MC920— Trabalho 5

Professor: Hélio Predini

Leonardo Rodrigues Marques `RA: 178610`

1 Especificação do Problema

O objetivo deste trabalho é detectar movimentos por meio da comparacão de imagens capturadas por uma camera de vídeo. A identificação de movimentos na cena é realizada pela verificação da diferença entre o quadro atual e o quadro anterior. Caso os conteúdos dos quadros sejam iguais, então não há a mudanças aparentes na cena. Quando há diferenças, os locais de mudanças (movimento) serão apresentados.

2 Implementação

A implementação foi feita seguindo os passos sugeridos no documento do trabalho. Inicialmente foi aberto a câmera com a função e verificado sucesso na operação:

```
video = cv2.VideoCapture(0)
    if not video.isOpened():
        print("Error opening camera")
        exit()
```

Após isso, foi criado um laço de iteração para verificar continuamente os frames obtidos pela câmera. Os frames foram obtidos usando a função `read` do OpenCv e transformados em escala de cinza. Um filtro gaussiano foi aplicado para remoção de ruído, e uma processo de limiarização para suavização da imagem. E por fim, foi calculado a diferença entre as duas imagens.

```
 frame_ok, frame = video.read()
if not frame_ok:
    print("Error receiving frame")
    break
frame_mono = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

if frame_prev is None:
    frame_prev = frame_mono
    continue
```

```
frame_difference = cv2.absdiff(frame_mono,frame_prev)


frame_gaussian_blur = cv2.GaussianBlur(frame_difference, (5, 5), 0)
frame_ok, frame_threshold = cv2.threshold(frame_gaussian_blur, 15, 255,
cv2.THRESH_BINARY, frame_gaussian_blur)
```

Após a preparação da imagem, foi aplicado a função `findCountours` para obter os os prontos de fronteira. Para contorno obtido, foi calculado uma área e um quadrado desenhado caso tivesse uma area de 1000.

```
 frame_contours = frame
        for contour in contours:
            area = cv2.contourArea(contour)
            if area > 1000:
                x, y, w, h = cv2.boundingRect(contour)
                frame_contours = cv2.rectangle(frame_contours, (x, y), (x + w, y + h), (0, 255, 0), 2
```

E por fim, os quadrados foram desenhados na captura do vídeo. Para sair, basta apertar a teclado **q**.

```
cv2.imshow('movement', frame_contours)
        if cv2.waitKey(1) == ord('q'):
            break
```

Para executar o programa, basta executar o comando:

```
python3 main.py
```

3 Resultados e Conclusão

O programa conseguiu detectar movimento de forma satisfatória. Algumas aprimorações, como: evitar desenhar quadrados pequenos dentro dos grandes, ajustar os limiares, poderiam otimizar a precisão da identificação de movimentos.

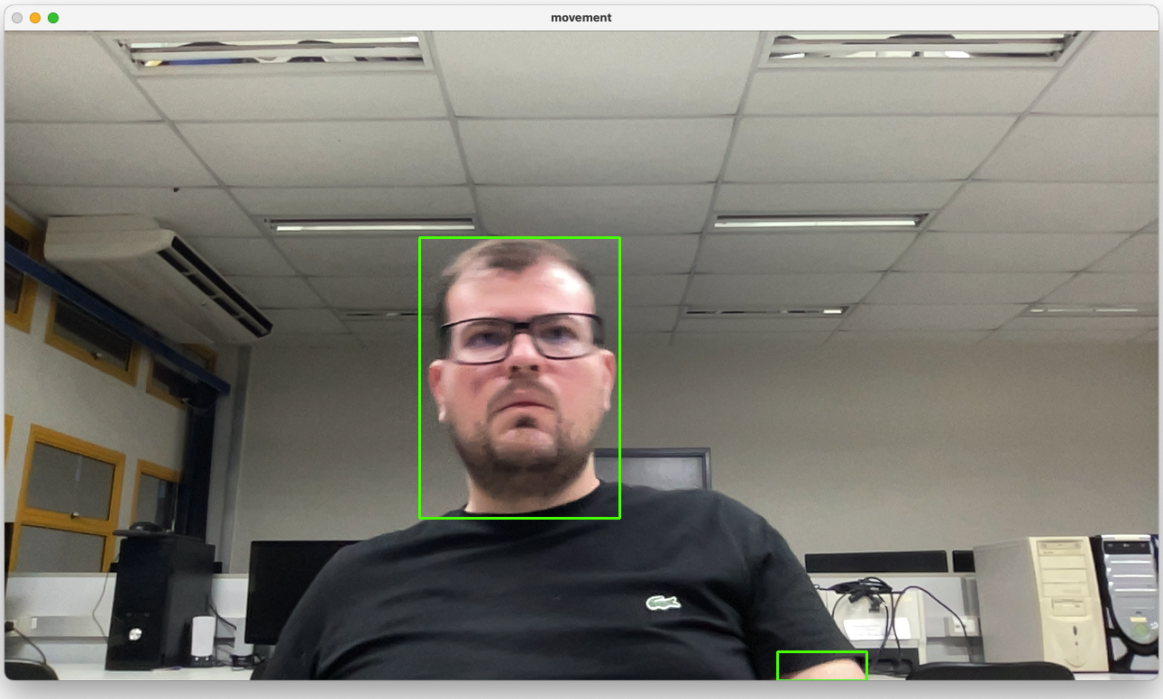

Figura 1: Identificação de movimentos em captura de vídeo.