

ISEL – INSTITUTO SUPERIOR DE ENGENHARIA DE LISBOA  
ADEETC – ÁREA DEPARTAMENTAL DE ENGENHARIA DE  
ELECTRÓNICA E TELECOMUNICAÇÕES E DE COMPUTADORES

---

MERCM  
MESTRADO EM ENG. DE REDES DE COMUNICAÇÃO E MULTIMÉDIA  
TRABALHO FINAL DE MESTRADO

---

## Sistema de Seguimento dos Movimentos Oculares para Interface Adaptada Pessoa-Máquina



Rafael Santos (37178)

*Júri*

---

<i>Presidente Professor Doutor</i>	Paulo Trigo
<i>Arguente Doutor</i>	André Lourenço
<i>Orientador Professor Doutor</i>	Arnaldo Abrantes
<i>Orientador Professor Doutor</i>	Pedro Jorge

---

*Dezembro, 2015*



# Resumo

Com a evolução da tecnologia as interfaces naturais são cada vez mais utilizadas e procuradas uma vez que a interacção com os dispositivos tendem a ser mais intuitivas, abandonando os tradicionais ratos e teclados.

O surgimento de novos dispositivos ou a evolução dos actuais, tem permitido um rápido desenvolvimento noutras formas naturais de interacção, que tendem a emergir, como interfaces que permitam interagir com o computador ou dispositivo electrónico através de gestos ou do olhar. Além disso, o aumento dos níveis de processamento têm permitido que as aplicações associadas às interfaces naturais realizem/detectem acções em tempo real cada vez mais sofisticadas.

No caso das interfaces que utilizam o olhar, o utilizador apenas terá que olhar para o ecrã e escolher a acção que pretende. Se for feita uma analogia para os sistemas actuais, rato e teclado, podemos considerar que o movimento dos olhos seria responsável por movimentar o rato e o piscar do olho por efectuar o clique. Entre muitas utilizações possíveis este tipo de interface permitiria pessoas com deficiência motora total ou parcial utilizar o computador e assim aumentar a sua qualidade de vida.

Assim, este trabalho tem por finalidade principal o estudo e implementação de um sistema que permita seguir o foco do olhar e assim desenvolver acções. Pretende-se neste projecto utilizar dispositivos de uso geral, nomeadamente, não intrusivos, o que apresenta dificuldades acrescidas na localização e seguimento preciso da pupila.

Compara-se neste trabalho uma câmara a cores normal de computador (webcam) e uma nova câmara, com informação de profundidade, que o fabricante (Intel) pretende que seja o novo padrão de câmaras para computador.

São apresentados testes comparativos entre os sistemas desenvolvidos e propõe-se um sistema robusto que, numa primeira fase, pretende simular a utilização do rato do computador.



# Abstract

With technology's evolution the natural interfaces are increasingly used and sought once the interaction with the devices tend to be more intuitive, abandoning traditional mouse and keyboard.

The emergence of new devices, or the existing one's evolution, has enabled rapid development in other natural forms of interaction, which tend to emerge, such as interfaces that allow interact with the computer or electronic device through gesture or eyes. Furthermore, increased levels of processing have enabled applications associated with natural interfaces to do or detect increasingly sophisticated actions in real time.

In the case of interfaces that use eye gaze, you just have to look at the screen and choose the action you want. If an analogy to existing systems is made, mouse and keyboard, we can consider that the eye movements would be responsible for moving the mouse and the blinking eye for making the click. Among many possible using this type of interface will allow people with total or partial physical disabilities to use the computer and thereby increase their quality of life.

This work has the principal purpose the study and implementation of a system that will ensure the eye gaze and doing actions. It's intended in this project using general purpose devices, namely, non-intrusive, which presents additional difficulties in locating and precise tracking of the pupil.

In this work is made a comparison within a color webcam and a new camera, with depth information, which the manufacturer (Intel) is to be the new standard for cameras to computer.

Comparative tests between the developed systems are shown. A robust system are presented and proposed that, initially, intended to simulate the use of the computer mouse.



# Agradecimentos

Este projeto, à semelhança tantos outros na minha vida, contam com o apoio importante de outras pessoas. Sem algumas delas o processo de desenvolvimento seria muito mais moroso e complicado.

Por esse motivo, quero prestar a minha gratidão aos meus pais, Luís Afonso Santos e Cristina Guerreiro Lopes Santos, pelo apoio incondicional e espírito de sacrifício em alturas menos boas, sem eles este projecto não seria possível de realizar.

Os meus agradecimentos à minha irmã, Joana Isabel Lopes Santos, pela amizade e compreensão. Não posso deixar de agradecer à minha esposa, Raquel Alexandra Cavaqueiro Lopes, pela força e motivação que me transmitiu.

Quero agradecer aos orientadores, Pedro M. Jorge e Arnaldo Abrantes, pelo conhecimento, motivação e apoio no desenvolvimento da dissertação.

Um obrigado ao Hélder Bastos e Paulo Martins pela ajuda e apoio.

Por último mas não menos importante, obrigado a todos os meus amigos que me ajudaram a evoluir como pessoa e que me apoiaram nos momentos bons e menos bons do meu percurso académico - Obrigado.

Para todos vós, um obrigado.



*Dedico este trabalho aos meus pais, irmã, esposa e familiares que me ajudaram a superar as minhas dificuldades e evoluir como pessoa e profissional.*



# Índice

Resumo	i
Abstract	iii
Agradecimentos	v
Índice	ix
Lista de Figuras	xi
<b>1 Introdução</b>	<b>1</b>
<b>2 Trabalho relacionado</b>	<b>5</b>
<b>3 Fundamentos Teóricos</b>	<b>11</b>
3.1 Haar Cascades . . . . .	11
3.2 Transformada de Hough . . . . .	15
3.2.1 Circunferência . . . . .	15
3.3 Filtro de Kalman . . . . .	17
3.3.1 Modelo, Processo e Iteração . . . . .	17
<b>4 Análise</b>	<b>21</b>
4.1 Requisitos Funcionais . . . . .	21
4.2 Requisitos Operacionais . . . . .	22
4.3 Casos de utilização . . . . .	22
4.4 Modelo Proposto . . . . .	24
4.4.1 Diagrama de estados . . . . .	25
4.5 Arquitectura . . . . .	27
4.5.1 Estrutura de ficheiros . . . . .	27

4.5.2	Diagramas de classes . . . . .	28
4.5.3	Diagrama classes - Teste . . . . .	31
4.5.4	Diagrama classes - bibliotecas externas . . . . .	31
<b>5</b>	<b>Implementação</b>	<b>33</b>
5.1	Desenvolvimento . . . . .	33
5.1.1	Câmara web - RGB . . . . .	35
5.1.2	Câmara web - Intel Real Sense . . . . .	36
5.1.3	Acções . . . . .	41
<b>6</b>	<b>Validação e testes</b>	<b>49</b>
6.1	Testes e validação de movimento do cursor com os olhos . . .	55
6.2	Testes e validação de movimento do cursor com o nariz . . . .	73
<b>7</b>	<b>Conclusões</b>	<b>83</b>
	<b>Bibliografia</b>	<b>87</b>

# Lista de Figuras

2.1	Exemplo de técnicas intrusivas(a) e não intrusivas(b) . . . . .	6
2.2	Exemplo do olho iluminado com luz infravermelha. . . . .	7
3.1	Funcionamento das <i>integral images</i> . . . . .	12
3.2	Algoritmo AdaBoost utilizado para construir o classificador de <i>Haar Cascade</i> - retirado de [19] . . . . .	13
3.3	Esquema de decisão das camadas no algoritmo Haar Cascades.	14
3.4	Exemplo das faces utilizadas para o treino . . . . .	14
3.5	Diagrama de blocos das várias etapas das transformadas de Hough. . . . .	15
3.6	Ilustração da abordagem ao círculos para utilização nas trans- formadas de Hough. . . . .	16
4.1	Casos de utilização gerais . . . . .	23
4.2	Casos de utilização após definição de prioridades . . . . .	24
4.3	Modelo proposto . . . . .	25
4.4	Modelo proposto . . . . .	26
4.5	Estrutura de ficheiros do sistema . . . . .	27
4.6	Diagrama de classes - <i>ControloApp</i> . . . . .	29
4.7	Diagrama de classes - <i>ControloAction</i> . . . . .	29
4.8	Diagrama de classes - <i>ControloFaceRSSDK</i> . . . . .	29
4.9	Diagrama de classes - <i>ControloUI</i> . . . . .	30
4.10	Diagrama de classes - Teste . . . . .	31
4.11	Diagrama de classes - Bibliotecas externas . . . . .	31
5.1	Face obtida a partir da câmara Intel Real Sense - Fluxo <i>RGB</i>	34
5.2	Imagem obtida a partir da câmara Intel Real Sense - Fluxo profundidade . . . . .	35

5.3	Relação entre face e a câmara - Pose da face . . . . .	37
5.4	Ângulos de Euler aplicados à aeronáutica . . . . .	37
5.5	Ponto de referência do nariz . . . . .	38
5.6	Pontos de referência do Olho - Pupila . . . . .	38
5.7	Comparação entre a detecção do olhos sem e com filtro de Kalman - Olho esquerdo . . . . .	39
5.8	Comparação entre a detecção do olhos sem e com filtro de Kalman - Olho direito . . . . .	39
5.9	Ponto referência da pupila com Kalman (vermelho) e sem Kalman (verde) . . . . .	40
5.10	Comparação entre a detecção com três fluxos de informação e o método apenas com o fluxo RGB - Olhar frente . . . . .	41
5.11	Comparação entre a detecção com três fluxos de informação e o método apenas com o fluxo RGB - Olhar cima . . . . .	41
5.12	Ilustração das acções implementadas com os olhos . . . . .	42
5.13	Ilustração das acções implementadas com o nariz e boca . . . . .	42
5.14	Movimento ocular a simular um rato, com movimentos diferenciais . . . . .	43
5.15	Ilustração do ecrã utilizado para calibrar . . . . .	44
5.16	Ilustração da associação das coordenadas do olho com posições do ecrã . . . . .	45
5.17	Pontos de referência do olho esquerdo fechado e aberto . . . . .	47
5.18	Ilustração da associação das coordenadas do nariz com posições do ecrã . . . . .	47
5.19	Pontos de referência da boca . . . . .	48
6.1	Diagrama de blocos do sistema . . . . .	49
6.2	Imagem do fluxo RGB com objecto compatível com a biblioteca OpenCV . . . . .	50
6.3	Imagem do fluxo de profundidade com objecto compatível com OpenCV . . . . .	50
6.4	Imagem do fluxo de infravermelho com objecto compatível com OpenCV . . . . .	51
6.5	Validação da detecção dos olhos e seus pontos de referência . . . . .	52
6.6	Oscilações da pupila - Kalman vs detecção . . . . .	52
6.7	Movimento do rato - olhar para esquerda do ecrã . . . . .	53

6.8	Movimento do rato - olhar para topo do ecrã . . . . .	54
6.9	Ilustração das posições do ecrã utilizadas para os testes . . . . .	54
6.10	Gráfico de dispersão - Ecrã 15" <i>FullHD</i> - Olho esquerdo e direito - Olhar para o centro . . . . .	55
6.11	Gráfico de dispersão - Ecrã 15" <i>FullHD</i> - Olho esquerdo e direito - Olhar para o canto superior e inferior direito. . . . .	56
6.12	Gráfico de dispersão - Ecrã 15" <i>FullHD</i> - Olho esquerdo e direito - Olhar para o canto superior e inferior esquerdo. . . . .	57
6.13	Gráfico de dispersão - Ecrã 15" <i>fullhd</i> - Rato - Quando o utilizador olha para os vários pontos de teste no ecrã. . . . .	58
6.14	Gráfico de dispersão - Ecrã 24" <i>FullHD</i> - Olho esquerdo e direito - Olhar para o centro . . . . .	59
6.15	Gráfico de dispersão - Ecrã 24" <i>FullHD</i> - Olho esquerdo e direito - Olhar para o canto superior e inferior direito. . . . .	60
6.16	Gráfico de dispersão - Ecrã 24" <i>FullHD</i> - Olho esquerdo e direito - Olhar para o canto superior e inferior esquerdo. . . . .	61
6.17	Gráfico de dispersão - Ecrã 24" <i>FullHD</i> - Rato - Quando o utilizador olha para os vários pontos de teste no ecrã. . . . .	62
6.18	Gráfico de dispersão - Ecrã 15" <i>HD</i> - Olho esquerdo e direito - Olhar para o centro . . . . .	63
6.19	Gráfico de dispersão - Ecrã 15" <i>HD</i> - Olho esquerdo e direito - Olhar para o canto superior e inferior direito. . . . .	64
6.20	Gráfico de dispersão - Ecrã 15" <i>HD</i> - Olho esquerdo e direito - Olhar para o canto superior e inferior esquerdo. . . . .	65
6.21	Gráfico de dispersão - Ecrã 15" <i>HD</i> - Rato - Quando o utilizador olha para os vários pontos de teste no ecrã. . . . .	66
6.22	Gráfico de dispersão - Ecrã 24" <i>HD</i> - Olho esquerdo e direito - Olhar para o centro . . . . .	67
6.23	Gráfico de dispersão - Ecrã 24" <i>HD</i> - Olho esquerdo e direito - Olhar para o canto superior e inferior direito. . . . .	68
6.24	Gráfico de dispersão - Ecrã 24" <i>HD</i> - Olho esquerdo e direito - Olhar para o canto superior e inferior esquerdo. . . . .	69
6.25	Gráfico de dispersão - Ecrã 24" <i>HD</i> - Rato - Quando o utilizador olha para os vários pontos de teste no ecrã. . . . .	70
6.26	Tabela do erro do posicionamento do rato através dos olhos . . . . .	71

6.27	Piscar do olho esquerdo - acção de clique do botão esquerdo do rato . . . . .	72
6.28	Piscar do olho direito- acção de clique do botão direito do rato	72
6.29	Oscilações da detecção do nariz - com e sem filtro de Kalman .	73
6.30	Gráficos de localização - Ecrã 15" <i>fullhd</i> - Nariz . . . . .	75
6.31	Gráficos de localização - Ecrã 24" <i>fullhd</i> - Nariz . . . . .	76
6.32	Gráfico de Posicionamento do cursor - Ecrã 15" <i>fullhd</i> - Nariz .	77
6.33	Gráfico de Posicionamento do cursor - Ecrã 24" <i>fullhd</i> - Nariz .	78
6.34	Tabela erro do posicionamento do rato através do nariz . . . .	79
6.35	Movimento do rato com o nariz - botão de minimizar janela .	79
6.36	Movimento do rato com o nariz - botão de pausar o vídeo . . .	80
6.37	Movimento do rato com o nariz, impossibilita a utilização do piscar de olhos para acções do sistema . . . . .	80
6.38	Clique esquerdo do rato com abertura e fecho da boca - simular um arrastar de ficheiro . . . . .	81
7.1	Reflexo da pupila sobre luz infravermelha . . . . .	84

# Capítulo 1

## Introdução

Com a envolvimento da tecnologia no quotidiano, tarefas rotineiras tornaram-se mais fáceis de executar, contudo, as interfaces de comunicação entre as pessoas e a máquina pouco mudaram nas últimas décadas.

Na actualidade existem inúmeras interfaces a serem testadas e à espera de maturidade para entrar no mercado, que utilizam sensor de profundidade já com alguma estabilidade como, por exemplo, a Kinetic da Microsoft ou o Leap motion, ou até mesmo, sistemas com mais complexidade que integram vários tipos de sensores como os Holo Lens da Microsoft e o Smart Helmet da Daqri. Estas tecnologias permitem imergir o utilizador no ambiente de realidade aumentada. Estas interfaces ajudam os seus utilizadores a efetuarem tarefas normais do quotidiano, que de outra forma não poderiam. Tornam-se assim muito úteis e importantes mas, têm um custo elevado para quem necessita de usufruir delas.

O trabalho apresentado nesta dissertação insere-se na área das interfaces pessoa-máquina adaptadas ao ser humano com dificuldades motoras e tem como finalidade principal estudar um sistema de seguimento ocular que permita ao utilizador controlar o computador através dos olhos.

O acto de olhar para o ecrã está presente no quotidiano de todos nós e, direta ou indiretamente, faz parte da maior parte dos processos de interação com as máquinas. Típicamente, é através da assimilação e processamento da informação disponibilizada no ecrã que são tomadas as decisões do que realizar para concluir o processo. Contudo, a informação que o olhar pode fornecer ainda não é muito utilizada, em virtude das dificuldades em se con-

seguir seguir o olhar com precisão necessário. Os sistemas que utilizam o olhar para executar acções têm valores elevadíssimos e uma utilização pouco natural/funcional. Esta temática tem vindo a ser muito estudada nos últimos anos uma vez que surgem equipamentos que permitem realizar seguimento dos olhos com maior detecção. Mas devido à elevada complexidade, requisitos em termos de equipamento e custos não são muito divulgados.

Procurou-se assim perceber de que modo a tecnologia recente e cada vez mais acessível ao utilizador comum pode ser utilizada para realizar esta forma de interacção pessoa-máquina.

Actualmente a maior parte dos dispositivos vêm equipados com câmara RGB<sup>1</sup> que pode ser usada para recolher informação e dar feedback ao utilizador. Contudo, a tecnologia utilizada nessas câmaras não é a mais indicada para se conseguir recolher informação com precisão e amostragem suficientes. Para seguir o olhar de um modo robusto é necessário possuir uma velocidade de processamento elevada para que seja possível efectuar o processamento de imagem em tempo real.

Por este motivo, no desenvolvimento deste projecto, foi utilizada uma nova câmara com a tecnologia 3D, a Intel Real-Sense F200 (RSF200). Esta câmara foi desenvolvida pela Intel, tendo sido anunciada no último trimestre de 2014 e disponibilizada para a comunidade científica em Janeiro de 2015. A RSF200 inclui três fontes de informação distintas, infravermelhos, *RGB*, de profundidade e possui, ainda, um microprocessador próprio da Intel. Para fazer uso desse dispositivo a Intel disponibiliza um SDK de código fechado.

A utilização de tecnologias muito recentes poderá ter a desvantagem de alguma instabilidade relacionada com a utilização de controladores e do *Software Development Kit* (SDK). No caso da RSF200, à data de escrita deste documento, o SDK encontra-se na *release* 4 versão 6 e os controladores que permitem o acesso à câmara na versão 1.4.

Para efetuar o estudo do sistema de seguimento de movimentos oculares foi utilizada, também, a *Open Source Computer Vision Library*, inicialmente desenvolvida pela Intel, a OpenCV. Em 2000 a Intel lançou um desafio de

---

<sup>1</sup>red, green e blue (vermelho, verde e azul)

criar uma biblioteca multi-plataforma. Esta biblioteca já conta com a versão 3.0 e é desenvolvida segundo a licença *Berkeley Software Distribution* (BSD), licença de código aberto que permite o uso para fins académicos e comerciais gratuitamente.

A OpenCV disponibiliza várias ferramentas de processamento de imagem, das quais se destacam:

- O classificador Haar Cascade de Viola & Jones [19] para detecção de objectos ou formas;
- As transformadas de Hough para detectar formas paramétricas;
- Filtro de Kalman para reduzir ruído nas observações com vista ao seguimento.

A biblioteca OpenCV e o SDK da RSF200 são disponibilizados para várias linguagens, incluindo Java, na sua versão 7 ou superior. Java foi a linguagem escolhida para este estudo devido às suas características que lhe transmitem robustez, facilidade de adaptação e bom desenho estrutural.

O desenvolvimento deste trabalho contribui para o estudo de novas interfaces de interacção com dispositivos electrónicos. No decorrer do desenvolvimento do sistema foram testados e revelados resultados promissores e conclusivos de que, a viabilidade de desenvolver este tipo de interfaces, cada vez se torna mais real e acessível a todos.

Esta dissertação explora as potencialidades de equipamento de última geração que em breve se tornará padrão em grande parte dos dispositivos móveis. O sistema permite assim aferir que a evolução tecnológica permite, cada vez mais, interagir com o mundo digital de forma natural e intuitiva.

O desenvolvimento do sistema encontra-se dividida em seis capítulos que são:

1. Trabalho relacionado;
2. Fundamentos Teóricos;
3. Análise;

4. Implementação;
5. Validação e Testes;
6. Conclusões.

# Capítulo 2

## Trabalho relacionado

Segundo Jaimes e Sebe [4] a evolução dos equipamentos têm vindo a desafiar a forma como as interfaces Pessoa-Máquina vão evoluir para além do tradicional rato e teclado.

O olhar é uma das formas mais natural para uma possível interacção com a máquina. O foco do olhar é uma boa fonte de informação sobre aquilo a que o utilizador está a prestar atenção. Como Poole e Ball referem em [6] é possível desenvolver interfaces práticas que utilizam o seguimento do olhar e o foco. O seguimento do olhar contempla o seguir o movimento dos olhos para onde a pessoa olha, mesmo sem dar nenhuma relevância. O foco, ao contrário do olhar, contempla uma situação à qual o utilizador dá relevância e fica focado na mesma. Estas abordagens têm vindo a ser estudadas à mais de cem anos.

Existem várias técnicas de detecção do foco e seguimento do olhar e podem ser divididas em duas categorias:

1. Intrusivas - que necessitam de equipamento em contacto com o utilizador para obter a informação necessária de modo a que o sistema funcione, ver figura 2.1;
2. Não-Intrusivas ou Técnicas Remotas - são maioritariamente baseadas em processamento de imagem, tipicamente utilizam câmaras para obter informação do olhar. Este tipo de técnicas não utiliza qualquer equipamento em contacto com o utilizador, ver figura 2.1.



Figura 2.1: Exemplo de técnicas intrusivas(a) e não intrusivas(b)

O trabalho de Rayner e Pollatsek [7] foi dos primeiros nesta área e utilizava electro-oculografia. Esta técnica, intrusiva, utiliza eléctrodos montados na pele em redor do olho a fim de medir as diferenças de potencial provocadas pelo movimentos dos olhos. Rayner e Pollatsek desenvolveram este trabalho com o intuito de estudar a psicologia da leitura, tentando perceber que relação tem o movimento ocular com a leitura que o utilizador faz.

Em 2003, Duchowski [2] propôs um método, semelhante ao [7], baseado na utilização de lentes de contacto, cobrindo toda a área da córnea e esclera, com um aro metálico. Com alguma semelhança ao método de Rayner e Pollatsek [7], o método apresentado por Duchowski utilizava o aro metálico da lente de contacto para medir as variações do campo electromagnético provocado pelo movimento do olho. Ainda em [2], Duchowski refere dois métodos de processamento para o seguimento do olhar, baseado no ecrã ou baseado no modelo. O primeiro, com base no ecrã, depende do processamento de imagem e utiliza o ecrã como referência para possibilitar a detecção do seguimento ocular. O segundo método depende dos modelos utilizados para efectuar a triangulação e obtenção do movimento ocular.

Morimoto e Mimica em 2004 [13] efetuaram um estudo sobre várias técnicas de seguimento ocular com intuito de desenvolver aplicações interactivas. Neste trabalho é detalhada a estrutura do olho humano e estudado o reflexo corneal.

As técnicas não intrusivas são mais confortáveis para o utilizador, mais fáceis e rápidas de operar, o que leva ao utilizador poder desfrutar do sistema durante mais tempo. Contudo, este tipo de técnicas tem algumas condicionantes como a iluminação, a qualidade da câmara ou o tipo de sensor

utilizado pela câmara. Schnipke e Todd [21] explicam no seu trabalho como é difícil e complexa a obtenção de informação fidedigna através de sistemas comerciais, que não são específicos para seguimento ocular. A experiência levada a cabo em [21] mostrou que, durante um ano de experiência com o sistema, o utilizador não era capaz de seguir dezasseis objectos.

Com o intuito de melhorar os sistemas com base em técnicas remotas é utilizada um tipo de iluminação específica, a infra-vermelha [18][8]. Este tipo de iluminação mostrou poder ter potencial visto que a pupila tem um ponto de reflexão próprio quando submetida a este tipo de onda electromagnética. Drewes em 2010 [18] apresentou um estudo sobre o comportamento da pupila quando iluminada por luz infra-vermelha. O fenómeno que ocorre é o chamado de efeito de "olhos vermelhos" presentes nas fotografias. Quando iluminada com luz infra-vermelha a pupila tem um reflexo branco na imagem obtida pela câmara, ver figura 2.2.



Figura 2.2: Exemplo do olho iluminado com luz infravermelha.

Neste tipo de imagem (figura 2.2), a diferença entre o reflexo corneal, também conhecido como imagens de Purkinje, e a pupila é utilizada para determinar a direcção do olhar [2]. Segundo o estudo efectuado por Wu Tunhua [8] a maioria dos sistemas baseados em técnicas remotas, utilizam a iluminação infra-vermelha para criar as imagens de Purkinje e o contraste da imagem para encontrar a pupila. Segundo o estudo apresentado, os sistemas com iluminação infra-vermelha possuem menor erro que os sistemas que apenas utilizam a imagem no espectro do visível. Em 2014 Xiong et al [22] efectuaram uma comparação entre a utilização de câmara RGB e RGB com luz infra-vermelha, respectivamente Webcam RGB e a Microsoft Kinect. Concluíram que a informação recebida directamente da Kinect oferecia mais precisão na informação para efectuar seguimento ocular do que a informação retirada e tratada de uma Webcam RGB, isto deve-se à possibilidade de obter

a imagem infra-vermelho proveniente da câmara da Microsoft.

Com o aumento da utilização de radiação na gama dos infra-vermelhos surgem questões sobre eventuais problemas para a saúde ocular quando exposta durante longos períodos de tempo a esta radiação. James e Sebe [4] mostraram a sua preocupação e revelaram ainda não haver resultados conclusivos sobre esta questão.

Os sistemas não intrusivos requerem calibração por parte do utilizador. A calibração na maioria dos sistemas é feita com base no tradicional sistema de calibração dos antigos ecrãs de toque, são desenhados pontos de referência no ecrã para os quais se diz ao utilizador para olhar, são colectadas amostras da posição da pupila e do reflexo corneal. Após se obter a informação necessária, é estimada a relação entre a pupila, reflexo corneal e as coordenadas do ecrã.

Com os vários estudos efectuados ao olho humano percebeu-se que não se pode apenas utilizar o olho humano como indicador da posição do cursor devido aos movimentos oculares involuntários e espasmos, que podem tornar um sistema com base nessa informação muito sensível e pouco preciso [3]. Além disso, os espasmos podem levar o utilizador a efectuar acções indesejadas, o que pode tornar a experiência de interacção pessoa-máquina frustrante.

Um possível complemento ao seguimento ocular, como meio de controlo do sistema, é a utilização de comandos por voz [5]. O estudo apresentado por Kaur em 2003 mostra que esta opção ajudaria a retirar a pressão do utilizador ao tentar controlar os seus movimentos oculares. Permitiria, também, uma interacção mais natural com o sistema visto utilizar uma ferramenta comum, a voz. Outro complemento que poderia ser utilizado seria a pose da face como forma de complementar o movimento dos olhos. Em 2014, Patrick et al [11] efectuaram um estudo sobre a estimação da pose da face com uma webcam. No trabalho apresentado, conseguiram um processamento em tempo real e a operar com condições de iluminação constantes, contudo, se houver alterações drásticas de movimento, o sistema não mantém um bom desempenho devido às alterações de iluminação e o procedimento tem que ser reinicializado.

Fanelli et al [14] efectuaram um trabalho onde estimam a pose da face

com base em informação de um câmara com sensor de profundidade. O sistema apresentado funciona em tempo real e consegue estimar a pose com informação de câmara de baixa qualidade, para isto são utilizados modelos de regressão aleatória que classificam a imagem de profundidade em relação à cabeça e ao resto do corpo. Segundo este autor, "A estimação da pose da face é um elemento chave da análise do comportamento humano"[14].

O trabalho apresentado por Wild em 2012 [10], tem um objectivo similar ao apresentado neste projecto. Segundo o estudo, os sistemas que se baseiam na utilização de tecnologias não intrusivas, acessíveis a qualquer pessoa, requerem ainda algum desenvolvimento. Os sistemas com utilidade diária são maioritariamente intrusivos ou utilizam câmaras desenvolvidas para esse fim, com valor por unidade demasiado elevado. O sistema proposto por Wild visa comprovar a viabilidade do desenvolvimento de um sistema de seguimento ocular com a utilização exclusiva de uma Webcam comum. Para o desenvolvimento do sistema foi utilizada a biblioteca OpenCV.

Neste trabalho, foi utilizada a transformada de Hough para a detecção da íris. Contudo, esta técnica revelou falhas de detecção em virtude das pálpebras e pestanas deformarem a forma circular da íris. Durante o desenvolvimento deste sistema verificou que é complicado para o utilizador mexer somente os olhos, inclinando, mesmo que ligeiramente, a cabeça para acompanhar o movimento dos olhos.

David Wild utiliza um processo de calibração para conseguir projectar o centro da íris em coordenadas do ecrã através de comparação dos valores das coordenadas. Contudo, este sistema apresenta um erro demasiado grande para ser utilizado em aplicações uteis e não consegue um processamento em tempo real. Mesmo assim, abre a possibilidade de utilização de um sistema não intrusivo muito difundido como é o caso de uma webcam.

No final de 2013, Santos et al [23] desenvolveram um sistema semelhante ao de David Wild. No sistema proposto foi utilizada a biblioteca a OpenCV e as ferramentas mais relevantes utilizadas foram:

- Haar *Cascades* - utilizada detectar face, nariz e olhos;
- Transformadas de Hough - utilizada para detectar a íris;

- *Mean Shift (Cam Shift)* - utilizada para fazer seguimentos dos pontos;
- Filtro de Kalman - utilizado para suavizar as alterações de posição dos pontos.

Com semelhança aos problemas de Wild, este trabalho também se deparou com alguns problemas derivados da qualidade de imagem e iluminação a que o utilizador se encontrava exposto. Para estimar as coordenadas do ecrã para onde o utilizador está a olhar, foi feito com um método de calibração onde a pessoa deve olhar para os quatro extremos do ecrã. Quando testado com um vídeo pré-gravado <sup>1</sup> do utilizador, a simular uma utilização do sistema, o erro médio é de sensivelmente trinta pixeis, dez vezes menor que o trabalho de Wild. Quando testado com webcam o sistema apresentou erros entre cento cinquenta e duzentos e cinquenta pixeis, pouco melhor que o sistema de Wild. O sistema permitiu um processamento em tempo real.

Em finais de 2014 a Intel apresentou uma câmara web com informação de profundidade e que pretende que substitua as câmaras web nos novos dispositivos, nomeadamente, nos computadores portáteis, tablets e ecrãs.

Pretende-se, neste trabalho, actualizar o sistema proposto em [23] para a utilização deste dispositivo, em virtude de poder vir a ser um elemento utilizado em grande escala.

---

<sup>1</sup>Vídeo previamente gravado com a webcam de um portátil

# Capítulo 3

## Fundamentos Teóricos

O desenvolvimento deste projeto assenta em processamento de imagem, trigonometria e álgebra. As imagens são capturadas pela RSF200 e processadas pela aplicação desenvolvida em Java. A trigonometria e álgebra são utilizadas para testar e simular a projecção do movimento ocular no ecrã com o intuito de efectuar acções. Para o processamento de imagem são utilizadas funcionalidades disponibilizadas pelo SDK da Intel e a biblioteca de processamento de imagem - OpenCV.

Em seguida apresentam-se e descrevem-se as ferramentas de processamento de imagem mais utilizadas no desenvolvimento deste projecto.

### 3.1 Haar Cascades

A detecção de objectos por classificadores em cascata que utilizam características *Haar* são métodos eficazes de detecção de objectos, nomeadamente, de faces.

Este método foi proposto por Viola e Jones [19] e utiliza características *Haar*, que se baseiam na variação do contraste dentro de um determinado grupo de pixels da imagem, salientando assim, regiões de tons mais escuros e mais claros.

O método pode ser descrito de forma sucinta pela equação 3.1. As características Haar podem ser obtidas de forma eficiente através das *integral images* e, o classificador de Haar para obter melhor performance utiliza o método AdaBoost, que cria uma árvore de decisão com base nas características Haar em cada nó.

$$HaarCascade = CaracteristicasHaar + ClassificadorHaar \quad (3.1)$$

A implementação de uma nova representação de imagem a *integral image* concedeu a este método um grande sucesso. Motivado pelo trabalho de Pappageorgiou et al. [28], para que as características *Haar* possam ser avaliadas rapidamente, foi introduzido a nova representação de imagem, *integral image*, que permite o cálculo eficiente das características *Haar*, esta variável é determinada com base na equação 3.2.

$$II(x, y) = \sum_{x' < x, y' < y} I(x', y') \quad (3.2)$$

As *integral images* funcionam por segmentos, o que permite encontrar o valor para cada segmento de forma rápida e eficiente. Na figura 3.1 podemos verificar o funcionamento desta abordagem. Para encontrar o valor representativo da região escura ( $I(d, b)$ ) é, apenas necessário, efectuar os seguintes cálculos:  $I(c, b) - I(d, a) + I(c, a)$ ; assim, a utilização deste tipo de imagem permite a optimização da obtenção de características *Haar*.

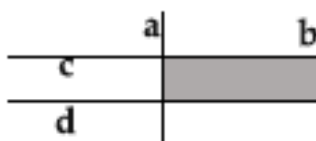


Figura 3.1: Funcionamento das *integral images*

Para a classificação é utilizada a técnica de AdaBoost, proposto em [29]. Numa imagem existem inúmeras características *Haar-like* e com a finalidade de conseguir um algoritmo rápido na detecção é necessário excluir a maior parte das características irrelevantes e focar o processamento nas que importam. Neste sentido o *AdaBoost* é uma ferramenta importante que disponibiliza um algoritmo de aprendizagem eficaz, robusto e com bom desempenho.

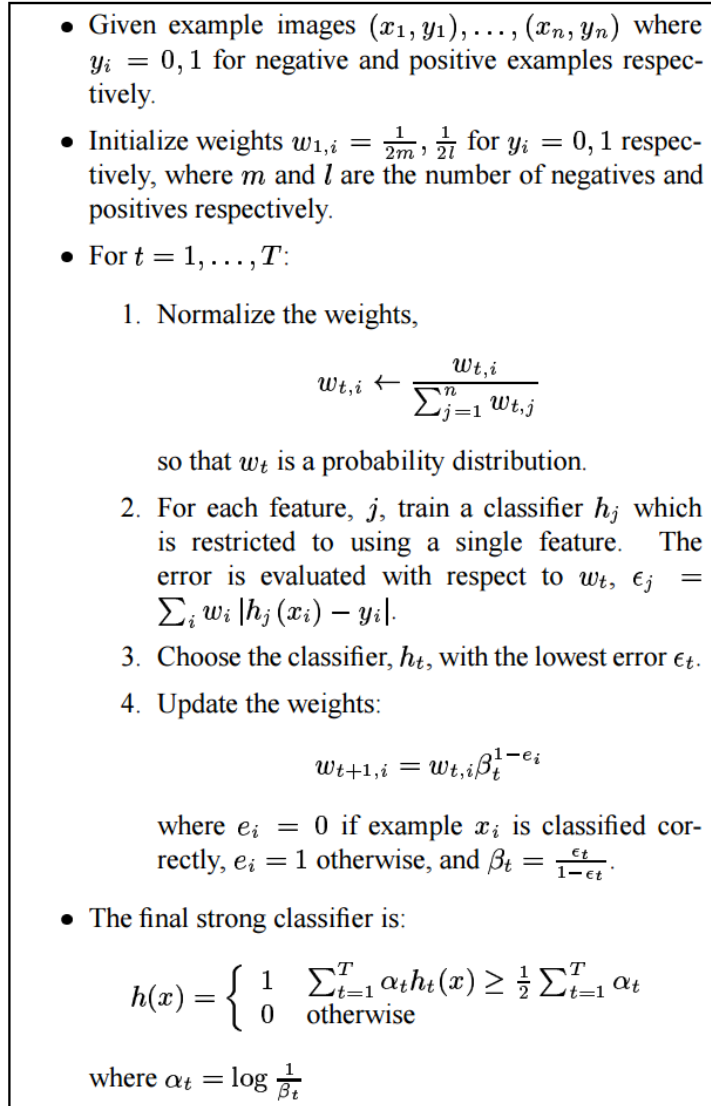


Figura 3.2: Algoritmo AdaBoost utilizado para construir o classificador de *Haar Cascade* - retirado de [19]

Na figura 3.2 é descrito o algoritmo *AdaBoost*, em cada ciclo o algoritmo escolhe uma característica.

Por fim, o algoritmo proposto por Viola & Jones utiliza uma sequência de classificadores (cascata) com vista a diminuir o tempo de processamento.

Na cascata é utilizada uma árvore de decisão, onde a análise e o resultado positivo da primeira característica despoleta a verificação no nível seguinte, o

nível seguinte, uma característica, já foi otimizado para obter altos níveis de desempenho [19]; por outro lado quando um nível retorna negativo a respectiva janela é descartada. A janela de análise consiste num grupo de pixels, com determinada dimensão onde o algoritmo de classificação vai executar uma série de avaliações e retornar o resultado.

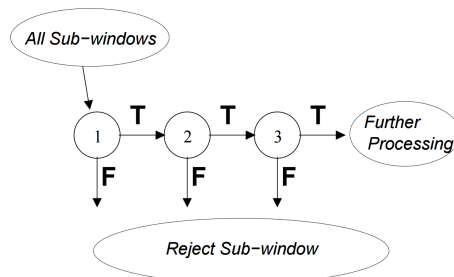


Figura 3.3: Esquema de decisão das camadas no algoritmo Haar Cascades.

Na figura 3.3 está esquematizada a sequência de acções realizadas pelo algoritmo proposto por Viola & Jones. Cada círculo representa uma característica Haar e respectiva numeração, nas transições existem setas que identificam o caminho do processamento face à detecção. Caso o resultado seja positivo o processamento avança para o nó seguinte, caso contrário a janela é descartada.



Figura 3.4: Exemplo das faces utilizadas para o treino

## 3.2 Transformada de Hough

A transformada de Hough foi desenvolvida por Paul Hough em 1962 e patenteada pela IBM [24]. Esta técnica foi desenvolvida com o intuito de detectar formas paramétricas, representadas em imagens binárias, como linhas, círculos e elipses. Com o desenvolvimento na área de processamento de imagem e visão artificial esta técnica tornou-se numa ferramenta clássica.

O conceito principal da transformada de Hough é a definição de um mapeamento entre o espaço da imagem e o espaço dos parâmetros (acumulador). Cada ponto de contorno da imagem é mapeado de modo a que seja possível determinar a sua correspondência no espaço de parâmetros. O valor do ponto de contorno é incrementado e o máximo local presente no acumulador identifica os parâmetros que corresponde à forma especificada.

Na figura 3.5 encontra-se resumido a sequência da informação quando se utiliza a transformada de Hough para a detecção dos objectos geométricos.

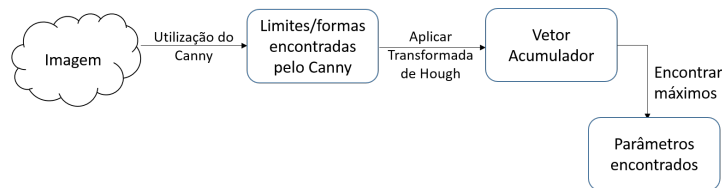


Figura 3.5: Diagrama de blocos das várias etapas das transformadas de Hough.

### 3.2.1 Circunferência

A transformada de Hough pode ser utilizada para detectar formas que possam ser descritas por um conjunto finito de parâmetros, por exemplo, uma circunferência, que no caso prático deste projecto será utilizado para representar a íris.

A circunferência é definida pelos parâmetros  $(x_0, y_0, r)$  onde  $(x_0, y_0)$  representam a localização do centro no eixo de coordenadas cartesianas e  $r$  representa o raio. Esta abordagem foi sugerida por E. R. Davies [24] e a figura 3.6 ilustra a abordagem considerada.

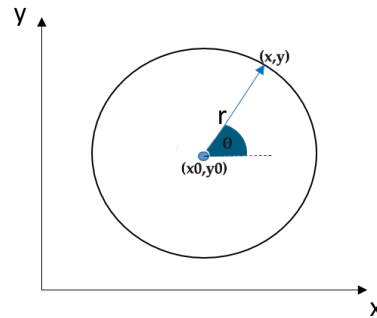


Figura 3.6: Ilustração da abordagem ao círculos para utilização nas transformadas de Hough.

Assim as equações que definem o centro do círculo são dadas por (ver equações 3.3 e 3.4):

$$(x - x_0)^2 + (y - y_0)^2 = r \quad (3.3)$$

$$\begin{cases} x_0 = x - r \cos \theta \\ y_0 = y - r \sin \theta \end{cases} \quad (3.4)$$

De modo a evitar a elevada complexidade computacional da execução de transformadas de Hough em três dimensões o problema é dividido em dois passos:

1. Encontrar pontos das circunferências na imagem - os contornos das circunferências podem ser encontradas através do algoritmo de Canny;
2. Encontrar o raio de cada círculo - Após obtenção das circunferências e seus contornos é encontrado o raio.

Para resolução do primeiro ponto são encontradas as rectas tangentes e respectiva normal dos pixels que representam o centro do círculo. Com os valores das várias normais, que cruzam o centro do círculo, é criado um histograma de  $(x, y)$  cujo os valores mais altos representarão o centro do círculo.

Encontrado o centro é necessário determinar o raio,  $r$ . Este é encontrado através de uma série de subtracções entre todos os pixels de contorno da imagem e o centro. Os valores são registados num histograma que, após estar concluído, nos indica o valor do raio.

Antes dos valores do centro e raio serem assumidos como verdadeiros é aconselhável que seja feito algum pós-processamento para validar a informação.

### 3.3 Filtro de Kalman

O filtro de Kalman é um algoritmo desenvolvido por Rudolf Kalman em 1960 [25] com o objectivo de poder estimar o estado do sistema dinâmico linear a partir de medições corrompidas por ruído branco. Por ruído branco entende-se um sinal aleatório com uma densidade espectral de potência uniforme que se caracteriza por, em instantes de tempo diferentes, o valor do sinal não guarda relação estatística.

No âmbito desta dissertação o filtro de Kalman é utilizado para estimar a trajectória da pupila na imagem, dadas as posições medidas ao longo da sequência de imagens.

O filtro de Kalman consiste num método recursivo para estimar o estado de um sistema linear, minimizando a média do erro quadrático. Este algoritmo é constituído por duas etapas, a predição e a filtragem.

Por predição entende-se o processo de inferir o estado do sistema a partir de observações anteriores, no caso específico deste projecto este estado será o ponto  $(x, y)$  representante das pupilas (coordenadas).

A filtragem pretende actualizar o estado actual do sistema a partir da leitura de dados ruidosos, melhorando a predição anterior com base na nova observação. Assim a filtragem serve para atenuar sinal indesejado, por exemplo ruído de leitura.

No caso específico deste projecto as observações utilizadas são os pontos  $(x, y)$  obtidos como representantes da pupila, de modo a que o ruído destas observações seja filtrado e seja obtido, assim, um seguimento mais suave da posição da pupila.

#### 3.3.1 Modelo, Processo e Iteração

O filtro de Kalman pode ser decomposto em três passos:

- Construção do modelo;

- Inicialização do processo;
- Iteração.

O primeiro passo, construção do modelo, é o passo mais importante. É neste passo que são construídos os modelos do mundo que são utilizados na filtragem. Podem ser, por exemplo, de velocidade constante ou aceleração constante.

O modelo do estado do sistema é dado pela equação 3.5 onde  $x_k$ , representa o estado actual do sistema no instante  $k$ ,  $A$ , a matriz que relaciona os estados em instantes consecutivos de tempo,  $B$  relaciona o controlo de entrada opcional  $u$  com o estado.

No caso do seguimento de um objecto, o seu vector do estado poderá ser a posição  $(x, y)$  e a velocidade  $(vx, vy)$  e considerando um movimento de velocidade constante.

$$x_k = A x_{k-1} + B u_{k-1} + w_{k-1} \quad (3.5)$$

$$z_k = H x_k + v_k \quad (3.6)$$

Para além da equação 3.5 o filtro de Kalman considera que a medida  $z_k$  do estado  $x_k$  é dado pela equação 3.6. Os parâmetros  $w_{k-1}$  e  $v_k$  representam o ruído gaussiano associado ao processo e à medição, respectivamente. As equações 3.7 e 3.8 representam as distribuições de probabilidades das variáveis  $w$  e  $v$  respectivamente, onde  $Q$  e  $R$  são as medidas de covariância para o processo de filtragem e medição.

$$p(w) = N(0, Q) \quad (3.7)$$

$$p(v) = N(0, R) \quad (3.8)$$

No segundo passo é a inicialização do processo desencadeado pelo filtro de Kalman. Após uma adaptação bem sucedida do modelo ao problema é desencadeado o processo de iteração.

As equações 3.9, 3.10, 3.11, 3.12 e 3.13 representam o algoritmo recursivo do filtro de Kalman. Onde a equação 3.9 corresponde ao ganho Kalman; a equação 3.10 representa a actualização da predição; a equação 3.11 representa a actualização da covariância; e as duas últimas equações 3.12 e 3.13 representam a predição em  $k + 1$ .

$$K_k = P_k H^T (H P_k H^T + R)^{-1} \quad (3.9)$$

$$\hat{x}_k = A \hat{x}_{k-1} + B u_k \quad (3.10)$$

$$P_k = A P_{k-1} A^T + Q \quad (3.11)$$

$$\hat{x}_k = \hat{x}_k + K_k (z_k - H \hat{x}_k) \quad (3.12)$$

$$P_k = (I - H K_k) P_k \quad (3.13)$$

Com a construção do modelo no passo anterior as matrizes  $A$ ,  $B$  e  $H$  são conhecidas na maioria das vezes. Restam, assim, duas matrizes a serem determinadas,  $R$  e  $Q$ . A matriz  $R$  é facilmente determinada porque por norma tem-se o conhecimento do ruído do sistema, esta representa a covariância do ruído da medição feita pelo sistema. A matriz  $Q$  por outro lado tem em conta o estado do sistema pelo que se torna mais difícil de determinar, esta representa covariância do ruído do processo.

Ambas são descritas nas equações 3.14 e 3.15 respectivamente.

$$R_k = E[w_k + w_k'] \quad (3.14)$$

$$Q_k = E[v_k + v_k'] \quad (3.15)$$



# Capítulo 4

## Análise

Neste capítulo apresenta-se a análise efectuada, para a concepção do sistema, ao nível de requisitos funcionais, casos de utilização, desenho aplicacional, estrutura de ficheiros e arquitectura de classes.

### 4.1 Requisitos Funcionais

Para isto pretende-se que o sistema tenha os seguintes requisitos:

1. Utilize hardware comercial;
2. Execução em tempo real;
3. Precisão suficiente para utilização comum do computador;
4. Seja intuitivo;
5. Com calibração simples e rápida.

Visto já existirem sistemas semelhantes ao pretendido mas que utilizam hardware dedicado e dispendioso, o primeiro ponto é crucial pois pretende-se utilizar hardware comum (câmara web). disponibilizado nos equipamentos actuais, na tentativa de democratizar a tecnologia devido facto dos sistemas existentes terem um custo demasiado alto para a público alvo.

Os restantes pontos são importantes devido à experiência que transmitem ao utilizador. Se o sistema não conseguir operar em tempo real não é usável para o utilizador, assim como, se não possuir a precisão necessária, a execução de acções torna-se difícil. Ser intuitivo e ter uma calibração simples ou praticamente nula realça a qualidade de utilização que o utilizador terá.

## 4.2 Requisitos Operacionais

Afim de garantir os requisitos funcionais o sistema tem ser capaz de:

1. Recolher a informação proveniente do utilizador;
2. Processar a informação recolhida;
3. Localizar o foco do olhar do utilizador na respectiva coordenada do ecrã;
4. Realizar acções básicas do rato;
5. Comunicar o estado.

O primeiro ponto está intimamente associado com o primeiro requisito funcional. Para recolher a informação com a maior acuidade possível o hardware utilizado tem que cumprir algumas premissas de qualidade. Por sua vez o segundo ponto relaciona-se muito proximamente com o segundo requisito funcional. O processamento da informação obtida deverá ser optimizado de forma a que o sistema tenha possibilidade de funcionar em tempo real. Por tempo real entende-se que o sistema consiga efectuar todo o processamento que necessita para funcionar dentro do refrescamento da câmara, de forma a não perder *frames* com informação relevante. Assim, o sistema tem disponível o intervalo de tempo entre a aquisição de duas imagens consecutivas para efectuar todo o processamento. Os requisitos operacionais permitem ao utilizador utilizar o sistema de forma precisa, intuitiva e simples.

## 4.3 Casos de utilização

Os casos de utilização estão apresentados na figura 4.1 e são:

1. Iniciar o sistema;
2. Iniciar calibração;
3. calibrar;
4. iniciar seguimento ocular;

5. movimento cursor;
6. realizar acções;
7. terminar seguimento ocular.

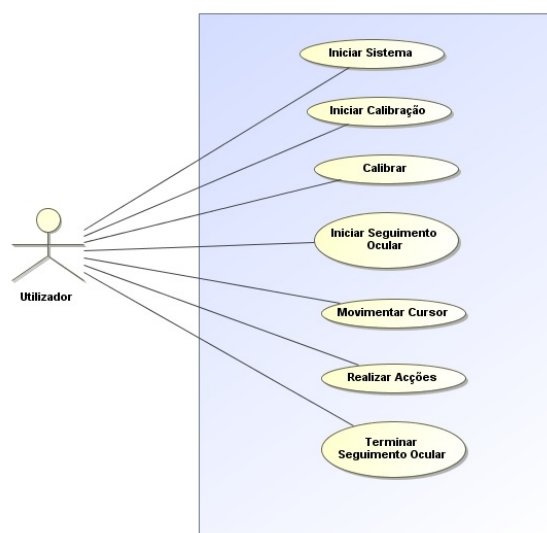


Figura 4.1: Casos de utilização gerais

No primeiro caso de utilização, o utilizador inicia o sistema de forma a poder usufruir da interface. No segundo caso de utilização, "iniciar calibração", o utilizador transmite a informação à aplicação para iniciar o processo de calibração. Em seguida, o utilizador efectua a calibração (caso de utilização - "Calibrar"). Com o sistema calibrado, o utilizador envia o comando de início do seguimento ocular, ou seja, inicia a interface, caso de utilização "Iniciar Seguimento Ocular". Em seguida efectua o movimento do cursor como forma de interacção com o sistema. Por fim, o utilizador pode realizar acções e parar o seguimento ocular, por exemplo, para efeito de descanso (casos de utilização "Realizar Acções" e "Terminar Seguimento Ocular" respectivamente).

Após avaliar os casos de utilização, estes foram redefinidos de forma a dar prioridades aos mais relevantes para a conclusão, com sucesso, do projecto. Os casos de utilização são apresentados consoante o esforço de investigação, desenvolvimento e a sua necessidade para a conclusão do estudo da interface pessoa-máquina (ver figura 4.2). Alguns casos de utilização foram retirados por não serem fundamentais para o bom funcionamento do sistema.

Mantiveram-se então os casos de utilização sem os quais o sistema não faz sentido existir, e são, dispostos por grau de importância:

1. Iniciar Sistema;
2. Calibrar;
3. Movimentar Cursos;
4. Realizar Acções.

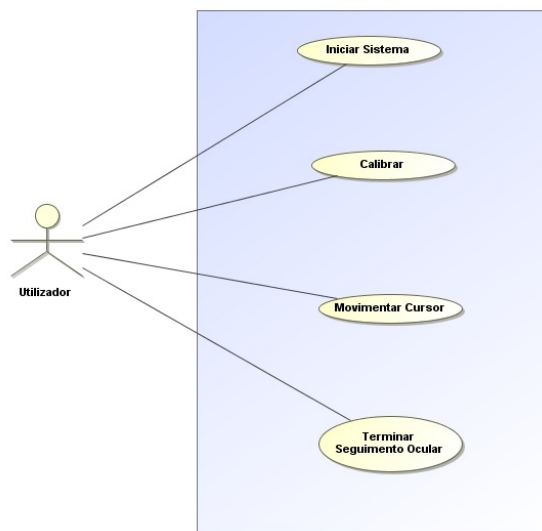


Figura 4.2: Casos de utilização após definição de prioridades

## 4.4 Modelo Proposto

Com os casos de utilização definidos foi projectado o modelo que o sistema deverá seguir. Na figura 4.3 apresentam-se os blocos que definem o sistema. O primeiro bloco é o responsável por adquirir as imagens da câmara. Os restantes representam os blocos responsáveis por extrair e processar a informação relevante das imagens adquiridas pela câmara. O último bloco, não só extrai informação como processa as acções que o utilizador pretende realizar.

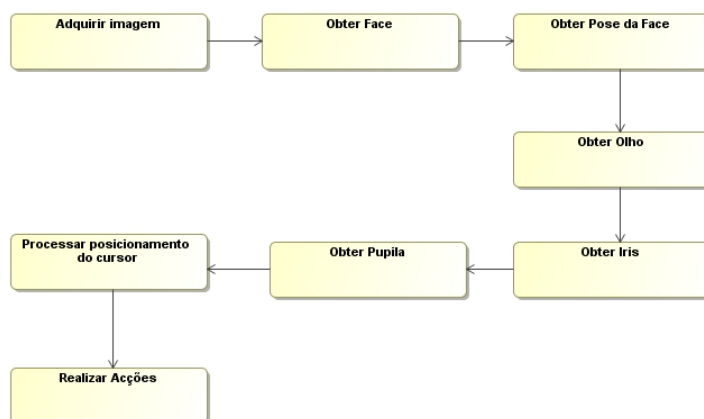


Figura 4.3: Modelo proposto

O primeiro bloco, "Adquirir imagem", é a obtenção da informação proveniente da câmara web utilizada. Em seguida efectua-se a detecção dos elementos da face necessários ao desenvolvimento do sistema (blocos "Obter face", "Obter Pose da Face", "Obter olho", "Obter Íris" e "Obter Pupila"). Com os elementos adquiridos é efectuado o processamento da informação de modo a obter a posição do cursor, bloco "Processar posicionamento do cursor". Em seguida é possível ao sistema efectuar acções, bloco "Realizar acções".

A implementação dos vários blocos poderá variar conforme o hardware utilizado, deste modo o diagrama 4.3 pretende expor uma visão global do funcionamento do sistema e será desenvolvido no capítulo 5.

#### 4.4.1 Diagrama de estados

Com o modelo apresentado na figura 4.3 é efectuado o diagrama de estados apresentado na figura 4.4. Este permite verificar a evolução dos estados do sistema, podemos assim melhor avaliar os vários passos do nosso sistema de forma um pouco mais pormenorizada do que através do modelo.

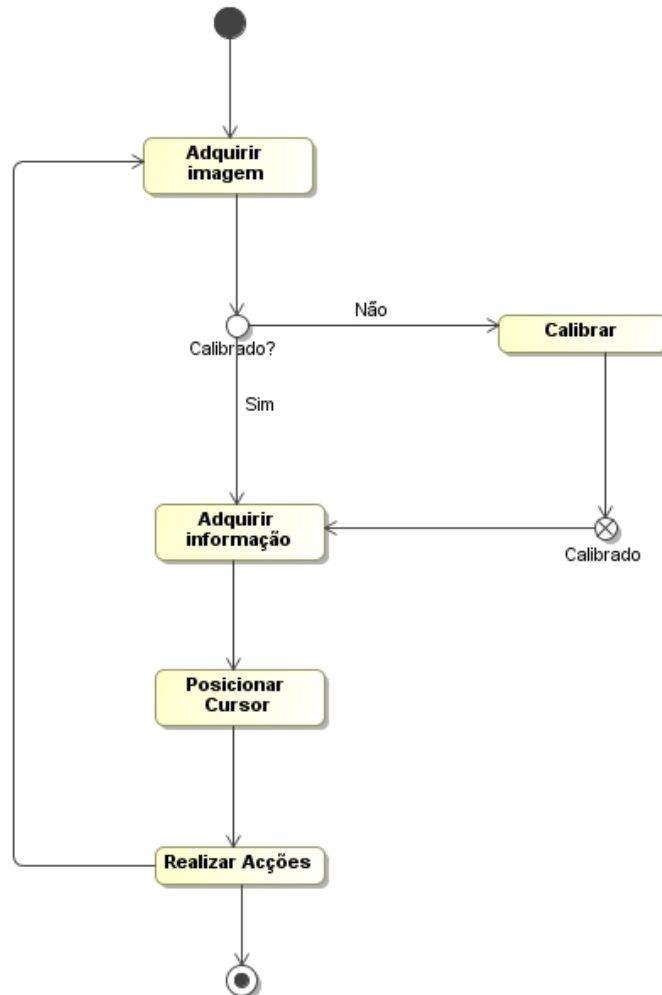


Figura 4.4: Modelo proposto

Verifica-se que o primeiro passo do sistema é a aquisição de imagens, em consonância com o modelo do sistema. O segundo passo do sistema pode variar dependendo se o sistema está ou não calibrado (primeira iteração) ou se é necessário proceder à calibração. Se for a primeira iteração o sistema terá que ser calibrado para o utilizador. Caso já tenha sido calibrado, o sistema vai obter a informação presente nas imagens adquiridas para o seu funcionamento. O estado seguinte pressupõe a utilização dessas informações adquiridas anteriormente para conseguir colocar o cursor no ecrã e consequentemente executar acções.

## 4.5 Arquitectura

Após a análise do sistema e com base nos diagramas de blocos, de estados e de casos de utilização foi desenvolvida a arquitectura do sistema. Esta tem como objectivo manter a coesão e diminuir a dificuldade de um sistema complexo como o que se pretende desenvolver.

### 4.5.1 Estrutura de ficheiros

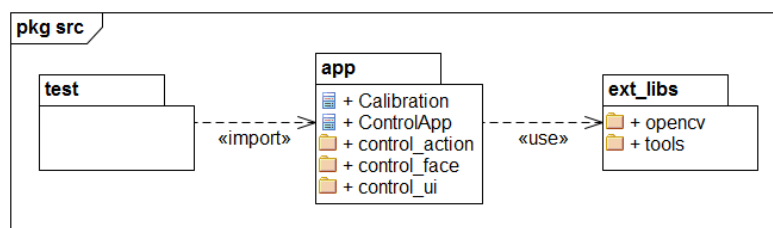


Figura 4.5: Estrutura de ficheiros do sistema

O sistema foi organizado em pastas e sub-pastas, figura 4.5, de forma a minimizar a complexidade, aumentando a coesão do sistema. A pasta *src* é a raiz de todo o código escrito para o funcionamento do sistema. Dentro desta podemos observar as pastas:

- *app* alias de aplicação - onde estão os pacotes e classes responsáveis pelo fluxo principal do sistema;
- *ext\_libs* alias de bibliotecas externas - onde estão os pacotes e classes utilizados, externos à aplicação. Contém ferramentas que podem ser alteradas sem se modificar o fluxo da aplicação;
- teste - onde está a classe responsável por executar a aplicação de forma a testar o seu funcionamento.

Ainda, dentro das três pastas citadas anteriormente podemos encontrar outras sub-pastas. Dentro de *app* encontram-se:

- *control\_face* alias de controlo de face - Neste pacote estão as classes responsáveis por efectuar todas as operações e processamento relativos à face. Como detecção de olhos, nariz e face;

- *control\_action* alias de controlo de acções - Este pacote detém as classes responsáveis por processar e executar as acções pretendidas pelo utilizador;
- *control\_ui* alias de controlo de interface de utilizador - Neste pacote estão as classes responsáveis por definir o que é apresentado ao utilizador a quando da utilização do sistema.

Dentro da pasta das bibliotecas externas encontram-se as pasta:

- *opencv* - nesta pasta estão as classes responsáveis pela utilização de várias funcionalidades da biblioteca OpenCV, assim como, algoritmos de processamento de imagem;
- *tools* alias de ferramentas - nesta pastas estão algumas ferramentas/algoritmos utilizados no desenvolvimento do sistema.

Fora da *src* encontra-se a pasta com o *sdk* utilizado, *RSSDK* é alias para *RealSense SDK*. Este *sdk* permite o acesso a várias funcionalidades da câmara da Intel.

### 4.5.2 Diagramas de classes

Enquanto a estrutura de ficheiros é construída, assim, se define a arquitetura do sistema em termos de classes. Na figura 4.6 encontra-se o diagrama de classes correspondente ao sistema desenvolvido. No mesmo diagrama podemos visualizar o diagrama da classe que faz o controlo da aplicação, este controlo é responsável por gerir e adquirir os recursos necessários para o sistema funcionar. Assim, a coesão e a complexidade do sistema são diminuídas.

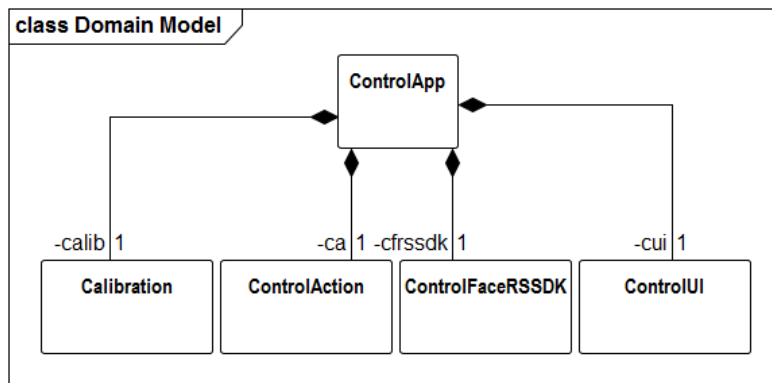


Figura 4.6: Diagrama de classes - *ControlApp*

De igual forma apresentam-se nas figuras 4.7, 4.8 e 4.9 os diagramas de classes correspondentes a cada controlo do sistema, respectivamente *control\_action*, *control\_face* e *control\_ui*. Cada uma das figuras contém a informação necessária para implementar a estrutura do sistema.

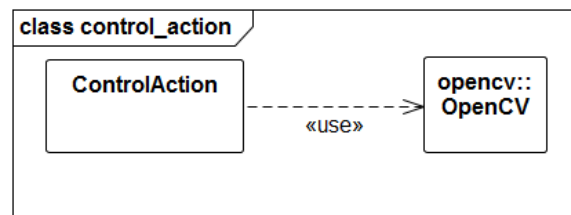


Figura 4.7: Diagrama de classes - *ControlAction*

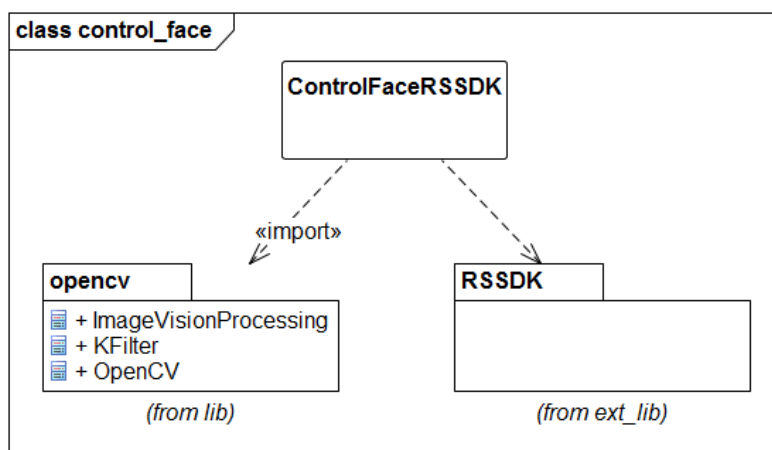


Figura 4.8: Diagrama de classes - *ControlFaceRSSDK*

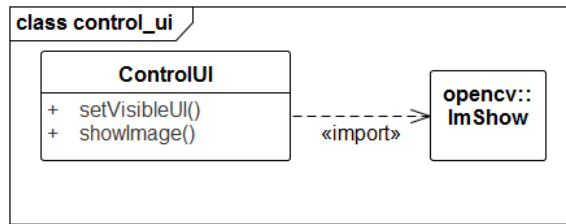


Figura 4.9: Diagrama de classes - *ControloUI*

No diagrama 4.7 verifica-se a presença da classe *ControlAction* e da biblioteca externa OpenCV, devido a algumas variáveis que são usadas serem implementadas pela biblioteca.

Esta classe guarda e processa a informação necessária para replicar a acção pretendida pelo utilizador no sistema. Por exemplo, na acção do piscar de olho, a classe controlo da aplicação 4.6 pergunta ao controlo de acção se o utilizador piscou o olho e qual a acção que o sistema deve executar. O controlo de acção verifica se o olho foi piscado e envia o comando para executar a funcionalidade do sistema.

Este diagrama (4.8) é o que representa a classe com mais peso e complexidade do sistema, *ControlFaceRSSDK*. O controlo de face, é a classe mais extensa pois é a responsável por efectuar todo o processamento de imagem necessário para transmitir a informação ao controlo de aplicação.

Verifica-se que a classe *ControlFaceRSSDK* utiliza várias funcionalidades disponibilizadas por outras entidades; utiliza a classe *ImageVisionProcessing* para conseguir incorporar a imagem proveniente da câmara com o tipo de dados suportado pelo OpenCV e para executar várias acções extensas de processamento de imagem. A classe *KFilter* é responsável por implementar o filtro de Kalman e possibilitar ao sistema o uso do mesmo de forma simples e eficaz.

É utilizado o *sdk* da câmara da Intel, *Real Sense*. Este permite ao sistema aceder a determinadas funcionalidades da câmara que possibilitam o processamento e aumentam o desempenho do sistema. É necessário salientar um tipo de objecto, o *pxcmStatus*, este permite ao sistema conhecer o estado da câmara, se está ligada, se houve algum erro, se as drivers não estão a funcionar e se a câmara está conectada. Este tipo de informação torna a utilização deste objecto fundamental para o bom funcionamento do sistema.

O diagrama 4.9 representa o controlo de interface de utilizador. Este

controlo activa ou desactiva a interface de utilizador. É este controlo que permite ao utilizador ver o fluxo de imagem. Utiliza uma classe, desenvolvida com ferramentas do OpenCV, para mostrar as imagens, *Imshow*. Esta classe reúne os dados e as funções necessárias para a imagem poder ser visualizada pelo utilizador.

### 4.5.3 Diagrama classes - Teste

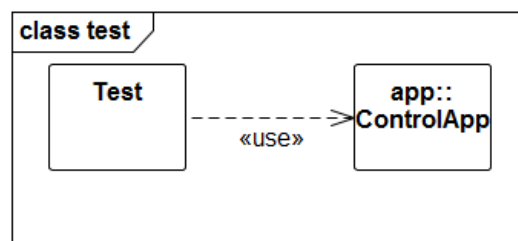


Figura 4.10: Diagrama de classes - Teste

Apresenta-se na figura 4.10 a arquitectura da classe responsável por arrancar o sistema e assim permitir efectuar os testes.

A classe Teste dispõe apenas de duas variáveis *ca* e *fechar* que representam, respectivamente, o controlo de aplicação e um booleano com a informação se é para terminar a aplicação.

### 4.5.4 Diagrama classes - bibliotecas externas

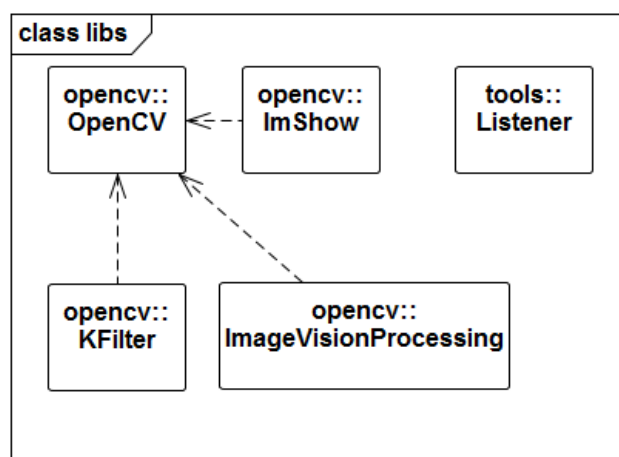


Figura 4.11: Diagrama de classes - Bibliotecas externas

No diagrama 4.11 encontra-se as classes que pertencem às bibliotecas externas. Dentro da pasta *opencv* estão as classes que utilizam funções da biblioteca OpenCV e que permitem auxiliar o controlo de face a processar as imagens provenientes da câmara.

A pasta *tools* contém o *listener* que permite ao controlo da aplicação verificar se o sistema ainda está a correr. É com esta funcionalidade que o método principal do sistema está a correr infinitamente até o *listener* dar a indicação contrária. Este herda as suas funcionalidades de uma classe embutida no *jre* utilizado.

# Capítulo 5

## Implementação

Após terminada a análise da arquitectura é executada a programação do sistema seguindo o modelo proposto na figura 4.3, no capítulo 4, secção 4.4.

### 5.1 Desenvolvimento

O primeiro passo da implementação do modelo é a implementação da classe de Teste e o controlo de aplicação. Estas duas classes vão permitir inicializar o sistema e controlar as acções realizadas pelo mesmo.

Assim, a classe de *Teste* é responsável por inicializar o controlo de aplicação e coloca-la a correr. Ao executar o método *run()* da classe *ControlApp* as funções vão ser executadas de forma sequencial, como descrito no modelo 4.4.

A primeira etapa da implementação do sistema é a aquisição de imagem *RGB* proveniente da câmara da Intel, a Real Sense. Para isso são utilizados vários métodos das classes *ControlFaceRSSDK*. Com a utilização de um *sdk* específico da Intel para a câmara são necessárias várias etapas.

A Real Sense constitui uma tecnologia recente, que implementa várias fontes de informação, RGB, profundidade e infravermelhos. Como objectivo principal, a Intel pretende difundir esta tecnologia e torná-la padrão para ser embutida em vários dispositivos. Este dispositivo está disponível apenas para desenvolvimento.

Para a utilização da câmara é necessário definir uma série de parâmetros. É necessário ter uma variável que vai guardar o estado da câmara. Por estado entende-se a situação da câmara, se está conectada, se houve algum problema a abrir algum dos fluxos disponíveis, etc. Em seguida é criada uma variável que permite fazer a gestão dos sensores da câmara, designada de *sense manager* alias gestor de sensores. Este objecto detém um papel fundamental na configuração da câmara.

Através desta variável são configurados os vários fluxos que vão ser utilizados, nomeadamente, profundidade, RGB ou infravermelhos. Embora o fluxo de infravermelhos possa parecer promissor, quando adquirido tem um resultado igual ao de profundidade (ver figura 5.2).

Para além da fonte de informação a utilizar o gestor, *ControlFaceRSSDK*, permite ainda passar informação ao micro processador da câmara que tipo de objecto vamos fazer o seguimento. Ao inicializar o gestor, após todas as configurações iniciais, este disponibiliza o estado dos sensores. Esta informação é importante para se compreender qual o estado da câmara (caso algo esteja mal programado, esta informação vai servir para fazer *debugging*).

De igual forma são inicializados os outros dois controlos do sistema, interface de utilizador e de acção. O controlo de interface de utilizador coloca os elementos visuais e atribui o nome à janela utilizada para dar o feedback ao utilizador. O controlo de acções apenas é inicializado e fica pronto para ser utilizado pelo controlo de aplicação.

Com os controlos inicializados é efectuado o passo seguinte, na realidade vamos executar o primeiro passo do modelo proposto 4.3. Assim a imagem é obtida do fluxo *RGB* da câmara, junto com a imagem é utilizada a *sdk* da Intel para obter a posição da face (figura 5.1).

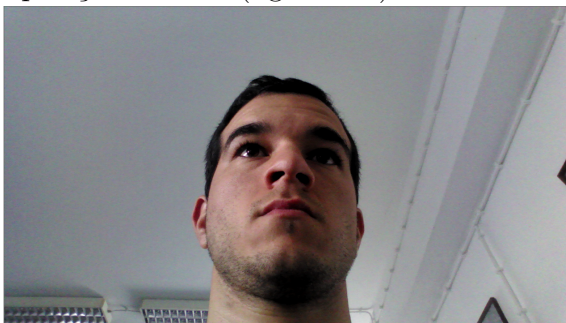


Figura 5.1: Face obtida a partir da câmara Intel Real Sense - Fluxo *RGB*

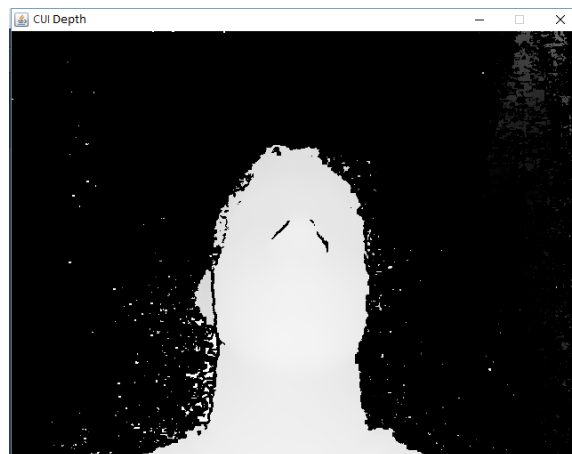


Figura 5.2: Imagem obtida a partir da câmara Intel Real Sense - Fluxo profundidade

Com a face detectada passamos à extracção das características de forma a obter os olhos e as pupilas. Para tal foram tomadas duas abordagens, a primeira utilizando a câmara Real Sense como uma webcam normal, utilizando apenas o fluxo *RGB*. A segunda abordagem consistiu na utilização das funcionalidades disponibilizadas pelo sdk, utilizando, nomeadamente, a informação de profundidade.

### 5.1.1 Câmara web - RGB

Para simular o sistema com uma abordagem testada anteriormente em [23] [10] foi utilizado apenas um fluxo de informação RGB, da Intel Real Sense. Assim, utilizando apenas a imagem RGB e algoritmos de processamento de imagem foram encontrados os olhos, iris e pupilas.

Os algoritmos utilizados neste processo encontram-se no método *GetPupils* da classe *ImageVisionProcessing* e foram implementados com o auxílio da biblioteca OpenCV.

A primeira etapa deste processamento consiste em efectuar tratamento da imagem (tons de cinzento, equalizar histograma e aplicar *blur*), de forma a ser possível a utilização dos classificadores de Haar, secção 3.1, e transformada de Hough, secção 3.2. Os classificadores de Haar foram utilizados para detectar

a face e os olhos, enquanto que a transformada de Hough foi utilizada para encontrar os círculos correspondentes à íris.

Uma vez encontrados os olhos é aplicada um pequeno *blur* para suavizar os contornos dos olhos e é aplicada a transformada de Hough para encontrar o círculo correspondente à íris.

Os resultados obtidos foram semelhantes aos de [10] [23], comprovando que com este tipo de abordagem os resultados não são os melhores, muito derivado pelas restrições do hardware.

Mais à frente são apresentadas duas figuras, 5.10 e 5.11, que permitem verificar a detecção da pupila através dos dois métodos abordados neste trabalho, detecções com os três fluxos da Intel Real Sense e o método tradicional, que utiliza apenas o fluxo RGB.

### 5.1.2 Câmara web - Intel Real Sense

Após a execução e experimentação de alguns exemplos da *sdk* e alguns problemas com os controladores procedeu-se à implementação do sistema.

É necessário voltar definir algumas variáveis no gestor de sensores da câmara, foi activado o modo de obter pontos de referência e permitida a detecção de face, olhos e nariz. Com esta configuração é encontrada a face e a respectiva pose. A pose da face tem como referência o eixo da câmara e a origem no centro da cabeça, na figura 5.3 ilustra-se esta relação.

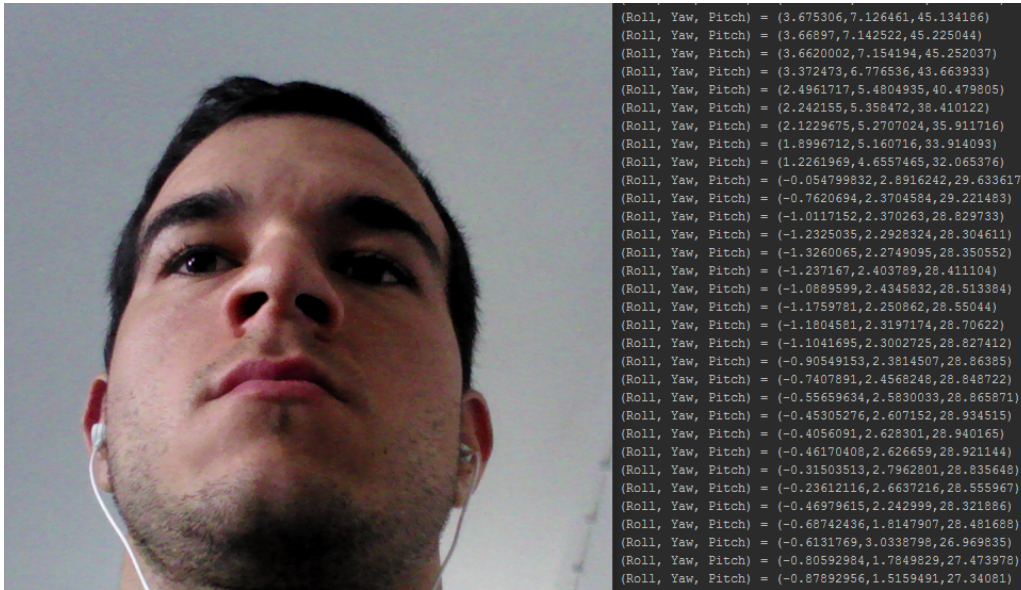


Figura 5.3: Relação entre face e a câmara - Pose da face

Para obtermos ângulos adaptados à nossa realidade foi utilizada uma função de Euler para os obter. Os ângulos de Euler foram introduzidos por Leonhard Euler [26] e utilizam três variáveis para descrever a orientação de um corpo rígido no espaço euclidiano. É a mesma tecnologia utilizada na aeronáutica para se conhecer a orientação da aeronave 5.4.

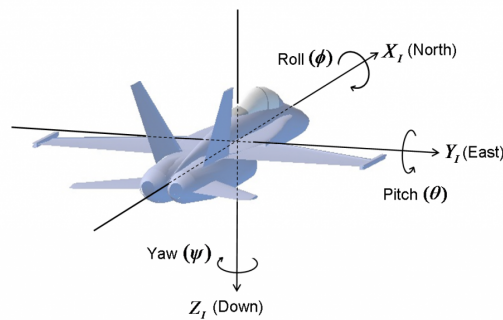


Figura 5.4: Ângulos de Euler aplicados à aeronáutica

Com a pose da face e os pontos de referência podemos passar à detecção do olho e dos respectivos pontos de referência. A câmara Real Sense funciona por profundidade, ou seja, detecta as várias partes da face devido à profundidade da cabeça. Em primeiro lugar é detectado o nariz, figura 5.5, e obtidos os seus pontos de referência, só depois são encontrados os olhos e os respectivos

pontos de referência da íris de forma semelhantes aos restantes pontos de referência 5.6.



Figura 5.5: Ponto de referência do nariz

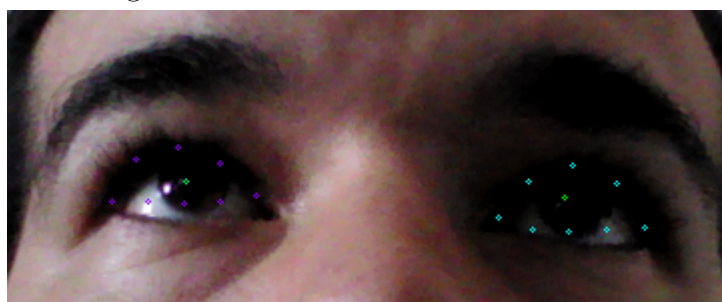


Figura 5.6: Pontos de referência do Olho - Pupila

Embora o *sdk* permita a obtenção dos pontos de referência dos olhos, a sua detecção tem muito ruído, o que iria impedir uma utilização estável de sistema. O ruído são as pequenas variações na detecção que fazem com que os pontos de referência oscilem, mesmo quando os olhos estão parados.

Para colmatar o ruído da detecção foi implementado o filtro de Kalman descrito no capítulo 3 na secção 3.3. Com a implementação do filtro de Kalman foi possível suavizar o seguimento dos pontos e assim obter uma trajectória mais natural do movimento dos olhos. Nas figuras 5.7 e 5.8 verifica-se a evolução do movimento dos olhos e a suavidade do movimento após o filtro de Kalman.

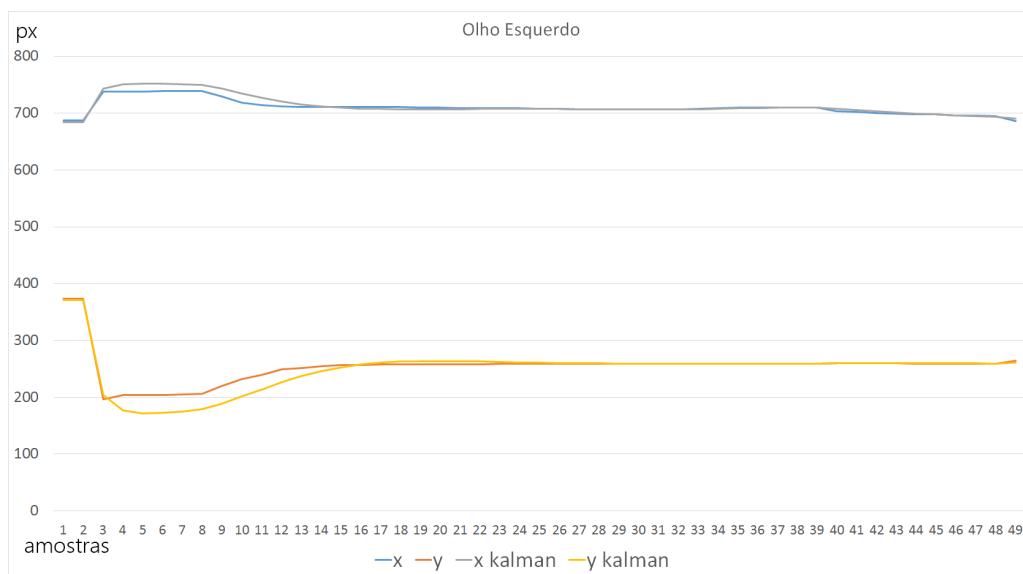


Figura 5.7: Comparação entre a detecção do olhos sem e com filtro de Kalman - Olho esquerdo

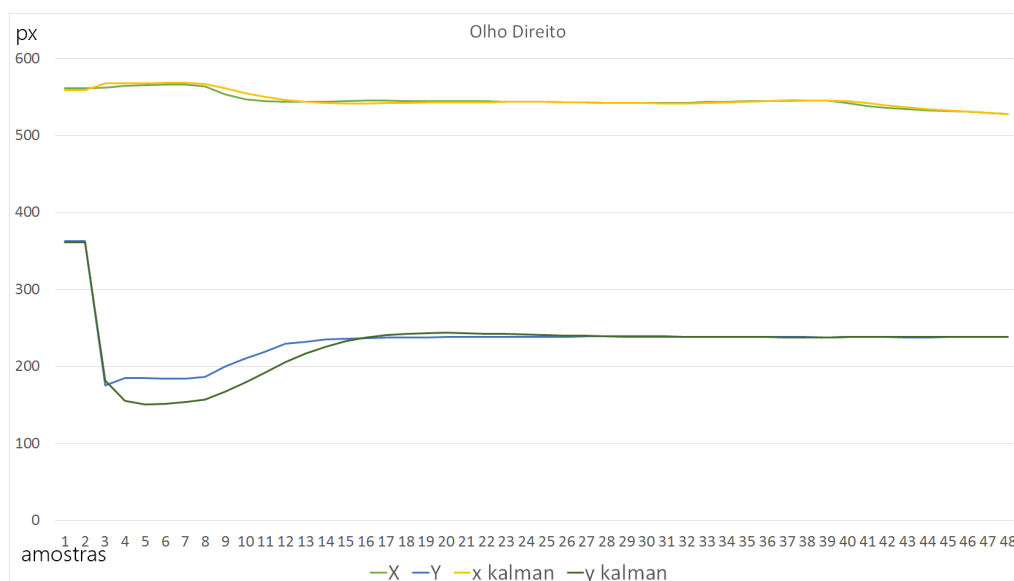


Figura 5.8: Comparação entre a detecção do olhos sem e com filtro de Kalman - Olho direito

Os gráficos das figuras 5.7 e 5.8 mostram que, sem a aplicação do filtro de Kalman, o seguimento dos movimentos oculares torna-se demasiado ruidoso e incapacita a possibilidade do sistema funcionar em condições. Nas imagens também é visível, como mostra na figura 5.9 onde se mostram os dois pontos

de referência da pupila, com e sem Kalman. O ponto vermelho corresponde à coordenada do olho com o filtro de Kalman aplicado e o ponto verde é o normal. A imagem 5.9 foi obtida após abrir os olhos, podendo verificar que o ponto vermelho está atrasado, isto deve-se ao facto do filtro de Kalman predizer qual a posição como consequência do movimento brusco dos olhos.

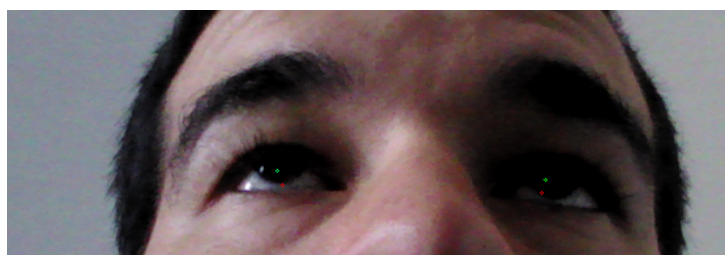


Figura 5.9: Ponto referência da pupila com Kalman (vermelho) e sem Kalman (verde)

Todas as operações acima referidas são efectuadas dentro do controlo de face comandado pelo controlo de aplicação. O controlo de aplicação apenas necessita de invocar quatro métodos do controlo de face, um para obter a imagem *RGB*, outro para criar um objecto que permite obter a face, outro para obter os dados para processar sobre a face e outro para obter os pontos de referência dos olhos.

Por fim é possível efectuar uma comparação entre os dois métodos. Nas figuras 5.10 e 5.11 apresenta-se a sequência, em que o utilizador está a olhar para a frente e em seguida olha para cima. Na abordagem tradicional é necessário abrir as pálpebras de forma a se conseguir obter, com melhor precisão, o círculo da íris. Quando o utilizador olha para frente ambos os métodos têm bons resultados mas, quando o utilizador olha para cima, a pupila é perdida.

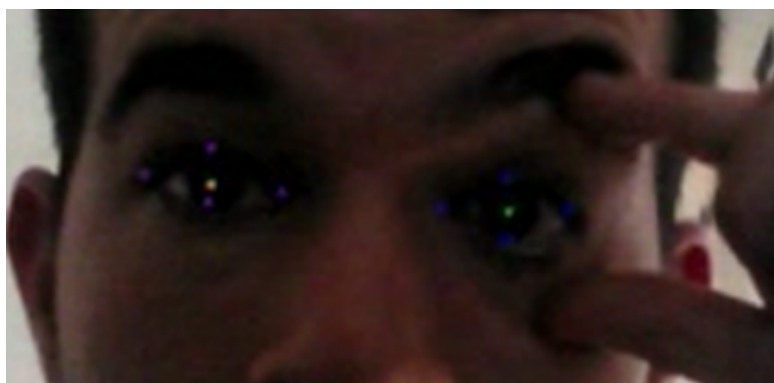


Figura 5.10: Comparação entre a detecção com três fluxos de informação e o método apenas com o fluxo RGB - Olhar frente

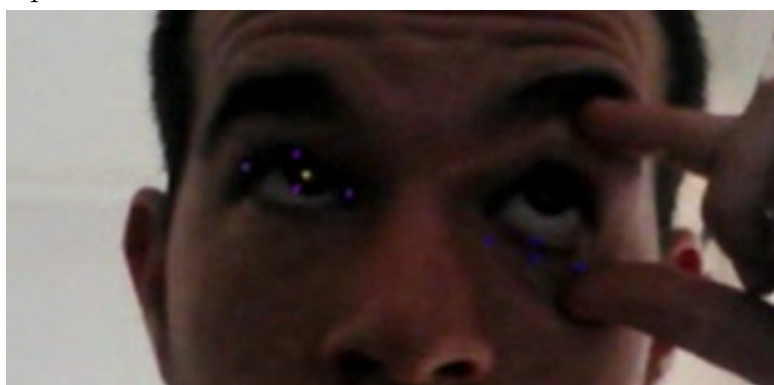


Figura 5.11: Comparação entre a detecção com três fluxos de informação e o método apenas com o fluxo RGB - Olhar cima

Verifica-se que a detecção pelo método da Intel consegue acompanhar a pupila e o método tradicional perdeu o seguimento da pupila, mal o utilizador olhou para cima. Tendo em conta estes resultados iniciais, terem sido pouco robustos, e com base nos trabalhos [10] e [23], a abordagem, só com o fluxo de informação RGB, foi abandonada.

### 5.1.3 Acções

Utilizando as técnicas de detecção e seguimento descritas nas secções anteriores, foram implementados métodos que permitem interagir com o computador - acções. Implementou-se:

- movimento do cursor através do seguimento ocular;

- clique do botão esquerdo e direito do rato através do fecho do olho esquerdo e direito respectivamente;
- movimento do cursor através do seguimento do nariz;
- clique esquerdo do rato com a abertura da boca.

Nas figuras 5.12 e 5.13 ilustra-se as acções que foram implementadas.

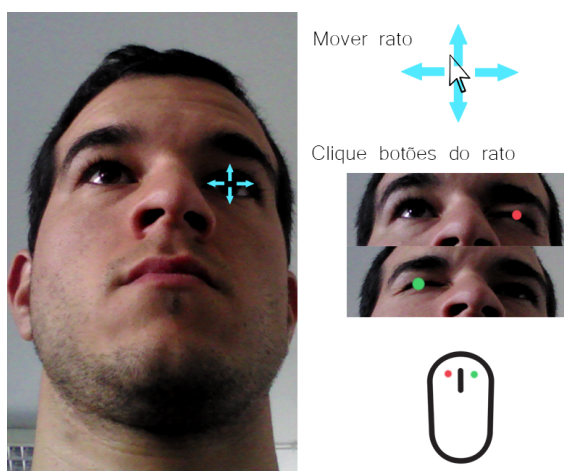


Figura 5.12: Ilustração das acções implementadas com os olhos

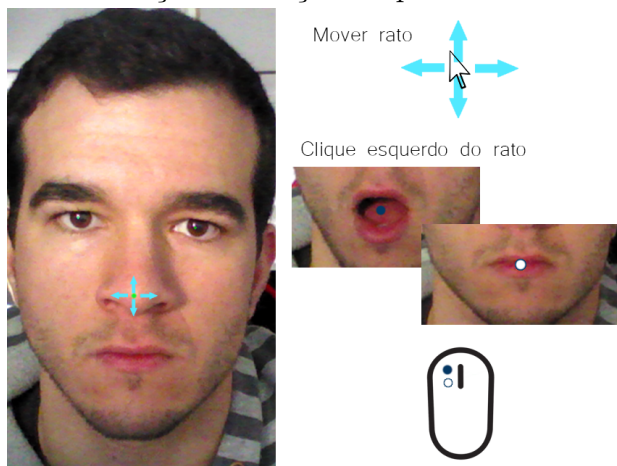


Figura 5.13: Ilustração das acções implementadas com o nariz e boca

A primeira acção definida é a mais complexa devido à sensibilidade dos movimentos oculares, assim como, ao facto da pupila se movimentar muito pouco dentro do globo ocular. Como o movimento da pupila dentro da área

de movimento não tem uma variação superior a quarenta pixeis, torna a sensibilidade do rato uma tarefa difícil de controlar.

Foram testados cinco métodos como tentativa de movimentar o cursor do rato com o seguimento dos olhos e um método, que se verificou eficiente, para movimentar o cursor com o nariz. Os métodos são:

- Olho como sendo um rato de computador;
- Olho com calibração;
- Olho com representação do ecrã;
- Olho esquerdo e direito com representação do ecrã;
- Nariz com representação do ecrã.

Foram implementadas, de igual forma, dois métodos para efectuar as acções de clique dos botões do rato, com os olhos, esquerdo e direito, e com o abrir e fechar da boca.

### **Olho como sendo um rato de computador**

Este método consiste na simulação de olho como sendo um rato, a funcionar de forma diferencial sobre o seu eixo ( ver figura 5.14).

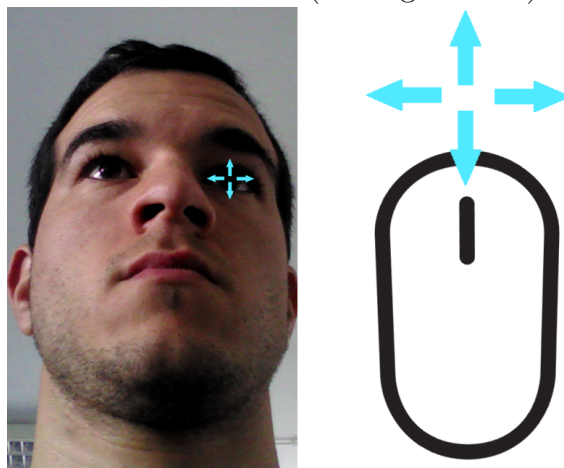


Figura 5.14: Movimento ocular a simular um rato, com movimentos diferenciais

Neste método há dois problemas que se salientam, o eixo sobre o qual o olho se movimenta não é possível de simular da forma pretendida e não é possível parar o rato de forma a ajustar o movimento, como se faz com o rato tradicional. Este factores fizeram com que os resultados fossem pouco práticos, não foi possível fazer um uso normal do rato.

### Olho com calibração

Esta abordagem, mais tradicional, semelhante à utilizada anteriormente em [23] e [10], consiste na utilização de um sistema de calibração. Para efectuar a calibração o utilizador tem que olhar para nove pontos no ecrã (ver figura 5.15), com o objectivo de se obter duzentas amostras das coordenadas dos olhos para cada ponto. Para cada nuvem de amostras é encontrado o ponto médio, de forma a se poder comparar o valor que deveria ser. Com o intuito de efectuar a distância entre o ponto médio da nuvem e o ponto medido dos olhos é utilizada a distância euclidiana e/ou de manhattan.

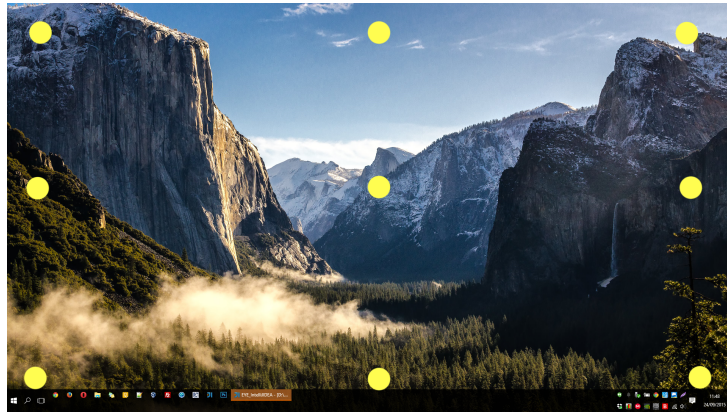


Figura 5.15: Ilustração do ecrã utilizado para calibrar

Dos testes realizados consideraram que o utilizador se encontrava sentado, numa posição confortável, em frente ao ecrã. Nestas circunstâncias o olho apenas mexe cerca de três pixels, o que é impraticável e inviabiliza este método.

### Olho com representação do ecrã

Este método foi o que obteve resultados mais promissores. Considerou-se os limites de movimento do olho como definição dos limites do ecrã. Assim sendo, o valor mínimo das coordenadas que o olho alcança, em relação aos extremos do olho (ver figura 5.16), definem a origem do ecrã (canto superior esquerdo); o mesmo se passa para o valor máximo que define a posição máxima no ecrã (canto inferior direito). Na figura 5.16 ilustra-se esta associação.

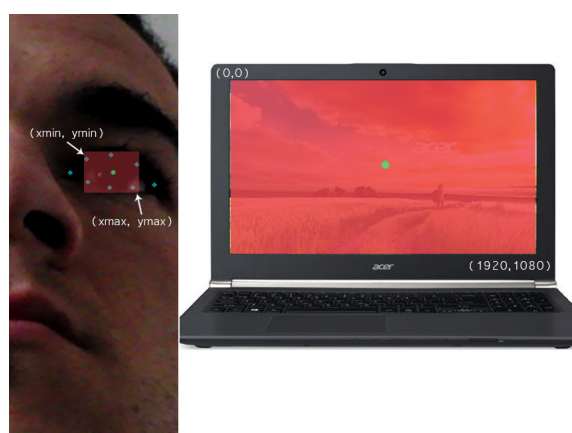


Figura 5.16: Ilustração da associação das coordenadas do olho com posições do ecrã

Por exemplo, considerando  $xmin = 100$ ,  $xmax = 150$ ,  $ymin = 75$  e  $yymax = 105$ , e que o olho está a olhar na posição central, ou seja,  $x = 125$  e  $y = 90$ . Temos que:

$$newx = \frac{x-xmin}{xmax-xmin} * screenwidth \quad (5.1)$$

$$newy = \frac{y-ymin}{ymax-ymin} * screenheight \quad (5.2)$$

Com um ecrã de resolução de alta definição onde  $screenwidth = 1920$  e  $screenheight = 1080$  temos que as coordenadas  $(newx, newy)$  são:

$$newx = \frac{125-100}{150-100} * 1920 = 960$$

$$newy = \frac{90-75}{105-75} * 1080 = 540$$

Neste caso o utilizador estava a olhar para o centro do ecrã e a projecção do rato é igualmente o centro do ecrã.

### Olho esquerdo e direito com representação do ecrã

Para complementar método anterior, foi introduzida a combinação dos dois olhos, o que permitiu obter melhores resultados. Para estimar a posição do cursor no ecrã foi encontrado o valor médio entre a posição do olho esquerdo e direito. A posição do olho esquerdo e direito foi encontrada utilizando o método anterior, 5.1.3. Para determinar o ponto médio e assim colocar o cursor na coordenada certa foi feita a diferença em  $x$  e  $y$  para ambas as posições.

Na equação 5.3 e 5.4 encontram-se as fórmulas usadas para o cálculo, onde  $le\_newx$  e  $re\_newx$  representam os valores de  $x$  para o olho esquerdo e direito, respectivamente; as variáveis  $le\_newy$  e  $re\_newy$  representam respectivamente os valores de  $y$  para o olho esquerdo e direito; as variáveis,  $bothx$  e  $bothy$ , originadas pela aplicação das formulas representam a nova coordenada para o posicionamento do cursor.

$$bothx = \frac{le\_newx + re\_newx}{2} \quad (5.3)$$

$$bothy = \frac{le\_newy + re\_newy}{2} \quad (5.4)$$

### Piscar de olho para simular o clique esquerdo e direito do rato

Após a implementação do movimento do rato implementou-se outra acção comum do rato, clique esquerdo e direito, respectivamente com olho esquerdo e direito.

O processo será explicado para o olho esquerdo, pois com o olho direito o processo repete-se. Para esta implementação foi necessário obter os pontos de referência em torno do olho. Após aquisição dos pontos de referência do olho foi calculada uma distância máxima e mínima a que os pontos de referência se encontram quando estão abertos e fechados. Essas distâncias foram calculadas através da observação dos valores que os pontos de referência apresentavam e conseqüentemente a sua média. Na figura 5.17 ilustra-se a obtenção dos valores mínimos a considerar.

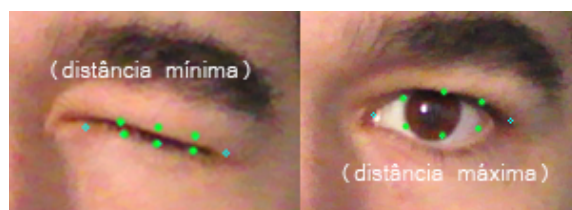


Figura 5.17: Pontos de referência do olho esquerdo fechado e aberto

Assim, com as distâncias calculadas, é possível definir um limiar onde o olho será considerado fechado e outro onde o olho será considerado aberto. Com estas funcionalidade podemos associar os cliques do rato ao estado do olho, aberto ou fechado.

### Nariz com representação do ecrã

A utilização do nariz como apontador para o cursor do rato é uma tarefa semelhante à anterior 5.1.3. É encontrado o ponto que representa a ponta do nariz e esse ponto é usado como apontador. Para fazer a projecção do movimento do nariz para o cursor do rato foi usada a mesma técnica do método 5.1.3. É definida uma janela na qual o nariz se move e essa janela representa o ecrã. A formula utilizada apresenta-se nas equações 5.1 e 5.2. Na figura 5.18 apresenta-se uma representação gráfica da metodologia descrita anteriormente.

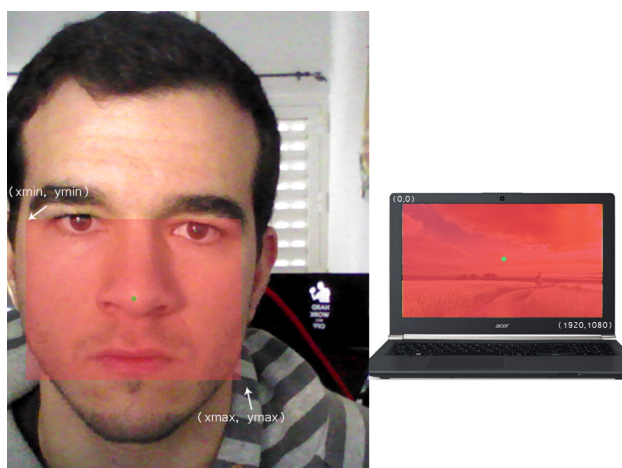


Figura 5.18: Ilustração da associação das coordenadas do nariz com posições do ecrã

Com este método obtiveram-se bons resultados, sendo possível movimentar o cursor pelo ecrã, como desejado pelo utilizador.

### **Abertura da boca para simular o clique esquerdo do rato**

Esta funcionalidade compreende a utilização da boca para simular a mesma funcionalidade que o piscar do olho esquerdo, ou seja, efectuar o clique esquerdo do rato. A implementação desta funcionalidade utilizando a boca é semelhante ao do olho esquerdo, contudo existe uma diferença. No método anterior 5.1.3 é o piscar do olho que simula o clique do rato, neste método é o abrir e fechar da boca mas, se o utilizador mantiver a boca aberta o botão do rato continua premido o que permite efectuar operações diferentes, mais elaboradas.



Figura 5.19: Pontos de referência da boca

Na figura anterior 5.19 apresentam-se os pontos que identificam a boca. Com os pontos de referência da boca é feito o mesmo processo de com os dos olhos para encontrar as distâncias mínimas à qual se atribui o estado boca fechada e boca aberta.

# Capítulo 6

## Validação e testes

O desenvolvimento deste projecto centrou-se na utilização da nova tecnologia da Intel, a câmara Real Sense. Esta câmara possui três fontes de informação: *RGB*, profundidade e infravermelhos.

O desenvolvimento deste sistema foi feito de forma incremental pelo que foram efectuados vários testes ao longo da implementação. Os primeiros testes consistiram na detecção da face, nas vertentes *RGB*, profundidade e infravermelhos.

Neste capítulo são efectuados testes e validações ao sistema. A abordagem que utiliza o fluxo *RGB* foi abandonada, devido aos resultados obtidos no capítulo 5, na secção 5.1.1, e aos resultados dos trabalhos [23] e [10].

Na figura 6.1 apresenta-se o diagrama de blocos que ilustra as etapas do sistema, desde a detecção da face até à realização de uma acção.



Figura 6.1: Diagrama de blocos do sistema

Para obter estes fluxos de dados é necessário efectuar vários tipos de configurações para que seja processado no microprocessador essa informação. Para o caso do fluxo *RGB* é necessário informar a câmara de qual a dimensão que se deseja obter na imagem, tendo uma resolução máxima de 1280x720. É necessário ainda informar a câmara que tipo de detecção se pretende (face e

mão, por exemplo) e qual o fluxo de informação a que se deseja aceder.

Após aplicar as configurações desejadas é possível ter acesso aos pixels da imagem. Para os objectivos deste projecto foi necessário converter este tipo de objecto para um outro tipo compatível com o a biblioteca OpenCV. Assim obteve-se as imagens correspondentes ao fluxo *RGB*, ver figura 6.2.

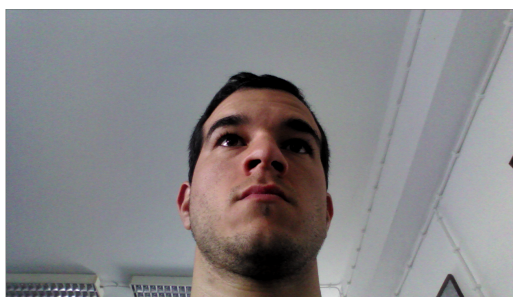


Figura 6.2: Imagem do fluxo RGB com objecto compatível com a biblioteca OpenCV

De forma muito semelhante é possível obter as imagens do fluxo de profundidade e infravermelhos. A resolução destes fluxos é definida pela câmara, sendo de 640x480.

Devido à tecnologia da câmara Intel Real Sense ser recente as imagens provenientes do fluxo profundidade e infravermelhos são iguais, ver figuras 6.3 e 6.4, para a linguagem de programação utilizada neste projecto (Java).

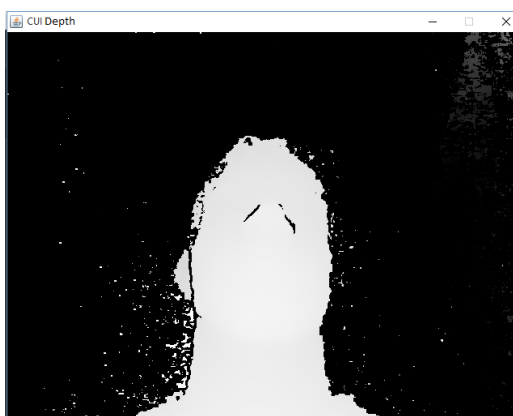


Figura 6.3: Imagem do fluxo de profundidade com objecto compatível com OpenCV

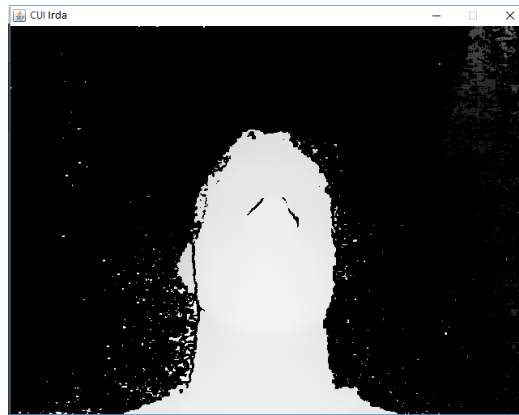


Figura 6.4: Imagem do fluxo de infravermelho com objecto compatível com OpenCV

O segundo passo consistiu na obtenção do olho e os seus pontos de referência. Esta etapa só é possível nas imagens *RGB* uma vez que as outras duas fontes não possuem informação suficiente para se conseguir aferir.

A detecção dos olhos funciona por profundidade, onde o sensor de profundidade emite um sinal e interpreta a sua reflexão. Esta tecnologia fica limitada em relação à distância a que os objectos se encontram do sensor, uma vez que se a face estiver muito perto não é possível fazer a interpretação correcta do sinal reflectido. Este factor leva a que exista uma distância mínima à qual deixa de ser possível detectar a posição dos olhos e os seus pontos de referência. A distância mínima foi aferida após alguns testes e é de, aproximadamente, vinte centímetros. Existe, igualmente uma distância máxima à qual a câmara deixa de conseguir obter o sinal com fiabilidade. Embora essa distância não seja tão relevante como a distância mínima, por razões de utilização e de restrições do sistema, foi calculada após alguns testes e verificou-se que a distância máxima é de sensivelmente um metro e meio. Um factor importante, que influencia a distância máxima, é a luminosidade. Se a compensação da luz adquirida pelo sensor de *RGB* tiver que ser realizada, pelo sensor de infravermelhos, esta distância diminui cerca de vinte centímetros devido às sombras e regiões escuras.

Cumprindo, então, a restrição da distância mínima é possível validar a detecção dos olhos e dos seus pontos de referência 6.5.

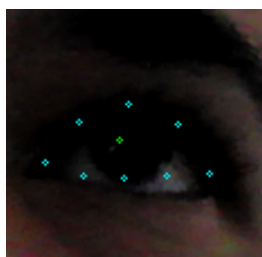


Figura 6.5: Validação da detecção dos olhos e seus pontos de referência

O passo seguinte é detectar a pupila e fazer o seu seguimento. A obtenção da pupila é feita de forma semelhante à dos olhos. É no entanto necessário ter em atenção que a sua detecção tem algumas oscilações devido à falta de contraste dentro do olho que a possa fazer salientar.

Ao ser obtido o ponto de referência da pupila, ponto central na figura 6.5, verificou-se que este sofria bastante oscilações entre as várias detecções, derivadas das oscilações de contraste na região da íris e de pequenos, quase imperceptíveis, movimentos da cabeça. Para colmatar este problema foi introduzido o filtro de Kalman, secção 3.3, com o intuito dos movimentos da pupila serem suavizados e diminuir o seu impacto no seguimento. No gráfico da figura 6.6 verifica-se a evolução ao longo do tempo das oscilações da detecção das coordenadas da pupila com e sem filtragem de kalman.

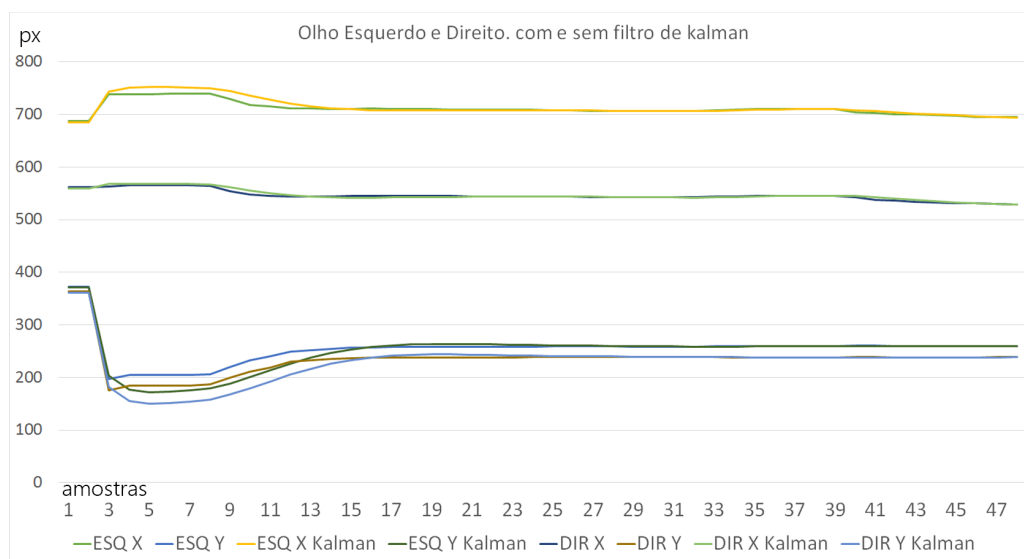


Figura 6.6: Oscilações da pupila - Kalman vs detecção

Para obter a comparação anterior, figura 6.6, foi considerado um padrão

de movimento realizado pelos olhos. O utilizador do sistema olha para os quatro cantos do ecrã na direcção dos ponteiros de relógio. As oscilações da posição dos olhos são visíveis através dos picos existentes no gráfico.

Após obter os pontos de referência dos olhos, de forma a ser possível efectuar os testes às várias implementações das acções do sistema, descritas no capítulo 5, foi executado um processo semelhante para o nariz, com o intuito de testar a implementação.

Assim, terminada a fase da detecção e processamento do seguimento dos olhos, são feitos os testes e validações às acções implementadas. Como referido no capítulo 5, foram implementados cinco métodos para movimentar o cursor, dos quais apenas dois mostraram resultados promissores.

O método que mostrou melhores resultados utiliza ambos os olhos para conseguir efectuar o processamento do cursor e encontra-se descrito na secção 5.1.3. Este método considera a janela definida pelos pontos de referência do olho (ver figura 5.6) como sendo os limites do ecrã, e faz o escalamento da posição da pupila para a posição no ecrã segundo as equações 5.1 e 5.2. Para este método foram efectuados ensaios de usabilidade e testes de precisão. As figuras 6.7 e 6.8 mostram um exemplo de utilização do foco do olhar para posicionar o cursor.

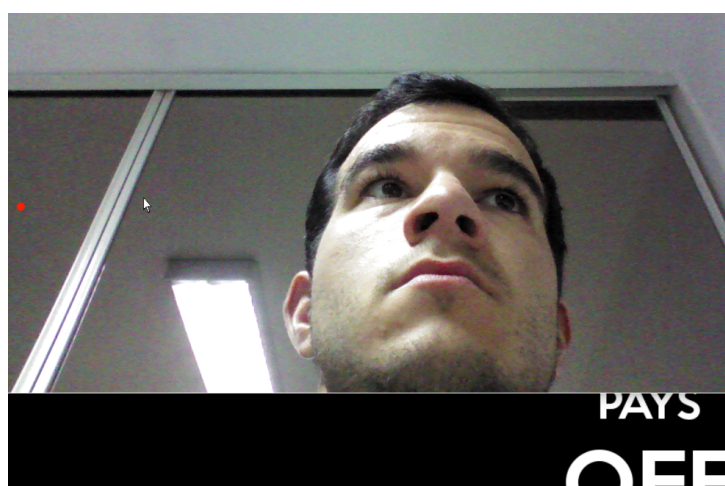


Figura 6.7: Movimento do rato - olhar para esquerda do ecrã

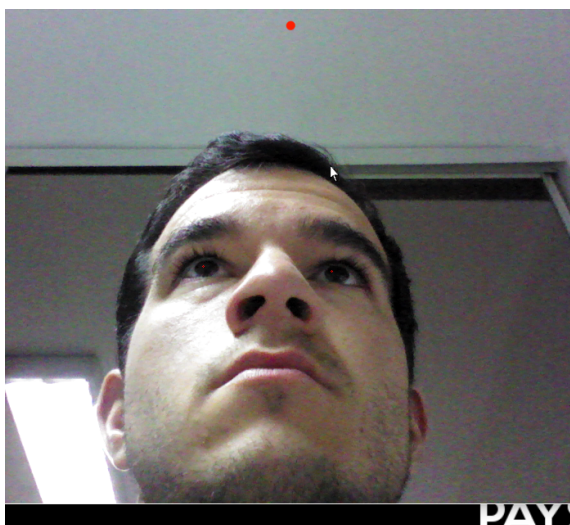


Figura 6.8: Movimento do rato - olhar para topo do ecrã

Como as figuras 6.7 e 6.8 são pouco elucidativas, apresenta-se na secção 6.1 os gráficos da posição dos olhos e do cursor obtidos nos testes. Estes incluem cem amostras para cada posicionamento definido pela pupila, com e sem filtro de kalman. As posições de teste podem ser vistas na figura 6.9.



Figura 6.9: Ilustração das posições do ecrã utilizadas para os testes  
Os teste efectuados foram executados em quatro ambientes diferentes:

- Com ecrã de resolução *FullHD* de quinze polegadas;

- Com ecrã de resolução *FullHD* de vinte e quatro polegadas;
- Com ecrã de resolução *HD* de quinze polegadas;
- Com ecrã de resolução *HD* de vinte e quatro polegadas;

## 6.1 Testes e validação de movimento do cursor com os olhos

Os primeiros teste efectuados foram com um ecrã de quinze polegadas com resolução de 1920 pixels de largura e 1080 de altura, para o método que utiliza os dois olhos para posicionar o rato no ecrã (secção 5.1.3). Nos gráficos apresentados nas figuras 6.10, 6.11 e 6.12 verifica-se a dispersão dos valores dos pontos de referência das pupilas, com e sem filtro de Kalman, a olhar para as várias posições utilizadas para teste.

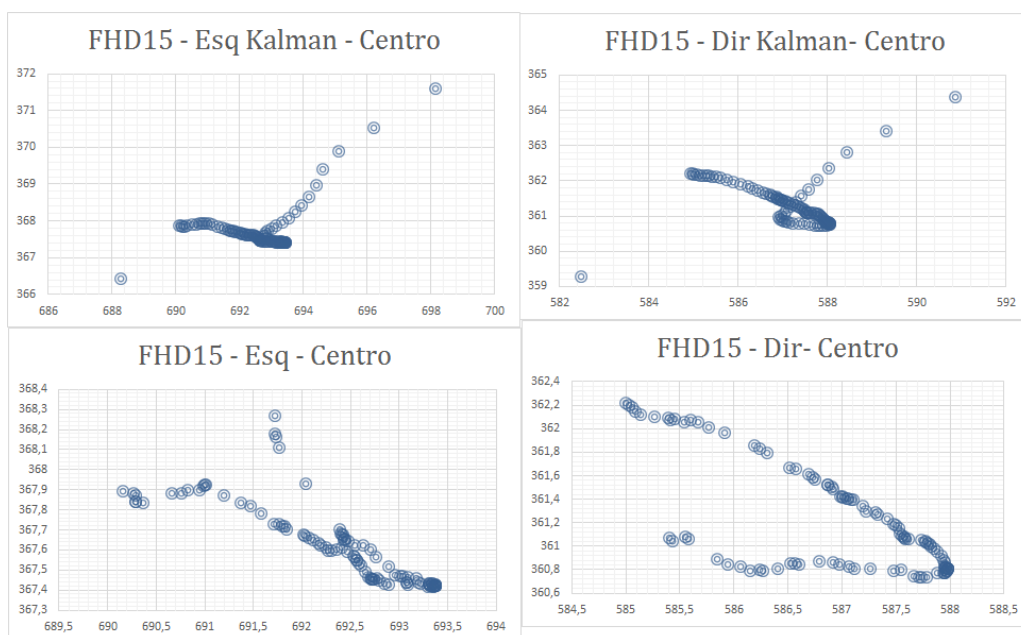


Figura 6.10: Gráfico de dispersão - Ecrã 15" *FullHD* - Olho esquerdo e direito - Olhar para o centro

No gráfico 6.10 verifica-se a evolução dos pontos de referência quando o utilizador está focado no centro do ecrã. Idealmente os pontos no gráfico não deveriam estar muito dispersos, mas, devido a ruído na detecção e luminosidade,

dade do ambiente, estes não se apresentam todos nas mesmas posições. Os pontos com menor densidade nos gráficos possibilitam verificar o movimento da pupila até convergir, idealmente, para o centro.

A mesma situação é verificada para os outros gráficos.

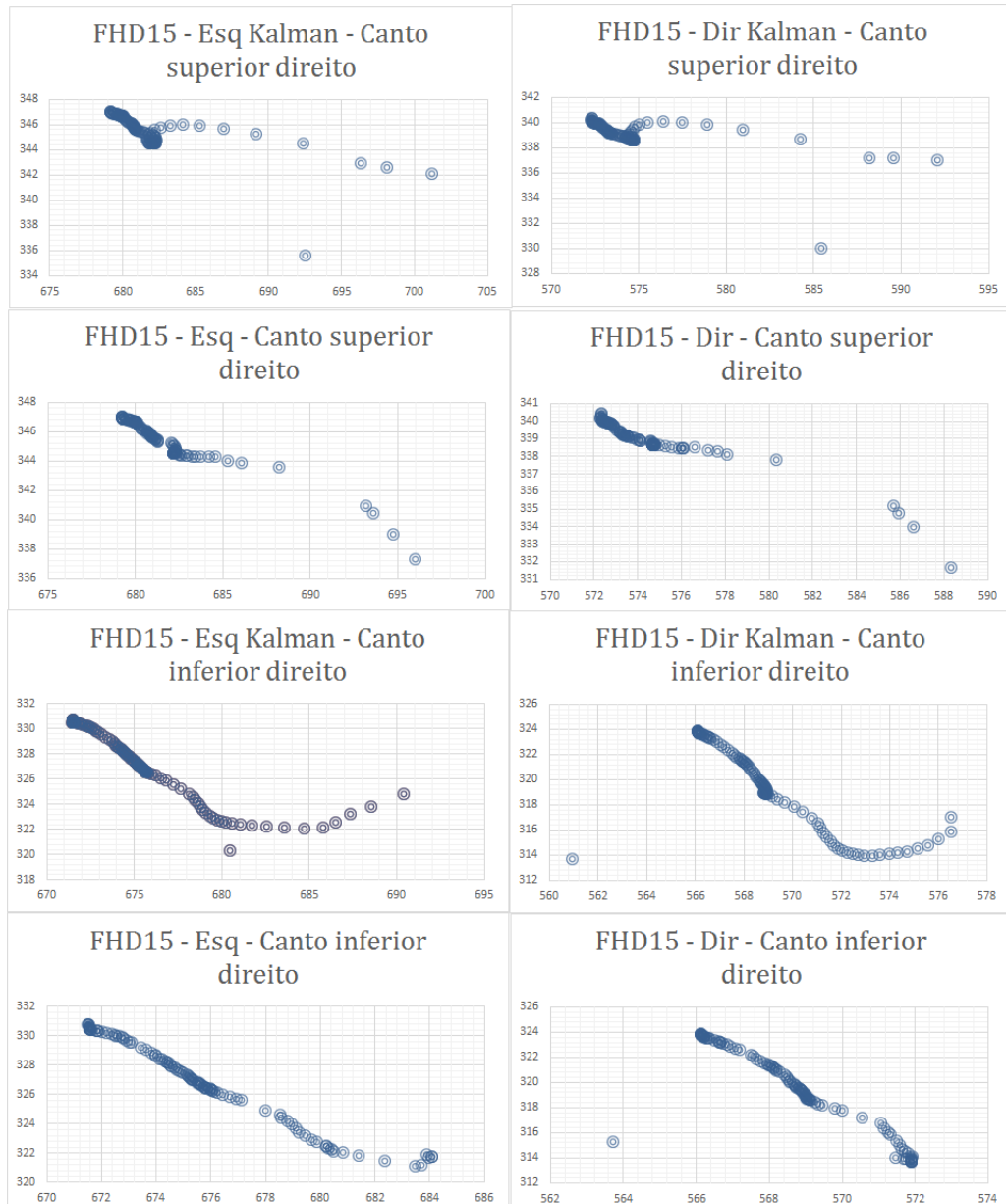


Figura 6.11: Gráfico de dispersão - Ecrã 15" *FuUHD* - Olho esquerdo e direito - Olhar para o canto superior e inferior direito.

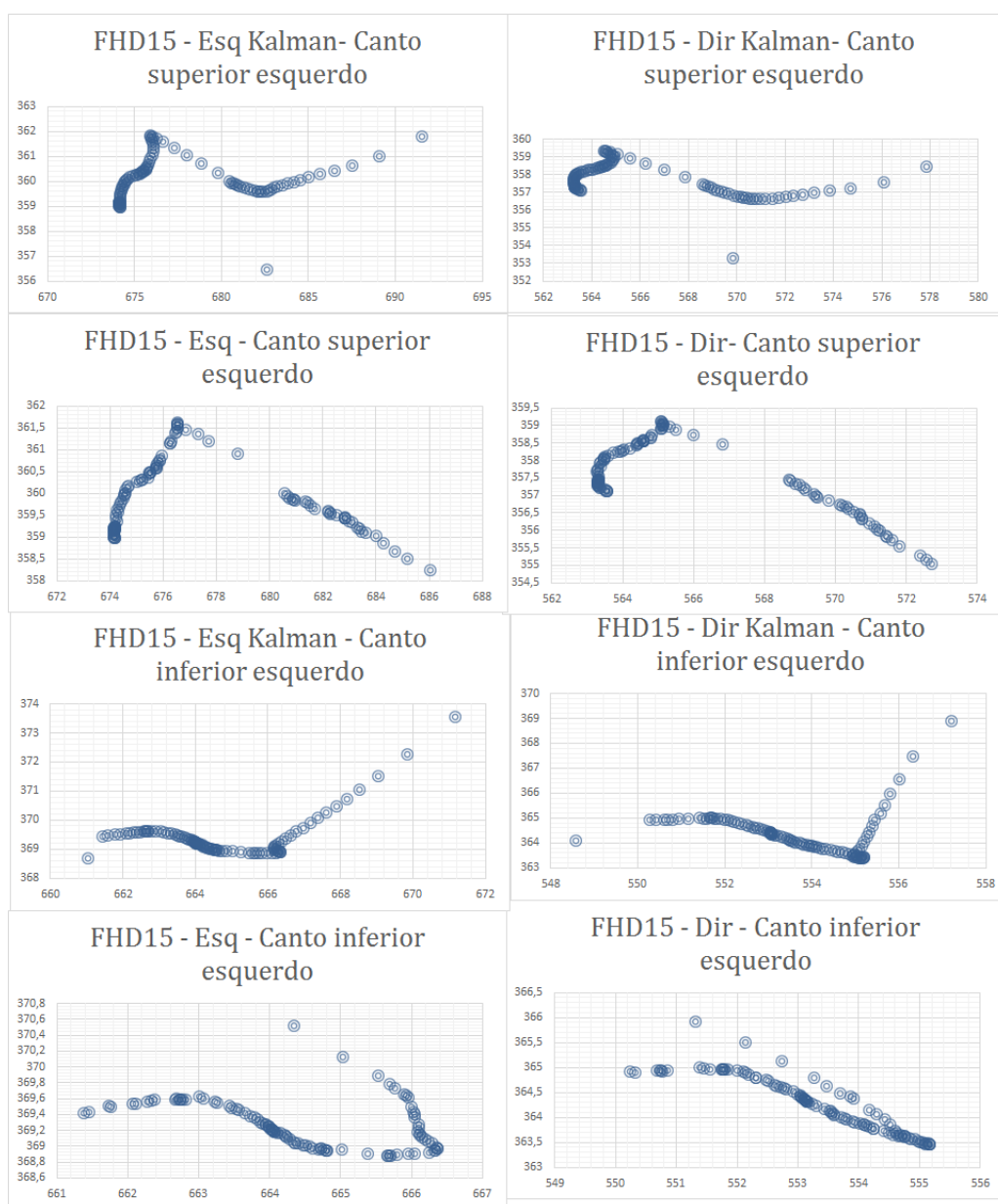


Figura 6.12: Gráfico de dispersão - Ecrã 15" *FullHD* - Olho esquerdo e direito - Olhar para o canto superior e inferior esquerdo.

O arrastamento e a dispersão nas densidades dos pontos verifica um erro grande no seguimento da pupila. Nos gráficos em 6.13 é apresentada a evolução da posição do rato para cada uma das posições de teste; o ponto vermelho representa o objectivo. Nestes gráficos a origem do ecrã (coordenada (0,0)) é situada no canto inferior esquerdo dos gráficos.

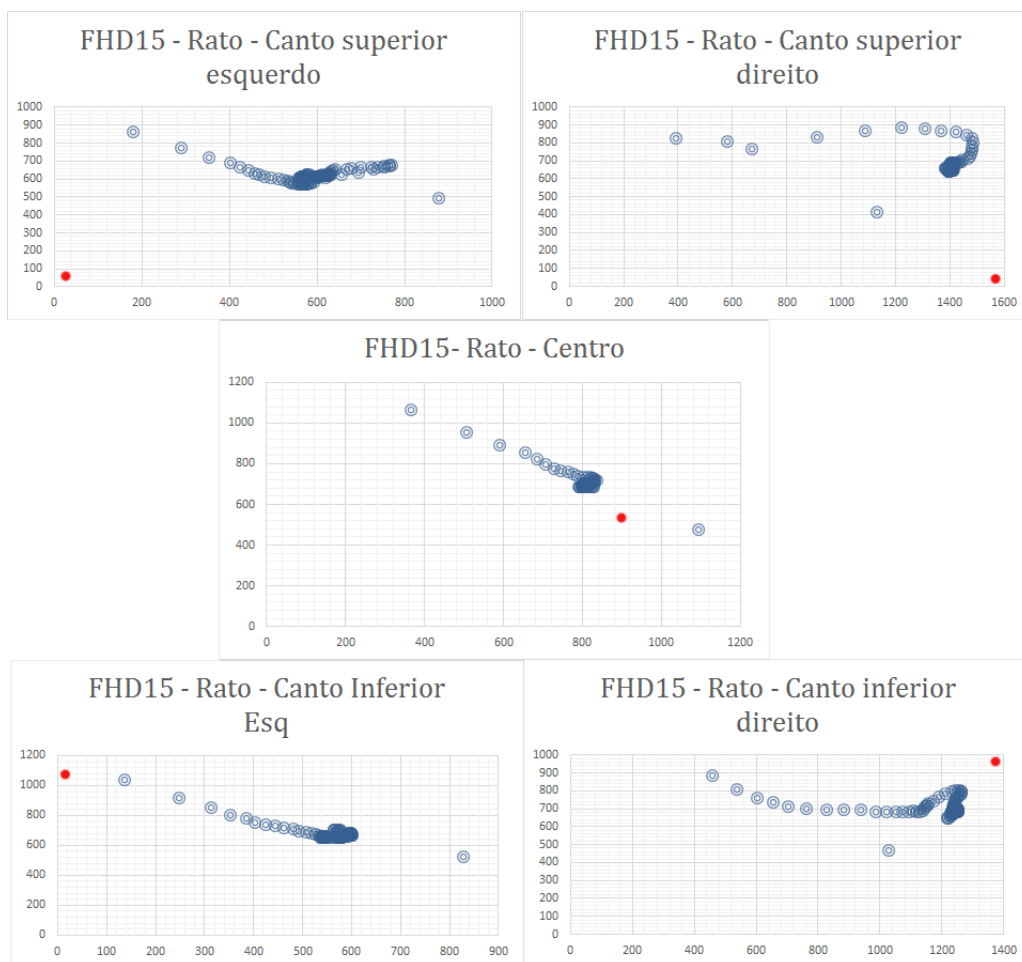


Figura 6.13: Gráfico de dispersão - Ecrã 15" *fullhd* - Rato - Quando o utilizador olha para os vários pontos de teste no ecrã.

Os mesmos testes foram efectuados para um ecrã de vinte e quatro polegadas com resolução *fullhd*. Os resultados obtidos, para o movimento da pupila, são apresentados nas figuras 6.14, 6.15, 6.16; os resultados para o posicionamento do rato são apresentados na figura 6.17.

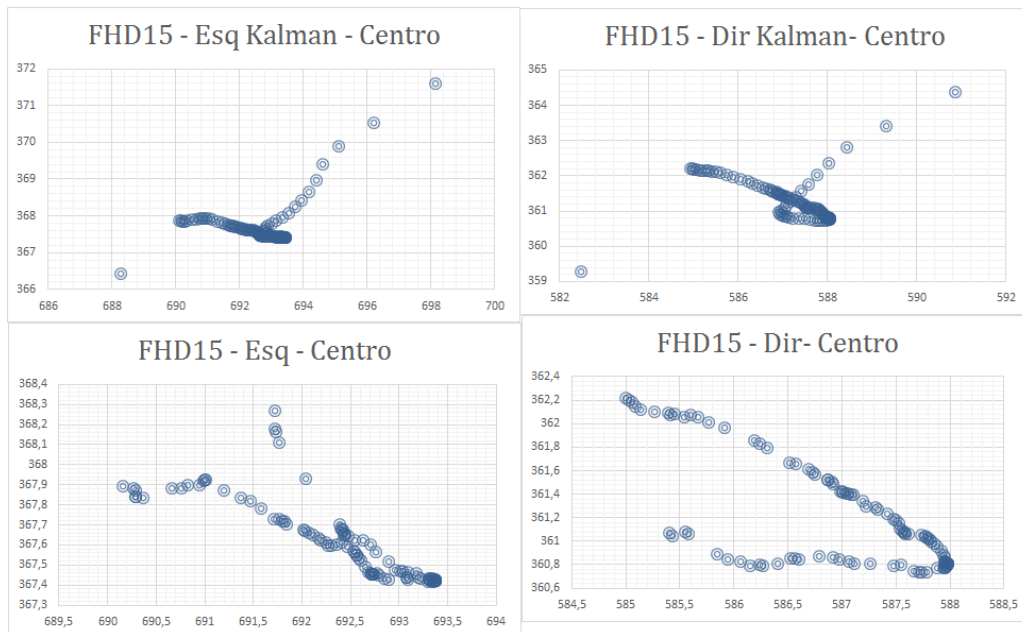


Figura 6.14: Gráfico de dispersão - Ecrã 24" *FullHD* - Olho esquerdo e direito - Olhar para o centro

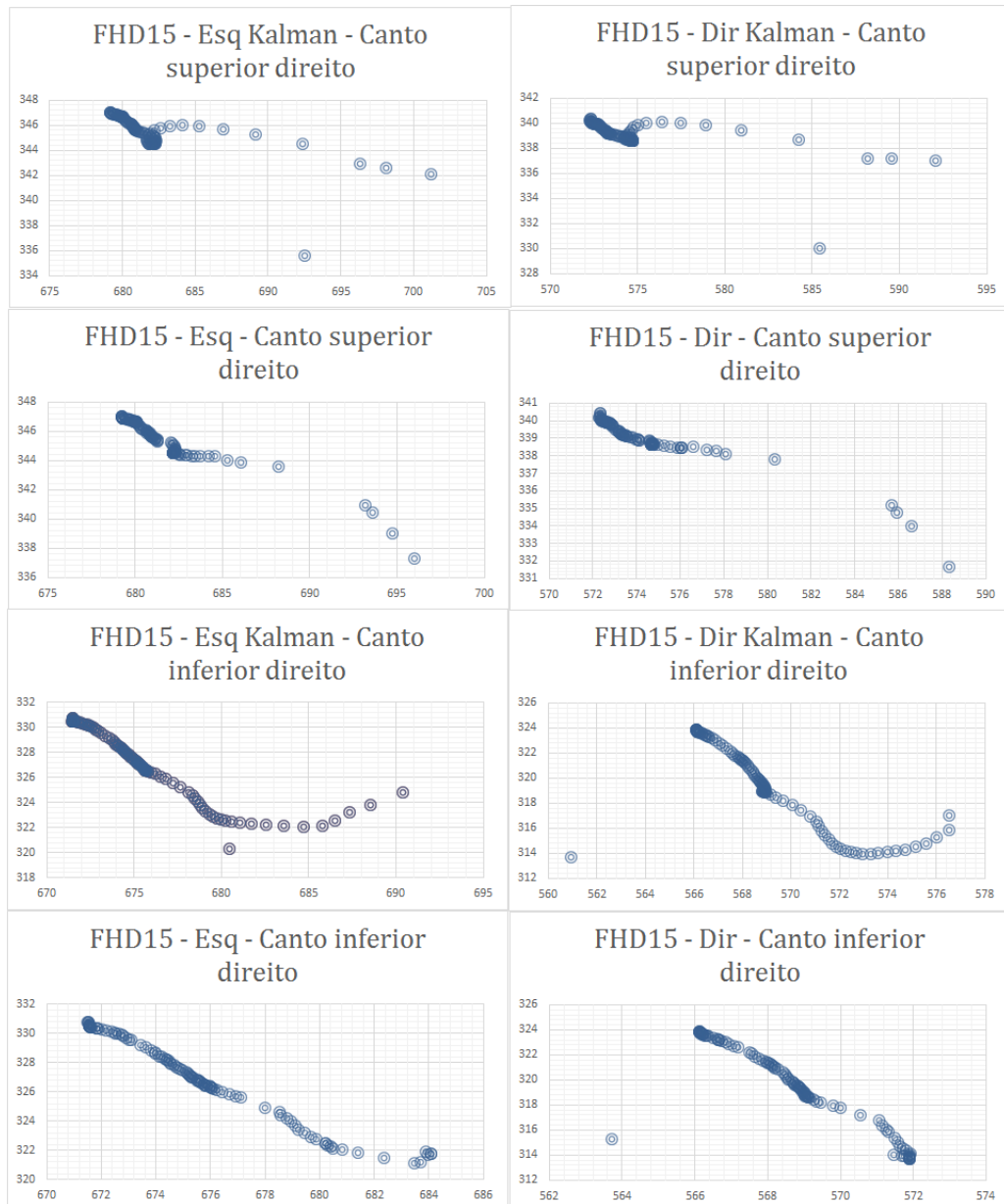


Figura 6.15: Gráfico de dispersão - Ecrã 24" *FullHD* - Olho esquerdo e direito - Olhar para o canto superior e inferior direito.

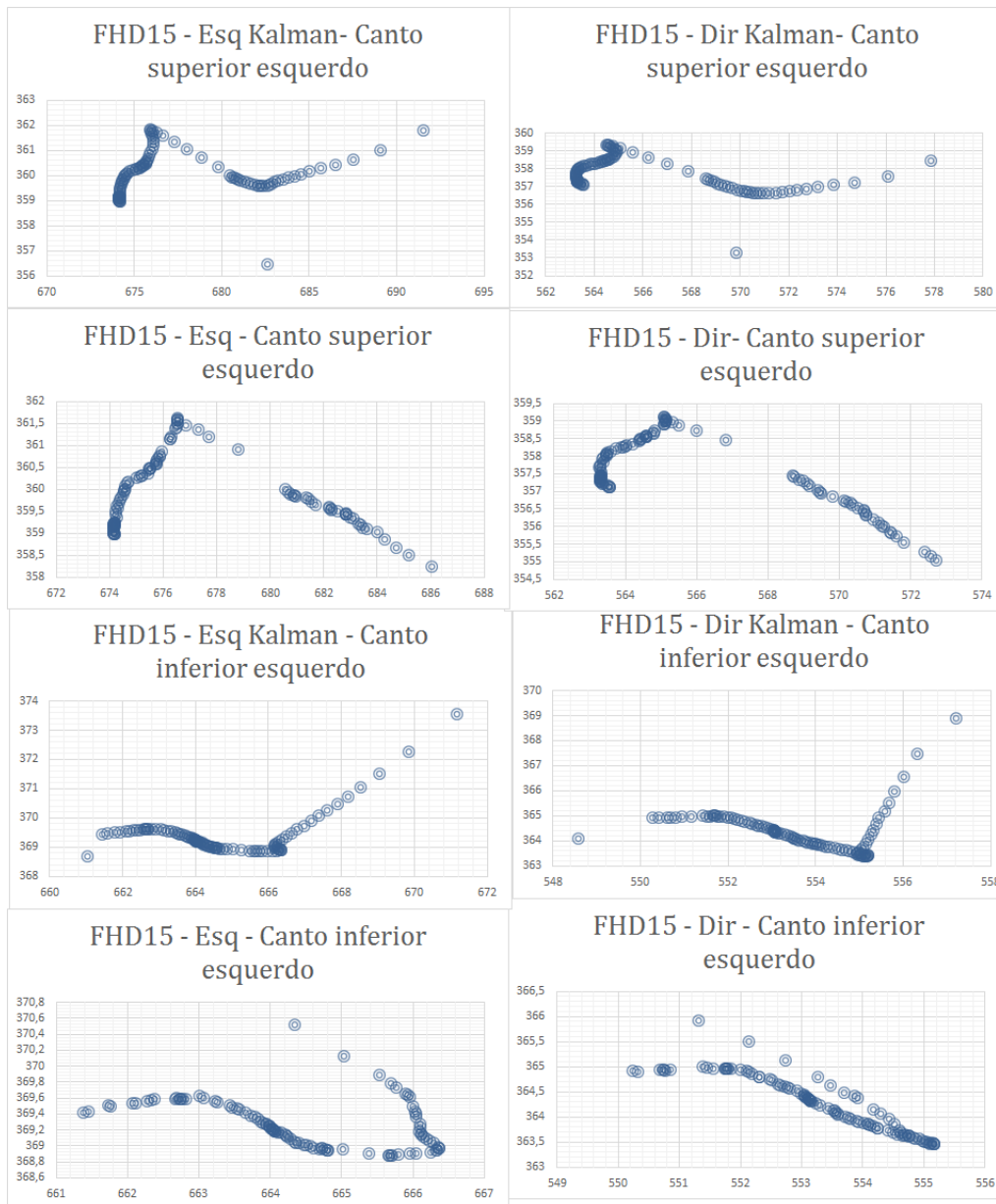


Figura 6.16: Gráfico de dispersão - Ecrã 24" *FullHD* - Olho esquerdo e direito - Olhar para o canto superior e inferior esquerdo.

Os resultados obtidos não variam muito em relação aos do ecrã mais pequeno. No posicionamento do rato, figura 6.17, o erro mantém-se, continuando a ser o centro com melhores resultados.

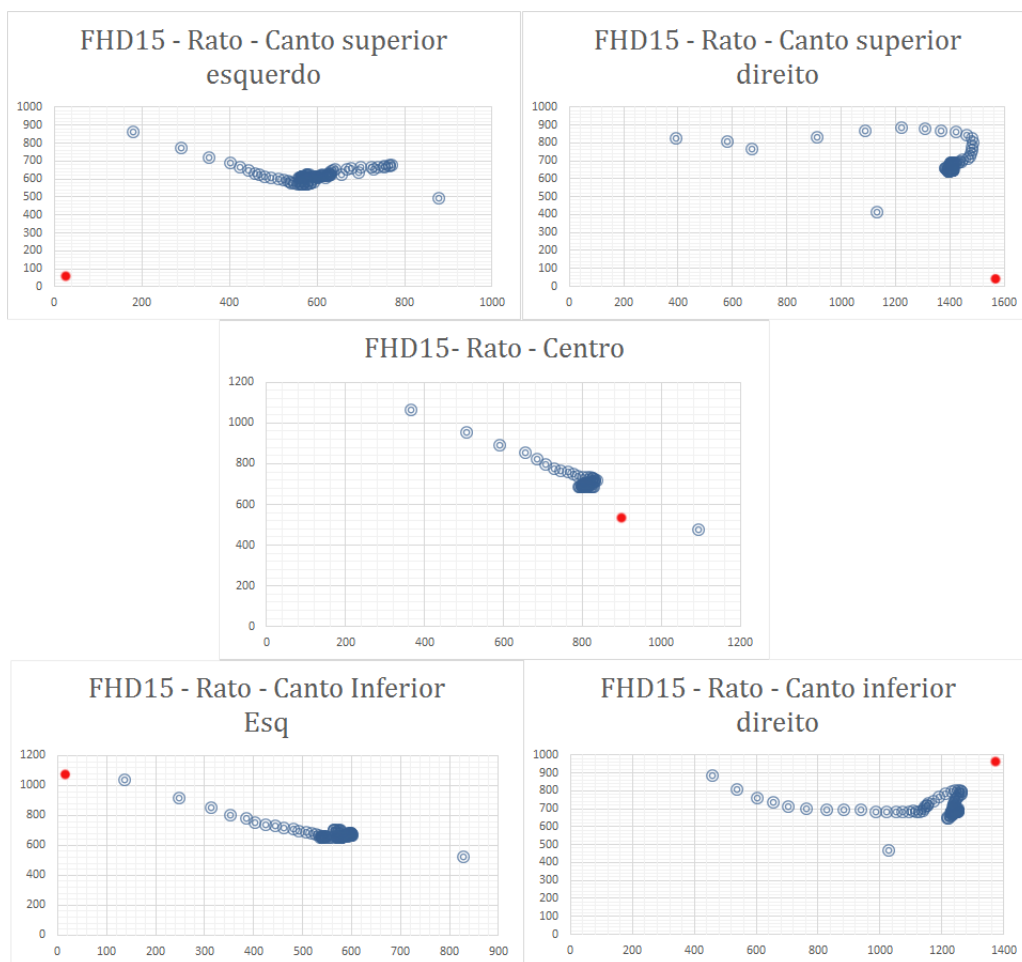


Figura 6.17: Gráfico de dispersão - Ecrã 24" *FullHD* - Rato - Quando o utilizador olha para os vários pontos de teste no ecrã.

Como os resultados obtidos na resolução de  $1920 \times 1080$  não foram os esperados efectuaram-se os mesmos testes para uma resolução mais baixa, de 1280 pixels de largura e 720 de altura.

Os resultados obtidos para o ecrã de quinze polegadas apresentam-se nos gráficos 6.18, 6.19, 6.20 e 6.21.

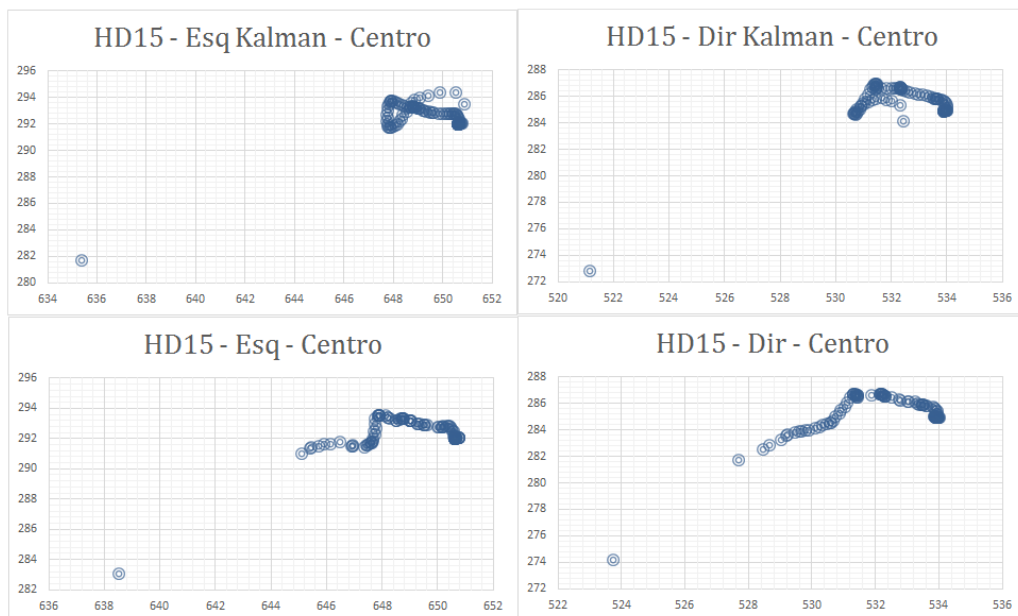


Figura 6.18: Gráfico de dispersão - Ecrã 15" *HD* - Olho esquerdo e direito - Olhar para o centro

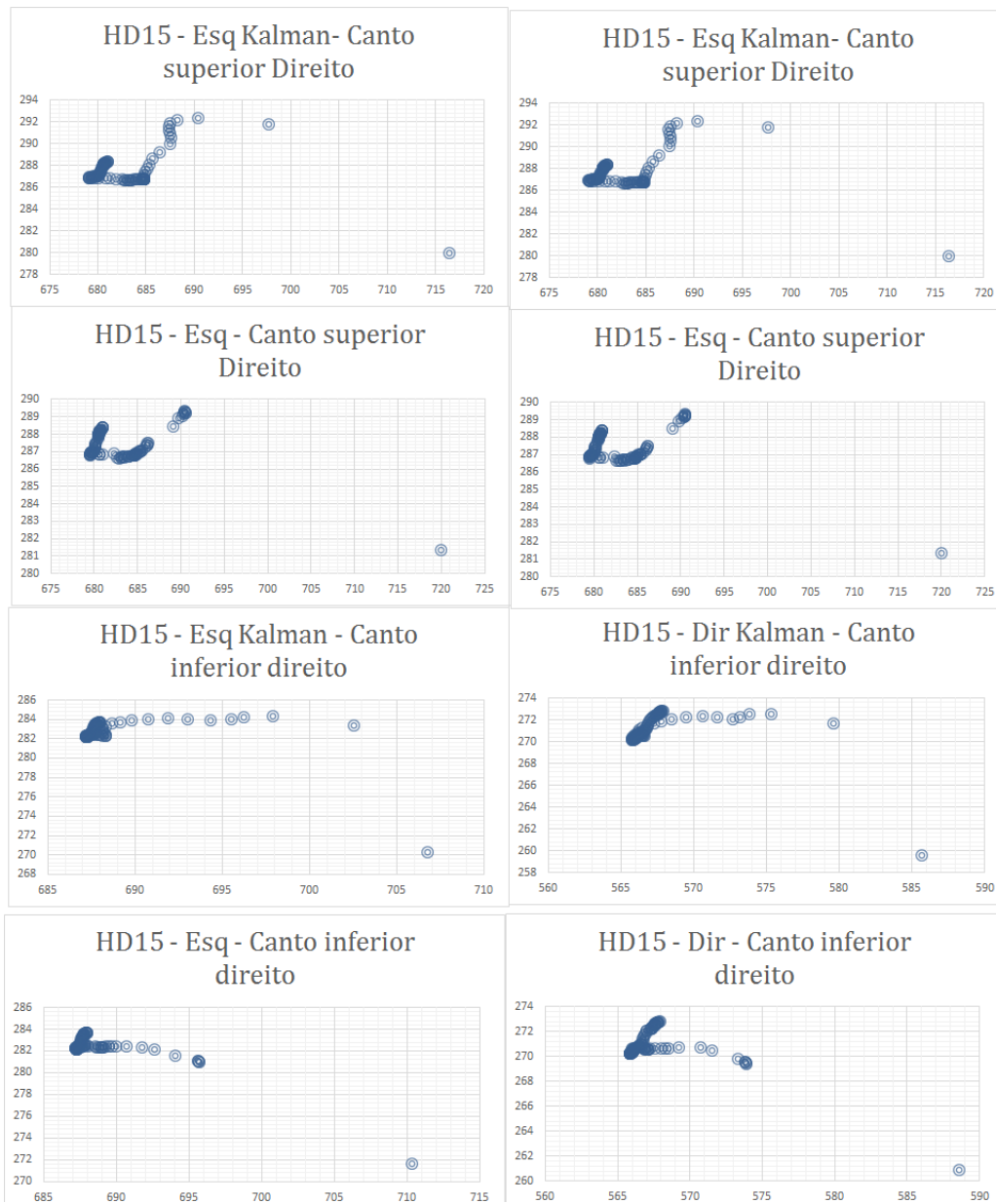


Figura 6.19: Gráfico de dispersão - Ecrã 15" *HD* - Olho esquerdo e direito - Olhar para o canto superior e inferior direito.

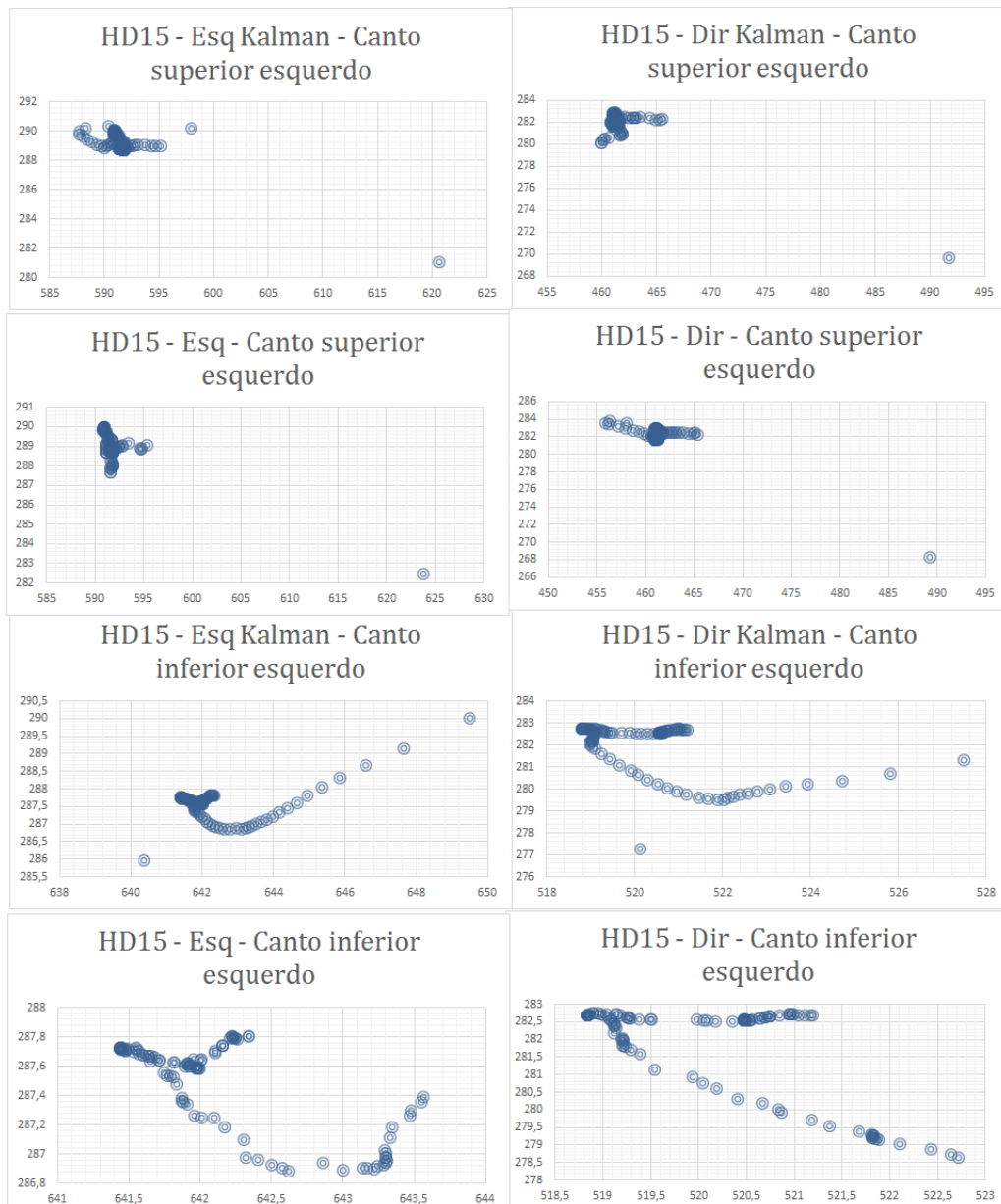


Figura 6.20: Gráfico de dispersão - Ecrã 15" *HD* - Olho esquerdo e direito - Olhar para o canto superior e inferior esquerdo.

Com a resolução mais baixa notam-se diferenças ao nível da dispersão dos valores nos gráficos. O centro continua a obter os melhores resultados para o posicionamento do rato.

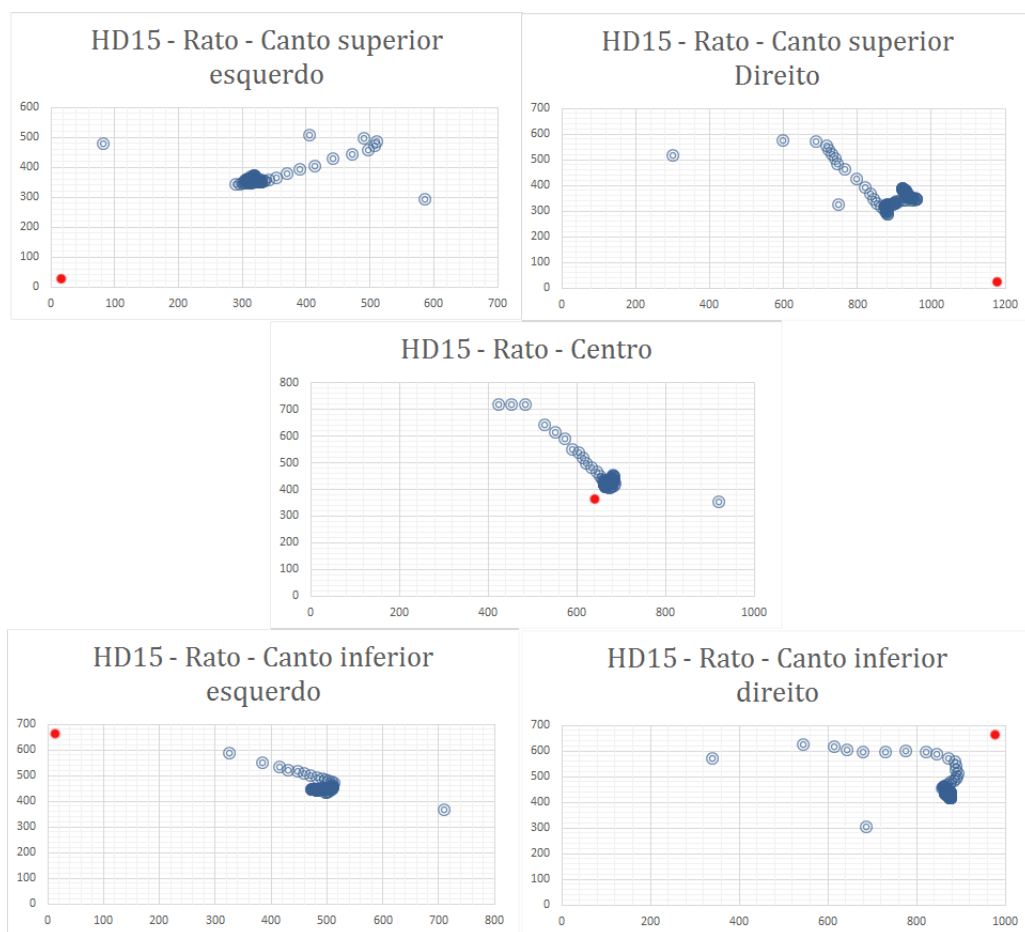
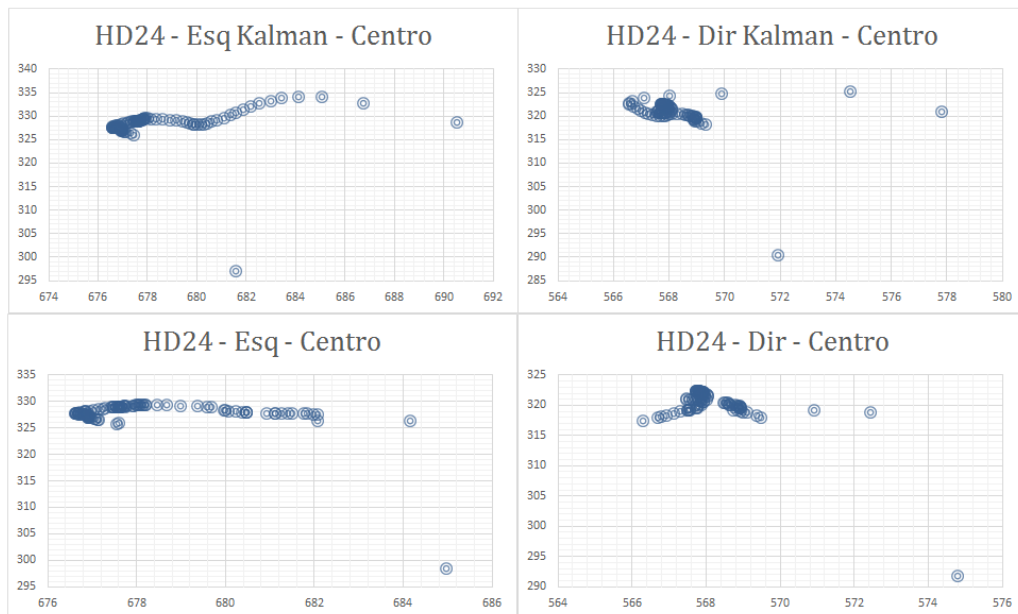


Figura 6.21: Gráfico de dispersão - Ecrã 15" *HD* - Rato - Quando o utilizador olha para os vários pontos de teste no ecrã.

Efectuaram-se os mesmos testes para o ecrã de vinte e quatro polegadas com resolução *hd*. Os valores obtidos são muito semelhantes aos anteriores continuando o centro a ser o que o obtém melhores resultados de posicionamento.



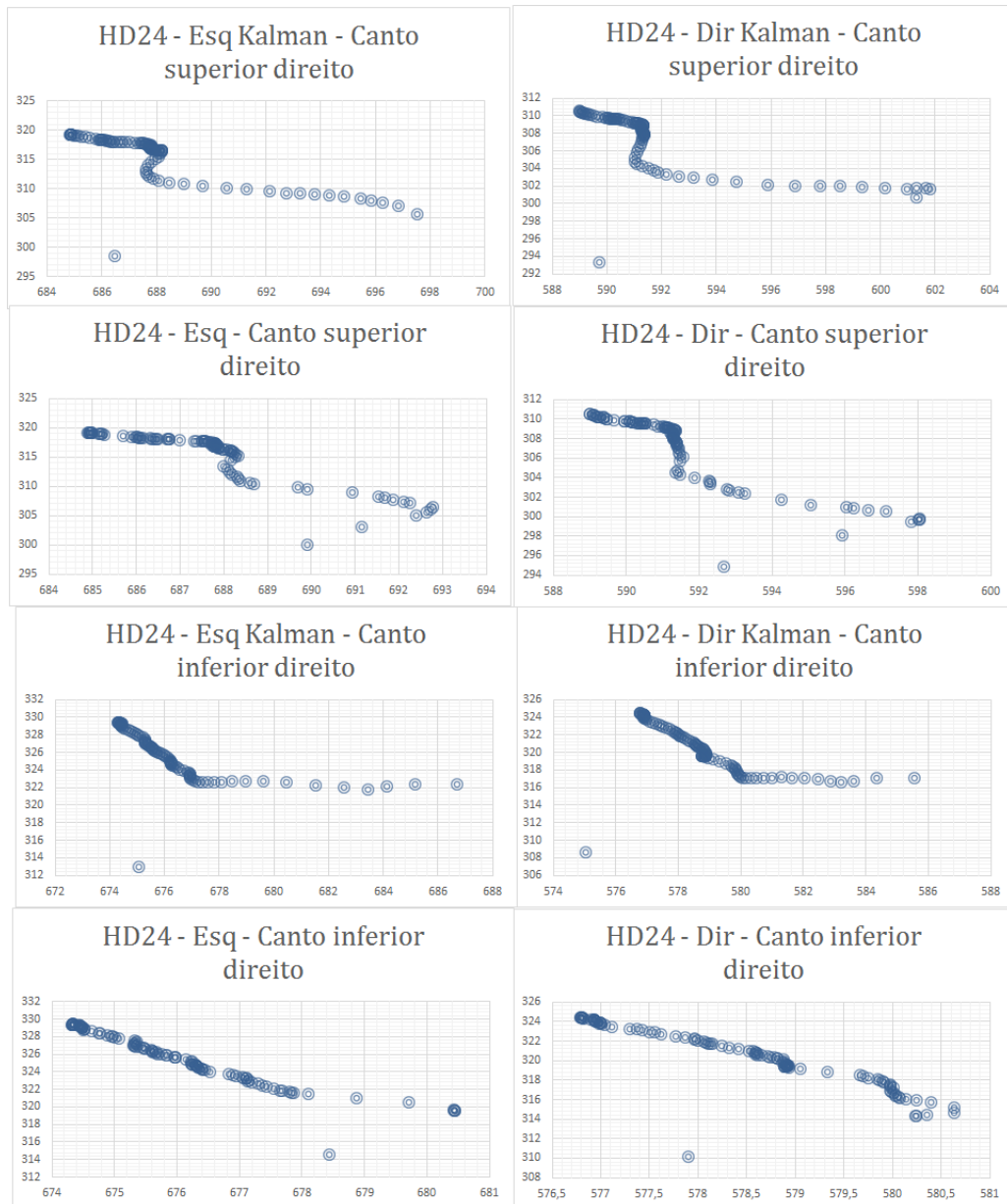


Figura 6.23: Gráfico de dispersão - Ecrã 24" *HD* - Olho esquerdo e direito - Olhar para o canto superior e inferior direito.

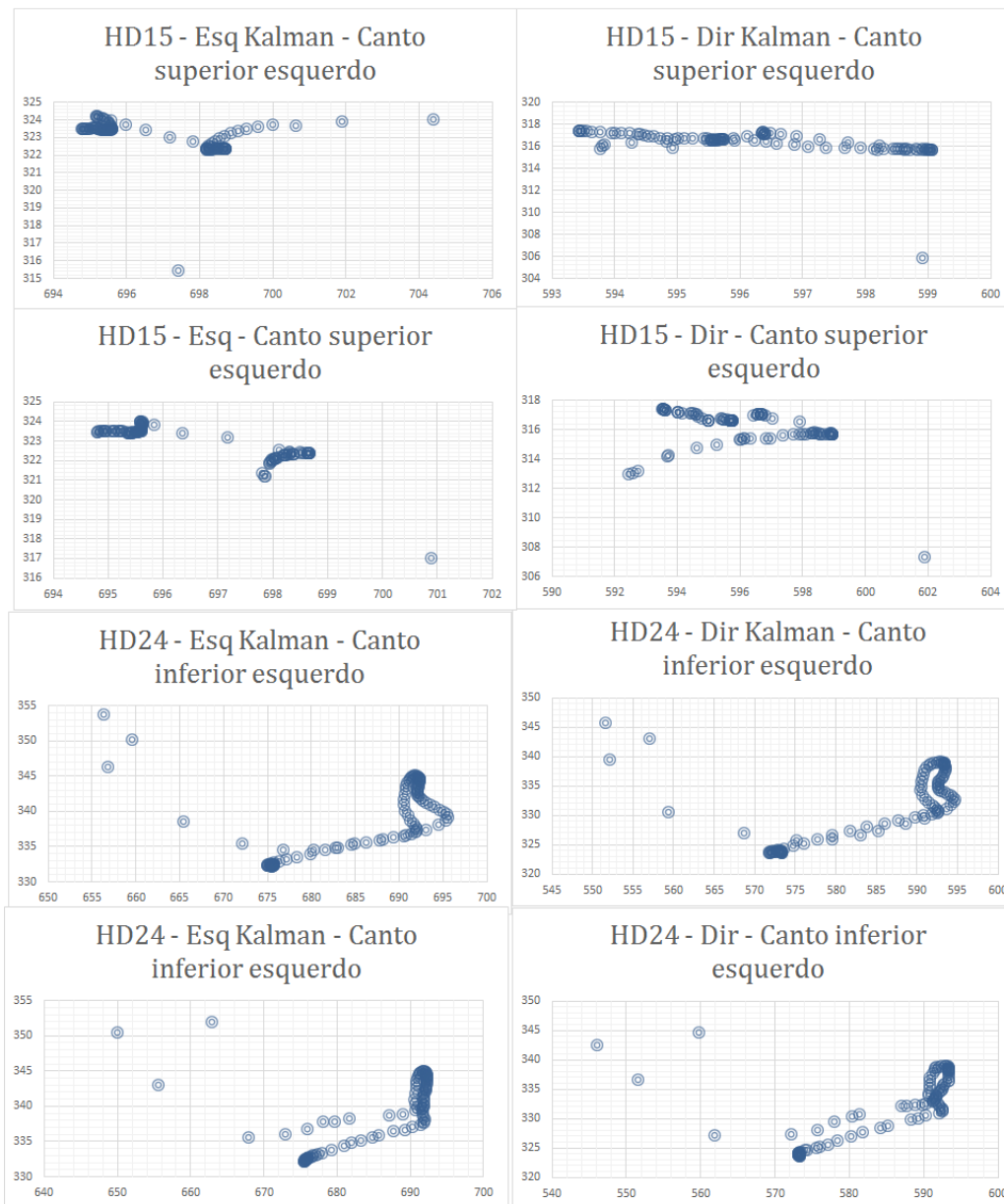


Figura 6.24: Gráfico de dispersão - Ecrã 24" *HD* - Olho esquerdo e direito - Olhar para o canto superior e inferior esquerdo.

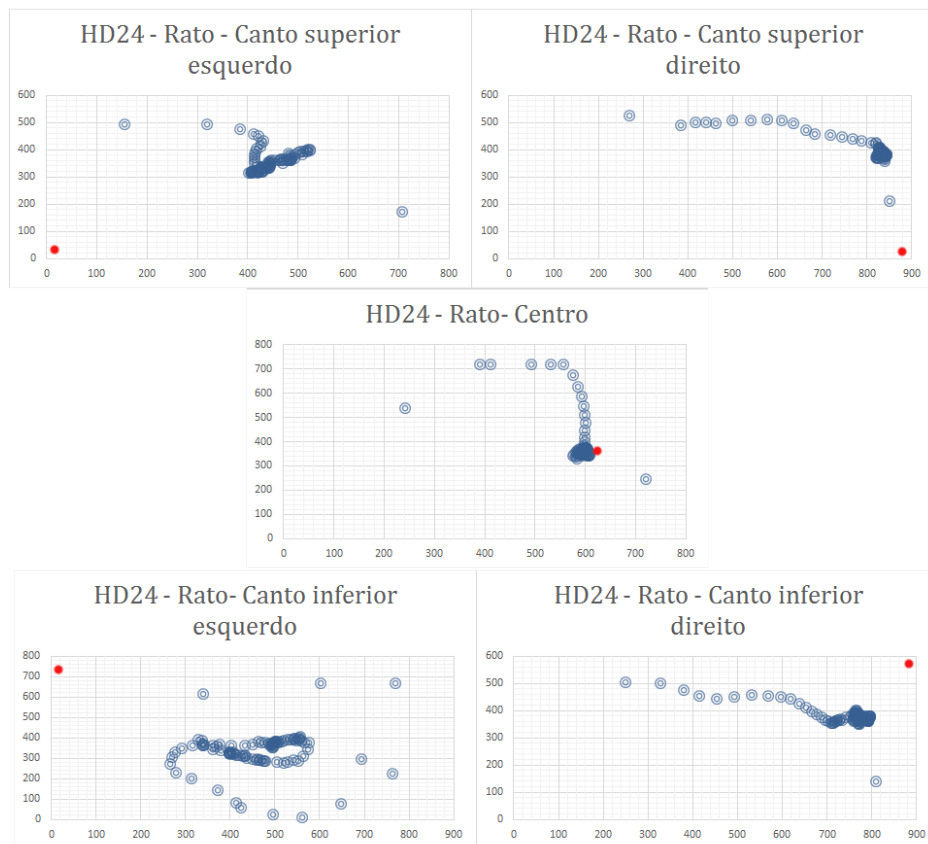


Figura 6.25: Gráfico de dispersão - Ecrã 24" *HD* - Rato - Quando o utilizador olha para os vários pontos de teste no ecrã.

O erro no posicionamento do rato é apresentado na figura 6.26. Para o cálculo do erro médio foram consideradas apenas as amostras entre as quais o foco do olhar no ponto de teste do ecrã se encontra mais estabilizado. O erro médio obtido é elevado, sendo de 408 pixels para  $x$  e de 336 pixels para  $y$ .

		Posição do Rato - Olho Esquerdo e Direito					ERRO TOTAL	
		Pontos de teste do ecrã						
		centro	canto inferior esquerdo	canto inferior direito	canto superior esquerdo	canto superior direito		
E R R O	FHD15	X	154	572	669	599	510	501
		Y	153	402	379	614	641	438
	FHD24	X	130	456	645	462	550	448
		Y	129	294	349	516	515	361
	HD15	X	27	479	399	307	343	311
		Y	62	262	279	344	351	260
	HD24	X	42	454	492	449	436	375
		Y	7	367	339	344	370	285
	ERRO TOTAL	X	88	491	551	454	459	409
		Y	88	331	336	455	469	336

Figura 6.26: Tabela do erro do posicionamento do rato através dos olhos

Nos testes efectuados anteriormente, os resultados obtidos foram os que se pretendiam. No controlo do rato o utilizador apenas consegue movimentá-lo da forma pretendida em relação à direcção horizontal. Verticalmente o controlo torna-se mais difícil. Este problema advém do facto dos músculos do olho *vertical rectus muscles* e *horizontal rectus muscles* terem capacidades de elasticidade de movimento diferentes [27]. Como o nome indica o *vertical rectus muscles* é responsável por ajudar no movimento vertical do globo ocular e a sua amplitude é inferior à exercida pelo *horizontal rectus muscles*, que é o responsável por ajudar no movimento horizontal do globo ocular. Assim, consegue-se discriminar o facto, das variação das posições no eixo do  $x$ , serem maiores e consequentemente ser possível ter maior sensibilidade ao movimento horizontal do rato. Em  $y$  o movimento da pupila é muito mais restrito, o que torna pouco viável a utilização do rato com comodidade nesta direcção.

A realização de acções como se estivesse a utilizar um rato normal, com clique esquerdo e clique direito dos botões, é uma das funcionalidades que muitos dos utilizadores destes sistemas pretendem e que raramente estão presentes nas aplicações comuns existentes. Com este objectivo em vista foram implementados o reconhecimento do piscar de olhos para efectuar o clique esquerdo e o clique direito do rato, com o piscar do olho esquerdo e do olho direito respectivamente, ver figura 6.27 e 6.28.

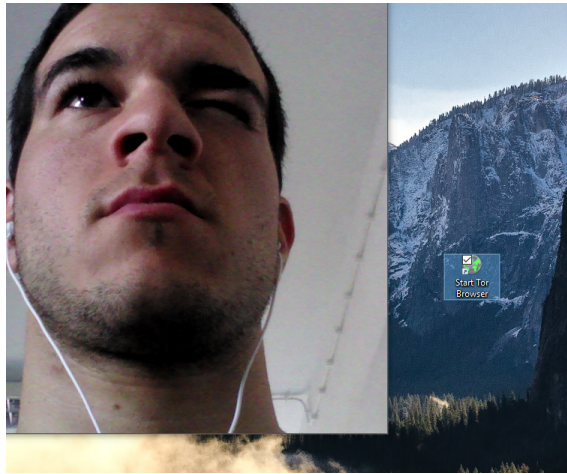


Figura 6.27: Piscar do olho esquerdo - acção de clique do botão esquerdo do rato

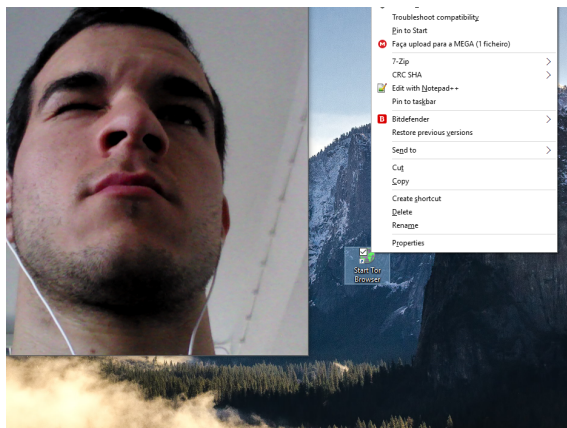


Figura 6.28: Piscar do olho direito- acção de clique do botão direito do rato

O piscar do olho esquerdo e respectivo reconhecimento tem uma taxa de sucesso 90%; enquanto que o piscar do olho direito tem um sucesso de 92%. A taxa de erro inerente a estas acções está relacionada com o facto de a pessoa poder fechar mais ou menos o olho, ou permanecer com o olho fechado mais ou menos tempo. Ainda, outro factor que contribui para que haja um erro médio de 9% é a distância a que a pessoa se encontra da câmara e respectiva variação de distância ao usar o sistema. Os resultados variam consoante a proximidade ou afastamento da câmara. Para se obter a taxa de sucesso anterior, o utilizador tem que permanecer confortável a cerca de 60 centímetros do ecrã. Se este se afastar, a resolução do olho diminui, o que torna o reconhecimento do piscar do olho menos fiável.

## 6.2 Testes e validação de movimento do cursor com o nariz

Como o resultado obtido com os olhos, para o movimento do cursor, não foi o pretendido, implementou-se outra metodologia. A movimentação do cursor pelo movimento do nariz, cuja implementação está descrita na secção 5.1.3. Esta solução também tem a vantagem de se poder continuar a utilizar os olhos, sem alterar a posição do cursor no ecrã.

Com semelhança à detecção da pupila, o ponto de referência do nariz também sofre pequenas oscilações na detecção. De forma a ser possível suavizar é usado um filtro de Kalman, ver gráfico na figura 6.29. Neste gráfico é comparada a detecção do ponto de referência do nariz, com e sem filtragem. Verifica-se que existem diferenças representativas do ruído atenuado quando o filtro de Kalman é utilizado.

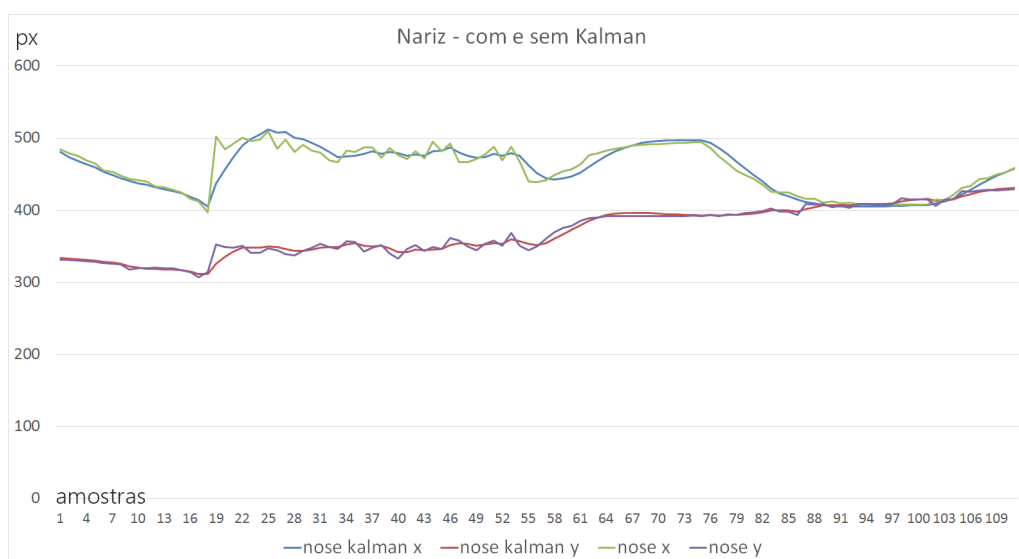


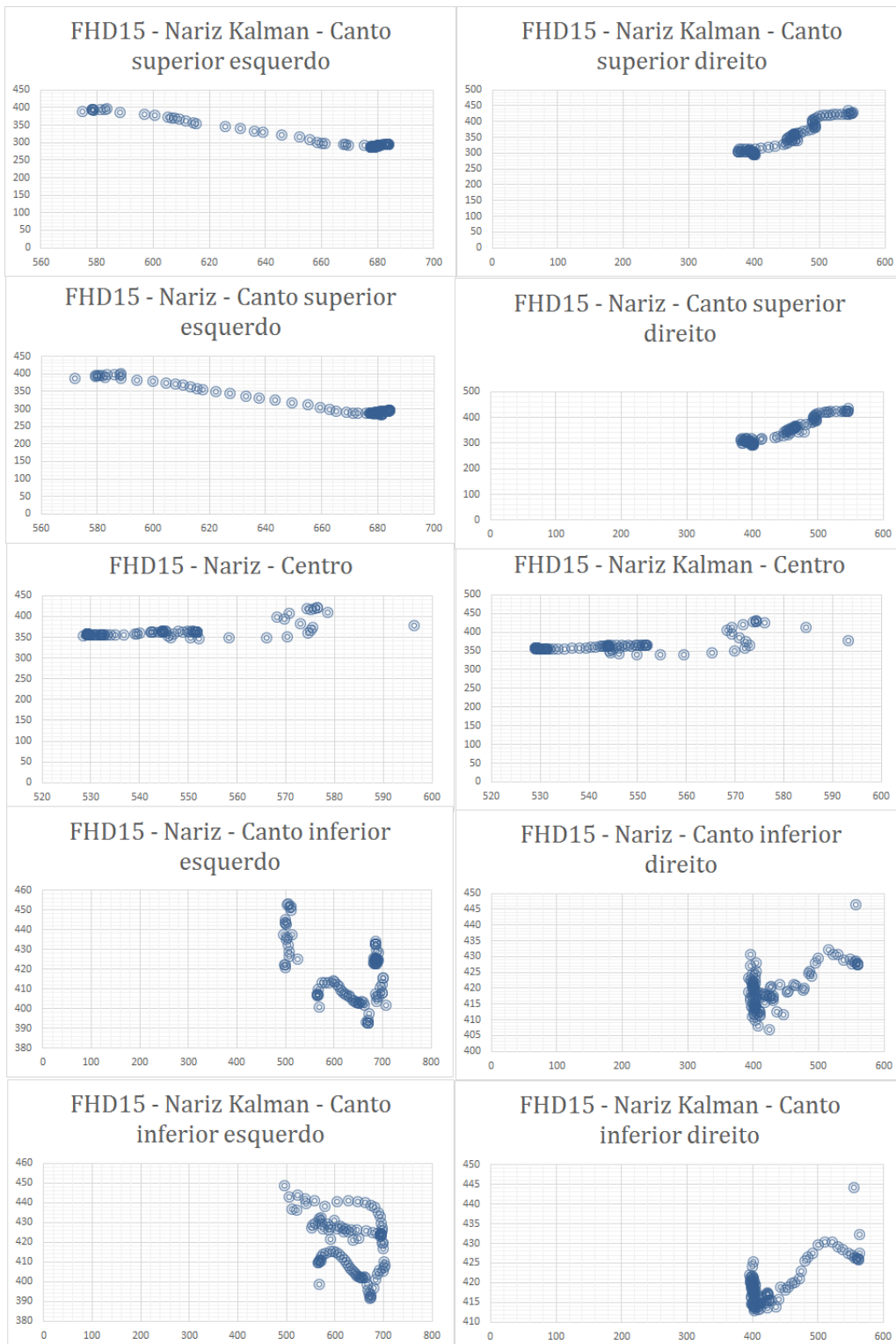
Figura 6.29: Oscilações da detecção do nariz - com e sem filtro de Kalman

Os resultados com filtro de Kalman, permitem um melhor efeito na estabilização da posição do cursor, quando se tem como objectivo a focagem de ícone pequeno.

Para testar a implementação deste método foram efectuados ensaios de usabilidade, semelhantes aos apresentados anteriormente na secção 6.1.

Para estes testes foram usados, igualmente, dois ecrãs com 15 e 24 polegadas, ambos com resolução de 1920 pixeis de largura e 1080 de altura.

Nos gráficos das figuras 6.30 e 6.31 apresentam-se os valores dos pontos de referência do nariz com e sem Kalman, respectivamente, enquanto o utilizador percorria os cinco pontos de teste do ecrã.

Figura 6.30: Gráficos de localização - Ecrã 15" *fullhd* - Nariz

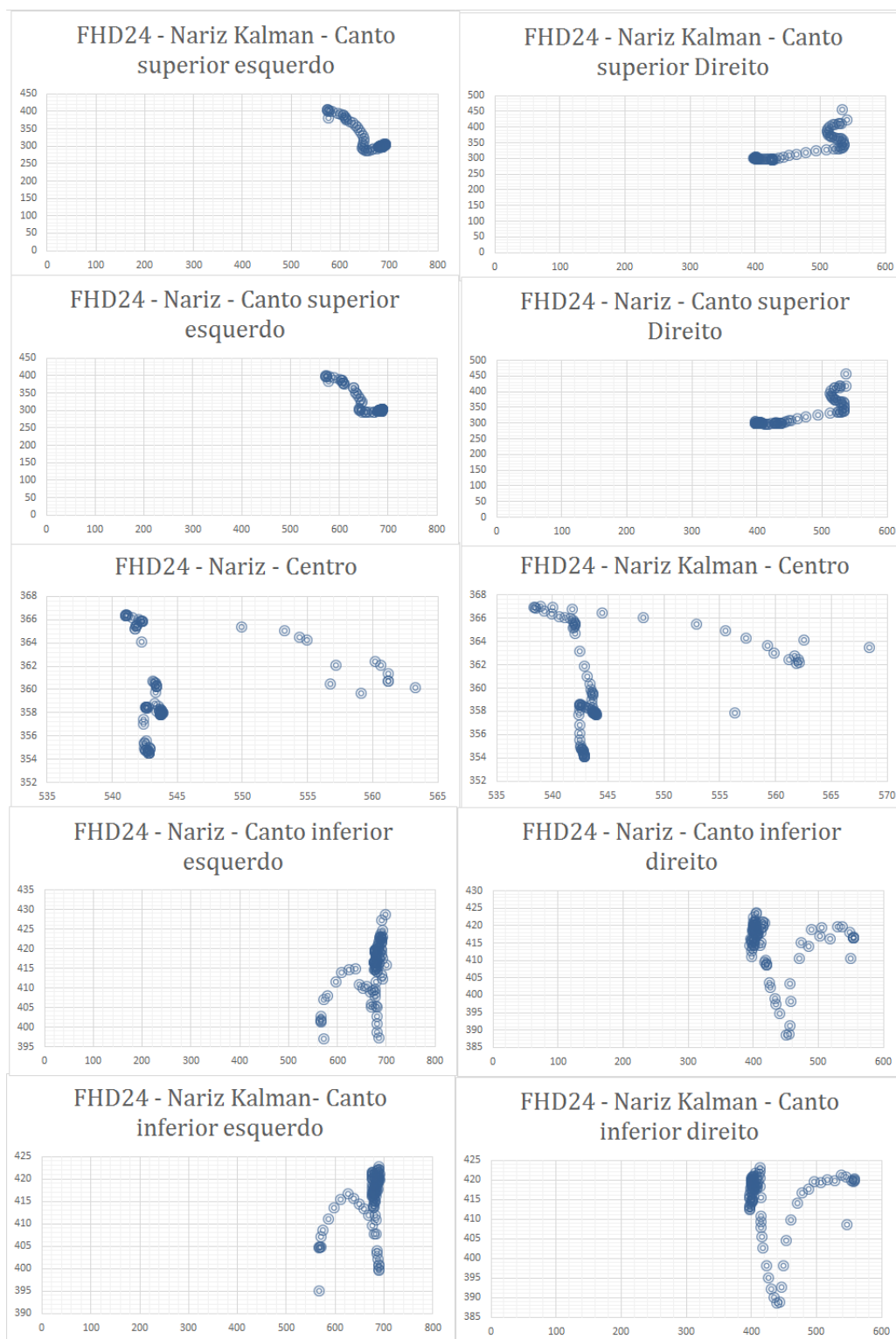


Figura 6.31: Gráficos de localização - Ecrã 24" *fullhd* - Nariz

O rasto que se nota nos gráficos corresponde ao percurso efectuado com o nariz para levar o cursor até ao ponto desejado. Os gráficos do posicionamento do cursor apresentam-se nas figuras 6.32 e 6.33.

Pela análise dos dados obtidos verificamos que o posicionamento do cursor com o nariz tem um erro muito baixo. Algum erro que possa aparecer está relacionado com o ruído inerente à detecção dos pontos de referência do nariz. Como o sistema permite um *feedback* em tempo real o utilizador tem facilidade em se adaptar e mover o cursor para o local desejado.

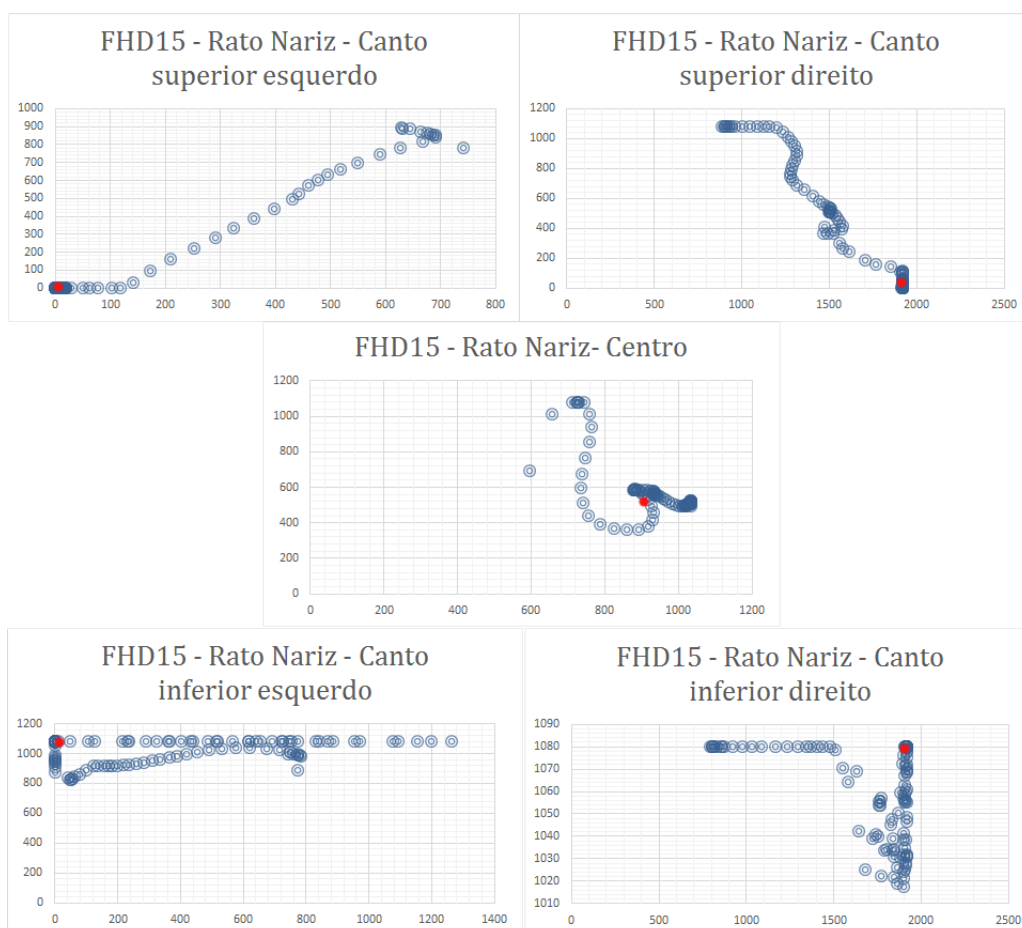


Figura 6.32: Gráfico de Posicionamento do cursor - Ecrã 15" *fullhd* - Nariz

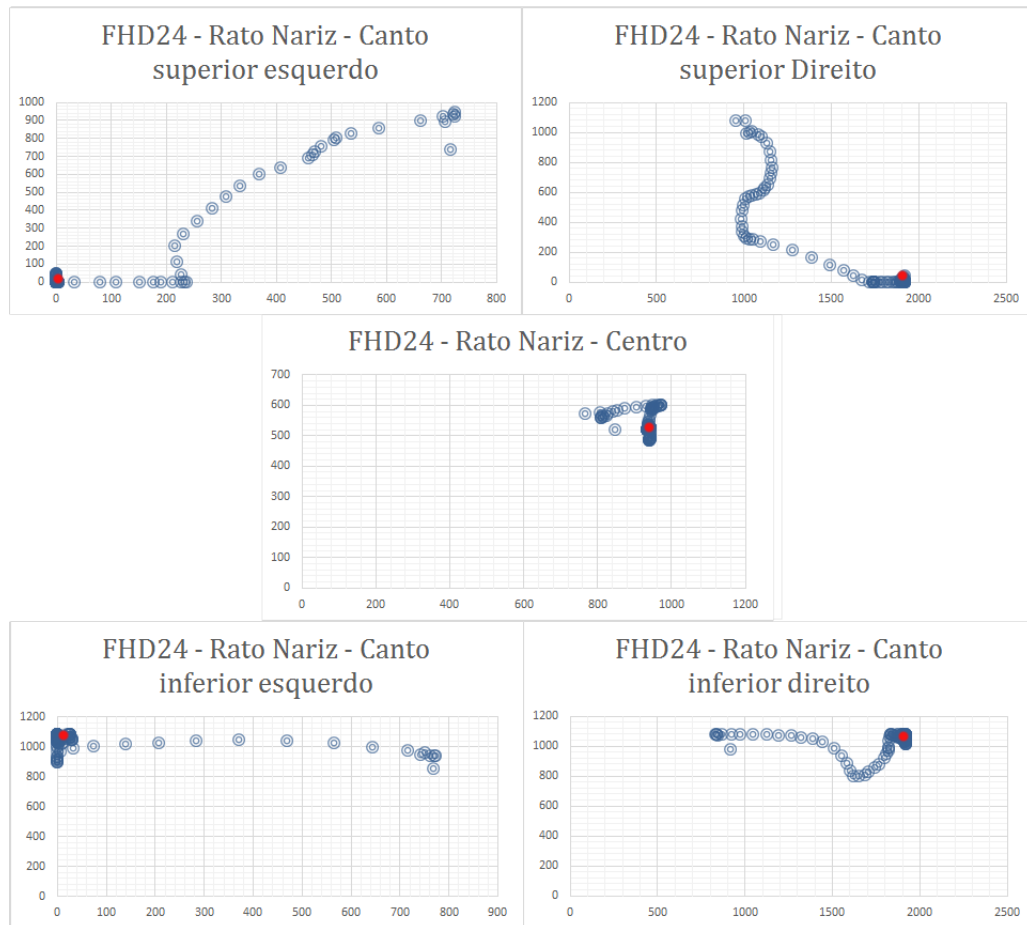


Figura 6.33: Gráfico de Posicionamento do cursor - Ecrã 24" *fullhd* - Nariz

O erro no posicionamento do rato apresenta-se na figura 6.34. Para o cálculo do erro médio seguiu-se as considerações anteriores, tendo sido consideradas um grupo de amostras a partir das quais o cursor já estava perto da posição desejada. O erro médio obtido é baixo, sendo de 44 pixels para  $x$  e de 19 pixels para  $y$ .

		Pontos de teste do ecrã					ERRO TOTAL	
		centro	canto inferior esquerdo	canto inferior direito	canto superior esquerdo	canto superior direito		
E R R O	FHD15	X	54	366	23	3	115	112
		Y	31	21	25	0	140	43
	FHD24	X	22	5	16	0	66	22
		Y	25	16	17	9	9	15
	ERRO TOTAL	X	25	123	13	1	60	45
		Y	19	12	14	3	50	20

Figura 6.34: Tabela erro do posicionamento do rato através do nariz

Embora o erro seja baixo não revela a realidade deste método, que se revelou com boa precisão para o utilizador comum, sem precisar de calibração e com pouco tempo de habituação.

É possível ao utilizador realizar todo o tipo de movimento com o cursor do rato em todo o ecrã. Nas figuras 6.35 e 6.36 ilustra-se a utilização deste método para movimentar o cursor pela ambiente de trabalho e abrir janelas do sistema operativo Windows 10.

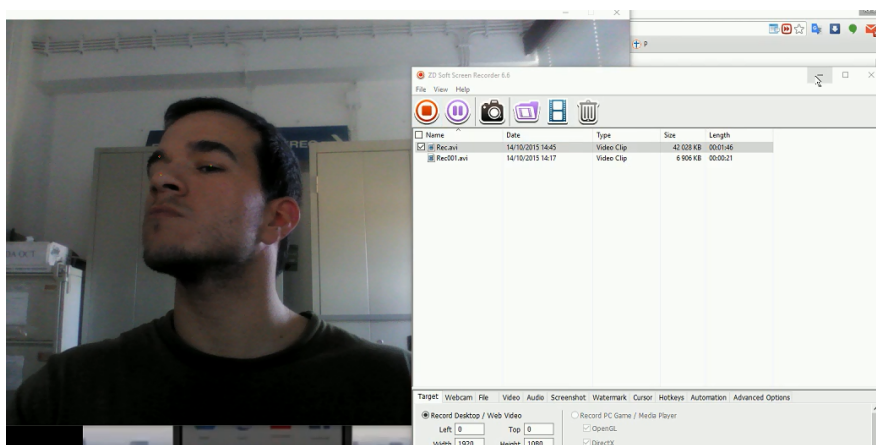


Figura 6.35: Movimento do rato com o nariz - botão de minimizar janela

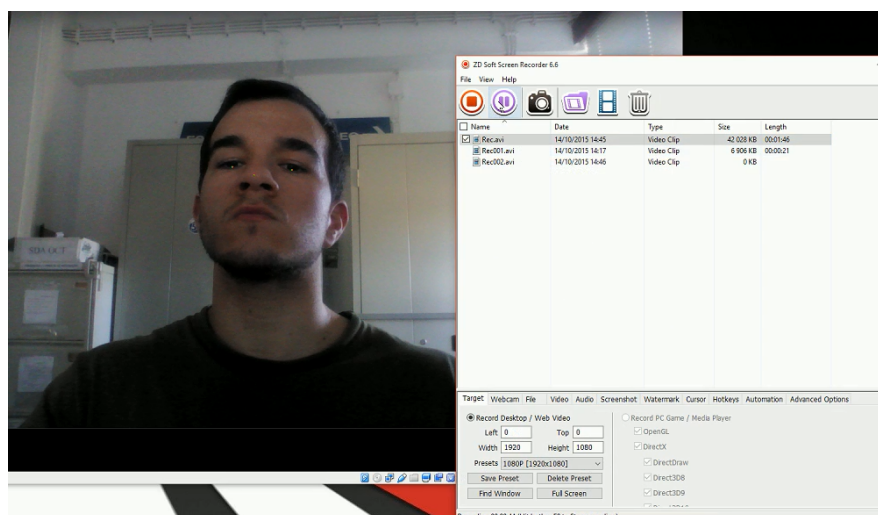


Figura 6.36: Movimento do rato com o nariz - botão de pausar o vídeo

Para realizar os cliques ao utilizar o nariz não é possível utilizar o piscar de olhos, devido à utilização no nariz obrigar a movimentação da cabeça. Na figura 6.37 podemos verificar que, ao rodar a cabeça, os olhos aparentam estar fechados, mas na realidade não estão. Esta discrepância invalida a utilização da acção piscar de olhos para controlar os botões do rato. Se analisarmos a imagem percebemos que um dos olhos está ocultado, novamente, limita a utilização do olho para efectuar alguma acção.

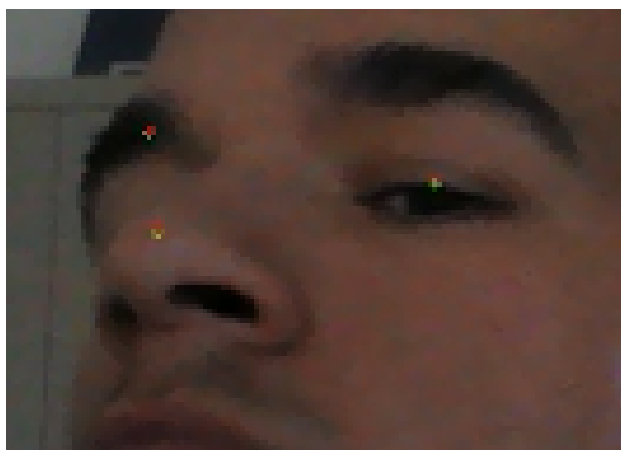


Figura 6.37: Movimento do rato com o nariz, impossibilita a utilização do piscar de olhos para acções do sistema

Por outro lado, temos um elemento na face que possibilita uma interacção com o sistema de forma eficaz, a boca. Com a impossibilidade da utilização

do piscar de olhos, foi implementada a acção de clique com o abrir e fechar da boca, descrita no capítulo anterior, na secção 5.1.3.

Assim, com a abertura e fecho da boca simulou-se duas acções:

- manter o botão esquerdo premido - abrir a boca;
- libertação do botão premido - fechar boca.

As duas acções anteriores, quando feitas sequencialmente representam um clique esquerdo do rato. Mas, enquanto a primeira acções está a decorrer o cursor pode ser utilizado para outras tarefas, como por exemplo, seleccionar texto. Estas acções podem ser verificadas na figura 6.38.

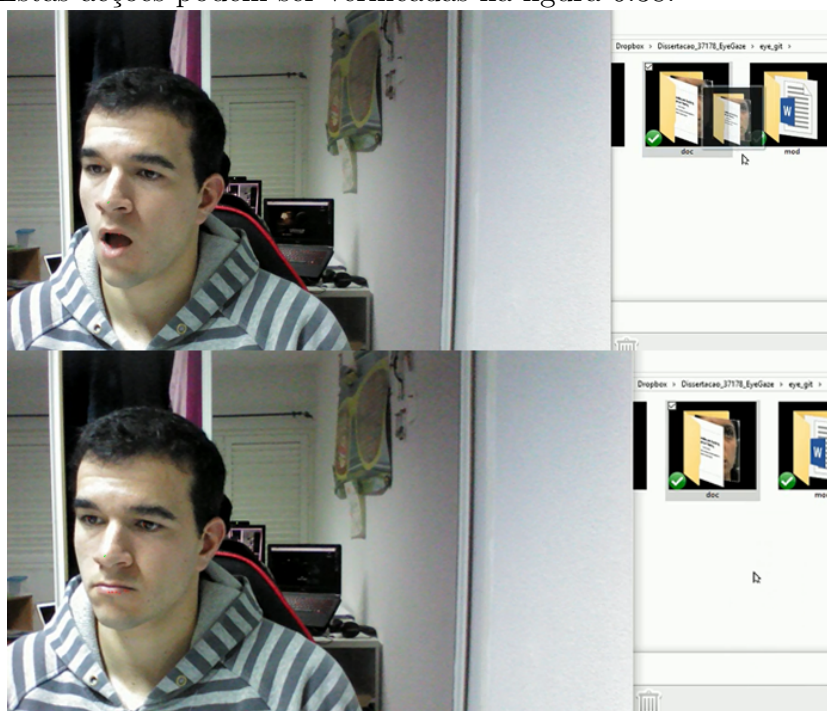


Figura 6.38: Clique esquerdo do rato com abertura e fecho da boca - simular um arrastar de ficheiro

Estas acções tem uma eficácia semelhante à dos piscar do olhos, tendo-se obtido uma taxa de sucesso de 95%. Contudo, esta eficácia varia, igualmente, com a distância e luminosidade da câmara. A implementação do clique do botão direito foi substituída pela interface de utilizador do Windows 8 ou superior, visto que este está preparado para dispositivos de toque e disponibiliza todas as opções na interface do explorador.



# Capítulo 7

## Conclusões

Nesta dissertação foi implementada uma interface para substituir as acções do rato utilizando os movimentos dos olhos, do nariz e da boca. É uma tarefa ambiciosa e complexa mas prova-se que é passível de ser concretizada.

Durante a realização deste projecto, foram estudadas as necessidades específicas dos utilizadores com necessidades especiais, nomeadamente, foram realizadas reuniões com especialistas desta área. Este estudo, revelou que um desejo, unânime, destes utilizadores, prende-se com a utilização comum do rato e teclado, visto ser de utilização comum pelos utilizadores. Assim, neste projecto tentou-se substituir a utilização do rato por um sistema alternativo. Centrado para o desenvolvimento de uma nova interface pessoa-máquina que utiliza os olhos, o nariz ou a boca. Dever-se-á estudar formas alternativas que não se prendam à utilização do rato e teclado, com a intenção para se desenvolver interfaces mais naturais.

Outra questão prende-se com a utilização de hardware dedicado. Um dos objectivos deste projecto prende-se com a utilização de hardware genérico, sem necessidade de aquisição de novo equipamento. Para a implementação deste sistema foi utilizada a câmara Intel Real Sense, cujo a Intel quer tornar padrão em novos dispositivos, substituindo as actuais câmaras web.

A iluminação é um dos factores relevantes para a qualidade do sistema. Embora a câmara da Intel tenha vários sensores o RGB é o mais utilizado para as operações concretizadas e, devido à necessidade de realizar compensação do RGB com infravermelhos, para melhorar a imagem, torna a ilu-

minação um factor importante. Outro aspecto fundamental é a distância do utilizador à câmara. Quanto mais longe o utilizador se encontrar da câmara menos resolução terá o olho, e assim, menos sensibilidade se consegue atribuir à acção de movimentar o rato. Por fim, um factor que relaciona os dois anteriores é a qualidade da câmara, quanto melhor a qualidade do sensor *RGB*, mais resolução e melhor adaptação luz, mais informação se consegue extrair das imagens.

Com a maturidade da tecnologia, este sistema poderá ser melhorado, pois, possivelmente, ter-se-à acesso às imagens provenientes do sensor de infravermelhos e assim utilizar o brilho da pupila proveniente do reflexo da luz infravermelha [6]. Na figura 7.1 mostra-se um exemplo de utilização de um sensor de infravermelhos.



Figura 7.1: Reflexo da pupila sobre luz infravermelha

O primeiro objectivo desta dissertação é obter um sistema de seguimento ocular fidedigno. Embora a precisão não tenha sido a mais desejada, este foi atingido e conseguiu-se um sistema a operar em tempo real e com velocidade suficiente para utilização diária.

A implementação e realização de acções foi concluída embora não com o sucesso desejado. Era pretendido o controlo do cursor com o movimento ocular como se tratasse de um rato vulgar, o qual não foi possível através do seguimento ocular.

A segunda parte deste objectivo, implementação de acções clique do rato, foi atingida com sucesso. O utilizador consegue aplicar acções comuns de um rato com o piscar dos olhos com precisão suficiente para a utilização diária. Foram implementadas acções básicas, como o clique dos botões do rato, por necessidade do público alvo deste dissertação, pessoas com deficiências motoras.

Com o intuito de obter melhores resultados foi criado um novo objectivo, de movimentar o cursor a partir do movimento do nariz e o clique do rato com a abertura e fecho da boca.

Este método foi o que obteve melhores resultados e se mostrou mais promissor. Esta metodologia é usável pelo utilizador comum sem qualquer tipo de calibração, basta que este assuma uma postura de trabalho correcta e que não faça muitas oscilações. A utilização do nariz requer pouco tempo de experiência e esforço por parte do utilizador.

De igual forma, a acção de clique implementada pelo abrir e fechar da boca mostrou-se simples de utilizar, eficaz e precisa.

Para trabalho futuro existe um aspecto que deve ser levado em conta. O trabalho deverá ser mais focado no algoritmo que permite fazer transposição do sistema de coordenadas do olho para o sistema de coordenadas do ecrã com precisão, de forma a viabilizar todo o sistema.

Considera-se que é possível implementar uma solução comercial utilizando o nariz e boca para simularem as acções realizadas com o rato. Além disso, poder-se-á utilizar o foco do olhar para realizar outras operações que envolvam menos precisão ou que auxiliam outro sistema.



# Bibliografia

- [1] Canny, J. (1986). A computational approach to edge detection. *IEEE transactions on pattern analysis and machine intelligence*, PAMI - 8(6):679–698.
- [2] Duchowski, A. T. (2003). *Eye tracking methodology: Theory and practice*. Springer-Verlag Ltd, London.
- [3] Jacob, R. J. K. e Karn, K. S. (2003). *Eye tracking in Human-Computer Interaction and usability research: Ready to deliver the promises*, cap.: The mind’s eye: Cognitive and applied aspects of eye movement research, p. 573–605. Elsevier, Amsterdam.
- [4] Jaimes, A. e Sebe, N. (2007). Multimodal humancomputer interaction: A survey. *Computer Vision and Image Understanding*, 1-2(108):116–134.
- [5] Kaur, M., Tremaine, M., Huang, N., Wilder, J., Gacovski, Z., Flippo, F., e Mantravadi, C. S. (2003). Where is “it”? event synchronization in gaze-speech input systems. In *Proceedings of the Fifth International Conference on Multimodal Interfaces (ICCSE)*, p. 151–158, NY. ACM Press.
- [6] Poole, A. e Ball, L. J. (2005). *Eye Tracking in HumanComputer Interaction and Usability Research: Current Status and Future Prospects*. Idea Group, Inc, Pennsylvania.
- [7] Rayner, K. e Pollatsek, A. (1989). *The psychology of reading*. Prentice Hall, NJ.
- [8] Tunhua, B. B. W., Changle, L. S. Z., e Kunhui, L. (2010). Real-time non-intrusive eye tracking for human-computer interaction. In *Proceedings of*

- the 5th International Conference on Computer Science and Education (ICCSE)*, p. 1092–1096. IEEE.
- [9] Welch, G. e Bishop, G. (2006). *An Introduction to the Kalman Filter*.
- [10] Wild, D. J. (2012). *Gaze tracking using a regular web camera*. Technical report, Department of Computer Science, Rhodes University, Grahamstown, South Africa.
- [11] Burguer, P., Rothbucher, M. e Diepold, K. (2014) *Self-Initializing Head Pose Estimation With 2D Monocular USB Camera*, Technical Report. Institute for Data Processing.
- [12] DeMethon, D. F. e Davis, L. S. University of Maryland, Computer Vision Laboratory. *Model-Based Object Pose in 25 Lines of Code*
- [13] Morimoto, C. H. e Mimica, M. R. (2004) *Eye Gaze tracking techniques for interactive applications*. Elsevier Inc.
- [14] Fanelli, G., Weise, T., Gall, J. e Van Gool, L. ETH Zurich, Suíça. EPFL Lausanne, Suíça. KU Leuven, Bélgica. *Real Time Head Pose Estimation from Consumer Depth Cameras*
- [15] Ji, Q. (2002) Department of Electrical, Computer, and System Engineering. Rensselaer Polytechnic Institute, Troy. USA. *3D Face pose estimation and tracking from a monocular camera*. 0262-8856/02/S Elsevier.
- [16] Moeslund, T. B. e Granum, E. (2000) Laboratory of Computer Vision and Media Technology, Aalborg University. *A Survey of Computer Vision-Based Human Motion Capture*. Academic Press.
- [17] Moeslund, T. B., Hilton, A. e Kruger, V. (2006) Laboratory of Computer Vision and Media Technology, Aalborg University. *A survey of advances in vision-based human motion capture and analysis*. 1077-3142/S Elsevier.
- [18] Drewes, H. (2010) *Eye Gaze Tracking for Human Computer Interaction*, Dissertation an der LFE Medien-Informatik der Ludwig-Maximilians-Universität München.

- 
- [19] Viola, P. e Jones, M. (2001) *Rapid Object Detection using a Boosted Cascade of Simple Features*, Conference on Computer Vision and Pattern Recognition.
- [20] Pinho, R. R., Tavares, J. M. R. S., Correia, M. F. P. V. (2005) *Filtro de Kalman no Seguimento de Movimento em Visão Computacional*, Universidade do Porto, Faculdade de Engenharia.
- [21] Schnipke, S. K. e Todd, M.W. (2000) *Trials and tribulations of using an eye-tracking system*. Proc. ACM SIGCHI - Human Factors in Computing Systems Conference, 2000, pp. 273-274
- [22] Xiong, X., Cai, Q., Liu, Z., Zhang Z. (2014) *Eye Gaze Tracking Using an RGBD Camera: A Comparison with a RGB Solution*, UBICOMP '14 ADJUNCT. Seattle, USA.
- [23] Santos, R., Santos, N., Jorge, P. e Abrantes, A. (2013) *Eye Tracking System With Common Webcam*. Lisboa, ISEL. Elsevier.
- [24] Antolovic, D. *Review of the Hough Transform Method, With an Implementation of the Fast Hough Variant for Line Detection*. Department of Computer Science, Indiana University and IBM Corporation.
- [25] Le Roux, J. (2003) *An introduction to Kalman Filtering: Probabilistic and Deterministic Approaches*. University of Nice.
- [26] Euler L. (1775) translated by Sten J. *General formulas for any translation of rigid bodies*. Imperial Academy of Sciences Petersburg.
- [27] Noorden, Gunter K. von e Campos, Emilio C. (1928) *Binocular Vision and Ocular Motility*. Mosby, Inc. St. Louis, Missouri.
- [28] Papageorgiou, C. P., Oren, M., Poggio T. (1998) *A general framework for object detection*. In International Conference of Computer Vision. Cambridge, MA.
- [29] Freund, Y. e Schapire, R. (1996) *A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting* AT&T Labs, New Jersey.
- [30] Lacey, T. *The Kalman Filter* MIT Literature.