

Visão por Computador – Eye Tracking e Processamento de Imagem Térmica

Universidade de Aveiro

José Ferrão N°63944

Ricardo Ribeiro N°64694

I. INTRODUÇÃO

O objetivo do nosso projeto é a implementação de um sistema de reconhecimento facial 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 3.1 e 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 (Cor e Térmica)
- Criação de imagem térmica de alta resolução com 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.

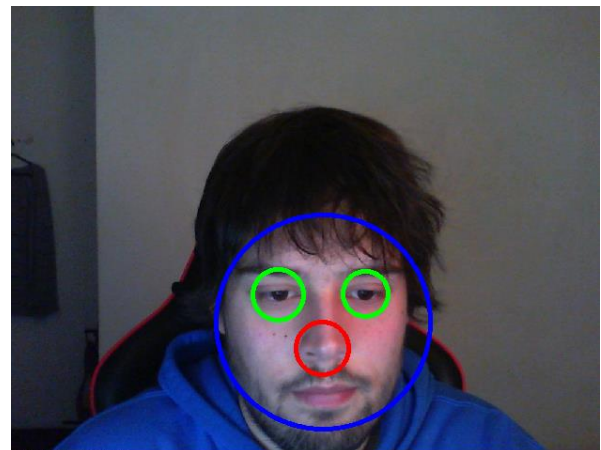


Figura 2. Resultado de Face Detection

Ao longo desta fase fomos experimentando várias cascadas já treinadas disponíveis na internet, tendo optado por 2 cascadas 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.

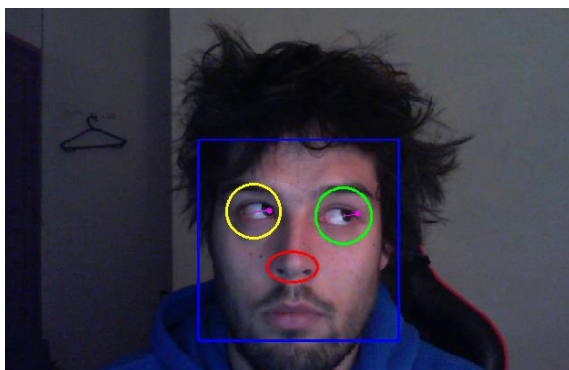


Figura 3. Resultado de eye tracking com vetor de direção do olhar

Testou-se ainda nesta fase captura de mais que uma face em simultâneo, a fig.4 representa o resultado obtido.

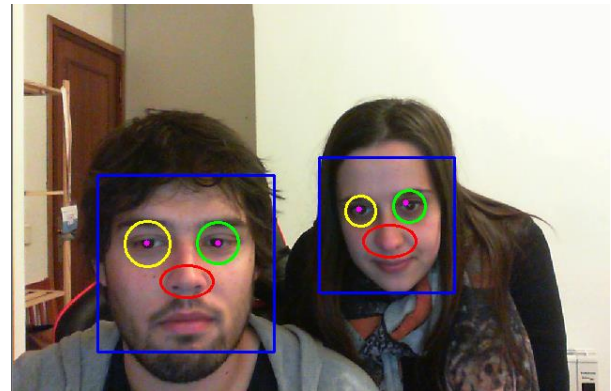


Figura 4. Captura de múltiplas faces

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.

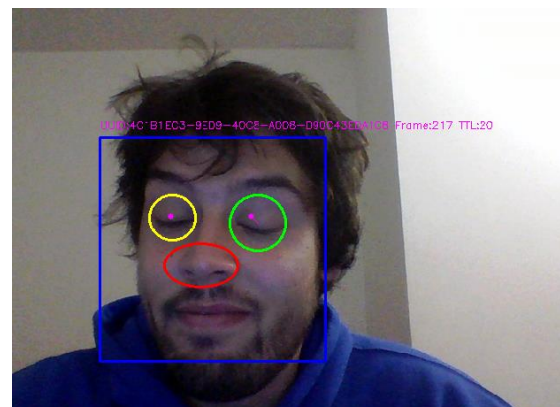


Figura 5. Face tracking com UUID representado acima da face

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.

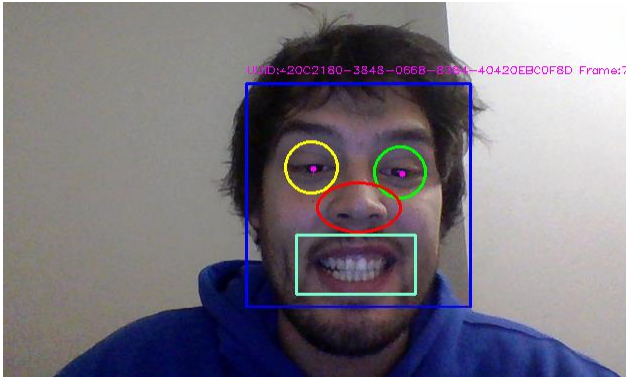


Figura 6. Deteção de sorrisos com haars-cascade

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 Lepton com uma resolução de 80x60, que captura radiação infravermelha com um comprimento de onda entre 8 μ m e 14 μ m. Usando uma breakout board, a camera é conectada ao raspberrypi através da interface SPI.

A imagem térmica obtida é de um canal, em grayscale, onde os pontos mais quentes estão a branco e os mais frios a preto.

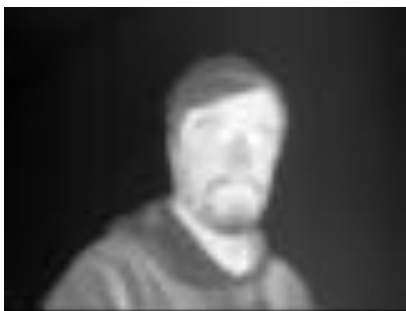


Figura 7. Imagem obtida com a camera térmica

A nossa visão é menos perceptível a mudanças numa imagem de grayscale, do que numa imagem com cores.

As imagens térmicas são imagens em grayscale mas são muitas vezes apresentadas como imagens com cor, que usam por exemplo o azul para as temperaturas mais frias e o vermelho para as mais quentes.

Desta forma, para uma melhor perceção das mudanças na imagem obtida da camera térmica tentamos colorir essa imagem. Uma forma simples de o fazer é usar a função `applyColorMap()` e o `ColormapTypes`, existente na biblioteca do opencv, usando assim um tipo de mapa de cores `COLORMAP_JET`.

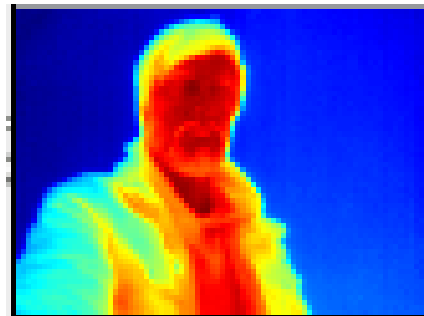


Figura 8. Imagem térmica colorida

VIII. IMAGE MATCHING

A calibração é usada para obter os parâmetros intrínsecos de uma ou mais camera, como distancia focal, projeção do ponto central, bem como a posição relativa e orientação entre as cameras.

No nosso projeto o objetivo é tentar através da imagem térmica e da imagem de uma camera do raspberrypi, obter os parâmetros de calibração de cada imagem. Estes parâmetros estabelecem uma relação entre os pontos 3D do mundo e os pontos 2D correspondentes das cameras, podendo assim criar uma reconstrução e retroprojeção da imagem térmica.

Para tal, é necessário também um objeto de calibração, que neste caso usamos uma imagem de um padrão de mesa de xadrez, como nas aulas.

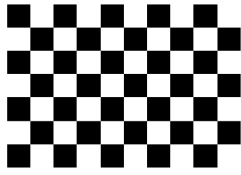


Figura 9. Padrão usado para calibração de camera

Como na imagem térmica não temos a distinção das cores da imagem de xadrez, pois estão todos a mesma temperatura, optamos por uma solução que nos dê uma distinção dos quadrados pretos e dos brancos.

Essa solução passa por usar um candeeiro apontado para a imagem de xadrez durante algum tempo, ficando assim com uma distinção dos diferentes quadrados.

Tal acontece devido a cor preta absorver a luz branca emitida pelo candeeiro e a cor branca reflete a luz branca, fazendo assim com que a temperatura dos quadrados pretos seja superior à dos quadrados brancos.

Posicionamos as duas camera lado a lado, tentando obter uma imagem semelhante, mas a imagem obtida num instante de tempo na camera térmica era diferente da imagem obtida na outra camera no mesmo instante de tempo. Isto deve-se ao facto de a camera do raspberrypi ter um atraso superior ao da camera térmica ao captar a imagem do mundo.

Para resolver este problema colocamos as cameras a apontar para o xadrez e deixamos estabilizar a imagem durante alguns segundos e fazemos a captura das duas imagens tendo assim das imagens semelhantes.



Figura 10. Imagens do padrão na camera do raspberrypi (esquerda) e na camera térmica (direita)

Como a resolução da imagem térmica é muito baixa 80x60 e mesmo apos aquecer a folha a diferença entre o branco e preto não é muito pronunciada não conseguimos, obter os cantos que são necessários no padrão de xadrez não

sendo possível obter parâmetros de calibração para a camera do raspberrypi.



Figura 11. Calibração da imagem da camera do raspberrypi.

Contudo, não podemos prosseguir com o objetivo de reconstruir uma projeção da imagem térmica, junto com a imagem da camera do raspberrypi.

Alternativas para este desafio no futuro será construção de uma nova folha com padrão de xadrez feita com papel de alumínio para onde poderemos projetar luz IR que será possivelmente bem visível na camera térmica, o uso de um padrão de dimensões mais pequeno e aumentado, e também o uso de outro material de cor preta que absorva mais energia do que a folha de papel.

IX. CONCLUSÃO

Foi possível concluir maior parte dos objetivos propostos para este projeto tendo sido implementado um sistema de reconhecimento facial capaz de detetar de forma eficaz vários elementos de uma face, estimar direção do olhar, seguir a cara, fazer deteção de expressões faciais e capturar imagem da camera térmica, não conseguindo chegar a um resultado na projeção da imagem térmica junta com uma imagem de melhor qualidade.

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 e de alguma melhoria para conseguir a detenção do padrão de xadrez para uma possível calibração.

Foi ainda possível testar o desempenho de openCV no raspberrypi e detetar 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.

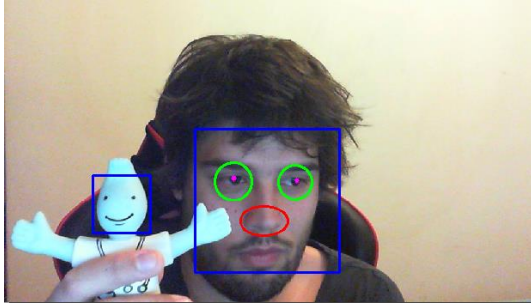


Figura 12. Exemplo de falso positivo na deteção de faces.

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/CameraReadyPaper63.pdf
- Emotime - github.com/luca-m/emotime
- Documentação OpenCV (docs.opencv.org/3.1.0/d3/d50/group__imgproc__colormap.html)
- cseweb.ucsd.edu/~yuc007/documents/thermal.pdf