

ObjectDetection

September 19, 2023

1 Background Subtraction for Object Detection

Este código implementa um sistema de subtração de fundo para a detecção de objetos em um vídeo. Ele começa lendo um vídeo de entrada e obtendo informações sobre seus quadros, como largura, altura e taxa de quadros. Em seguida, configura um gravador de vídeo para salvar o resultado da subtração de fundo.

O algoritmo utiliza um modelo de fundo que é continuamente atualizado com base nos quadros do vídeo, usando uma taxa de aprendizado (parâmetro “alpha”). A diferença entre o quadro atual e o modelo de fundo é calculada e um limiar é aplicado para criar uma imagem binária que destaca os objetos em movimento. O código exibe as imagens originais, o modelo de fundo, a imagem de diferença e a imagem binária em janelas separadas enquanto grava o resultado em um arquivo de vídeo. O processo continua até o final do vídeo ou até que o usuário pressione a tecla Esc para sair.

Esse sistema é útil para detecção de movimento em cenários de vigilância e rastreamento de objetos em vídeos.

```
[ ]: import cv2
import numpy as np

# Lê o vídeo
video_path = 'surveillance.avi'
cap = cv2.VideoCapture(video_path)

# Obtém informações do vídeo
frame_width = int(cap.get(3))
frame_height = int(cap.get(4))
frame_rate = 30

# Configura o gravador de vídeo
fourcc = cv2.VideoWriter_fourcc(*'MJPG')
out = cv2.VideoWriter('Background_Subtraction.avi', fourcc, frame_rate,
    ↪(frame_width, frame_height), 0)

# Inicializa parâmetros
alpha = 0.95
theta = 0.1

# Inicializa o fundo com o primeiro quadro
```

```

ret, background = cap.read()
if not ret:
    raise ValueError("Erro ao ler o primeiro quadro do vídeo.")

background = cv2.cvtColor(background, cv2.COLOR_BGR2GRAY).astype(np.float32)

while True:
    ret, frame = cap.read()
    if not ret:
        break

    # Converte o quadro atual para escala de cinza
    gray_frame = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY).astype(np.float32)

    # Atualiza o modelo de fundo
    background = alpha * background + (1 - alpha) * gray_frame

    # Calcula a diferença entre o quadro atual e o fundo
    diff_img = np.abs(gray_frame - background)

    # Aplica um limiar para criar uma imagem binária
    thresh_img = (diff_img > theta).astype(np.uint8) * 255

    # Mostra os quadros em janelas separadas
    cv2.imshow('New frame', gray_frame.astype(np.uint8))
    cv2.imshow('Background frame', background.astype(np.uint8))
    cv2.imshow('Difference image', diff_img.astype(np.uint8))
    cv2.imshow('Thresholded difference image', thresh_img)

    # Grava o quadro atual no arquivo de saída
    out.write(diff_img.astype(np.uint8))

    if cv2.waitKey(30) & 0xFF == 27: # Pressione Esc para sair
        break

# Libera os recursos
cap.release()
out.release()
cv2.destroyAllWindows()

```

```

[ ]: %cd /content/drive/MyDrive/Colab Notebooks/aula7.1
! sudo apt update
! sudo apt-get install texlive-full
! jupyter nbconvert --to pdf Detection.ipynb

```

```

[ ]: from google.colab import drive
drive.mount('/content/drive')

```