

9ª Atividade

Rastreamento de objeto por cor - Contornos

O código abaixo mostra como utilizar a biblioteca OpenCV para exibir um vídeo em uma janela a partir de um arquivo.

```
# display original video
cap = cv.VideoCapture('corgi_race.mp4')

while cap.isOpened():
    ret, frame = cap.read()
    # if frame is read correctly ret is True
    if not ret:
        print("Can't receive frame (stream end?). Exiting ...")
        break

    frame = cv.resize(frame, (640, 480))
    cv.imshow('frame', frame)
    if cv.waitKey(1) == ord('q'):
        break

cap.release()
cv.destroyAllWindows()
```

Utilizando o código acima como base e os conceitos apresentados em sala de aula, implemente um código que execute as seguintes tarefas:

1. Detecte e desenhe todos os contornos de uma cor à sua escolha no vídeo.
2. Rastreie os objetos desta mesma cor a partir destes contornos. Os objetos rastreados devem ser identificados a partir de um retângulo limite desenhado em volta destes. Deve haver apenas um retângulo por objeto detectado.
3. Repita a etapa anterior, desta vez utilizando o círculo limite e as elipses circunscritas dos contornos. Foi mais fácil traçar apenas um objeto geométrico por objeto rastreado?

Funções de referência

Algumas funções de referência necessárias para implementar as tarefas acima são descritas abaixo:

```
contours, hierarchy = cv.findContours(color_mask, cv.RETR_TREE,  
    cv.CHAIN_APPROX_SIMPLE)
```

- Encontra os contornos e a hierarquia dos contornos dos objetos em uma imagem binária.

```
cv.drawContours(img, contours, x, (r, g, b), thickness)
```

- Traça um ou mais contornos em *img* com a cor (r, g, b) e grossura *thickness*. O parâmetro *x* define qual dos contornos contidos em *contours* será desenhado. Para $x = -1$ todos os contornos são desenhados.

```
x = np.vstack(y)
```

- Retorna o vetor *x* obtido a partir do empilhamento vertical dos vetores contidos em *y*.

```
hull = cv.convexHull(contour)
```

- Retorna a casca convexa que engloba um contorno.

```
cv.polylines(img, [points], isClosed, (r, g, b), thickness)
```

- Traça um polígono em *img* a partir de um conjunto de pontos [*points*], que pode ser fechado para *isClosed* = *True* e aberto caso contrário. A linha traçada tem cor (r, g, b) e grossura *thickness*.

```
x, y, w, h = cv.boundingRect(contour)
```

- Retorna as coordenadas do canto superior esquerdo e a largura e a altura do retângulo que engloba um contorno.

```
rect = cv.minAreaRect(contour)  
box = cv.boxPoints(rect)  
box = np.intp(box)
```

- *cv.minAreaRect()* encontra o retângulo rotacionado de menor área que engloba o contorno. As funções seguintes retornam as coordenadas inteiras dos seus vértices.

```
(x,y),radius = cv.minEnclosingCircle(contour)
```

- Retorna as coordenadas do centro e o raio do menor círculo que engloba um contorno.

```
ellipse = cv.fitEllipse(contour)  
# or  
(x_el,y_el),(MA,ma),angle = cv.fitEllipse(contour)
```

- Retorna a elipse circunscrita a um contorno, definida pelo seu ponto central, eixos maior e menor e ângulo.

```
cv.ellipse(img,ellipse,(r,g,b),thickness)
```

- Desenha uma elipse em *img* de acordo com os parâmetros especificados.