

UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO TECNOLÓGICO
DEPARTAMENTO ENGENHARIA ELÉTRICA E ELETRÔNICA
CURSO DE ENGENHARIA ELÉTRICA

Luiz Henrique Fernandes Salmonson

Processamento de imagem em código aberto para extração de características cinemática e morfológicas do comportamento animal

Florianópolis
2019

**Ficha de identificação da obra elaborada pelo autor,
através do Programa de Geração Automática da Biblioteca Universitária da UFSC.**

Salmonson, Luiz Henrique Fernandes

Processamento de imagem em código aberto para extração de características cinemática e morfológicas do comportamento animal / Luiz Henrique Fernandes Salmonson ; orientador, Daniela Ota Hisayasu Suzuki, 2019.

44 p.

Trabalho de Conclusão de Curso (graduação) -
Universidade Federal de Santa Catarina, Centro Tecnológico,
Graduação em Engenharia Elétrica, Florianópolis, 2019.

Inclui referências.

1. Engenharia Elétrica. 2. Etologia. 3. Processamento de imagens. 4. descritores morfológicos por análise de vídeo. 5. descritores cinemáticos por análise de vídeo. I. Suzuki, Daniela Ota Hisayasu . II. Universidade Federal de Santa Catarina. Graduação em Engenharia Elétrica. III. Título.

Luiz Henrique Fernandes Salmonson

Processamento de imagem em código aberto para extração de características cinemática e morfológicas do comportamento animal

Este Trabalho de Conclusão de Curso (TCC) foi julgado adequado para obtenção do Título de Bacharel em Engenharia Elétrica e aprovado, em sua forma final, pela Banca Examinadora

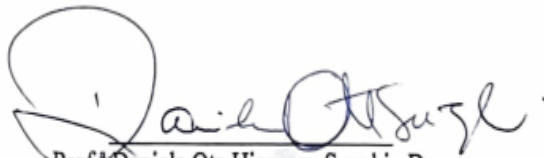
Florianópolis, 11 de dezembro de 2019.



Prof. Jean Vianei Leite, Dr.

Coordenador do Curso de Graduação em Engenharia Elétrica

Banca Examinadora:



Prof.ª Daniela Ota Hisayasu Suzuki, Dra.


Orientadora

Universidade Federal de Santa Catarina



Eng. João Antônio Marcolan, M.e.

Universidade Federal de Santa Catarina



Eng. Guilherme Brasil Pintarelli, M.e.

Universidade Federal de Santa Catarina

Este trabalho é dedicado aos meus professores, amigos ,colegas de classe e querida família.

AGRADECIMENTOS

Gostaria de agradecer à minha orientadora, Prof. Dra. Daniela Suzuki pela confiança, direcionamento e todo esforço para completar essa etapa na minha formação.

Agradeço também aos membros IEB e especial do laboratório de Bioengenharia, por me aceitarem no convívio do dia a dia e auxiliarem em minhas dúvidas.

Agradeço a minha família e os meus amigos por estarem ao meu lado nos momentos de dificuldades e de alegria.

RESUMO

A ferramenta computacional EthoWatcher OpenSource é utilizada em pesquisas com animais para gerar relatórios de etologia, analisar experimentos em vídeo e extrair variáveis morfológicas e cinemáticas da filmagem. A ferramenta utiliza métodos de processamento de imagens digitais e entradas do operador para extração dos dados desejados. Por meio de análise dos módulos de processamento de imagens da ferramenta foram identificadas técnicas de tratamentos utilizados e como os dados morfológicos são extraídos. Foram propostas modificações para os tratamentos e descritores existentes, assim como a adição de novos descritores. O descritor de ângulo do objeto de interesse era limitado de 0° a 90° e a nova forma de cálculo obtém valores entre 0° e 180° . Adicionando ao tratamento da imagem a dilatação resultou em menos cenários em que os dados extraídos eram perdidos. Foi criado o descritor de distância do animal do experimento (OI) a área de interesse ao observador, o qual adiciona as capacidades de análise da ferramenta. Os descritores modificados e criados apresentaram resultados consistentes e são reprodutíveis independentemente para o mesmo vídeo e calibração. A adição de 90° ao descritor de ângulo torna sua informação mais ampla e o descritor de distância do animal do experimento (OI) a área de interesse ao observador poderá ser usado em pesquisas para indicar a proximidade de um animal a área de interesse ou áreas de aversão.

Palavras-chave: Etologia. Processamento de imagens. Análise de vídeo. Rastreamento em vídeo.

ABSTRACT

The computational tool EthoWatcherOpenSource is used in research with animals to generate ethology reports, analyze video experiments and extract morphological and cinematic information from video. The software uses digital image processing methods and operator input to generate the intended data. Analyzing the image processing modules from the tool, it was identified the techniques and treatment used and how morphological data is extracted. Changes are proposed to the image treatment and existing descriptors, and new descriptors are also added. The angle descriptor was limited for values in the range of 0° to 90° and the proposed method outputs a range of 0° to 180° . Adding dilation in the image treatment process resulted in less cases of data loss. It is created the distance from the experiment animal (OI) to the area of interest for the observer, which adds to the descriptors output of the software. The proposed modifications and created descriptor have shown to work as intended and are reproducible independently for the same video and calibration. Adding 90° to the angle descriptor gives it a larger range and the distance from the experiment animal (OI) to the area of interest for the observer can be used in research to describe the animal proximity to areas of interest or aversion.

Keywords: Ethology. Image Processing. Video analysis. Video tracking.

LISTA DE FIGURAS

Figura 1 - Elemento estruturante com tamanho 3.....	29
Figura 2 – Retorno de “cv::minAreaRect” e seus parâmetros.....	31
Figura 3 – Em (a) há três recortes do primeiro vídeo, contendo um retângulo no centro de um relógio. Em (b) um recorte do segundo vídeo, contendo distribuição espacial de moscas <i>Drosophila Megalonaster</i> (imagem de 8 minutos e 45 segundos do vídeo).	32
Figura 4 – Em (a) não há dilatação Relógio 10 – Roi, Resultado da subtração do plano de fundo, Zoom do resultado da erosão e frame original com caixa e indicação de ângulo). Em (b) (com dilatação relógio 10 - Roi, Resultado da subtração do plano de fundo, Zoom do resultado da erosão seguida de dilatação e frame original com caixa e indicação de ângulo)	36
Figura 5 – Histograma da distância entre mosca e área de interesse experimento.....	38
Figura 6 – Comparativo de valores para descritor de ângulo.	38

LISTA DE TABELAS

Tabela 1 – As siglas utilizadas durante construção dos códigos.	28
Tabela 2 – Nomeação de classes e variáveis utilizadas durante construção dos códigos.	28
Tabela 3 – Dados extraídos do primeiro vídeo.....	35
Tabela 4 – Dados extraídos do segundo vídeo.	36

LISTA DE ABREVIATURAS E SIGLAS

LANC - Laboratório de Neurociência Computacional

IEB - Instituto de Engenharia Biomédica

UFSC - Universidade Federal de Santa Catarina

INPI - Instituto Nacional da Propriedade Industrial

ROI - Região objeto de interesse

OI - Objeto de interesse

AIO - Área de interesse ao observador

AE - Área do experimento

CM - Centro de massa

CSV – Arquivo de valores separados por vírgula (*Comma Separated Values*)

SUMÁRIO

1	INTRODUÇÃO	15
1.1	OBJETIVOS.....	17
1.1.1	Objetivo geral	17
1.1.2	Objetivos Específicos.....	17
2	FUNDAMENTAÇÃO TEÓRICA.....	18
2.1	PROCESSAMENTO DE IMAGENS	18
2.1.1	Subtração do Plano de Fundo	18
2.1.2	Conversão para Escala de Cinza.....	19
2.1.3	Limiarização	19
2.1.4	Processo Morfológico	19
2.1.5	Momentos de um objeto.....	20
2.2	A ORGANIZAÇÃO DO ALGORITMO ETHOWATCHER OS	21
2.3	DESCRITORES CINEMÁTICOS E MORFOLÓGICOS.....	22
2.4	FERRAMENTAS DE DESENVOLVIMENTO.....	23
2.4.1	Qt.....	24
2.4.2	C++.....	24
2.4.3	OpenCV	24
2.5	BOAS PRÁTICAS EM CÓDIGOS DE PROGRAMAÇÃO.....	25
3	MATERIAIS E MÉTODOS.....	27
3.1	ASPECTOS UTILIZADOS NO ALGORITMO	27
3.1.1	Filtros morfológicos utilizados	29
3.1.2	Descritores Cinemáticos e Morfológicos	30
3.1.3	Validação dos descritores implementados	31
4	RESULTADOS.....	35

4.1	COMPARAÇÃO DO ALGORITMO COM ERODE E DILATAÇÃO E SOMENTE ERODE	35
4.2	DESCRITOR DISTÂNCIA DO OBJETO DE INTERESSE A REGIÃO DE INTERESSE AO OBSERVADOR	37
4.3	DESCRITOR DE ÂNGULO	38
5	DISCUSSÃO	39
6	CONCLUSÃO	41
	REFERÊNCIAS	43

1 INTRODUÇÃO

O estudo de comportamento animal é importante para o entendimento do efeito de uma série de fármacos no campo da neurociência. Porém, o uso de animais vertebrados em pesquisa tem sofrido críticas no campo ético. O que tem levado que a comunidade científica venha tendo uma série de iniciativas para tentar reduzir o uso de animais, refinar as metodologias usadas e substituir o animal vertebrado(ECKERT, 2014).

Essas iniciativas éticas já resultaram em alguns esforços. Como por exemplo na filmagem dos experimentos comportamentais. A filmagem diminuiu o risco de o experimentador ter apenas uma oportunidade de visualizar a experiência e poder reusar o sinal biológico extraído do animal para uma série de pesquisas. Com a disponibilidade da captura digital, grandes bancos de imagens foram formados. Porém classificar comportamento de forma categórica tem se mostrado um trabalho laborioso sujeito a uma certa incerteza, que atualmente vem sendo estudada (MARCHESINI et al., 2019), muitas vezes causadas por vieses humanos (MARCOLAN, 2017). Com isso tem se explorado técnicas para mitigar essas incertezas da análise humana.

Uma das técnicas para reduzir a influência humana e refinar a análise de dados extraídos dos vídeos etológicos, é utilizar extratores baseados nas técnicas de processamento de imagem. O objetivo da análise de imagens digitais é obter resultados objetivos e reprodutíveis, influenciando o experimento ao mínimo e proporcionando dados equivalentes aos métodos científicos consolidados. A filmagem e coleta de dados etográficos por análise da imagem proporciona uma riqueza na quantidade de dados obtidos em menor tempo, devido a exclusão do fator humano. Em contrapartida, os dados de interesse são definidos pelo experimentador e a ferramenta utilizada deve ser compatível com seus objetivos(ANDERSON; PERONA, 2014).

Existem algumas ferramentas pagas que permitem extrair variáveis cinemáticas e morfológicas do experimento animal como o Ethovision produzidos pela Noldus(NOLDUS, [s.d.]). E outras gratuitas como JWatcher (BLUMSTEIN; DANIEL; EVANS, [s.d.]) . Recentemente o Laboratório de Neurociência Computacional (LANC) do Instituto de Engenharia Biomédica (IEB) da Universidade Federal de Santa Catarina (UFSC) também registrou no Instituto Nacional da Propriedade Industrial (INPI) dois softwares etográficos que permitem realizar a análise cinemática, morfológica e categórica dos experimentos. O EthoWatcher (CRISPIM JUNIOR et al., 2012) registrado no número 09285-3. E a sua versão

de código aberto o EthoWathcer Open-Source registrado no número BR512019002103-7(MARCOLAN, 2017).

A primeira versão do EthoWatcher foi baixada por mais de de 1100 pessoas em mais de 50 países, totalizando mais de 1600 downloads. Essa ferramenta já foi utilizada em diversos tipos de estudos comportamentais. As anotações e críticas feitas pelos usuários do programa incentivaram a criação de várias características do EthoWatcher Open Source(CROZARA, 2017). A versão mais atual foi pensada para atender essas críticas e requisições e também permitir que a comunidade científica contribua mais ativamente do processo de construção de funcionalidades do software. A alternativa de criar a ferramenta em um projeto de código aberto vem para amenizar o problema causada pelo abandono do software (do ingles *abandonware*). Essa situação é encontrada em alguns softwares, que por algum motivo os desenvolvedores parem de atualizar, mas ainda há uma comunidade ativa de usuários;

Pelas questões éticas, mencionadas anteriormente, tem se tentado substituir o uso de camundongos no experimento do nado forçado por moscas *Drosophila Melanogaste* (ECKERT, 2014). Esse experimento do nado forçado é muito usado para analisar o efeito de substâncias químicas potencialmente antidepressivas(DOMINGUES, 2015). Eckert, testou uma nova padronização no teste com moscas para fármacos com efeito antidepressivos. Em seu trabalho, Eckert usou o EthoWacher OS versão Beta 48 e fez considerações apenas a respeito do sinal categórico do seu experimento. Isso aconteceu devido ao fato que a o processamento de múltiplas áreas de interesse do experimento, o processamento que permite a análise de vários animais, demorava muito para ser realizado com o software.

Esse fato moveu o desenvolvimento de uma biblioteca de processamento de imagem em código aberto para extração de características cinemática e morfológicas do comportamento animal. Que também, construiu descritores cinemáticos e morfológicos que pudessem ser utilizados nesse experimento.

Este trabalho trás evolução das atividades realizadas no laboratório de Laboratório de Bioengenharia na área de pesquisa - Desenvolvimento de sistema de etografia e de análise comportamental baseados em processamento digital de imagens. O desenvolvimento de um software é um processo agregador de contribuições no planejamento, código, documentação e pesquisa. O software EtowatcherOS atualmente apresenta funcionalidades e documentação como descrito em (MARCOLAN, 2017). Um dos módulos fundamentais do seu funcionamento é a biblioteca de análise de imagem, a qual está defasada em relação as versões mais atuais das ferramentas onde foi desenvolvida, o QtCreator e a biblioteca de visão computacional

OpenCV. Assim, foi realizada análise, desenvolvimento de modificações e documentação desse módulo utilizando ferramentas e métodos atualizados.

1.1 OBJETIVOS

São listados os seguintes objetivos gerais e específicos.

1.1.1 Objetivo geral

Desenvolver e testar um algoritmo em C++ de código aberto para a extração de variáveis cinemáticas e morfológicas de vídeos de animais em situações experimentais. O algoritmo segue a mesma ordem de operações dos trabalhos anteriores (CRISPIM JUNIOR et al., 2012; CROZARA, 2017; MARCOLAN, 2017), adicionando etapas de cálculo onde modificado. Seguindo as diretrizes ágeis de prototipação de software e programação orientada ao objeto.

1.1.2 Objetivos Específicos

Para atingir os objetivos é necessário conclusão dos seguintes objetivos específicos:

- Analisar as ferramentas de análise de imagem do software EthoWatcher OS
- Reimplementar as ferramentas de processamento de imagem do EthoWatcher OS e adicionar novos descritores em um código aberto;
- Validação dos descritores desenvolvidos;

2 FUNDAMENTAÇÃO TEÓRICA

As imagens digitais têm parâmetros diferentes dependendo do sensor, digitalização e compressão usados. Existem diversas técnicas para análise de imagens digitais, considerando arquivos de imagens estáticas e vídeos, as informações de interesse gravadas no arquivo são a matriz de intensidade das cores, altura, largura, duração e número de quadros por segundo. A partir dessas informações são realizadas as funções de tratamento e extração de dados nas imagens. Para um obter o resultado adequado é necessário adequar os parâmetros de entrada dos métodos as especificidades de cada cenário.

Nessa secção são apresentadas técnicas de processamento de imagens aplicáveis ao trabalho, organização do EthoWatcher, descritores cinemáticos e morfológicos, ferramentas de desenvolvimento e boas práticas de programação.

2.1 PROCESSAMENTO DE IMAGENS

Observa-se que as técnicas a seguir foram selecionadas por *brainstorm* com o pesquisador João Marcolan do Laboratório de Bioengenharia. Estas são para auxiliar em problemas levantados como perda de informação e dificuldades em determinar o objeto de interesse, assim como extrair informação com os descritores propostos por Crispim (CRISPIM JUNIOR, 2011)

2.1.1 Subtração do Plano de Fundo

A subtração do plano de fundo entrega a diferença absoluta entre um pixel na mesma posição em imagens diferentes. Para isso é necessária uma imagem de referência de plano de fundo e a imagem a ser tratada. O resultado entrega uma imagem onde elementos de fundo são diminuídos, portanto itens de interesse não pertencentes ao plano de fundo são destacados.

Esse processo é influenciado por fatores de intensidade de cores e luz e a posição da imagem. A diferença de cor do plano de fundo e do objeto a ser adicionado a cena determina o valor máximo de intensidade do resultado. Variação na posição do plano de fundo gera artefatos inconsistentes com o esperado, dependendo das variações originais da imagem.

2.1.2 Conversão para Escala de Cinza

Para certas análises e técnicas de processamento é necessário que o parâmetro de entrada esteja em escalas de cinza. O processo de conversão para escala de cinza realiza a média ponderada dos canais coloridos, convertendo para um valor de intensidade em escala única de cinza. Os valores de ponderação da média dependem do método utilizado (QUEIROZ; GOMES, 2001).

2.1.3 Limiarização

Para executar algoritmos binários na imagem é necessário representá-la de forma binária, ou seja, preto (de valor “zero”) e branco (de valor máximo “255”). A conversão por limiarização funciona analogamente a um filtro passa alta, onde valores de intensidade abaixo do limiar são mapeados como zero (preto) e valores iguais ou acima do limiar são mapeados como valor máximo.

2.1.4 Processo Morfológico

Processos morfológicos são métodos para extração de componentes, descrição ou transformação da topologia de objetos em imagens utilizando um elemento estruturante. Um elemento estruturante é um conjunto, usualmente caracterizado como uma forma geométrica, para comparar com subconjuntos da imagem sendo analisada e executar operações dependentes do resultado dessas comparações (QUEIROZ; GOMES, 2001). São explicadas nessa subseção erosão e dilatação.

2.1.4.1 Erosão

Operação iterativa onde o elemento estruturante é comparado para cada pixel da imagem analisada e gera uma nova imagem. O valor do pixel na nova imagem é zero caso um dos pixels do elemento estruturante esteja sobreposto a um pixel com valor nulo. O valor do pixel na nova imagem é positivo caso todos os pixels do elemento estruturante esteja sobreposto

a um pixel com valor positivo. A erosão retira pixels da periferia do objeto, fazendo com que pequenas partículas sejam apagadas e objeto de interesse tenha suas dimensões reduzidas (QUEIROZ; GOMES, 2001).

2.1.4.2 Dilatação

Método iterativo onde o elemento estruturante é comparado para cada pixel da imagem analisada e gera uma nova imagem. O valor do pixel na nova imagem é positivo caso um dos pixels do elemento estruturante esteja sobreposto a um pixel com valor positivo. O valor do pixel na nova imagem é nulo caso todos os pixels do elemento estruturante esteja sobrepostos a pixels com valor nulo. O filtro por dilatação adiciona pixels na borda dos contornos em que é aplicado e aumenta suas dimensões.

2.1.5 Momentos de um objeto

Os momentos de um objeto em uma imagem são características espaciais de representação em um espaço cartesiano. Em imagens digitais os cálculos de momentos binários são utilizados para encontrar a área e o centro de massa para uma forma.

Definindo que a matriz de imagem $I(x,y)$ contém valores de 0 ou 1, tem tamanho horizontal i e vertical j , os momentos m_{pq} são calculados pela equação (1).

$$m_{pq} = \sum_i \sum_j x_i^p * y_j^q * I(x, y) \quad (1)$$

A área do objeto na imagem é calculada pela soma de todos os pixels que compõem o objeto, ou seja, o momento m_{00} , conforme equação (2).

$$\text{Área} = m_{00} = \sum_i \sum_j x_i^0 * y_j^0 * I(x, y) \quad (2)$$

Para calcular o centro de massa em cada eixo, pode ser utilizada a soma dos pixels relativa a posição e dividido pelo número de pixels do objeto, o m_{00} , para obter a posição relativa do centro de massa. Os momentos m_{10} e m_{01} são calculados com as equações (3) e (4).

O centro de massa do objeto de interesse é um descritor da sua posição no plano cartesiano (CROZARA, 2017). Os centros de massas “x” e “y” é calculado com as equações (5) e (6).

$$m_{10} = \sum_i \sum_j x_i^1 * y_j^0 * I(x, y) \quad (3)$$

$$m_{01} = \sum_i \sum_j x_i^0 * y_j^1 * I(x, y) \quad (4)$$

$$\text{CentroMassa}_x = \frac{m_{01}}{m_{00}} \quad (5)$$

$$\text{CentroMassa}_y = \frac{m_{10}}{m_{00}} \quad (6)$$

2.2 A ORGANIZAÇÃO DO ALGORITMO ETHOWATCHER OS

O fluxo do processamento da informação inicia com cadastro do arquivo de vídeo a ser analisado e configuração da calibração do experimento pelo usuário. Com as informações necessárias o ciclo de conversão da imagem para o formato da biblioteca OpenCV e envio para o módulo de processamento é executado.

O módulo de processamento do EtowatcherOS possui uma série de dependências para gerar dados morfológicos e cinemáticos e relatórios. Nos arquivos, ou módulo, principais (“moduloprocessamento.h” e “moduloprocessamento.cpp”) estão definidas as classes utilizadas no tratamento de imagens para extração desses descritores e o cálculo deles. Para compreensão dos parâmetros de entrada e saída do módulo principal foram analisados os módulos antecessores e sucessores responsáveis por fornecer as imagens, adquirir as informações de calibração, acionar as funções de processamento e gravar os dados da análise (estes são os módulos “telacadastrofilme”, “telaEtografiaProce” e “moduloCaptador”¹). Todas as classes

¹ O processo de leitura do vídeo é executado pelo “moduloCaptador”, a determinação das variáveis do experimento é inserida na “telacadastrofilme” e o ciclo de análise das imagens é controlado pelo módulo “telaEtografiaProce”.

foram estudadas buscando identificar possíveis melhorias seguindo as boas práticas de programação discutidas na subseção 2.3 (CROZARA, 2017; MARTIN, 2009).

No módulo de ciclo de análise das imagens (“telaEtografiaProce”) há uma função de conversão (“processamentoDeVideoTodo”) da imagem para o tipo “cv::Mat”², que será utilizado ao longo do processamento. O controle da conversão dos dados e processamento é feito por função auxiliar “processamentoMorfologico”. Durante essa etapa, a imagem passa pelo seguinte processo de tratamento:

- a) Conversão de cor para escala de cinza;
- b) Conversão por limiarização para imagem binária;
- c) Os objetos são erodidos (filtro morfológico de erosão);

Em seguida a imagem é recortada para conter somente a área definida como câmera virtual e inicia o seguinte processo para encontrar o contorno do animal do experimento (OI):

- a) Encontra todos os contornos na imagem;
- b) Verifica se algum contorno foi encontrado;
- c) Classifica qual o contorno com maior área;
- d) Calcula a área desse contorno.

Uma vez obtida as informações do contorno do OI os descritores morfológicos são extraídos e os cinemáticos são calculados. Ao final todas as informações relevantes são enviadas ao módulo de gravação de dados e se inicia o processamento da imagem seguinte, caso não seja o final do experimento.

2.3 DESCRITORES CINEMÁTICOS E MORFOLÓGICOS

Os descritores são calculados pela análise de cada frame tratado para extração desses. Os descritores morfológicos são dependentes das informações no frame analisado e a relação de pixel para comprimento real da imagem. Os descritores cinemáticos são calculados pela diferença dos descritores morfológicos em frames consecutivos. Os descritores morfológicos utilizados no EtowatcherOS são:

- Área do objeto de interesse;
- Centro de massa do objeto de interesse
- Ângulo do objeto de interesse;

² Classe da biblioteca OpenCV para matrizes, geralmente contendo informações de imagens.

- Altura do objeto de interesse;
- Largura do objeto de interesse.

E os descritores cinemáticos utilizados EtowatcherOS são:

- Variação área do objeto de interesse;
- Variação centro de massa do objeto de interesse
- Variação ângulo do objeto de interesse;
- Variação altura do objeto de interesse;
- Variação largura do objeto de interesse;
- Variação da distância do objeto de interesse.

Durante a detecção da área do objeto de interesse, que ocorre antes de iniciar a análise morfológica, é verificado se existe um contorno detectado na imagem. Isso é verificado pelo cálculo da área do contorno identificado na área com a função “cv::contourArea()”. O cálculo do centro de massa do objeto de interesse é feito com a função cv::moments(), que usa as equações (5) e (6). O cálculo do ângulo do objeto de interesse usa a função “cv::minAreaRect()”, a qual retorna qual o menor retângulo que contém o OI. A altura e largura do objeto de interesse são os parâmetros height e width respectivamente resultante do menor retângulo que contém o OI. Os descritores cinemáticos são calculados pela subtração dos descritores morfológicos na análise presente pelo valor da análise anterior. Para cada tipo de grandeza sendo calculada é considerada operação equivalente a subtração . Ou seja, as variações do OI (de área, centro de massa, ângulo, altura) são computadas a partir de subtração do seu valor presente pelo seu valor anterior. A variação da distância do objeto de interesse (distância da variação do centro de massa pela equação de comprimento de uma hipotenusa) é computada pela equação (7).

Variação da Distancia

$$= \sqrt{(x_{atual} - x_{anterior})^2 + (y_{atual} - y_{anterior})^2} \quad (7)$$

2.4 FERRAMENTAS DE DESENVOLVIMENTO

São apresentadas nessa subseção as ferramentas de desenvolvimentos aplicáveis e utilizadas durante esse trabalho.

2.4.1 Qt

Framework de desenvolvimento para aplicação de código aberto. As vantagens da ferramenta são sua integração para compilação de aplicativos facilitada pela disponibilidade de biblioteca própria abrangente, com funcionalidades preparadas para construção de um código modular, ampla documentação com exemplos de aplicação e apoio recorrente dos desenvolvedores (THE QT COMPANY, 2019).

2.4.2 C++

Linguagem de programação difundida no desenvolvimento de software elegante e eficiente (MARTIN, 2009). O EthoWatcher OS, assim como a biblioteca de análise de imagens OpenCV foram criadas para essa linguagem.

2.4.3 OpenCV

Biblioteca de funções para visão computacional para aplicações acadêmicas e comerciais. A biblioteca tem módulos desenvolvidos para resolver problemas específicos de visão computacional de forma computacionalmente eficiente. A biblioteca OpenCV possui uma série de módulos, serão detalhados nessa subseção apenas os módulos utilizados, que são o “Core”, “Imgprocess” e o “Highgui” (OPENCV TEAM, 2019).

O módulo Core contém a base para se trabalhar com a biblioteca OpenCV. As funções e definições de interesse são as estruturas básicas de dados, funções de erro, funções matemáticas para matrizes (adição, subtração e multiplicação) e funções estatísticas (máximo, mínimo, média e normalização). As funções definidas no módulo Imgprocess executam transformações, conversões e filtragem de imagens. É o principal módulo utilizado, por fornecer funções otimizadas dos tratamentos e extração dos descritores das imagens. Para visualização das imagens, iteração de objetos na tela do usuário e acoplamento das estruturas de dados do OpenCV com o sistema operacional do usuário o módulo HighGUI fornece os métodos para trabalhar com interface gráfica do usuário.

2.5 BOAS PRÁTICAS EM CÓDIGOS DE PROGRAMAÇÃO

As boas práticas em programação são recomendadas dependendo da filosofia de código a ser seguida. Segundo Martin no livro “*Clean Code: A Handbook of Agile Software Craftsmanship*” boas regras práticas do desenvolvimento e documentação são (MARTIN, 2009):

- Funções, métodos e variáveis devem ser nomeados de forma autoexplicativa;
- Nomes devem ser preferencialmente pronunciáveis e procuráveis;
- Nomes devem ser o mais curto possível;
- Nome de classes e objetos de classes devem ser substantivos;
- Nome de métodos e funções devem ser verbos;
- Classes e métodos devem ter função única e clara;
- Use uma palavra diferente para cada conceito;
- Métodos devem fazer somente uma função;
- Objetos devem ter parâmetros relacionados a função da sua classe;
- Evitar o uso de comentários para melhorar a legibilidades;
- Métodos não devem modificar parâmetros de objetos fora de sua classe;

3 MATERIAIS E MÉTODOS

Foram utilizados os materiais (software EtowatcherOS e documentações) de Marcolan e Crozara (CROZARA, 2017; MARCOLAN, 2017). Foram utilizadas as bibliotecas Qt, C++ e OpenCV e o ambiente de desenvolvimento QT Creator. Os softwares foram atualizados para suas versões mais recentes e de interesse (a data da atualização foi o segundo semestre de 2019).

O estudo prévio do código fonte do EtowatcherOS detectou que ele está organizado em módulos, onde estão definidas as classes, os parâmetros e os métodos seguindo a necessidade funcional do programa (conforme explícito na Fundamentação teórica). O algoritmo foi desenvolvido utilizando a plataforma de desenvolvimento QT Creator 4.10.1, com a linguagem de programação C++, biblioteca de visão computacional OpenCV 4.1.1 e baseado no funcionamento da ferramenta ETHOWATCHER descrito em *ETHOWATCHER: validation of a tool for behavioral and video-tracking analysis in laboratory animals* (CRISPIM JUNIOR et al., 2012; CROZARA, 2017; MARCOLAN, 2017).

3.1 ASPECTOS UTILIZADOS NO ALGORITMO

O algoritmo foi desenvolvido em um arquivo de biblioteca para C++, contendo um *header*³ e um *body*⁴. Nesses arquivos estão declaradas as classes, objetos e métodos utilizados. Durante os testes do algoritmo foram desenvolvidas versões de um programa que executa os métodos na ordem para funcionamento.

Para manter a organização do código foram definidas diversas nomenclaturas seguindo as boas práticas para um código limpo e buscando um dicionário consistente para melhor entendimento interpretação do código por terceiros (MARTIN, 2009). As siglas que foram utilizadas nesse trabalho objetivaram trazer o mesmo significado, sem a necessidade da escrita por extenso ou já eram utilizadas na literatura. Essas são de acordo com a Tabela 1.

³ *Header* é o arquivo onde informações iniciais para compilação de um código são declaradas.

⁴ *Body* arquivo com os métodos e funções para as classes criadas.

Tabela 1 – As siglas utilizadas durante construção dos códigos.

Sigla	Descrição
ROI	Do inglês região objeto de interesse
OI	Objeto de interesse
AIO	área de interesse ao observador
AE	área do experimento (arena de experimento)
CM	Centro de massa

Fonte: Próprio autor (2019).

As classes criadas têm nome descritivo para sua função e se limitam a essa. Classes com mais de uma função foram divididas em classes diferentes ou foram criadas subclasses para tratar as funções divergentes da classe original. Mantendo a nomenclatura utilizada no código fonte do EthoWatcherOS onde possível e modificando onde necessário. A nomeação de classes, variáveis e métodos é de acordo com a Tabela 2.

Tabela 2 – Nomeação de classes e variáveis utilizadas durante construção dos códigos.

Conteúdo da variável	Regra de nomenclatura	Exemplo do código
Matriz contendo imagens	frame + “função da imagem ou conteúdo”	frameTratada
Pontos da classe cv::Points2f	ponto + “Informação contida no ponto”	pontoCM
Vetores contendo pontos de uma área da classe cv::Rect	Região + “objeto delimitado pelo retângulo”	regiaoExperimento

Fonte: Próprio autor (2019).

Na iteração entre imagens analisadas, os objetos com a informação do frame anterior foram nomeados da seguinte forma: “antiga” + “nome do objeto” e “nova” + “nome do objeto”. Por exemplo, a variável de altura foi criada da seguinte forma:

```
float novaVarAltura, novaVarLargura;
float antigaAltur, antigaLargur;
```

onde float é o tipo de variável (de ponto flutuante).

Os nomes de objeto foram atualizados para o formato “nome do objeto” + “_anterior” e “nome do objeto” + “presente”. Indicando conter informações da iteração anterior e atual da análise, respectivamente.

As variáveis de região e pontos da imagem referentes apenas a região do objeto de interesse era terminada em “ROI”. Para melhor distinguir as variáveis com referência do frame completo e as imagens com referência a um subconjunto de pontos da imagem, foi adotado o pós fixo “_ROI”.

3.1.1 Filtros morfológicos utilizados

Nesse trabalho foi utilizado o filtro de erosão *morphcross*, um elemento em forma de cruz. A função utilizada para gerar o elemento estruturante é a “getStructuringElement”. Durante seu uso é necessário especificar o tamanho do elemento estruturante⁵. A sua representação gráfica, se tamMorfológico possui 3, é conforme a Figura 1. Esse elemento estrutural é utilizado na função “cv::erode()” para executar o método na imagem de entrada.

Figura 1 - Elemento estruturante com tamanho 3.

0	0	0	250	0	0	0
0	0	0	250	0	0	0
0	0	0	250	0	0	0
250	250	250	250	250	250	250
0	0	0	250	0	0	0
0	0	0	250	0	0	0
0	0	0	250	0	0	0

Fonte: Próprio autor (2019).

⁵ A função deve ser chamada da seguinte maneira:

“getStructuringElement(cv::MORPH_CROSS,cv::Size(2*tamMorfologico+1, 2*tamMorfologico+1),cv::Point(tamMorfologico, tamMorfologico))”

Após passagem pelo filtro de erosão é esperado que pequenas partículas, representadas por região com poucos pixels, sejam totalmente eliminados da imagem e áreas conectadas indevidamente sejam separadas.

A próxima etapa é o filtro por dilatação. A quantidade de pixels resultante também depende do elemento estruturante. O elemento utilizado é o mesmo da etapa anterior, a erosão. A função que executa o método é a “cv::dilate()”. Após sua aplicação a área do objeto de interesse retorna para um valor próximo ao tamanho antes da erosão.

3.1.2 Descritores Cinemáticos e Morfológicos

Os descritores morfológicos e cinemáticos são os mesmos da subseção 2.3.2 (Descritores cinemáticos e morfológicos).

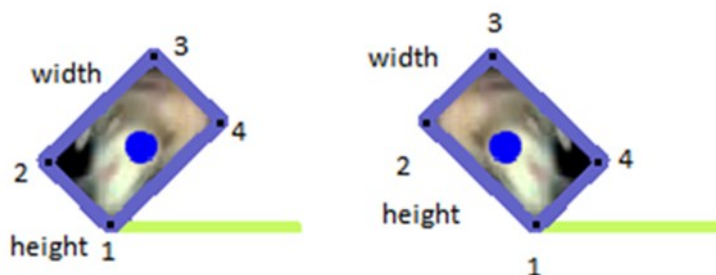
A área do animal é determinada como o momento m_{00} resultante do contorno detectado como objeto de interesse. Ela é um resultado direto das seguintes funções:

- “cv::findContours”, que detecta os contornos em uma imagem binária, classifica em uma relação de hierarquia contém e está contido.
- “cv::moments”, que retorna os momentos até terceiro grau de um contorno.

O centro de massa é obtido pelas relação dos resultados da função “cv::moments” (que executa as equações 5 e 6).

O ângulo do objeto de interesse tem uma faixa de valores entre 0° a -180° . Para seu cálculo é utilizada a função “cv::minAreaRect”. Esta encontra o menor retângulo rotacionado que contém todos os pontos do objeto de interesse. O retângulo encontrado é composto por quatro pontos na ordem horária: esquerda baixo, acima esquerda, acima direita e abaixo direita. A altura é a distância entre o primeiro e segundo ponto e a largura a distância entre o segundo e terceiro ponto. Estes são conforme a Figura 2. O parâmetro ângulo do retângulo da Figura 2 é calculado como ângulo entre o primeiro e último ponto em relação ao eixo horizontal. Sua faixa de valores é de 0° a -90° . Para obter o valor referenciado ao lado mais comprido do retângulo e o eixo vertical é verificado se o *width* é maior do que *height*, ou seja, o ângulo já tem o referencial correto, ou *height* é maior do que *width* e se subtrai 90° ao parâmetro ângulo resultante.

Figura 2 – Retorno de “cv::minAreaRect” e seus parâmetros



Fonte: Próprio autor (2019).

A altura do objeto de interesse é o comprimento do menor lado no menor retângulo que contém o objeto de interesse. Para o seu cálculo é utilizada a mesma função do descritor anterior. Os valores de altura e largura resultantes da função são comparados e o menor é definido como altura. Já a largura do objeto de interesse é o comprimento do maior lado no menor retângulo que contém o objeto de interesse. Para o seu cálculo é utilizada a mesma função do descritor de ângulo em relação ao eixo vertical do menor retângulo que contém o objeto de interesse. Os valores de altura e largura resultantes da função são comparados e o maior é definido como largura.

O descritor composto criado é calculado com informações dos descritores morfológicos do OI e um ponto de interesse na região do experimento. O descritor de experimento usado é a distância do objeto de interesse até a região de interesse ao observador. O descritor de distância do animal do experimento a poderá ser usado em pesquisas para indicar a proximidade de um animal a área de interesse experimental, como bebedouro, comedouro e entretenimento ou áreas de aversão como local gelado, escuro ou sujeito a descarga elétricas.

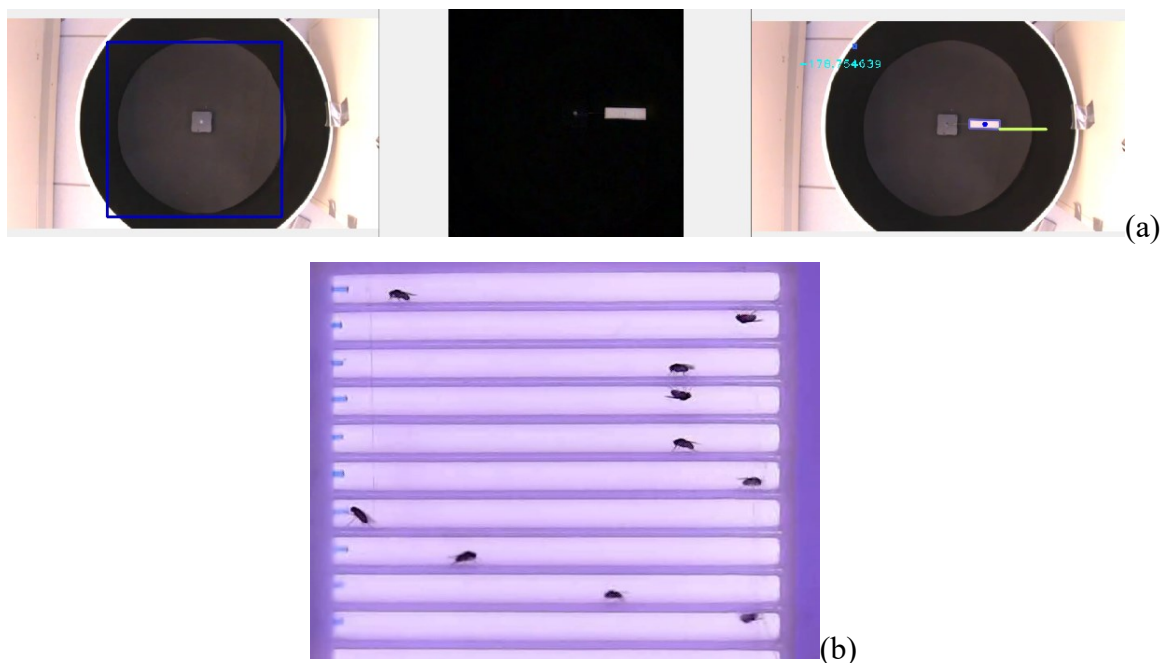
3.1.3 Validação dos descritores implementados

Para realizar os testes comparativos os algoritmos foram modificados para obter resultados sem as modificações propostas e comparar com o resultado do algoritmo completo. Os resultados que serão apresentados nos resultados são extraídos pelo algoritmo desenvolvido

completo e uma versão modificada onde a função “tratarFrame” salta a etapa da dilatação. Os resultados do descritor ângulo foram calculados realizando o cálculo inverso do novo descritor.

Para análise do algoritmo foram realizadas análises de dois vídeos do Laboratório de Bioengenharia da UFSC. O primeiro é o vídeo⁶ é de um retângulo posicionado a 10cm do centro de um relógio no ponteiro de segundos em um fundo preto simples, onde são testados os limites de tamanho para elemento estruturante e identificado os casos onde o algoritmo proposto consegue extrair informações e o antigo gerava um caso de erro. O segundo vídeo⁷ é um experimento com *Drosophilas Megalonaster*, representando um caso real com características naturais de experimento e suas problemáticas. São expostos aqui recortes do primeiro e segundo vídeo na Figura 3.

Figura 3 – Em (a) há três recortes do primeiro vídeo, contendo o plano de fundo com ROI desenhado, o resultado da operação subtração e o retângulo no centro de um relógio com descritores morfológicos desenhados(imagem de 43 segundos do vídeo). Em (b) um recorte do segundo vídeo, contendo distribuição espacial de moscas *Drosophilas Megalonaster* (imagem de 8 minutos e 45 segundos do vídeo).



Fonte: Próprio autor.

⁶ Título “ponteiroSegundos10cvVetical.avi” do banco de dados do Laboratório de Bioengenharia da UFSC.

⁷ Título “V1femeas2108201911h.avi” do banco de dados do Laboratório de Bioengenharia da UFSC.

Para análise de comportamento animal e sua distribuição espacial foi utilizado a filmagem de um estudo com *Drosophila Megalonastris*. Ele contém 7 raia de experimento (conforme Figura 3b), onde os 4 animais no topo são o grupo de controle e os 3 animais no final foram tratadas com Escitalopram, fármaco antidepressivo, o vídeo foi filmado em Agosto de 2019. Nos resultados foi utilizado as informações extraídas da quinta raia para verificar as diferenças das mudanças em tratamento da imagem, novo descritor distância do objeto de interesse a região de interesse ao observador e descritor de ângulo.

4 RESULTADOS

4.1 COMPARAÇÃO DO ALGORITMO COM ERODE E DILATAÇÃO E SOMENTE ERODE

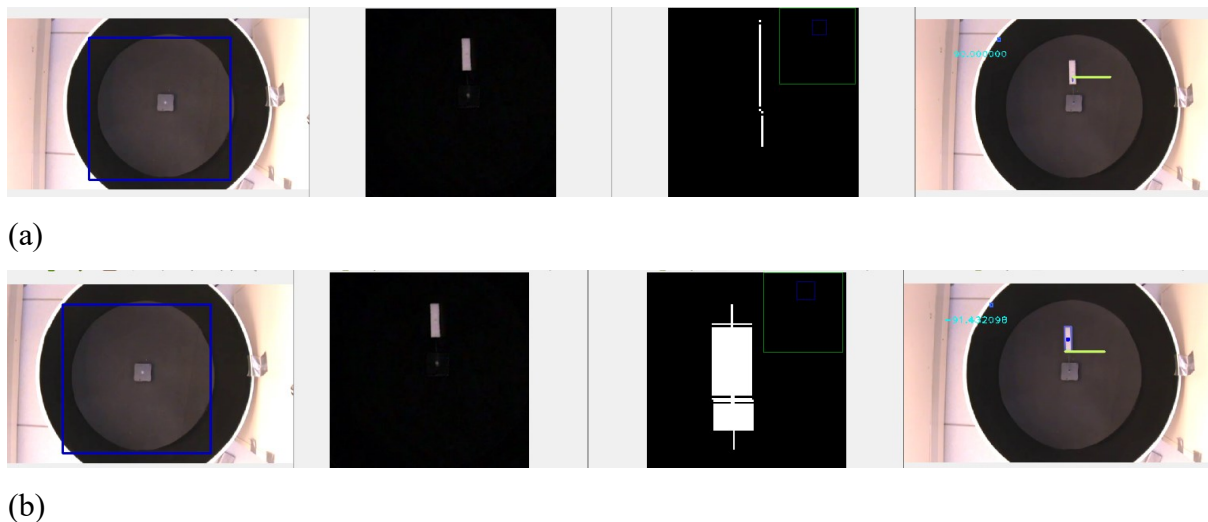
Na tabela 3 são apresentados os dados extraídos do primeiro vídeo utilizando uma calibração com valor de limiarização constante (de valor igual à 60 em escala de 8bits ou 255) e variando o tamanho do elemento estruturante para os filtros morfológicos. Observa-se que na tabela há um erro (área com valor zero), indicando que a aplicação da erosão diminuiu o OI a um valor de pixels inferior ao necessário para encontrar o contorno. Esse erro é tratado no EthoWatcher por verificar se existe um OI na ROI analisada e, caso negativo, copiar os descritores do frame anterior para o novo frame. Utilizando a dilatação esse cenário de erro não ocorreu.

Tabela 3 – Dados extraídos do primeiro vídeo.

Análise	Operação morfológica	Área OI			
		Min.	Max.	Média	Desvio padrão
Elemento estruturante 2	Erosão e Dilatação	1457	1665	1563.63	56.84
	Erosão	1107.5	1327	1233.24	64.73
Elemento estruturante 2	Erosão e Dilatação	304	1536.5	1250.47	222.75
	Erosão	0	394	140.118	117.63

Fonte: Próprio autor (2019).

Figura 4 – Em (a) não há dilatação e o tamanho do elemento estruturante é 10– Roi, Resultado da subtração do plano de fundo, Zoom do resultado da erosão e frame original com caixa e indicação de angulo). Em (b) (com dilatação relógio 10 - Roi, Resultado da subtração do plano de fundo, Zoom do resultado da erosão seguida de dilatação e frame original com caixa e indicação de angulo)



Fonte: Próprio autor (2019).

Realizando a análise do segundo vídeo, com nível de limiarização constante (de valor igual à 20 em escala de 8bits ou 255) e variando o tamanho do elemento estruturante para funções morfológicas as estatísticas dos resultados estão na Tabela 4.

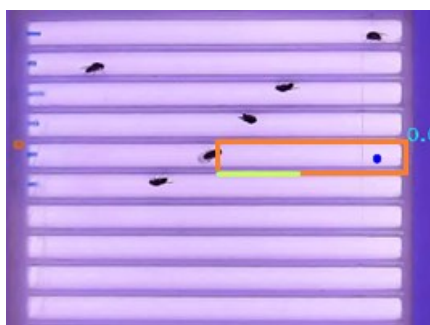
Tabela 4 – Dados extraídos do segundo vídeo.

Análise	Operação morfológica	Área OI			
		Min.	Max.	média	DP a
Elemento estruturante 0	Nenhuma	199	303	252.53	20.68
Elemento estruturante 3	Erosão e Dilatação	175.5	270.5	222.96	16.04
	Erosão	55.5	121.5	84.95	11.06
Elemento estruturante 4	Erosão e Dilatação	46	262.5	171.6	18.47
	Erosão	3.5	81	26.42	27.36

Fonte: Próprio autor (2019).

Quando analisado o efeito das funções morfológicas na área do objeto de interesse, no algoritmo com dilatação para o tamanho do elemento estruturante 3 e 4 existe uma maior variação entre o máximo e mínimo dos dados. Traçando os descritores nas imagens é identificado que ocorre um erro na identificação do OI, mostrado na Figura 5 . A falha do filtro de erosão em desconectar elementos indesejados da imagem, causa detecção errônea do objeto de interesse e extrai descritores com valores referentes ao erro. Consequentemente, os resultados estatísticos com o elemento estruturante tamanho 3 e 4 serão influenciados por essas ocorrências. Comparando os dados gerados com elemento estruturante 4 e aplicando somente a erosão, a área tem valores dentro de uma faixa limitada. Nos casos sem dilatação o valor do elemento estruturante é inversamente proporcional a média da área, pois o filtro de erosão subtrai pixels e não há compensação para isso.

Figura 5 - Erro de identificação da ROI com elemento estruturante 3

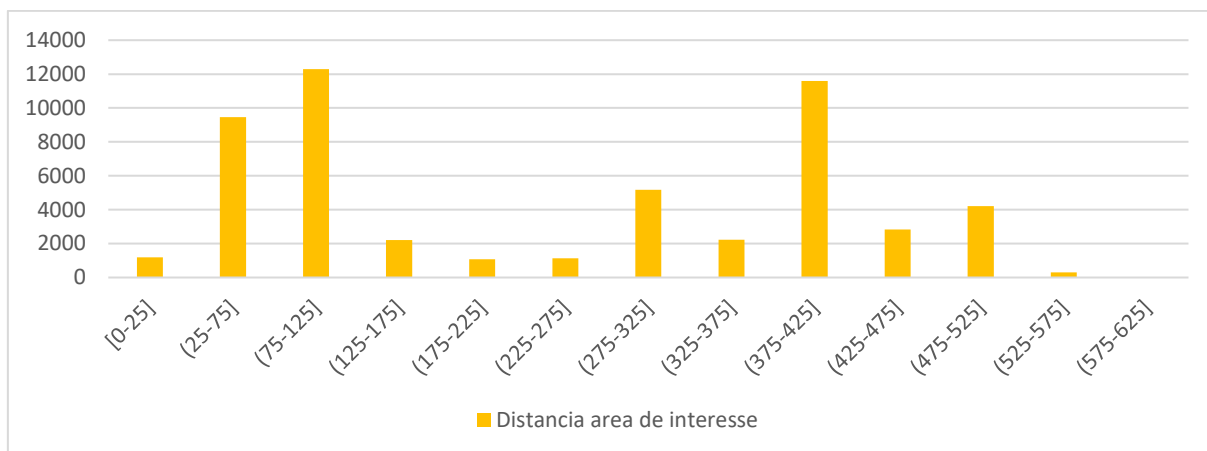


Fonte: Próprio autor (2019).

4.2 DESCRITOR DISTÂNCIA DO OBJETO DE INTERESSE A REGIÃO DE INTERESSE AO OBSERVADOR

O descritor distância do objeto de interesse a região de interesse ao observador representa a distancia do animal a um ponto de interesse no experimento. A Figura 5 é o resultado da análise do segundo video utilizando nível de limiarização constante (de valor igual à 60 em escala de 8bits ou 255) e tamanho do elemento estruturante 3.

Figura 6 – Histograma da distância entre mosca e área de interesse experimento.



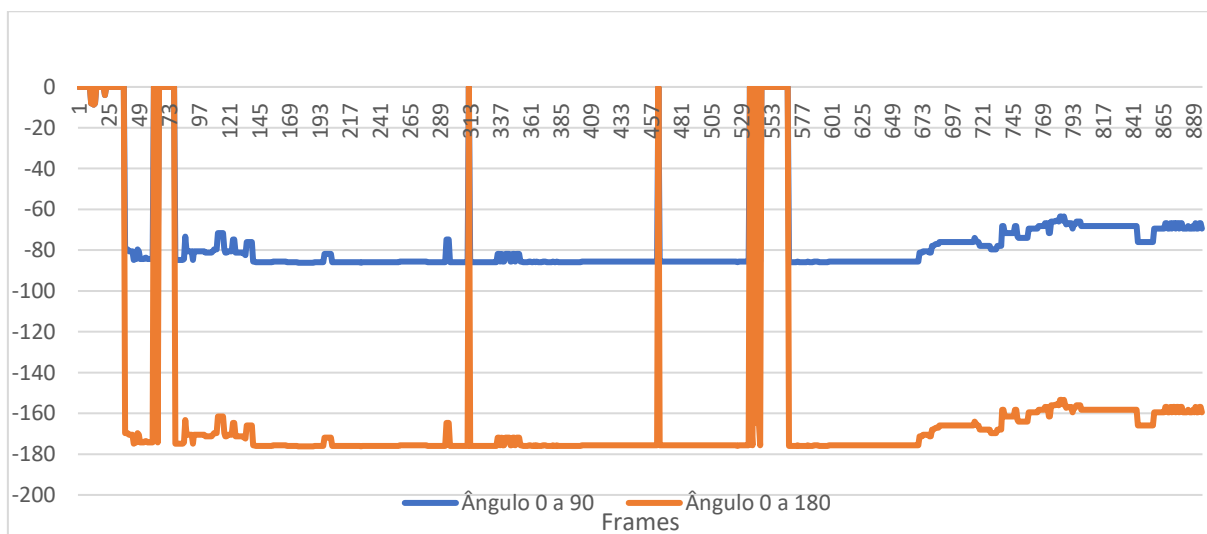
Fonte: Próprio autor (2019).

Os dados foram extraídos sem valores aparentes de erro. Em casos onde o OI não é encontrado, como indicado pela área 0 na Tabela 4 o descritor também gera erro. Esse erro é evitado pelo EthoWatcher OS na verificação se o OI foi encontrado.

4.3 DESCRITOR DE ANGULO

Os resultados do descritor geram dados com maior faixa de valores como na Figura 6. Os resultados são consistentes com o esperado e não apresentaram cenários de erro diferenciados.

Figura 7 – Comparativo de valores para descritor de ângulo.



Fonte: Próprio autor (2019).

5 DISCUSSÃO

O projeto evoluiu para uma estrutura mais coerente com as intenções do laboratório e ferramentas atualizadas disponíveis. As técnicas propostas podem ser um avanço nas capacidades de extração de dados para o EthoWatcher OS.

Os métodos para extração de descritores morfológicos modificados apresentaram menos cenários de erro do que em trabalhos anteriores. Os passos adicionados foram considerados influenciarem positivamente na análise, diminuindo o erro do sistema.

O cálculo da área do OI é influenciado pelas operações de erosão e dilatação. Como o método anterior apenas retirava uma área do objeto na operação de erosão, o resultado sempre é menor do que o objeto detectado. Com aplicação da correção por dilatação, o método proposto busca compensar o cálculo da área, assim como mostrou gerar menos cenários de erro para OI representados por poucos pixels.

No cálculo de área a simplificação da entrada como um contorno de imagem binária faz com que seu cálculo seja semelhante ao utilizado por (CROZARA, 2017), onde é calculada a área pela contagem de bits nulos na matriz contendo o objeto de interesse. Porém, foi identificado na análise de ratos, devido ao seu olho preto, o resultado é impactado por não contar esses pixels e o método de momentos compensa esse erro por não considerar as informações no interior do OI.

O cálculo de ângulo é dependente da função de menor retângulo que contém o OI, então não existe uma distinção entre frente e costas do animal. Apenas a angulação da sua maior dimensão com o eixo horizontal.

Cenários em que o OI não é encontrado ou existem artefatos na imagem que geram detecção errônea do OI ainda ocorrem.

Analisando visualmente o histograma gerado e mostrado na Figura 6 pode-se notar que a curva de distribuição espacial da distância entre a mosca e o comedor de sua raia é formada pelo somatório de duas ondas normais com a média deslocada. Uma hipótese é que a mosca tenha, então, duas regiões de preferências. Essa informação pode ser muito útil na hora de demarcar distâncias de interesse para segmentar o relatório categórico produzido pelo EthoWatcher OS. Isso pode permitir explorar quais comportamentos categóricos foram realizados nessas distâncias, áreas. E averiguar estatisticamente se existe diferença entre a

duração dos comportamentos das moscas nelas. Outra coisa a ser explorada é se existe diferença na distribuição da mosca tratada em relação a mosca controle. Para ser possível essas e outras análises seria necessário produzir mais dados de moscas em diferentes condições. Seria de interesse etológico construir outros descritores composto como o de ângulo do animal em referência a uma linha reta traçada pelo usuário. Porque, a direção em que o animal exprime o comportamento pode ser um discriminador em duas sequências comportamentais parecidas. Por exemplo, mexer as patas em direção ao comedouro pode significar algum comportamento de ingestão de alimento enquanto mexer as patas em direção a parede pode significar um comportamento de fuga. Como o descritor composto de posição em relação a um ponto ele também poderia ser útil para segmentar o relatório de comportamento animal.

6 CONCLUSÃO

O algoritmo criado tem as funcionalidades propostas para adicionar as capacidades do Ethowatcher OS. As técnicas de análise de imagem utilizadas mostraram ser as melhores para a aquisição da informação desejada, sem utilizar técnicas que requerem grande poder computacional, e podem ser evoluídas com mais contribuições.

Descritores compostos podem ser úteis para entender mais nuances do comportamento animal. Um uso seria o de segmentação do relatório de comportamento por distâncias de interesse. Sendo assim esse tipo de descritor pode ser usado para refinar a análise de comportamento animal. Mais descritores compostos devem ser construídos como por exemplo o de ângulo em relação a uma linha traçada pelo o usuário. O algoritmo deve ser passado em mais vídeos que contenham moscas para ser possível realizar análises estatísticas. Isso pode permitir refinar a exploração de conhecimento dos dados etográficos.

A biblioteca possui melhorias na nomenclatura de algumas variáveis e na organização do código. Mas, seria interessante desenvolver outras práticas de desenvolvimento ágil como por exemplo o de testes automáticos. Para que garantisse que as modificações ao longo do tempo feitas nelas ainda fossem válidas, uma vez que isso tenha sido averiguado por testes experimentais. Ainda faltaria adicionar elementos na interface gráfica do EthoWatcher OS para suportar as adições de funcionalidades que a biblioteca possui. Como por exemplo, uma área para que o usuário informasse aonde fica o ponto de referência para área de interesse ao observador.

Outra possibilidade de evolução é a diferenciação na calibração para animais (mamíferos, aves, etc.) e insetos (moscas, mosquitos, etc.). Insetos tem o corpo rígido e podem ser utilizadas técnicas onde é analisado o formato do inseto e usado para identificar a posição, facilitando a detecção de sua posição e ângulo. Para animais, devido a variação no seu formato, orientação e comportamentos, é necessária uma técnica mais avançada que utilizam de probabilidade estatísticas para determinar pareamento das informações de calibração com o animal. Essas técnicas envolvem a calibração de um algoritmo com imagens de todos os ângulos e comportamentos do animal para se realizar uma comparação com o vídeo sendo analisado e corretamente extrair os descritores de interesse. Adicionalmente, as técnicas de

comparação com modelos exigem maior poder computacional e afetariam o acesso de pesquisadores ao uso do software.

Como conclusão, os objetivos propostos foram atingidos, identificando as forças e fraquezas dos métodos utilizados, destacando cenários de perdas de dados e são propostos novos trabalhos para continuidade da pesquisa.

REFERÊNCIAS

ANDERSON, D. J.; PERONA, P. Toward a science of computational ethology. **Neuron**, 2014.

BLUMSTEIN, D. T.; DANIEL, J. C.; EVANS, C. S. **JWatcher**. Disponível em: <<http://www.jwatcher.ucla.edu/>>. Acesso em: 12 dez. 2019.

CRISPIM JUNIOR, C. F. **Uso de descritores morfológicos e cinemáticos na identificação automática de comportamentos de animais de laboratório**. [s.l.] UFSC, 2011.

CRISPIM JUNIOR, C. F. et al. ETHOWATCHER: Validation of a tool for behavioral and video-tracking analysis in laboratory animals. **Computers in Biology and Medicine**, 2012.

CROZARA, M. G. N. **Um sistema de código aberto para registro e análise de dados comportamentais categóricos, morfológicos e cinemáticos em animais de laboratório**. [s.l.] UFSC, 2017.

DOMINGUES, K. **Influência do dimorfismo sexual na resposta ao tratamento com Fluoxetina em ratos no nado forçado repetido: contribuição do metabolismo da serotonina**. [s.l.] UFSC, 2015.

ECKERT, F. B. **Criação e padronização de métodos para análises etológicas de Drosophila melanogaster**. [s.l.] UFSC, 2014.

MARCHESINI, G. et al. **The use of reliability measurements in the Forced Swim Test (FST) in rats: probing the effects of experience on raters performance**. The use of reliability measurements in the Forced Swim Test (FST) in rats: probing the effects of experience on raters performance. **Anais...**2019

MARCOLAN, J. A. **Ferramenta de código aberto para análise de comportamento e aquisição de vídeo em tempo real usando técnicas de visão**. [s.l.] UFSC, 2017.

MARTIN, R. C. **Clean Code: A Handbook of Agile Software Craftsmanship**. [s.l.] Person, 2009.

NOLDUS. **EthoVision**. Disponível em: <<https://www.noldus.com/animal-behavior-research/products/ethovision-xt>>. Acesso em: 12 dez. 2019.

OPENCV TEAM. **opencv.org**. Disponível em: <OpenCV team>. Acesso em:

10 dez. 2019.

QUEIROZ, J. E. R.; GOMES, H. M. Introdução ao Processamento Digital de Imagens. **RITA**, v. 8, n. 1, 2001.

THE QT COMPANY. **QT Creator**. Disponível em: <<https://www.qt.io/>>. Acesso em: 10 dez. 2019.