



UNIVERSIDADE FEDERAL DO PARÁ
INSTITUTO DE TECNOLOGIA
FACULDADE DE ENGENHARIA DA COMPUTAÇÃO E TELECOMUNICAÇÕES

**Caracterização de comportamento visual: implementação e teste de métricas de
similaridade de scanpaths**

Gustavo Damasceno da Silva

BELÉM - PA

2022



UNIVERSIDADE FEDERAL DO PARÁ
INSTITUTO DE TECNOLOGIA
FACULDADE DE ENGENHARIA DA COMPUTAÇÃO E TELECOMUNICAÇÕES

Gustavo Damasceno da Silva

Caracterização de comportamento visual: implementação e teste de métricas de similaridade de scanpaths

Trabalho de Conclusão de Curso apresentado para obtenção do grau de Bacharel em Engenharia da Computação pela Universidade Federal do Pará. Área de concentração: Engenharias.

Orientador: Prof. Dr. Ronaldo de Freitas Zampolo

BELÉM - PA
2022

Gustavo Damasceno da Silva

Caracterização de comportamento visual: implementação e teste de métricas de similaridade de scanpaths

Trabalho de Conclusão de Curso apresentado para obtenção do grau de Bacharel em Engenharia da Computação pela Universidade Federal do Pará. Área de concentração: Engenharias.

Data da Defesa: _____

Conceito: _____

Banca Examinadora

Prof. Dr. Ronaldo de Freitas Zampolo

Faculdade de Engenharia da Computação e Telecomunicações - UFPA
Orientador

Prof. Dr. Agostinho Luiz da Silva Castro

Faculdade de Engenharia da Computação e Telecomunicações - UFPA
Membro da Banca

**Prof. Dr. Francisco Carlos Bentes Frey
Müller**

Faculdade de Engenharia da Computação e Telecomunicações - UFPA
Membro da Banca

Belém - PA
2022

AGRADECIMENTOS

A meu pai e minha mãe, dedico este trabalho como símbolo da conclusão de uma etapa que teve início através de confiança e incentivo. Agradeço à Amanda, minha namorada e melhor amiga ao longo dos últimos 7 anos e, sobretudo, ao longo destes 5 anos de graduação. Agradeço à dona Carmen e à tia Vânia por me acolherem todos os domingos e terem sido parte da minha segunda família. Por último gostaria de retribuir através de agradecimentos minha amiga Giulia, por acreditar no meu esforço e me apoiar em todos os momentos.

“I have spoken.”

(*Kuil, from The Mandalorian*)

RESUMO

Técnicas de rastreamento ocular, do inglês *eye-tracking*, permitem o estudo da atenção visual de um observador, mediante identificação de áreas de atenção, tempo de fixação em determinada região da cena observada e a sequência de exploração visual. A representação em *scanpaths* permite a ordenação de sacadas e fixações (movimentos oculares pertinentes em processos de rastreamento ocular), facilitando a compreensão do deslocamento do olhar do participante e permitindo inferências a respeito da sequência de processos cognitivos empregados. Utilizar o processo de *eye-tracking* na medicina pode aprimorar o entendimento sobre rotinas visuais de médicos e estudantes de medicina com diferentes níveis de experiência durante a análise de exames como ultrassonografia e eletroencefalograma. Neste trabalho é feita uma introdução aos tipos de movimento ocular de maior interesse para caracterização de comportamento visual e aos métodos de como identifica-los com dados obtidos por rastreadores oculares. Além disso, é apresentado o conceito de *scanpath* e são avaliadas duas métricas de comparação de *scanpaths*. Resultados experimentais sugerem que o algoritmo *ScanMatch* é superior ao *Edit distance* ao computar os relacionamentos existentes entre as regiões visualizadas e a atenção do observador. Dessa forma, são discutidos elementos que subsidiam a aplicação do rastreamento ocular à caracterização do comportamento visual aplicada à análise de imagens médicas, motivação original do trabalho.

Palavras-chave: Rastreamento ocular, *scanpaths*, imagens médicas, *ScanMatch*, *Edit distance*.

ABSTRACT

Eye-tracking techniques allows the study of an observer's visual attention, by identifying areas of attention, fixation time in a certain region of the observed scene and the sequence of visual exploration. The representation in scanpaths allows the ordering of saccades and fixations (eye movements relevant in eye tracking processes), facilitating the understanding of the participant's gaze displacement and allowing inferences about their sequence of cognitive processes . Using the eye tracking process in medicine is possible to improve the understanding of visual routines of physicians and medical students with different levels of experience when analyzing tests such as ultrasound and electroencephalogram. In this work, an introduction is made to the types of eye movement of greatest interest for characterization of visual behavior and the methods of how to identify them with data obtained by eye trackers. Furthermore, the scanpath concept and two scanpath comparison metrics are evaluated. Experimental results suggest that the ScanMatch algorithm is superior to Edit distance when computing the existing relationships between the visualized regions and the observer's attention. Thus, elements that support the application of eye tracking to the characterization of visual behavior applied to the analysis of medical images, the original motivation for the work, are discussed.

Keywords: Eye-tracking, scanpaths, medical images, ScanMatch, Edit distance

LISTA DE ILUSTRAÇÕES

Figura 1 – Imagem do olho com destaque da pupila e do reflexo da córnea	6
Figura 2 – Posições relativas da pupila e imagens de Purkinje vistas pela câmera de um rastreador ocular	7
Figura 3 – Opções de personalização durante a síntese de um estudo no RealEye	10
Figura 4 – Opções de personalização para exibição de cada imagem	11
Figura 5 – Detecção de face e pupila para execução da calibragem	12
Figura 6 – Telas durante processo de calibragem. Em (a) tela branca, em (b) tela preta e em (c) tela cinza	13
Figura 7 – Fase final do processo de calibragem	13
Figura 8 – Mapa de calor de rastreamento ocular.	14
Figura 9 – Menu de preferências para a exibição do mapa de calor	15
Figura 10 – Gráfico em função do tempo exibindo coordenadas e velocidades antes da retirada de ruído	15
Figura 11 – Gráfico em função do tempo exibindo fixações, coordenadas e velocidades após a retirada de ruído	16
Figura 12 – Ângulos horizontal θ_1 e vertical θ_2 , a partir do observador O até o centro do monitor (representado pelo retângulo)	18
Figura 13 – Disposição da distância D entre o observador e a tela de dimensão T	19
Figura 14 – Disposição dos pontos descritos na Tabela 1	20
Figura 15 – Representação de fixações pós I-VT	22
Figura 16 – Identificação de fixações com I-VT. Representação em duas dimensões em que os pontos maiores representam as fixações, enquanto os pontos menores representam as sacadas.	23
Figura 17 – Fluxograma do agrupamento de fixações durante a filtragem	25
Figura 18 – Fluxograma da eliminação de fixações durante a filtragem	25
Figura 19 – Exemplo de grade para conversão de scanpaths em strings	27
Figura 20 – Exemplo de comparação entre strings para obter dissimilaridade	27
Figura 21 – Transformação de fixações para string considerando-se uma variação temporal de referência de 50 ms. Ao lado, strings obtidas segundo o método de edição de string e segundo o método ScanMatch, respectivamente	30

Figura 22 – Exemplos de alinhamento entre strings com a aplicação de penalidades de gap e de uma matriz de substituição para guiar as pontuações de alinhamento	31
Figura 23 – Matriz C (disposição de pares no alinhamento)	34
Figura 24 – Imagem utilizada de estímulo visual	37
Figura 25 – Disposição do estímulo visual ao centro de uma tela de 1920 pixels por 1080 pixels	37
Figura 26 – <i>Scanpaths</i> dos observadores 1 (a), 2 (b), 3 (c), 4 (d) e do observador de referência (e). Os círculos vermelhos representam as fixações, seus números denotam a ordem de observação e as letras denominam as células na grade.	39
Figura 27 – Tempo gasto em cada célula durante as fixações para os observadores 1 (a), 2 (b), 3 (c), 4 (d) e para observador de referência (e). Os círculos vermelhos representam as fixações, seus números denotam a ordem de observação e as letras denominam as RoIs	40
Figura 28 – Exemplo de função de janelamento	50
Figura 29 – Janelamento da mesma slice, com centros em 40 HU e comprimentos de: 80 HU em (a) e de 40 HU em (b)	51
Figura 30 – Fluxograma do tratamento de imagem descrito neste capítulo	51

LISTA DE TABELAS

Tabela 1 – Correspondência entre índices obtidos pelo RealEye, eixo cartesiano e índices em pixels (em uma imagem 1920x1080 px)	20
Tabela 2 – Sequências arbitrárias em alinhamento. Cada sequência representa um <i>scanpath</i>	32
Tabela 3 – Contagem de pares por coluna. Cada coluna representa a respectiva posição no <i>scanpath</i> de origem	33
Tabela 4 – Contagem total de pares	33
Tabela 5 – Tamanho das strings (sem expansão) relacionadas ao <i>scanpath</i> de cada observador	41
Tabela 6 – Pontuação de similaridade entre <i>strings</i> através do método <i>Edit distance</i> (sem replicação de caracteres)	41
Tabela 7 – Tamanho das strings (com expansão) relacionadas ao <i>scanpath</i> de cada observador	42
Tabela 8 – Pontuação de similaridade entre <i>strings</i> através do método <i>Edit distance</i> (com replicação de caracteres)	42
Tabela 9 – Alinhamentos presentes para a geração da matriz BLOSUM	43
Tabela 10 – Pontuação de similaridade entre strings através do método <i>ScanMatch</i>	43
Tabela 11 – Valor Hounsfield (HU) de substâncias comuns	49

LISTA DE ABREVIATURAS E SIGLAS

PoR	<i>Point of Regard</i>
I-VT	<i>Velocity-Threshold Identification</i>
RoI	<i>Region of Interest</i>

SUMÁRIO

1	INTRODUÇÃO	1
1.1	Contexto	1
1.2	Justificativa	1
1.3	Objetivos	2
1.3.1	Objetivo Geral	2
1.3.2	Objetivos Específicos	2
1.4	Metodologia	2
1.5	Estrutura do trabalho	3
2	MOVIMENTOS OCULARES E REALEYE	4
2.1	Introdução	4
2.2	Movimentos oculares	4
2.2.1	Sacadas	4
2.2.2	Fixações	5
2.3	Eye-tracking e análise de reflexão da pupila e da córnea	5
2.4	RealEye	7
2.4.1	Parâmetros de estudo	8
2.4.2	Calibragem	12
2.4.3	Recursos pós-experimento e disposição de dados	13
2.4.4	RealEye e limitações de projeto	16
3	DETECÇÃO DE FIXAÇÕES	18
3.1	Introdução	18
3.2	Ângulo visual e variação angular	18
3.3	Identificação de limite de velocidade (I-VT)	21
3.4	Filtragem de fixações	23
4	SCANPATHS E ALGORITMOS DE ANÁLISE	26
4.1	Introdução	26
4.2	Edit distance	26
4.3	ScanMatch	28
4.3.1	Matrix BLOSUM	32
5	ARRANJO EXPERIMENTAL E RESULTADOS	36
5.1	Introdução	36

5.2	Hipótese	36
5.3	Estímulo visual e parâmetros de exibição	37
5.4	Participantes	38
5.5	Tratamento e disposição dos dados obtidos	38
5.6	Estratégias e resultados de comparação	40
5.6.1	Aplicação de <i>Edit distance</i>	40
5.6.1.1	<i>Edit distance</i> sem replicação de caracteres	41
5.6.1.2	<i>Edit distance</i> com replicação de caracteres	41
5.6.2	<i>ScanMatch</i>	42
6	CONCLUSÃO	44
6.1	Considerações Finais	44
6.2	Trabalhos futuros	45
	REFERÊNCIAS	46
A	- IMAGENS MÉDICAS	48
A.1	Introdução	48
A.2	Imagens de tomografia computadorizada	48
A.3	Tomografia computadorizada e unidades Hounsfield	49

1 INTRODUÇÃO

1.1 Contexto

Sistemas de *eye-tracking* permitem caracterizar estratégias visuais a partir do comportamento ocular de um observador utilizando a coleta dos pontos de vista (PoR, do inglês *Point of Regard*). Desta forma, é possível registrar os PoRs durante múltiplos processos de visualização com observadores distintos e posteriormente compará-los através de algoritmos que avaliem o grau de semelhança entre as estratégias visuais adotadas. Com o intuito de oferecer novos instrumentos para aprimorar o treinamento de estudantes e profissionais da medicina na interpretação de imagens médicas, este trabalho propõe um sistema de coleta e análise de PoRs para comparar rotinas visuais através de diferentes métricas de similaridade.

1.2 Justificativa

A modelagem de comportamentos visuais quando da análise de imagens médicas, seja por especialistas ou profissionais em treinamento, representa uma abordagem promissora de ensino e consolidação da aprendizagem na área de diagnósticos por imagem. Nesse sentido, deve-se avaliar o desempenho de observadores em treinamento com base em estratégias de visualização adotadas por observadores experientes em contextos de diagnose via imagem.

Através do progresso científico é possível aprimorar o exercício da medicina. Tal contexto abre portas para a aplicação das ciências computacionais em prol de diagnósticos mais precisos e tratamentos mais adequados. A experiência de um profissional da medicina é essencial na avaliação de exames provenientes de áreas como a radiologia, que envolvem o diagnóstico a partir das leituras visuais de exames. Portanto, é relevante associar a experiência médica ao *eye-tracking* com o objetivo de entender e avaliar padrões de visualização para aperfeiçoar o aprendizado de médicos menos experientes ou em formação. Trata-se de um processo complexo de caracterização de estratégias visuais, onde métricas de similaridade sobre a sucessão de movimentos oculares e as regiões visualizadas por diferentes pessoas possuem um papel relevante.

1.3 Objetivos

1.3.1 Objetivo Geral

- Desenvolver um experimento para capturar PoRs e implementar funções para a análise de *scanpaths*.

1.3.2 Objetivos Específicos

- Identificar e analisar eventos a partir de PoRs para obter *scanpaths*.
- Implementar métricas para a comparação entre *scanpaths*.
- Estabelecer parâmetros para um experimento de visualização de imagens médicas.

1.4 Metodologia

A elaboração deste trabalho iniciou-se com pesquisa bibliográfica em artigos envolvendo manipulação de imagens DICOM, *eye-tracking* e algoritmos de comparação entre *scanpaths*, além da leitura do livro *Eye Tracking Methodology. Theory and Practice* escrito por Andrew T. Duchowski.

Em seguida foram formulados códigos em *Python* para realizar o tratamento de imagens médicas e tratar dados provenientes do *eye-tracking* para gerar *scanpaths*.

Fez-se necessário selecionar, através do estudo bibliográfico, métricas de similaridade de *scanpaths* visando sua implementação e avaliação posteriores.

O experimento de testes foi concebido levando-se em consideração a possibilidade de escala do experimento em relação ao número de participantes, número de imagens a serem exibidas e parâmetros de observação a serem aplicados.

Após múltiplas execuções do experimento e execução dos códigos, os resultados das comparações entre *scanpaths* foram reunidos e discutidos, levando em consideração as métricas de similaridade utilizadas.

1.5 Estrutura do trabalho

Além deste capítulo introdutório, o trabalho foi dividido em mais 5 capítulos, lista de referência e anexo, descritos como se segue.

O Capítulo 2 descreve os princípios, as características e os métodos de funcionamento de algumas tecnologias voltadas ao propósito de *eye-tracking*.

O Capítulo 3 aborda a detecção de fixações oculares a partir dos dados adquiridos de um dispositivo de *eye-traking*, com foco no método de análise angular.

O Capítulo 4 conceitua *scanpath* e duas métricas de similaridade usadas na comparação de *scanpaths*

O Capítulo 5 descreve os experimentos de teste, detalhando problemáticas e resultados.

O Capítulo 6 apresenta uma conclusão geral dos resultados obtidos e apresenta propostas para trabalhos futuros de continuidade.

2 MOVIMENTOS OCULARES E REALEYE

2.1 Introdução

O avanço da tecnologia possibilitou o aprimoramento de técnicas de *eye-tracking* em relação à aquisição e análise de dados. Este capítulo aborda os bases do rastreamento ocular: técnicas mais comuns, parâmetros de análise dos dados adquiridos e problemáticas pertinentes ao estabelecimento de uma pesquisa, como seleção de estímulos e participantes, tempo de visualização e atribuição de tarefas.

2.2 Movimentos oculares

Quase todos os movimentos oculares normais utilizados para reposicionar a fóvea (região central da retina) em primatas podem ser classificados em cinco tipos básicos: sacada, perseguição suave, vergência, vestibular e nistagmo fisiológico (movimentos em miniatura associados a fixações) (ROBINSON, 1968 apud DUCHOWSKI, 2007).

Para introduzir os principais métodos de monitoramento ocular considerando a observação de imagens estáticas, serão abordados dois movimentos primordiais à sua análise: sacadas e fixações.

2.2.1 Sacadas

Sacadas são movimentos oculares rápidos usados para reposicionar a fóvea (região central da retina) em um novo local no ambiente visual (GREGORY, 1997 apud DUCHOWSKI, 2007). Estes movimentos duram, em média, de 10 a 100 milissegundos.

Sacadas possuem duas características principais: são balísticas e estereotipadas. O termo balístico diz respeito à presunção de que os destinos de sacada são pré-programados de forma que, uma vez que o movimento sacádico para o próximo local de fixação desejado tenha sido calculado, as sacadas não podem ser alteradas. O termo estereotipado, por sua vez refere-se à observação de que padrões de movimento específicos podem ser evocados repetidamente. Tais presunções sustentam-se pois, durante a execução da sacada, não há tempo suficiente para o *feedback* visual guiar o olho até sua posição final. (LAURUTIS; ROBINSON, 1986 apud DUCHOWSKI, 2007).

2.2.2 Fixações

Fixações podem ser descritas como os movimentos oculares que estabilizam a retina sobre um objeto estacionário de interesse e são caracterizadas por movimentos oculares em miniatura: tremores, desvios e micro sacadas (variando em 1–2 min de arco em amplitude). Estes movimentos oculares em miniatura podem ser considerados ruídos presentes no sistema de controle tentando manter o olhar firme. Este ruído aparece como uma flutuação aleatória sobre a área de fixação, normalmente não maior do que 5° de ângulo visual. Desta forma, como o nome sugere, fixações estão relacionadas ao interesse visual em um determinado instante (DUCHOWSKI, 2007).

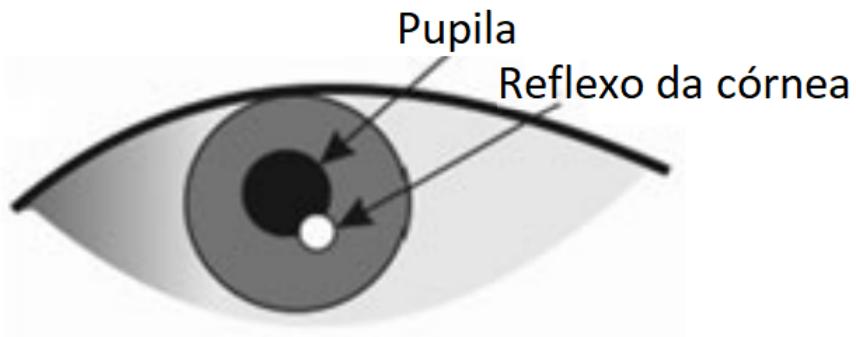
2.3 Eye-tracking e análise de reflexão da pupila e da córnea

Monitorar o olhar através da orientação do olho no espaço, ou seja, do ponto de vista, representa uma das principais técnicas de rastreio ocular.

Métodos de análise da reflexão de pupila e córnea baseados em vídeo são técnicas não invasivas que não oferecem medições diretas do ponto de vista. Entretanto, através da fixação da posição da cabeça é possível avaliar a posição do olho em relação à cabeça de modo a estimar o ponto de vista. Também é possível utilizar processamento computacional para contornar ruídos na movimentação da cabeça e compensá-los.

Existem duas estratégias fundamentais para técnicas desta natureza: reflexão da córnea e o centro da pupila. Geralmente tais estratégias são realizadas com auxílio de uma fonte luminosa e câmeras na faixa do infravermelho (Figura 1) mas é possível utilizar iluminação ambiente (lâmpadas de led) e câmeras de baixo custo para diminuir o orçamento de um projeto.

Figura 1 – Imagem do olho com destaque da pupila e do reflexo da córnea



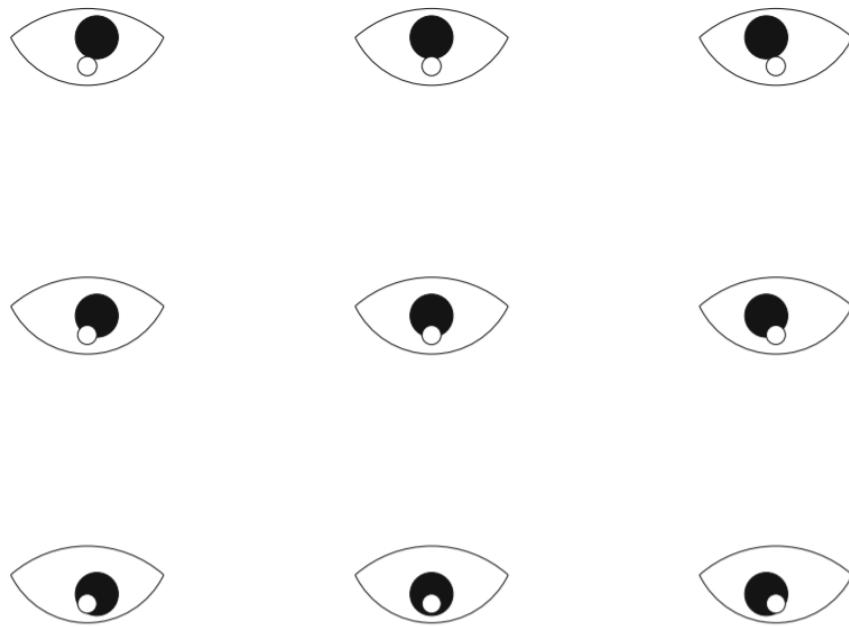
Fonte: Feng e Cai, 2017. Adaptado.

A distância entre o reflexo da córnea, também chamado de imagem de Purkinje, e o centro da pupila é utilizada para estimar a direção do olhar em dado momento (DUCHOWSKI, 2007). Cabe ressaltar que a estrutura do olho implica na formulação de quatro imagens de Purkinje e que o *eye-tracker* baseado em vídeo utiliza apenas a primeira.

Outro critério primordial à acurácia de *eye-trackers* baseados em vídeo diz à posição do plano de visualização: melhores resultados são obtidos se o plano de visualização permanecer paralelo ao rosto.

Usualmente, a calibração de um *eye-tracker* é realizada com a apresentação de pontos estáticos em posições conhecidas na tela. Durante a visualização dos pontos para calibração, a posição da pupila e do reflexo da córnea se modificam, adotando padrões adotados padronizados variáveis (Figura 5) que associados às posições de calibração pré-definidas, permitem a configuração do sistema para estimar PoRs para uma dada região de visualização.

Figura 2 – Posições relativas da pupila e imagens de Purkinje vistas pela câmera de um rastreador ocular



Fonte: Duchowski, 2007.

2.4 *RealEye*

Atualmente, as técnicas mais difundidas em rastreadores oculares fazem uso de fontes de luz e câmeras que operam no espectro infravermelho (FERHAT; VILARIÑO, 2016).

Uma importante proposta empregada por rastreadores oculares atuais é investir em tecnologias mais escalonáveis, como plataformas web com uso de câmeras típicas de notebooks ou computadores. Neste contexto, a experimentação online vem surgindo em muitas áreas da psicologia cognitiva como uma alternativa viável ou suplemento à experimentação clássica em laboratório (SEMMELMANN; WEIGELT, 2017).

Através de rastreamento do olhar por meio de luz visível, a necessidade de um hardware especial é parcialmente contornada, abrindo portas para o estabelecimento de monitoramento ocular como um serviço web (FERHAT; VILARIÑO, 2016).

Ainda que tais avanços não resolvam o problema de baixa precisão devido a condições ambientais imprevisíveis, é possível registrar dados com precisão suficiente, a depender da aplicação, para serem pós-processados através de algoritmos de análise.

Este contexto é a base que fomentou o desenvolvimento de diversas tecnologias para monitoramento ocular, entre elas o *RealEye* (<https://www.realeye.io>): uma plataforma online projetada para conduzir pesquisas de rastreamento ocular de *webcam* com base na tela. Nesta plataforma a criação dos estudos, a coleta dos resultados e a análise dos dados são feitas no navegador da web através de uma biblioteca de rastreamento ocular escrita em *JavaScript* chamada *WebGazer*. Esta biblioteca monitora o olhar através da localização do olho e, em seguida, da pupila, que é encontrada através de três suposições: a cor da íris é mais escura em relação a cor da área circundante, a pupila é circular e por fim está localizada no centro do globo ocular.

O *RealEye* usa o poder de computação de um PC / laptop comum para rodar uma rede neural que analisa as imagens provenientes de uma *webcam*. A inteligência artificial através da rede neural detecta o rosto do usuário, suas pupilas e prevê um PoR. Tudo é realizado inteiramente em um navegador web em tempo real.

O software do sistema *RealEye* não necessita de downloads e qualquer estudo pode ser conduzido remotamente. Os testes são feitos nos *laptops* ou PCs dos participantes usando suas *webcams* (não pode ser feito em dispositivos móveis). Para participar do estudo é necessário um computador com *Microsoft Windows* 10 ou *Mac* (Macbook, iMac) com *MacOS* com *webcam* integrada ou *webcam* USB. A resolução de tela exigida é de 1024x968 pixels ou maior. Apenas orientações de tela horizontais são suportadas. Além disso, a página não pode ser ampliada.

2.4.1 Parâmetros de estudo

Segundo o site oficial do *RealEye* (<https://www.realeye.io/>), existem fatores fundamentais para que um estudo seja executado de maneira correta:

- A *webcam* deve ser montada sobre tela que exibe os estímulos;
- A imagem da *webcam* deve ser estável;
- O participante deve sentar-se em frente à *webcam*, que deve ficar um pouco abaixo do nível dos olhos;
- O rosto do participante deve estar bem iluminado;
- A posição da fonte de luz não pode mudar durante todo o teste;

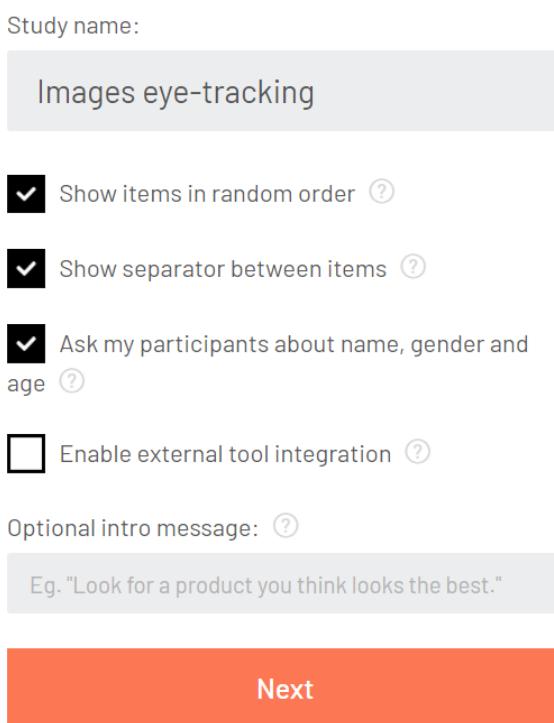
- Deve haver apenas um rosto capturado pela *webcam*;
- Os olhos do participante devem estar claramente visíveis na imagem da *webcam*;
- O tamanho do rosto do participante deve corresponder a pelo menos 1/3 da imagem da *webcam*;
- Recomenda-se que o participante não use óculos, se possível. Caso não seja possível evitar os óculos, estes não podem limitar a visibilidade das pupilas pela *webcam*;
- O participante deve sentar-se confortavelmente e deve tentar manter a cabeça imóvel;
- É altamente recomendável o uso de cadeira com apoio de cabeça - o participante deve descansar a cabeça. Caso contrário, o participante pode apoiar a cabeça no braço;
- O teste deve ser executado em uma *webcam* que forneça boa qualidade de imagem: resolução de pelo menos 640x480. Recomenda-se uma *webcam* de pelo menos 720p a 30 FPS;
- O teste deve ser executado no navegador *Google Chrome*;
- O navegador da Web deve ter apenas uma guia / janela com o teste de rastreamento ocular para garantir que a maior parte da capacidade da CPU esteja disponível para o IA de rastreamento ocular;
- Pelo menos 4 GB de RAM e processadores Intel i5 (mínimo de 2 núcleos, clock de 3 GHz ou superior).
- Se o estudo ocorre em laptop, ele deve ser conectado ao adaptador de energia para garantir que a energia total da CPU esteja disponível;
- A velocidade mínima da Internet deve ser de 2 Mb / s;
- Todos os tipos de aplicativos e notificações do sistema devem ser desativados durante o teste. Recomenda-se ativar o recurso do Windows denominado "Assistente de foco" ou semelhante no Mac OS;

Os dados são capturados com uma taxa de amostragem de cerca de 30 Hz. Esta taxa pode ser aumentada para 60 Hz através de *webcams* melhores, porém este trabalho manterá sua execução através de *webcams* de 30 Hz.

Nenhuma imagem de uma *webcam* é capturada ou transmitida, pois a *webcam* é utilizada apenas para calcular as posições dos olhos que são armazenadas na forma de dados de texto básicos.

Em relação aos estímulos, o *RealEye* possibilita três opções: é possível implementar imagens, vídeos e websites. Para imagens, os testes podem contar até 100 imagens por estudo. Em relação aos vídeos, é possível aplicá-los em sequência, com visualizações de até 10 minutos acompanhando interações livres pela página. Em relação a imagens, pode-se optar por exibições randômicas de imagens separadas por um separador (uma tela de transição preenchida por uma cor sólida) ou pela solicitação de informações (nome, sexo, idade) ao participante (Figura 3).

Figura 3 – Opções de personalização durante a síntese de um estudo no RealEye



Fonte: www.realeye.io.

A função do separador é permitir que os olhos do participante possam relaxar entre as exibições dos estímulos.

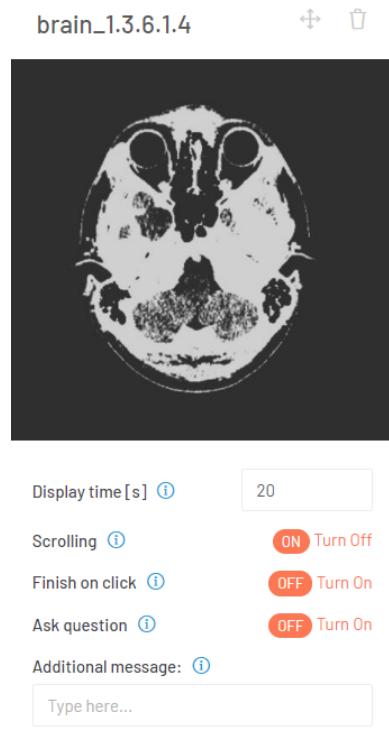
Também é possível exibir mensagens de introdução após a calibragem e antes da exibição de estímulos.

Especificamente sobre o uso de imagens como estímulos, um único estudo pode contar com até 100 imagens. São aceitos os formatos JPEG e PNG, com tamanho máximo de 30 MB.

A resolução mínima de cada imagem é de 500 x 500 e a resolução máxima é de 15.000 x 15.000.

Para cada imagem pode-se definir opções de exibição, tais como tempo de exibição, função de *scrolling*, finalização por click, habilitação de perguntas após a exibição e mensagens a serem exibidas antes da exibição (Figura 4).

Figura 4 – Opções de personalização para exibição de cada imagem



Fonte: www.realeye.io.

Cada estudo dispõe de um número máximo de participantes que depende do plano de assinatura *RealEye* contratado. O plano de teste, por exemplo, conta com um máximo de 5 participantes por experimento.

Após configurar o estudo, um link é fornecido e pode ser distribuído entre os participantes escolhidos. Através do link, o participante é conduzido ao estudo para realizar a detecção de face e pupila (Figura 5).

Figura 5 – Detecção de face e pupila para execução da calibragem

Sua face está dentro do quadro verde?



Ajuda

Sim

Credits: WebGazer.js (LGPLv3), TensorFlow.js (Apache 2.0).

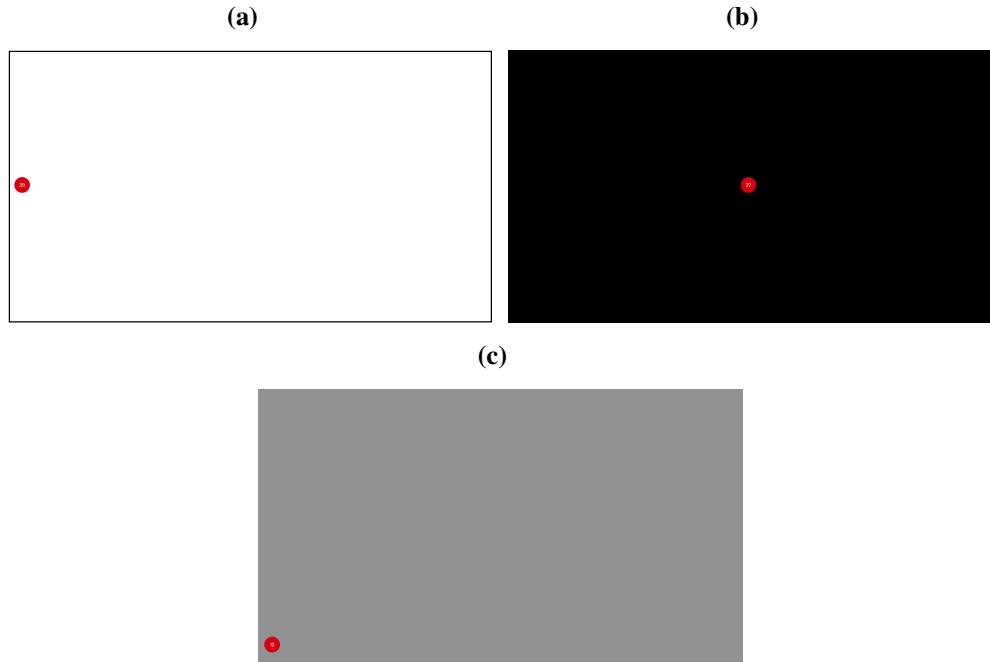
Fonte: Autor.

2.4.2 Calibragem

Após a detecção de face e de pupila, as instruções básicas são exibidas e a calibragem de 40 pontos começa. Trata-se de um processo em tela cheia, em que ocorre exibição alternada de pontos em vermelho. É conduzida em 3 fundos diferentes - branco, preto e cinza (Figura 6) - para minimizar a influência da intensidade da luz do monitor durante o teste de rastreamento ocular.

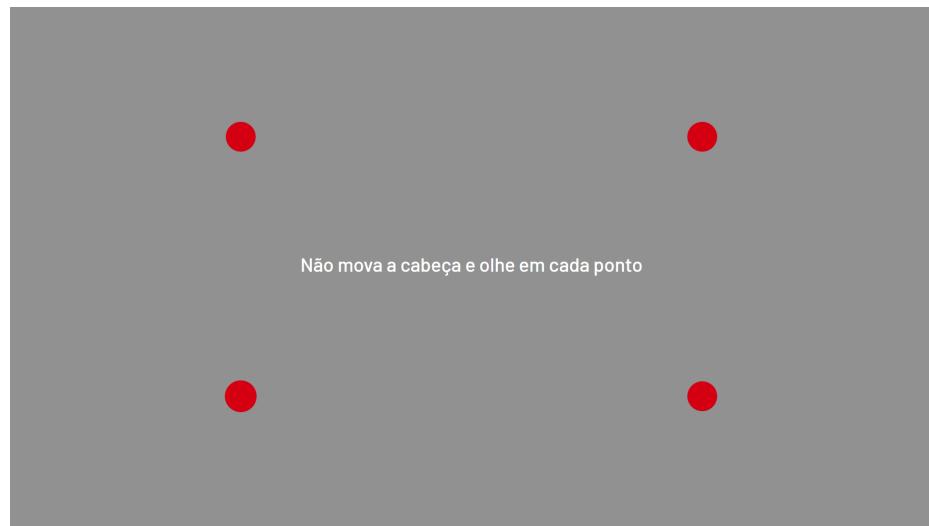
A fase final de calibragem ocorre com a exibição de quatro pontos vermelhos, como mostra a Figura 7, que devem ser focados pelo participante em qualquer ordem até que sumam da tela.

Figura 6 – Telas durante processo de calibragem. Em (a) tela branca, em (b) tela preta e em (c) tela cinza



Fonte: Autor.

Figura 7 – Fase final do processo de calibragem



Fonte: www.realeye.io.

2.4.3 Recursos pós-experimento e disposição de dados

Após a finalização do estudo, é possível obter acesso aos dados em um quadro específico.

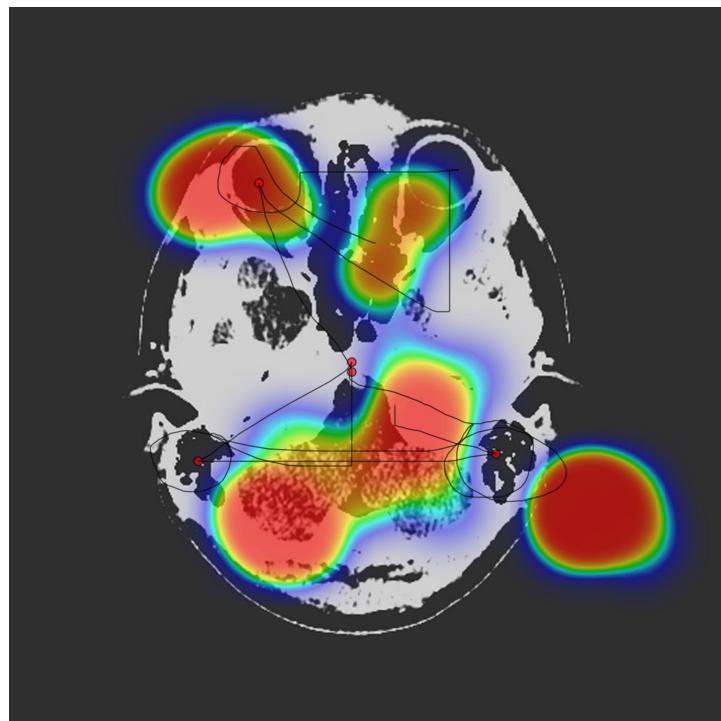
Os dados podem ser dispostos e visualizados online de duas formas: através de mapas de

calor e através de gráficos em função do tempo.

Mapas de calor, do inglês *heatmap*, são relatórios que utilizam as cores como referência para facilitar o entendimento das informações de forma que, lugares com maior densidade de PoRs são representados por cores quentes, geralmente o amarelo, o laranja e o vermelho enquanto lugares com menor densidade de PoRs são representados por cores mais frias, como o azul.

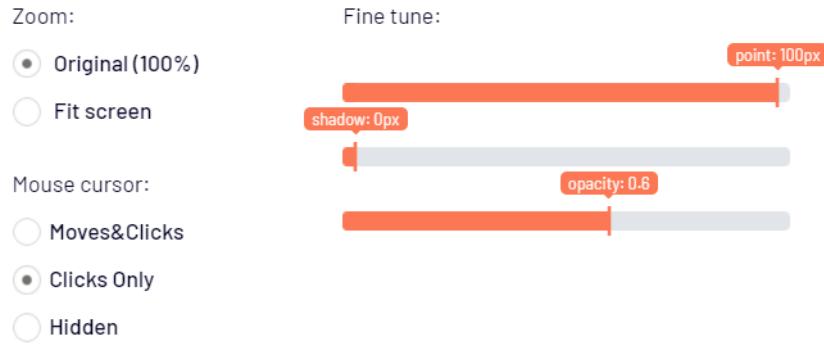
Na Figura 8 é possível observar regiões sobrepostas por círculos, cuja cor varia de acordo com o quanto visualmente interessante a região pareceu ao observador (quanto mais quente a cor do círculo, mais visualmente interessante a região). Além disso, linhas dispostas sobre a imagem representam a movimentação do cursor, com utilização do mouse, durante a execução do experimento, enquanto pontos vermelhos menores, envoltos por uma circunferência preta, exibem as coordenadas onde o usuário clicou.

Figura 8 – Mapa de calor de rastreamento ocular.



Fonte: Autor.

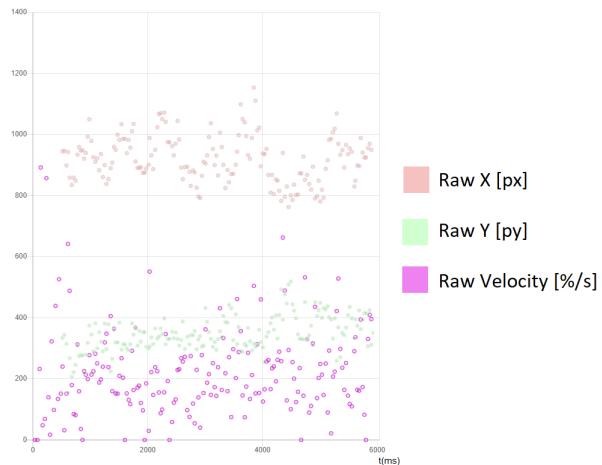
Segundo o menu presente na Figura 9, também é possível exibir o mapa de calor sem as linhas ou os cliques relacionadas ao mouse:

Figura 9 – Menu de preferências para a exibição do mapa de calor

Fonte: Autor.

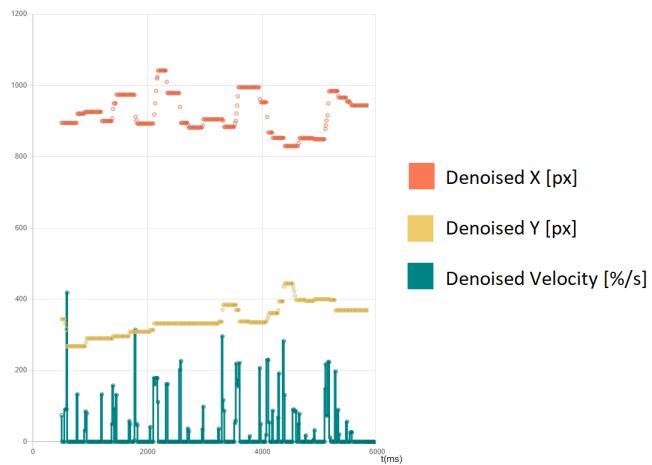
Cabe destacar que a Figura 9 demonstra erros perceptíveis em termos de calibragem pois são resultados de um monitoramento ocular em condições adversas, realizado apenas com o objetivo de demonstrar como é exibido o mapa de calor.

De acordo com as Figuras 10 e 11, através da análise dos dados em função do tempo é possível observar parâmetros específicos como velocidade e coordenadas (x e y) antes e depois de filtragem para redução de ruído. Também é possível observar fixações dispostas ao longo do tempo. Ao sobrepor o cursor do mouse sobre qualquer ponto disposto nos gráficos é possível observar o tempo em milissegundos e o parâmetro disposto.

Figura 10 – Gráfico em função do tempo exibindo coordenadas e velocidades antes da retirada de ruído

Fonte: Autor.

Figura 11 – Gráfico em função do tempo exibindo fixações, coordenadas e velocidades após a retirada de ruído



Fonte: Autor.

É possível realizar download dos dados obtidos durante o monitoramento ocular. Os dados são armazenados em um mesmo arquivo CSV (*comma-separated-values*) e diferenciados através de um cabeçalho sempre que uma nova rotina de observação é iniciada. Existem algumas opções de disposição dos dados: é possível optar por um arquivo com as fixações ou por um arquivo apenas com os PoRs.

2.4.4 RealEye e limitações de projeto

O sistema *RealEye* possui um ambiente complexo e completo, capaz de oferecer diversos recursos para amparar a síntese de um experimento e facilitar a compreensão e o processamento dos dados. Entretanto, trata-se de uma plataforma por assinatura, com planos custosos e que portanto não são acessíveis ao estabelecimento de um estudo escalonável sem um orçamento maior à disposição.

A proposta inicial deste trabalho de conclusão de curso em termos e monitoramento ocular era baseada na disponibilidade de um *eye-tracker* tradicional de baixo custo inicialmente projetado para jogos. Entretanto, por imprevistos técnicos que envolvem a política da empresa fornecedora do *eye-tracker*, não seria possível obter os dados crus obtidos. Além disso, a pandemia do novo coronavírus (COVID-19) tornou-se um impedimento ao recrutamento de participantes para realizarem um experimento em laboratório, levando à escolha de uma técnica de monitoramento ocular que pudesse ocorrer à distância, respeitando o isolamento social.

Ainda que ofereça a opção de tratar os dados, resultando apenas nas fixações de cada usuário, um dos objetivos deste trabalho é desenvolver o próprios algoritmo de detecção de fixações. Desta forma, optou-se por utilizar apenas os dados pré-processados sem ruído e explicar o processamento para detecção de fixações nos capítulos seguintes.

Em relação à exibição de um estímulo estático, ou seja de uma imagem, é necessário frisar que o sistema *RealEye* tenta otimizá-lo através de um zoom com o objetivo de permitir que imagens menores que as dimensões da tela sejam exibidas com aproveitamento máximo. Como é primordial manter o controle sobre qualquer alteração na imagem exibida, optou-se por criar uma imagem com as mesmas dimensões do monitor: o estímulo desejado é formulado pela sobreposição da imagem de interesse, com suas dimensões originais, ao centro de um fundo que tenha da mesma cor do fundo da imagem e que tenha dimensões iguais às do monitor.

3 DETECÇÃO DE FIXAÇÕES

3.1 Introdução

O processo de classificação básica de PoRs em sacadas e fixações é uma parte essencial da análise de movimentos oculares (SALVUCCI; GOLDBERG, 2000).

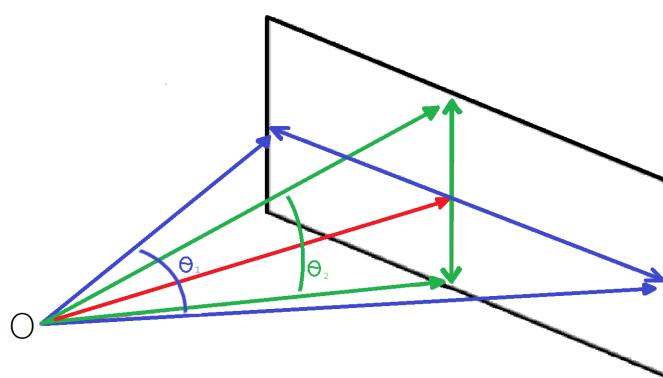
Dentre os diversos algoritmos de detecção de fixações, optou-se por uma técnica baseada na análise de velocidade angular. Não se trata do algoritmo mais robusto proposto até hoje, porém fornece uma análise suficientemente acurada para o sistema proposto neste trabalho visto que a identificação de sacadas e fixações baseada na velocidade ponto a ponto é o método mais simples de implementar (STUART et al., 2014).

3.2 Ângulo visual e variação angular

Para distinguir sacadas de fixações é necessário analisar velocidades angulares relacionadas aos PoRs obtidos após o monitoramento ocular. Para concretizar esta análise é primordial obter as proporções do monitor (originalmente em pixels) em ângulos visuais.

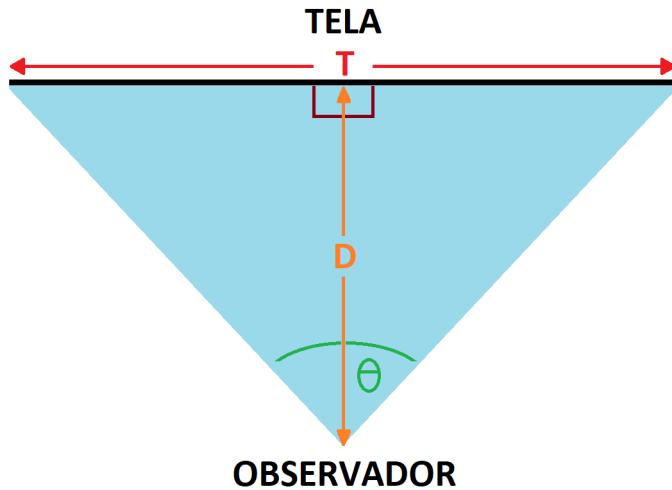
Para tanto, utiliza-se de referência o centro geométrico da tela, conforme a Figura 12.

Figura 12 – Ângulos horizontal θ_1 e vertical θ_2 , a partir do observador O até o centro do monitor (representado pelo retângulo)



Fonte: Autor.

Tomaremos como exemplo a tela que representará o padrão para o estudo: resolução de 1920 x 1080 px com 35 centímetros de comprimento e 20 centímetros de altura a uma distância de 35 centímetros do rosto do participante.

Figura 13 – Disposição da distância D entre o observador e a tela de dimensão T

Fonte: Autor.

Para calcular a correspondência angular da tela é necessário analisar o tamanho de cada dimensão, bem como a distância da tela até o observador (T e D na Figura 13, respectivamente). As equações 4.1 e 4.2 representam o cálculo dos ângulos horizontal (com T_h representando o comprimento) e vertical (com T_v representando a altura), respectivamente:

$$\theta_h = 2 \times \arctan \frac{T_h}{2D} \quad (3.1)$$

$$\theta_v = 2 \times \arctan \frac{T_v}{2D} \quad (3.2)$$

Após aplicar os valores de referência anteriormente mencionados, chegamos aos valores de 53° para θ_h e 31.8° para θ_v

O próximo passo é descobrir a variação angular para cada pixel. O processo é simples ocorre através das fórmulas 4.3 e 4.4:

$$\delta_{\theta_h} = \frac{\theta_h}{P_h} \quad (3.3)$$

$$\delta_{\theta_v} = \frac{\theta_v}{P_v} \quad (3.4)$$

onde δ_{θ_h} representa a variação angular horizontal para cada pixel de acordo com o total de pixels da dimensão horizontal P_h ; δ_{θ_v} representa a variação angular vertical para cada pixel de acordo com o total de pixels da dimensão vertical P_v .

Desta forma, podemos atribuir o valor de 0.027° à variável $\delta_{\theta h}$ e 0.029° à variável $\delta_{\theta v}$

Para determinar a velocidade angular necessária à classificação de sacadas e fixações é indispensável obter primeiramente variação angular entre dois pontos.

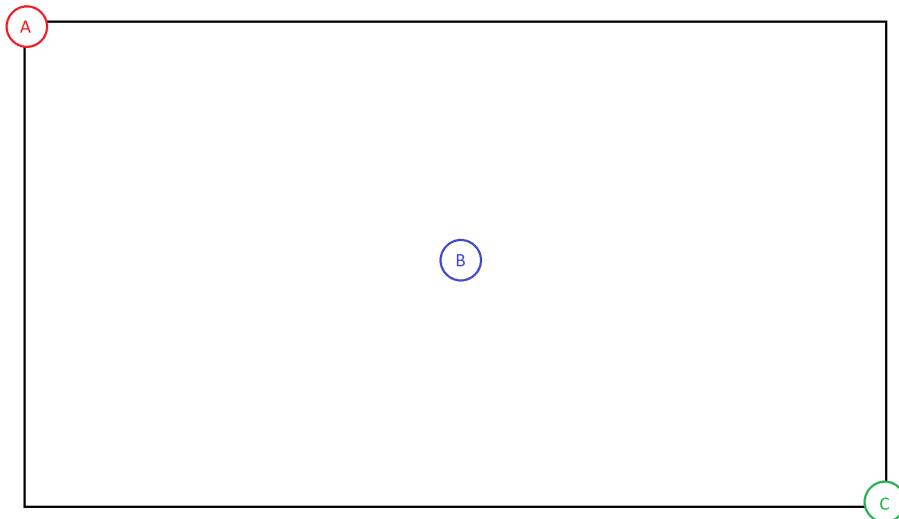
Conforme abordado no Capítulo 2, os dados advindos do sistema *RealEye* são dispostos em coordenadas x e y, normalizadas em relação às dimensões da imagem exibida. Ou seja, valores atribuídos a x e y dizem respeito a porcentagens das dimensões do estímulo. Para elucidar, a Tabela 1 demonstra através de pontos arbitrários (Figura 14) a correspondência entre os índices advindos do *RealEye*, a disposição segundo os eixos cartesianos e a disposição segundo os pixels de uma imagem com 1920 x 1080 px.

Tabela 1 – Correspondência entre índices obtidos pelo RealEye, eixo cartesiano e índices em pixels (em uma imagem 1920x1080 px)

Ponto (NOME)	Índices RealEye (x e y normalizados)	Índice em Px (Linha , Coluna)
Ponto A	0 , 0	0 , 0
Ponto B	0.5,0.5	540 , 960
Ponto C	1 , 1	1080 , 1920

Fonte: Autor.

Figura 14 – Disposição dos pontos descritos na Tabela 1



Fonte: Autor.

Antes de obter a variação angular é necessário garantir que os pontos obtidos no monitoramento ocular estejam centralizados em relação à tela, tornando o centro da tela o ponto inicial

de um mapa cartesiano. Considerando-se as dimensões de referência (1920 x 1080 px) temos:

$$x_c = x - \frac{1920}{2} \quad (3.5)$$

$$y_c = y - \frac{1080}{2} \quad (3.6)$$

em que x_c e y_c representam as coordenadas centralizadas.

Para descobrir a variação angular, Baseia-se na equação a seguir (STUART et al., 2014):

$$D = \sqrt{(x_{c2} - x_{c1})^2 + (y_{c2} - y_{c1})^2} \quad (3.7)$$

em que a D representa a variação espacial e angular entre dois pontos.

Entretanto, utilizamos como referência as coordenadas centralizadas e utilizamos as variáveis $\delta_{\theta h}$ e $\delta_{\theta v}$ como coeficientes de transformação de pixels para graus:

$$\Delta_\Theta = \sqrt{\delta_{\theta h}(x_{c2} - x_{c1})^2 + \delta_{\theta v}(y_{c2} - y_{c1})^2} \quad (3.8)$$

em que Δ_Θ representa a variação angular em uma aproximação válida, dada a razão entre distância de visualização e as dimensões da tela.

3.3 Identificação de limite de velocidade (I-VT)

Velocity-Threshold Identification (I-VT) é um método baseado em um valor de velocidade limite que distingue pontos de fixação e sacada com base em suas velocidades ponto a ponto.

Os perfis de velocidade dos movimentos oculares mostram essencialmente duas distribuições de velocidades: baixas velocidades para fixações (<100 graus / seg) e altas velocidades (> 300 graus / seg) para sacadas (SALVUCCI; GOLDBERG, 2000). Este aspecto dos movimentos oculares torna a distinção baseada na velocidade bastante direta e robusta.

O I-VT inicialmente calcula as velocidades ponto a ponto para cada coordenada através de monitoramento ocular. Para tanto, a velocidade V é calculada através da divisão da variação angular Δ_Θ (cujo processo de obtenção foi descrito na Seção 4.2) pela variação temporal Δ_t , conforme demonstra a equação a seguir:

$$V = \frac{\Delta_\Theta}{\Delta_t} \quad (3.9)$$

O algoritmo então classifica cada ponto como um ponto de fixação ou sacada com base em um limiar de velocidade: se a velocidade do ponto estiver abaixo do limiar, o PoR é classificado como ponto de fixação, caso contrário, ponto de sacada.

Em seguida, os pontos de fixação consecutivos são agrupados, enquanto as sacadas são descartadas.

Finalmente, o algoritmo I-VT traduz cada grupo de fixação para a representação da Figura 15, segundo a qual o tempo do primeiro ponto, o início da fixação, é representado por t e a duração do grupo de fixações é representada por d (SALVUCCI; GOLDBERG, 2000).

Figura 15 – Representação de fixações pós I-VT

<x,y,t,d>

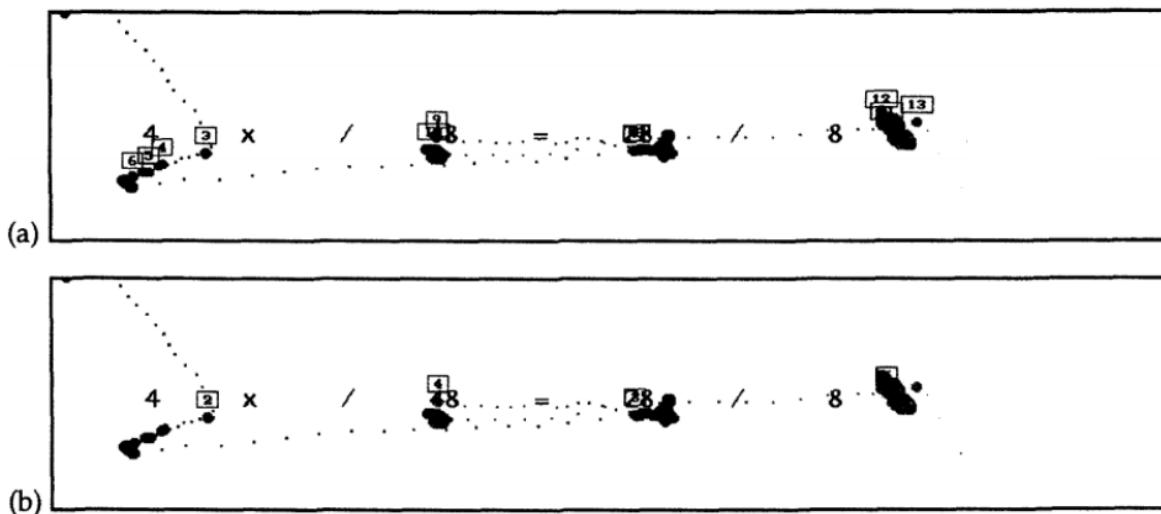
Fonte: Autor.

O I-VT é simples de implementar, funciona com muita eficiência e pode ser executado facilmente em tempo real. No entanto, às vezes ele encontra problemas quando as velocidades do ponto oscilam perto do limite devido ao ruído do rastreador ocular. Especificamente, quando as velocidades de ponto estão próximas do limite. A Figura 16 (a) ilustra esse problema: as fixações 3-6 são interpretadas como quatro fixações separadas, ao passo que parece mais provável que haja apenas uma ou duas fixações reais representadas nesses pontos de dados. Da mesma forma, as fixações 11-13 parecem representar uma única fixação (SALVUCCI; GOLDBERG, 2000).

No passado, um modo de aliviar o problema foi agregar fixações consecutivas sobre um único alvo, como mostrado na Figura 16 (b) (SALVUCCI; GOLDBERG, 2000).

Outros trabalhos propõem como solução considerar durações mínimas para fixações ou sacadas, como impor um limiar mínimo de 10 ms para pontos sacádicos (SEN; MEGAW, 1984 apud SALVUCCI; GOLDBERG, 2000).

Figura 16 – Identificação de fixações com I-VT. Representação em duas dimensões em que os pontos maiores representam as fixações, enquanto os pontos menores representam as sacadas.



Fonte: Salvucci; Goldberg, 2000.

É possível resumir o algoritmo I-VT através dos seguintes passos:

- Calcular velocidades ponto a ponto.
- Rotular cada ponto abaixo do limite de velocidade como um ponto de fixação, caso contrário, como um ponto sacádico.
- Agrupar os pontos de fixação consecutivos, removendo os pontos sacádicos.
- Mapear cada grupo de fixações para uma fixação no centroide de seus pontos.

3.4 Filtragem de fixações

Conforme mencionado na seção anterior, o método I-VT precisa lidar com fatores de imprecisão durante a detecção e o agrupamento de fixações. Neste contexto, para lidar com fixações únicas que, ao passar pelo algoritmo, são compreendidas como múltiplas, é possível trabalhar com limiares temporal e angular para analisar dispersão.

Os limiares de dispersão normalmente usados durante a pesquisa são motivados por características fisiológicas. Dependendo da natureza da tarefa, normalmente é recomendado

que o limite para a duração mínima de fixação seja entre 100 e 200 milissegundos (MANOR; GORDON, 2003 apud BLIGNAUT, 2009).

Além disso, cada fixação é composta de três submovimentos: tremor, deriva e micros-sacadas. Cada submovimento inherentemente adiciona ruído. Portanto, é possível aplicar uma variação angular limite para definir um mesmo grupo de fixações: no máximo, em torno de $0,5^\circ$ (KOMOGORTSEV et al., 2010).

Também é importante notar que a duração do intervalo de tempo entre dois grupos de fixação serve como um filtro para piscadas, cuja duração máxima ocorre na faixa de 75–425 ms (EVINGER; MANNING; SIBONY, 1991 apud KOMOGORTSEV et al., 2010).

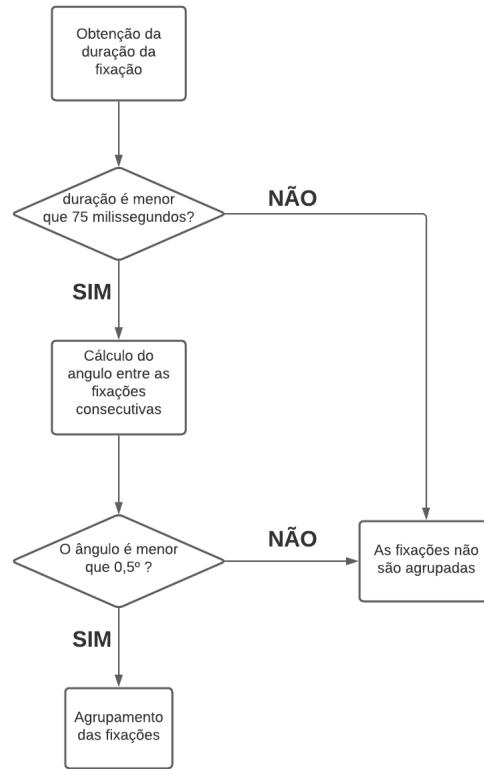
A Seção 3.3 também demonstrou que cada segmento de fixações ao final do algoritmo I-VT comprehende ao agrupamento de fixações consequentes e anteriores a uma sacada. Neste contexto, a duração da fixação passa a denotar o intervalo temporal entre a primeira e a última fixações agrupadas.

Neste estudo, utilizou-se um limiar de 100 ms para descarte de fixações, ou seja, fixações com duração inferior à duração mínima (100 ms) são descartadas da análise.

Em suma, existem dois procedimentos necessários para uma filtragem bem sucedida: Agrupar fixações que tenham sido separadas por piscadas e eliminar fixações com durações abaixo do limiar.

Para honrar os procedimentos descritos, primeiramente as fixações com menos de 75 ms de duração são agrupadas segundo o limiar angular: caso a variação angular entre duas fixações tenha sido menor que $0,5^\circ$, as fixações são novamente agrupadas (Figura 17).

Em seguida, é realizada a análise do limiar temporal e todas as fixações resultantes do agrupamento anterior que ainda possuam duração inferior a 100 ms são descartadas (Figura 18).

Figura 17 – Fluxograma do agrupamento de fixações durante a filtragem

Fonte: Autor.

Figura 18 – Fluxograma da eliminação de fixações durante a filtragem

Fonte: Autor.

4 SCANPATHS E ALGORITMOS DE ANÁLISE

4.1 Introdução

Os primeiros estudos sobre os movimentos dos olhos durante observação de cenas demonstraram que a exploração visual não ocorre de forma aleatória, mas que os movimentos dos olhos estão relacionados ao conteúdo da cena e à tarefa atribuída ao observador (YARBUS, 1967 apud BRANDT; STARK, 1997).

Neste contexto, surge o estudo de *scanpaths*: sequências de fixações e sacadas relacionadas à rotina visual de um indivíduo ao visualizar um estímulo (BRANDT; STARK, 1997).

Desta forma, a comparação de *scanpaths* representa a possibilidade de compreender e comparar peculiaridades entre rotinas visuais de diferentes observadores mediante observação da mesma cena, ou até mesmo avaliar a progressão temporal da rotina visual para uma mesma imagem.

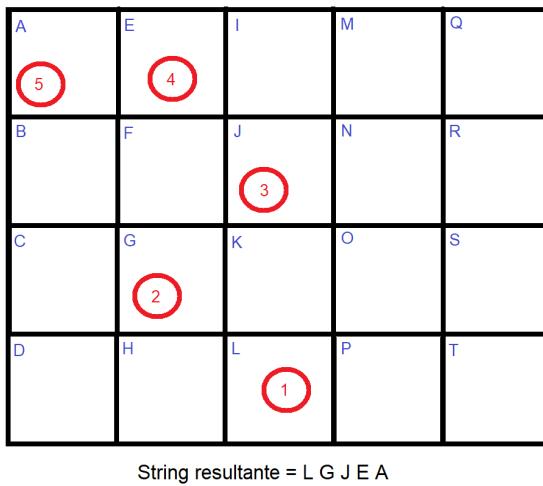
Este capítulo discute duas técnicas para analisar similaridade entre *scanpaths*: *Edit distance* e *ScanMatch*

4.2 *Edit distance*

Uma das primeiras maneiras bem sucedidas de comparar *scanpaths* é baseada em distância de edição da *string* (WAGNER; FISCHER, 1974 apud ANDERSON et al., 2014). Trata-se da medição de dissimilaridade de strings (sequências de caracteres) associadas a *scanpaths*.

O primeiro passo deste algoritmo é sobrepor uma grade à imagem visualizada, de forma que cada célula da grade receba um caractere único. Em seguida, a sequência de fixações é transformada em uma sequência de caracteres com base na posição da fixação em relação às células da grade (ANDERSON et al., 2014).

A Figura 19 demonstra a conversão das fixações de um *scanpath* (indicadas em ordem numérica pelos círculos em vermelho) em sequência de caracteres.

Figura 19 – Exemplo de grade para conversão de scanpaths em strings

Fonte: Autor.

O próximo passo na comparação entre dois *scanpaths* diferentes, cada um representado por sua respectiva *string*, é utilizar uma sequência de transformações chamada distância de Levenshtein: um método que mede o custo de edição para transformar uma *string* em outra, utilizando, em sua forma básica, um conjunto de três operações (inserção, deleção, substituição), cada operação com custo unitário cujo objetivo de obter o custo mínimo de edição entre duas strings (CRISTINO et al., 2010). Neste contexto, a distância de Levenshtein representa a dissimilaridade: a diferença entre duas strings em função do número de caracteres que difere. A Figura 20 demonstra o cálculo da medida de dissimilaridade entre duas strings arbitrárias.

Figura 20 – Exemplo de comparação entre strings para obter dissimilaridade

String 1 = A C F G T Y
 String 2 = A B F T Y

MANIPULAÇÃO DA STRING 1

Passo 1 = SUBSTITUIÇÃO (C por B)

String 1 = A B F G T Y
 String 2 = A B F T Y

Passo 2 = EXCLUSÃO (G)

String 1 = A B F T Y
 String 2 = A B F T Y

Número de passos = dissimilaridade = 2

Fonte: Autor.

A similaridade pode ser representada em função da dissimilaridade e do número de

caracteres da maior *string* envolvida na comparação. Neste caso, podemos criar uma pontuação de similaridade segundo a fórmula a seguir:

$$Sim(S, T) = 1 - \frac{D}{Ta_{max}} \quad (4.1)$$

em que *Sim* representa a pontuação de similaridade entre as strings *S* e *T*, *D* representa a dissimilaridade (número de transformações) e *Ta_{max}* representa o tamanho (número de caracteres) da maior *string* envolvida na comparação.

Apesar de capturar a noção intuitiva da similaridade entre *scanpaths* de uma maneira simples, existe uma problemática em relação às bases do método *Edit distance*: a grade é definida independentemente do conteúdo da imagem e pode separar um elemento do estímulo visual de modo a apontar *scanpaths* semelhantes como totalmente divergentes (ANDERSON et al., 2014).

Em razão dos aspectos negativos associados a este método de comparação, foram propostas variantes. Um exemplo consiste em atribuir caracteres a áreas de interesse pré-definidas com base no formato e na continuidade dos objetos disponíveis na imagem, permitindo ao pesquisador adicionar informações semânticas ao processo de quantização (JOSEPHSON; HOLMES, 2002 apud ANDERSON et al., 2014).

Em uma segunda possibilidade de variação, permite-se repetir caracteres em posições vizinhas, caso uma fixação permaneça em uma célula durante um tempo acima de um dado limite.

Outra modificação interessante inclui a incorporação de uma nova operação ao cálculo de dissimilaridades: transposição entre duas letras presentes na mesma *string*. Trata-se de um princípio que altera a disposição da sequência de caracteres, em detrimento da análise sequencial dos caracteres do *scanpath*.

4.3 ScanMatch

O *ScanMatch* surgiu com o objetivo de contornar algumas deficiências do *Edit distance*. Trata-se de um método de comparação de *scanpaths* que alinha as sequências de movimentos dos olhos com base no algoritmo Needleman-Wunsch, muito utilizado em bioinformática para comparar sequências de DNA, que contabiliza a pontuação por cada transformação entre as letras correspondentes de duas strings através de uma análise estatística (NEEDLEMAN; WUNSCH,

1970 apud CRISTINO et al., 2010). Sua principal vantagem é fornecer pontuações variáveis de acordo com o número de vezes em que uma substituição ocorre entre um par de letras, ou seja, não depende exclusivamente da igualdade entre as letras comparadas para pontuar a similaridade entre os *scanpaths*.

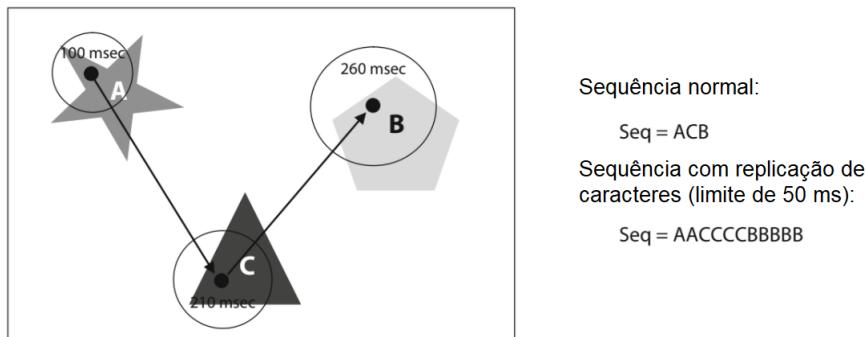
Embora o método *Edit distance* tenha se mostrado uma ferramenta útil e relativamente rápido de calcular, uma de suas principais desvantagens é a falta de relacionamento entre as regiões de interesse, do inglês *Regions of Interest* (RoIs) representadas por células da grade. Como resultado, o algoritmo carece de flexibilidade, uma vez que não pode diferenciar entre regiões próximas e distantes ou, mais geralmente, entre regiões de interesse semelhantes e diferentes (CRISTINO et al., 2010). Uma segunda desvantagem de utilizar *Edit distance* é não levar em consideração a duração da fixação: todas as fixações, sejam curtas ou longas, são tratadas igualmente, embora seja indiscutível que a duração da fixação represente um indicador importante (HENDERSON; GRAHAM, 2008 apud CRISTINO et al., 2010).

No *ScanMatch*, os *scanpaths* são armazenados espacial e temporalmente e, em seguida, codificados para criar uma sequência de letras que representa o local de fixação, a duração e as informações de sequência (ANDERSON et al., 2014).

A codificação para criar a sequência de letras ocorre de modo semelhante ao que ocorre no método de edição de *string*, com uma diferença primordial: o *ScanMatch* calcula a *string* através da análise temporal da sequência de fixações. Esta análise temporal é realizada com base em uma variação temporal que assumirá caráter unitário: o caractere associado a uma célula será replicado caso a fixação permaneça na célula por um tempo maior que determinado limite (variação temporal de referência), sendo o número de repetições do caractere definido pela razão entre tempo de permanência e a variação temporal de referência.

A Figura 21 demonstra a transformação de um *scanpath* arbitrário em *string*, primeiramente apenas em função das regiões e por fim em função também da duração de cada fixação.

Figura 21 – Transformação de fixações para string considerando-se uma variação temporal de referência de 50 ms. Ao lado, strings obtidas segundo o método de edição de string e segundo o método ScanMatch, respectivamente



Fonte: Cristiano et al. 2010, adaptado.

Os algoritmos de alinhamento de sequência global buscam o alinhamento ideal ao longo de todo o comprimento de duas sequências, maximizando sua pontuação. Diferentemente dos algoritmos básicos envolvendo distância de *Levenshtein*, as sequências são alinhadas de acordo com uma matriz de substituição que dá pontuações de acordo com cada alinhamento.

A proposta do algoritmo de Needleman-Wunsch é utilizar programação dinâmica para resolver o problema de alinhamento. Trata-se de uma abordagem dividida em duas partes: primeiro cria-se uma matriz com todas as possibilidades de pontuação baseadas em uma matriz de substituição e uma penalidade de lacuna, e em seguida ocorre a busca do alinhamento ideal. Desta forma, a aplicação de Nedleman-Wunsch exige apenas dois parâmetros a serem definidos: a matriz de substituição e a penalidade de gap (penalidade associada quando não há um correspondente a um elemento na outra *string* envolvida).

Conforme mencionado, uma das principais desvantagens do método de edição de *string* é a falta da computação de relacionamentos entre as regiões de interesse. Esse problema pode ser contornado através da matriz de substituição, uma vez que ela pode codificar informações sobre a relação entre cada ROI. Em bioinformática, várias matrizes estão disponíveis, sendo as mais populares PAMxx e BLOSUMxx (MOUNT, 2008 apud CRISTINO et al., 2010). A função da matriz de substituição é fornecer ao algoritmo uma pontuação (quanto maior, melhor) para alinhar duas letras. Em suma, a pontuação entre duas letras funciona como um fator de semelhança para julgar limiares de proximidade entre regiões de interesse diferentes, mas que possuam alguma relação.

A Figura 22 fornece dois exemplos de alinhamento, cujas pontuações podem ser obtidas através das penalidades de gap e da matriz de substituição.

Figura 22 – Exemplos de alinhamento entre strings com a aplicação de penalidades de gap e de uma matriz de substituição para guiar as pontuações de alinhamento

Comparação entre sequências arbitrárias **A_B** e **AC_**:

Penalidade de gap = 0  Pontuação = $10 + 0 + 0 = 10$

		A	B	C
A	10	-1	-5	
	-1	10	-1	
C	-5	-1	10	

Comparação entre sequências arbitrárias **AB** e **AC_**:

Penalidade de gap = -2  Pontuação = $10 + (-1) = 9$

Fonte: Cristiano et al. 2010, adaptado.

A pontuação de alinhamento fornecida pelo algoritmo Needleman-Wunsch é altamente dependente da matriz de substituição e do comprimento das sequências comparadas. Na verdade, comparando duas strings idênticas com 20 caracteres darão naturalmente uma pontuação mais alta do que duas sequências idênticas com 2 caracteres. Para superar esses problemas, uma normalização da pontuação é realizada (CRISTINO et al., 2010):

$$N = \frac{Sc}{M_m \times T_s} \quad (4.2)$$

em que N representa a pontuação normalizada, Sc representa a pontuação segundo a matriz de substituição, M_m representa o maior valor da matriz de substituição e T_s representa o tamanho da maior *string* envolvida na comparação.

Uma grande vantagem do método *ScanMatch* é que ele pode levar em consideração a similaridade espacial, temporal e sequencial entre os scanpaths. Além disso, as informações semânticas podem ser facilmente adicionadas usando a matriz de substituição. Entretanto, uma desvantagem desse método é que ele sofre com os problemas de quantização inerentes a qualquer medida que use regiões de interesse ou grades (ANDERSON et al., 2014)).

4.3.1 Matrix BLOSUM

Para contar com a eficiência do algoritmo *ScanMatch* é fundamental escolher cautelosamente o método de geração da matriz de substituição.

A matriz de substituição BLOSUM é amplamente usada para marcar alinhamentos de sequência de proteínas. Os valores da matriz são baseados nas substituições de aminoácidos observadas em um grande conjunto de 2.000 padrões de aminoácidos conservados, chamados de blocos. Esses blocos foram encontrados em um banco de dados de sequências de proteínas representando mais de 500 famílias de proteínas relacionadas e atuam como assinaturas dessas famílias de proteínas (MOUNT, 2008). Trata-se de uma abordagem amplamente relacionada ao tamanho do grupo amostral de sequências, retornando uma noção cuja eficiência é diretamente proporcional ao número de dados fornecidos para processamento.

Em termos de análise de *scanpaths*, a metodologia BLOSUM representa uma das mais comuns estratégias adaptadas para a construção de matrizes de substituição (CRISTINO et al., 2010).

Ao final de sua concepção, a matriz BLOSUM fornece valores logarítmicos relacionados às razões de possibilidades (*log-odds ratio*) como uma medida de similaridade. O objetivo da matriz BLOSUM é penalizar alinhamentos com menor probabilidade e recompensar alinhamentos com maior probabilidade, considerando-se as sequências analisadas (*scanpaths*).

Para a construção de uma matriz BLOSUM, o primeiro passo é eliminar sequências repetidas. O segundo passo envolve contabilizar todos os pares dispostos nos alinhamentos, através de uma análise coluna à coluna, conforme exemplifica a Tabela 3 em conjunto com a Tabala 2.

Tabela 2 – Sequências arbitrárias em alinhamento. Cada sequência representa um *scanpath*

Sequência 1	A	A	A
Sequência 2	B	E	A
Sequência 3	A	E	
Sequência 4	C	D	

Fonte: Autor.

É importante destacar que os pares são contabilizados como conjuntos de dois elementos, cuja disposição não implica em pares distintos. Também é importante compreender que a proposta BLOSUM não comprehende ao processamento de sequências de tamanhos diferentes,

Tabela 3 – Contagem de pares por coluna. Cada coluna representa a respectiva posição no *scanpath* de origem

Pares	coluna 1	coluna 2	coluna 3
A e B	2	0	0
A e A	1	0	1
A e C	2	0	0
B e C	1	0	0
A e E	0	2	0
A e D	0	1	0
E e E	0	1	0
E e D	0	1	0

Fonte: Autor.

Tabela 4 – Contagem total de pares

Pares	Contagem total
A e B	2
A e A	2
A e C	2
B e C	1
A e E	2
A e D	1
E e E	1
E e D	1

Fonte: Autor.

pois foi desenvolvida para a análise de sequências de aminoácidos. Entretanto, é possível ignorar as diferenças de tamanho com base na proposta de comparação entre *scanpaths*, assumindo que o fator primordial de análise não está no tamanho das strings analisadas e sim nas substituições observadas entre as células (caracteres) de diferentes *scanpaths*.

Após contabilizar os pares observados em cada coluna, os resultados são somados em uma contagem total, conforme descrito pela Tabela 4 com base nos dados supostos nas Tabelas 2 e 3.

Figura 23 – Matriz C (disposição de pares no alinhamento)

$C =$

	A	B	C	D	E
A	2	2	2	1	2
B			1		
C					
D					
E					1

Fonte: Autor.

A Figura 23 exibe a matriz C como disposição do número total de cada par obtido no alinhamento.

Em seguida, é necessário calcular a frequência de cada par. Este cálculo é realizado através da razão entre os valores dispostos na matriz C e o número de pares possíveis. As fórmulas para o cálculo das frequências e para o cálculo do número de pares possíveis estão descritas a seguir:

$$T = \frac{1}{2}[Co \cdot Ro \cdot (Ro - 1)] \quad (4.3)$$

$$Q(x,y) = \frac{C(x,y)}{T} \quad (4.4)$$

em que T representa o número total de pares, Co representa o número de colunas, Ro representa o número de linhas e $Q(x,y)$ representa a frequência para o elemento de índices x e y descrito por $C(x,y)$. Desta forma, Q representa uma matriz composta pelas frequências de cada par presente na matriz C .

O próximo passo é realizado através do cálculo da probabilidade esperada de cada caracterere:

$$P(x) = Q(x,x) + \frac{1}{2} \times \sum_{x \neq y} Q(x,y) \quad (4.5)$$

em que x e y representam os índices da matriz Q (e portanto representam os caracteres do alfabeto analisado).

Após o cálculo das probabilidades esperadas, é realizado o cálculo das frequências esperadas. Utiliza-se (5.6) para índices iguais e (5.7) para índices diferentes:

$$E(x,x) = P(x)^2 \quad (4.6)$$

$$E(x,y) = 2 \times P(x) \times P(y) \quad (4.7)$$

Os valores das frequências esperadas dão origem a nova matriz E . Entretanto, para gerar valores inteiros, os coeficientes da matriz E são processados por uma transformação logarítmica para a obtenção de uma nova matriz composta por valores de razão de probabilidade logarítmica esperada, segundo a fórmula:

$$L(x,y) = \log_2 \frac{Q(x,y)}{E(x,y)} \quad (4.8)$$

5 ARRANJO EXPERIMENTAL E RESULTADOS

5.1 Introdução

Este capítulo descreve a comparação entre cinco *scanpaths* relacionados à observação de uma imagem médica, neste caso uma tomografia da cabeça, através dos métodos *Edit distance* e *ScanMatch*.

5.2 Hipótese

Uma imagem pode possuir pontos chave que atuem como gatilhos à atenção visual, moldando *scanpaths* diferentes em padrões semelhantes. Entretanto, para estudos envolvendo atenção visual é primordial diferenciar padrões de visualização com base em suas particularidades, compreendendo as características e aspectos que levam a divergências ou convergências entre estratégias visuais de observadores distintos.

A rotina visual de um indivíduo pode ser moldada: é possível instruir pontos importantes a serem visualizados e inspecionados. Observar imagens médicas não faz parte do cotidiano de qualquer indivíduo, trata-se de uma tarefa específica geralmente restrita ao cotidiano de trabalhadores da área da saúde após anos de estudo acerca da fisiologia e anatomia das estruturas a serem visualizadas. Neste contexto, é empírico supor que indivíduos sem contato ou sem muita experiência com ofícios da área da saúde possuam rotinas visuais destoantes das rotinas exercidas por médicos experientes.

Ao analisar comparativamente *scanpaths* de diversos observadores ao visualizar uma imagem médica é possível obter dados quantitativos acerca das rotinas visuais e propor novas estratégias para aprimorá-las de acordo com um objetivo previamente estabelecido.

Desta forma, o experimento propõe a utilização das técnicas *Edit distance* e *ScanMatch* para analisar dados de diferentes observadores em relação ao mesmo estímulo visual (uma imagem médica) e compará-los com o intuito de compreender suas particularidades, semelhanças e divergências.

Em virtude da pandemia do novo Coronavírus (COVID-19), tornou-se impraticável recrutar especialistas da área da saúde como participantes, portanto, o experimento representa uma abordagem em pequena escala, com um número restrito de participantes.

5.3 Estímulo visual e parâmetros de exibição

Para estímulo visual, um arquivo DICOM proveniente de um processo de tomografia computadorizada da cabeça foi tratado através dos passos descritos no Apêndice A e transformado em uma imagem quadrada de 1024 pixels. A exibição foi padronizada com a disposição da imagem no centro da tela, durante 15 segundos.

A exibição, bem como a coleta de dados, ocorreu através da plataforma RealEye, abordada no Capítulo 2.

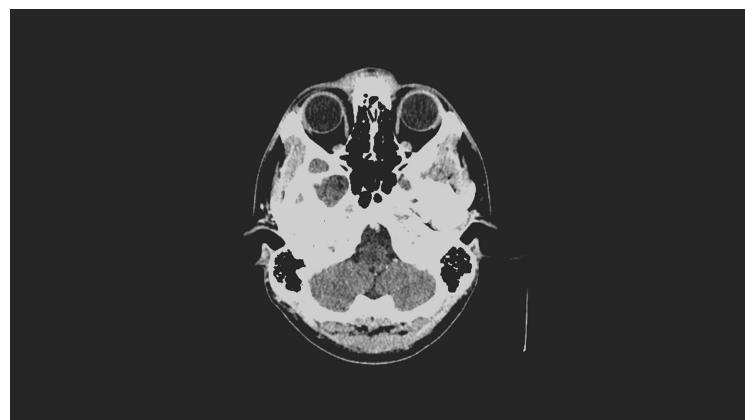
As Figuras 24 e 25 demonstram respectivamente a imagem obtida após o tratamento do arquivo DICOM e a sua disposição para exibição ao centro da tela.

Figura 24 – Imagem utilizada de estímulo visual



Fonte: Autor.

Figura 25 – Disposição do estímulo visual ao centro de uma tela de 1920 pixels por 1080 pixels



Fonte: Autor.

5.4 Participantes

Foram recrutados 5 participantes sem qualquer experiência em relação à análise de imagens médicas com idades variando entre 18 e 25 anos.

Por problemas de conexão durante o acesso à plataforma *RealEye*, um dos voluntários recrutados pôde visualizar a imagem duas vezes. Durante esta segunda visualização, observou-se que a estratégia visual sofreu modificações e que o foco do observador migrou para uma região diferente. Para compensar em parte a falta de um especialista, a familiaridade desse usuário durante a segunda observação da imagens foi considerada como proveniente de um indivíduo mais experiente e tratada como observação de referência.

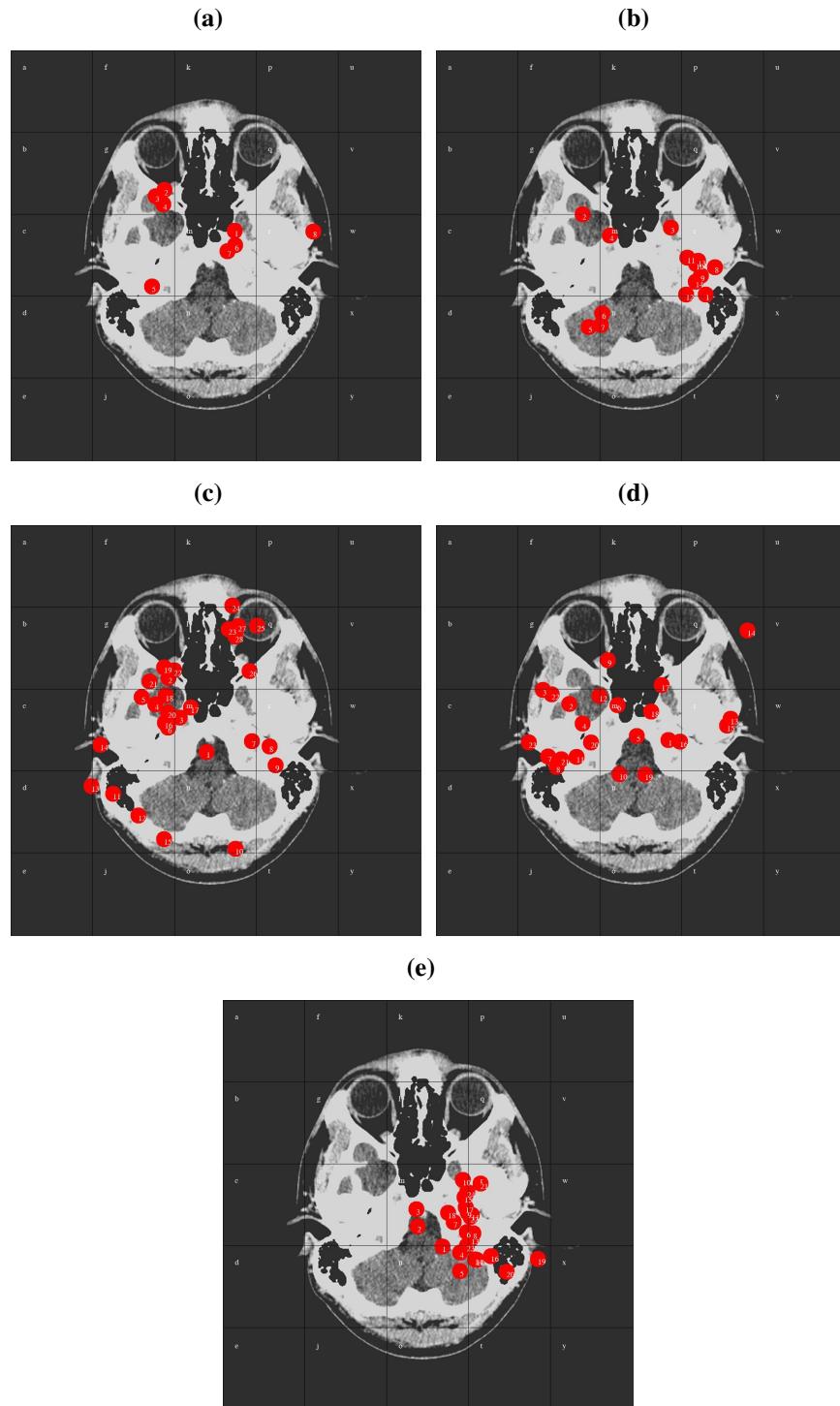
5.5 Tratamento e disposição dos dados obtidos

O capítulo 2 abordou a disposição de dados após o processo de *eye-tracking* e o tratamento para a obtenção de fixações em meio aos dados pré-filtrados.

Após obter e agrupar fixações, a imagem de teste foi dividida em 25 RoIs iguais (matriz 5x5), atribuindo uma letra para cada célula afim de agrupar fixações espacialmente.

A Figura 26 exibe os *scanpaths* dos 5 voluntários, dispostos na grade.

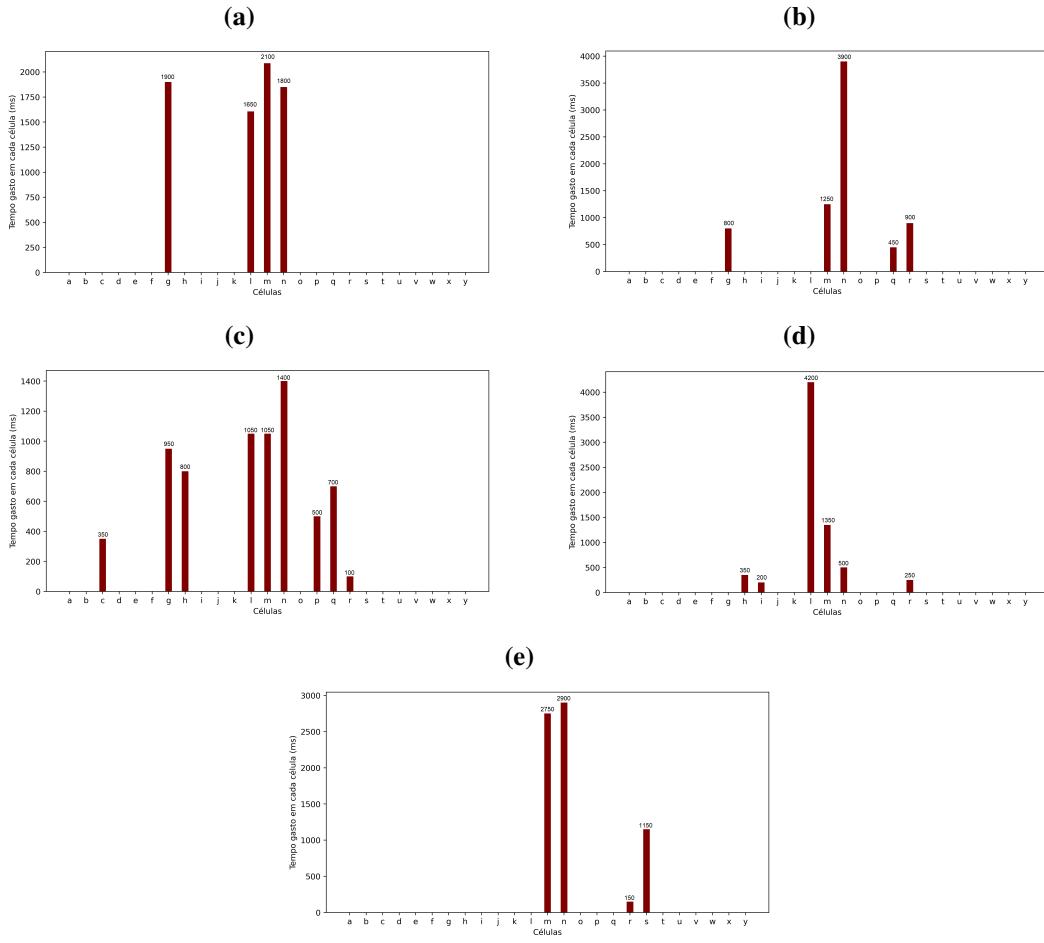
Figura 26 – Scanpaths dos observadores 1 (a), 2 (b), 3 (c), 4 (d) e do observador de referência (e).
Os círculos vermelhos representam as fixações, seus números denotam a ordem de observação e as letras denominam as células na grade.



Fonte: Autor.

O tempo gasto em cada célula durante as fixações pode ser descrita através dos gráficos a seguir, denotando o tempo em milissegundos.

Figura 27 – Tempo gasto em cada célula durante as fixações para os observadores 1 (a), 2 (b), 3 (c), 4 (d) e para observador de referência (e). Os círculos vermelhos representam as fixações, seus números denotam a ordem de observação e as letras denominam as RoIs



Fonte: Autor.

A Figura 27 demonstra uma forte tendência à observação das regiões g, m e n, enquanto demonstra interesse atípico pelas regiões r e s.

5.6 Estratégias e resultados de comparação

Para comparar os *scanpaths*, optou-se pelas estratégias *Edit string* (sem e com replicação de caracteres) e *ScanMatch*.

5.6.1 Aplicação de *Edit distance*

Aplicam-se neste experimento duas versões diferentes do algoritmo de *Edit distance*, descritas em termos de suas particularidades: a primeira versão envolve comparar strings cujos caracteres são resultado exclusivo da transformação de cada fixação e a segunda versão envolve

replicar os caracteres da *string* com base na duração das fixações. Cabe destacar que, para as duas versões, este experimento optou por utilizar o tamanho da menor *string* como critério de normalização, visto que não há interesse em punir os observadores com base apenas no número de regiões diferentes observadas durante o monitoramento ocular.

5.6.1.1 *Edit distance* sem replicação de caracteres

Para demonstrar as consequências de comparações através do método *Edit distance* sem replicação de caracteres, a Tabela 5 apresenta o tamanho de cada *string* obtida:

Tabela 5 – Tamanho das strings (sem expansão) relacionadas ao scanpath de cada observador

Observador	Tamanho da String
Observador 1	8
Observador 2	14
Observador 3	28
Observador 4	23
Observador de referência	24

Fonte: Autor.

A tabela 6 apresenta os resultados para comparações entre *strings* através de *Edit distance*, atribuindo o valor 1 como pontuação máxima de similaridade:

Tabela 6 – Pontuação de similaridade entre strings através do método *Edit distance* (sem replicação de caracteres)

	Observador 1	Observador 2	Observador 3	Observador 4	Observador de referência
Observador 1	1	0,14	0,178	0,173	0,16
Observador 2		1	0,142	0,217	0,29
Observador 3			1	0,25	0,170
Observador 4				1	0,166
Observador de referência					1

Fonte: Autor.

É possível observar que as pontuações relacionadas à comparações entre strings diferentes são muito baixas. Isso ocorre porque, ao ignorar a duração de cada fixação, a comparação perde parte da acurácia relacionada à disposição temporal das regiões visualizadas.

5.6.1.2 *Edit distance* com replicação de caracteres

A replicação obedece a um caráter objetivo: cada caractere será replicado com base na duração da fixação correspondente durante o processo de *eye-tracking*. Desta forma, utilizando-se de um limite de 50 milissegundos (semelhante à abordagem do método de *ScanMatch* descrito

no Capítulo 4), o número de caracteres relacionados à uma fixação após a replicação é descrito pelo quociente entre a duração da fixação e o limite de 50 milissegundos.

Tabela 7 – Tamanho das strings (com expansão) relacionadas ao scanpath de cada observador

Observador	Tamanho da String
Observador 1	150
Observador 2	146
Observador 3	138
Observador 4	136
Observador de referência	139

Fonte: Autor.

A Tabela 7 explicita o tamanho de cada *string* após a replicação de caracteres e demonstra que, apesar do tamanho consideravelmente diferente antes do processo de replicação, o somatório do tempo despendido por cada observador apresenta valores muito semelhantes.

A Tabela 8 exibe os resultados de todas as possíveis comparações entre os *scanpaths* através do método *Edit distance* com replicação de caracteres, atribuindo o valor 1 como pontuação máxima:

Tabela 8 – Pontuação de similaridade entre strings através do método *Edit distance* (com replicação de caracteres)

	Observador 1	Observador 2	Observador 3	Observador 4	Observador de referência
Observador 1	1	0,353	0,07	0,19	0,32
Observador 2		1	0,06	0,15	0,26
Observador 3			1	0,217	0,32
Observador 4				1	0,13
Observador de referência					1

Fonte: Autor.

Através da análise entre as Tabelas 6 e 8 é possível observar que as mudanças mais bruscas entre pontuações obtidas ocorreram entre strings cujos tamanhos anteriores à expansão apresentavam maior diferença. Além disso, é possível observar como a comparação entre strings de tamanhos muito diferentes impacta negativamente a pontuação, gerando uma falsa impressão de disparidade entre *scanpaths*.

5.6.2 ScanMatch

Para o método *ScanMatch*, optou-se por uma matriz de substituição baseada no algoritmo BLOSUM através dos alinhamentos presentes entre os caracteres dos *scanpaths* (Tabela 9). O objetivo é pontuar similaridade com base na correspondência entre caracteres alinhados e

também nas probabilidades de alinhamentos entre caracteres não correspondentes (aspectos abordados no Capítulo 4).

Tabela 9 – Alinhamentos presentes para a geração da matriz BLOSUM

	<i>l</i>	<i>m</i>	<i>n</i>	<i>r</i>
<i>g</i>	15	12	9	14
<i>l</i>		12	13	18
<i>m</i>			7	10
<i>n</i>				13

Fonte: Autor.

A principal vantagem do *ScanMatch* em relação ao método de *Edit distance* está na pontuação de alinhamentos entre caracteres diferentes. Desta forma, a matriz de substituição contabiliza, através de todos os *scanpaths* coletados, os alinhamentos mais comuns entre células diferentes, atribuindo caráter positivo durante a análise da pontuação.

A Tabela 10 explicita os resultados das comparações entre *scanpaths* através do método *ScanMatch*, atribuindo o valor 1 como pontuação máxima:

Tabela 10 – Pontuação de similaridade entre strings através do método ScanMatch

	Observador 1	Observador 2	Observador 3	Observador 4	Observador de referência
Observador 1	1	0,654	0,35	0,54	0,53
Observador 2		1	0,292	0,525	0,412
Observador 3			1	0,521	0,527
Observador 4				1	0,57
Observador de referência					1

Fonte: Autor.

Os resultados apresentados (Tabela 10) permitem notar o impacto da atribuição de pontuações positivas para alinhamentos entre caracteres diferentes visto que, através do *ScanMatch*, é possível aprofundar a análise de similaridade entre *scanpaths* com muitos alinhamentos entre RoIs distintas.

6 CONCLUSÃO

Este trabalho propôs um sistema obter e classificar PoRs, e comparar os *scanpaths* oriundos da observação de imagens médicas. O objetivo foi aplicar conhecimentos de *eye-tracking* e de comparação de rotinas visuais para avaliar similaridades e divergências entre os *scanpaths* coletados.

Através da elaboração e execução do experimento, foi possível aprimorar o conhecimento em tecnologias de *eye-tracking*, classificação de movimentos oculares e métodos de comparação para *scanpaths*, estendendo conteúdos teóricos para aplicação prática em um sistema funcional e futuramente escalonável. O sistema gerou resultados consistentes com o contexto apresentado: voluntários aleatórios, em número reduzido e sem experiência com análise de imagens médicas. Os algoritmos para comparação de *scanpaths* utilizados neste trabalho, reiteram a maior eficácia do *ScanMatch* em relação ao *Edit distance* de acordo com os dados apresentados (Tabelas 6,8 e 10) e das discussões expostas no Capítulo 5 em virtude da análise de similaridade capaz de pontuar melhor o relacionamento entre RoIs diferentes.

6.1 Considerações Finais

O presente trabalho apresenta resultados preliminares e subsídios para uma proposta mais ampla de modelagem de comportamento visual, inicialmente voltada para o treinamento médico na área de diagnóstico por imagem. Trata-se de uma proposta voltada ao escalonamento futuro, que ainda carece de dados e meios necessários mas que conta com resultados palpáveis para justificar trabalhos futuros em continuidade com a hipótese, métodos e algoritmos inicialmente desenvolvidos.

Foi possível obter múltiplos dados relacionados ao comportamento visual de diversos observadores e gerar pontuações de similaridade consistentes, baseadas em algoritmos de uso consolidado para comparação entre *scanpaths*.

Além disso, por agrupar diferentes técnicas em diversas etapas (monitoramento ocular, geração de *scanpaths* e comparação entre rotinas visuais), é possível reorganizar os métodos aplicados, adicionando ou modificando as técnicas para análise dos movimentos oculares ou os algoritmos para comparação de *scanpaths*. Desta forma é viável adaptar o estudo à obtenção e classificação automatizada de *scanpaths* através de *eye-trackers* mais sofisticados, implementar

novos algoritmos de análise de similaridade ou até mesmo compensar a utilização de *eye-trackers* de baixo custo com novas etapas de processamento.

6.2 Trabalhos futuros

Para trabalhos futuros:

- Recrutar voluntários da área da saúde com diversidade em termos de tempo de experiência no ofício da medicina e em análise de imagens médicas.
- Dispor de dois grupos de participantes: um grupo composto por especialistas em diagnóstico através de imagens médicas e um grupo composto por estudantes ou profissionais menos experientes.
- Aprimorar a acurácia da classificação de movimentos oculares.
- Aplicar novos e diversos algoritmos para comparação entre *scanpaths*.
- Sintetizar e aplicar entrevistas de caráter subjetivo e objetivo para fornecer novas nuances aos dados coletados
- Incluir lógica difusa ao sistema para incorporar novas nuances ao processamento de dados.
- Criar um sistema de baixo custo móvel ou online para pautar estudos e coleta de dados de modo escalonável.

REFERÊNCIAS

- ANDERSON, N. et al. A comparison of scanpath comparison methods. **Behavior research methods**, v. 47, 12 2014.
- BLIGNAUT, P. Fixation identification: The optimum threshold for a dispersion algorithm. **Attention, perception psychophysics**, v. 71, p. 881–95, 06 2009.
- BRANDT, S. A.; STARK, L. W. Spontaneous eye movements during visual imagery reflect the content of the visual scene. **Journal of Cognitive Neuroscience**, v. 9, n. 1, p. 27–38, 1997.
- CRISTINO, F. et al. Scanmatch: A novel method for comparing fixation sequences. **Behavior research methods**, v. 42, p. 692–700, 08 2010.
- DUCHOWSKI, A. T. Eye tracking methodology. **Theory and practice**, Springer, v. 328, n. 614, p. 2–3, 2007.
- EVINGER, C.; MANNING, K.; SIBONY, P. Eyelid movements. mechanisms and normal data. **Investigative ophthalmology visual science**, v. 32, p. 387–400, 03 1991.
- FERHAT, O.; VILARIÑO, F. Low cost eye tracking: The current panorama. **Computational Intelligence and Neuroscience**, v. 2016, p. 1–14, 03 2016.
- GONÇALVES, L. et al. System and protocol for collecting subjective data on clinical quality of compressed dicom images. In: . [S.l.: s.n.], 2017.
- GREGORY, R. L. **Eye and Brain: The Psychology of Seeing - Fifth Edition**. Rev - revised, 5. Princeton University Press, 1997. ISBN 9780691048376. Disponível em: <<http://www.jstor.org/stable/j.ctvc77h66>>.
- HENDERSON, J. M.; GRAHAM, P. L. Eye movements during scene viewing: evidence for mixed control of fixation durations. **Psychonomic bulletin amp; review**, v. 15, n. 3, p. 566—573, June 2008. ISSN 1069-9384. Disponível em: <<https://doi.org/10.3758/pbr.15.3.566>>.
- JOSEPHSON, S.; HOLMES, M. Attention to repeated images on the world-wide web: Another look at scanpath theory. **Behavior Research Methods Instruments and Computers**, v. 34, p. 539–548, 12 2002.
- KOMOGORTSEV, O. et al. Standardization of automated analyses of oculomotor fixation and saccadic behaviors. **Biomedical Engineering, IEEE Transactions on**, v. 57, p. 2635 – 2645, 12 2010.
- LAURUTIS, V. P.; ROBINSON, D. A. The vestibulo-ocular reflex during human saccadic eye movements. **The Journal of Physiology**, v. 373, n. 1, p. 209–233, 1986. Disponível em: <<https://physoc.onlinelibrary.wiley.com/doi/abs/10.1113/jphysiol.1986.sp016043>>.
- MANOR, B. R.; GORDON, E. Defining the temporal threshold for ocular fixation in free-viewing visuocognitive tasks. **Journal of Neuroscience Methods**, v. 128, n. 1, p. 85–93, 2003. ISSN 0165-0270. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0165027003001511>>.
- MOUNT, D. Using blosum in sequence alignments. **CSH protocols**, v. 2008, p. pdb.top39, 06 2008.

- MUSTRA, M.; DELAC, K.; GRGIC, M. Overview of the dicom standard. In: **2008 50th International Symposium ELMAR.** [S.l.: s.n.], 2008. v. 1, p. 39–44.
- NEEDLEMAN, S. B.; WUNSCH, C. D. A general method applicable to the search for similarities in the amino acid sequence of two proteins. **Journal of Molecular Biology**, v. 48, n. 3, p. 443–453, 1970. ISSN 0022-2836. Disponível em: <<https://www.sciencedirect.com/science/article/pii/0022283670900574>>.
- PHAN, A.-C.; VO, V.-Q.; PHAN, T.-C. A hounsfield value-based approach for automatic recognition of brain haemorrhage. **Journal of Information and Telecommunication**, Taylor Francis, v. 3, n. 2, p. 196–209, 2019. Disponível em: <<https://doi.org/10.1080/24751839.2018.1547951>>.
- PIANYKH, O. S. **Digital Imaging and Communications in Medicine (DICOM): A Practical Introduction and Survival Guide.** 1st. ed. [S.l.]: Springer Publishing Company, Incorporated, 2010. ISBN 3642094007.
- ROBINSON, D. A. The oculomotor control system: A review. **Proceedings of the IEEE**, v. 56, n. 6, p. 1032–1049, 1968.
- SALVUCCI, D.; GOLDBERG, J. Identifying fixations and saccades in eye-tracking protocols. In: . [S.l.: s.n.], 2000. p. 71–78.
- SEMMELMANN, K.; WEIGELT, S. Online webcam-based eye tracking in cognitive science: A first look. **Behavior research methods**, v. 50, 06 2017.
- SEN, T.; MEGAW, T. The effects of task variables and prolonged performance on saccadic eye movement parameters. In: GALE, A. G.; JOHNSON, F. (Ed.). **Theoretical and Applied Aspects of Eye Movement Research.** North-Holland, 1984, (Advances in Psychology, v. 22). p. 103–111. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0166411508618245>>.
- STUART, S. et al. Quantifying saccades while walking: Validity of a novel velocity-based algorithm for mobile eye tracking. In: . [S.l.: s.n.], 2014. v. 2014.
- WAGNER, R. A.; FISCHER, M. J. The string-to-string correction problem. **J. ACM**, Association for Computing Machinery, New York, NY, USA, v. 21, n. 1, p. 168–173, jan. 1974. ISSN 0004-5411. Disponível em: <<https://doi.org/10.1145/321796.321811>>.
- YARBUS, A. L. **Eye Movements and Vision.** [S.l.]: Plenum. New York., 1967.

A - IMAGENS MÉDICAS

A.1 Introdução

Computadores estão presentes em quase todos os aspectos de nossas vidas, incluindo áreas da medicina, a exemplo do desenvolvimento de diagnóstico auxiliado por computador (CAD). Neste contexto, em decorrência de muitas direções no desenvolvimento de equipamentos de imagens médicas, mostrou-se importante criar um padrão para a conexão e troca de informações entre médicos e aparelhos. Após diversas revisões, DICOM (Digital Imaging and Communications in Medicine) se tornou o mais comum conjunto de normas para padronizar o formato eletrônico utilizado no armazenamento e comunicação de imagens (MUSTRA; DELAC; GRGIC, 2008).

O DICOM é um padrão grande e complexo, pois deve ser adequado para diferentes ramos médicos como oftalmologia, cardiologia, ultrassom e tomografia computadorizada (MUSTRA; DELAC; GRGIC, 2008).

Este trabalho baseou-se na utilização de imagens no padrão DICOM geradas para tomografia computadorizada. Ao longo do capítulo serão abordadas particularidades pertinentes ao uso deste padrão.

A.2 Imagens de tomografia computadorizada

Para aproximar este estudo das condições reais para observação de imagens médicas optou-se pela utilização imagens de tomografia computadorizada. Tais imagens foram obtidos através da plataforma Patient Contributed Image Repository (PCIR), uma base de dados gratuita criada para compartilhamento de arquivos DICOM diretamente pelos pacientes, com o objetivo de permitir o livre uso para consultas e pesquisas.

Devido à maior correspondência com artigos encontrados na fase de pesquisa bibliográfica deste trabalho, foi escolhida uma tomografia do crânio para constituir o experimento descrito em capítulos posteriores.

A.3 Tomografia computadorizada e unidades Hounsfield

Em uma tomografia computadorizada cada voxel (pixel tridimensional) possui um valor relacionado à densidade do tecido exposto.

Cada pixel é relacionado a um valor numérico denominado *Número CT*, que denota a média de todos os valores de atenuação contidos no voxel correspondente. Este valor numérico é comparado ao valor de atenuação da água e exibido em uma escala denominada de unidades Hounsfield.

O arquivo DICOM é capaz de registrar em 16 bits mais de 65.536 níveis de cinza para exibição de uma imagem monocromática, capturando assim as menores nuances em imagens médicas. Em comparação, a conversão de DICOM em JPEGs limita a exibição em 256 níveis de cinza (8 bits por pixel) e frequentemente os torna inadequados para leitura diagnóstica. Além disso, muitos monitores não suportam níveis de cinza suficientes para exibir com fidelidade suficiente uma imagem DICOM (PIANYKH, 2010).

Neste contexto, é necessário realizar uma operação denominada janelamento, a saber, uma transformação de intensidade de pixels, onde a faixa de valores dos pixels correspondentes a toda a escala de cinza, branco e preto é ajustada para destacar a estrutura anatômica de interesse. Tal processo é essencial, pois esta função de mapeamento torna os valores de pixel consistentes com a capacidade do dispositivo de exibição através de uma transformação linear envolvendo unidades Hounsfield, permitindo enfatizar diferentes tipos de tecidos (GONÇALVES et al., 2017).

Os valores dos pixels em unidades Hounsfield associados a diferentes substâncias podem ser observados na Tabela 11:

Tabela 11 – Valor Hounsfield (HU) de substâncias comuns

Substância	HU
Água	0
Osso	1000
Ar	-1000
Matéria cinza	35-40
Matéria Branca	20
Sangue	40-90
Calcificação	> 120

Fonte: Phan, Vo, Phan, 2019.

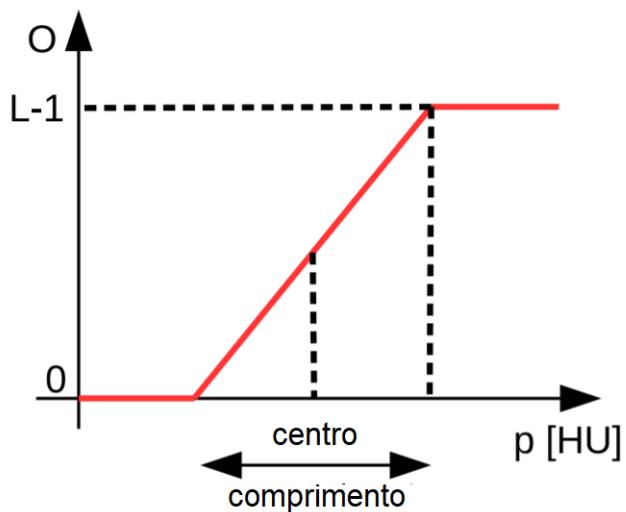
Para guiar a transformação para unidades Hounsfield, os arquivos DICOM contam com as tags *Rescale intercept*, e *rescale slope*) que podem ser aplicadas na seguinte equação linear (PHAN; VO; PHAN, 2019)):

$$HU = PixelValue \times RescaleSlope + RescaleIntercept \quad (\text{A.1})$$

onde *PixelValue* é o valor de cada ponto da imagem, enquanto *RescaleSlope* e *RescaleIntercept* são os valores dos parâmetros fornecidos nas imagens DICOM.

Após obter os valores dos pixels em unidades Hounsfield, é necessário aplicar o processo de janelamento através de um valor central e um valor de comprimento para definir o intervalo que será representados dentro da capacidade do monitor à disposição.

Figura 28 – Exemplo de função de janelamento

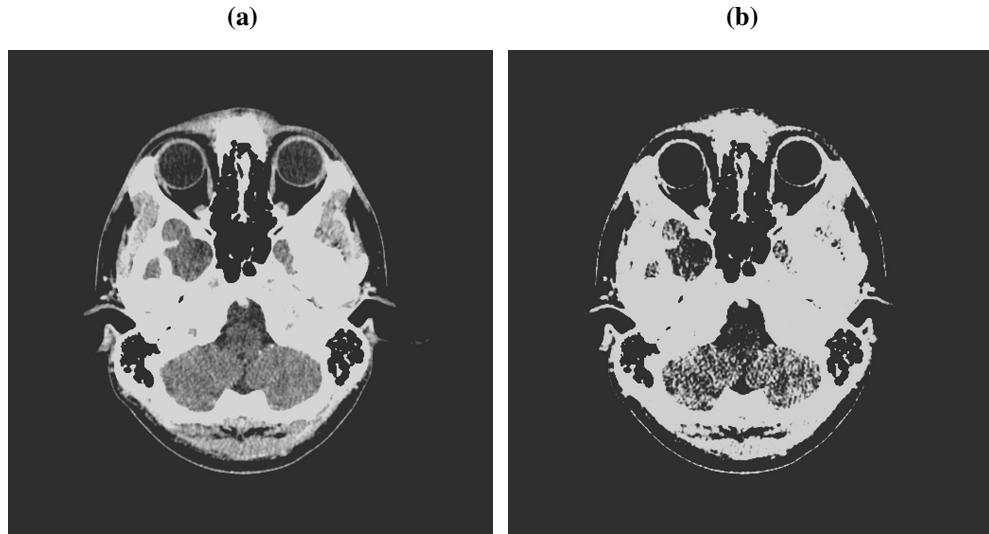


Fonte: Gonçalves, 2017.

Na Figura 28 é possível observar, além do centro e do comprimento da janela, que $p[\text{HU}]$ e O denotam os valores de pixel antes (em valeres Hounsfield) e depois do processo de janelamento, respectivamente. O intervalo $[0, L-1]$, por sua vez, representa a faixa de exibição do dispositivo de visualização de preto a branco (GONÇALVES et al., 2017).

Tratando-se de uma tomografia, este estudo optou por selecionar uma faixa em unidades Hounsfield que destacasse a exibição de matéria cinza e matéria branca. Portanto, tomamos 40 HU como centro de uma janela em uma faixa de 80 HU, resumindo-se de 0 HU até 80 HU. Para fins de comparação, a Figura 29 apresenta o mesmo slice após passar por janelamentos diferentes.

Figura 29 – Janelamento da mesma slice, com centros em 40 HU e comprimentos de: 80 HU em (a) e de 40 HU em (b)



Fonte: Autor.

A Figura 29 denota janelamentos diferentes oriundos da mesma *slice* (fadia horizontal da tomografia), ambos realizados com o centro em 40 HU. Na imagem da esquerda é possível reconhecer um mapeamento mais complexo em tons de cinza, diretamente relacionado ao comprimento da janela, em uma faixa de 80 HU (0 HU a 80 HU). A imagem da direita possui transições mais bruscas entre os tons de cinza, por mapear a faixa de 40 HU (20 HU a 60 HU).

É importante destacar que, após o janelamento, a imagem passa por um processo de interpolação para adquirir as dimensões desejadas. Neste estudo optou-se por gerar imagens finais com dimensões iguais a 1024 pixels de largura e 1024 pixels de altura. O método escolhido foi o de interpolação bicúbica por tratar-se do método academicamente mais usual (PIANYKH, 2010).

Por conta da plataforma utilizada para rastreamento ocular, a imagem usada neste trabalho precisou ser salva em JPEG, conforme será detalhado no Capítulo 3.

Figura 30 – Fluxograma do tratamento de imagem descrito neste capítulo



Fonte: Autor.