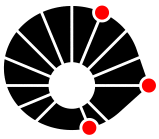

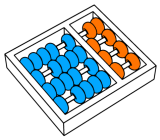

Controle de Mouse Baseado em Rastreamento Ocular

F.H. Matoba *H. Pedrini*

Relatório Técnico - IC-PFG-22-31
Projeto Final de Graduação
2022 - Dezembro

UNIVERSIDADE ESTADUAL DE CAMPINAS
INSTITUTO DE COMPUTAÇÃO

Controle de Mouse Baseado em Rastreamento Ocular

Felipe Hideki Matoba* Hélio Pedrini*

Resumo

Este projeto visa investigar e aplicar técnicas de visão computacional e de aprendizado de máquina para a detecção e o rastreamento dos olhos por meio de uma câmera de vídeo, além de oferecer um protótipo para o controle do mouse a partir do olhar. O protótipo pode auxiliar usuários com deficiência motora a interagir com o computador, dispensando a necessidade de equipamento especializado. Diferentes abordagens de visão computacional e aprendizado de máquina são avaliadas e aprimoradas, buscando-se melhorar a precisão do rastreamento ocular, sem a necessidade de ajustes manuais. Vários desafios estão associados ao problema, tais como a ocorrência de grandes movimentos da cabeça e condições do ambiente.

1 Introdução

O avanço cada vez mais acelerado da tecnologia que vem ocorrendo nos últimos anos tem gerado uma necessidade crescente da inclusão de pessoas tanto à comunidade de usuários das mais variadas tecnologias quanto ao mercado de trabalho. Vários equipamentos e métodos foram criados para auxiliar os usuários à utilização de aparelhos eletrônicos, tais como o Talk Back [1] para deficientes visuais que utilizam o sistema Android ou o VoiceOver [2] para usuários do sistema iOS, assim como diversas outras ferramentas para auxiliar diferentes tipos de deficiências.

Há ainda muitos desafios que precisam ser enfrentados para que pessoas com deficiência motora consigam interagir adequadamente com o ambiente em que vivem, incluindo o ambiente digital. A existência de equipamentos especializados permite que alguns usuários interajam com o computador de maneira compatível com suas necessidades, utilizando os pés por exemplo, mas nem sempre isso é possível ou acessível para as pessoas.

O rastreamento ocular oferece um meio alternativo para possibilitar a interação do usuário com o computador. Apesar de comumente estar relacionado a equipamentos caros e sofisticados disponíveis no mercado, várias tentativas surgiram para realizar o rastreamento utilizando somente uma câmera, de modo a facilitar o acesso a essa tecnologia.

Sendo assim, este projeto visa investigar diferentes técnicas automáticas para a detecção dos olhos e o rastreamento ocular por meio de uma câmera de vídeo para auxiliar usuários com alguma deficiência motora a interagir com seu ambiente. O movimento dos olhos e a fixação do olhar serão explorados pelo sistema para permitir interação do usuário com o

*Instituto de Computação, Universidade Estadual de Campinas, Campinas, SP, 13083-852.

equipamento. Algumas características desejáveis do protótipo a ser desenvolvido incluem baixo custo, fácil manuseio e rápida adaptação.

Este texto está organizado da seguinte forma. A Seção 2 traz alguns conceitos importantes para a realização deste trabalho. A Seção 3 descreve brevemente alguns trabalhos relacionados ao tema presentes na literatura. A Seção 4 descreve o método desenvolvido para a detecção dos olhos, o rastreamento ocular e o controle do mouse. A Seção 5 apresenta o protótipo desenvolvido. Finalmente, a Seção 6 apresenta as considerações finais deste projeto.

2 Conceitos

Esta seção descreve brevemente alguns conceitos de processamento de imagens e visão computacional utilizados no desenvolvimento do trabalho.

2.1 Processamento de Imagens

Nem sempre analisar as imagens brutas é suficiente ou adequado, então ao longo dos anos foram desenvolvidas diversas técnicas para processar as imagens e extrair informações ou prepará-las para a aplicação de outros algoritmos. Serão introduzidas a seguir algumas dessas técnicas de processamento de imagem que foram utilizadas neste projeto.

2.1.1 Filtro Bilateral

Utilizado principalmente para remoção de ruído em imagens, o filtro bilateral é similar ao filtro gaussiano, mas possui a vantagem de preservar as bordas da imagem, que ficam borradas no caso gaussiano. Isso é alcançado por considerar, além da proximidade espacial, a diferença de valor entre os pixels analisados, o que Tomasi e Manduchi [11] chamaram de *domain filtering* e *range filtering*, respectivamente, ao propor o método em seu trabalho.

2.1.2 Abertura Morfológica

A abertura morfológica consiste de uma erosão seguida de uma dilatação. Ambas funcionam passando um *kernel* de tamanho arbitrário pela imagem. Na primeira, um pixel da imagem só permanece como 1 caso todos os pixels no *kernel* centrado nele também sejam 1; caso contrário, eles são alterados para zero. A segunda operação faz o contrário, transformando um pixel em 1 caso ao menos um pixel do *kernel* também seja 1.

Efetivamente, uma abertura pode ser utilizada para remoção de ruído branco da imagem, pois pixels de valor 1 são removidos pela erosão, enquanto as bordas que foram prejudicadas são posteriormente restauradas pela dilatação.

2.1.3 Cálculo do Centroide

O centroide de um *blob* em uma imagem binária pode ser facilmente obtido a partir dos momentos M_{ij} de uma imagem I, dado pela fórmula:

$$M_{ij} = \sum_x \sum_y x^i y^j I(x, y) \tag{1}$$

No caso de uma imagem binária, em que cada pixel $I(x, y)$ é igual a um ou zero, tem-se que os momentos M_{10} e M_{01} correspondem à soma das posições x e y, respectivamente. Além disso, o momento M_{00} é equivalente a contar o número de pixels não-nulos. Assim, o centroide C é dado pelo ponto $(\frac{M_{10}}{M_{00}}, \frac{M_{01}}{M_{00}})$, que é igual a posição média dos pixels não-nulos da imagem em cada eixo.

2.2 Visão Computacional

Visão computacional pode ser entendida como uma área de estudo que busca desenvolver métodos que possibilitem computadores de analisar, identificar e entender o conteúdo de imagens, similarmente a como um ser humano o faria. Ela utiliza amplamente desde métodos de processamento de imagem a complexas redes neurais para classificação de objetos [12].

2.2.1 Detecção de Faces

Um ramo muito estudado da visão computacional é o de detecção de faces, possuindo diversas aplicações, como monitoramento de multidões, reconhecimento facial e identificação de partes do rosto. Dalal e Triggs [4] mostraram em seu trabalho a eficácia da utilização de histogramas de gradientes orientados (*Histograms of Oriented Gradients* - HOG) para obter *features* da imagem e usá-las em classificadores. Desde então, a ideia tem sido aplicada para outros objetivos, incluindo a detecção de rostos, como foi implementado na biblioteca dlib[1], utilizada neste trabalho.

2.2.2 Detecção de Pontos Fiduciais da Face

Pontos fiduciais da face (*face landmarks*) são pontos chaves que caracterizam um rosto, muito utilizados no problema de alinhmanento facial (*face alignment*), que busca identificar a estrutura geométrica de faces. Há vários modelos diferentes de pontos fiduciais que utilizam quantidades variadas de pontos, sendo que, para os propósitos deste trabalho, foi suficiente um que usa 68 pontos no total, o qual pode ser encontrado na base de dados iBUG 300-W [10].

Para a detecção propriamente dita, Kazemi e Sullivan [6] propuseram um conjunto de árvores de regressão em seu trabalho. A regressão é feita em uma abordagem iterativa, em cascata, em que *features* extraídas da predição da forma do rosto são usadas para melhorar a própria predição, o que por sua vez aumenta a qualidade das *features*. A implementação

[1] http://dlib.net/

deste algoritmo da biblioteca `dlib` treinada com o conjunto de dados mencionado foi utilizada neste trabalho.

3 Trabalhos Relacionados

Várias abordagens têm sido exploradas nos últimos anos para o rastreamento ocular, com diferentes exigências de equipamentos. No trabalho de Hennessey et al. [5], o rastreamento é obtido a partir da interseção entre vetor do centro da córnea ao centro da pupila e o plano do monitor. Para isso, um modelo 3D dos olhos e do ambiente é utilizado, contando com uma única câmera auxiliada por luzes infravermelhas que causam um brilho na córnea para poder identificar pontos usados na estimação de seu centro.

Outra proposta é a utilização de aprendizado de máquina, mais notavelmente de redes neurais convolucionais (*convolutional neural networks* - CNN), muito eficazes para tarefas que envolvem processamento de imagens. Krafka et al. [7] treinaram uma CNN com recortes de olhos, do rosto e suas posições retiradas de imagens capturadas por *tablets* e *smartphones* em larga escala, buscando aumentar a capacidade de generalização de sua rede neural.

Técnicas de visão computacional também podem ser utilizadas, como foi feito no trabalho de Lamé [8], que foi usado como base para este projeto. Primeiramente, foram identificadas as partes da imagem que contêm os olhos, o que pode ser feito utilizando modelos pré-treinados disponíveis nas bibliotecas `OpenCV` ou `dlib`. É aplicada então a limiarização (*thresholding*) da imagem para detectar a região da íris, a partir da qual é calculado seu centroide, que corresponde aproximadamente ao centro da pupila. Com isso, é possível inferir a direção geral do olhar do usuário.

Nasor et al. [9] também propuseram uma maneira de controlar o mouse por meio dos olhos. Em seu trabalho, é considerado o movimento relativo da íris a partir de uma posição central para obter a direção do olhar, informação que é combinada com o piscar dos olhos para realizar os cliques.

4 Metodologia

Para a implementação deste trabalho, foi utilizado como base o código de Lamé [8], a partir do qual foram feitas algumas modificações e adição de funcionalidades. Será discutida a seguir em mais detalhes a implementação do protótipo.

4.1 Detecção da Face

Inicialmente, é feita a detecção do rosto para poder aplicar o algoritmo que faz a detecção dos olhos. Foi utilizada neste trabalho a implementação disponível na biblioteca `dlib` do algoritmo brevemente comentado na Seção 2. A única operação realizada na imagem antes de ser feita a detecção é a conversão para escala de cinza, necessária para o funcionamento da função.

4.2 Detecção dos Olhos

A detecção inicial dos olhos facilita significativamente a identificação das pupilas, reduzindo drasticamente a região de busca e o risco de detecções falsas. Para isso, primeiro foram obtidos os pontos fiduciais (*landmarks points*) dos olhos, cuja função também é disponível na biblioteca `dlib`, que traz uma implementação do método proposto por Kazemi e Sullivan [6].

Os olhos direito e esquerdo correspondem aos conjuntos de pontos (37-42) e (43-48), respectivamente. Uma vez identificadas a região de cada olho, ela é recortada da imagem aplicando uma máscara, obtida com a função `fillPoly`, que recebe os *landmarks* e cria um polígono. O resultado pode ser observado na Figura 1.

Figura 1: Exemplo de olho recortado da imagem.

4.3 Detecção da Pupila

A partir do recorte do olho, diversas operações de processamento de imagens são realizadas para extrair as informações necessárias para a identificação da pupila. Primeiramente, o filtro bilateral é aplicado para reduzir possíveis ruídos presentes na imagem. A seguir, uma abertura morfológica é aplicada, que consiste em uma erosão seguida de dilatação, o que evita problemas caso haja reflexos na região da íris e pupila, como pode ser observado na Figura 1.

Por fim, é aplicada a limiarização (*thresholding*) na imagem, transformando-a em uma imagem binária. A escolha do limiar (*threshold*) é um desafio, pois o resultado depende fortemente das condições de iluminação do ambiente. Um método simples para a escolha desse valor foi adotado, testando diversos valores diferentes em um dado intervalo e ficando com aquele que produz uma região da íris mais próxima a um valor esperado de 48% da região total do olho.

Depois de processadas, as imagens resultantes são binárias e devem conter somente a região da íris e pupila. A partir disso, é calculado o contorno, que serve de entrada para o cálculo dos momentos da imagem. Como discutido na Subseção 2.1.3, é possível obter o centroide a partir dos momentos, o que equivale aproximadamente ao centro da pupila.

Na Figura 2, de cima para baixo, temos diferentes direções do olhar: no centro, para a direita e para a esquerda, respectivamente. As imagens mais à esquerda são obtidas depois da operação de abertura, enquanto as do meio passaram pela limiarização (*thresholding*). Nas imagens da direita, o centro da pupila que foi estimado é destacado pelo pixel em branco.

Nas imagens da segunda linha, é possível observar uma das dificuldades encontradas no método usado, que é a iluminação não homogênea. A área mais escura no canto esquerdo do olho cria uma região extra que se mistura com a da íris ao fazer a limiarização, o que altera um pouco a posição do centro estimado.

Figura 2: Imagens do olho durante as etapas de processamento.

4.4 Determinação da Direção do Olhar

O protótipo tenta estimar a direção do olhar para cinco direções da tela: parte inferior e superior da tela, centro, direita e esquerda, além de combinações entre elas, como superior esquerda, inferior direita, etc., para um total de 9. Para melhorar a precisão do protótipo, uma calibração inicial é realizada, na qual é solicitado que o usuário clique em um quadrado em cada região mencionada, exceto o centro, enquanto olha para ele. Ao fazer isso, é capturada uma foto, a qual é analisada para obter a posição da pupila, que fica salva para cada caso e serve de base para comparar com a posição atual.

Para as direções horizontais, é considerada a posição da pupila relativa ao canto do olho. Isso torna o protótipo um pouco menos sensível a movimentos da cabeça, mas idealmente o usuário deve mover apenas o olho para olhar em cada região. No caso vertical, como os olhos são mais alongados do que altos, o movimento da pupila é menor para cima e para baixo do que para direita e para esquerda. Então, o sistema dificilmente consegue a precisão adequada para esses movimentos pequenos, especialmente para câmeras de baixa resolução. Por isso, foi usada a posição absoluta da pupila nesses casos, o que também permite que o usuário levante ou abaixe a cabeça levemente para melhor controlar o protótipo.

4.5 Detecção do Piscar de Olhos

A checagem do piscar de olhos foi baseada no trabalho de Soukupová e Čech [3], que utiliza os pontos fiduciais para calcular a razão entre a parte vertical e horizontal do olho, que diminui drasticamente quando o usuário pisca. No trabalho mencionado, é utilizado o classificador chamado máquinas de vetores de suporte (*support vector machines* - SVM) para a detecção, mas devido a limitações de tempo neste projeto foi utilizado um valor pré-determinado da razão mencionada para a identificação do piscar.

Uma modificação feita em relação ao código inicial de Lamé [8] é a consideração do estado de cada olho de maneira independente. Anteriormente, era calculada a média da razão de ambos os olhos, mas isso limitava as opções de detecção, pois exigia que ambos os olhos estivessem fechados. Assim, foi possível que algumas operações do controle do mouse fossem feitas usando o piscar de um único olho, permitindo que o usuário mantivesse o outro aberto para poder ver melhor o resultado de suas ações.

4.6 Controle do Mouse

Um sistema simples que combina a direção do olhar e o piscar do olho foi implementado no protótipo para realizar o controle do mouse. A direção dos últimos cinco quadros é salva em uma fila. Quando o usuário pisca, um ou ambos os olhos por três ou mais quadros seguidos e, enquanto ele mantém o piscar, o mouse é movido na direção que aparece com mais frequência nessa lista.

Essa lógica serve para evitar que o usuário realize ações não intencionais quando piscar naturalmente. Também evita o problema de que, como a pupila sempre está sendo rastreada, durante o piscar há um momento em que sua posição estimada pode ficar errada, o que poderia fazer com que o mouse se movimentasse na direção errada.

5 Resultados

Nesta seção, será apresentado o funcionamento do protótipo, que foi utilizado com uma câmera de resolução 720p e 3 MP que grava até 30 fps. Devido ao número de operações feitas nas imagens, não foi possível processá-las com a frequência máxima permitida pela câmera, entretanto, o desempenho foi suficiente para o controle do mouse de maneira satisfatória.

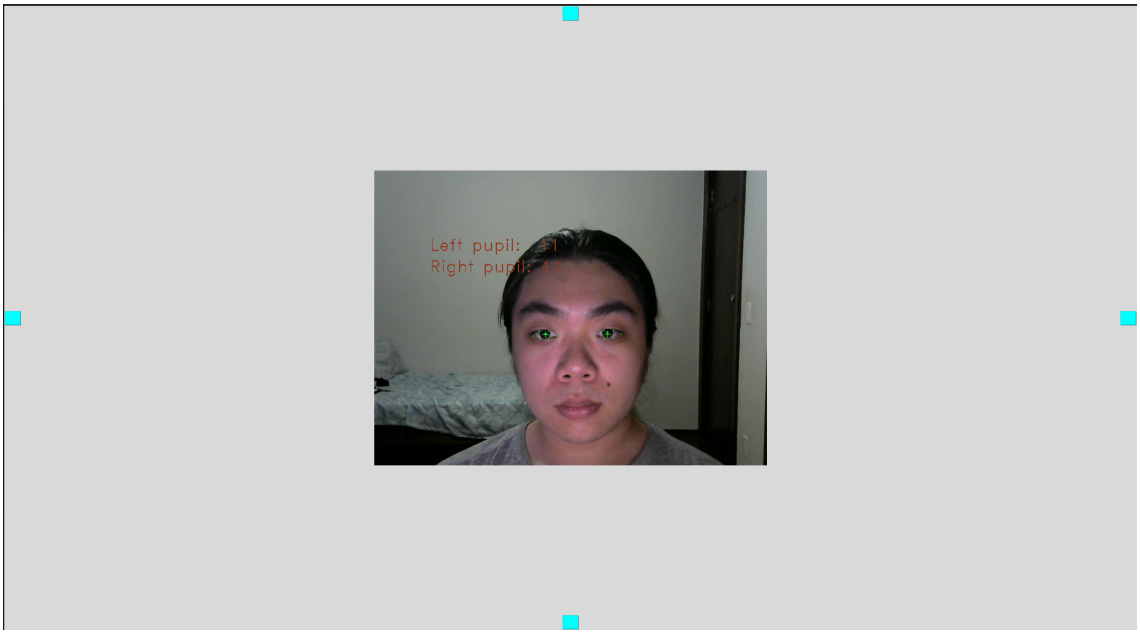

Figura 3: Tela de calibração.

A tela de calibração utilizada é mostrada na Figura 3. Os quadrados azuis indicam os pontos para onde o usuário deve olhar para que o protótipo obtenha as referências necessárias para determinar a direção do olhar. Por simplicidade, é necessário que esses quadrados sejam clicados, pois nesse momento uma foto é capturada para extrair as informações exigidas, mas idealmente esse processo seria automático, dando apenas ins-

truções ao usuário. O vídeo da câmera no centro foi adicionado para referência, permitindo verificar o funcionamento da detecção da pupila nas condições presentes.

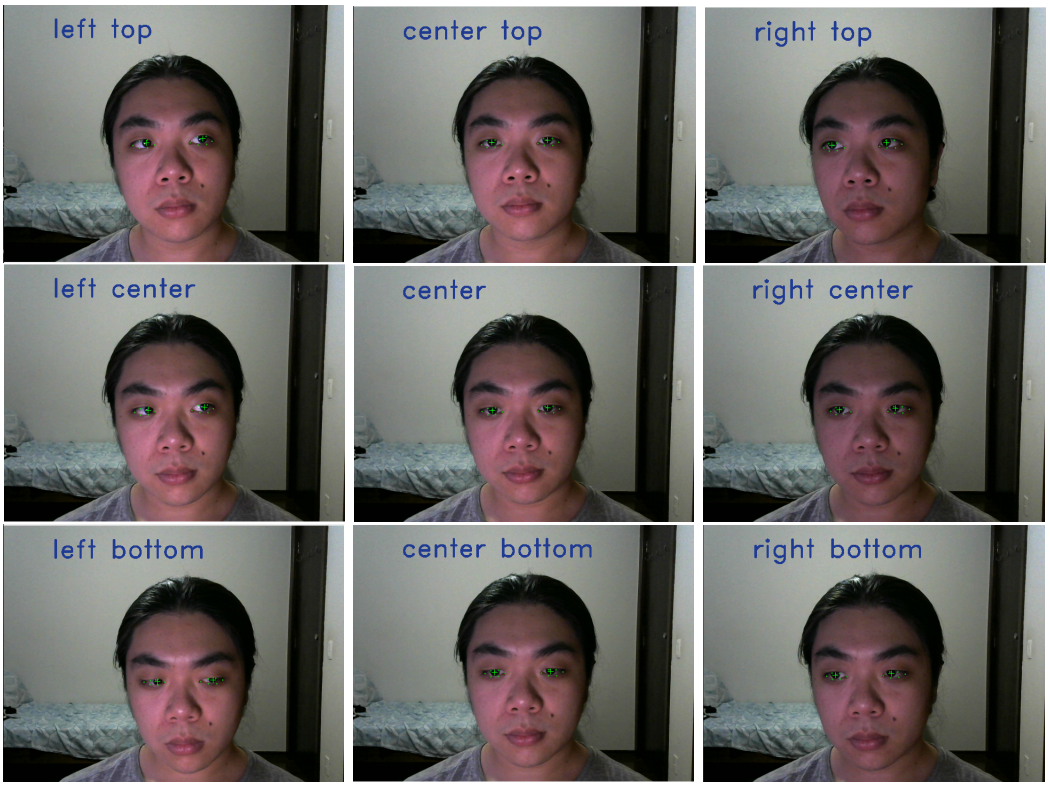

Figura 4: Detecção da direção do olhar pelo protótipo.

A Figura 4 mostra todas as direções que são detectadas pelo protótipo, indicadas pelo texto em azul. Também estão destacados em verde os *facial landmarks* detectados, além do centro estimado da pupila. Depois de fixar por um momento o olhar em uma direção e manter um ou os dois olhos fechados, o mouse se movimenta nessa direção, incluindo as diagonais, em uma taxa fixa. No caso do centro, o piscar ativa o clique do mouse ao invés de movimentá-lo.

Conforme mencionado na Subseção 4.4, um leve movimento da cabeça nas direções verticais ajuda em sua detecção, enquanto para as direções horizontais o movimento do olho é mais importante. No geral, o protótipo foi preciso na detecção da direção do olhar, desde que não fossem feitos movimentos muito grandes da cabeça, o que requer uma recalibração do sistema.

O piscar de olhos é detectado para um ou ambos os olhos fechados, como pode ser observado na Figura 5, o que permite que o usuário mantenha um dos olhos abertos para poder melhor controlar o mouse, sem a necessidade de adivinhar quando ele deve abrir os olhos para parar o mouse.

Conforme comentado na Seção 4.5, quando o olho está fechado, os pontos dos *facial landmarks* da parte de cima e de baixo do olho se aproximam, o que leva à detecção. No

Figura 5: Detecção do piscar pelo protótipo.

entanto, foi encontrada uma dificuldade na detecção do piscar, devido ao posicionamento incorreto dos *landmarks* dos olhos em algumas situações, especialmente quando o usuário possui olheiras, como pode ser visto no olho direito (à esquerda) na segunda imagem da Figura 5. Isso pode fazer o protótipo deixar de identificar o piscar, pois pode parecer que o olho não está fechado.

6 Conclusões e Trabalhos Futuros

Neste projeto, um protótipo foi desenvolvido que, a partir de uma calibração simples, consegue fornecer uma interface para interação entre o usuário e o computador usando apenas os olhos e sem exigir grandes investimentos em equipamentos caros e especializados. Vários conceitos e algoritmos foram usados em conjunto para poder fazer funcionar as diversas etapas de processamento necessárias para partir da imagem da câmera até chegar na detecção do centro da pupila e o controle do mouse.

Há ainda diversos aspectos do protótipo que podem ser refinados, resolvendo alguns dos problemas mencionados, como o funcionamento em condições de luminosidade adversas e uma calibração mais autônoma, que podem ser explorados futuramente.

Referências

[1] Talk Back, 2022. `https://en.wikipedia.org/wiki/Google_TalkBack`.

[2] Voice Over, 2022. `https://en.wikipedia.org/wiki/VoiceOver`.

[3] J. Cech and T. Soukupova. Real-time eye blink detection using facial landmarks. *Cent. Mach. Perception, Dep. Cybern. Fac. Electr. Eng. Czech Tech. Univ. Prague*, pages 1–8, 2016.

[4] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 886–893. IEEE, 2005.

[5] C. Hennessey, B. Noureddin, and P. Lawrence. A single camera eye-gaze tracking system with free head motion. In *Symposium on Eye Tracking Research & Applications*, pages 87–94, 2006.

[6] V. Kazemi and J. Sullivan. One millisecond face alignment with an ensemble of regression trees. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1867–1874, 2014.

[7] K. Krafka, A. Khosla, P. Kellnhofer, H. Kannan, S. Bhandarkar, W. Matusik, and A. Torralba. Eye tracking for everyone. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2176–2184, 2016.

[8] A. Lamé. Gaze tracking. Github repository, February 2019.

[9] M. Nasor, K. M. Rahman, M. M. Zubair, H. Ansari, and F. Mohamed. Eye-controlled mouse cursor for physically disabled individual. In *Advances in Science and Engineering Technology International Conferences*, pages 1–4. IEEE, 2018.

[10] C. Sagonas, E. Antonakos, G. Tzimiropoulos, S. Zafeiriou, and M. Pantic. 300 faces in-the-wild challenge: Database and Results. *Image and Vision Computing*, 47:3–18, 2016.

[11] C. Tomasi and R. Manduchi. Bilateral filtering for gray and color images. In *Sixth International Conference on Computer Vision (IEEE Cat. No. 98CH36271)*, pages 839–846. IEEE, 1998.

[12] B. Zhao, J. Feng, X. Wu, and S. Yan. A survey on deep learning-based fine-grained object classification and semantic segmentation. *International Journal of Automation and Computing*, 14(2):119–135, 2017.