

UNIVERSIDADE ESTADUAL PAULISTA "JÚLIO DE MESQUITA FILHO"

FACULDADE DE CIÊNCIAS - CAMPUS BAURU

BACHARELADO EM SISTEMAS DE INFORMAÇÃO

WALTER DALLA TORRE NETO

Ferramenta para análise comportamental de insetos

Bauru - SP

2024

WALTER DALLA TORRE NETO

Ferramenta para análise comportamental de insetos

Trabalho de Conclusão de Curso apresentada à Faculdade de Ciências de Bauru da Universidade Estadual Paulista (UNESP), como parte dos requisitos para a obtenção do título de Bacharel em Sistemas de Informação.

Orientador: Prof. Dr. Marcelo de Paiva Guimarães

Bauru - SP

2024

T691f Torre Neto, Walter Dalla
Ferramenta para análise comportamental de insetos / Walter Dalla Torre Neto. -- Bauru, 2024
59 f. : il.

Trabalho de conclusão de curso (Bacharelado - Sistemas de Informação) - Universidade Estadual Paulista (UNESP), Faculdade de Ciências, Bauru
Orientador: Marcelo de Paiva Guimarães

1. Sistemas de computação. 2. Visão por computador. 3. Processamento óptico de dados. 4. Artropodes. 5. Insetos Comportamento. I. Título.

Sistema de geração automática de fichas catalográficas da Unesp.
Dados fornecidos pelo autor(a).

Dedico este trabalho a minha família e amigos;

Em especial, minha mãe Helena Maria, por me proporcionar e incentivar em tudo;

Ao meu pai Adilson Pereira que me ensinou o sentido de família;

A minha avó Carmem por me ensinar o prazer da leitura e ao meu avô Walter por me mostrar que tudo é possível com um pouco de desingripante e gambiarra;

A minha prima Amanda por ser minha referência e sempre estar ao meu lado;

Aos meus amigos que me aguentaram enquanto eu escrevia a seguinte monografia; e

Aos moradores da minha república por compartilhar a vida universitária comigo.

AGRADECIMENTOS

A todos os professores desde a pré escola, fundamental, médio, técnico e superior. Sem vocês nada seria possível e nunca chegaria até aqui.

À todas as pessoas que passaram pela minha vida, mesmo as que não me lembro mais... querendo ou não todos são parte essencial de quem sou hoje.

À Universidade Estadual Paulista “Júlio de Mesquita Filho” (UNESP), instituição de ensino superior que me formou no Bacharelado.

Ao Colégio Técnico de Limeira - UNICAMP (COTIL) , instituição de ensino médio e técnico que me formou no técnico em informática e me proporcionou muitas memórias.

À Universidade Estadual de Campinas (UNICAMP), instituição de ensino superior que sempre me inspirou e me proporcionou minha iniciação científica em química.

À EMEB Profª Alairce Ciani Deangelo e à EMEB Cecília Meireles, que proporcionaram a base curricular que me levou até aqui.

“Nenhum homem pode banhar-se duas vezes no mesmo rio...
pois na segunda vez o rio já não é o mesmo,
nem tão pouco o homem”
Heráclito de Éfeso

“Eu nunca penso no futuro. Ele não tarda a chegar.”
Albert Einstein

RESUMO

TORRE NETO, W. D. Ferramenta para análise comportamental de insetos. 2024. Monografia (Bacharelado) - Faculdade de Ciências, Universidade Estadual Paulista (UNESP), Bauru, 2024.

O presente trabalho apresenta uma ferramenta para a análise comportamental de insetos, como abelhas e moscas, em um ambiente controlado, como uma caixa experimental e duas câmeras. A ferramenta permite a avaliação de parâmetros como rotas percorridas, tempo de voo, tempo de permanência nas bordas da caixa e velocidade. Além disso, ela recria visualmente, por meio de gráficos tridimensionais (3D), o percurso realizado pelos insetos. Esses dados fornecerão subsídios para estudos aprofundados sobre tomada de decisão, diferenças de comportamento entre gêneros e faixas etárias, além de aspectos relacionados à natureza gregária, cooperativa e afetiva, entre outros. A arquitetura modular da ferramenta permite que novos recursos sejam incorporados de forma flexível, ampliando suas possibilidades de uso.

Palavras-Chave: comportamento; análise; inseto; visão computacional; ferramenta.

ABSTRACT

TORRE NETO, W. D. Tool for behavioral analysis of insects. 2024. Monograph (Bacharelado) - Faculdade de Ciências, Universidade Estadual Paulista (UNESP), Bauru, 2024.

The present work presents a tool for the behavioral analysis of insects, such as bees and flies, in a controlled environment, such as an experimental box and two cameras. The tool allows the evaluation of parameters such as routes taken, flight time, time spent on the edges of the box and speed. Furthermore, it visually recreates, through three-dimensional (3D) graphics, the path taken by insects. These data will provide support for in-depth studies on decision-making, differences in behavior between genders and age groups, as well as aspects related to the gregarious, cooperative and affective nature, among others. The tool's modular architecture allows new features to be incorporated flexibly, expanding its possibilities of use.

Keywords: Behavior; analysis; insect; computer vision; tool.

LISTA DE FIGURAS

Figura 1 - Esquema do ambiente controlado proposto por Ching-Hsin	8
Figura 2 - Esquema do ambiente controlado	11
Figura 3 - Ações do projeto	13
Figura 4 - Interface do sistema	15
Figura 5 - UML do projeto	16
Figura 6 - JSON do perfil de análise	17
Figura 7 - Tela da ferramenta de corte	19
Figura 8 - Código de correção de perspectiva	21
Figura 9 - Exemplo de máscara de background	22
Figura 10 - Exemplo de imagem tratada com ruídos	23
Figura 11 - Exemplo de imagem tratada sem ruídos	24
Figura 12 - Exemplo de imagem com o inseto detectado	25
Figura 13 - Código de remoção de background	25
Figura 14 - Projeção criada a partir das câmeras	27
Figura 15 - algoritmo de invocação de módulos de metadados	29
Figura 16 - Interface válida para módulo de metadata	29
Figura 17 - Gráfico de exemplo do Matplotlib	31
Figura 18 - Imagem de exemplo do pdf	32
Figura 19 - Primeiro frame do vídeo superior da mosca urbana	34
Figura 20 - Primeiro frame do vídeo lateral da mosca urbana	35
Figura 21 - Gráfico de rota visão diagonal	36
Figura 22 - Gráfico de rota visão lateral	37
Figura 23 - Gráfico de rota visão superior	38
Figura 24 - Relatório de metadados do cenário 1	39
Figura 25 - Primeiro frame do vídeo superior da Barata	40
Figura 26 - Primeiro frame do vídeo simulando a movimentação lateral	41
Figura 27 - Gráfico de resultados visão superior	42
Figura 28 - Resultados dos metadados	43
Figura 29 - Gráfico de resultados visão diagonal	44
Figura 30 - Gráfico de rota ângulo superior do cenário 3	45

LISTA DE SIGLAS

BM	Módulos básicos
EM	Módulos de exportação
FPS	Quadros por segundo
HTML	HyperText Markup Language
IA	Inteligência artificial
IM	Módulos de interface
MM	Módulos de metadados
OS	Sistema operacional
PDF	Portable Document Format
ROI	Região de interesse
CV	Visão computacional

SUMÁRIO

1. INTRODUÇÃO	1
1.1 Objetivos	2
1.2 Objetivos específicos	2
1.3 Organização do trabalho	2
2. FUNDAMENTAÇÃO TEÓRICA	4
2.1 Definição de variáveis a serem monitoradas	4
2.1.1 Rota	4
2.1.2 Tempo em voo e tempo em pouso	5
2.1.3 Velocidade	5
2.2 Fundamentação teórica computacional	6
2.2.1 Visão computacional	6
2.2.2 Quadros (Frame)	6
2.2.3 Região de interesse	7
2.2.4 Extração de fundo	7
3. TRABALHOS CORRELATOS	8
4. IMPLEMENTAÇÃO	10
4.1 Ambiente controlado	10
4.2 Módulos	12
4.2.1 Módulos de interface	14
4.2.1.1 Perfil de análise	17
4.2.1.2 Importar vídeo	18
4.2.1.3 Cortar bordas	18
4.2.2 Módulos básicos	20
4.2.2.1 Processar vídeo	20
4.2.2.2 Rota	26
4.2.3 Módulos de metadados	27

4.2.4 Módulos de exportação	30
4.2.4.1 Sistema de persistência de dados	30
4.2.5 Sistema de exibição dos dados	30
5. RESULTADOS	33
5.1 Cenário 1	33
5.2 Cenário 2	39
5.3 Cenário 3	43
6. CONCLUSÃO	46
REFERÊNCIAS	47

1. INTRODUÇÃO

O agronegócio é um dos principais setores da economia brasileira, chegando a compor 27,4% do PIB nacional em 2021 (CEPEA, 2022). Segundo Araújo e Oliveira (2017), o setor é responsável pelo uso em altíssima escala dos mais variados tipos de agrotóxicos para o controle de fungos, doenças e insetos. O uso é para garantir a qualidade e saúde do plantio. Há agrotóxicos específicos para cada estágio do crescimento das plantas e da maturação dos frutos, variando em quantidade, concentração e princípio ativo.

No ambiente urbano há um uso indiscriminado de inseticidas e repelentes para o controle de insetos, como moscas e baratas, que são categorizados como indesejáveis. Os inseticidas usados nas cidades não são específicos e atingem uma grande gama de animais, inclusive é comum a intoxicação de crianças. Isto levanta questões sobre o uso de inseticidas que podem contaminar solo e afluentes, afetando a fauna e flora do ambiente urbano e rural, conforme é discutido por Belchior, et. al. (2014).

Agrotóxicos podem alterar o comportamento de insetos e outros seres que não são o alvo principal do veneno, como é explorado no trabalho de Lopes (2024), onde é feita a correlação entre herbicidas e o impacto que ocorre na qualidade de vida de abelhas nativas do Brasil. Esses impactos podem variar de acordo com o inseto, tipo do agrotóxico, concentração que entrou em contato, período de exposição, e outros fatores.

O estudo acerca do comportamento de insetos expostos a inseticidas auxiliam a diminuir o efeito colateral que inseticidas possuem, uma vez que podemos compreender como atingem os insetos e como utilizar de forma mais eficiente sem desperdícios ou superdoses, ajudando na prevenção de doenças, proliferação de insetos invasores e contaminação do meio ambiente, deixando assim o uso mais responsável e melhorando a eficácia da produção e do uso do inseticida.

Atualmente há grupos de pesquisas como o de Lopes, Nocelli, Monquero (2024) e Haynes (1988) focados no estudo comportamental de insetos sob o efeito de inseticidas, analisando como o veneno altera o comportamento em diferentes espécies. Estes grupos de pesquisa utilizam softwares privados e proprietários, como o Ethoflow desenvolvido por Bernardes et al. (2021), que identificam padrões e

analisam grandes quantidades de dados. Como são proprietários, há um custo financeiro para cada nova análise e um bloqueio para o desenvolvimento de novos módulos para atender as necessidades dos pesquisadores.

A partir deste cenário surge a necessidade de uma ferramenta que seja facilmente adaptável para diversas situações dentro do escopo de análise comportamental de insetos, que permita configurações customizadas, permitindo que o pesquisador altere ou crie novos parâmetros para fácil adição e remoção de funções customizadas para que se molde às necessidades do projeto.

1.1 Objetivos

O objetivo foi desenvolver uma ferramenta de software livre que permita extrair dados e metadados de rastreamento de insetos em vídeos, incluindo informações como trajetória, tempo de voo e tempo de repouso, para análise detalhada do comportamento dos insetos quando expostos a inseticidas.

1.2 Objetivos específicos

Objetivos específicos para o sucesso do projeto incluem:

- A ferramenta deve permitir que novos módulos de funcionalidades sejam adicionados em seu código fonte;
- Levantar com precisão dados e metadados de rastreio a partir das informações de entrada;
- Exportar os dados coletados em formato legível para o usuário final; e
- Exportar os dados coletados em formato padronizado para uso em outras ferramentas.

1.3 Organização do trabalho

O texto do trabalho está organizado de modo a introduzir gradativamente os conceitos utilizados para a solução do problema apresentado na introdução.

O Capítulo 2 possui a fundamentação teórica utilizada no presente trabalho subdividida em duas frentes. A primeira sendo a fundamentação teórica do

problema, apresentando a importância das variáveis que devem ser monitoradas e extraídas, pois são utilizadas com frequência na análise comportamental de insetos. A segunda frente da fundamentação teórica trata das questões relacionadas a área da Computação, como Visão Computacional (CV) e ferramentas utilizadas no trabalho.

O Capítulo 3 traz os trabalhos correlacionados a ferramenta proposta e desenvolvida na presente monografia, citando inspirações e referências que foram seguidas.

O Capítulo 4 apresenta o método que foi utilizado para a implementação da ferramenta proposta nos capítulos anteriores.

O Capítulo 5 expõe a ferramenta e os resultados obtidos a partir de alguns vídeos de testes.

O Capítulo 6 mostra os resultados obtidos do desenvolvimento da ferramenta e discute novas melhorias futuras.

2. FUNDAMENTAÇÃO TEÓRICA

2.1 Definição de variáveis a serem monitoradas

Há diversos dados úteis para pesquisadores de análise comportamental de insetos, sendo os mais comuns aqueles relacionados aos movimentos e comportamentos. Isso ocorre porque os inseticidas são conhecidos por estimular ou reduzir comportamentos locomotores e de autolimpeza (*grooming*), conforme descrito por Haynes (1988).

A ferramenta desenvolvida neste trabalho é focada em recolher informações da posição e locomoção de insetos dos vídeos importados para análise. O foco é processar essas informações nos Módulos Básicos (*Basic Modules - BM*) e Módulos de Metadados (*Metadata Module - MM*) com finalidades diversas a fim de gerar metadados que podem ser úteis para indicar padrões no comportamento do inseto analisado.

Os dados e metadados levantados são não somente para uso de análise humana, mas também para exportar para ferramentas de terceiros, a fim de enriquecer os dados apresentados junto aos vídeos.

A seguir, serão apresentados os dados relevantes para a análise comportamental de insetos, com foco na localização em três dimensões e nos seus subprodutos, conhecidos como metadados.

2.1.1 Rota

Rota é o caminho percorrido pelo inseto dentro da caixa. Esta variável é importante porque pode revelar informações como a predominância de movimento circular, caótico, letárgico ou estacionário. Além disso, permite notar se o inseto, por possuir algum mecanismo de detecção de concentração, prefere ficar afastado do bocal de onde o veneno é inserido.

É necessário extrair e exportar esses dados para que o pesquisador possa analisar de diversos ângulos e reproduzir a rota diversas vezes. Os dados devem estar disponíveis para serem exportados para outros sistemas como um conjunto de pontos cartesianos em três dimensões, facilitando o uso em novos

módulos, softwares terceiros, e para leitura humana. O uso da rota em softwares de terceiros auxilia sistemas a detectar padrões nas imagens e estabelecer ligações entre a posição indicada pela ferramenta deste trabalho com a imagem do vídeo.

2.1.2 Tempo em voo e tempo em pouso

O tempo de voo e em pouso é relevante para determinar se o inseto analisado possui preferência de comportamento em determinadas situações. Isso pode auxiliar o pesquisador a identificar se esse comportamento muda ao alterar as condições do ambiente. Por exemplo, se há mudança no comportamento de um inseto que prefere ficar em repouso quando não está sob o efeito de um inseticida e se esse comportamento muda ao entrar em contato com o veneno.

O sistema identifica com base na posição do inseto qual a borda que está pousada e consegue marcar no objeto de dados final quanto tempo o inseto esteve fora da borda, ou seja, voando.

2.1.3 Velocidade

Essa variável consiste no cálculo da velocidade média entre dois pontos consecutivos, permitindo identificar a velocidade e a aceleração alcançadas pelo inseto em um intervalo de tempo específico. Com esses dados, o pesquisador pode utilizar representações gráficas para exibir a variação de velocidade e aceleração ao longo do tempo.

A velocidade é um dado relevante para o pesquisador, pois permite identificar se o inseto apresenta sinais de letargia ou reatividade, além de possibilitar a estimativa de sua velocidade tanto em voo quanto durante o movimento no pouso.

A velocidade é um dado que para essa pesquisa caracterizamos de metadado, pois ela é o subproduto do módulo de rotas com o tempo, isso permite ao pesquisador identificar a velocidade do inseto analisado e se o comportamento com relação a velocidade muda, podendo auxiliar a identificar super estímulos e doses letárgicas.

O cálculo da velocidade baseia-se na determinação da velocidade média entre dois pontos consecutivos, a partir da qual é possível identificar a velocidade e

a aceleração alcançadas pelo inseto em um determinado intervalo de tempo. Isso permite ao pesquisador utilizar o modo gráfico para exibir a variação da velocidade e aceleração.

2.2 Fundamentação teórica computacional

2.2.1 Visão Computacional

Faz parte da teoria da computação a análise de imagens para detecção de objetos e reconstrução de dados. Pesquisadores na área da CV desenvolveram técnicas matemáticas para reconhecimento e reconstrução de formas e objetos. Há várias técnicas matemáticas que podem ser utilizadas para diversos fins, algumas possuem aplicações mais específicas e outras mais generalizadas, mas todas possuem um caráter estatístico para reconstruir dados com base em informações incompletas e com variáveis desconhecidas, como é descrito por Szeliski (2010).

A área utiliza de teoremas matemáticos e físicos para compor algoritmos de análise de imagens para problemas genéricos que preveem solucionar problemas específicos, conforme é discutido no livro de Szeliski (2010).

O presente trabalho apresenta uma solução utilizando de CV para identificar e definir a posição do inseto em três dimensões a partir de dois vídeos. A seguir, são descritos conceitos que foram utilizados para implementar o software.

2.2.2 Quadros (*Frame*)

Um vídeo é composto por diversas imagens estáticas que são alternadas de acordo com o tempo, dando a ilusão de fluidez e movimento. Na área da CV cada imagem é chamada de frame. O tempo de alternância entre elas é denominado *frames* por segundo. Neste trabalho é considerado constante a quantidade de *frames* por segundo.

2.2.3 Região de interesse

Região de Interesse (*Region of Interest* - ROI) é uma área específica de uma imagem destacada para análise ou processamento mais detalhado. O objetivo desta técnica é diminuir o custo computacional, recontando a imagem original a fim de construir uma nova figura menor com somente a região relevante para a análise matemática poupando recursos do computador.

Em um cenário ideal, a câmera seria posicionada para capturar o ambiente controlado de forma precisa. No entanto, não é possível garantir a ausência de distorções, posicionamentos incorretos ou incompatibilidades entre o formato do ambiente controlado e o campo de visão da câmera. É responsabilidade do ROI remover tais distorções a fim de gerar uma nova imagem pronta para a análise.

2.2.4 Extração de fundo

O plano de fundo de uma imagem muitas vezes não é relevante para os algoritmos de análise utilizados para resolver determinados problemas. De modo geral, a extração de fundo seleciona um objeto de interesse e o remove da imagem como um todo. Os vídeos capturados não possuem informações de profundidade, portanto o plano de fundo é qualquer informação externa ao objeto de interesse.

O campo de pesquisa da CV gerou diversos algoritmos ao longo dos anos para remover o fundo de imagens e vídeos como o algoritmo de segmentação bicamada ou o algoritmo z-keying e multi-view stereo, que foi discutido por Szeliski (2010).

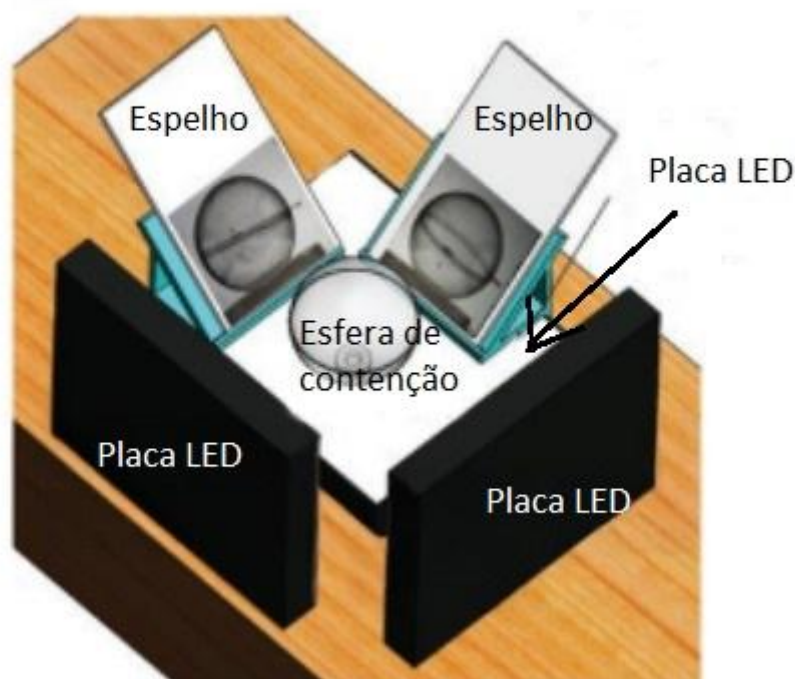
3. TRABALHOS CORRELATOS

A área de pesquisa da análise comportamental de insetos utiliza ferramentas computacionais há diversos anos para monitorar e registrar o posicionamento do inseto. Há diversas ferramentas proprietárias e públicas disponíveis aos pesquisadores.

Um dos trabalhos correlatos mais relevantes é o software descrito no trabalho de Chen, Chiang e Tsai (2021), que se baseia em analisar a movimentação em três dimensões utilizando a menor quantidade de câmeras possível. O grupo de pesquisa utilizou uma única câmara apontada para uma bio esfera oca e transparente de acrílico, o que normalmente resultaria em uma projeção bidimensional, porém eles utilizaram de dois espelhos posicionados em ângulos específicos para criar duas novas câmeras virtuais, dessa forma obtendo três ângulos de visão em uma só imagem.

No artigo de Chen, Chiang e Tsai (2021) é descrito o ambiente controlado, sendo uma esfera de acrílico com duas placas de LED para manter a iluminação uniforme e dois espelhos angulados para refletir a esfera, conforme a Figura 1. Esta pesquisa foi a maior influência para a monografia atual, pois nele o grupo descreve os métodos para a remoção da imagem de fundo e ROI utilizados no projeto.

Figura 1 - Esquema do ambiente controlado proposto por Ching-Hsin



Fonte: Adaptado de Chen, Ching-Hsin et. al. (2021)

Outra pesquisa que tem grande correlação com o presente trabalho é do grupo de pesquisa de Fry et al. (2000), que busca sanar questões acerca do balanço entre a qualidade de vídeo e o campo de visão. A solução apresentada por ele foi a utilização de uma câmera de vigilância com eixo de angulação para seguir o inseto, rotacionando a câmera a fim de reduzir a imagem analisada para o ROI. O artigo se propõe a discutir e executar o monitoramento de insetos em duas e três dimensões, usando duas câmeras de vigilância para monitorar o posicionamento em três dimensões do inseto analisado.

Vários trabalhos foram utilizados como referência para a elaboração da atual monografia, como Farder-Gomes et al. (2021) e Gomes et al. (2024). Eles utilizam o software proprietário Ethoflow, desenvolvido no trabalho de Bernardes et al. (2021).

O software Ethoflow utiliza Inteligência Artificial juntamente com CV para descrever a rota e comportamentos de múltiplos insetos. Ele é capaz de analisar diversos animais como peixes, cupins e ratos em diversas condições de luz, cor de fundos e qualidade de vídeo como é descrito por Bernardes et al. (2021).

4. IMPLEMENTAÇÃO

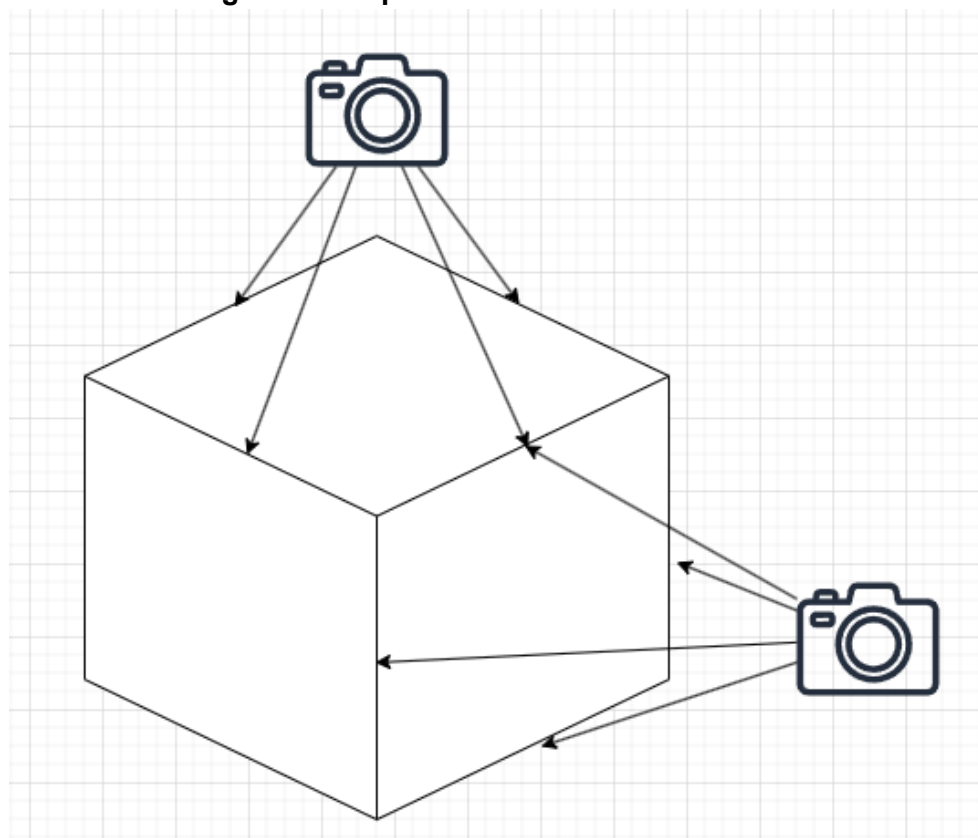
O projeto foi desenvolvido seguindo os procedimentos de levantamento de requisitos funcionais e não funcionais para a definição do escopo, levantamento bibliográfico e projetos correlacionados.

Como cerne dos requisitos funcionais está a questão de auxiliar o pesquisador a levantar e analisar os dados e metadados obtidos a partir dos vídeos e parametrizações. Os dados relevantes a serem monitorados foram discutidos no Seção 2.1. Esses dados foram escolhidos por serem os mais relevantes na análise comportamental, que podem ser extraídos dentro das limitações expressadas pelos requisitos não funcionais, como o de poder ser executado em um desktop ou servidor.

A ideia para tornar isso possível é que com análises puramente matemáticas de CV é possível chegar a uma precisão aceitável da posição do inseto.

4.1 Ambiente controlado

O sistema foi projetado para operar em um ambiente controlado, com duas câmeras posicionadas em um ângulo de 90 graus entre si, apontando para uma caixa retangular transparente com fundo branco e iluminação uniforme, sem gradientes, conforme ilustrado na Figura 2.

Figura 2 - Esquema do ambiente controlado

Fonte: Elaborado pelo Autor (2024)

A caixa (Figura 2) servirá de confinamento para conter o inseto dentro do ambiente, podendo ser de qualquer tamanho, desde que seja grande o suficiente para que o inseto fique livre para exibir seu comportamento de pouso e voo a critério do pesquisador. A caixa deve ser transparente nos lados em que a câmera estiver posicionada, sendo feita de vidro ou acrílico, a fim de permitir que a câmera consiga ver toda a extensão das bordas.

O fundo dos vídeos deve ser branco, portanto os lados da caixa que não possuem a câmera podem ser opacos na cor branca ou então posicionar a caixa em cima de um plano branco e fundo branco, ou até cobrir as laterais com tecidos. O fundo não pode conter distorções que possam criar sombras.

A iluminação ideal é sem gradientes e igual para toda a extensão da caixa, porém a forma como a extração de fundo do software foi implementada aceita iluminações com gradientes. Com tudo a iluminação não deve variar durante o experimento, pois variações na intensidade de luz podem interferir com a precisão da posição do inseto.

As câmeras devem ser posicionadas em lados adjacentes, formando um ângulo de 90 graus em relação a caixa. Idealmente posicionadas no topo e em uma lateral, elas devem ser acionadas ao mesmo tempo por meios externos. É imprescindível que elas gravem na mesma velocidade de quadros por segundo (*Frames Per Second* - FPS), a resolução das câmeras podem variar, porém quanto maior a resolução melhor o resultado do algoritmo de posicionamento do inseto. O enquadramento ideal da câmera deve capturar as bordas da caixa.

4.2 Módulos

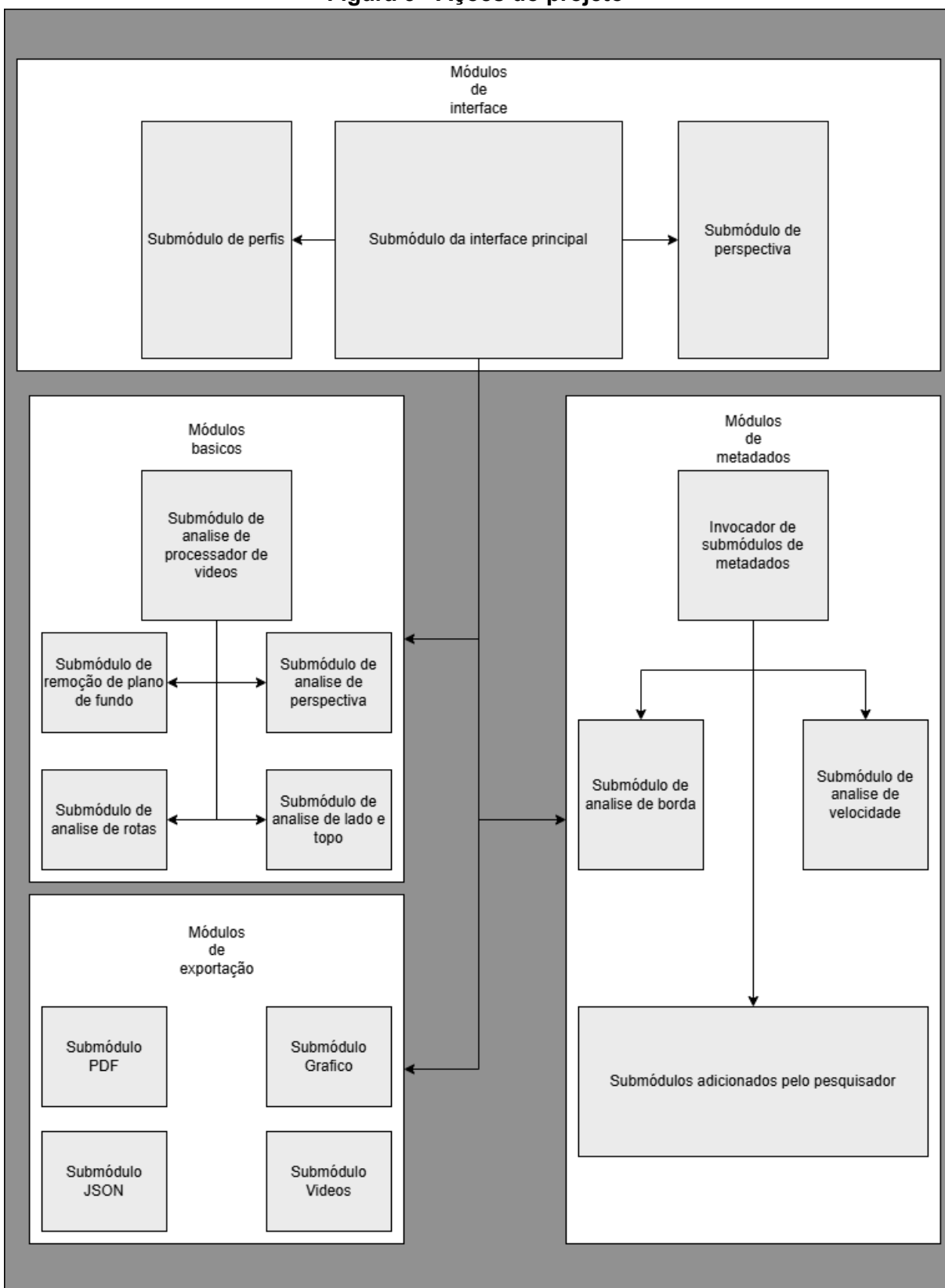
A ferramenta de rastreamento de insetos foi construída usando uma arquitetura modular, sendo que cada parte é um módulo que existe independente dos outros. A única junção entre os módulos é a camada de interface ou visualização do sistema. O projeto foi desenvolvido em quatro módulos principais, com submódulos de acordo com a finalidade a ser executada. De modo geral, os módulos estão estruturados conforme a Figura 3.

Os módulos estão distribuídos em:

- Módulos de interface (*Interface Modules* - IM) responsáveis pelas interfaces sistema-usuário;
- Módulos de extração de dados básicos (*Basic Modules* - BM), que executam a extração de dados básicos dos vídeos importados;
- Módulos de metadados (*Metadata Modules* - MM) que extraem novos dados com base nas informações básicas do vídeo; e
- Módulos de exportação (*Export Modules* - EM) que geram PDFs (*Portable Document Format* - PDF) e exibe o gráfico de rota.

Os quatro módulos trabalham em conjunto para permitir que o software desempenhe todos os requisitos funcionais e não funcionais já descritos. As ações que os usuários podem executar são processar vídeos, processar dados e criar novo perfil de análise, que nos próximos capítulos serão explicados com mais detalhes.

Figura 3 - Ações do projeto



Fonte: Elaborado pelo Autor (2024)

4.2.1 Módulos de interface

A interface do sistema (Figura 4) foi feita no Tkinter, uma biblioteca Python feita para exibir interfaces. A sua escolha ocorreu porque, conforme a sua documentação oficial da Python Software Foundation (2024), ela integra diretamente com o Sistema Operacional (*Operating System* - OS) do usuário, permitindo que usuários de Windows, Linux e MacOs possam utilizar o software deste projeto sem problemas de compatibilidade.

A interface da Figura 4 segue a ordem lógica das ações dentro do software. Contudo, a interface foi concebida com a intenção de facilitar a depuração e criação de novos módulos de metadados ao dar ao desenvolvedor a possibilidade de não executar novamente os módulos básicos, a fim de economizar tempo durante o desenvolvimento e testes dos módulos de metadados.

A interface principal do sistema permite que o usuário interaja e execute todos os fluxos do projeto, que estão expressados no Linguagem de Modelagem Unificada (UML) da Figura 5.

Figura 4 - Interface do sistema

The image shows a software interface titled "Ferramenta para a análise comportamental de insetos". The interface is organized into several sections:

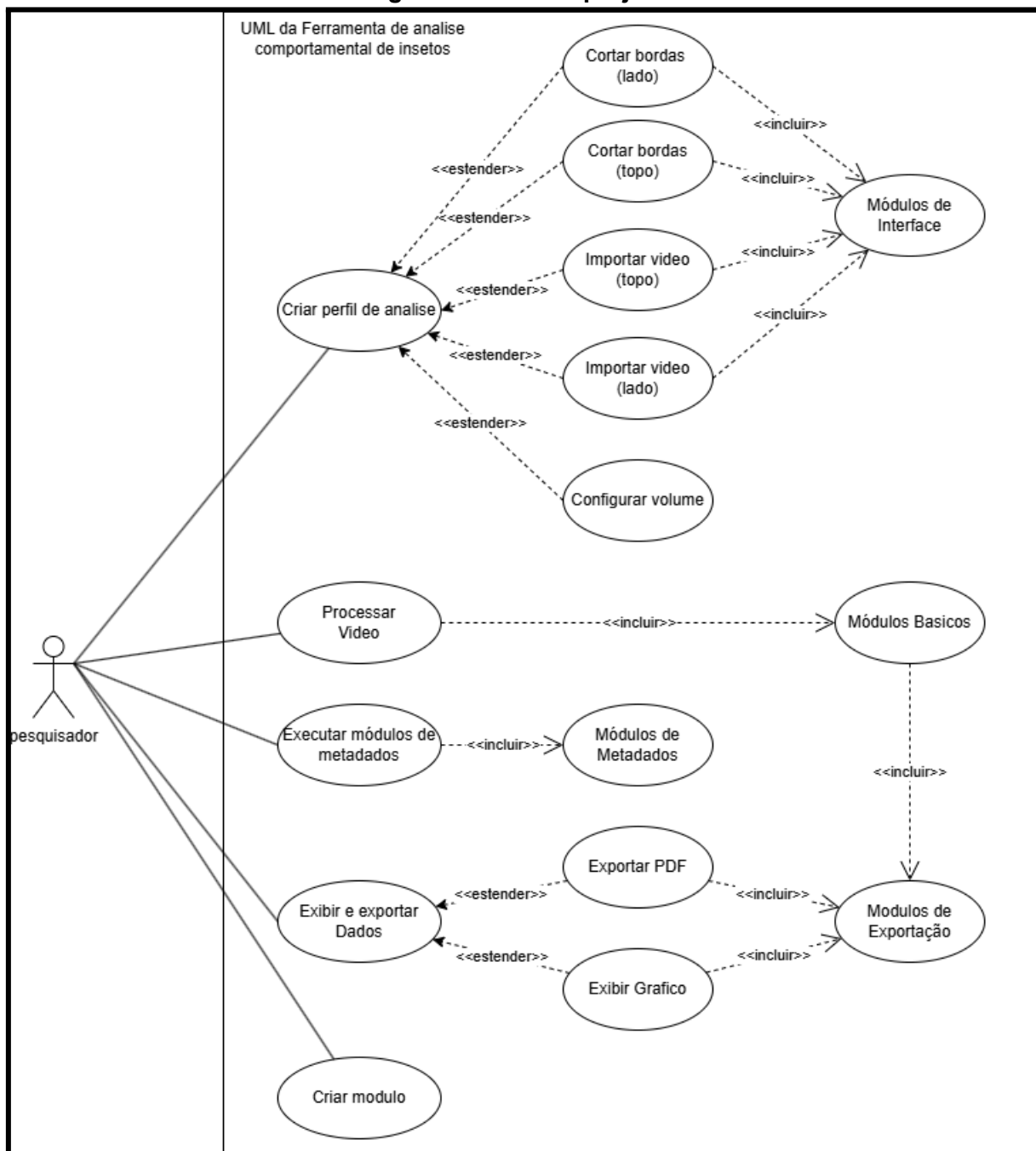
- Seleção de perfil de análise:** A dropdown menu with the option "Novo perfil de análise".
- Seleção de local do arquivo de vídeo topo:** A text input field.
- Configuração de bordas (topo):** A button.
- Seleção de local do arquivo de vídeo lado:** A text input field.
- Configuração de bordas (lado):** A button.
- Medidas da caixa:** Three input fields labeled "Altura (cm)", "Largura (cm)", and "Profundidade (cm)".
- Botões de ação:** A series of buttons including "Salvar configurações", "Processar vídeo (Módulos Básicos)", "Executar módulos de metadados", "Exibir gráfico de rota", and "Exportar para PDF".

Fonte: Elaborado pelo Autor (2024)

Todos os fluxos partem do pesquisador (conforme a Figura 5) que é o usuário principal do trabalho. Ele possui poder total na ferramenta e nenhum módulo é executado sem ser iniciado pela ação do usuário. A interface permite selecionar e cadastrar os perfis de análise, permitindo ao usuário enviar os locais dos arquivos de vídeo e configurá-los, adicionar medidas da caixa como altura, largura e profundidade em centímetros. Estes dados são usados pelos módulos para calcular

os metadados. Nesta tela é possível reprocessar os dados, exibir um gráfico de rota do perfil selecionado e exportar os dados e metadados para um PDF para leitura humana.

Figura 5 - UML do projeto



Fonte: Elaborado pelo Autor (2024)

Os fluxos apresentados no UML da Figura 5 são descritos com detalhes nas seções a seguir.

4.2.1.1 Perfil de análise

A ferramenta possui uma função onde é possível salvar perfis de análise, que possibilitam ao pesquisador configurar e executar uma análise e guardá-la para reproduzir em outro momento.

O perfil de análise guarda todos os dados de configuração necessários para iniciar o processamento da análise ou exibir os dados e metadados em qualquer momento. Permitindo assim, que o pesquisador possa executar com facilidade análises sem a necessidade de configurar o sistema a todo momento. Isso o permite testar o funcionamento de novos módulos ou compartilhar perfis com seus pares.

No momento que se abre o sistema o perfil “Novo perfil de análise” é selecionado, como é demonstrado na Figura 5. Após isso é possível trocar de perfil ou realizar uma análise sem criá-lo.

É possível criar um perfil clicando no botão “Salvar Configurações” na tela principal do sistema, como indicado na Figura 5. Cada perfil salvo fica guardado na memória em um arquivo JSON, armazenado em `./cache/configs.json`. Neste arquivo há um objeto que salva cada perfil, como mostrado na Figura 6. Os dados `frame_perspective_points_side` e `frame_perspective_points_top` são usados para correção de perspectiva contendo quatro pontos em duas dimensões. Os dados `side_video_path` e `top_video_path` são os caminhos internos para os vídeos de cada perfil. E `width_box_cm`, `depth_box_cm` e `height_box_cm` são o tamanho do comprimento, largura e altura, respectivamente, em centímetros da caixa.

Figura 6 - JSON do perfil de análise

```
{  
  "frame_perspective_points_side": List<(int, int)>,  
  "frame_perspective_points_top": List<(int, int)>,  
  "side_video_path": string,  
  "top_video_path": string,  
  "width_box_cm": int,  
  "depth_box_cm": int,  
  "height_box_cm": int  
}
```

Fonte: Elaborado pelo Autor (2024)

Os dados salvos são usados caso ocorra a troca do perfil de análise ou durante a abertura do sistema. Esses dados são enviados para o módulo básico e módulo de metadados no início do processamento do perfil de análise.

4.2.1.2 Importar vídeo

Na primeira tela do sistema (Figura 5) há o botão “Selecione o local do arquivo de vídeo topo”, que ao ser clicado é chamado uma função do OS que possibilita selecionar o arquivo .mp4 gravado com a câmera no topo da caixa. Há também o botão “Selecione o local do arquivo de vídeo lado”, que possibilita selecionar o arquivo .mp4 gravado de um dos lados da caixa. Os caminhos dos vídeos são salvos no perfil de análise selecionado.

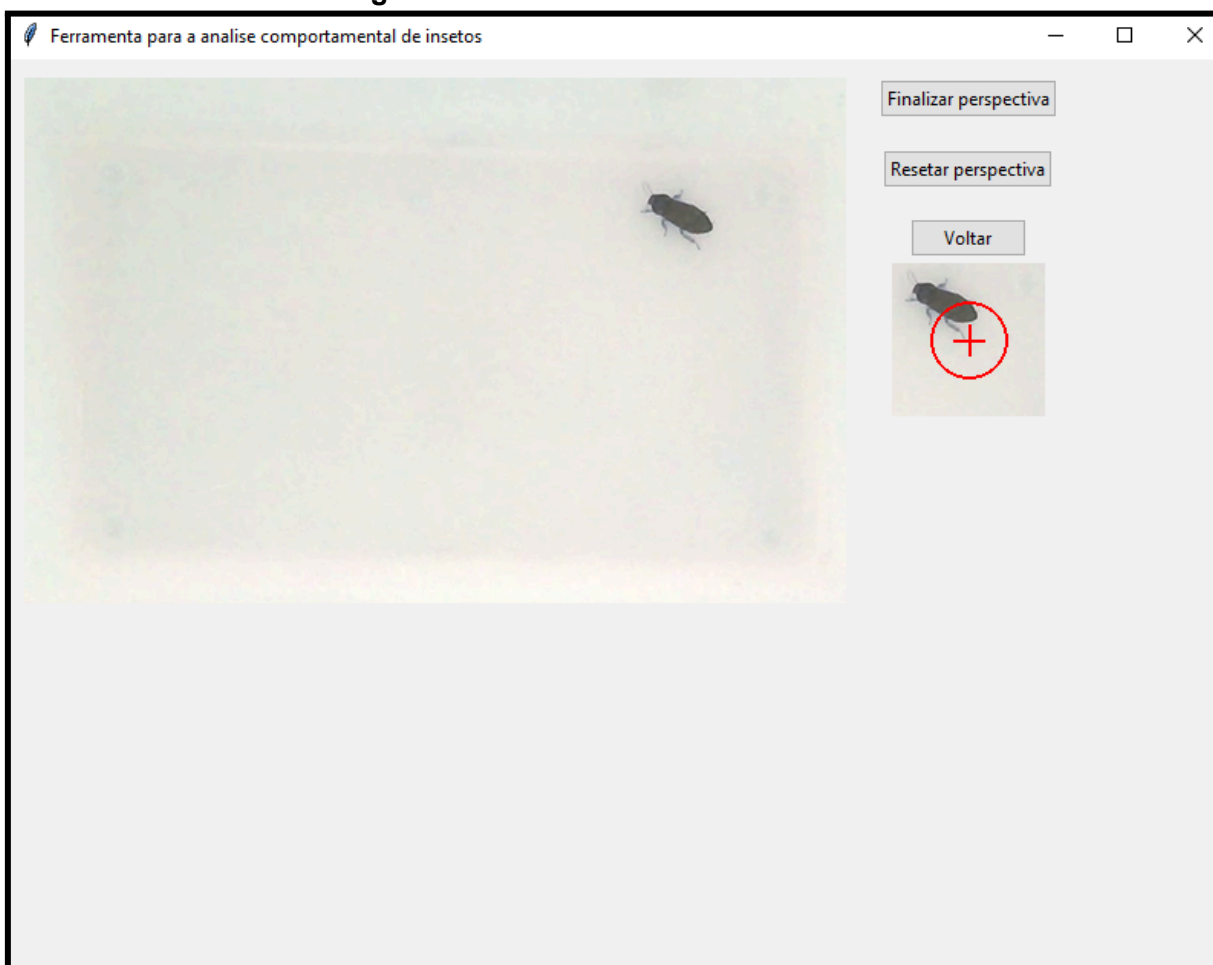
Os vídeos não são copiados ou duplicados, só ocorrem interações com o arquivo de vídeo durante o processamento da análise.

4.2.1.3 Cortar bordas

Este passo é essencial para a configuração da ferramenta, pois permite que o pesquisador corrija as bordas limites da análise e diminua o ROI sem a necessidade de sistemas externos de edição de vídeo. Após selecionar os vídeos, é possível clicar no botão “configurar bordas” com o subtítulo de “topo” e “lado”. Ao clicar no botão, uma nova tela será exibida (Figura 7) contendo o primeiro frame do vídeo indicado na configuração da interface da Figura 5.

É possível interagir com a imagem clicando nela para definir quatro pontos de correção de perspectiva na ordem: ponto superior direito, ponto superior esquerdo, ponto inferior direito e ponto inferior esquerdo.

Ao movimentar o cursor em cima da imagem principal, outra imagem será exibida ao canto, mostrando um recorte com bordas de 50 pixels exatamente onde o cursor está posicionado com uma cruz centralizada para aumentar a precisão do clique.

Figura 7 - Tela da ferramenta de corte

Fonte: Elaborado pelo Autor (2024)

Após finalizar o clique dos quatro pontos, a imagem será cortada e sua perspectiva será corrigida de acordo com os pontos selecionados. É possível reiniciar a perspectiva clicando no botão “resetar perspectiva”, onde a tela será recarregada com os valores zerados para uma nova configuração. Após selecionar os quatro pontos de perspectiva é possível sair da tela salvando as configurações clicando no botão “finalizar perspectiva”.

Se o usuário optar por não modificar nenhuma perspectiva e clicar no botão “voltar”, os pontos de perspectiva selecionados serão automaticamente os mais extremos da imagem, resultando na ausência de correção de perspectiva. Em ambos os casos, os pontos de perspectiva são salvos no perfil de análise.

Definir o ROI é importante para as análises posteriores no BM pois, remover partes da imagem que não são relevantes para a análise comportamental

do inseto auxilia no processamento do vídeo, uma vez que não é necessário gastar tempo de processamento com elas.

4.2.2 Módulos básicos

Os BM são os que fazem a manipulação e análise do vídeo, realizando a extração de fundo, ROI, correção de perspectiva e localização do objeto de interesse, e levantamento dos dados de FPS e a quantidade de *frames*. No atual estágio do projeto, esses módulos não são de acesso público e não é esperado que o pesquisador os manipule-os.

Os BM foram criados em cima da biblioteca `opencv-python` (Open Source Computer Vision Library). Ela implementa uma gama de algoritmos de CV de uso gratuito que rodam somente na CPU.

Os BM e MM utilizam a biblioteca `NumPy`, que é responsável por implementar algoritmos complexos de computação numérica como álgebra linear e operações estatísticas.

4.2.2.1 Processar vídeo

Após configurar o perfil de análise, inserindo os vídeos, cortando as bordas e preenchendo os dados da caixa, a ferramenta está pronta para iniciar a análise de comportamento do inseto, que é dividida em duas etapas. A primeira etapa é o processamento do vídeo onde o sistema interage com o vídeo salvo na memória do computador, envia para o submódulo de correção de perspectiva. A correção de perspectiva executa *frame a frame* do vídeo, corrigindo a perspectiva utilizando os métodos `getPerspectiveTransform` e `warpPerspective` da biblioteca `cv2`.

Conforme o código da Figura 8, estes métodos geram a matriz de correção da perspectiva e como subproduto removem todas as informações que estão fora dos pontos de perspectiva gerando o ROI que será analisado. Após isso, ocorre a remoção do plano de fundo (*background*). Há diversas formas de fazer a remoção do plano de fundo de um vídeo, porém nem todas funcionam no contexto de insetos, que são objetos muito pequenos e que se movem muito rapidamente.

Figura 8 - Código de correção de perspectiva

```
for frame in frames:  
    matrix = cv2.getPerspectiveTransform(...)  
    output_frame = cv2.warpPerspective(frame, matrix, ROI_size)
```

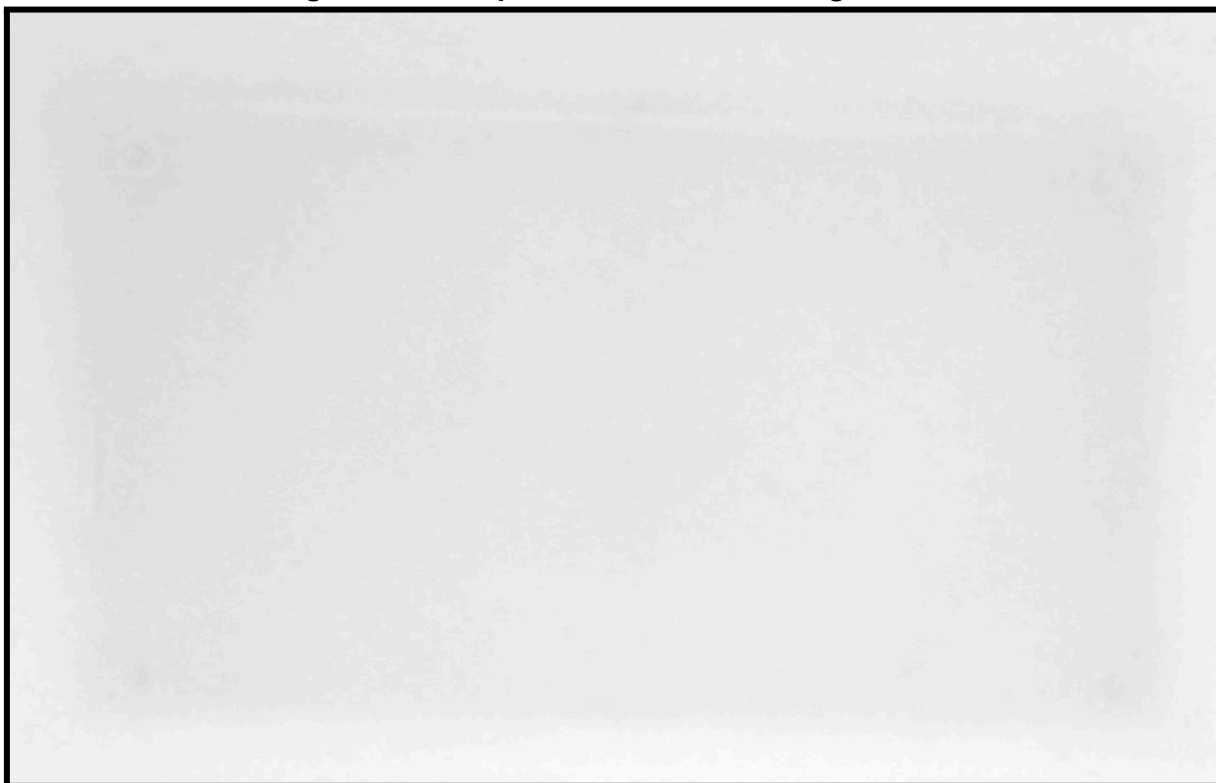
Fonte: Elaborado pelo Autor (2024)

Cabe notar que insetos costumam ter coloração escura quando sua imagem é tratada para cinza. Isto não é uma regra, pois há insetos que possuem diversas cores, porém os insetos mais comuns de serem analisados como abelhas, moscas e baratas seguem esse padrão.

Uma das ideias exploradas para localizar o inseto é selecionar o pixel mais escuro dentro do ROI, porém essa abordagem é imprecisa uma vez que a iluminação pode gerar falsos positivos nas bordas e principalmente na sombra do inseto.

A ideia de Chen, Chiang e Tsai (2021) para remover o plano de fundo é usar um algoritmo dinâmico no qual ocorre o recalibramento a cada 500 imagens. Primeiro todos os *frames* do vídeo são tratados para a escala de cinza (grayscale), que vai de 0 a 255, e então é selecionado o primeiro *frame* de cada bloco gerando uma lista de *frames* pseudo randômicos. Para cada posição de pixel é selecionado o pixel com o maior valor dentre todas as imagens, isto é, o pixel mais claro, para constituir uma máscara de fundo (Figura 9). A imagem é formada por uma matriz numérica seguindo o valor da escala de cinza.

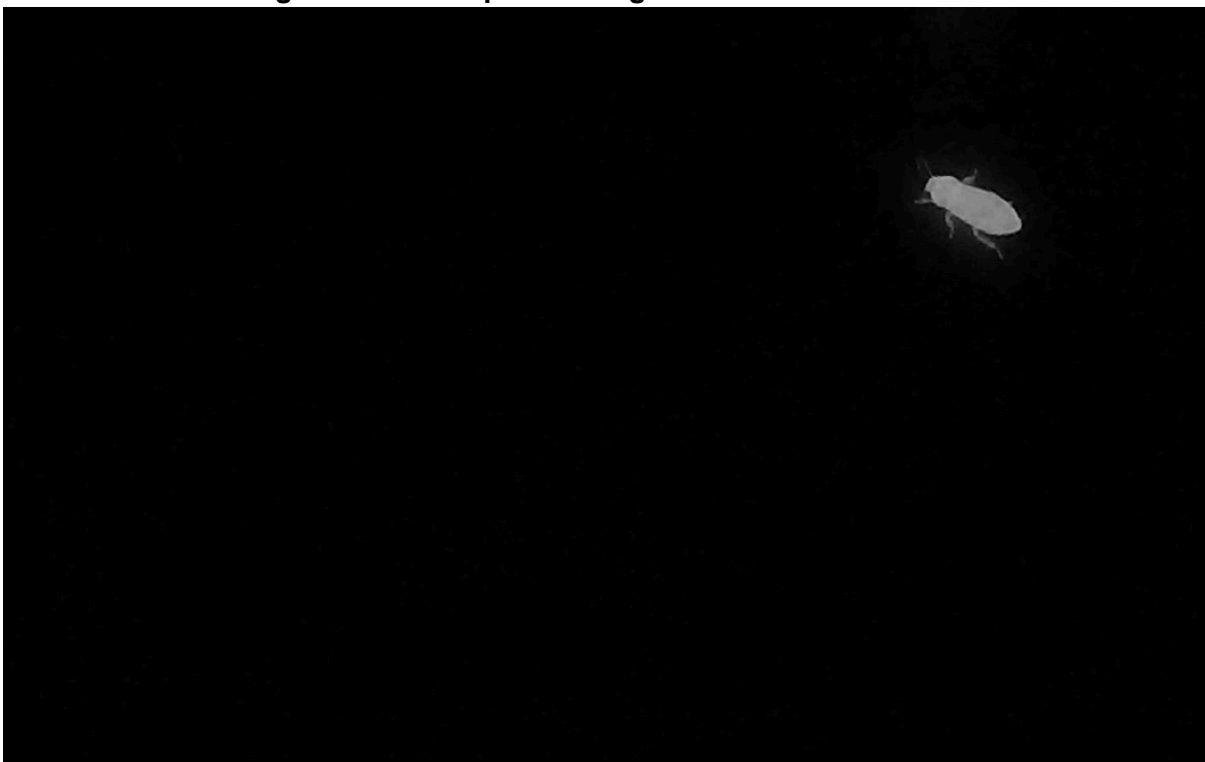
Figura 9 - Exemplo de máscara de background



Fonte: Elaborado pelo Autor (2024)

Os *frames* são formados por uma matriz numérica seguindo o valor da escala de cinza, portanto é possível subtrair as matrizes para cada *frame* do resto do vídeo. As imagens resultantes são formadas por pixels com o valor próximo do zero, portanto escuros no local onde o plano de fundo coincide com a máscara. Já no local em que o inseto estiver presente a subtração resultará em um valor próximo do branco, conforme o exemplo da Figura 10.

Figura 10 - Exemplo de imagem tratada com ruídos



Fonte: Elaborado pelo Autor (2024)

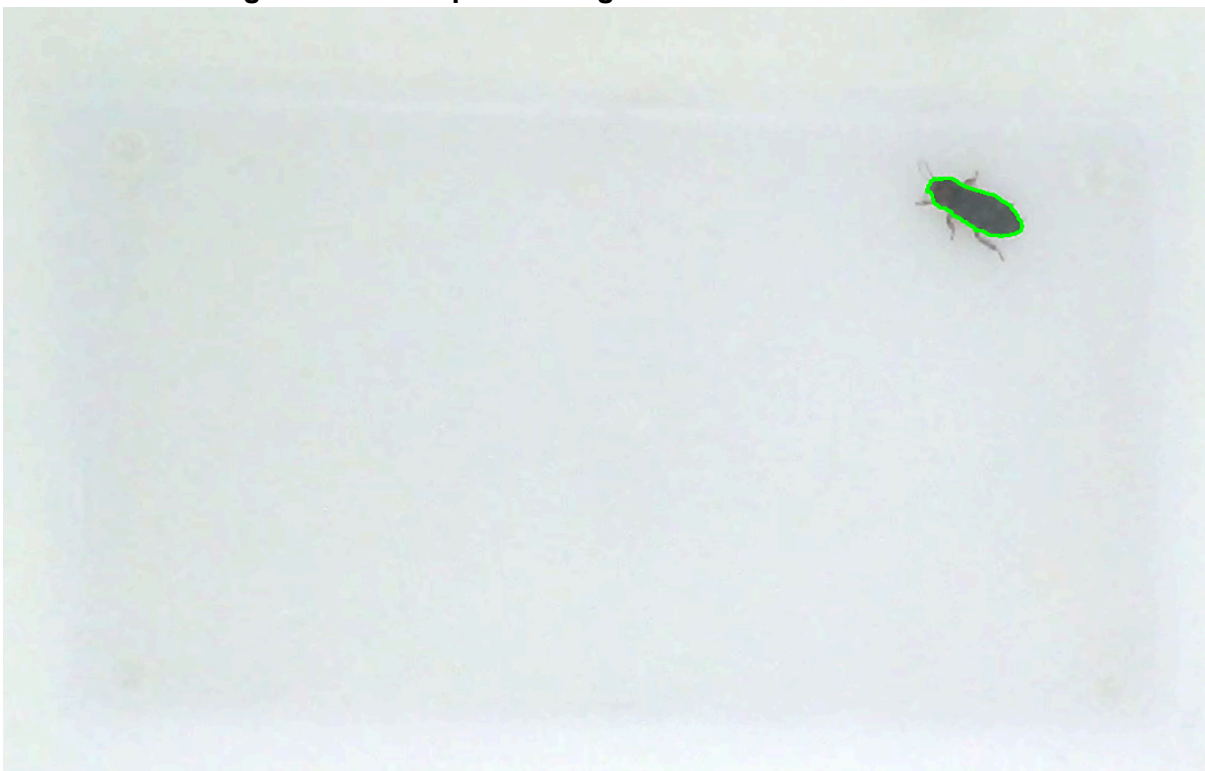
Para um ser humano, esse tratamento é o suficiente para evidenciar o local do inseto, porém para a máquina é necessário limpar os ruídos que ocorrem perto do inseto. Então é removido todos os pixels com valor menor a 80, cujo valor foi escolhido após testes, o que resulta em uma imagem como a da Figura 11.

Figura 11 - Exemplo de imagem tratada sem ruídos



Fonte: Elaborado pelo Autor (2024)

Após esses tratamentos temos uma imagem em escala cinza constituída de manchas escuras pequenas e, idealmente, uma única mancha grande contendo o inseto. Por isso, é usado um algoritmo do OpenCv para encontrar o contorno de todas as manchas e escolher a mancha com o maior área de contorno que será o inseto. A Figura 12 representa a imagem de entrada com o contorno encontrado pelo algoritmo de CV.

Figura 12 - Exemplo de imagem com o inseto detectado

Fonte: Elaborado pelo Autor (2024)

Coletando a posição central do maior contorno, é possível definir a posição do inseto a cada *frame*. Como limitação técnica, não é possível garantir que não ocorra mudanças na ordem dos pontos na importação e exportação em uma lista (*array*) no padrão JSON. Portanto, cada ponto está localizado em um objeto cuja chave é o número do *frame*.

O algoritmo descrito anteriormente é sumarizado na Figura 13.

Figura 13 - Código de remoção de background

```
selected_fist_frame_batch = ... # Gerar o lote de 500 frames
white_mask = np.max(selected_fist_frame_batch)
for i, frame in enumerate(frame_count):
    dif_frame = cv2.absdiff(white_mask, frame)
    _, diff = cv2.threshold(dif_frame, minThreshold, ...)

    ... Tratamentos de imagem

    contours, _ = cv2.findContours(frame, ...)
    max_contour = ... # Seleciona o contorno maximo
```

Fonte: Elaborado pelo Autor (2024)

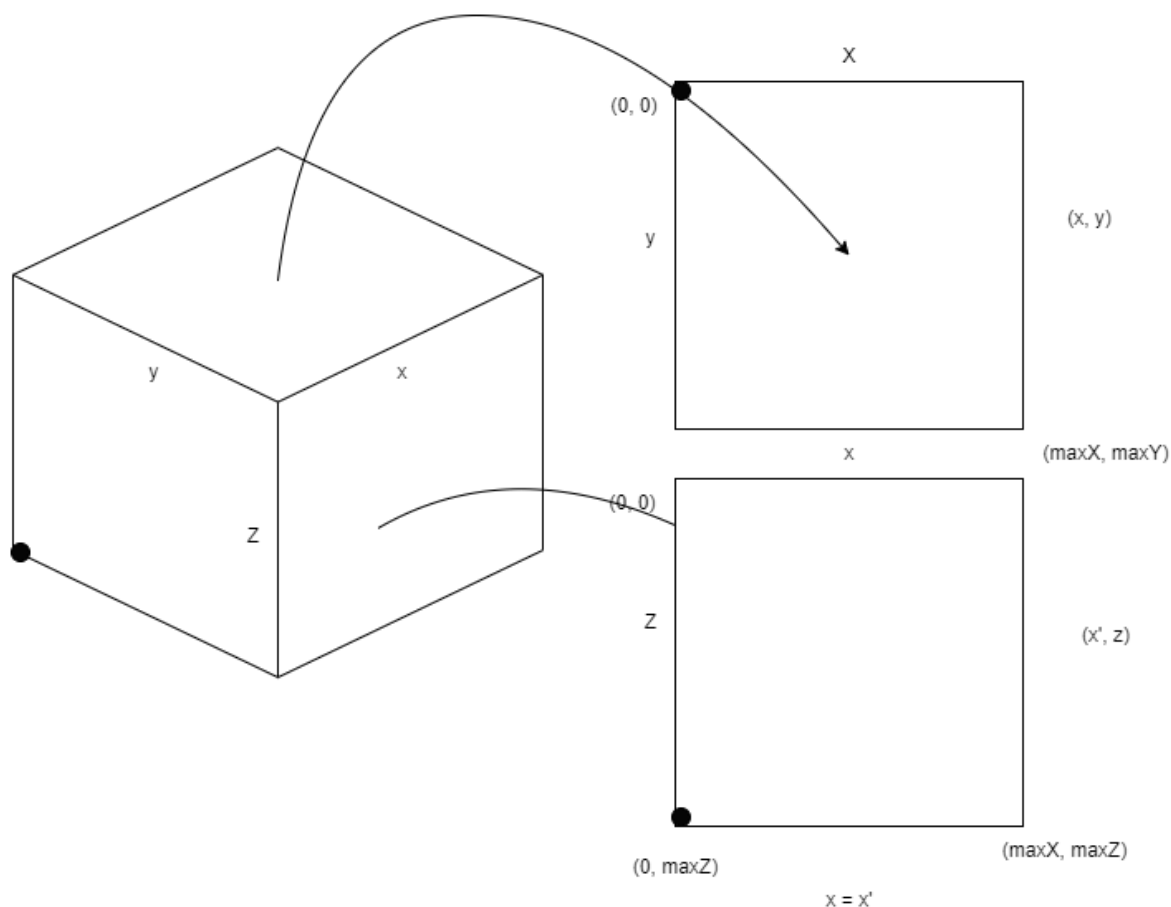
O BM também extrai do vídeo o FPS e a quantidade de *frames*. Todos os dados são salvos em um arquivo JSON com o nome do perfil no caminho da pasta “./cache/output”.

4.2.2.2 Rota

A rota do inseto é descrita a partir de pontos em um plano cartesiano com três dimensões. Ela é definida a partir do centro de uma área com maior concentração de pixels pretos nos vídeos de topo e de lado.

A câmera montada no topo da caixa define a posição do inseto na largura (x) e profundidade (z). A câmera montada no lado da caixa define a altura (y), conforme a Figura 14. Após isso, os dados são unidos, gerando um ponto (x, y, z) e sendo salvos na estrutura de dados final, que é exibida no formato JSON. Nesse JSON, a chave principal é denominada “Rota”, e os valores consistem em uma lista de objetos. Cada objeto tem como chave o número do *frame* do vídeo, contendo sua posição em um sub objeto com chaves correspondentes aos eixos (x, y, z) e valores definidos pelas coordenadas do ponto.

Figura 14 - Projeção criada a partir das câmeras



Fonte: Elaborado pelo Autor (2024)

O sistema de rotas é portanto uma lista com objetos indexados em ordem contendo os valores da posição em três dimensões (x , y , z) do inseto, que quando enviados ao módulo de exibição na taxa correta, exibe a rota do inseto em tempo real limitado somente à taxa de atualização do sensor.

O objeto da rota é a base das análises dos módulos seguintes de velocidade, preferência de local e tempo de voo. Após a definição da rota, é iniciado o fluxo de processamento de metadados, onde são executados os módulos de metadados do sistema que estão apresentados a seguir.

4.2.3 Módulos de metadados

Os MM são responsáveis por analisar a rota e extrair metadados obtidos a partir da análise dos dados básicos como rota e FPS, junto a configurações, *thresholds* e lógicas específicas. Exemplos de MM implementados neste trabalho

são a velocidade e o tempo em borda. É possível e esperado que o pesquisador possa alterar e criar novos metadados utilizando lógicas específicas para a análise de seu projeto, por isso é possível adicionar novos MM que podem atuar como uma esteira de processamento de dados com MM dependendo de outros MM, criando um código em python e adicionando no sistema.

O módulo de metadados é responsável por calcular novas informações com base nos dados levantados pelo BM. Este módulo é dividido em duas estruturas: a primeira é o invocador de submódulos e a segunda são os submódulos invocados.

O MM foi projetado para ser manipulado pelo usuário, isto é, o usuário pode escrever novos módulos e adicioná-los para que o software execute no momento do processamento.

Portanto, foi necessário desenvolver um invocador de módulos conforme a Figura 15, onde esses módulos podem vir de arquivos python com nomes desconhecidos. O invocador deve encontrar e executar o arquivo passando o objeto contendo todos os dados já calculados pelo BM e pelos MM anteriores, atualizando o objeto com o resultado obtido.

Figura 15 - Algoritmo de invocação de módulos de metadados

```
# carrega todos os arquivos no diretorio
for filename in os.listdir(...):

    if filename.endswith(".py"): # Pesquisa os arquivos python
        module = load_module(...) # Carrega para memoria

        # Executa a interface
        if hasattr(module, 'module_call'):
            data = module.module_call(data)
```

Fonte: Elaborado pelo Autor (2024)

Para criar um novo módulo, basta inserir o arquivo contendo o código (script) na pasta “./src/MetadataModule”, seguindo o padrão da interface mostrado na Figura 16. O arquivo deve conter uma função chamada “*module_call*” que receba um objeto genérico contendo os dados de saída dos outros módulos e retorne o mesmo objeto, alterando os valores via ponteiro.

Figura 16 - Interface válida para módulo de metadata

```
def module_call(data):
    data["isRabbit"] = False
    return data
```

Fonte: Elaborado pelo Autor (2024)

O valor de retorno do módulo deve ser no mesmo formato do parâmetro de entrada, pois ele será utilizado para o módulo seguinte. É importante notar que não há segurança alguma no caso de um módulo alterar o processamento de um módulo anterior ou apagar os dados. A ferramenta foi projetada assim para que o pesquisador possa utilizar em seu potencial máximo sem travas e restrições.

O trabalho prevê a implementação de dois módulos. O primeiro é responsável pelo tempo em voo e tempo em pouso, sendo chamado de módulo de borda (*BorderModule*). Já o segundo é responsável pelo cálculo da velocidade (*SpeedModule*).

O módulo de borda calcula o tempo de voo e pouso. Ele é definido por um valor limite (*threshold*) configurado pelo pesquisador o que gera linhas virtuais. O

threshold definido por padrão é de 100 pixels afastado da borda para o centro. Caso o ponto referente a posição do inseto se localize fora do *threshold*, o inseto está voando. Caso contrário, ele está pousado em uma das bordas. O módulo contabiliza qual foi o tempo em cada borda e então salva esses dados no objeto de saída.

O módulo de velocidade calcula a distância entre dois pontos e calcula o tempo baseado no FPS do módulo básico. Em seguida, é calculada a velocidade e salvo os valores do objeto de saída.

4.2.4 Módulos de exportação

Os módulos de exportação são a interface do sistema com o disco rígido do computador, salvando e carregando arquivos para a memória principal. Também é responsável pelas conversões, geração dos arquivos PDF e exibição dos relatórios.

4.2.4.1 Sistema de persistência de dados

O foco da ferramenta é no uso local, sem dependências externas como internet, permitindo sua execução em computadores sem placas gráficas ou processadores específicos. Por isso, os dados do sistema não possuem proteções como senhas, criptografia ou ofuscamento, facilitando o acesso ao trabalho caso o usuário queira exportar o código-fonte, adicionar um novo MM ou compartilhar perfis de análise.

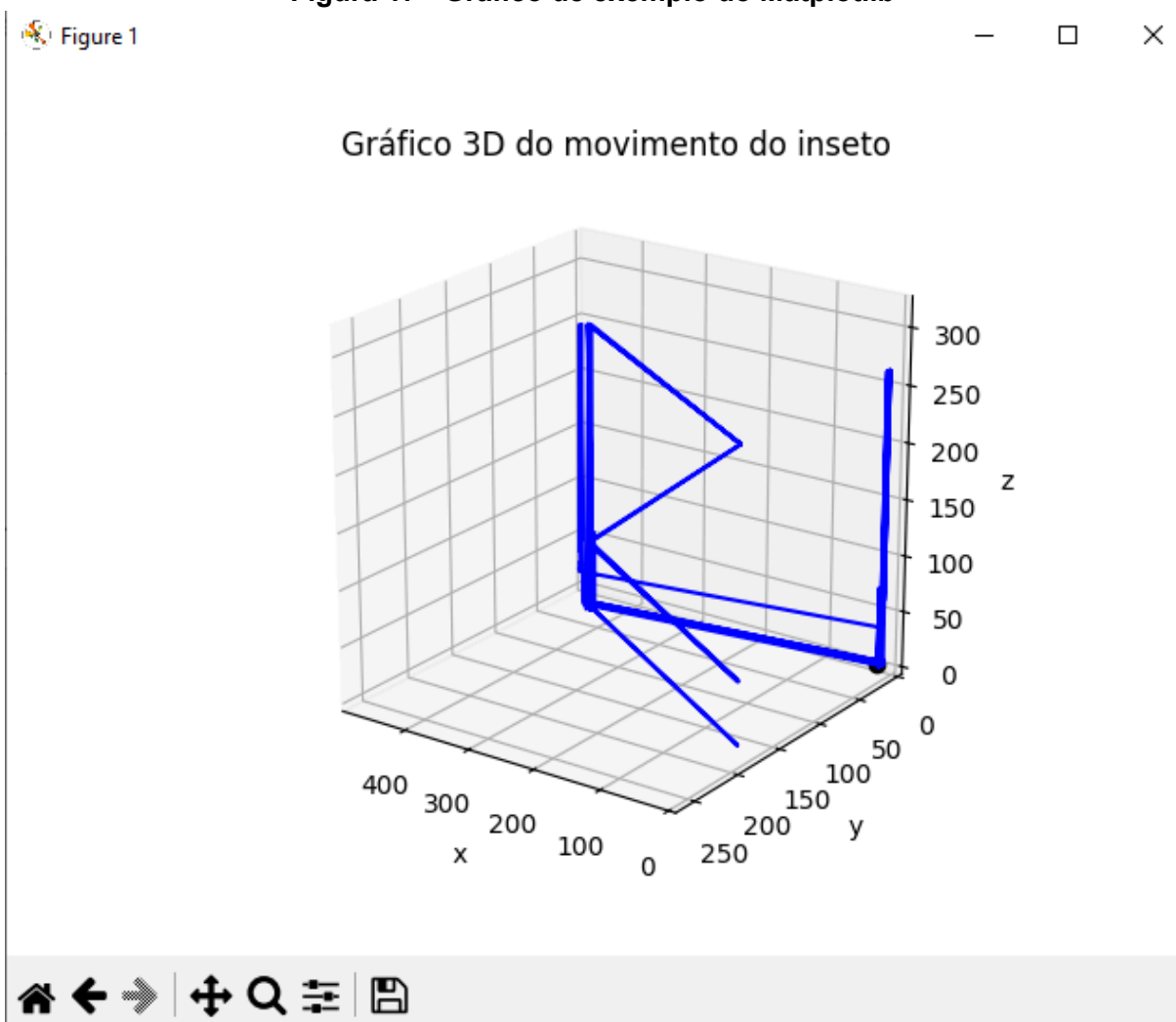
Os dados e metadados gerados pelo sistema com base nas informações fornecidas pelo usuário são salvos localmente em pastas de *cache*, em formato JSON, que são de fácil leitura humana e computacional.

4.2.5 Sistema de exibição dos dados

A rota do inseto é o requisito mais importante para o software, pois sua análise é de extrema importância para os pesquisadores da área comportamental de insetos. Por isso, ela deve ser exibida para o usuário em um gráfico tridimensional interativo.

A exibição desses dados em tela é feita usando a biblioteca Matplotlib desenvolvida por Hunter (2007), que é capaz de renderizar os dados de rota em um gráfico 3D interativo *frame a frame*, onde é possível alterar o ângulo da visão com o mouse, conforme a Figura 17.

Figura 17 - Gráfico de exemplo do Matplotlib



Fonte: Elaborado pelo Autor (2024)

O sistema conta com um submódulo para gerar o relatório dos metadados em um documento PDF. O módulo integra os dados a um template HTML (*HyperText Markup Language*), que é salvo no formato PDF pela biblioteca *Pisa* conforme a Figura 18.

Figura 18 - Imagem de exemplo do pdf

Propriedade	Valor
Quantidade de frames	1574
Largura da Caixa (cm)	1.0
Altura da Caixa (cm)	1.0
Profundidade da Caixa (cm)	1.0
Razão Pixel para cm	558.0
FPS	26
Tempo Borda X	0
Tempo Borda Y	0
Tempo Borda Z	0

Fonte: Elaborado pelo Autor (2024)

5. RESULTADOS

Para a apresentação dos resultados, foram elaborados dois cenários distintos: um para fins de teste durante a elaboração do projeto; e outro para fins práticos, simulando o ambiente controlado esperado no uso prático do pesquisador.

5.1 Cenário 1

O primeiro cenário foi elaborado de forma rústica para ser utilizado como base dos testes durante o desenvolvimento do software, portanto o ambiente não é totalmente controlado.

Para a construção do ambiente foi utilizado um aquário de vidro. Nas paredes foram colados com fita crepe um papel dobrado de folhas A4. Foi utilizado dois celulares para captar os dois ângulos, superior e lateral. Como inseto foi utilizado uma mosca urbana de espécie não identificada. As Figuras 19 e 20 são do primeiro frame do vídeo.

A gravação possui diversos defeitos que não são encontrados em um ambiente profissional e completamente controlado. Contudo esses defeitos destacados em números vermelhos, na Figura 19 e 20, foram chave para o desenvolvimento, pois eles apontaram a necessidade do software utilizar as técnicas de remoção de background, explicadas na Seção 4.2.2.1.

Figura 19 - Primeiro *frame* do vídeo superior da mosca urbana



Fonte: Elaborado pelo Autor (2024)

O ponto número 1 e 2 da Figura 19 são referentes ao reflexo da luz no vidro do aquário. Caso a mosca passasse em baixo desse ponto, o software deixaria de reconhecê-la. Contudo, no vídeo do caso 1, isto não ocorre.

Figura 20 - Primeiro *frame* do vídeo lateral lateral da mosca urbana

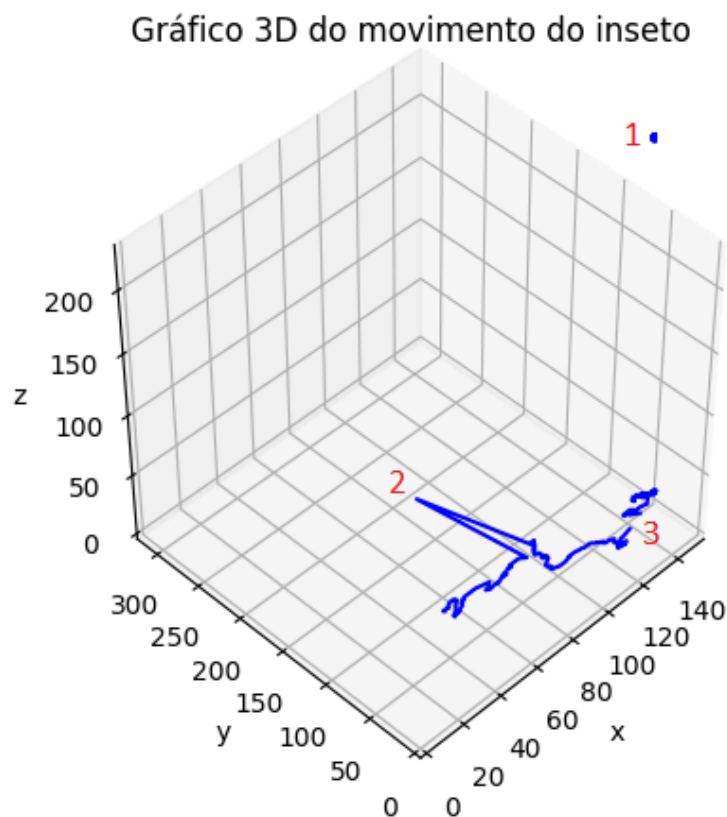


Fonte: Elaborado pelo Autor (2024)

O ponto número 1 da Figura 20 é referente a sombra formada pela iluminação irregular. No momento em que a mosca entra nessa sombra o software parava de reconhecê-la. Da mesma forma, o software constantemente perdia a mosca com pontos escuros na imagem, conforme o ponto número 2 das bordas do aquário de vidro. Para resolver essas questões, foi implementado o algoritmo que gera a máscara de background da Figura 9, que trata das questões estáticas do plano de fundo e soluciona os dois problemas.

O ponto número 3 da Figura 19 e 20 é a fita crepe utilizada para prender a folha A4 ao fundo da imagem. Por conta de sua alta diferença de contraste, foi necessário implementar o algoritmo de remoção de ruídos exemplificado na Figura 10.

O perfil de análise foi configurado com os vídeos e os valores referentes ao tamanho do aquário, os resultados e sua discussão estão a seguir. Observando as Figuras 21, 22 e 23, é possível compreender que o software identificou com sucesso a rota do inseto, mesmo em um vídeo de testes em um ambiente não controlado, demonstrando resiliência na identificação mesmo em cenários não ideais.

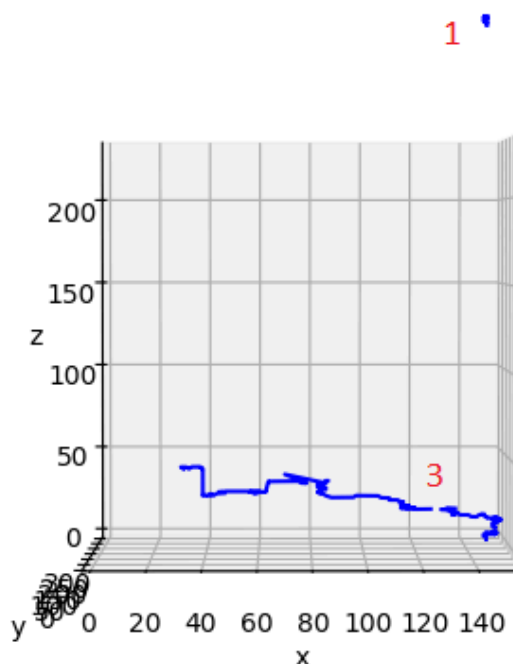
Figura 21 - Gráfico de rota visão diagonal

Fonte: Elaborado pelo Autor (2024)

Contudo, os gráficos possuem três pontos a serem discutidos identificados com números vermelhos. O ponto número 1 denota o momento final do vídeo em que o momento em que o inseto voou rapidamente e o software não conseguiu identificar ele durante alguns *frames*. Quando o software conseguiu identificar o inseto novamente, ele já estava na parte superior do ambiente.

Figura 22 - Gráfico de rota visão lateral

Gráfico 3D do movimento do inseto



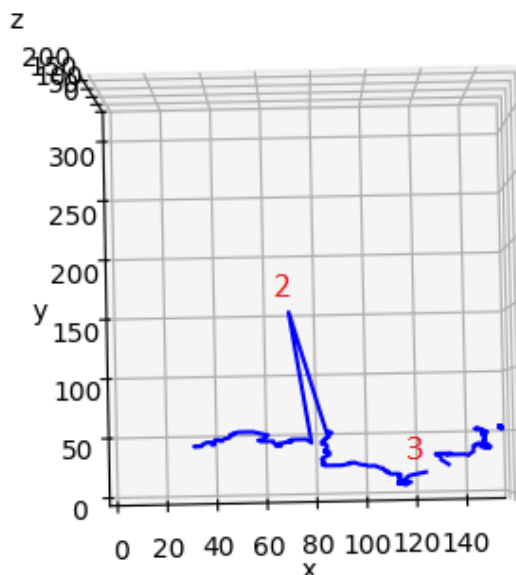
Fonte: Elaborado pelo Autor (2024)

O ponto número 2, visível nas Figuras 21 e 23, é relevante porque representa um único frame em que o sistema confunde o plano de fundo com o inseto. Isso ocorre quando a mosca começa a bater as asas gerando um ruído leve na máscara de remoção de fundo. Ao mesmo tempo, o ruído de fundo provocado pela luz irregular refletida na fita crepe da Figura 19 fica mais intenso, gerando o erro momentâneo da posição.

O ponto número 3, destacado na Figura 23, refere-se a um conjunto de 13 *frames* nos quais o inseto fica indetectável para o software. Isso ocorre porque o inseto está de costas para a câmera lateral e suas asas refletem a luz de forma a se tornar indistinguível do plano de fundo.

Figura 23 - Gráfico de rota visão superior

Gráfico 3D do movimento do inseto



Fonte: Elaborado pelo Autor (2024)

A quebra de linha pode ser vista tanto no ponto número 1 quanto no ponto número 3, onde a linha azul não segue a posição do inseto. Já no ponto número 2, o inseto foi identificado em todos os *frames*, mesmo que erroneamente, o software não introduz uma quebra e entende que só foi uma movimentação rápida.

A Figura 24 apresenta o relatório dos metadados do cenário 1 gerados pelo módulo de exportação. A quantidade de *frames* é um valor único uma vez que o sistema para de contabilizar os *frames* ao final do vídeo mais curto. O tamanho da caixa foi configurado com o valor fantasia de 10 cm para todos os lados, a fim de simplificar o resultado. Ambos os vídeos foram gravados a 30 FPS.

Figura 24 - Relatório de metadados do cenário 1

Dados do processamento: aquario

Propriedade	Valor
Quantidade de frames	562
Largura da Caixa (cm)	10.0
Altura da Caixa (cm)	10.0
Profundidade da Caixa (cm)	10.0
Razão Pixel para cm	30.8
FPS	30
Tempo Borda X	444

Fonte: Elaborado pelo Autor (2024)

5.2 Cenário 2

Os segundo e terceiro cenários de teste foram elaborados junto a um pesquisador da área. O vídeo consiste em uma barata dentro de um recipiente de plástico, sendo capturada por uma única câmera, conforme mostrado na Figura 25.

Figura 25 - Primeiro *frame* do vídeo superior da Barata

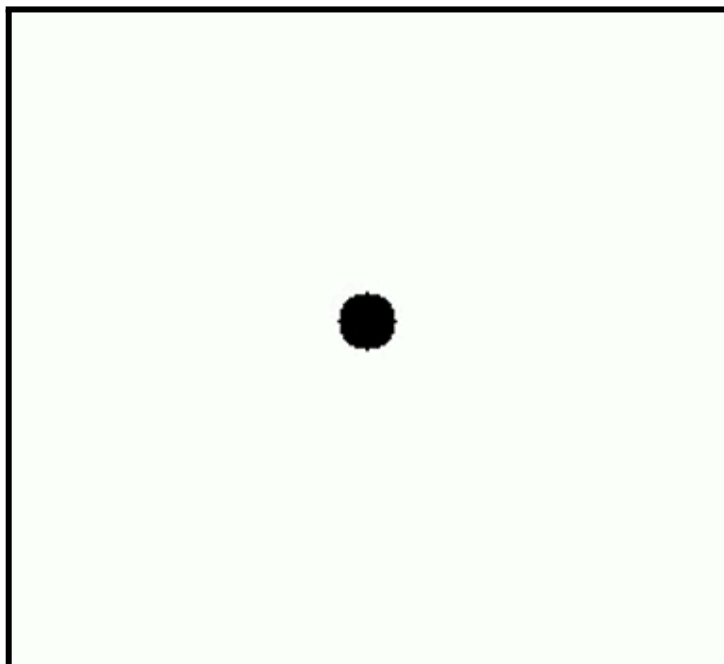


Fonte: Disponibilizado por Cháriston André Dal Belo (2024)

Como não havia outro vídeo contendo a visão lateral, foram elaborados dois vídeos de um círculo em um fundo branco. No primeiro vídeo, o círculo permanece estático, permitindo a análise apenas da movimentação da barata em um ambiente 2D. No segundo vídeo, o círculo se move de cima para baixo em um movimento retilíneo e uniforme, com duração de 1 segundo, simulando uma movimentação nas paredes, conforme ilustrado na Figura 25.

O primeiro cenário, com o vídeo estático, gerou os pontos de rota nos eixos x e y do vídeo mostrado na Figura 25. No entanto, o ponto referente ao eixo z (altura) não apresentou dados, pois o sistema não conseguiu identificar o objeto de interesse, como ilustrado na Figura 26, devido à ausência de movimento. Isso indica que, caso o inseto não se mova ou permaneça inativo em um dos eixos, o sistema não o identifica corretamente. O gráfico gerado é exibido sem nenhum ponto, uma vez que o ponto tridimensional da rota está incompleto.

Figura 26 - Primeiro *frame* do vídeo simulando a movimentação lateral

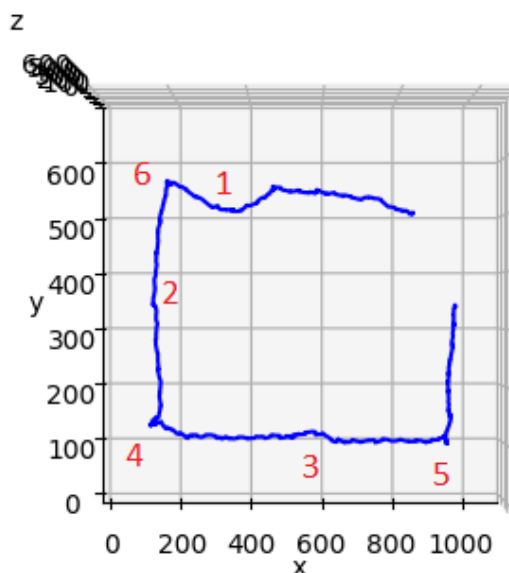


Fonte: Elaborado pelo Autor (2024)

Contudo, para exemplificar um comportamento e limitação do software, foi necessário burlar o problema inserindo manualmente o valor de altura nos pontos da rota, o que resultou no gráfico apresentado na Figura 27.

Figura 27 - Gráfico de rota visão superior

Gráfico 3D do movimento do inseto



Fonte: Elaborado pelo Autor (2024)

Com a visualização em ângulo superior, é possível compreender o caminho que o inseto fez durante o vídeo. O sistema registrou com sucesso as mudanças relevantes na rota, como mostrado na Figura 27. Na marcação 1, o inseto desviou do limite da caixa, fazendo uma curva. Nas posições 2 e 3, houve uma correção de curso por parte do inseto quando sua antena tocou o canto da caixa, levando-o a desviar para o centro. Nos momentos 4, 5 e 6, o inseto ficou parado por um breve período enquanto inspecionava os cantos da caixa com suas antenas.

A Figura 28 exibe os dados recuperados a partir da execução dos metadados. A quantidade de *frames* é um valor único, pois o sistema interrompe a contagem ao final do vídeo mais curto. O tamanho da caixa foi configurado com o valor fantasia de 10 cm (aleatório) para todos os lados. Os vídeos foram gravados a 26 FPS, os tempos em borda foram calculados com base somente na posição x (largura) e y (comprimento). O tempo em borda Z (altura) não foi configurado para esse cenário.

Figura 28 - Resultados dos metadados**Dados do processamento: monografia-cenario-2**

Propriedade	Valor
Quantidade de frames	1573
Largura da Caixa (cm)	10.0
Altura da Caixa (cm)	10.0
Profundidade da Caixa (cm)	10.0
Razão Pixel para cm	39.8
FPS	26
Tempo Borda X	435
Tempo Borda Y	773
Tempo Borda Z	0

Fonte: Elaborado pelo Autor (2024)

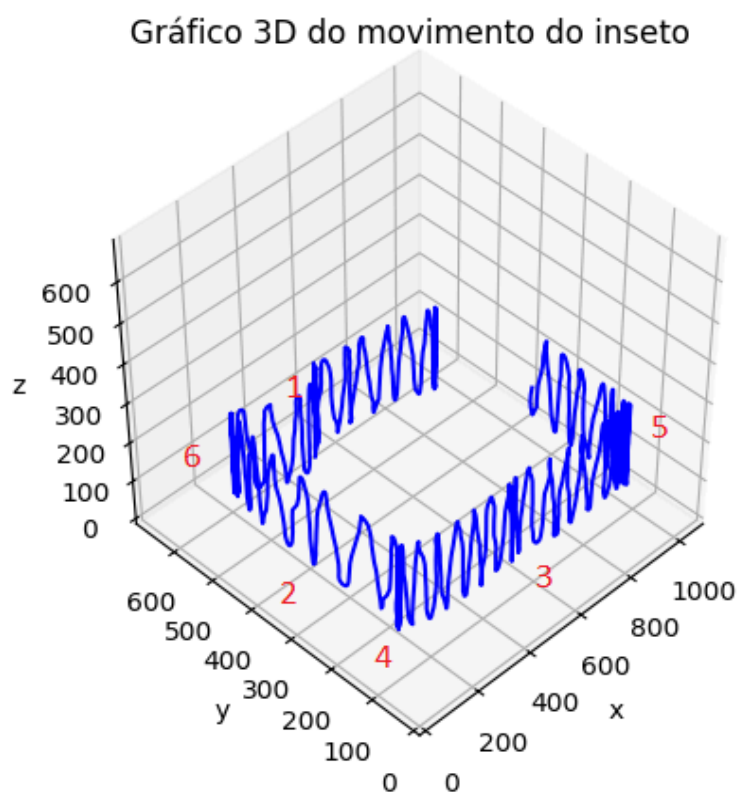
5.3 Cenário 3

O cenário 3 foi elaborado utilizando os mesmos dados de posição nos eixos x e y gerados na análise dos módulos básicos mostrados na Figura 25. No entanto, desta vez foi utilizado o segundo vídeo artificial, onde o círculo se move de cima para baixo em um caminho retilíneo, completando o movimento em 1 segundo a 30 FPS.

É possível ver como o sistema integrou as posições dos eixos de largura (y) e profundidade (x), referentes à movimentação do inseto, com a movimentação artificial no eixo de altura (z).

É possível observar, na Figura 29, que o comportamento analisado no cenário anterior continua sendo exibido no gráfico junto ao movimento oscilante da altura. Os pontos de interesse enumerados em vermelho coincidem com o da Figura 27, porém ficam mais evidentes quando observados na mesma angulação, conforme a Figura 30.

Figura 29 - Gráfico de rota visão diagonal

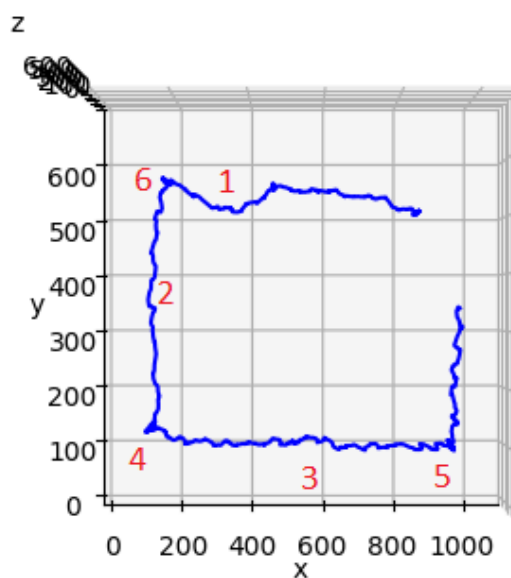


Fonte: Elaborado pelo Autor (2024)

Ao comparar o gráfico da Figura 27, que possui altura única, com o gráfico da Figura 28, onde a altura oscila, é visível que o primeiro gráfico está mais nítido, já que o segundo gráfico apresenta maior oscilação. Isto ocorre pois a diferença na altura somada à perspectiva imposta pelo gráfico gera uma distorção visual nos gráficos. Contudo, após analisar os pontos da rota do cenário 2 e 3, é possível compreender que a única mudança é no ponto z de altura. Portanto o artefato gerado no gráfico é uma limitação visual do módulo de exportação e não um erro no algoritmo de CV.

Figura 30 - Gráfico de rota ângulo superior do cenário 3

Gráfico 3D do movimento do inseto



Fonte: Elaborado pelo Autor (2024)

Os resultados do cenário 3 são exatamente iguais aos do cenário 2, uma vez que os vídeos base são idênticos.

6. CONCLUSÃO

A área de pesquisa da análise comportamental de insetos é relevante para compreender não só como insetos se comportam mas também como agrotóxicos rurais e inseticidas urbanos impactam suas vidas. Entender o efeito de produtos químicos sobre insetos traz benefícios, como a prevenção de superdosagens de venenos e a melhoria da eficiência de seu uso em contextos urbanos e rurais.

O software construído atinge o objetivo deste trabalho, que é criar uma ferramenta para auxiliar pesquisadores na extração de metadados para a análise do comportamento de insetos.

A capacidade do sistema de aceitar novos módulos de análise de forma simples, bastando adicionar o arquivo contendo o código Python em uma pasta, será de grande valor para a criação de um ecossistema de módulos, onde pesquisadores possam utilizar, compartilhar e reutilizar módulos customizados entre si.

No entanto, há possibilidades de melhorar o desempenho do sistema substituindo partes de sua implementação por bibliotecas desenvolvidas em linguagens mais rápidas e eficientes, como C e Rust.

Há diversas melhorias que podem ser feitas em trabalhos futuros, como listado abaixo:

- Melhorar o sistema de exportação de dados a fim de facilitar o compartilhamento de perfis de análise;
- Implementar testes automatizados padronizados;
- Padronizar os módulos de interface, módulos básicos e modos de exportação, para que sejam de fácil alteração assim como os módulos de metadados;
- Implementar um sistema para que o usuário possa configurar a ordem de execução dos metadados;
- Implementar otimização no módulo básico para dividir o frame em sub regiões de análise com base na movimentação do inseto; e
- Implementar otimização no módulo básico para a análise do vídeo lateral, simplificando-o para uma única coluna de altura ao invés de uma matriz.

REFERÊNCIAS

ARAÚJO, Isabelle Maria Mendes De; OLIVEIRA, Ângelo Giuseppe Roncalli Da Costa. AGRONEGÓCIO E AGROTÓXICOS: IMPACTOS À SAÚDE DOS TRABALHADORES AGRÍCOLAS NO NORDESTE BRASILEIRO. Trabalho, Educação e Saúde, v. 15, n. 1, p. 117–129, 2016.

BELCHIOR, Diana Cléssia Vieira; SARAIVA, Althiéris de Souza; LÓPEZ, Ana Maria Córdova; et al. IMPACTOS DE AGROTÓXICOS SOBRE O MEIO AMBIENTE E A SAÚDE HUMANA. v. 34, n. 1, 2014.

CHEN, Ching-Hsin; CHIANG, Ann-Shyn; TSAI, Hung-Yin. Three-Dimensional Tracking of Multiple Small Insects by a Single Camera. Journal of Insect Science, v. 21, n. 6, p. 14, 2021.

FARDER-GOMES, Cliver Fernandes; FERNANDES, Kenner Moraes; BERNARDES, Rodrigo Cupertino; et al. Harmful effects of fipronil exposure on the behavior and brain of the stingless bee *Partamona helleri* Friese (Hymenoptera: Meliponini). Science of The Total Environment, v. 794, p. 148678, 2021.
FILHO, Júlio De Mesquita. UNIVERSIDADE ESTADUAL PAULISTA.

FRY, S.N.; BICHSEL, M.; MÜLLER, P.; et al. Tracking of flying insects using pan-tilt cameras. Journal of Neuroscience Methods, v. 101, n. 1, p. 59–67, 2000.

GOMES, Davy Soares; MIRANDA, Franciane Rosa; FERNANDES, Kenner Moraes; et al. Acute exposure to fungicide fluazinam induces cell death in the midgut, oxidative stress and alters behavior of the stingless bee *Partamona helleri* (Hymenoptera: Apidae). Ecotoxicology and Environmental Safety, v. 282, p. 116677, 2024.

HAYNES, K F. Sublethal Effects of Neurotoxic Insecticides on Insect Behavior. Annual Review of Entomology, v. 33, n. 1, p. 149–168, 1988.

LOPES, Lucas Teodoro; NOCELLI, Roberta Cornélio Ferreira; MONQUERO, Patricia Andrea. Os impactos dos herbicidas em abelhas nativas brasileiras. 2024.

SZELISKI, Richard. Computer Vision: Algorithms and Applications.

HUNTER, John D. Matplotlib: A 2D Graphics Environment. Computing in Science & Engineering, v. 9, n. 3, p. 90–95, 2007.

PYTHON SOFTWARE FOUNDATION. *tkinter* — Python interface to Tcl/Tk. 2024. Documentação Python 3.13.0. Disponível em: <https://docs.python.org/3/library/tkinter.html>. Acesso em: 10 out. 2024.

BERNARDES, Rodrigo Cupertino; LIMA, Maria Augusta Pereira; GUEDES, Raul Narciso Carvalho; et al. Ethoflow: Computer Vision and Artificial Intelligence-Based Software for Automatic Behavior Analysis. Sensors, v. 21, n. 9, p. 3237, 2021.