**Visão por Computador – Eye Tracking e Processamento de Imagem Térmica**

**Universidade de Aveiro**

José Ferrão Nº63944

Ricardo Ribeiro Nº64694

I. INTRODUÇÃO

O objetivo do nosso projeto é a implementação de um sistema de reconhecimento fácil e imagem térmica usando um raspberrypi.

Sendo o objetivo final correr o projeto com um raspberrypi 3 muitas das decisões terão que ser tomadas com base no desempenho dos diversos algoritmos.

O projeto será usado como base ao desenvolvimento de uma dissertação que tem como objetivo a criação de um sistema a ser instalado numa mesa (numa sala de aula por exemplo) capaz de monitorizar em tempo real a pessoa sentada nessa mesma mesa, fornecendo informação acerca da direção do olhar, expressões faciais, temperatura facial, etc.

Ao longo do projeto foi usado OpenCV 3.2 tendo sido o desenvolvimento do trabalho feito maioritariamente em computadores (x86) e testado posteriormente no raspberrypi de forma a medir desempenho das diversas soluções implementadas e ter acesso á camera térmica.

II. OBJETIVOS

* Reconhecimento facial
* Face tracking
* Eye tracking
* Reconhecimento de expressões faciais
* Obtenção de imagem térmica
* Image matching
* Construção de imagem térmica de resolução alta recorrendo a informação de contornos da imagem a cores.

III.RECONHECIMENTO FACIAL

Para reconhecimento foram usadas haars-cascades em conjunto com alguma informação de posicionamento relativo dos elementos de uma face.

Assumimos sempre que os olhos surgem na parte superior da face 60% da parte superior, e que o nariz aparece abaixo dos olhos nos 60% inferiores da face, desta forma poupamos algum tempo no processamento da imagem e evitamos alguns falsos positivos na deteção destes elementos.

# 

Figura 1 – Separação parte Superior e inferior da face

Permitimos ainda no máximo 1 nariz e 2 olhos por face detetada.

A imagem abaixo representa o resultado obtido após esta fase de implementação.



Figura 2. Resultado de Face Detection

Ao longo desta fase fomos experimentando várias cascatas já treinadas disponíveis na internet, tendo optado por 2 cascatas disponíveis nos exemplos de openCV para deteção de olhos e face (construída com imagem de pessoas com e sem óculos) e uma outra externa para deteção do nariz.

IV. EYE TRACKING

O nosso objetivo para esta etapa de eye tracking é obter um vetor com a direção do olhar da pessoa.

Começamos com imagens da zona obtida pela cascata de deteção de olhos, para identificar o centro da pupila começamos por experimentar deteção de blobs a partir da zona central da imagem, com algum sucesso, mas muitas vezes com centro da pupila muito afastado da realidade.

Experimentamos de seguida um algoritmo desenvolvido por Fabian Timm e Erhardt Barth (Documento indicado na secção de referencias) capaz de tratar melhor nos nossos testes casos em que não existe visibilidade completa sobre a pupila.

A implementação do algoritmo usada tem como base a implementação disponível em (github.com/ trishume/eyeLike) tendo sido, no entanto, alterada, de forma a simplificar alguns cálculos reduzir a quantidade de testes feitos na fase de definição do centro da pupila.

Adicionamos também ainda nesta fase distinção entre olho direito e esquerdo verificando para isso a posição relativa dos olhos.

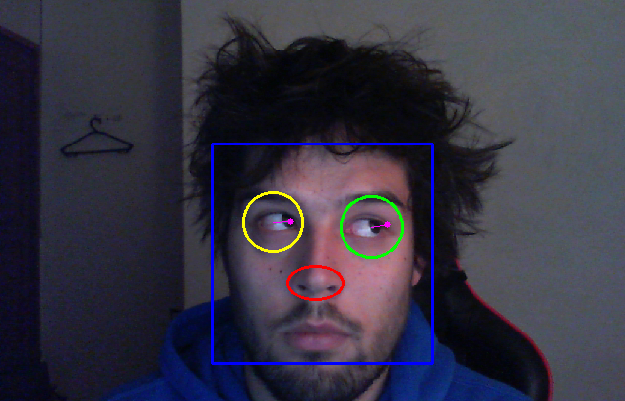


Figura 3. Resultado de eye tracking com vetor de direção do olhar

Testou-se ainda nesta fase captura de mais que uma face em simultâneo, a imagem abaixo representa o resultado obtido:

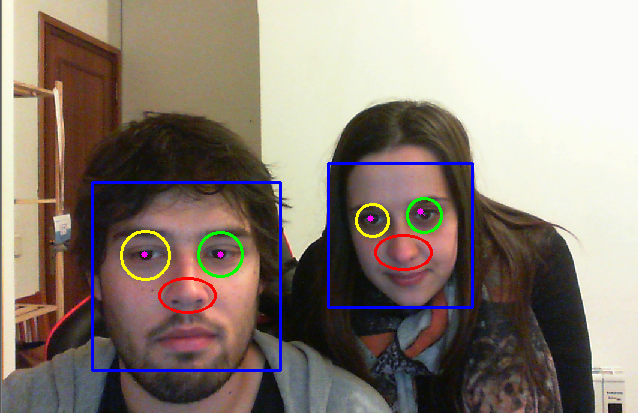


Figura 4. Captura de múltiplas faces

V. FACE TRACKING

Para face tracking e garantir que o programa consegue detetar as faces e também acompanhar e distinguir múltiplas faces, sem aumentar demasiado a penalização de desempenho vamos usar um historio de posicionamento das caras com calculo continuo de velocidade e localização relativa dos elementos da face.

Para isso foram criadas estruturas para armazenar a posição, tamanho e velocidade da face olhos e nariz sendo atribuído a cada um destes elementos um UUID para identificação.

É armazenado ainda um histórico de posições e cada elemento tem um TTL associado (time to live) para casos de perda de visibilidade e um contador temporal.

Após implementação podemos ver que acima de cada face aparece representado o UUID e parâmetros temporais dessa mesma face que a acompanha ate a mesma desaparecer da camera.

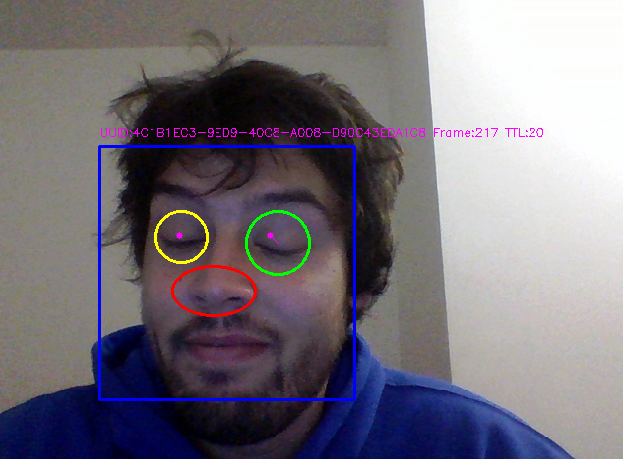


Figura 5. Face tracking com UUID representado acima da face

VI. FACIAL EXPRESSIONS

Para esta fase do projeto o objetivo é detetar expressões faciais e trabalhar essencialmente na zona da boca. Começou-se por experimentar deteção de sorrisos com haars-cascades de forma semelhante ao que foi feito no primeiro ponto deste trabalho.

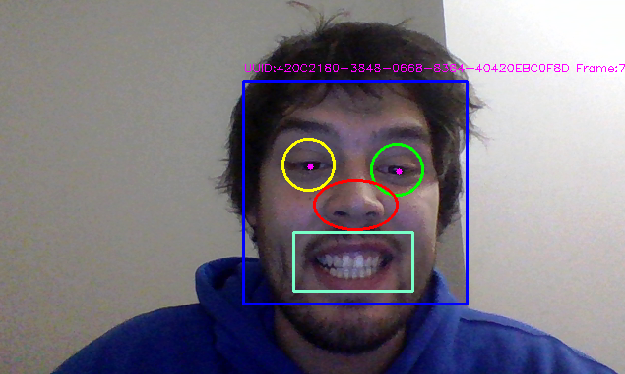


Figura 6. Deteção de sorrisos com haars-cascade

Experimentamos posteriormente um projeto de avaliação de estado emocional com opencv, mas com resultados um pouco aquém do esperado.

VII. IMAGEM TERMICA

Para captura de imagem térmica estamos a usar uma camera FLIR Leptron 50 conectada via SPI ao raspberrypi.

VIII. IMAGE MATCHING

//TODO

IX. CONCLUSÃO

Foi possível concluir todos os objetivos propostos para este projeto tendo sido implementado um sistema de reconhecimento facial capaz de detetar de forma eficaz vais elemento de uma face, estimar direção do olhar, seguir a cara, fazer deteção de expressões faciais e capturar imagem da camera térmica.

Apesar de tudo existe ainda grande margem para melhorias neste trabalho, poderíamos ter usado optical flow para compensar melhor casos de perda total de visibilidade de caras e melhor resolução de oclusão de varias caras, os resultados obtidos para a imagem térmica necessitam ainda de algum trabalho.

Foi ainda possível testar o desempenho de openCV no raspberrypi e detectar alguns dos desafios relacionados com o uso da camera térmica, como por exemplo a dificuldade em conseguir objetos com morfologia igual quando vistos com a camera térmica e com uma camera normal.



Figura X. Exemplo de falso positivo na deteção de faces.

X. REFERÊNCIAS

* Documentação da disciplina
* Accurate Eye centre localization by means of Gradients - Fabian Timm and Erhardt Barth – inb.uni-luebeck.de/publikationen/pdfs/TiBa11b.pdf
* Haar Cascades - alereimondo.no-ip.org/OpenCV/34
* Face and facial feature detection evaluation alereimondo.no-ip.org/OpenCV/uploads/37/Camera ReadyPaper63.pdf
* Emotime - github.com/luca-m/emotime