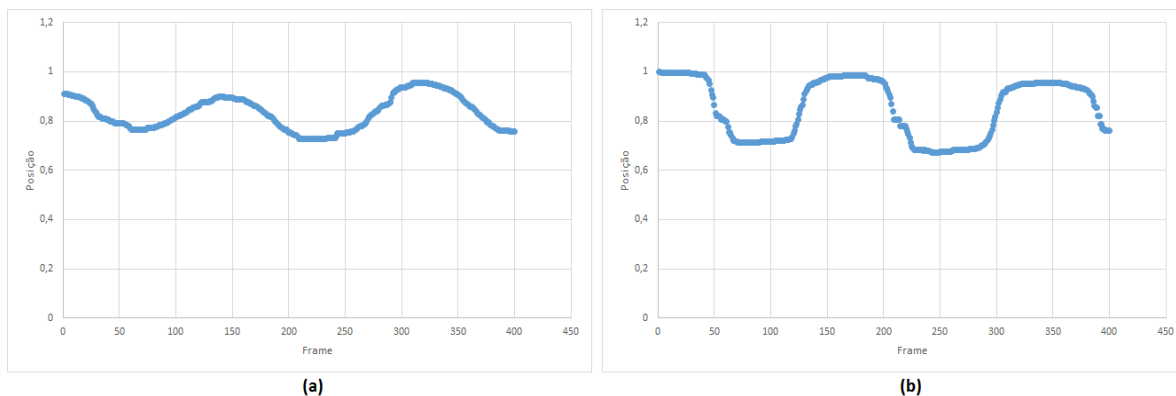


Figura 31 - Gráfico da posição por frame da movimentação em quadrado em ambiente real. (a) Deslocamento em x e (b) deslocamento em y.

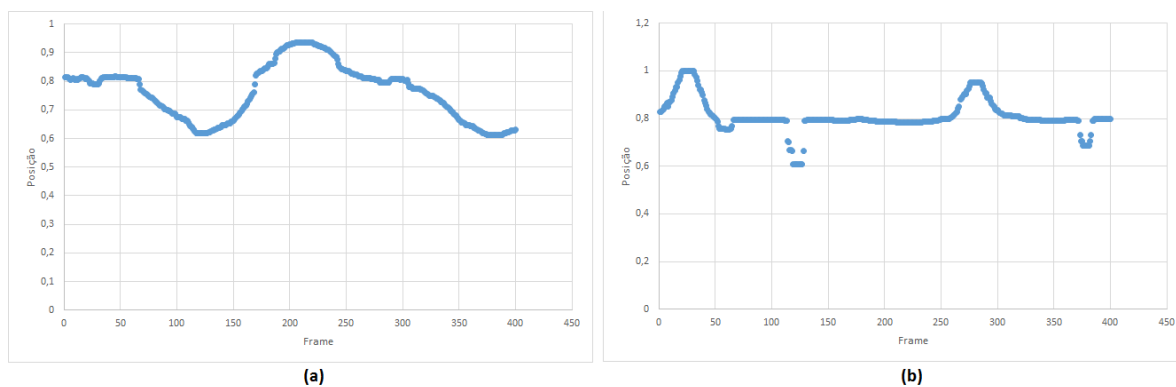


Figura 33 - Gráfico da posição por frame da movimentação T em ambiente real. (a) Deslocamento em x e (b) deslocamento em y.

Como pode-se notar, fica claro a tendência dos deslocamentos mostrado nos gráficos. Alguns movimentos possuem um período maior do que os outros devido a sua complexidade de execução. De qualquer forma, o algoritmo de segmentação foi validado pelos dados.

4.5. Classificação dos movimentos

Para a classificação dos movimentos foi utilizado a técnica de *template matching*, discutida na seção 2. Primeiramente buscou-se o modelo ideal do *template* para cada tipo de movimento.

Como explicado na seção 3, os dados dos movimentos foram separados em dois conjuntos, denominados de conjunto de treinamento e conjunto de testes. O conjunto de treinamento foi utilizado para inferir trajetórias ideais de cada movimento, visto que o controle do drone aconteceu de forma manual, ou seja, foi humanamente controlado.

Ainda com base nos dados de treinamento, foi definido os limiares de acerto para cada movimentação que foram determinados como sendo a média dos erros de cada amostra. A Tabela 1 mostra os menores erros obtidos com cada modelo para cada um dos movimentos analisados.

Tabela 1 – Valores de erros mínimos obtidos do conjunto de treinamento

Modelo	Vertical				Horizontal			
	Conjunto 1		Conjunto 2		Conjunto 1		Conjunto 2	
	x1	y1	x2	y2	x1	y1	x2	y2
vx	0,030383	0,187768	0,034908	0,200146	0,058168	0,008015	0,099892	0,004995
vy	0,135205	0,020318	0,141184	0,094091	0,117629	0,141521	0,112492	0,138229
hx	-	-	-	-	0,021702	0,176377	0,089396	0,176062
hy	-	-	-	-	0,191767	0,024774	0,215504	0,009983
sx	-	-	-	-	0,040092	0,119735	0,124363	0,132592
sy	-	-	-	-	0,077299	0,131629	0,119206	0,143326
tx	-	-	-	-	0,089323	0,168662	0,068461	0,17946
ty	-	-	-	-	0,081266	0,139329	0,117893	0,139609

Continuação da Tabela 1

Modelo	Quadrado				T			
	Conjunto 1		Conjunto 2		Conjunto 1		Conjunto 2	
	x1	y1	x2	y2	x1	y1	x2	y2
vx	0,026205	0,00365	0,017287	0,081715	0,031304	0,129988	0,022966	0,115431
vy	0,103123	0,0821	0,109685	0,10294	0,080237	0,049364	0,106574	0,067651
hx	0,068845	0,043388	0,053919	0,045362	0,096532	0,094955	0,058815	0,090301
hy	0,114158	0,096429	0,067943	0,141841	0,13583	0,199876	0,112703	0,175286
sx	0,021986	0,048555	0,023189	0,072067	0,058135	0,096914	0,0548	0,079993
sy	0,040624	0,04233	0,043524	0,080147	0,085755	0,121791	0,039359	0,098897
tx	0,0916	0,103211	0,139477	0,093753	0,098546	0,100987	0,02802	0,090526
ty	0,079162	0,103296	0,082981	0,114716	0,110179	0,046373	0,068814	0,036499

Os conjuntos de dados da movimentação vertical não puderam ser analisados pelo modelo de nenhum outro movimento, pois o tamanho em frames do conjunto era menor do que o tamanho dos demais modelos. Com base nos resultados obtidos, conseguimos calcular a média dos erros de cada conjunto, sendo esses apresentados na Tabela 2.

Tabela 2 - Média dos erros de cada modelo

Modelo	Média dos erros
vx	0,032645661
vy	0,057204464
hx	0,055549104
hy	0,017378813
sx	0,022587539
sy	0,061238661
tx	0,063283
ty	0,041436054

Os valores apresentados na Tabela 2 correspondem ao limiar de acerto de cada movimentação, ou seja, ao testar um conjunto de dados que contenha uma oscilação vertical é esperado que o erro encontrado seja menor que 0,0326 em x e menor que 0,0572 em y. Dessa forma, é possível afirmar que o movimento que aqueles dados retratam é uma oscilação vertical.

De posse desses limiares, utilizamos o conjunto de teste para verificar a validade desses valores. A Tabela 3 contém os erros obtidos de cada modelo em cada movimentação.

Tabela 3 - Valores de erros mínimos obtidos do conjunto de teste

Modelo	Vertical		Horizontal		Quadrado		T	
	x3	y3	x3	y3	x3	y3	x3	y3
vx	0,023567	0,188498	0,12094	0,00191	0,018536	0,02343	0,027719	0,086886
vy	0,130501	0,04609	0,100518	0,0882	0,108928	0,080592	0,113753	0,078628
hx	-	-	0,039748	0,18465	0,07092	0,040353	0,081474	0,096744
hy	-	-	0,237513	0,008062	0,107203	0,14498	0,131552	0,143433
sx	-	-	0,097987	0,114354	0,025119	0,033345	0,100899	0,075854
sy	-	-	0,140554	0,064828	0,049756	0,027967	0,059289	0,097938
tx	-	-	0,115717	0,138922	0,082219	0,104791	0,028919	0,083364
ty	-	-	0,098278	0,16471	0,063044	0,097988	0,08729	0,035401

As células realçadas em vermelho representam valores que ficaram abaixo do limiar para aquele modelo. Como pode ser observado, o modelo de **vx** teve grande quantidade de falsos positivos, apresentando valores até abaixo do obtido pelo conjunto em que era esperado o menor valor, a movimentação vertical em x. Entretanto, como um acerto é definido como sendo uma combinação de acertos em ambos os eixos, a movimentação vertical foi encontrada.

Analisando os demais movimentos, concluímos que a movimentação em quadrado não foi reconhecida. Nenhum conjunto conseguiu ficar abaixo do limiar estabelecido para o modelo sx. Pode-se notar, no entanto, que o valor obtido no conjunto esperado ficou bem próximo do limiar enquanto que nos demais movimentos os valores ficaram mais distantes. É provável que esse erro esteja associado a uma amostra de teste com muita interferência ou que o limiar restrinja muito a qualidade da amostra.

Para todos os outros movimentos o algoritmo se mostrou funcional, reconhecendo o movimento correto. Outro ponto importante é que não houve nenhum caso onde mais de um movimento foi reconhecido. Dessa forma, como resultado final temos três acertos e um erro, fazendo com que a taxa de assertividade do algoritmo fosse de 75%.

5 Conclusão

A utilização de enxames de robôs será corriqueira em um futuro próximo. A disponibilidade de acesso a esse tipo de tecnologia facilitará a execução de tarefas complexas e perigosas, reduzindo o grau de exposição de humanos a riscos.

O trabalho conseguiu demonstrar a viabilidade de um conceito de comunicação visual como forma alternativa de interação entre membros desses enxames, que poderá ser utilizado como contingência em caso de falha.

Primeiramente, o desafio era encontrar uma forma de segmentação de objetos em movimento. Nos testes realizados na simulação, o algoritmo de Lucas-Kanade modificado, já utilizado para segmentar drones em ambiente de simulação (CUNHA, 2016), se mostrou eficiente. Os resultados obtidos foram bastante satisfatórios e relevantes, demonstrando mais uma vez a capacidade desse algoritmo no uso de segmentação.

Entretanto, na utilização desse mesmo algoritmo com imagens reais, onde o nível de detalhes, formas e texturas aumentam consideravelmente, o algoritmo não foi satisfatório. Não foi possível obter a posição do objeto em movimento. Os limitadores de ruído foram insuficientes para o êxito da segmentação e, portanto, se mostrou ineficiente para segmentar objetos em movimento de imagens reais.

Desse modo, uma nova solução foi desenvolvida, baseada em um dos princípios de segmentação que é a subtração de imagens. Com essa subtração e o uso do algoritmo de Harris para detecção de objetos, foi obtido um resultado excelente. Considerando a imprevisibilidade do ambiente real, o algoritmo identificou o drone com médio nível de ruído fazendo com que o uso de filtro de alta frequência fosse suficiente para obtenção de uma informação limpa do sinal de deslocamento gerado.

Essa forma de segmentação de objetos em movimento se mostrou útil até mesmo em ambientes com muita interferência do ambiente.

Com a localização do drone definida em cada frame, foi possível gerar curvas de deslocamento da sua posição ao longo do tempo. De posse desses deslocamentos foi observado um ruído causado pela forma de aquisição das imagens. O fato do drone ter sido

controlado de forma manual, causou períodos de movimentos irregulares, bem como as respectivas amplitudes. Para reduzir essa interferência, medidas de normalização foram adotadas.

A classificação dos movimentos realizadas através de comparação de modelos pré-definidos se mostrou uma abordagem eficiente, visto que a porcentagem de acerto foi de 75%. Como não foi realizado estudos suficientes de reprodutibilidade dos movimentos, foi considerado a idealidade da segmentação como fator de controle.

O fato da velocidade e da amplitude da movimentação do drone não ser controlada, faz com que a comparação de modelos seja dificultada. Um controle dessas características tende a melhorar a eficaz dessa abordagem, pois os modelos de comparação serão mais bem definidos.

Essa forma de classificação utilizando comparação de modelos é apenas uma em universo de possibilidades para a identificação do tipo de movimento realizado, sendo esse uma das sugestões de trabalhos futuros.

Vale ressaltar que o modelo de segmentação não foi testado com o uso de vários objetos de interesse se movimentando no campo de visão, o que é válido também para trabalhos futuros.

Portanto, a realização desse trabalho demonstrou uma forma de comunicação visual específica, baseada na comunicação das abelhas, que poderá ser útil na comunicação entre enxames de robôs na execução de tarefas complexas.

Referências Bibliográficas

ACKERMAN, S. **What is Matlab**, 1997. Disponível em: <<http://cimss.ssec.wisc.edu/wxwise/class/aos340/spr00/whatismatlab.htm>>. Acesso em: 10 jan. 2018.

AMAZON. **Amazon Prime Air. Amazon**, 2018. Disponível em: <<https://www.amazon.com/Amazon-Prime-Air/b?node=8037720011>>. Acesso em: 02 jan. 2018.

ANBARJAFARI, G. **Introduction to image processing** | Digital Image Processing. University of Tartu, 2018. Disponível em: <<https://sisu.ut.ee/imageprocessing/book/1>>. Acesso em: 25 jan. 2018.

AYTUG, H.; KHIUJA, M.; VERGARA, F. E. **Use of genetic algorithms to solve production and operations management problems: a review**. [S.l.]. 2003.

BIANCHI, M. F. **Extração de características de imagens de faces humanas através de wavelets, PCA e IMPCA**. Universidade de São Paulo. São Carlos. 2006.

BOSQUE, M. M. **Implementação do Controle de Enxames de Robôs Utilizando a Hidrodinâmica de Partículas Suavizadas**. Universidade Federal de Minas Gerais. Belo Horizonte, p. 8. 2009.

BOUGUET, J.-Y. **Pyramidal Implementation of the Lucas Kanade Feature Tracker Description of the algorithm**. Intel Corporation - Microprocessor Research Labs. [S.l.], p. 9. 2000.

COLIN, R. R.; JONATHAN, E. R. **Genetic algorithms-Principles and perspectives**. [S.l.]. 2002.

COPPELIA ROBOTICS. **Coppelia Robotics V-REP: Create. Compose. Simulate. Any Robot**. Coppelia Robotics, 2018. Disponível em: <<http://www.coppeliarobotics.com/>>. Acesso em: 20 dez. 2017.

COX, G. S. **Template Matching and Measures of Match in Image Processing**. University of Cape Town. Cape Town, p. 24. 1995.

COXWORTH, B. **Canadian police save a man's life, using a drone**. New Atlas, 2013. Disponível em: <<https://newatlas.com/rcmp-quadcopter-locate-victim/27488/#gallery>>. Acesso em: 27 dez. 2017.

CRAIG, J. **Top 12 non military uses for drones**. Air Drone Craze, 2018. Disponível em: <<https://www.airdronecraze.com/drones-action-top-12-non-military-uses/>>. Acesso em: 7 jan. 2018.

CUNHA, F. O. **Emprego de Computação Bioinspirada no processo de comunicação entre membros de uma colônia de robôs**. Universidade Federal de Uberlândia. Uberlândia, p. 53. 2016.

DE CASTRO, L. N. **Fundamentals of natural computing: an overview**. [S.l.]. 2007.

DEPARTAMENTO DE CONTROLE DO ESPAÇO AÉREO. **ICA 100-40: Sistemas de Aeronaves Remotamente Pilotadas e o Acesso ao Espaço Aéreo Brasileiro**. Rio de Janeiro. 2017.

DICIONÁRIO ONLINE DE PORTUGUÊS. **Dicionário online de Português**, 2018. Disponível em: <<https://www.dicio.com.br/canto/>>. Acesso em: 11 jan. 2018.

DJI. Buy Magic Pro. **DJI Store**, 2018. Disponível em: <<https://store.dji.com/product/mavic-pro>>. Acesso em: 20 jan. 2018.

FRISCH[†], K. V. **Decoding The Language of The Bee**. Nobel Lecture. Munique: [s.n.]. 1973. p. 76-87.

GONZALEZ, R. C.; WOODS, R. E. **PROCESSAMENTO DIGITAL DE IMAGENS**. 3ª ed. ed. [S.l.]: Pearson, 2010.

GRUMMAN, N. Northrop Grumman Photos - **RQ-4 Block 10 Global Hawk**. Northrop Grumman, 2018. Disponível em: <http://www.northropgrumman.com/MediaResources/Pages/Photo.aspx?pid%3DGH-10020_008%26rel%3D%2F%26name%3DPhotos>. Acesso em: 15 jan. 2018.

JASMINE. **Swarm Communication**. SwarmRobot, 2015. Disponível em: <<http://www.swarmrobot.org/Communication.html>>. Acesso em: 10 dez. 2017.

KADHM, M. S. **Pattern Recognition: Template Matching**. University of Technology. Bagdá, p. 17. 2014.

KARI, L.; ROZENBERG, G. **Many facets of natural computing**. [S.l.]. 2008.

LAMY, S. et al. **Introduction to Global Politics**. 4ª. ed. New York: Oxford University Press, 2016.

LOPEZ-RUBIO, E.; PALOMO, E. J.; DOMINGUEZ, E. **Bregman divergences for growing hierarchical self-organizing networks**. [S.l.]. 2014.

LUCAS, B. D.; KANADE, T. **An Iterative Image Registration Technique with an Application to Stereo Vision**. Proceedings of the 7th International Joint Conference on Artificial Intelligence, Vancouver, v. 2, p. 674-679, 1981.

MEDEIROS, T. H. et al. **Computação Bioinspirada aplicada à Robótica**. [S.l.]: [s.n.]. 2015.

NAVARRO, I.; MATÍA, F. **An Introduction to Swarm Robotics**. ISRN Robotics, Madri, 19 Junho 2012. 10.

PHILLIPS, A. **Commercial Drones Market Projected to Reach \$3.5 billion by 2024**. DroneLife, 2017. Disponível em: <<https://dronelife.com/2017/09/01/commercial-drones-market-projected-reach-3-5-billion-2024/>>. Acesso em: 10 dez. 2017.

ROSTEN, E.; DRUMMOND, T. **Machine Learning for High-Speed Corner Detection**. Cambridge University. Cambridge, p. 14. 2006.

SIDDIQUE, N.; HOJJAT, A. **Nature Inspired Computing: An Overview and Some Future Directions**. Cognitive Computation, p. 706-714, 2015.

SINHA, U. **Fundamentals of Features and Corners: Harris Corner Detector**. AI Shack, 2012. Disponível em: <<http://aishack.in/tutorials/harris-corner-detector/>>. Acesso em: 05 jan. 2018.

TAN, Y.; ZHENG, Z.-Y. **Research Advance in Swarm Robotics**. Defence Technology, Pequim, 2 Março 2013. 18-39.

THE MATHWORKS INC. MATLAB - MathWorks. **MathWorks**, 2018. Disponível em: <<https://www.mathworks.com/products/matlab.html>>. Acesso em: 10 jan. 2018.

THEODORIDIS, S.; KOUTROUMBAS, K. **Pattern Recognition**. 4ª. ed. [S.l.]: Elsevier, 2008.

TICE, B. P. **UNMANNED AERIAL VEHICLES**. Airpower Journal, 1991.

TUTORIALS POINT. **Applications and Usage**. Tutorials Point, 2018. Disponível em: <https://www.tutorialspoint.com/dip/applications_and_usage.htm>. Acesso em: 03 fev. 2018.

WEDEL, A.; CREMERS, D. **Stereo Scene Flow for 3D Motion Analysis**. 1ª. ed. Londres: Springer, 2011.

X DEVELOPMENT LLC. Project Wing. **Project Wing**, 2017. Disponível em: <<https://x.company/wing/>>. Acesso em: 02 jan. 2018.