UNIVERSIDADE DE SÃO PAULO
INSTITUTO DE FÍSICA DE SÃO CARLOS

ISABELA MARIA DE OLIVEIRA

Desenvolvimento de ferramenta computacional para obtenção automática de deslocamento e anotação de outros parâmetros em testes comportamentais do tipo campo aberto

São Carlos
2018

ISABELA MARIA DE OLIVEIRA

Desenvolvimento de ferramenta computacional para obtenção automática de deslocamento e anotação de outros parâmetros em testes comportamentais do tipo campo aberto

Dissertação apresentada ao Programa de Pós-Graduação em Física do Instituto de Física de São Carlos da Universidade de São Paulo, para obtenção do título de Mestra em Ciências.

Área de Concentração: Física Aplicada
Opção: Física Biomolecular
Orientador: Prof. Dr. Fernando Fernandes Paiva

Versão Corrigida

(versão original disponível na Unidade que aloja o Programa)

São Carlos
2018

AUTORIZO A REPRODUÇÃO E DIVULGAÇÃO TOTAL OU PARCIAL DESTE TRABALHO, POR QUALQUER MEIO CONVENCIONAL OU ELETRÔNICO PARA FINS DE ESTUDO E PESQUISA, DESDE QUE CITADA A FONTE.

Oliveira, Isabela Maria de
Desenvolvimento de ferramenta computacional para
obtenção automática de deslocamento e anotação de outros
parâmetros em testes comportamentais do tipo campo aberto
/ Isabela Maria de Oliveira; orientador Fernando
Fernandes Paiva - versão corrigida -- São Carlos, 2018.
68 p.

Dissertação (Mestrado - Programa de Pós-Graduação em
Física Aplicada Biomolecular) -- Instituto de Física de São
Carlos, Universidade de São Paulo, 2018.

1. Teste campo aberto. 2. Visão computacional. 3.
Comportamento animal. I. Paiva, Fernando Fernandes,
orient. II. Título.

À minha mãe, cujo empurrão inicial
culminou neste ponto

AGRADECIMENTOS

Inicialmente à minha família, que ao longo da vida proveram todas as condições que possibilitaram este passo.

Ao Prof. Dr. Fernando, pela definição do projeto, busca de colaborações e compreensão quanto às condições atípicas de realização do trabalho.

À Comissão de Pós-Graduação do Instituto de Física de São Carlos pelo atendimento ao pedido de extensão do prazo para depósito.

Ao CCBS – Centro de Ciências Biológicas e da Saúde do Departamento de Fisioterapia da UFSCar – Universidade Federal de São Carlos pela colaboração e fornecimento dos vídeos e análises sem os quais este trabalho não seria possível, em especial à Prof^a Dr^o Anna Carolyn Lepesteur Gianlourengo, M^a Evelyn Maria Guilherme e ao Prof. Dr. Thiago Luiz de Russo.

Aos colegas de trabalho na Diretoria de Ensino Região de São Carlos pela transigência quanto às saídas necessárias para disciplinas e outras necessidades.

Aos amigos, sem os quais nada do trabalho seria possível: Ruan pelas incontáveis operações de resgate, detecções de problemas de lógica nos códigos e resoluções aos bugs de instalação. Natália, que mesmo sem entender dois terços da metodologia foi capaz de me socorrer nos momentos de desespero e fazer enxergar um caminho. Tamiris, sempre cuidando do cenário para que focar no necessário fosse possível. Prê, pelas conversas com batata frita nos dias difíceis (ou nem tão difíceis assim, afinal é batata frita). Hipo, pelas conversas convertidas em soluções práticas do início do trabalho. E ainda aqueles que conheci na ong Teto Brasil, que além de dividir os aprendizados pós imersões ainda mostram que a vida pode sim ser mais que trabalhar pra pagar boletos e viajar.

RESUMO

OLIVEIRA, I. M. **Desenvolvimento de ferramenta computacional para obtenção automática de deslocamento e anotação de outros parâmetros em testes comportamentais do tipo campo aberto**. 2018. 68 p. Dissertação (Mestrado em Ciências) - Instituto de Física de São Carlos, Universidade de São Paulo, São Carlos, 2018.

Testes comportamentais do tipo Campo Aberto figuram dentre os mais básicos e consolidados para estudos de comportamento animal, sendo, portanto, amplamente utilizados em pesquisas de Neurociência e Farmacologia. Através dele se pode avaliar os efeitos da imposição de diferentes condições no comportamento de animais, bem como comparar ou caracterizar perfis típicos de comportamento entre linhagens de uma mesma espécie. Consistindo de soltar animal em uma caixa e observar, dentre outros parâmetros, seu deslocamento e preferência em permanecer nas áreas próximas às paredes ou nas centrais, a extração desses dados pela forma tradicional acabar por ser limitante à obtenção plena de resultados. Isso posto, este trabalho visa oferecer aos pesquisadores um meio para obter os dados de maneira automatizada, sem o desgaste de ficar assistindo o vídeo do teste para anotar o deslocamento. Escrita em Python, utilizando técnicas consolidadas de Visão Computacional e operações de compreensão acessível, esta ferramenta além de ser *Open Source* pode ser ajustada à extração de outros parâmetros e à aplicação em outras modalidades do teste, além de oferecer a medida do deslocamento de forma mais precisa.

Palavras-chave: Teste campo aberto. Visão computacional. Comportamento animal.

ABSTRACT

OLIVEIRA, I. M. **Development of software to automatically obtain displacement and annotation of other parameters in open field behavioral tests**. 2018. 68 p. Dissertação (Mestrado em Ciências) - Instituto de Física de São Carlos, Universidade de São Paulo, São Carlos, 2018.

Open Field Test is a basic and consolidated test used in Animal Behavior studies, especially Neuroscience and Pharmacology. Through it, it's possible to evaluate the effects in behavior of imposing different conditions, as well to compare or characterize typical behavior profiles between lineages of the same species. Consisting of releasing an animal in a box and observing, among other parameters, its displacement and preference in remaining in the areas near the walls or in the central ones, the extraction of these data in the traditional way ends up being limiting to obtaining full results. That said, this essay aims to offer researchers a way to get the data in an automated way, without need watching the video of the test to note the displacement. Written in Python, using consolidated techniques of Computational Vision and operations of accessible comprehension, this tool besides being Open Source can be adjusted to the extraction of other parameters and application in other modalities of the test, besides offer a more accurate measure of displacement.

Keywords: Open field test. Computer vision. Animal behavior.

LISTA DE FIGURAS

Figura 1 – Esquema de Cores RGB.....	21
Figura 2 – Vizinhança 3 x 3 ao redor de um ponto (x, y) em uma imagem f	24
Figura 3 – Uma imagem com predominância de tons claros e outra com pouco contraste antes e após Equalização do Histograma.	25
Figura 4 – Correção de Ruído Sal e Pimenta em imagem de raios X de um circuito:	26
Figura 5 – Perfil de intensidades de uma região em uma imagem com os três tipos de bordas descritos na literatura.	27
Figura 6 – Esquema geral dos dados extraídos pelo algoritmo.....	35
Figura 7 – Plotagem do ganho em similaridade conforme novos frames são adicionados ao conjunto a partir do qual se calcula a média para obter um modelo de como seria a imagem sem o animal.....	37
Figura 8 – Esquema simplificado dos métodos utilizados na determinação da localização do animal num dado instante T	38
Figura 9 – Quadro extraído de vídeo de Teste realizado em gerbilo – “alto contraste”	43
Figura 10 – Quadro extraído de vídeo de Teste realizado em camundongo – “baixo contraste”	43
Figura 11 – Aspecto Visual da adição de mais frames para compor o modelo, bem como da implicação do contraste entre animal e caixa.	45
Figura 12 – Modelo Final do Fundo do Vídeo obtido ao fim da etapa respectivamente para vídeos com gerbilo e camundongos.	46
Figura 13 – Histogramas calculados a partir dos quadros utilizados com exemplo. A linha vermelha indica o limiar obtido.....	47
Figura 14 – Resultados da Subtração em Módulo, sua Suavização e posterior Limiarização. O número indicado corresponde ao valor de corte obtido a partir do histograma.	48
Figura 15 – Resultados das etapas de Limiarização, Seleção de Objeto e Contorno.	49
Figura 16 – Ilustração da saída final da etapa de detecção: desenho de ponto no centro geométrico calculado.....	49

Figura 17 – Exemplo de Gráfico de Dispersão das Posições (x,y) durante a realização do teste com o camundongo 2.	50
Figura 18 – Gráfico de Dispersão de (x,y) para o camundongo 3:	51
Figura 19 – Gráfico de Dispersão da Posição para o gerbilo: Há uma distribuição mais homogênea das posições, dado o perfil de deslocamento diferenciado da espécie. Os eixos têm intervalos maiores por se tratar de imagem com maior número de pixels.	51
Figura 20 – Variação do Deslocamento a cada segundo durante a realização do teste.	52
Figura 21 – Outro exemplo de perfil de Deslocamento extraído: não há momentos de predominância de pouca motilidade, o que em conjunto com outros parâmetros pode ser um indicador de estresse.	53
Figura 22 – Perfil de Deslocamento do Gerbilo – por tratar-se de uma imagem maior, as variações de deslocamento em pixels se dão em outra escala. ...	53
Figura 23 – Gráfico da Variação de Deslocamento (em azul) e Informação sobre Quadrantes (em vermelho) para o Camundongo 5.	54
Figura 24 – Sistema de Referência e Nomenclatura de Quadrantes para o XploRat e a Ferramenta Proposta.	55
Figura 25 – Gráfico comparativo com melhor correspondência entre a porcentagem de tempo de permanência – Vídeo do teste realizado com o camundongo 8	56
Figura 26 – Comparação entre as porcentagens de permanência em cada quadrante extraídas pela XploRat e pela Ferramenta proposta para o camundongo 3.	56
Figura 27 – Gráfico de Ocorrências de Entrada para o mesmo Vídeo do Gráfico anterior: observar que quase não há correspondências (camundongo 8).	57
Figura 28 – Melhor resultado de comparação para as ocorrências de entrada em cada quadrante – camundongo 2.	57
Figura 29 - Comparação entre sistemas de referência: da Xplorat, o utilizado na Ferramenta e o definido em função das regiões periférica (P), intermediária (I) e central (C).	58
Figura 30 – Gráfico simplificado da ocorrência e tempo de permanência nas macrorregiões criadas.	59
Figura 31- Dados obtidos a partir do vídeo da realização do teste com o camundongo 4.	59

LISTA DE TABELAS

Tabela 1 – Deslocamento Total obtido a partir de cada um dos vídeos.....	54
Tabela 2 – Número de ocorrências de Entrada em cada Quadrante – comparação entre XploRat e a Ferramenta Desenvolvida	67
Tabela 3 – Porcentagem do Tempo total do teste que o animal ficou em cada quadrante	68

SUMÁRIO

1	INTRODUÇÃO.....	17
1.1	Noções de Visão Computacional.....	19
1.1.1	Imagem digital monocromática e colorida	19
1.1.2	Tipos de Operações com Imagens	22
1.1.3	Operações no Domínio Espacial	23
1.1.4	Operações no Domínio da Frequência	25
1.1.5	Suavização	26
1.1.6	Aguçamento e Detecção de Bordas	27
1.1.7	Segmentação de Imagens.....	28
1.1.8	Similaridade entre Duas Imagens.....	28
1.2	Ferramentas Disponíveis no Mercado	29
1.2.1	X-PloRat	29
1.2.2	EthoVision	30
1.2.3	EthoWacher.....	30
1.2.4	Mouse Move	31
1.2.5	Kinoscope.....	31
2	DESENVOLVIMENTO.....	34
2.1	Python	31
2.1.1	NumPy	32
2.1.2	SciPy	33
2.1.3	Matplotlib	33
2.1.4	scikit-image.....	33
2.1.5	OpenCV.....	34
2.2	Breve Histórico e Visão Geral do Algoritmo.....	35
2.3	Modelo de Fundo da Imagem – Frame Médio.....	36
2.4	Rastreamento do Animal	38
2.5	Tratamento da Saída de Dados.....	40
3	RESULTADOS	43
3.1	Detecção do Animal e Saída de Dados	44
3.2	Validação da Ferramenta	55
4	CONCLUSÃO	61
	REFERENCIAS	63
	APÊNDICE	67

1 INTRODUÇÃO

Testes Comportamentais tornaram-se etapa obrigatória em diversos campos de pesquisa, com destaque para os do tipo Campo Aberto, pela simplicidade de construção do aparato e realização do experimento. Consistindo em se soltar um animal de pequeno porte numa caixa e observar algum comportamento – via de regra deslocamento – oferece meios para análise de efeitos comportamentais com relação à imposição de condições.

As primeiras ocorrências na literatura de Testes Comportamentais do tipo Campo Aberto se dão com (1) e (2), comparando os padrões de deslocamento quando se colocava alimento no centro da caixa e quando não no primeiro, e correlacionando respostas fisiológicas – micção e defecação a estresse emocional no segundo. Ao longo do tempo, graças a alterações nos protocolos e nos aparatos, contamos atualmente com mais de 100 tipos de testes comportamentais para roedores (3) e, suas aplicações, antes restritas à psicologia comportamental, hoje passam por diversas áreas, com destaque à neurociência e psicofarmacologia.

Em (4), temos na introdução uma discussão sobre em que efetivamente consiste o Teste Comportamental do Tipo Campo Aberto, na qual se conclui que apenas uma definição genérica é capaz de englobar todas as aplicações já à época: “*an enclosed open area where an animal is placed and some form of behavior, usually activity, is measured*”.

Há incontáveis variáveis quanto à realização do teste, passando pela caixa – formato, tamanho e material; duração e horário da tomada dos dados; condições de iluminação; presença de algum outro elemento como abrigos, odores, alimentos, objetos e ainda o grau de familiaridade do animal com o teste ou seu sexo, por exemplo. Essas variações são moduladas conforme o interesse do estudo, necessidades e dimensão do animal em estudo e disponibilidade de materiais do laboratório.

O grande objetivo dos testes comportamentais é detectar diferenças causadas por algum fator imposto ao animal, por exemplo administração de fármacos (4), condições de iluminação (5) ou ainda caracterizar o perfil de uma nova linhagem que apresente características recessivas e possivelmente aja de maneira diferente se comparada à média dos organismos de sua espécie. (6) Usualmente se anota o

deslocamento do animal e outros comportamentos bem caracterizados para a espécie em estudo, tal como autolimpeza ou elevação de tronco para camundongos.

A obtenção da distância percorrida se dá com pouca precisão: tradicionalmente se desenha linhas na caixa utilizada, e se anota quantas vezes o animal cruzou de uma região para outra, contabilizando a distância entre os centros de cada região como tendo sido percorrida pelo animal. Em (7) se cita um aparato em que a divisão da caixa em regiões é feita por feixes paralelos de luz infravermelha, registrando o número de vezes que dois adjacentes deles são interrompidos através de sensores. É bastante comum que se gravem vídeos da realização dos testes, para que posteriormente sejam assistidos e então os dados anotados.

Obter resultados consistentes não é simples no método tradicional: demanda tempo e especialização por parte do pesquisador encarregado. Ainda, a interferência do estado emocional do analista e a não interpretação como similar de um mesmo evento por pessoas distintas trazem risco à fidedignidade dos resultados com as ocorrências durante o experimento e a sua reprodutibilidade. Argumenta-se em (4) que por vezes não fica claro quais são os dados coletados, qual o grau de automação nessa etapa e como é feita a análise.

A demanda de tempo citada anteriormente merece ainda mais destaque: em outras colaborações do laboratório, teve-se contato com projetos de dezoito meses em que três deles foram dedicados apenas à análise dos testes. Há, portanto, uma demanda de instrumentos que auxiliem o pesquisador na obtenção e análise dos dados. Será feita uma descrição daqueles localizados durante este trabalho posteriormente, todavia antecipa-se que por vezes exigem compra de licenças. É válido destacar que esse tipo de teste não costuma ser o cerne dos projetos que o envolve, de modo que usualmente não é prioridade no emprego de esforços ou recursos.

Isto posto, apresenta-se uma Ferramenta para Detecção Automatizada de Dados a partir dos vídeos da realização dos experimentos. Escrita em Python, com interface simplificada, propõe-se a auxiliar o pesquisador que dependa de Testes Comportamentais do Tipo Campo Aberto a extrair os dados de maneira rápida e automatizada, sem licenças ou sigilo sobre seu funcionamento, e ainda com deslocamento dado de maneira mais precisa e independente do desenho de regiões. Outrossim, o código poderá ser complementado para determinar outras variáveis ou ampliado para outros testes por qualquer pessoa que tenha interesse.

1.1 Noções de Visão Computacional

Não há consenso sobre as fronteiras entre Visão Computacional, Processamento e a Análise de Imagens. Visão Computacional é definida em (8) como construção de descrições explícitas e significativas de objetos físicos a partir de imagens, enquanto o Processamento Digital de Imagens seriam as operações computacionais necessárias para a obtenção de descritores intermediários, e a Análise de Imagens a extração de informações mais refinadas a partir deles. Em (9), cita-se como meta da Visão Computacional emular a visão humana, incluindo o aprendizado, inferência e ação a partir de informação visuais – pertinente às definições de Inteligência Artificial. No mesmo livro temos uma alegoria da existência de um contínuo de métodos computacionais, com Visão Computacional num extremo e Processamento de Imagens no outro. Os métodos se dividiriam então em Baixo-Médio e Alto Nível – no nível mais baixo estariam as operações mais primitivas, com entradas e saídas correspondendo a imagens: redução de ruídos, otimização de contraste ou nitidez. No nível médio, teríamos saídas diferentes de imagens: segmentação, reconhecimento de objetos e descrição simplificada destes para que o computador possa processar de forma mais eficiente informações sobre ele e no mais alto haveria então busca de um sentido cognitivo a partir dos dados intermediários.

1.1.1 Imagem digital monocromática e colorida

Em (9) se define uma imagem monocromática como uma função bidimensional $f(x, y)$, em que x e y são coordenadas espaciais, e a amplitude de f em qualquer par de coordenadas (x, y) , corresponde à intensidade nesse ponto. Em imagens digitais, os valores $f(x, y)$ são quantidades escalares positivas limitadas e compõem uma matriz 2D de M linhas e N colunas, com x assumindo valores inteiros de 0 a $M - 1$ e y de 0 a $N - 1$. Imagens digitais são então compostas por um número finito de elementos com localização e valores específicos, denominados pixels, elementos pictóricos ou elementos de imagem.

Na formação da imagem, o valor da intensidade do pixel será proporcional à energia irradiada pela superfície a que ele corresponde (9): considerando o espectro visível, nas imagens em preto e branco, um maior nível de cinza estará relacionado

à uma maior luminosidade e, portanto, ao branco e regiões mais claras do cenário. Imagens digitais podem ainda ser criadas a partir de comprimentos de onda fora do espectro visível à humanos, sendo uma alternativa de representação das informações para interpretação.

Quando lidamos numa mesma imagem com mais de um intervalo de comprimentos de onda – por exemplo cores, deixamos de ter uma matriz bidimensional e passamos a uma tridimensional. Quando imprimimos algo colorido em papel, cada pixel da imagem é decomposto em 4 cores: ciano, magenta, amarelo e preto. Algo parecido acontece na representação digital da imagem, cada pixel corresponde à soma dos seus canais de cores – a dimensão acrescida, de modo que haverá uma superfície semelhante à função $f(x, y)$ descrita anteriormente em cada um deles.

Há diferentes modelos, sistemas ou espaço de cores cuja utilização se dá conforme a aplicação (9), notadamente o CMYK (*cyan, magenta, yellow, black*) para impressões e o RGB (*red, green, blue*) em monitores de vídeo. O modelo RGB se baseia em um sistema de coordenadas cartesianas, em que vermelho, verde, azul, ciano, magenta e amarelo ocupam seis dos oito vértices. Os outros dois são ocupados por preto na origem do sistema, e branco – quando todas as componentes têm suas intensidades máximas. Forma-se então um contínuo tridimensional que contém todas as nuances de cores possíveis, com destaque à diagonal entre preto e branco conter todos os níveis de cinza - ver a Figura 1 extraída de. (10)

Figura 1 – Esquema de Cores RGB

Fonte: BAYKAN (10)

Um dos efeitos de se trabalhar com imagens coloridas em lugar das monocromáticas é os pixels deixarem de ser grandezas escalares e tornarem-se vetores: seja c um vetor arbitrário que se estende da origem a um ponto qualquer no espaço de cores RGB (9), os componentes de c serão os componentes RGB de um ponto qualquer da imagem, que por sua vez podem ser escritos em função das coordenadas (x, y) deste ponto, ou seja:

$$c = \begin{bmatrix} c_R \\ c_G \\ c_B \end{bmatrix} = \begin{bmatrix} R \\ G \\ B \end{bmatrix} \text{ (espaço de cores)} \quad (1)$$

$$c = \begin{bmatrix} c_R(x, y) \\ c_G(x, y) \\ c_B(x, y) \end{bmatrix} = \begin{bmatrix} R(x, y) \\ G(x, y) \\ B(x, y) \end{bmatrix} \text{ (imagem)} \quad (2)$$

Uma preocupação necessária ao se trabalhar com imagens policromáticas é se os dados adicionais obtidos com o processamento a cores recompensam a alocação de memória e tempo de processamento exigidos. É bom ponderar ainda se os algoritmos trabalham com os canais individualmente ou diretamente no espaço vetorial, sobretudo para otimização da análise. Neste trabalho optou-se por

transformar a imagem para tons de cinza, uma vez que parte dos métodos computacionais utilizados não comportavam matrizes tridimensionais em sua implementação padrão, e a determinação de posição do animal ocorreu de forma satisfatória nessa condição.

1.1.2 Tipos de operações com imagens

Por vezes, os processos de aquisição e armazenamento de imagens distorcem as informações originais ou as apresenta em formatos que não são ideais para a aplicação desejada, podendo-se aplicar métodos matemáticos e estatísticos para adequação. Em (9), os autores estabelecem diferenças entre realce e restauração de imagens: o primeiro tem protocolo indefinido e delimitado conforme o conceito de “imagem adequada” do usuário, enquanto no segundo ele se baseia em modelos teóricos de degradação.

Outra subdivisão é que os métodos podem ser aplicados diretamente na imagem, também chamado de espacial, ou no domínio da transformada: calcula-se uma nova função $T(u, v) = T(f(x, y))$ a partir da $f(x, y)$ descrita anteriormente, fazendo então a operação em $T(f(x, y))$ e obtendo $g_r(T)$ que será transformada numa $f_r(x, y)$ análoga à $f(x, y)$. Em processamento de imagens é bastante comum o uso de transformadas integrais com kernels separáveis e simétricos, permitindo que as transformadas bidimensionais sejam computadas como combinação de unidimensionais e a imagem possa ser totalmente recuperada a partir de sua transformada direta.

Matematicamente, transformadas integrais possuem a seguinte forma geral (9):

$$T(u, v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) r(x, y, u, v) \quad (3)$$

onde $f(x, y)$ é a imagem de entrada, $r(x, y, u, v)$ o kernel de transformação direta, u e v são chamadas de variáveis de transformada, $u = 0, 1, 2, \dots, M - 1$ e $v = 0, 1, 2, \dots, N - 1$, sendo M o número de linhas e N de colunas de $f(x, y)$.

$T(u, v)$ é então chamada de transformada direta de $f(x, y)$ e, de posse dela, é possível recuperar a imagem original através da transformada inversa de (u, v) :

$$f(x, y) = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} T(u, v) s(x, y, u, v) \quad (4)$$

em que $s(x, y, u, v)$ é o kernel de transformação inversa, $x = 0, 1, 2, \dots, M - 1$ e $y = 0, 1, 2, \dots, N - 1$. Se os kernels r e s forem separáveis e simétricos, podem ser escritos conforme k abaixo:

$$k(x, y, u, v) = k_1(x, u)k_2(y, v) = k_1(x, u)k_1(y, v). \quad (5)$$

Considerando $f(x, y)$ uma imagem quadrada de tamanho $M \times M$, a equação (3) de $T(u, v)$ poderá ser reescrita como:

$$\mathbf{T} = \mathbf{AFA} \quad (6)$$

com $\mathbf{F}$ sendo a matriz $M \times M$ contendo os elementos de $f(x, y)$, $\mathbf{A}$ matriz $M \times M$ com elementos $a_{ij} = k_1(i, j)$ e $\mathbf{T}$ é a transformada $M \times M$ resultante com valores $T(u, v)$ para $u, v = 0, 1, 2, \dots, M - 1$.

Para obter a transformada inversa, multiplicamos antes e depois a anterior por $\mathbf{B}$, matriz de transformação inversa:

$$\mathbf{BTB} = \mathbf{BAFAB} \xrightarrow{\text{Se } \mathbf{B}=\mathbf{A}^{-1}} \mathbf{F} = \mathbf{BTB} \quad (7)$$

indicando que é possível recuperar a imagem a partir de uma transformada que utilize kernel separável e simétrico. Essa descrição se aplica às transformadas de Fourier, de Walsh, de Hadamard, discreta de cosseno, de Haar e de slant (9), fundamentais em processamento de imagens.

1.1.3 Operações no domínio espacial

Os métodos que operam no plano contendo os pixels da imagem, ou seja, no domínio espacial, são comumente chamados de Filtros Espaciais. Podem ser expressos por:

$$g(x, y) = T[f(x, y)] \quad (8)$$

onde $f(x, y)$ é a imagem de entrada, $g(x, y)$ a imagem de saída e T um operador em f definido na vizinhança do ponto (x, y) . (9)

Na figura abaixo podemos visualizar o que seria uma vizinhança 3×3 ao redor de um ponto (x, y) qualquer da imagem, aproveitando ainda para ilustrar onde convencionou-se a origem do plano cartesiano e de suas variáveis. Em muitos dos métodos aplicados ao domínio espacial, a intensidade (x, y) em $g(x, y)$ é calculado aplicando o operador T à vizinhança de (x, y) em $f(x, y)$. Assim, para se obter $g(x, y)$, move-se a máscara espacial, *template*, kernel ou janela por todos os pixels da imagem.

Figura 2 – Vizinhança 3 x 3 ao redor de um ponto (x, y) em uma imagem f .

Fonte: GONZALEZ (11)

Uma subclasse das operações no domínio espacial são aquelas baseadas em seu histograma: uma representação gráfica de quantas vezes cada intensidade ocorre na imagem. O histograma de uma imagem digital com níveis de intensidade contidos no intervalo $[0, L - 1]$ é uma função discreta $h(r_k) = n_k$, onde r_k corresponde ao k -ésimo valor de intensidade e n_k o número de pixels da imagem com intensidade r_k . Ele fornece estatísticas úteis e pode ser utilizado para identificar e corrigir imagens com predominância de tons claros ou escuros, bem como aquelas com pouco contraste.

Assim, uma possível operação para melhorar o contraste de uma imagem é equalizar o histograma. Nela, é calculada uma função de transformação que busca homogeneizar os valores n_k da imagem e acaba por fazer que os níveis de intensidade da imagem equalizada cubram um maior intervalo da escala de intensidades. (9) Abaixo, imagens com predominância de tons claros e com pouco contraste e seus respectivos histogramas antes e após a equalização. Esta técnica não foi utilizada no trabalho pois o realce do contraste não foi um limitante e os resultados não foram uniformes a todos os frames, comprometendo a consistência dos resultados para o vídeo com um todo.

Figura 3 – Uma imagem com predominância de tons claros e outra com pouco contraste antes e após Equalização do Histograma.

a) Histograma de uma imagem clara; b) Histograma da imagem clara após Equalização;
c) Histograma de uma imagem com pouco contraste; d) Histograma de uma imagem com pouco contraste após Equalização.

Fonte: Adaptada de GONZALES (11)

1.1.4 Operações no domínio da frequência

Uma alternativa por conjectura mais eficiente que mover janelas por toda a imagem é modificar a transformada de Fourier dela e calcular a transformada inversa para obter a imagem processada (9). Seja uma imagem digital $f(x, y)$ de dimensão $M \times N$, a equação básica de filtragem terá a seguinte forma:

$$g(x, y) = \mathfrak{F}^{-1}[H(u, v)F(u, v)] \quad (9)$$

Com $\mathfrak{F}^{-1}$ sendo a Transformada Inversa, $F(u, v)$ a Transformada inicialmente aplicada à imagem de entrada, $H(u, v)$ uma função filtro e $g(x, y)$ a imagem de saída.

Baixas frequências na transformada estão relacionadas a variação lenta de intensidade numa imagem, enquanto altas às transições abruptas. Assim, ao definir $H(u, v)$ como um filtro passa-baixa – que atenua altas frequências, temos a imagem $g(x, y)$ suavizada relação à $f(x, y)$ e, similarmente, ao usar um filtro passa alta, bordas seriam destacadas.

1.1.5 Suavização

Uma das principais maneiras de se suavizar a imagem $f(x, y)$ é atribuir algum momento estatístico, por exemplo média ou mediana, da intensidade da vizinhança de (x, y) em f ao pixel (x, y) em g . É possível atribuir peso maior ou menor a cada posição do kernel, conforme o estilo de ruído que se deseja amenizar, bem como utilizar de distribuições estatísticas como a de Gauss. A título de ilustração, observe-se a imagem abaixo extraída de (11), em que temos uma imagem corrompida com ruído sal e pimenta e os resultados das correções com filtros que usam média e mediana.

Figura 4 – Correção de Ruído Sal e Pimenta em imagem de raios X de um circuito:
Imagem original corrompida; b) Redução de ruído com filtro de média da vizinhança 3x3; c) Redução do ruído com filtro de mediana.

Fonte: GONZALEZ (11)

Para este trabalho, foi escolhido um filtro que utiliza média e fixou-se a dimensão do kernel conforme o tamanho da imagem: seja $f(x, y)$ uma imagem de M linhas e N colunas, o template será um retângulo de $(M \div 100)$ por $(N \div 100)$. Assim, espera-se que a suavização mantenha sua eficiência em diversas condições de dimensão da imagem.

1.1.6 Aguçamento e detecção de bordas

Se o cálculo de médias para suavização é análogo à integração, para aguçamento se trabalha com diferenciação. A força da resposta do operador será proporcional ao nível de descontinuidade na intensidade da imagem, de modo que bordas e outras inconstâncias como ruídos serão realçadas e áreas com variação suave atenuadas. Em imagens digitais, as derivadas de primeira e de segunda ordens podem ser escritas respectivamente nas formas (10) e (11), em função de diferenças:

$$\frac{\partial f}{\partial x} = f(x+1) - f(x) \quad (10)$$

$$\frac{\partial^2 f}{\partial x^2} = f(x+1) + f(x-1) - 2f(x) \quad (11)$$

Tipicamente (9) se definem três modelos de bordas: rampa, linha e telhado. Na rampa, há uma variação entre as intensidades de uma e outra região de forma pouco abrupta, em contraste com as do tipo degrau. A principal característica da borda estilo telhado é sua simetria, tal como se pode observar na figura abaixo no ponto identificado como “Thin line”.

Figura 5 – Perfil de intensidades de uma região em uma imagem com os três tipos de bordas descritos na literatura.

Fonte: GONZALEZ (11)

Neste trabalho, detectores de borda foram utilizados para construção de contornos em imagens binárias – que contém apenas as intensidades máxima e

mínima, de modo que a borda será no estilo degrau, bastante simples de ser localizada. Especificamente utilizou-se o Detector de Bordas de Canny, cujas definições foram deduzidas a partir da representação matemática daquilo que ele considerava ser as características ideais de um detector de bordas. Há descrição detalhada em (9), capítulo 10 seção 2, e dela se destaca tratar-se de um algoritmo multi-estágio e de desempenho superior a outros mais simples.

1.1.7 Segmentação de imagens

Segmentar uma imagem consiste em subdividi-la nas regiões que a compõe (9), por vezes referindo-se aos objetos nela constantes. É uma das tarefas mais difíceis no processamento de imagens: o nível de detalhamento deve parar quando houver detecção das regiões de interesse e não há garantias de que um protocolo acertado para uma dada aplicação funcione bem para outra, ainda que com alto grau de semelhança.

A segmentação subdivide-se em duas estratégias principais: descontinuidade e similaridade. A primeira baseia-se em grandes alterações de intensidade, considerando-as como divisoras de regiões – bordas, e a segunda em quão semelhante um grupo de pixels é de outro próximo – tipicamente quanto à intensidade ou textura. É possível ainda aliar essas táticas umas com as outras ou com outros elementos descritivos da imagem, tal como seu histograma.

Na Limiarização, dado um valor de corte, haverá uma redefinição dos níveis de intensidade dos pixels da imagem: tipicamente os que estiverem acima deste limiar assumirão intensidade máxima, e os abaixo, mínima. (12) De posse da imagem binarizada, pelo critério de se os pixels são conexos ou não se define as regiões, facultando utilizar de processamento morfológico para lidar com ruídos.

A maior dificuldade na Limiarização ou *Thresholding* é definir o valor que será utilizado para corte. Nesse sentido, em (13) Otsu buscou resolver a questão aplicando estatística ao histograma da imagem: o limiar k ideal é aquele que maximiza a variância entre classes. Inspirado por ele, para este trabalho concebeu-se uma sequência de operações que estabelece o valor de corte conforme a população do histograma.

1.1.8 Similaridade entre duas imagens

Um problema recorrente em Visão Computacional é determinar quão semelhantes duas imagens são entre si, sobretudo em virtude de ruídos. Um primeiro instinto é calcular o Desvio Médio Padrão (*MSD* em inglês) em associação com a Relação Sinal-Ruído de Pico (*PSNR* em inglês) - simples de se calcularem e com significados físicos bem estabelecidos. Todavia seus resultados não coincidem muito bem com aspecto visual, tal como indicado em. (14)

Nesta mesma publicação se propõe então o Índice de Similaridade Estrutural (*SSIM* em inglês): através de comparações entre a luminância l , o contraste c e estrutura s , resulta um decimal entre -1 e 1, sendo o 1 obtido para imagens idênticas.

Sejam duas janelas A e B de dimensão $N \times N$, μ_A a média de A , μ_B média de B , σ_A^2 a variância de A , σ_B^2 a variância de B , σ_{AB} a covariância de A e B e c_1 , c_2 , c_3 constantes, a luminância é definida em função das médias:

$$l(A, B) = \frac{2\mu_A\mu_B + c_1}{\mu_A^2 + \mu_B^2 + c_1} \quad (12)$$

Por sua vez, o contraste em termos das variâncias:

$$c(A, B) = \frac{2\sigma_A\sigma_B + c_2}{\sigma_A^2 + \sigma_B^2 + c_2} \quad (13)$$

E finalmente a estrutura em função da covariância e raízes das variâncias:

$$s(A, B) = \frac{\sigma_{AB} + c_3}{\sigma_A\sigma_B + c_3} \quad (14)$$

O Índice de Similaridade estrutural será então uma combinação em que o peso de cada descritor acima é dado pelos expoentes α , β e γ , que tipicamente assumem valores iguais a 1.

$$SSIM(A, B) = [l(A, B)^\alpha \cdot c(A, B)^\beta \cdot s(A, B)^\gamma] \quad (15)$$

1.2 Ferramentas Disponíveis no Mercado

1.2.1 X-PloRat

Em 2002 foi lançada a primeira versão do X-PloRat (15), em que, após alimentar o software com informações como formato da caixa, número de divisões, em qual quadrante o animal será solto e duração do teste, o usuário assiste ao vídeo e anota

as transições entre as regiões utilizando setas direcionais do teclado, obtendo ao fim do teste, planilhas com “Frequência” e “Tempo” em cada região, com possibilidade de se configurar atalhos no teclado para comportamentos específicos como erguer o tronco ou autolimpeza, de modo que ao acionar (e segurar) a tecla se registra a duração e em qual quadrante ocorreu. Uma das restrições desse software é o número limitado de caixas a que pode servir, sobretudo com relação ao formato: retangular, quadrado, em cruz ou de “I”. É justo ainda o destaque ao cálculo do deslocamento se dar de uma maneira imprecisa, através da soma das distancias entre os centros dos quadrantes, bem como ao viés de um analisador para outro na interpretação das ocorrências para anotação e até mesmo por um mesmo usuário por condições de cansaço ou pressão, por exemplo. Representa, no entanto, grande avanço se comparado à anotação manual dos comportamentos e deslocamentos. Foi disponibilizado por terceiros para download no GitHub após não estar mais disponível em seu repositório original.

1.2.2 EthoVision

Corresponde a umas das ferramentas mais completas, pois além de fazer o *tracking* do animal pelo seu centro geométrico, automaticamente detecta outros comportamentos bem caracterizados para ratos e camundongos, e pode ser utilizado para rastreamento de outros animais. Pioneira e em constante otimização (16), teve sua primeira versão lançada em *DOS*, usando linhas de comando para execução. É um produto da *Noldus Information Technology*, uma empresa que desenvolve soluções para pesquisa comportamental. Dificultam uma utilização mais ampla e difundida a pouca transparência dos algoritmos utilizados e licenças a partir de \$5850, sem incluir todas as funcionalidades descritas.

1.2.3 EthoWacher

Trata-se de iniciativa de reprodução das funcionalidades do *Ethovision* de forma aberta e gratuita, tendo sido escrita em C++ (17). Promete grandes avanços comparados a seu modelo, mas exige alimentação qualificada por parte do usuário – seja dando um perfil de comportamentos já nos moldes do programa ou treinando-o a partir de vídeo por indicações de onde começa e termina cada ação. Mesmo que se deseje apenas fazer o *tracking* do animal é preciso calibrar o software,

carregando o vídeo no programa e marcando frames que correspondam à caixa vazia, ao animal, podendo exigir outros parâmetros de visão computacional nem sempre bem compreendidos pelo usuário. É possível marcar os instantes de outros comportamentos como erguer o tronco ou autolimpeza e usar a Ethowacher para detectá-los, todavia não se encontrou dados a respeito do sucesso da ferramenta nessas atribuições.

1.2.4 Mouse Move

Mais um fruto da crescente demanda de ferramentas para tomada automática de dados em Testes do tipo Campo Aberto, sustentada por (7). Constitui alternativa a sistemas comerciais, trazendo inovações como fornecer informações qualitativas e quantitativas de forma semiautomática e com alto rendimento (18), abrindo mão da detecção de comportamentos para tal. Traz ainda novos parâmetros como angulação, raio e direção dos deslocamentos, fatores significativos ao estudo utilizado para a validação da ferramenta: análise da atividade motora de ratos em que se induziu derrame cerebral e devem apresentar paralisia em apenas um dos lados do corpo. Não apresenta informações claras no artigo sobre seu funcionamento em caixas não circulares, sugerindo reproduzir o tamanho de caixa e posicionamento de câmera para obtenção de resultados. Outro fator restritivo a seu acesso é ter sido desenvolvida em LabVIEW, cuja licença básica anual custa próximo a \$1000 e só funciona na plataforma Windows – há outras opções sem limitação de prazo e sistema operacional, com custos proporcionais.

1.2.5 Kinoscope

Mais recente das ferramentas localizadas durante a realização deste trabalho, busca uma maior constância na análise dos resultados por diferentes membros da equipe, através da apresentação de mapas e gráficos visuais (19) em lugar de dados brutos. Entretanto, até onde se pode consultar, não suportava o Teste Comportamental do tipo Campo Aberto, embora esteja disponível para *download* e edição (foi escrito em C#), de modo que com ajustes possa servir a este propósito.

1.3 Python

Toda a ferramenta foi desenvolvida em linguagem de programação Python: com vasta biblioteca padrão já inclusa no código fonte (20), e diversos módulos de parceiros, traz métodos estatísticos e matemáticos de uso recorrente em implementações direcionadas a aplicações específicas. Além disso possui uma ampla comunidade que por sua vez disponibiliza diversos tutoriais na web e em livros, sendo bastante acessível para propósitos científicos. Além disso é conhecida por permitir fazer muito com poucas linhas de comando, tornando-se bastante atrativa a pessoas com pouca experiência em programação de computadores.

Foi criada no início dos anos 90 por Guido van Rossum (21), e todas suas versões são *open source*, ou seja, não só é possível acessar e modificar o código fonte – permitindo alterações e criação de trabalhos derivados – como não há restrições a sua redistribuição (22). É considerada uma linguagem de alto nível – mais próxima à humana e distante do código de máquina (binário); interpretada – executada concomitantemente à leitura e tradução pelo sistema operacional ou processador; e flexível a vários estilos ou paradigmas de programação, notoriamente as formas imperativa, funcional e orientada a objetos.

Outra característica de destaque da Python é sua interoperabilidade com outras linguagens, permitindo utilizar das melhores características de cada plataforma e incorporar novas funcionalidades sem grandes reescritas de código fonte. É possível, escrevendo algoritmos em Python, desenvolver aplicações .NET (23) ou Java, bem como usufruir de todas as funcionalidades de bibliotecas como OpenCV e NumPy – originalmente da C/C++.

Foi escolhida a versão 2.7 em 32 bits, pois algumas das bibliotecas não funcionaram plenamente na fase inicial do projeto em Python 3.x e códigos em 32 bits são processados tanto por sistemas de arquitetura 32 quanto de 64 bits. A seguir, descrição detalhada de todas as bibliotecas utilizadas.

1.3.1 NumPy

Segundo seus criadores, trata-se de biblioteca *open source* desenvolvida para atender necessidades específicas da computação científica no início da popularização da Python, especialmente lidar com dados em formato de matrizes (24). Definir um objeto do tipo *array* numérico *N*-dimensional foi um dos grandes

avanços provido por ele, bem como ampla coleção de funções matemáticas de alto nível para processá-los, tais como a transformada de Fourier e outras referentes a conceitos da álgebra linear.

Um array N -dimensional consiste de uma coleção de itens indexados por N inteiros na forma de tuplas, uma para cada dimensão – por exemplo uma imagem de 20x30 pixels terá 20 tuplas com 30 intensidades em cada uma delas. Esta biblioteca faz-se essencial ao processamento computacional de imagens digitais, tanto que o próprio tutorial de instalação da OpenCV inclui passo para download da NumPy.

1.3.2 SciPy

Escrita para trabalhar com os arrays da NumPy, busca oferecer ao usuário comandos de mais alto nível e classes para manipulação e visualização de dados, também se voltando à problemas da computação científica (25) – vide seu nome. Inclui, dentre outros, métodos de aglomeração (*clustering*), cálculo integral, processamento de imagens, estatística, álgebra linear, otimização e transformadas discretas. Foi amplamente utilizada para estudo estatístico de variação de parâmetros e posterior definição desses na construção do algoritmo da ferramenta, bem como para algumas etapas do processamento de quadros e cálculo do centro geométrico.

1.3.3 Matplotlib

Conforme o nome indica, trata-se de biblioteca para produção de gráficos, voltando-se à exportação de imagens de boa qualidade e utilizando-se de scripts interativos (26), inclusive em comunicação com outras linguagens ou softwares. Propõe-se a fazer coisas fáceis fáceis, e coisas difíceis possíveis. (27) Através da função pyplot pode-se visualizar o gráfico interativamente como no MATLAB, oferecendo ainda controle completo a todas as características de um gráfico através de simples comandos para aplicações mais robustas.

1.3.4 scikit-image

Inicialmente escrita para complementar a SciPy, foi desenvolvida e distribuída independentemente desta, intentando processamento de imagens para aplicações em pesquisa, educação e indústria (28). Seu desenvolvimento é mais recente que da OpenCV, tendo sido menos empregado que esta última, ainda que sem uma análise mais objetiva e fundamentada para basear esta escolha. Todavia foi fundamental à construção da ferramenta por ter o método de cálculo do Índice de Similaridade Estrutural implementado.

1.3.5 OpenCV

Seu nome é um acrônimo de *Open Source Computer Vision*, e almeja obter resultados relacionados à Visão Computacional em tempo real, contendo algoritmos complexos – por exemplo realidade aumentada e reconhecimento de faces e gestos de maneira otimizada em seu código fonte. Ao contrário da scikit-image – amplamente escrita em Python, é redigida em C++ (29) e possui *bindings* com a primeira. Junto com a NumPy, foi a biblioteca mais utilizada em toda a ferramenta, seja para abrir o vídeo, extrair características, tratar ruídos, segmentar frames ou desenhar resultados para visualização.

2 MÉTODOS

2.1 Breve Histórico e Visão Geral do Algoritmo

Houveram três momentos bem distintos de desenvolvimento do trabalho. Num primeiro, com poucas técnicas de Visão Computacional efetivamente dominadas, a localização do animal era calculada pela diferença entre frames subsequentes, com bons resultados apenas quando havia alta taxa de movimentação. Num segundo, buscou-se na literatura técnicas de segmentação a serem aplicadas diretamente no frame e que pudessem servir ao propósito desejado, com efeitos satisfatórios para os vídeos a que se tinha acesso naquele instante. Todavia, em nova colaboração detectou-se pouca amplitude de aplicação da ferramenta, adotando-se uma nova estratégia híbrida das anteriores.

A efetiva localização do animal num dado instante passou a basear-se na subtração entre o frame correspondente e um modelo de como seria a imagem sem o animal – único objeto que se deseja detectar. Deste modo há dois passos cruciais para a qualidade da análise: a construção deste modelo do fundo e a segmentação da diferença entre ele e o frame a ser analisado.

Abaixo temos um fluxograma da extração e utilização dos dados: vários parâmetros dos métodos matemático-estatísticos utilizados são definidos conforme propriedades do vídeo, por exemplo sua taxa de quadros, dimensão e número total dos frames, estando sua extração indicada como “características”.

Figura 6 – Esquema geral dos dados extraídos pelo algoritmo.

Fonte: Elaborada pela autora

2.2 Modelo de Fundo da Imagem – Frame Médio

A construção de uma imagem que correspondesse a todo o cenário de realização do teste se baseou em fazer a média de um número suficiente de frames aleatórios do vídeo. Assim, as variações da intensidade que correspondem à presença do animal naquela região são dissolvidas pelo cálculo da média, e se pode então detectá-lo independentemente de ser mais claro ou escuro que a caixa em que é colocado para realização do teste.

Foram tomadas providências para evitar que um frame seja selecionado duas vezes para compor o conjunto a partir do qual se constrói a imagem média. Sempre que alguma etapa do algoritmo envolver escolha aleatória de um frame, esta obedecerá a condição de haver no mínimo 5 (cinco) segundos de diferença entre o instante dele e de todos os demais já escolhidos anteriormente.

Inicialmente se toma um número arbitrário N de quadros do vídeo e se compara a imagem média obtida a partir de todos os N frames com aquela a partir de $N - 1$: caso sejam muito diferentes entre si, adiciona-se mais um frame ao conjunto e se refaz o teste.

Para critério de diferença, escolheu-se trabalhar com o Índice de Similaridade Estrutural (*SSIM* em inglês), descrito com maiores detalhes na Introdução. Realizou-se análise estatística da distribuição dos valores de similaridade obtidos em função da adição de novos frames, adotando-se como critério de parada o incremento no *SSIM*:

1. calcula-se as imagens médias a partir de N , $(N - 1)$ e $(N - 2)$ frames, respectivamente as imagens A, B e C;
2. obtém-se os valores de similaridade S_1 entre A e B e S_2 entre B e C;
3. determina-se o ganho em similaridade através da diferença $|S_1 - S_2|$;
4. se o ganho for inferior a um limiar, considera-se a imagem B como modelo de *background* do vídeo;
5. caso o ganho seja superior, adiciona-se mais um frame ao conjunto usado para cálculo da média e se reinicia o procedimento.

A fim de determinar qual o limiar de ganho suficientemente pequeno para passagem à próxima etapa, plotaram-se gráficos da diferença $|S_1 - S_2|$ para dois vídeos representantes das duas condições a que se tinha acesso – um com muito pouco contraste entre o animal e a caixa – camundongo branco, e outro com valores

mais altos – gerbilo. Após a observação de alguns deles, fixou-se como limiar de parada a variação de similaridade de 0,00035, quando a curva perde inclinação.

Figura 7 – Plotagem do ganho em similaridade conforme novos frames são adicionados ao conjunto a partir do qual se calcula a média para obter um modelo de como seria a imagem sem o animal.

Fonte: Elaborada pela autora.

Quando o ganho é maior que o limiar, entende-se que adicionar um novo frame ao conjunto modifica consideravelmente a imagem média resultante. Assim, a intensidade da região correspondente ao animal não teria sido dissolvida o suficiente – observação dos frames médios ao longo da operação corroboram o raciocínio.

Considerando a possibilidade de o valor escolhido a partir do gráfico não ser atingido, adicionou-se uma condição de interrupção alternativa: quando o conjunto de frames utilizado para o cálculo do modelo atingir o máximo de elementos possível dada a condição de distância temporal de cinco segundos.

É relevante a discussão de que existem métodos bem estabelecidos para se obter um modelo de fundo para vídeos, inclusive já implementados de maneira que se dispensaria as próximas etapas da ferramenta – *GMM*, *MOG* e *MOG II*. Todavia,

em testes preliminares ainda na primeira etapa de desenvolvimento do projeto, observou-se que aqueles que por atualizarem o fundo para a subtração acabavam por perder o animal em momentos de pouca motilidade.

Além disso, tratam-se de algoritmos complexos, de compreensão e alimentação de parâmetros menos acessíveis a pessoas com pouca experiência em computação, contrariando dois dos paradigmas da ferramenta: clareza do algoritmo e possibilidade de expansão para novas funcionalidades. Outrossim, a maneira simplificada de subtração de fundo teve resultados bastante satisfatórios, tal como se mostrará posteriormente.

2.3 Rastreamento do Animal

De posse da imagem que corresponde a uma aproximação de o que seria o vídeo sem o animal e de outras propriedades, se inicia agora a análise frame a frame. A seguir temos um fluxograma das principais etapas constituintes para visualização do processo em totalidade.

Figura 8 – Esquema simplificado dos métodos utilizados na determinação da localização do animal num dado instante T .

Fonte: Elaborada pela autora.

Observar o destaque ao fato de que o frame é conexo a um instante T e, portando, o centro geométrico calculado que corresponderá à localização do animal também o será. Assim, se poderá fornecer ao usuário informações como velocidade

do animal, não obtidas na análise tradicional do Testes Comportamentais do Tipo Campo Aberto.

Após abrir o vídeo, pode-se trazê-lo ao instante em que efetivamente o teste começa – informação dada pelo usuário. Em seguida, o frame correspondente a este momento passará por suavização para tratar possíveis ruídos – operação de borramento (*blur*) com substituição do valor do pixel pela média das intensidades dos pontos da vizinhança, de dimensão definida em função do número de linhas e colunas.

A diferença é calculada com operação matriz $\times$ matriz em que cada elemento da resultante será o módulo da subtração entre os respectivos pixels no frame a ser analisado e no modelo de background. Caso se fizesse sem este cuidado, valores de intensidade negativa poderiam ser obtidos – seriam visualizados como um ponto preto na imagem, e comprometeriam os passos seguintes.

Para trazer maior continuidade e fluidez à superfície obtida, aplica-se uma máscara de suavização por média com o dobro do tamanho da anteriormente utilizada. O objetivo é reduzir flutuações isoladas e unir regiões que correspondam ao animal e possam estar separadas, otimizando os resultados da segmentação, crucial ao êxito da ferramenta.

A etapa sucessora é construir um histograma a partir da matriz da diferença – ao contrário de imagens digitais e graças à operação de suavização, poderão ocorrer valores não inteiros. Assim, colocou-se condição de o histograma ter no mínimo cinco bins, mesmo que não se tenha verificado na prática ocorrência de distribuições dos elementos que a fizessem necessária.

De posse do histograma, procura-se pelo valor de intensidade do bin cuja soma de sua população com a dos de intensidade superiores não ultrapasse 3,5% da população total da distribuição: percorrendo o histograma de trás para frente, soma-se a quantidade de pontos com intensidade no intervalo daquele bin até que ultrapasse 3,5% da quantidade de pontos total da imagem, seguindo para a próxima etapa do algoritmo.

Essa porcentagem foi definida considerando que as caixas costumam conter de 16 a 36 regiões e que a dimensão da região deve ser próxima à do corpo do animal. Não se utiliza, entretanto, o valor deste bin como limiar na próxima etapa, e sim o do bin com intensidade imediatamente inferior – assim se garante mais pontos que a estimativa da área correspondente ao animal na imagem binarizada.

De posse do *array* da diferença em módulo e do limiar acima descrito, pode-se então criar uma imagem binária: todas as intensidades no *array* maiores que o valor de corte levarão a um pixel de intensidade máxima na posição correspondente na imagem resultante. Analogamente valores inferiores corresponderam a pixels de intensidade nula. Este método é popularmente conhecido por *thresholding* ou limiarização e foi descrito na Introdução.

Faz-se agora um pequeno tratamento de ruídos sempre presentes após limiarização. Através de operação morfológica de erosão, pequenas regiões desconexas e irregularidades nas bordas daquelas maiores são eliminadas. Logo em seguida, utilizando-se de operação de fechamento, pequenas lacunas são fechadas nas regiões detectadas. Assim, espera-se que hajam poucas regiões a serem detectadas na próxima etapa, bem como não se tenham muitos buracos nas mesmas – importante ao desempenho da ferramenta.

Como explicita-se no parágrafo anterior, não se espera que se obtenha uma imagem com apenas a região correspondente ao animal em branco e o restante em preto. Dessarte, é necessário estabelecer algum critério para selecioná-la dentre as demais, adotando-se a área, em concordância com os passos para construção do resultado até este momento. Com auxílio do algoritmo de Canny (30) detecta-se as bordas das regiões, e então se constrói descritores para cada região conexa dos pontos pertencentes ao contorno.

Usualmente, na própria biblioteca que contém o método para determinação dos contornos há outros para que se extraiam características deles: por exemplo área e centro geométrico. Com a área selecionamos a região com maior potencial de corresponder ao animal, e do centro geométrico determinamos a posição do animal no frame. Feitas todas estas operações com o frame de um instante T basta repeti-la recursivamente ao $(T + 1)$ até o término do vídeo para rastrear a posição do animal durante toda a realização do teste comportamental.

2.4 Tratamento da Saída de Dados

Um típico vídeo gravado com câmeras digitais modernas ou até mesmo smartphones oferecem taxa de amostragem de 24 a 60 frames por segundo. Assim, o algoritmo acima consegue extrair mais informações do vídeo do que o usuário realmente precisa, tornando necessário se pensar em alternativas para saída de

dados. Escolheu-se então fornecer um ponto (x,y) de localização na imagem por segundo. Isso vem ainda de encontro a tratar possíveis flutuações nas posições calculadas, geradas principalmente na etapa de seleção de objeto pela maior área.

Esse cuidado consiste em analisar a variação da posição durante o curto intervalo de tempo em questão, antes de entregá-la ao usuário. Calcula-se o desvio médio padrão da série de dados e, caso este seja alto, elimina-se o elemento que mais contribuiu para ele. Com um elemento a menos, o desvio é calculado novamente e, até que seja atingido um valor aceitável – só então se considera como posição do animal a média naquele segundo.

Além da saída de dados como uma planilha, oferece-se um gráfico da dispersão de (x,y) ao longo da realização do teste. Ele traz consigo uma correspondência à dimensão física do Teste Campo Aberto: pode-se visualizar a predominância em ficar nas regiões das paredes ou centrais da caixa.

A próxima informação a ser extraída é o deslocamento: calcula-se a distância euclidiana – Teorema de Pitágoras, entre as localizações de segundos subsequentes, com unidade de pixels. Ainda que sem correspondência a unidades de distância físicas como centímetros ou polegadas, esta informação serve aos propósitos comparativos dos efeitos causados por distintas condições dos testes comportamentais, desde que o posicionamento da câmera seja mantido em relação à caixa, inclusive quanto à angulação.

Ao observar um gráfico da distribuição dos deslocamentos ao longo do tempo, é possível identificar intervalos de maior motilidade, o que juntamente com a informação da posição dá importantes indicações do grau de conforto do animal com o teste ou sua ansiedade. Não é inacessível ao usuário determinar, a partir do frame médio salvo no computador, quantos pixels da imagem correspondem ao comprimento da caixa, de modo que simples regra de três é suficiente para ter uma correspondência dos deslocamentos com uma unidade física de distância.

Para que os resultados desta ferramenta pudessem ser comparados à metodologia tradicional e ao X-PloRat, buscou-se a partir dos dados reconstruir a caixa com suas respectivas regiões. Partindo dos pressupostos de que:

1. animal esteve em contato com todas as paredes da caixa;
 2. a caixa tinha formato retangular ou quadrado;
 3. a câmera esteve posicionada no centro da caixa;
- seus vértices foram obtidos através dos máximos e mínimos em x e y .

Para determinação das regiões, além dos pressupostos acima é necessário saber qual a grade utilizada para dividi-la. De posse dessa informação, basta dividir espacialmente os pontos obtidos para se saber quanto tempo o animal esteve em cada quadrante.

Seja (i, j) o índice da matriz representante da grade de m linhas e n colunas que divide a caixa, é possível reescrevê-lo como um único índice $k = 0, 1, 2, \dots, i * j$ através de uma transformada, a fim de que nesse novo sistema se determine a qual quadrante o ponto pertence por divisão simples, sem gastar processamento com i^j condicionais:

$$k = mi + j + 1 \quad (16)$$

Dado o ponto (x, y) correspondente a uma localização na imagem, os índices i e j referente a qual região da caixa ele está será dado por:

$$(i, j) = \left((m - 1) \frac{x}{x_{\text{máx}} - x_{\text{min}}}, (n - 1) \frac{y}{y_{\text{máx}} - y_{\text{min}}} \right) \quad (17)$$

De posse de k é simples construir um vetor com a sequência de seus valores obtidos. Com ele é possível verificar se há transições entre quadrantes: ao comparar cada item a partir do segundo com seu anterior, repetindo-se a informação de quantas vezes se entra em alguma região dada pelo XploRat. Ainda, através da contabilização do número total de ocorrências de cada valor de k se determina quanto tempo o animal ficou na região correspondente.

Em associação com a informação da distribuição dos deslocamentos ao longo do tempo, pode-se gerar gráficos com o deslocamento e o quadrante da posição – podendo-se identificar num mesmo gráfico se as janelas de pouca motilidade foram em região periférica ou central

3 RESULTADOS

Esta seção subdivide-se em três etapas principais do algoritmo: resultados intermediários, para ilustrar o funcionamento e construção passo a passo do mesmo; resultados finais – rastreamento ao longo do tempo; e validação da ferramenta – comparação com a XploRat. Para contextualização, apresenta-se frames extraídos de vídeos de Testes Comportamentais aplicados utilizando-se da mesma caixa pelo CCBS¹ a camundongos (*Mus Musculus*) – serão referidos como “Baixo Contraste” e gerbilos (*Meriones unguiculatus*) – indicados como “Alto Contraste”, animais de espécies e tonalidade de pelagem distintas.

Figura 9 – Quadro extraído de vídeo de Teste realizado em gerbilo – “alto contraste”
Fonte: Elaborada pela autora.

Figura 10 – Quadro extraído de vídeo de Teste realizado em camundongo – “baixo contraste”
Fonte: Elaborada pela autora.

¹ Centro de Ciências Biológicas e da Saúde – Departamento de Fisioterapia – Universidade Federal de São Carlos (UFSCar)

De início já se pode notar que os quadros têm razão altura $\times$ largura diferentes: 720 $\times$ 1280 no primeiro e 480 $\times$ 704 pixels no segundo. Outrossim, o posicionamento da câmera em relação à caixa e, conseqüentemente, a região ocupada por ela no quadro também diferem substancialmente. A fim de reduzir o impacto desses elementos, trabalhou-se apenas com a Região de Interesse (*ROI* em inglês), um recorte do quadro original que contenha apenas a caixa, a ser selecionada pelo usuário.

Trabalhar com *ROIs* é ainda vantajoso do ponto de vista computacional, pois diminui a dimensão das informações a serem manipuladas para obtenção dos resultados. Entretanto, não a definir não implica necessariamente em falha total do algoritmo, conforme pôde ser observado em testes sem sua definição para alguns dos vídeos. Dois elementos foram identificados como cruciais para sucesso da ferramenta sem definição das regiões de interesse: controle do ambiente – evitando que outros objetos apareçam fora da caixa e a área ocupada pela mesma na imagem.

Houveram tentativas de desenvolver uma etapa de algoritmo que fosse capaz de detectar a caixa automaticamente, eliminando dependência do usuário, sem muito sucesso. Nas poucas vezes que se detectou a caixa, diversas operações extremamente direcionadas aos ruídos que ali ocorriam foram utilizadas, preferindo-se que a informação seja obtida externamente.

3.1 Detecção do Animal e Saída de Dados

No segundo momento de desenvolvimento do projeto, a detecção do animal era baseada essencialmente na segmentação por intensidade a cada frame, o que tornaria a ferramenta ou pouco flexível ou muito mais dependente do usuário para inserir parâmetros. Frente a isso e a se ter utilizado a diferença entre frames subsequentes para detecção do animal num primeiro momento, chegou-se à abordagem híbrida apresentada em detalhes na metodologia.

A primeira etapa então do algoritmo com resultados exibíveis é a construção do modelo de background do vídeo. Tal como descrito na Metodologia, a principal questão era definir o número adequado de frames para cálculo da média. É necessário ressaltar que esse número não será constante dentro de um mesmo

vídeo, uma vez que envolve sorteio de frames, preferindo-se apenas ilustrar o aspecto visual do acréscimo de novos quadros ao conjunto que comporá a média.

Figura 11 – Aspecto Visual da adição de mais frames para compor o modelo, bem como da implicação do contraste entre animal e caixa.

Fonte: Elaborada pela autora.

Visualmente a média de nove frames extraídas do vídeo com o camundongo parece um modelo adequado de background, ao contrário daquele resultante de um mesmo número de iterações a partir do vídeo do gerbilo. Este efeito confere com as expectativas e reforça necessidade de se estabelecer um critério estatístico para finalização desta etapa e seguimento à próxima.

Sem o critério estatístico de parada – variação do ganho no Índice de Similaridade Estrutural causado pelo acréscimo de mais um frame ao cálculo do fundo do vídeo, a condição de contraste animal-caixa poderia comprometer a aplicabilidade da ferramenta. Um alto contraste animal-caixa, aliado a um número insuficiente de frames, poderia levar a um modelo com sombras, comprometendo a detecção especialmente nos frames em que o animal ocupa posição próxima a elas. Ainda, estabelecer um número arbitrariamente grande teria impacto computacional pelo volume de informações a serem processadas desnecessariamente.

Figura 12 – Modelo Final do Fundo do Vídeo obtido ao fim da etapa respectivamente para vídeos com gerbilo e camundongos.

Fonte: Elaborada pela autora

De posse dos resultados acima, pôde-se seguir ao *tracking* do animal. Cada fase desta etapa estará apresentada com relação à um mesmo instante do vídeo, para que melhor se possa observar os efeitos de cada uma delas. Ao final, se obterá uma coordenada e, ao percorrer todo vídeo – ou a parte indicada como de realização do teste, se obtém como resultado uma distribuição da posição em função do tempo.

De início, calcula-se a diferença em módulo entre o modelo de background do vídeo e frame referente ao instante t , com expectativas de que a região ocupada pelo animal apresente maiores variações. A seguir, este resultado passa por suavização, numa tentativa de reduzir descontinuidades de regiões e eliminar flutuações em pontos isolados. Em seguida é calculado um histograma da distribuição de valores da mesma, a partir do qual se definirá o limiar de corte para a limiarização que sucederá esta operação.

A escolha de se segmentar o frame a partir de um modelo de fundo resolve parte da dependência de se obter bons resultados apenas a partir de vídeos com razoável contraste entre animal e rato, observada no segundo momento de desenvolvimento da ferramenta. A priori, essa deficiência da ferramenta não seria uma demanda imediata pois o pesquisador poderia controlar este parâmetro em novos testes, mas a alternativa pensada provou-se factível e com boas respostas.

Para a representação dos resultados da etapa de cálculo do limiar de corte a partir do histograma, preferiu-se mantê-los com suas amplitudes de eixos automáticas em lugar de trazê-los a uma mesma escala, a fim de permitir uma melhor visualização da curva. Como as diferenças entre modelo de fundo e frame estão em forma de *array*, podem ser representadas como imagens, a fim de se prover uma visualização do resultado, tal como poderá ser verificado mais à frente.

Figura 13 – Histogramas calculados a partir dos quadros utilizados com exemplo. A linha vermelha indica o limiar obtido.

Fonte: Elaborada pela autora.

No segundo histograma 6,56% dos pixels tinham intensidade acima do limiar obtido pelo algoritmo, e o valor máximo para essa variável era 51. No primeiro, a porcentagem foi de 3,60% - bem próxima àquela arbitrária utilizada pelo algoritmo, e o valor de corte 44 numa amplitude máxima de 87. A obtenção de quase o dobro de

pontos do que se pensaria em tomar inicialmente não pode ser considerado um problema sem antes olhar a distribuição de pixels na imagem binarizada.

Optou-se por apresentar o resultado das diferenças entre modelo de fundo e frame sem normalização – assim poderia melhor se visualizar o quão distinto são os resultados a partir de contrastes animal-caixa diferentes.

Figura 14 – Resultados da Subtração em Módulo, sua Suavização e posterior Limiarização. O número indicado corresponde ao valor de corte obtido a partir do histograma.

Fonte: Elaborada pela autora.

Quanto à limiarização, pode-se observar que a região ocupada pelo animal foi ressaltada conforme se pretendia, juntamente com ruídos, tal como previsto na metodologia. Assim, o sucesso da localização fica em função da escolha da região correta para calcular o centro geométrico – em virtude de toda a construção do algoritmo, utilizou-se como critério de seleção a maior área, conseguindo-se marcar satisfatoriamente o animal, como pode ser visualizado a seguir.

Figura 15 – Resultados das etapas de Limiarização, Seleção de Objeto e Contorno.

Fonte: Elaborada pela autora

Com a informação do contorno, desenhado em azul na imagem colorida para fins de ilustração, pode-se calcular o centro geométrico da região nele contida. É uma boa aproximação para a localização do animal, tal como se pode visualizar nas imagens abaixo, em que se desenhou o ponto verde na coordenada calculada.

Figura 16 – Ilustração da saída final da etapa de detecção: desenho de ponto no centro geométrico calculado.

Fonte: Elaborada pela autora.

Como destacado em diversos momentos do texto, ao executar iterativamente as operações acima, uma vez a cada frame, conseguimos uma localização a cada instante do vídeo. Todavia, trata-se de muito mais coordenadas de posição por segundo do que o suficiente para extração dos dados de deslocamento, preferindo-se fornecer então uma média das posições a cada janela temporal, tomando-se providências para minimizar influências dos frames em que a escolha de região pela maior área não foi um bom critério.

Apresente-se então gráficos da distribuição dos pontos obtidos a partir de vídeo de teste aplicado em camundongos. O teste tem duração de cinco minutos e foi realizado no período diurno, um horário que o roedor tende a ser menos ativo. O vídeo tem amostragem de 30 quadros por segundo, altura de 408 pixels e largura de 704 pixels. Para efeitos de comparação, inclui-se ainda gráfico da dispersão de uma janela de cinco minutos do teste para o gerbilo, que originalmente tinha 15 minutos e foi gravado em vídeo também de amostragem 30 quadros por segundo e com altura de 720 por largura de 1280 pixels.

Figura 17 – Exemplo de Gráfico de Dispersão das Posições (x, y) durante a realização do teste com o camundongo 2.

Fonte: Elaborada pela autora

Figura 18 – Gráfico de Dispersão de (x, y) para o camundongo 3: observar a concentração de pontos na área central da caixa.

Fonte: Elaborada pela autora

Figura 19 – Gráfico de Dispersão da Posição para o gerbilo: Há uma distribuição mais homogênea das posições, dado o perfil de deslocamento diferenciado da espécie. Os eixos têm intervalos maiores por se tratar de imagem com maior número de pixels.

Fonte: Elaborada pela autora

A representação de resultados acima é particularmente interessante pela proximidade entre ela e a configuração física dos Testes Comportamentais. Através dela é simples a análise da predominância de se permanecer nas regiões periféricas e próximas à borda em comparação com a presença em regiões mais centrais, indicativos do grau de conforto com o teste e tendência à exploração por parte do animal.

Mais explicitamente, na figura 17 há predominância em se permanecer no canto inferior esquerdo da caixa, enquanto na 18 podemos observar um comportamento exploratório das áreas centrais, possíveis indicadores de estresse e conforto, respectivamente. Na figura 19 temos uma distribuição mais homogênea das posições, possível indicação de o animal estar habituado ao teste, não buscando as áreas de proteção.

Outra saída bastante interessante que se pode dar aos dados é um gráfico da variação de posição em função do tempo. Através dele é possível identificar janelas temporais de maior ou menor atividade locomotora, bem como avaliar a detecção da posição pela ferramenta: o animal não pode se deslocar de um canto a outro da caixa em um segundo.

Figura 20 – Variação do Deslocamento a cada segundo durante a realização do teste. Observar intervalo de valores menores: pouca motilidade do animal pode ser indicação de outros comportamentos como levantar o tronco ou autolimpeza.

Fonte: Elaborada pela autora

Figura 21 – Outro exemplo de perfil de Deslocamento extraído: não há momentos de predominância de pouca motilidade, o que em conjunto com outros parâmetros pode ser um indicador de estresse.

Fonte: Elaborada pela autora

Figura 22 – Perfil de Deslocamento do Gerbilo – por tratar-se de uma imagem maior, as variações de deslocamento em pixels se dão em outra escala.

Fonte: Elaborada pela autora

Além do gráfico, oferece-se ainda o deslocamento total do animal, dado pela soma de todos os deslocamentos obtidos pelo algoritmo. Como os resultados são altamente influenciados pelo tamanho da imagem, foi necessário definir algum fator de normalização. Escolheu-se então a diagonal da ROI definida pelo usuário: para os vídeos de testes com camundongos, tinha dimensão 337x289, e com o gerbilo 766x710.

Tabela 1 – Deslocamento Total obtido a partir de cada um dos vídeos.

Animal	Deslocamento Total em Pixels	Deslocamento Normalizado
Camundongo 1	6577	14,84
Camundongo 2	5729	12,93
Camundongo 3	9777	22,07
Camundongo 4	8001	18,06
Camundongo 5	8505	19,20
Camundongo 6	10073	22,74
Camundongo 7	11772	26,57
Camundongo 8	15519	35,03
Gerbilo	47699	14,86

Fonte: Elaborada pela autora

Uma informação que, correlacionada ao deslocamento, possa ser bastante substancial é o posicionamento do animal na caixa: um intervalo de pouca motilidade pode ter interpretações bastante distintas se tiverem ocorrido em região central ou periférica. Uma maneira de se indicar isto visualmente é colocar também no gráfico que quadrante o animal está, tal como na Figura 23.

Figura 23 – Gráfico da Variação de Deslocamento (em azul) e Informação sobre Quadrantes (em vermelho) para o Camundongo 5. Especificamente para este vídeo, a janela de pouca motilidade se dá junto a um canto da caixa.

Fonte: Elaborada pela autora

Abaixo, imagem indicativa da Nomenclatura dos Quadrantes para o XploRat e aquele gerado pela ferramenta proposta. Ela é fundamental para avaliação do gráfico da Figura 23 e daqueles que serão apresentados na seção de Validação da Ferramenta – fez-se uma conversão ao formato de dados do XploRat.

A1	B1	C1	D1	E1
A2	B2	C2	D2	E2
A3	B3	C3	D3	E3
A4	B4	C4	D4	E4
A5	B5	C5	D5	E5

Sistema de Referência XploRat

1	2	3	4	5
6	7	8	9	10
11	12	13	14	15
16	17	18	19	20
21	22	23	24	25

Sistema de Referência
Ferramenta Desenvolvida

Figura 24 – Sistema de Referência e Nomenclatura de Quadrantes para o XploRat e a Ferramenta Proposta.

Fonte: Elaborada pela autora

3.2 Validação da Ferramenta

A fim de se fazer uma avaliação objetiva da ferramenta aqui apresentada, buscou-se comparar os resultados de análise obtidos através da XploRat e da ferramenta proposta a partir de um mesmo grupo de vídeos. Para tanto converteram-se os dados de posição à uma planilha com a Frequência de Entrada e Tempo de Permanência em cada quadrante, tal como seria dada pela XploRat que constará no Apêndice.

De posse dela, plotaram-se dezesseis gráficos – dois para cada um dos vídeos já caracterizados. Optou-se por não apresentar todos eles aqui, e sim expor as informações de forma pragmática e exemplificativa. Não se pôde observar uma ótima correlação entre os dados das duas ferramentas, tal como se pode ser verificado nos gráficos abaixo. Mesmo quando as porcentagens de tempo de permanência se aproximam para a maioria dos quadrantes, não se observa fenômeno parecido para a Frequência de Entrada, tal como se evidencia com as figuras 25 e 26.

Figura 25 – Gráfico comparativo com melhor correspondência entre a porcentagem de tempo de permanência – Vídeo do teste realizado com o camundongo 8

Fonte: elaborada pela autora.

Figura 26 – Comparação entre as porcentagens de permanência em cada quadrante extraídas pela XploRat e pela Ferramenta proposta para o camundongo 3.

Fonte: Elaborada pela autora

Figura 27 – Gráfico de Ocorrências de Entrada para o mesmo Vídeo do Gráfico anterior: observar que quase não há correspondências (camundongo 8).

Fonte: Elaborada pela autora

Figura 28 – Melhor resultado de comparação para as ocorrências de entrada em cada quadrante – camundongo 2.

Fonte: Elaborada pela autora.

Dentre as causas levantadas para a baixa correlação entre os dados do XploRat e da Ferramenta Desenvolvida, a definição do quadrante a partir dos próprios dados foi expressiva. É possível que distorções da caixa causadas pela angulação da lente ou a um posicionamento não central da câmera levem a uma representação não equânime de cada quadrante, de modo que a simples divisão não corresponda aquilo observado durante o teste.

Além disso, o modo de definição do efetivo início do vídeo pode ter influenciado os resultados: na ferramenta há uma pequena lacuna entre o animal estar livre para se movimentar e não haver obstáculo como a mão do pesquisador, e em ambos os casos o início é marcado pelo avaliador, sendo possível a ocorrência de não sincronidade entre eles.

Todavia não se pode excluir da discussão a respeito da pouca correlação a origem dos dados do XploRat. Uma única analista assistiu a todos os vídeos e, enquanto os assistia, utilizando das setas direcionais do teclado deslocava um quadrado vermelho na caixa virtual do XploRat conforme a movimentação do animal observado. Uma das maiores dificuldades levantada pela analista é como definir de fato quando o animal cruzou de um quadrante a outro, bem como sincronizar a exibição do vídeo e o início do teste na ferramenta.

Conforme levantado na defesa deste trabalho, a análise dos testes comportamentais do tipo campo aberto por vezes se baseia na preferência por permanecer nas regiões: periféricas, centrais e intermediárias, e não necessariamente em relação aos quadrantes ocupados. Assim, fez-se nova manipulação dos resultados obtidos, somando as ocorrências e permanência nos quadrantes que compõe cada região.

Figura 29 - Comparação entre sistemas de referência: da Xplorat, o utilizado na Ferramenta e o definido em função das regiões periférica (P), intermediária (I) e central (C).

Fonte: Elaborada pela autora

O resultado obtido desta forma foi semelhante a quando não houve a aglomeração em regiões maiores: não é possível identificar correspondências entre os dados obtidos com a Xplorat e a Ferramenta Desenvolvida. Assim reitera-se alguns dos fatores já levantados: a natureza da coleta de informações é muito distinta em uma ferramenta e outra, bem como a divisão dos dados nas regiões não considerarem distorções de inclinação da caixa e grade angular da câmera.

Outrossim, expõe-se o gráfico com maior correspondência entre as ferramentas, com dados obtidos a partir do camundongo 7:

Figura 30 – Gráfico simplificado da ocorrência e tempo de permanência nas macrorregiões criadas.

Fonte: Elaborada pela autora

A fim de exemplificar a extração de informação comparativa a partir deste tipo de gráfico, trazemos abaixo os resultados obtidos com o vídeo do camundongo 4: considerando os resultados da ferramenta desenvolvida, é possível identificar uma maior permanência na região central em comparação com o de número 7.

Figura 31- Dados obtidos a partir do vídeo da realização do teste com o camundongo 4.

Fonte: Elaborada pela autora

4 CONSIDERAÇÕES FINAIS E PERSPECTIVAS FUTURAS

Testes Comportamentais são etapa obrigatória em diversas áreas de pesquisa e, o do tipo Campo Aberto, pela simplicidade de sua construção, se estabelece como dos mais recorrentes entre eles. Todavia, a extração de dados desse tipo de teste é controversa e por vezes não muito bem estabelecida, comprometendo a reprodutibilidade dos experimentos. Nesse sentido há uma demanda de ferramentas para extração dos dados, seja de forma automatizada ou apenas como auxiliar.

Nesse contexto, há diversas limitações como licenças ou restrição de plataformas, bem como descontinuação dos projetos, justificando-se a proposição de mais uma Ferramenta para Tomada Automatizada de Dados. Além disso a escolha de fazê-la em linguagem Python, das mais acessíveis e com comunidade bastante ativa, permite que o usuário possa adaptá-la e ampliá-la a novas funcionalidades e outras modalidades de testes. Contribui para a criação de trabalhos derivados dar-se preferência por operações simples e intuitivas na construção do algoritmo.

Outrossim a Ferramenta traz consigo avanços como trazer o deslocamento de maneira mais precisa, bem como diversas saídas gráficas para interpretação facilitada do usuário, poupando tempo de fazer gráficos a partir dos dados fornecidos. A exportação dos dados neste formato pode ainda aumentar a quantidade de informações extraídas do teste – ainda que sejam derivadas da determinação da posição em função do tempo, não se pode esperar que todos os usuários pensassem em calculá-las.

Todavia é omissa quanto a detecção automatizada de outros comportamentos que não deslocamento, tais como levantamento de tronco, autolimpeza e congelamento, de modo que se pretende incluir na plataforma atalhos para anotação destes comportamentos enquanto o usuário assiste ao vídeo com a posição anotada – talvez a única forma de avaliação dos resultados do algoritmo para as condições de realização do teste.

Uma das justificativas de não se incluir outros comportamentos distintos de deslocamento é a dificuldade de defini-los sem experiência prática de realização dos testes, bem como a existência de diversos perfis conforme as espécies ou mesmo entre linhagens dentro delas. Um dos objetivos da ferramenta é ser capaz de ser

aplicada a uma larga variedade de condições, de modo que a anotação através de atalhos é mais democrática e funcional.

Embora a ferramenta tenha sido construída visando ser aplicável a amplo espectro de condições – sobretudo coloração da caixa e animal, o ambiente de execução dos testes é quase controlado. O pesquisador poderia modificar a coloração da caixa a fim de aumentar o contraste entre ela e o animal, e assim localizá-lo segmentando por intensidade a cada frame. Outra modificação no aparato a ser considerada, é utilizar iluminação com difusores para evitar sombras e câmeras que gravem em melhor qualidade.

REFERENCIAS

- 1 HALL, C.; BALLACHEY, E. A study of the rat's behavior in a field: a contribution to method in comparative psychology. **University of California Publications in Psychology**, v. 6, n. 1, p. 1-12, 1932.
- 2 HALL, C. Emotional behavior in the rat. I. defecation and urination as measures of individual differences in emotionality. **Journal of Comparative Psychology**, v. 18, n. 3, p. 385-403, 1934.
- 3 HÅNELL, A.; MARKLUND, N. Structured evaluation of rodent behavioral tests used in drug discovery research. **Frontiers in Behavioral Neuroscience**, v. 8, p. 252, 2014. doi: 10.3389/fnbeh.2014.00252.
- 4 CHOLERIS, E. et al. A detailed ethological analysis of the mouse open field test: effects of diazepam, chlordiazepoxide and an extremely low frequency pulsed magnetic field. **Neuroscience and Biobehavioral Reviews**, v. 25, n. 3, p. 235-260, 2001.
- 5 VALENTINUZZI, V. et al. Locomotor response to an open field during C57BL/6J active and inactive phases: differences dependent on conditions of illumination. **Physiology & Behavior**, v. 69, n. 3, p. 269–275, 2000.
- 6 CAROLA, V. et al. Evaluation of the elevated plus-maze and open-field tests for the assessment of anxiety-related behaviour in inbred mice. **Behavioural Brain Research**, v. 134, n. 1-2, p. 49–57, 2002.
- 7 STANFORD, S. The Open field test: reinventing the wheel. **Journal of Psychopharmacology**, v. 21, n. 2, p. 134-135, 2007.
- 8 BALLARD, D.; BROWN, C. **Computer vision**. Rochester: Prentice Hall, 1982.
- 9 GONZALEZ, R.; WOODS, R. **Processamento digital de imagens**. 3a. ed. São Paulo: Pearson Prentice Hall, 2010.
- 10 BAYKAN, A.; YILMAZ, N.; KANSUN, G. Case study in effects of color spaces for mineral identification. **Scientific Research and Essays**, v. 5, n. 11, p. 1243-1253, 2010. Disponível em: <https://www.researchgate.net/RGB-color-cube_fig1_228677646>. Acesso em: 7 jun. 2018.
- 11 GONZALES, R.; WOODS, R. **Digital image processing**. 2nd ed. New Jersey: Prentice Hall, 2001.
- 12 HARALICK, R.; SHAPIRO, L. Image segmentation techniques. **Computer Vision, Graphics and Image Processing**, v. 29, n. 1, p. 100-132, 1985.
- 13 OTSU, N. A threshold selection method from gray-level histograms. **IEEE Transactions on Systems, Man, and Cybernetics**, v. 9, n. 1, p. 62-66, 1979.

- 14 WANG, Z. et al. Image quality assessment: from error visibility to structural similarity. **IEEE Transactions on Image Processing**, v. 13, n. 4, p. 600-612, 2004.
- 15 TEJADA, J.; CHAIM, K. T.; MORATO, S. X-PloRat: a software for scoring animal behavior in enclosed spaces. **Psicologia: teoria e pesquisa**, v. 33, p. e3322, 2017. doi.org/10.1590/0102.3772e3322.
- 16 NOLDUS, L.; SPINK, A.; TEGELENBOSCH, R. EthoVision: a versatile video tracking system for automation of behavioral experiments. **Behavior Research Methods, Instruments, & Computers**, v. 33, n. 3, p. 398-414, 2001.
- 17 CRISPIM JUNIOR, C. et al. ETHOWATCHER: validation of a tool for behavioral and video-tracking analysis in laboratory animals. **Computers in Biology and Medicine**, v. 42, n. 2, p. 257-264, 2012.
- 18 SAMSON, A. et al. MouseMove: an open source program for semi-automated analysis of movement and cognitive testing in rodents. **Scientific Reports**, v. 5, p. 16171, 2015. doi: 10.1038/srep16171.
- 19 KOKRAS, N. et al. Kinoscope: an open-source computer program for behavioral pharmacologists. **Frontiers in Behavioral Neuroscience**, v. 11, p. 88, 2017. doi: 10.3389/FNBEH.2017.00088.
- 20 PYTHON SOFTWARE FOUNDATION. **The Python Tutorial 2001-2018**. Disponível em: <<https://docs.python.org/3/tutorial/index.html#tutorial-index>>. Acesso em: 08 jun. 2018.
- 21 PYTHON SOFTWARE FOUNDATION. **History and License 1990-2018**. Disponível em: <<https://docs.python.org/3/license.html>>. Acesso em: 07 jun. 2018.
- 22 OPEN SOURCE INITIATIVE. **OSI**. Disponível em: <<https://opensource.org/>>. Acesso em: 07 jun. 2018.
- 23 GITHUB, INC. **Python for NET**. Disponível em: <<https://pythonnet.github.io/>>. Acesso em: 08 jun. 2018.
- 24 OLIPHANT, T. E. **A guide to NumPy**. Birmingham: Packet Publishing, 2006. v. 1.
- 25 OLIPHANT, T. E. SciPy: open source scientific tools for Python. **Computing in Science and Engineering**, v. 9, n. 1, p. 10-20, 2007.
- 26 HUNTER, J. D. Matplotlib: a 2D graphics environment. **Computing in Science & Engineering**, v. 9, n. 3, p. 90-95, 2007.
- 27 HUNTER, J. et al. **Matplotlib**: Python plotting - Matplotlib 2.2.2 documentation. 2012-2018. Disponível em: <<https://matplotlib.org/>>. Acesso em: 18 jun. 2018.
- 28 VAN DER WALT, S. et al. scikit-image: image processing in Python. **PeerJ**, v. 2, n. 453, 2014. doi.org/10.7717/peerj.453.

- 29 BRADSKI, G. K. A. **Learning OpenCV**: computer vision with the OpenCV library. California: O'Reilly Media, Inc., 2008.
- 30 CANNY, J. A Computational approach to edge detection. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 8, n. 6, p. 679-698, 1986.
- 31 CHOW, C.; KANEKO, T. Boundary detection of radiographic images by a threshold method. **Frontiers of Pattern Recognition**, p. 61-82, 1972. doi.org/10.1016/B978-0-12-737140009-90-5.5.
- 32 PRUT, L.; BELZUNG, C. The open field as a paradigm to measure the effects of drugs on anxiety-like behaviors: a review. **European Journal of Pharmacology**, v. 463, n. 1-3, p. 3-33, 2003.

APÊNDICE

Tabela 2 – Número de ocorrências de Entrada em cada Quadrante – comparação entre XploRat e a Ferramenta Desenvolvida

	Vídeo	A1	B1	C1	D1	E1	A2	B2	C2	D2	E2	A3	B3	C3	D3	E3	A4	B4	C4	D4	E4	A5	B5	C5	D5	E5
XploRat	1	5	6	5	4	6	9	7	6	10	11	14	10	10	13	11	11	6	6	9	13	9	11	10	11	10
Ferramenta Proposta	1	3	6	4	7	8	3	2	2	4	7	4	1	3	4	3	4	4	2	4	5	1	4	4	6	4
XploRat	2	5	5	4	5	5	6	4	2	4	8	4	2	2	1	6	7	4	3	4	7	8	7	5	7	6
Ferramenta Proposta	2	5	5	5	6	6	4	2	2	1	4	6	2	1	1	3	6	4	3	0	4	2	2	3	4	4
XploRat	3	12	14	14	15	7	10	6	9	15	7	12	7	9	13	6	12	6	10	13	7	11	15	15	15	8
Ferramenta Proposta	3	3	4	7	7	4	2	2	4	5	8	4	3	5	1	6	11	2	5	10	6	8	8	9	12	6
XploRat	4	5	4	6	6	3	6	2	4	3	3	6	2	2	2	5	5	1	1	0	4	5	5	5	4	4
Ferramenta Proposta	4	5	6	5	6	7	3	0	3	2	7	5	0	3	2	4	8	6	2	5	4	6	4	4	7	5
XploRat	5	5	5	7	9	10	6	3	6	4	10	10	6	8	6	9	10	4	6	8	13	8	9	9	7	8
Ferramenta Proposta	5	4	3	3	5	5	2	2	3	8	7	7	3	1	4	2	7	4	2	11	7	6	5	6	6	6
XploRat	6	6	5	5	6	6	6	0	0	4	8	6	1	3	2	6	6	2	3	2	9	6	6	4	4	7
Ferramenta Proposta	6	6	8	8	10	4	4	3	4	5	4	5	2	2	2	3	10	7	8	6	6	7	6	7	5	3
XploRat	7	5	6	6	7	7	4	4	6	6	10	3	4	4	2	8	4	7	7	9	11	5	8	8	12	10
Ferramenta Proposta	7	8	7	6	7	9	8	5	3	6	8	10	5	2	4	4	9	5	4	7	9	7	4	5	10	4
XploRat	8	6	6	6	9	7	8	5	7	8	10	11	8	7	7	11	10	5	3	8	10	7	7	6	6	5
Ferramenta Proposta	8	10	6	6	10	9	7	2	4	3	4	4	2	1	2	5	9	4	5	3	11	8	4	5	9	9

Fonte: Elaborada pela autora

Tabela 3 – Porcentagem do Tempo total do teste que o animal ficou em cada quadrante

	Vídeo	A1	B1	C1	D1	E1	A2	B2	C2	D2	E2	A3	B3	C3	D3	E3	A4	B4	C4	D4	E4	A5	B5	C5	D5	E5
XploRat	1	4,9	2,2	1,9	0,9	5,5	2,5	2,8	1,4	4,3	5,7	6,0	1,7	2,4	4,2	5,4	4,3	1,8	0,8	2,4	10,0	9,7	2,1	2,2	4,3	10,8
Ferramenta Proposta	1	5,0	6,7	2,3	5,3	16,3	1,7	1,0	0,7	5,7	4,3	4,3	0,7	5,7	2,7	1,7	5,0	6,0	1,3	3,3	4,0	1,0	2,7	5,3	4,3	3,0
XploRat	2	6,4	1,6	1,9	4,2	7,0	6,0	1,2	0,5	1,3	4,4	1,2	0,4	4,4	0,3	5,7	2,6	4,3	3,6	1,9	5,4	19,5	4,1	2,1	3,4	6,5
Ferramenta Proposta	2	19,3	5,7	4,7	2,7	7,0	2,0	3,0	3,3	0,3	3,7	7,7	1,0	4,7	1,7	4,0	6,0	2,7	1,3	0,0	2,3	3,3	2,3	2,7	2,3	6,3
XploRat	3	7,2	4,9	6,9	9,3	9,1	3,4	0,8	2,8	3,8	3,9	3,3	2,6	1,7	1,3	1,3	2,1	1,5	2,0	3,0	2,5	7,0	4,6	4,0	3,9	7,3
Ferramenta Proposta	3	6,0	2,3	6,7	5,3	3,3	0,7	1,0	2,0	2,3	5,0	1,7	2,0	10,7	0,3	3,7	9,7	3,0	1,7	6,3	2,3	5,0	3,7	4,3	7,0	4,0
XploRat	4	20,8	2,2	4,9	6,3	7,0	4,3	2,8	1,4	0,8	1,6	4,1	3,4	4,9	1,1	3,8	1,4	0,3	0,8	0,0	1,3	8,2	4,7	3,4	2,2	8,4
Ferramenta Proposta	4	8,5	5,1	5,1	6,1	8,8	1,0	0,0	4,8	0,7	4,4	3,4	0,0	2,7	1,0	2,0	7,5	4,8	1,0	3,1	2,0	7,8	2,7	6,8	5,4	5,1
XploRat	5	5,7	1,0	2,5	4,8	9,2	2,3	0,5	1,5	2,1	3,8	4,4	1,6	11,7	1,0	4,9	5,1	0,6	1,3	4,9	6,5	6,1	5,1	4,0	1,6	7,9
Ferramenta Proposta	5	19,0	4,0	2,0	3,0	3,3	1,0	1,0	3,7	4,3	6,0	3,7	2,0	2,3	1,3	1,3	6,7	3,7	0,7	6,7	7,3	3,7	3,3	2,7	2,7	4,7
XploRat	6	9,8	0,9	1,8	4,7	12,8	3,1	0,0	0,0	1,2	5,6	2,8	2,4	2,4	1,2	2,0	6,6	1,5	4,0	0,9	9,1	9,2	3,9	2,0	1,6	10,5
Ferramenta Proposta	6	6,3	9,0	6,3	8,3	4,7	1,7	2,7	3,3	2,3	2,3	3,0	0,7	0,7	1,3	1,0	10,7	3,0	4,7	2,3	4,0	7,0	3,7	5,0	2,7	3,3

Fonte: Elaborada pela autora