

# Visão por Computador

## 09 - Motion

Diogo Corte, Diogo Silva  
 DETI, Universidade de Aveiro  
 Aveiro, Portugal  
 {diogo.corte, dbtds}@ua.pt

**Resumo –** Este relatório foi realizado para a unidade curricular de Visão por Computadores com o objetivo de demonstrar a percepção da aula (VC Exercises 09). Sendo que o objetivo principal é análise de algoritmos existentes que resolvem problema como optical flow, separação do foreground/background e tracking de objectos.

### NOTA SOBRE AS IMAGENS

Apesar de as imagens terem aparência pequena, dispõem de boa resolução sendo que se for aplicado zoom sobre a imagem é possível ver com detalhe que se pretende.

### I. DESCRIÇÃO DOS FICHEIROS

O conteúdo da pasta src contém os seguintes ficheiros:

1. farneback.cpp
2. foregroundDetection.cpp
3. tracker.cpp
4. lucaskanade\_tracking.cpp
5. Video\_001.avi
6. Video\_003.avi

### II. OPTICAL FLOW

Para além dos exemplos referidos neste capítulo, também existe outro exemplo de optical flow que é apresentado na secção de Object Tracking, devido ao intuito do seu uso ser para object tracking em vez do exemplo seguinte de Farneback que apenas representa o fluxos calculados. O exemplo do Lukas-Kanade encontra-se na secção IV-B.

#### A. Farneback Optical flow

Source: <http://funvision.blogspot.pt/2016/02/opencv-31-tutorial-optical-flow.html>

O algoritmo de Gunnar Farneback é útil para obter uma estimativa de fluxos de movimento entre duas frames consecutivas, baseado em expansão polinomial. Para este efeito o OpenCV fornece o método calcOpticalFlowFarneback.

Para o cálculo de fluxos de movimento são necessárias duas imagens em escala de cinzento, a representação dos fluxos é pontual no caso do bloco permanecer estático ou linhas a começar no ponto e com o sentido do deslocamento em caso de movimento. Os fluxos ópticos representados na Figura 1 significam o movimento da cabeça do sujeito da direita para a esquerda.

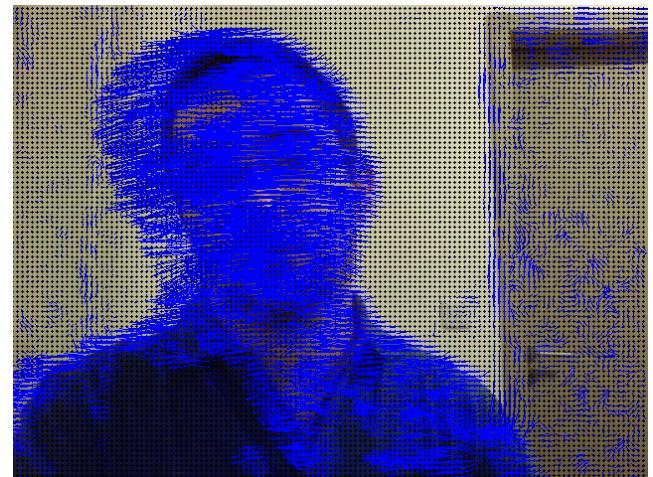


Figura 1: fluxos ópticos de Farneback

Farneback retorna para uma matriz com um conjunto de fluxos, que neste caso, de forma a serem representadas foi usada uma grelha com uma unidade de medida de 5 pixels, para cada ponto dessa mesma grelha é representado numa linha cujo o fluxo é obtido diretamente da matriz obtida pela função de Farneback.

### III. BACKGROUND/FOREGROUND SEPARATION

#### A. OpenCV Background Subtraction Tutorial

Source: [http://docs.opencv.org/trunk/d1/dc5/tutorial\\_background\\_subtraction.html](http://docs.opencv.org/trunk/d1/dc5/tutorial_background_subtraction.html)

Videos: [http://bmc.iut-auvergne.com/?page\\_id=24](http://bmc.iut-auvergne.com/?page_id=24)

Subtração de fundo é uma técnica largamente usada para gerar uma máscara do foreground (imagens de pixels com valores binários, em que preto significa estático e branco movimento) com uma câmera estacionária.

Na imagem 2 é possível verificar que o fundo que é a relva, o passeio e o edifício é subtraído ao foreground com sucesso, sendo que as pessoas que causam o movimento na imagem são as únicas que aparecem na imagem do lado direito.



Figura 2: Subtração de fundo em ficheiro de vídeo

Na imagem 3 é bastante similar ao interior, mas sendo que a leitura foi efetuada de uma câmara em vez de um ficheiro vídeo. Sendo que o fundo do quarto é questão desaparece da imagem ficando apenas o foreground que contém a pessoa em movimento.



Figura 3: Subtração de fundo em captura de vídeo

Para este efeito o OpenCV fornece o `BackgroundSubtractor` que cria um modelo dinâmico de background, realimentado com as frames passadas, ao qual subtrai a frame atual. Pode ser inicializado de duas formas distintas, respetivamente `cv::bgsegm::BackgroundSubtractorMOG` e `cv::BackgroundSubtractorMOG2`. Enquanto MOG é estreitamente binário, limitando o seu resultado ao binário preto ou branco, MOG2 utiliza a escala de cinzentos para representar o grau de alteração da cena, usando cores intermédias para representar por exemplo sombras.

#### IV. OBJECT TRACKING

##### A. OpenCV Tracker

Source: [http://docs.opencv.org/3.1.0/d2/d0a/tutorial\\_introduction\\_to\\_tracker.html](http://docs.opencv.org/3.1.0/d2/d0a/tutorial_introduction_to_tracker.html)

O OpenCV fornece uma ferramenta que permite fazer o tracking de objectos, sendo que o Tracker precisa de ser iniciado com uma box que se denomina de ROI, sendo que a imagem 4 mostra a inicialização do ROI (Region of Interest) na face de uma pessoa.

Após o tracker ser inicializado basta a cada frame que se pretende localizar o objecto (o mais provável é querer fazer o tracking frame a frame) é necessário invocar o método seguinte:

---

```
tracker->update(frame, roi);
```

---

Sendo que o tracker recebe a frame e envia para um ponteiro ROI (Region of Interest) que é um retângulo a nova localização do ROI. Ainda retorna true se o ROI não tiver sido modificado entre ciclos, o que ajuda para evitar processamento desnecessário.

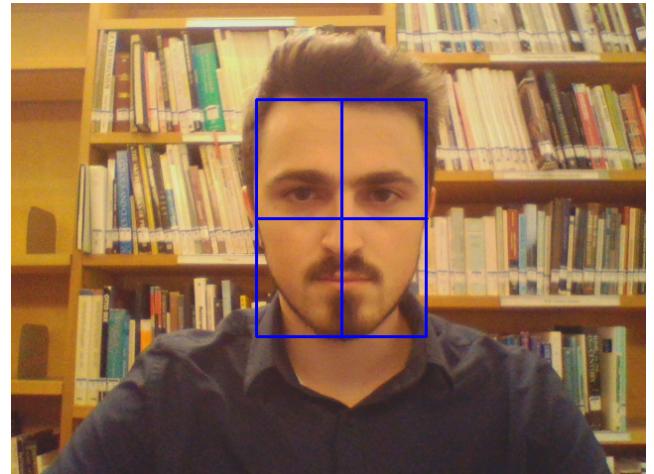


Figura 4: Inicialização do Tracker

Na imagem 5 é possível verificar que o Tracker permite um bocado de margem antes de perder o tracking do objecto em questão.

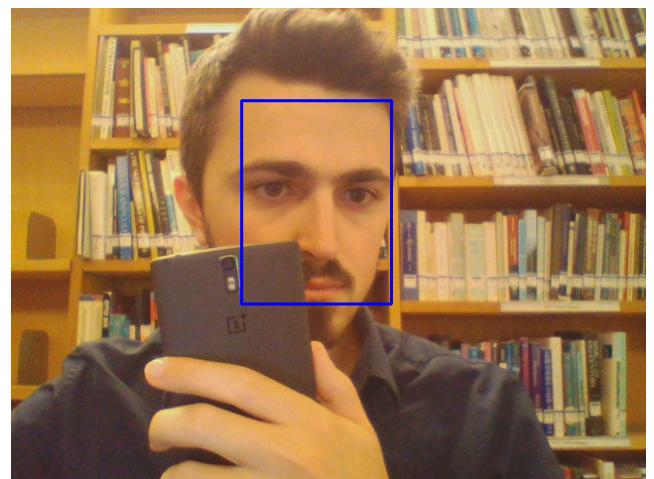


Figura 5: Tracking de cara

No entanto, se o objecto for algo pequeno como um olho, é possível que um movimento brusco faça com que o Tracker perca o tracking do mesmo. Ou seja, objectos mais pequenos são mais susceptíveis a serem perdidos durante o tracking, como se pode verificar no exemplo seguinte, na imagem 8.

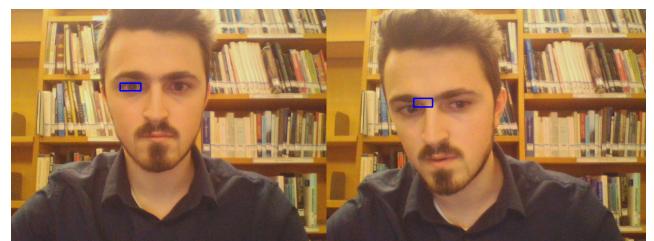


Figura 6: Tracking do olho, antes e depois de perder o tracking

### B. Using Lucas-Kanade optical flow to track

Source: <https://github.com/opencv/opencv/blob/master/samples/cpp/lkdemo.cpp>

O algoritmo de Lucas-Kanade é um método diferencial para o cálculo de uma estimativa de fluxo ótico.

Para tal o algoritmo assume que o deslocamento de uma imagem entre duas frames consecutivas é pequeno e aproximadamente constante para uma vizinhança próxima do ponto de interesse, pelo que os resultados de tracking deste algoritmo obtêm melhores resultados para velocidades baixas e constantes e/ou frame rates elevados.



Figura 7: Lucas Kanade Tracking de objecto em movimento

Para velocidades de movimento relativamente altas o algoritmo perde os pontos de referência, para velocidade baixas o algoritmo segue os pontos de referências com uma precisão bastante boa, isto para movimentos de translação (Figura 7), no entanto tal não acontece para movimentos de rotação sobre objetos irregulares uma vez que a localização espacial dos pontos de interesse definidos em relação à sua vizinhança também é alterada.



Figura 8: Lucas Kanade Tracking de objectos com cena em movimento

O algoritmo de tracking Lucas-Kanade de objectos com a cena em movimento a uma velocidade constante obtém bons resultados (Figura 8).