

5ª Atividade

Filtragem no domínio da frequência

O código abaixo mostra como utilizar a biblioteca OpenCV para abrir um vídeo colorido e exibi-lo convertido para escala de cinza. Além disso, também é calculada a transformada de Fourier, e o seu espectro (o módulo da transformada) também é exibido como um vídeo. Neste segundo caso, é utilizada uma transformação de intensidade logarítmica para melhorar a visualização.

```
import numpy as np
import cv2 as cv
import matplotlib.pyplot as plt

%matplotlib inline

# display original video in grayscale and video spectrum
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

    # resize current frame and convert to grayscale
    frame = cv.resize(frame, (640, 480))
    frame = cv.cvtColor(frame, cv.COLOR_BGR2GRAY)

    # calculate shifted dft
    dft = cv.dft(np.float32(frame), flags = cv.DFT_COMPLEX_OUTPUT)
    dft_shift = np.fft.fftshift(dft)

    real = dft_shift[:, :, 0]
    imag = dft_shift[:, :, 1]

    # calculate spectrum
    spectrum = 20*np.log(cv.magnitude(real, imag)+0.0001)
    spectrum_norm = spectrum/np.max(spectrum)
```

```

# show grayscale frame and corresponding spectrum
cv.imshow('spectrum', spectrum_norm)
cv.imshow('frame', frame)
if cv.waitKey(30) == ord('q'):
    break

cap.release()
cv.destroyAllWindows()

```

Utilizando o código acima como base, assim como o que foi explorado na atividade anterior, escreva um código que realize as tarefas abaixo. Em todos os casos, mantenha o vídeo em dimensão 640×480 (ou menos), para evitar um custo computacional excessivo.

1. Implemente um filtro passa-baixa ideal, aplique o filtro à transformada de Fourier do vídeo e mostre o vídeo e o vídeo do espectro resultantes para diferentes valores de D_0 .
2. Implemente um filtro passa-baixa de Butterworth, aplique o filtro à transformada de Fourier do vídeo e mostre o vídeo e o vídeo do espectro resultantes para diferentes valores de D_0 e de n .
3. Implemente um filtro passa-alta gaussiano, aplique o filtro à transformada de Fourier do vídeo e mostre o vídeo e o vídeo do espectro resultantes para diferentes valores de D_0 .

Funções de referência

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

```

dft = cv.dft(np.float32(frame), flags = cv.DFT_COMPLEX_OUTPUT)
dft_shift = np.fft.fftshift(dft)

```

- A sequência de comandos acima retorna o objeto *dft_shift* contendo a transformada de Fourier de uma imagem *frame* qualquer.

```

real = dft_shift[:, :, 0]
imag = dft_shift[:, :, 1]

```

- objeto *dft_shift* contém dois canais, sendo *dft_shift[:, :, 0]* a parte real da transformada e *dft_shift[:, :, 1]* a parte imaginária.

```

cv.magnitude(real, imag)

```

- O espectro da transformada é o seu módulo. O módulo de um número complexo pode ser obtido a partir da função `cv.magnitude()`, utilizando as suas partes real e imaginária, como descrito acima.

```
f_ishift = np.fft.ifftshift(dft_shift)
frame = cv.idft(f_ishift)
```

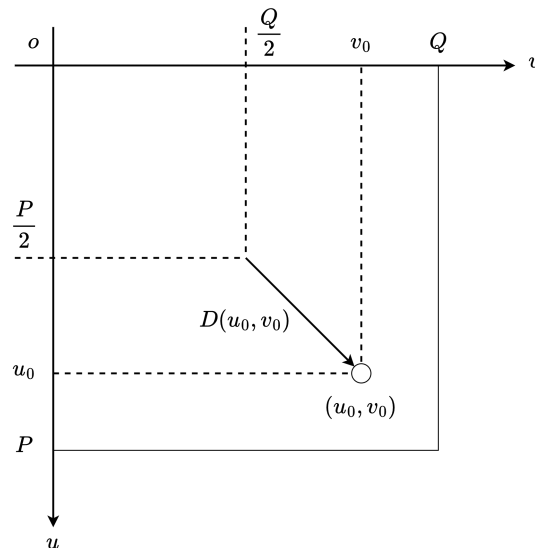
- A sequência de comandos acima retorna o objeto *frame* contendo a imagem correspondente a uma transformada de Fourier qualquer.

Referência sobre filtragem no domínio da frequência

A filtragem do domínio da frequência segue as seguintes etapas :

1. Calcular a transformada de Fourier da imagem.
2. Multiplicar a transformada da imagem por uma máscara, que é alguma forma geométrica no plano da frequência (a multiplicação ocorre termo a termo).
3. Calcular a transformada inversa.

As máscaras do processo acima são os filtros. Os filtros no domínio da frequência são parametrizados com relação à distância euclidiana de um ponto qualquer no espectro com relação ao seu centro geométrico, como ilustrado na figura abaixo. P e Q são as dimensões da imagem, (u_0, v_0) é um ponto qualquer na imagem e $D(u_0, v_0)$ é a distância euclidiana com relação ao centro geométrico da imagem, localizado em $(\frac{P}{2}, \frac{Q}{2})$.



A fórmula para a distância euclidiana é apresentada abaixo:

$$D(u, v) = \sqrt{\left(u - \frac{P}{2}\right)^2 + \left(v - \frac{Q}{2}\right)^2} \quad (1)$$

Para os filtros passa-baixa, por exemplo, todos os pontos cuja distância para o centro é menor (frequência menor) do que um determinado limiar são mantidos, enquanto os pontos mais distantes do que este limiar (frequência mais alta) tem o seu valor diminuído.

O filtro passa baixa ideal (ILPF) rejeita todas as componentes com frequência acima de um certo valor limiar (uma distância D_0), não havendo atenuação. Esta distância D_0 é tipicamente chamada de **frequência de corte**. Quanto maior o D_0 , menor será o efeito do filtro para o caso dos filtros-passa baixa. Ele é expresso pela seguinte função matemática:

$$H(u, v) = \begin{cases} 1, & \text{se } D(u, v) \leq D_0 \\ 0, & \text{c.c.} \end{cases} \quad (2)$$

O filtro ideal apresenta uma alta incidência de *ringing*, um efeito negativo onde aparecem anéis de variação de intensidade na imagem. Uma alternativa é o filtro passa-baixa de Butterworth (BLPF) de ordem n e frequência de corte a D_0 da referência é especificado pela seguinte função de transferência:

$$H(u, v) = \frac{1}{1 + \left(\frac{D(u, v)}{D_0}\right)^{2n}} \quad (3)$$

Para valores de n altos, os filtros passa-baixa de Butterworth apresentam os mesmos problemas dos filtros ideais. O filtro passa-baixa gaussiano (GLPF) corrige estes problemas, e tem a forma:

$$H(u, v) = e^{-\frac{D^2(u, v)}{2D_0^2}} \quad (4)$$

Os filtros passa-alta podem ser obtidos pelo complemento do filtro passa-baixa correspondente. Ou seja, um filtro passa-alta de Butterworth seria dado por:

$$H(u, v) = 1 - \frac{1}{1 + \left(\frac{D(u, v)}{D_0}\right)^{2n}} \quad (5)$$

Então, um filtro em frequência recebe como parâmetros a frequência de corte D_0 (e a ordem n no caso Butterworth) e usa esses parâmetros para calcular as funções $H(u, v)$ acima, denominada comumente de máscara. A filtragem ocorre quando esta máscara é multiplicada pela transformada de Fourier de uma imagem.