

EyePursuitLinks - an Eye-pursuit Based Interface for Web Browsing Using Smart Targets

Alex Torquato Souza Carneiro
alex@alumni.usp.br
Department of Computer Science
University of São Paulo
São Paulo, Brazil

Candy Veronica Tenorio
Gonzales
candytg@ime.usp.br
Department of Computer Science
University of São Paulo
São Paulo, Brazil

Carlos H. Morimoto
hitoshi@ime.usp.br
Department of Computer Science
University of São Paulo
São Paulo, Brazil



Figura 1: EyePursuitLinks sendo usado com hyperlinks numerados do 0-7, 0-4, 0-1, e 0 nos 4 níveis de navegação na web. A interface à direita ilustra os alvos que estão se movendo sobre a trajetória circular. Para selecionar um link, basta o usuário perseguir o alvo correspondente até em 4 níveis. Em cada nível os links estão representados por um conjunto de números.

ABSTRACT

Web accessibility and digital inclusion are still one of the great research challenges in computer science. For people with severe motor disabilities, eye-gaze interaction enables hands-free operation of graphical interfaces. Nonetheless, gaze tracking devices typically require frequent user calibration that weakens the overall user experience and performance. In this paper we propose EyePursuitLinks, an eye-pursuit based technique that allows people with severe motor disabilities to browse the web by following multiple moving targets corresponding to the links they want to follow. The main contribution of this paper is the use of Smart Targets (ST) to select potentially large number of hyperlinks within a relatively small area. We have performed a user experiment with 10 volunteers to evaluate the performance of the Smart Targets selection against conventional pursuit selection mechanism using 4, 8, and 16 simultaneous targets. Our results show that the use of ST is significantly more robust than the conventional method for larger number of targets.

CCS CONCEPTS

• **Human-centered computing** → **Interaction techniques**; *Ubiquitous and mobile computing*; • **Computing methodologies** →

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

WebMedia '23, October 23–27, 2023, Ribeirão Preto, Brazil

© 2023 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM ISBN 979-8-4007-0908-1/23/10...\$15.00

<https://doi.org/10.1145/3617023.3617058>

Graphics systems and interfaces; • Mathematics of computing → Probabilistic algorithms.

KEYWORDS

Eye-gaze interaction, Responsive interfaces, Probabilistic algorithm, Smooth pursuits

ACM Reference Format:

Alex Torquato Souza Carneiro, Candy Veronica Tenorio Gonzales, and Carlos H. Morimoto. 2023. EyePursuitLinks - an Eye-pursuit Based Interface for Web Browsing Using Smart Targets. In *Proceedings of the Brazilian Symposium on Multimedia and the Web (WebMedia '23)*, October 23–27, 2023, Ribeirão Preto, Brazil. ACM, New York, NY, USA, 9 pages. <https://doi.org/10.1145/3617023.3617058>

1 INTRODUÇÃO

Acessibilidade e integração digital continuam sendo grandes desafios de pesquisa no Brasil e no mundo [1]. Para pessoas com deficiências motoras severas, as interfaces baseadas no olhar podem permitir a sua integração digital [14, 16]. As interfaces baseadas no olhar utilizam rastreadores de olhar para estimar a posição instantânea do olhar sobre o monitor do computador. Esses dispositivos podem ser usados para substituir o mouse para a função de apontamento, mas a seleção pelo olhar (clique do mouse) pode ser um desafio. Para certas pessoas, o clique pode ser feito por meio de uma chave mecânica acionada pelos pés ou pela boca por exemplo. Dentre as técnicas de seleção feitas exclusivamente pelo olhar, a mais comum é conhecida como tempo de latência (dwell-time), onde o usuário deve manter seu olhar sobre o objeto desejado durante um certo tempo (latência) até o objeto ser selecionado [13].

Dentre os desafios para o uso do tempo de latência está a escolha do intervalo de tempo necessário para cada seleção. Quando ele

é muito curto, a seleção dos objetos pode ser feita de forma involuntária e quando esse tempo é muito longo, a interface se torna muito vagarosa. Outra limitação dessa interface é devida a falta de acurácia dos rastreadores de olhar comparado ao mouse, e também da necessidade frequente de calibração dos dispositivos. O método de seleção por perseguição de alvos (SP - Smooth Pursuit), foi sugerido por Vidal et al. [20] no sistema Pursuits como alternativa para evitar a necessidade de calibração por meio do reconhecimento de padrões de movimento dos objetos. A ideia é apresentar vários alvos móveis, cada um com um movimento particular, distintos entre si. A partir de uma medida de similaridade entre os movimentos dos alvos e os movimentos do olhar é possível realizar a seleção de um alvo. As vantagens apresentadas pelo método baseado em SP serviram de inspiração para vários outros trabalhos na área de interação pelo olhar que estão apresentados na Seção 2.

A Figura 1 ilustra como um navegador poderia utilizar uma interface baseada em SP para selecionar um dentre vários hyperlinks. Cada link pode ser associado a um número (ou rótulo). Esses rótulos podem ficar dispostos em um círculo, como mostrado no lado direito da figura, onde os rótulos podem ficar girando com velocidade constante, por exemplo. Para selecionar um link, basta o usuário perseguir o rótulo desejado. Observe que essa interface permite que o usuário observe toda a página sem se preocupar com seleções involuntárias e o uso de perseguição de alvos móveis dispostos sobre um círculo permite que a seleção seja realizada sem necessidade de calibração do rastreador de olhar e usando uma área relativamente pequena da interface. O uso desses alvos circulares foi introduzido por Esteves et al. [8] e vários algoritmos para estimar o alvo perseguido foram sugeridos por Velloso et al. [17].

Uma grande limitação desses algoritmos é com relação ao número de objetos selecionáveis pois, quanto maior o número de objetos se movendo ao redor do círculo, maior a similaridade entre o movimento do objeto desejado e de seus vizinhos mais próximos. Diante desses desafios, elaboramos uma solução a qual denominamos *Smart Targets* que consiste em uma interface responsiva aos movimentos do usuário e um algoritmo probabilístico que identifica os alvos com maior probabilidade de perseguição. A combinação da interface responsiva e do algoritmo probabilístico reduz o risco de seleções involuntárias ou equivocadas, onde o alvo selecionado não é o desejado, e torna a solução robusta ao aumento na quantidade de alvos exibidos simultaneamente. Os conceitos implementados no *Smart Targets* estão descritos na Seção 3. Para avaliar o desempenho da interface usando *Smart Targets* e para evitar a interferência da tarefa de navegar na Internet, desenvolvemos uma aplicação para inserir 4 caracteres usando *Smart Targets*. Esta aplicação foi avaliada em um experimento com 10 voluntários, descrito na Seção 4. Os resultados obtidos no experimento são apresentados e discutidos na Seção 5. Por fim, a Seção 6 conclui este trabalho.

2 TRABALHOS CORRELATOS

A seleção pelo olhar possui diversas características que influenciam o desempenho e a experiência do usuário: a técnica e algoritmo de seleção, a interface, e a tarefa a ser realizada pelo usuário. A seguir dividimos esta seção segundo estas características.

2.1 Seleção por Perseguição

Um dos trabalhos inspirados pelo Pursuits foi o *Orbits*, proposto por Esteves et al. [8]. Este trabalho propõe o uso de trajetórias circulares concêntricas para os alvos que são exibidos em uma interface pequena, compatível com relógios inteligentes. Os autores investigaram o efeito de diferentes atributos como tamanho da trajetória e velocidade angular, para a robustez na seleção dos alvos. Os resultados mostraram que o *Orbits* permite seleções com precisão próxima de 80% quando a quantidade de alvos é limitada ao máximo de 8, o que foi verificado por outros autores, como Verweij et al. [19] no sistema *WaveTrace*.

Os sistemas *Orbits* e *WaveTrace* utilizaram alvos que se movem sobre uma mesma trajetória circular, no mesmo sentido e com velocidade constante, o que denominamos Alvos Convencionais (*Conventional Targets*).

Caso a interface da aplicação seja grande, como um monitor de computador ou uma televisão, e o rastreador utilizado seja previamente calibrado, existem soluções que permitem selecionar confiavelmente um alvo dentre muitos apresentados simultaneamente, como o sistema *DialPlates* proposto por Drewes et al. [6] que apresentou resultados satisfatórios, com taxas de acerto próximas a 80%, com até 160 alvos simultâneos.

Visando melhorar os resultados existentes até então, Velloso et al. [17] propuseram dois novos algoritmos denominados 2D Correlation (2D) e Profile Matching (PM) que calculam a similaridade entre os movimentos que o usuário realiza com os olhos e os movimentos dos alvos dentro de uma janela temporal de duração predeterminada, usualmente de 1s. Esses algoritmos têm como objetivos identificar quando o usuário está perseguindo um alvo e discriminar o alvo perseguido. Os desempenhos dos algoritmos propostos foram comparados com os desempenhos dos algoritmos Basic Correlation (BC), proposto para o sistema Pursuits por Vidal et al. [21, 22], e Rotated Correlation (RC), proposto para o sistema PathSync por Carter et al. [3]. Dentre os quatro algoritmos, o 2D Correlation apresentou os melhores resultados.

Outro trabalho que contém uma contribuição com potencial de melhorar o desempenho dos algoritmos que calculam similaridades entre movimentos é o algoritmo probabilístico proposto por Williamson e Smith [23, 24], elaborado para interfaces de aplicações para computadores baseadas em movimentos de mouse. Velloso e Morimoto [18] demonstram como essa abordagem probabilística pode reduzir as seleções involuntárias e aumentar a robustez na seleção dos alvos por SP quando os dados apresentam distorções de natureza espaço-temporais inerentes à captura feita pelo rastreador ocular sem calibração. No contexto da seleção por SP, o algoritmo probabilístico atua como uma camada de processamento posterior aos algoritmos que calculam similaridades: BC, RC, 2D e PM, fornecendo evidências mais fortes de que o usuário está ou não realizando uma perseguição, além de reduzir a incerteza acerca de qual alvo está sendo perseguido. Assim, optamos por adotar o algoritmo probabilístico, onde adaptamos a proposta original de Williamson e Smith, e o algoritmo 2D Correlation para a implementação da seleção de alvos no *Smart Targets*.

2.2 Interfaces para Seleção por Perseguição

Além das técnicas de seleção dos alvos por perseguição, o design da interface também tem grande influência na robustez para seleção de alvos por SP. Esse aspecto foi abordado em poucos trabalhos, por exemplo o sistema Smoovs, proposto por Lutz et al. [12]. Através de novas dinâmicas de movimentação dos alvos, é possível aumentar a quantidade de alvos na interface, todavia a proposta do Smoovs onera o tempo para seleção, pois consiste em agrupar múltiplos alvos fazendo com que cada grupo se mova como um “macroalvo”, desta forma o usuário precisa selecionar primeiro o grupo que contém o alvo desejado para depois selecionar o alvo propriamente.

Uma alternativa ainda não explorada no design de interfaces para interação por perseguição do olhar são as interfaces reativas ao olhar (*gaze contingent displays*) [7] capazes de se adaptarem dinamicamente ao olhar do usuário. Esse tipo de interface cria uma interação em malha fechada ao ajustar a quantidade de informação presente na região onde mais provavelmente o usuário está olhando em tempo-real, sendo uma das inspirações para a interface dos Smart Targets.

2.3 Navegação na Web pelo Olhar

Outro aspecto é a definição da tarefa realizada pelo usuário. A acessibilidade na web para pessoas com deficiências motoras graves é viável através da interação pelo olhar, permitindo a operação de interfaces web sem usar as mãos. Poucos trabalhos exploraram essa tarefa. Lutteroth et al. explorou este tipo de interação pela web através do *ActiGaze* [11]. Utilizando um algoritmo de distribuição de cores para cada hyperlink é possível diferenciá-los para evitar seleções involuntárias. Utilizando um tempo de permanência curto o usuário pode explorar hyperlinks e confirmar a seleção através do botão de confirmação na cor correspondente ao link desejado. Os botões de confirmação são apresentados num painel lateral à página web explorada.

Uma extensão do *Actigaze* é o trabalho desenvolvido por Figueiredo e Fonseca através do *EyeLinks* [9]. Esta proposta estende o tipo de objetos para qualquer objeto clicável dentro de uma página web e ao invés de cores utiliza numeração de objetos. Ambas as propostas apresentaram desempenhos semelhantes por tratar-se de uma técnica de seleção baseada em tempo de permanência e em duas etapas (exploração e confirmação).

Uma vez que dispomos das abordagens para o algoritmo de seleção dos alvos, a interface dos Smart Targets e a tarefa do usuário, desenvolvemos uma aplicação para avaliar seu desempenho e compará-lo ao dos Conventional Targets. Nesta avaliação, utilizamos um protocolo semelhante ao adotado por Carneiro et al.[2] para avaliação do PursuitPass, no qual o usuário deveria selecionar alvos por SP para inserir 4 dígitos previamente definidos, simulando os níveis de controle na navegação pela web. Porém, diferente desses autores que definiram uma sequência fixa para ser inserida, optamos por criar sequências aleatórias, o que corresponde a um cenário de uso mais corriqueiro, onde o usuário seleciona os objetos sem uma ordem determinada ou quantidade de hyperlinks dinâmicos que podem estar uma página web.

3 DESCRIÇÃO DOS SMART TARGETS

Propomos os Smart Targets como uma solução para seleção robusta de alvos por SP, que necessita de uma área pequena para exibição dos alvos, compatível com telas pequenas, e que não tenha o desempenho comprometido com o aumento na quantidade de alvos. Analisando o comportamento dos Conventional Targets durante uma perseguição, observamos que é possível que dois ou mais alvos apresentem valores elevados de similaridade durante a perseguição, em particular, quando alvos vizinhos estão dispostos muito próximos entre si, com distância inferior à precisão do sensor utilizado na captura dos dados do usuário (aproximadamente 1° de ângulo visual para a maioria dos rastreadores oculares comerciais). Assim, elaboramos um mecanismo de modulação das distâncias angulares entre os alvos que isola o alvo com maior probabilidade de perseguição dos demais por uma distância de 90° de seus adjacentes, esse mecanismo torna a seleção robusta ao assegurar que os movimentos descritos pelo alvo com maior probabilidade de perseguição são suficientemente diferentes para evitar a confusão na seleção do alvo perseguido.

3.1 Interface

Ao contrário dos alvos convencionais, que possuem comportamentos periódicos e independentes das ações do usuário conforme ilustrado na Figura 2, os Smart Targets são capazes de modificar os movimentos dos alvos em resposta às ações do usuário. Com isso, introduzimos possibilidades que não foram abordadas na literatura até então e conseguimos aumentar a robustez para discriminação do alvo perseguido e também aumentar a quantidade de alvos sem causar prejuízos à seleção.

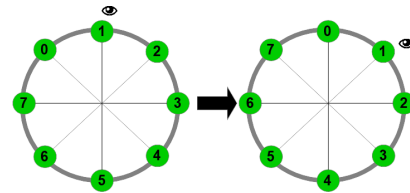


Figura 2: Comportamento dos alvos convencionais durante a perseguição.

De forma genérica, para selecionar um dentre múltiplos alvos por SP, o usuário realiza duas ações: busca e perseguição do alvo desejado. Durante a busca, o usuário faz breves perseguições para identificar a informação associada a cada alvo, por exemplo caracteres alfanuméricos. Por causa dessas breves perseguições, o risco de seleções involuntárias aumenta caso a janela temporal, usada para o cálculo da similaridade, seja pequena ou o usuário não tenha ciência de que um alvo brevemente perseguido está na iminência de ser selecionado. Já durante a perseguição após a localização do alvo desejado, o usuário realiza movimentos similares aos do alvo perseguido por um tempo maior para selecioná-lo. Assim, o tempo de perseguição é o que permite a diferenciação entre uma perseguição breve (para busca) e a perseguição para seleção.

A ambiguidade entre os movimentos descritos pelos alvos próximos ocorre quando o número de alvos impede uma separação

espacial maior que a incerteza das medidas provenientes do rastreador utilizado. Nesta situação, observa-se um novo desafio: distinguir exatamente o alvo perseguido. A ideia presente nos Smart Targets que facilita a discriminação do alvo desejado com maior robustez consiste em modular as distâncias angulares entre os alvos de modo a reduzir essa ambiguidade.

Assim, os alvos se organizam para aumentar a distância angular para o alvo supostamente perseguido durante a perseguição. Para isso propomos alterar dinamicamente o comportamento dos alvos uma vez que é identificado que o usuário está realizando uma perseguição. Em uma aplicação com muitos alvos (8 ou mais), os coeficientes de similaridade calculados pelos algoritmos de similaridade podem apresentar valores próximos para alvos adjacentes, além disso as distorções espaço-temporais e as limitações de precisão do sensor podem fazer com que o alvo com maior similaridade não seja exatamente o alvo perseguido. Então, além de introduzir a manipulação dos movimentos dos alvos, também é necessário que o usuário perceba qual alvo está para ser selecionado e possa decidir por manter ou corrigir a seleção.

O mecanismo proposto para modular as distâncias entre os alvos é uma função do acúmulo de evidências de perseguição associadas a cada alvo ao longo do tempo. Na presença de muita incerteza, consideramos os alvos igualmente espaçados entre si. Quando o sistema detecta alguma evidência que o usuário está realizando uma perseguição, os alvos com menor probabilidade alteram sua velocidade para aumentar a distância angular para o alvo com maior probabilidade \hat{i} . Este movimento dos alvos é denominado afastamento. A velocidade angular do alvo com maior probabilidade de perseguição $\omega_{\hat{i}}$ é mantida durante o afastamento.

O princípio de sobreposição de movimentos [10] foi utilizado para manipular as velocidades dos demais alvos, de modo que a distância angular destes para o alvo \hat{i} aumente até um limite previamente estabelecido de 90° para os alvos adjacentes ao \hat{i} e os demais acomodados nos 180° restantes. A Figura 3 ilustra como ocorre a sobreposição dos dois movimentos enquanto o alvo perseguido é isolado enquanto o usuário o persegue. Durante a perseguição, o alvo perseguido (indicado pela proximidade do olho na Figura 3) mantém a velocidade constante enquanto os demais alvos modificam suas velocidades para provocar o distanciamento.

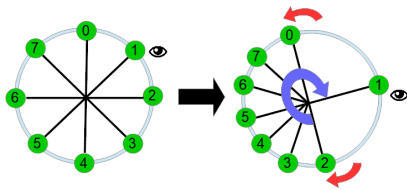


Figura 3: Movimentos sobrepostos realizados pelos Smart Targets para isolar um alvo durante a perseguição: movimento circular com velocidade angular constante (seta azul) e afastamento (setas vermelhas).

A distância de 90° entre \hat{i} e seus adjacentes foi adotada com base nos resultados obtidos para os Conventional Targets com 4 alvos simultâneos, que fornece baixa ocorrência média de erros (15% ou menor).

Para a definição dos parâmetros do movimento de afastamento, o alvo com maior probabilidade de perseguição \hat{i} foi usado como referência e a distância k ($k \geq 0$) foi definida para os alvos que estão a uma distância de k alvos de \hat{i} , por exemplo, os alvos adjacentes a \hat{i} apresentam $k = 1$, seus adjacentes diferentes de \hat{i} apresentam $k = 2$ e assim sucessivamente. Observe que $k = 0$ para \hat{i} . Durante a perseguição, a distância angular entre \hat{i} e os alvos $k = 1$ aumenta até 90° (máxima distância angular adotada), a distância angular entre os alvos $k = 1$ e $k = 2$ aumenta até 45° e os 90° restantes são divididos entre os outros alvos em uma progressão geométrica com razão $\frac{1}{2}$. Assim, garantimos que todos os alvos diferentes de \hat{i} estão acomodados no espaço angular de 180° centrado no ponto diametralmente oposto ao alvo \hat{i} . Esta disposição permite que o usuário mude a perseguição para os alvos vizinhos de \hat{i} com facilidade, já que estes estão a, no mínimo, 45° de distância angular dos seus adjacentes.

A variação da distância angular entre os alvos é feita linearmente, o que gera um efeito mais previsível para o usuário. A Figura 4 apresenta a variação da menor distância angular entre os alvos, em relação ao alvo com maior probabilidade de perseguição, ao longo do tempo para uma perseguição que é detectada em $t = 0,5$ s. A duração para o afastamento escolhida foi 1 s, assim o usuário tem tempo para perceber que algum alvo está sendo isolado e pode reagir de acordo com sua intenção, seja para interromper o isolamento do alvo, mudar o alvo perseguido, ou manter a perseguição para selecionar o alvo. Devido ao tempo de afastamento dos alvos, pode ser que o tempo por seleção seja pelo menos 1 s maior que a janela temporal para o cálculo da similaridade.

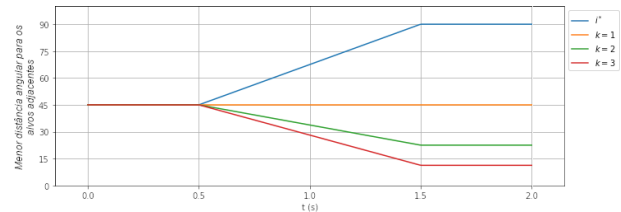


Figura 4: Variação da distância angular entre os alvos durante uma perseguição.

O movimento de afastamento é iniciado quando uma perseguição é detectada. Por isso, é fundamental que a detecção da perseguição seja robusta e não apresente erros enquanto os alvos com menores probabilidades de perseguição estão se afastando. De acordo com a literatura, o alvo perseguido pode ser identificado pelos algoritmos baseados em métricas de similaridade com alta acurácia, desde que a distância angular entre os alvos seja suficientemente grande [6, 8, 19, 20]. Assim, escolhemos aplicar os algoritmos baseados em similaridade, porém adicionamos o algoritmo probabilístico, que utiliza os coeficientes de similaridade para calcular probabilidades de perseguição para cada alvo, como uma camada adicional de processamento. As probabilidades de perseguição são utilizadas no cálculo da entropia que fornece evidências de que o usuário está realizando uma perseguição.

O algoritmo probabilístico melhora significativamente o desempenho dos Smart Targets uma vez que, além do erro de discriminação, também há os erros causados por seleções involuntárias e

por perseguições não identificadas. Esses erros ocorrem, respectivamente, quando o algoritmo usado para seleção do alvo perseguido confunde movimentos realizados ocasionalmente pelo usuário com os movimentos de um alvo, gerando os falsos positivos, e quando os dados estão afetados, por exemplo pelas distorções espaço-temporais, de modo que os algoritmos que calculam métricas de similaridade não são capazes de identificar os padrões que caracterizam a perseguição, gerando os falsos negativos. A seguir descrevemos como o algoritmo probabilístico proposto por Williamson e Smith [24] foi utilizado para tornar a seleção de alvos por SP robusta a esses erros.

3.2 Algoritmo probabilístico

Como mostrado na Seção 2, os principais algoritmos de seleção de alvos por SP se baseiam no cálculo de alguma medida de similaridade entre os movimentos de cada um dos alvos e os movimentos descritos pelo usuário. Quando as medidas associadas a um ou mais alvos se tornam maiores que um determinado limiar, o alvo com maior similaridade é selecionado.

A ideia de usar um algoritmo probabilístico é acumular as evidências de perseguição até que a incerteza sobre a intenção do usuário seja baixa o suficiente para selecionar um dos alvos, o que pode adiar a decisão de seleção em teoria. O algoritmo descrito a seguir foi inspirado pelos trabalhos de Williamson e Murray-Smith [23, 24] e Velloso e Morimoto. [18], e adaptado para seleção de alvos por SP.

Como evidência de que o usuário está perseguido um alvo, vamos assumir que as medidas de similaridade são proporcionais às probabilidades de perseguição $p_i(t)$ associadas a cada um dos N alvos e estão sujeitas a incertezas. O conjunto de probabilidades $P = \{p_i(t), 1 \leq i \leq N\}$ que indicam a probabilidade do usuário estar perseguido cada um dos alvos no instante t é definido para as manipulações algébricas apresentadas para os Smart Targets. O cálculo das probabilidades $p_i(t)$ depende do acúmulo de evidências de que o usuário está perseguido um alvo, o que reduz a ocorrência de seleções involuntárias, porém pode introduzir um atraso maior para que um alvo seja selecionado no caso de múltiplos alvos apresentarem altas similaridades com o movimento realizado pelo usuário.

Inicialmente (em $t = 0$), a probabilidade de perseguição de cada alvo é a mesma, i.e., $p_i(0) = 1/N$ para todo alvo i . Esses valores iniciais de probabilidade são mantidos enquanto $t < T_w$ (T_w corresponde à duração da janela temporal para o cálculo da similaridade entre os movimentos dos alvos e do usuário). Passado um tempo correspondente à janela T_w , os coeficientes de similaridade podem ser utilizados para atualizar as probabilidades. Para as probabilidades de perseguição em $t \geq T_w$, uma variável intermediária $\gamma_i(t)$ é definida a partir das probabilidades $p_i(t-1)$ e dos coeficientes de similaridade $s_i(t)$ como

$$\gamma_i(t) = \begin{cases} \alpha s_i(t) + p_i(t-1), & \text{se } s_i(t) > \lambda; \text{ e} \\ \beta s_i(t) p_i(t-1), & \text{caso contrário.} \end{cases} \quad (1)$$

Em seguida, os valores de $\gamma_i(t)$ são corrigidos para um valor positivo e próximo de 0 nos casos em que $s_i(t)$ é negativo. Por fim, os valores de $\gamma_i(t)$ são normalizados para fornecerem as probabilidades $p_i(t)$ conforme a equação a seguir

$$p_i(t) = \frac{\gamma_i(t)}{\sum_{j=1}^N \gamma_j(t)}. \quad (2)$$

As constantes α e β ($0 < \alpha, \beta < 1$) utilizadas na definição de $\gamma_i(t)$, assim como a constante λ que corresponde a um limiar aplicado sobre o coeficiente de similaridade $s_i(t)$, permitem controlar a relação entre a robustez e a velocidade da seleção. O ajuste das constantes α e β permite controlar a velocidade de variação das probabilidades $p_i(t)$, reduzindo o tempo para seleção ou tornando as seleções mais demoradas, o que pode aumentar a robustez às mudanças de movimento do usuário. Já a constante λ atua como limiar sobre o valor $s_i(t)$ para elevar ou diminuir $p_i(t)$. Esse limiar é diferente do limiar de similaridade adotado por outros autores [3, 4, 8, 17, 19, 20] uma vez que λ não determina diretamente quando o usuário está perseguido um alvo, mas um mínimo de similaridade a ser observada para elevar as probabilidades $p_i(t)$.

Para identificar quando o usuário está perseguido um alvo, Williamson e Murray-Smith utilizam um limiar $L_{\mathcal{H}}$ sobre a entropia do conjunto P [24]. A partir da definição de entropia:

$$\mathcal{H}(P) = - \sum_{i=1}^N p_i(t) \log_2(p_i(t)), \quad (3)$$

percebe-se que a máxima incerteza para seleção de um dentre N alvos ocorre nos instantes em que todos os alvos apresentam a mesma probabilidade de perseguição, i.e., $p_i(t) = \frac{1}{N} \forall i$, o que fornece o valor máximo de entropia $\mathcal{H}(P) = \log_2(N)$. Na medida em que os valores do conjunto $P = \{p_i(t), 1 \leq i \leq N\}$ são atualizados durante uma perseguição, as probabilidades do alvo perseguido e/ou de seus adjacentes aumentam enquanto as demais diminuem, o que reduz a incerteza e consequentemente a entropia. Assim, uma opção para identificar quando o usuário está perseguido um alvo é verificar se $\mathcal{H}(P) < L_{\mathcal{H}}$, para um certo limiar $L_{\mathcal{H}}$.

A aplicação deste limiar sobre o valor de entropia permite a identificação do instante em que há evidência suficiente para caracterizar uma perseguição, assim podemos determinar o alvo perseguido \hat{i} como aquele com maior probabilidade de perseguição

$$\hat{i} = \arg \max_{\forall i} \{p_i(t)\}. \quad (4)$$

Com isso, concluímos a descrição dos Smart Targets e seguimos com a condução de um experimento com voluntários para avaliação dos Smart Targets e comparação de seu desempenho com os Conventional Targets.

4 EXPERIMENTO PARA AVALIAÇÃO DOS SMART TARGETS

Para o experimento de avaliação dos Smart Targets, desenvolvemos uma aplicação que implementa a interface responsiva em uma área pequena e o algoritmo probabilístico para seleção de alvos por SP. Conduzimos um teste preliminar com integrantes do nosso grupo de pesquisa no qual coletamos dados referentes a perseguições para definição dos parâmetros do algoritmo probabilístico, onde obtemos os valores $\alpha = 0,8$ e $\beta = 0,5$, que foram definidos de modo que a entropia $\mathcal{H}(P)$ mantenha-se estável, com valor próximo de

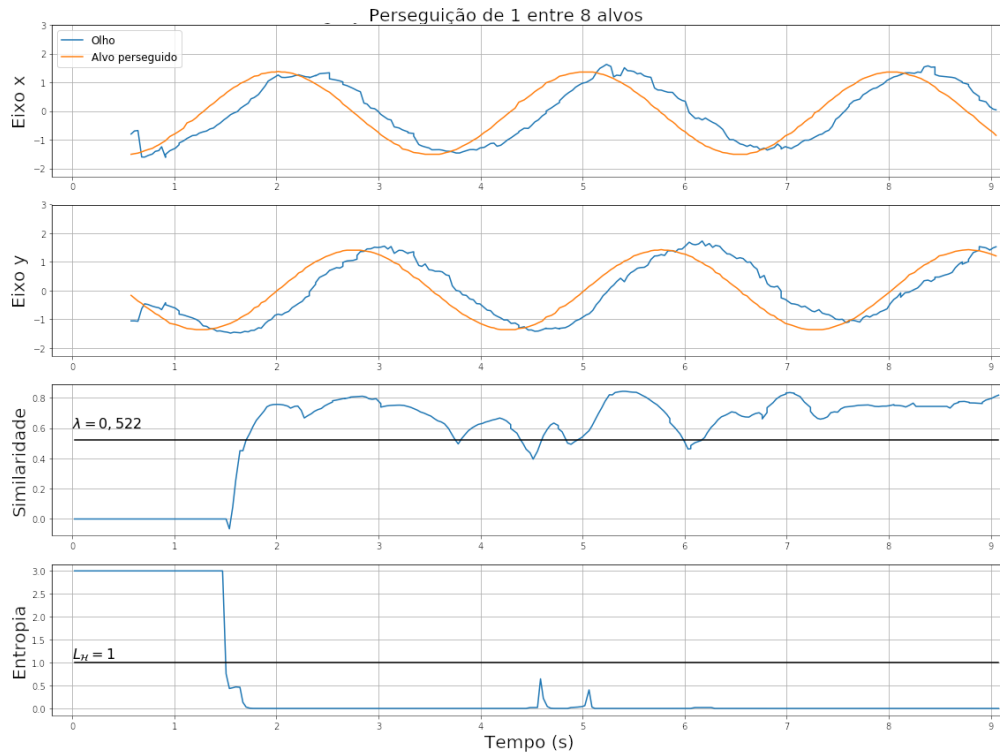


Figura 5: Exemplo de dados capturados no teste preliminar com a respectiva entropia calculada com 8 alvos usando os parâmetros $\alpha = 0,8$ e $\beta = 0,5$, e o algoritmo 2D com $\lambda = 0,522$.

0 (e abaixo do limiar $L_H = 1$) durante a maioria das perseguições, mesmo com pequenas distorções observadas nos dados capturados pelo rastreador ocular.

A Figura 5 ilustra um exemplo usando dados coletados no teste preliminar e os parâmetros apresentados onde é possível observar valores de similaridade abaixo do limiar $\lambda = 0,522$, definido a partir da métrica VP10, adaptada da VP5 utilizada por Velloso et al. [17]. A métrica VP10 corresponde à maior taxa de verdadeiros positivos (VP) tal que a taxa de falsos positivos (FP) esteja limitada ao máximo de 10%, assim definimos o limiar com base em VP10, o que fornece a maior taxa de verdadeiros positivos tal que a taxa de falsos positivos seja menor ou igual à 10%. O valor de λ escolhido produziu valores de entropia dentro do intervalo $[0, 1]$ durante todo o período de perseguições, o que podemos interpretar como: a incerteza para detectar a perseguição se manteve abaixo de 2 alvos, caracterizada por $\mathcal{H}(P) = 1$, reforçando as evidências de perseguição. O uso da entropia em vez da similaridade torna a seleção mais robusta como podemos observar nos gráficos de similaridade e entropia na Figura 5, onde os valores de similaridade associados ao alvo perseguido apresentam maiores variações durante a perseguição, ficando abaixo do limiar por 4 vezes, enquanto a entropia mantém-se abaixo de 1.

4.1 Protocolo experimental

Uma vez que os parâmetros do algoritmo probabilístico foram obtidos a partir dos dados coletados no teste preliminar, definimos a

velocidade dos alvos como $60^\circ/s$ e, além da interface descrita na Seção 3, também introduzimos uma barra de progresso nos alvos indicando o tempo que falta para o alvo perseguido ser selecionado. Essa barra de progresso foi exibida sempre que uma perseguição foi detectada para permitir aos voluntários corrigir seus movimentos durante as seleções.

Para a condução do experimento, utilizamos um rastreador remoto Pupil, desenvolvido pela empresa Pupil-Labs, configurado para captura de dados a 120Hz, montado na cabeça dos voluntários. Os dados referentes ao rastreamento dos olhos foram coletados através do software desenvolvido pelo fabricante do rastreador. O rastreador foi conectado a um computador desktop que dispõe de dois monitores posicionados lado a lado com altura de 22 cm da superfície de apoio, de modo que o centro do monitor esteja aproximadamente alinhado com os olhos dos voluntários. Um dos monitores foi usado para exibir a interface da aplicação desenvolvida para o experimento, ilustrada na Figura 7, e o outro para exibir a sequência de 4 caracteres sorteados aleatoriamente que os voluntários deveriam inserir. Utilizando os dois monitores, pudemos reproduzir um cenário mais próximo do uso real de uma navegação na web com o EyePursuitLinks, em que o usuário alterna sua atenção entre o conteúdo da página web e os alvos móveis.

Tivemos a colaboração de 10 voluntários (4 mulheres), com idades entre 21 e 38 anos (média = 31,1 anos) e sem deficiências motoras, sendo alunos de graduação e pós-graduação oriundos da instituição

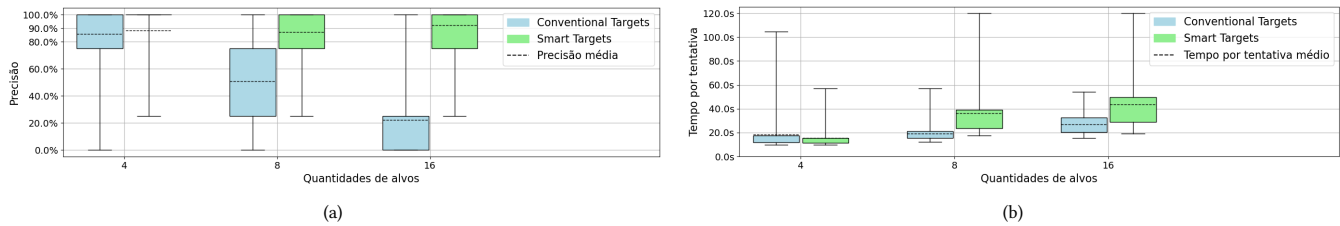


Figura 6: Resultados obtidos quanto à (a) precisão por caracter e (b) tempo para inserção.

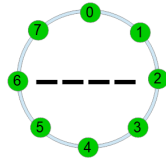


Figura 7: Interface da aplicação desenvolvida para a avaliação.

a qual somos vinculados. Apenas 1 voluntário relatou ter experiência prévia com o uso de rastreadores oculares, sendo também o único que utiliza lentes para correção da visão.

A participação de cada voluntário no experimento iniciou com a explicação do experimento e seus objetivos, em seguida os voluntários foram posicionados em uma cadeira de modo que os rostos ficassem a aproximadamente 40 cm de distância do monitor onde a interface da aplicação era exibida, de modo que o tamanho da interface corresponde a 3° de ângulo visual o que equivale ao tamanho de uma tela de relógio inteligente. Decidimos não utilizar apoios de queixo ou cabeça para avaliar o desempenho da solução com um cenário mais desafiador, onde o usuário pode mover a cabeça enquanto interage com a aplicação.

Então, os voluntários vestiram o rastreador ocular e realizaram duas inserções usando 4 alvos para familiarização com a aplicação, em seguida cada voluntário teve 15 tentativas de inserir os 4 caracteres utilizando tanto os Conventional Targets como os Smart Targets divididas da seguinte forma: para as primeiras 5 tentativas, foram exibidos 4 alvos, para as 5 tentativas seguintes foram exibidos 8 alvos e para as 5 últimas foram exibidos 16 alvos, simulando a navegação com 4, 8 e 16 links em uma página web. Decidimos usar essa ordem crescente na quantidade de alvos para que a experiência adquirida no cenário mais simples, com menos alvos, pudesse contribuir para os cenários mais difíceis, com mais alvos. Cada participante levou aproximadamente 40 minutos para realizar as tarefas propostas, incluindo um período de descanso entre as tarefas.

Após realizarem todas as tentativas de inserção, os voluntários responderam um formulário com informações pessoais: sexo, idade, experiência prévia com rastreadores oculares e uso de lentes ou óculos para correção da visão, e suas impressões acerca do experimento e uma comparação qualitativa entre os Conventional Targets e os Smart Targets. Este experimento foi avaliado e aprovado pelo comitê de ética da EACH/USP (parecer¹: 5693728).

¹Este parecer pode ser verificado na Plataforma Brasil <https://plataformabrasil.saude.gov.br/>

Com este experimento, coletamos os seguintes dados referentes ao desempenho dos Conventional Targets e dos Smart Targets: sequências que os voluntários deveriam inserir, sequências que foram inseridas e o tempo que cada tentativa levou para ser concluída. A partir desses dados, obtivemos os resultados apresentados na próxima seção.

5 RESULTADOS

A partir dos dados de desempenho coletados no experimento descrito na seção anterior, obtivemos os seguintes resultados para os Conventional Targets e os Smart Targets separados por quantidade de alvos exibidos: precisão por caracteres, calculada a partir da distância de Levenshtein entre a sequência apresentada e a sequência inserida em cada tentativa, taxa de erro por sequência, calculada a partir das sequências inseridas que têm ao menos 1 erro, e tempo por tentativa de inserção.

Os gráficos das Figuras 6(a) e 6(b) contêm, respectivamente, os resultados referentes às precisões por caracter inserido e aos tempos por tentativa de inserção obtidos pelos voluntários. Esses gráficos correspondem aos diagramas de caixa (*boxplots*), em que cada caixa mostra os valores máximo e mínimo, os limites do 2° e 3° quartis como retângulo colorido, e os valores de média (linha pontilhada). O intervalo de valores que compreende o 2° e 3° quartis é denominado intervalo interquartil (*IQR*, *Interquartile Range*), seguindo a definição original proposta por Tukey et al. [5, 15].

A Tabela 1 apresenta os resultados médios gerais obtidos no experimento quanto às métricas avaliadas. Análises ANOVA de uma via foram conduzidas para comparar os efeitos dos Smart Targets sobre os Conventional Targets e da quantidade de alvos nos resultados de desempenho obtidos. Observamos que o desempenho quanto à precisão dos Smart Targets não sofre variações significativas com o aumento da quantidade de alvos ($F(147, 2) = 0,841, p = 0,433$) sendo que a maior média de precisão $M = 92\%$ ($STD = 14,7\%$) foi obtida com os Smart Targets em uma interface com 16 alvos, o que revela o efeito do aprendizado dos voluntários durante o experimento. Comparando o desempenho quanto à precisão dos Conventional Targets com os Smart Targets, observamos que não há diferenças significativas para 4 alvos ($F(98, 1) = 0,259, p = 0,612$), porém as diferenças são significativas para 8 ($F(98, 1) = 44,46, p < 0,001$) e 16 ($F(98, 1) = 266,18, p < 0,001$) alvos.

As taxas de sequências erradas indicam a concentração dos erros ao longo das tentativas de inserção, por exemplo, foram obtidos 24% de sequências erradas com os Smart Targets com 4 alvos na interface, porém a precisão foi de 88%, isso significa que maioria das

Tabela 1: Resultados gerais do experimento para os Conventional Targets e os Smart Targets.

Quantidades de alvos	Métrica	Conventional Targets	Smart Targets
4	Precisão por tentativa / Percentual de erros	M = 85,5%, STD = 25,8% / 30%	M = 88%, STD = 23,3% / 24%
	Tempo por tentativa	M = 18,08 s, STD = 15,21 s	M = 15,49 s, STD = 8,29 s
8	Precisão por tentativa / Percentual de erros	M = 50,5%, STD = 31,7% / 90%	M = 87%, STD = 22,2% / 30%
	Tempo por tentativa	M = 19,24 s, STD = 7,01 s	M = 35,97 s, STD = 20,44 s
16	Precisão por tentativa / Percentual de erros	M = 22%, STD = 26,6% / 96%	M = 92%, STD = 14,7% / 28%
	Tempo por tentativa	M = 26,59 s, STD = 8,87 s	M = 43,45 s, STD = 22,74 s

sequências com erros apresenta apenas 1 caracter errado. Por outro lado, os altos percentuais de sequências erradas e a baixa precisão obtidos com os Conventional Targets para 8 e 16 alvos reforçam que esta solução tem limitações para aplicações com muitos alvos.

Já para os tempos por tentativa, observamos que não há diferença significativa entre os resultados obtidos para os Conventional Targets e os Smart Targets com 4 alvos ($F(98, 1) = 1, 12, p = 0, 292$). Mas há uma diferença significativa entre os resultados obtidos com os Conventional Targets com o aumento do número de alvos ($F(147, 2) = 8, 89, p < 0, 001$), porém o teste post-hoc Tukey HSD mostrou que esta diferença somente é significativa entre os resultados com 16 alvos e os demais. Analisando o efeito da quantidade de alvos sobre os tempos por tentativa obtidos com os Smart Targets, observamos também diferenças significativas ($F(147, 2) = 31, 30, p < 0, 001$), todavia, o teste post-hoc Tukey HSD mostrou que esta diferença somente é significativa entre os resultados com 4 alvos e os demais. Os tempos por tentativa de inserção com os Conventional Targets quando há 8 e 16 alvos na interface são significativamente mais baixos que os tempos obtidos com os Smart Targets uma vez que eles não apresentam o movimento de afastamento dos alvos e muitas vezes ocorrem seleções indesejadas, o que é reforçado pelas altas taxas de erros observadas.

Os Smart Targets não apresentam movimento de afastamento quando há 4 alvos na interface, dado que a distância angular entre eles é de 90° . Assim, é esperado que o tempo para seleção aumente quando a quantidade de alvos é maior que 4 devido ao tempo de acomodação dos alvos com as menores probabilidades de perseguição. Todavia, observamos não haver diferenças significativas no tempo por tentativa de inserção com os Smart Targets para 8 e 16 alvos ($F(98, 1) = 2, 98, p = 0, 087$), sugerindo que os Smart Targets também são robustos quanto ao tempo por tentativa quando a quantidade de alvos é superior a 4. Contudo, o experimento que conduzimos não é suficiente para concluir que o tempo por tentativa será sempre semelhante para qualquer quantidade de alvos.

Quanto à avaliação qualitativa feita com os voluntários ao final do experimento, os voluntários utilizaram uma Escala de Likert com valores de 1 à 5 para reportar a facilidade percebida no uso dos Conventional Targets e dos Smart Targets. A avaliação média dos Conventional Targets foi de 3,5 com mediana de 3 enquanto os Smart Targets obtiveram média de 4,6 com mediana de 5. Os participantes puderam registrar comentários, onde relataram como principal vantagem dos Smart Targets a facilidade em verificar e corrigir o alvo perseguido durante o afastamento, o que não é possível fazer com os Conventional Targets. Os participantes não relataram quaisquer desconfortos durante a realização do experimento.

6 CONCLUSÃO

Neste trabalho, propomos o EyePursuitLinks, uma técnica de interação que permite a seleção de hyperlinks usando apenas o olhar. A técnica é baseada em movimentos de perseguição do olhar, que permitem a interação sem a necessidade de calibração do rastreador de olhar. A grande contribuição desse trabalho está na solução denominada Smart Targets, que permite um aumento no número de alvos sem aumentar a área reservada à interface nem prejuízos quanto à precisão na seleção dos alvos. Para isso, introduzimos a ideia de modulação dos movimentos dos alvos e implementamos uma interface de seleção responsiva, que se modifica a partir dos movimentos oculares realizados pelo usuário, e um algoritmo probabilístico que permite a seleção de um dentre muitos alvos por meio de SP. Através da modulação dos movimentos, é possível modificar as distâncias angulares entre os alvos de modo a isolar o alvo perseguido, reduzindo as ocorrências de erros. Os Smart Targets foram avaliados em um experimento com 10 voluntários e seu desempenho foi comparado com os Conventional Targets, onde a interface não é responsiva.

Os resultados obtidos com o experimento mostram que os Smart Targets permitem uma seleção robusta em páginas web que não é afetada pela quantidade de alvos presentes na interface, visto que os voluntários conseguiram o melhor desempenho médio com a maior quantidade de alvos avaliada, sugerindo um efeito de aprendizagem ao longo do tempo que contribuiu para um melhor desempenho na medida em que os voluntários ganharam mais experiência. Por outro lado, o desempenho dos Conventional Targets decaiu significativamente com o aumento da quantidade de alvos na interface o que acontece devido à maior semelhança entre os movimentos de alvos adjacentes na medida em que a quantidade de alvos aumenta. Além do melhor desempenho quanto à precisão na seleção dos alvos, os voluntários reportaram que a interface dos Smart Targets é mais fácil de usar para a seleção de alvos por Smooth Pursuit. Em trabalhos futuros, o desempenho dos Smart Targets será avaliado considerando pessoas com deficiência motora dentre os voluntários.

AGRADECIMENTOS

Este trabalho recebeu suporte financeiro da Fundação para a Pesquisa do Estado de São Paulo, projetos 2016/10148-3 e 2017/50121-0. O segundo autor recebeu suporte financeiro da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – código 88882.328178/2018-01 e do programa Google Latin American Research Awards (LARA).

REFERÊNCIAS

- [1] M. Cecilia Baranauskas, Clarisse de Souza, and Roberto Pereira. 2014. *I GranDIHC-BR — Grandes Desafios de Pesquisa em Interação Humano-Computador no Brasil*. <https://doi.org/10.13140/2.1.3651.9201>
- [2] Alex Torquato S. Carneiro, Carlos Eduardo L. Elmadjian, Candy Gonzales, Flavio L. Coutinho, and Carlos H. Morimoto. 2019. PursuitPass: A Visual Pursuit-Based User Authentication System. In *2019 32nd SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*. IEEE, Brazil, 226–233. <https://doi.org/10.1109/SIBGRAPI.2019.00038>
- [3] Marcus Carter, Eduardo Velloso, John Downs, Abigail Sellen, Kenton O'Hara, and Frank Vetere. 2016. PathSync: Multi-User Gestural Interaction with Touchless Rhythmic Path Mimicry. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (San Jose, California, USA) (*CHI '16*). ACM, New York, NY, USA, 3415–3427. <https://doi.org/10.1145/2858036.2858284>
- [4] Christopher Clarke, Alessio Bellino, Augusto Esteves, Eduardo Velloso, and Hans Gellersen. 2016. TraceMatch: a computer vision technique for user input by tracing of animated controls. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. 298–303.
- [5] Robert Dawson. 2011. How significant is a boxplot outlier? *Journal of Statistics Education* 19, 2 (2011).
- [6] Heiko Drewes, Mohamed Khamis, and Florian Alt. 2019. DialPlates: enabling pursuits-based user interfaces with large target numbers. In *Proceedings of the 18th International Conference on Mobile and Ubiquitous Multimedia*. 1–10.
- [7] Andrew T Duchowski, Nathan Cournia, and Hunter Murphy. 2004. Gaze-contingent displays: A review. *CyberPsychology & Behavior* 7, 6 (2004), 621–634.
- [8] Augusto Esteves, Eduardo Velloso, Andreas Bulling, and Hans Gellersen. 2015. Orbits: Gaze interaction for smart watches using smooth pursuit eye movements. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*. ACM, 457–466.
- [9] Pedro Figueiredo and Manuel J. Fonseca. 2018. EyeLinks: A Gaze-Only Click Alternative for Heterogeneous Clickables. In *Proceedings of the 20th ACM International Conference on Multimodal Interaction* (Boulder, CO, USA) (*ICMI '18*). Association for Computing Machinery, New York, NY, USA, 307–314. <https://doi.org/10.1145/3242969.3243021>
- [10] David Halliday, Robert Resnick, and Jearl Walker. 2016. *Fundamentos de física. Vol. 1*. Grupo Gen-LTC.
- [11] Christof Lutteroth, Moiz Penkar, and Gerald Weber. 2015. Gaze vs. Mouse: A Fast and Accurate Gaze-Only Click Alternative. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology* (Charlotte, NC, USA) (*UIST '15*). Association for Computing Machinery, New York, NY, USA, 385–394. <https://doi.org/10.1145/2807442.2807461>
- [12] Otto Hans-Martin Lutz, Antje Christine Venjakob, and Stefan Ruff. 2015. SMO-OVS: Towards calibration-free text entry by gaze using smooth pursuit movements. *Journal of Eye Movement Research* 8, 1 (2015), 1–11.
- [13] P. Majaranta, U.K. Ahola, and O. Špakov. 2009. Fast gaze typing with an adjustable dwell time. In *Proceedings of the 27th international conference on Human factors in computing systems (CHI '09)*. ACM, New York, NY, 357–360. <https://doi.org/10.1145/1518701.1518758>
- [14] Katharina Reiter, Ken Pfeuffer, Augusto Esteves, Tim Mittermeier, and Florian Alt. 2022. Look & Turn: One-handed and Expressive Menu Interaction by Gaze and Arm Turns in VR. In *2022 Symposium on Eye Tracking Research and Applications*. 1–7.
- [15] John W Tukey et al. 1977. *Exploratory data analysis*. Vol. 2. Reading, Mass.
- [16] JJ Van Middendorp, F Watkins, C Park, and H Landymore. 2015. Eye-tracking computer systems for inpatients with tetraplegia: findings from a feasibility study. *Spinal cord* 53, 3 (2015), 221–225.
- [17] Eduardo Velloso, Flavio Luiz Coutinho, Andrew Kurauchi, and Carlos H Morimoto. 2018. Circular orbits detection for gaze interaction using 2D correlation and profile matching algorithms. In *Proceedings of the 2018 ACM Symposium on Eye Tracking Research & Applications*. ACM, 25.
- [18] Eduardo Velloso and Carlos H Morimoto. 2021. A probabilistic interpretation of motion correlation selection techniques. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 1–13.
- [19] David Verweij, Augusto Esteves, Vassilis-Javed Khan, and Saskia Bakker. 2017. WaveTrace: motion matching input using wrist-worn motion sensors. In *Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems*. 2180–2186.
- [20] Mélodie Vidal, Andreas Bulling, and Hans Gellersen. 2013. Pursuits: Spontaneous Interaction with Displays Based on Smooth Pursuit Eye Movement and Moving Targets. In *Proceedings of the 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing* (Zurich, Switzerland) (*UbiComp '13*). ACM, New York, NY, USA, 439–448. <https://doi.org/10.1145/2493432.2493477>
- [21] Mélodie Vidal, Andreas Bulling, and Hans Gellersen. 2015. Pursuits: Spontaneous eye-based interaction for dynamic interfaces. *GetMobile: Mobile Computing and Communications* 18, 4 (2015), 8–10.
- [22] Mélodie Vidal, Ken Pfeuffer, Andreas Bulling, and Hans W Gellersen. 2013. Pursuits: eye-based interaction with moving targets. In *CHI '13 Extended Abstracts on Human Factors in Computing Systems*. 3147–3150.
- [23] John Williamson. 2006. *Continuous uncertain interaction*. Ph. D. Dissertation. University of Glasgow (United Kingdom).
- [24] John Williamson and Roderick Murray-Smith. 2004. Pointing without a pointer. In *CHI '04 Extended Abstracts on Human Factors in Computing Systems*. ACM, 1407–1410.