

ESTI019-17 - Codificação de Sinais Multimídia

Roteiro 01 - Captura de Video

Prof. Celso Kurashima

Prof. Mario Minami

04 de outubro de 2019

Identificação do Aluno

Nome Completo

ESCREVA AQUI A SUA RESPOSTA

RA

ESCREVA AQUI A SUA RESPOSTA

Atividades em Aula - Parte 01

1. No seu diretório "**Documentos**", crie uma pasta com seu primeiro nome e uma sub-pasta "**lab1**".
2. Dentro desta pasta, copie todos arquivos fornecidos, incluindo os videos e imagens,
3. Entre no **jupyter notebook** e abra o arquivo de programa fornecido em python:
lab1_video_python.ipynb.
4. Siga as instruções a seguir.

1: Compilar e executar um programa OpenCV

- O código abaixo realiza a leitura de um arquivo de **imagem** e mostra numa janela. Estude passo a passo os comandos.

In []:

```
# Estude este código e Execute
# (clique em "Run", com esta célula selecionada)
import numpy as np
import cv2 as cv

# Leitura e apresentação de uma imagem
img = cv.imread('messi5.jpg')
cv.imshow('Imagem',img)

k = cv.waitKey(0)

# Aperte a tecla ESC para sair e fechar o programa

if k == 27:
    cv.destroyAllWindows()
```

- O código abaixo realiza a leitura de um arquivo de imagem e mostra na mesma tela.
- O código obtém a resolução da imagem em pixels, e imprime o valor na tela.
- (OBS.: execute duas vezes o mesmo código, caso a imagem não apareça na primeira vez.)

In []:

```
# Estude este código e Execute
# (clique em "Run", com esta célula selecionada)
import numpy as np
import cv2 as cv
from matplotlib import pyplot as plt

# Leitura da imagem e obtenção da sua resolução
imgmessi = cv.imread('messi5.jpg')
altura,largura,camadas= imgmessi.shape
print("Resolução: ", largura, " x ", altura, " PIXELS. ", camadas, " camadas.")

# Mostra a imagem como gráfico do tipo "plot"
plt.imshow(imgmessi)
plt.xticks([]), plt.yticks([])
plt.show()
```

- Agora, passe para o próximo código.

2. Leitura de um arquivo de vídeo:

- Esse código realiza a leitura de um vídeo gravado em arquivo e mostra o vídeo em uma nova janela.
- Observe e modifique a taxa de quadros do vídeo no código.
- Observe e modifique a "escala", isto é a resolução da imagem, no código.

In []:

```
import time
import numpy as np
import cv2 as cv

# QPS = taxa do video. Digite o valor
QPS = 25

# Inicializa contador de quadros
Qcont= 0

# Escala da resolução do video
ESCALA = 1.0

# Objeto que indica o arquivo de video a ser lido
cap = cv.VideoCapture('videoCC.mp4')

# Modifique o arquivo por este abaixo
# comentando a linha anterior e descomentando a linha abaixo
#cap = cv.VideoCapture('big_buck_bunny.mp4')

# Inicia a marcação de tempo
start_time = time.time()

while(cap.isOpened()):
    # leitura da imagem no video
    ret, imagem = cap.read()

    if ret==True:
        # redimensiona a imagem multiplicando a largura e altura pela Escala
        img_nova = cv.resize(imagem, None, fx=ESCALA, fy=ESCALA, interpolation = cv.INTER_CUBIC)

        # obtenção da resolução das imagens original e redimensionada
        altura,largura,camadas= imagem.shape
        alt,larg,cam= img_nova.shape

        # mostra a imagem redimensionada na janela
        cv.imshow('VIDEO',img_nova)

        # Temporização: aguarda o proximo quadro durante (1/QPS) segundos
        # exemplo: se QPS= 25 fps, tempo de espera = 40 ms
        time.sleep(1/QPS)
        Qcont = Qcont+1

        if cv.waitKey(1) & 0xFF == ord('q'):
            break
    else:
        break

# Fim da marcação do tempo decorrido e cálculo da diferença
elapsed_time= time.time() - start_time

# Libera o objeto de leitura e fecha a janela
cap.release()
cv.destroyAllWindows()

# resolução original
print("Resolução original: ", largura, " x ", altura, " PIXELS. ")

# resolução após redimensionalmento da imagem
```

```
print("Resolução modificada: ", larg, " x ", alt, " PIXELS. ")

# tempo decorrido na exibição do video
print("Tempo total: %.2f segundos." % elapsed_time)

# quantidade total de quadros do video
print("Quantidade total de quadros do video: %d quadros." % Qcont)
```

3. Gravação de video em arquivo:

- Esse código realiza a leitura de imagens, mostra a sequência na janela, e salva essa sequência num arquivo no formato **mp4** de video.

In []:

```
import numpy as np
import cv2 as cv

# Digite a largura e altura do video
LARG= 640
ALT= 360
# Digite a escala desejada
ESCALA = 0.25

# Calculo da nova resolução da imagem
width= round(LARG * ESCALA)
height= round(ALT * ESCALA)
# QPS = taxa do video. Digite o valor
QPS = 25.0

# Indique o arquivo de video a ser lido
cap = cv.VideoCapture('big_buck_bunny.mp4')

# ----- #
# Estes comandos definem o codec e cria o objeto de gravar video
# Indique o arquivo mp4 a ser criado
fourcc = cv.VideoWriter_fourcc(*'MP4V')
out = cv.VideoWriter('saidavideo.mp4',fourcc, QPS, (width,height))
# ----- #

while(True):
    # leitura da imagem
    ret, frame = cap.read()

    if ret==True:
        # redimensiona a imagem multiplicando a largura e altura pela Escala
        frame2 = cv.resize(frame, None, fx=ESCALA, fy=ESCALA, interpolation = cv.INTER_CUBIC)

        # salva a imagem modificada
        out.write(frame2)

        # mostra a imagem redimensionada na janela
        cv.imshow('VIDEO redimensionado',frame2)

        if cv.waitKey(1) & 0xFF == ord('q'):
            break

    else:
        break

# Libera os objetos e fecha a janela
cap.release()
out.release()
cv.destroyAllWindows()
```

Atividades em Aula - Parte 02

1. Obtenção de videos:

(a) Obtenha dois videos de você mesmo, com fundo claro e boa iluminação, com aprox. 10 (dez) segundos de duração cada.

a.1. O primeiro video com movimentos em geral.

a.2. O segundo video com movimentos mais rápidos que o primeiro.

(b) Salve os videos na mesma pasta dos demais arquivos deste roteiro.

2. Modificação dos videos:

(c) Elaborar um novo programa, re-utilizando os codigos estudados na **Parte 01**, com os seguintes objetivos:

c.1. Executar o primeiro video na resolução original e na taxa original.

c.2. Executar o segundo video numa resolução maior OU menor que a original.

c.3. Executar o primeiro video numa taxa de quadros maior OU menor que a taxa original.

c.4. Salvar o segundo video com resolução menor que a original e com taxa de quadros maior que a taxa original.

(d) Anotar os resultados obtidos de todos os parametros para todos casos acima (taxa de quadros, resolução, tempo de execução, quantidade de quadros).

3. Elaborar o relatório das atividades práticas, no formato HTML5:

(e) Descreva todos os procedimentos realizados.

(f) Disponibilize o código elaborado na forma de arquivo **.ipynb**, e coloque o "link" para o mesmo no relatório html5.

(g) Insira os videos salvos no item (d), na forma multimídia do relatório html5.

(h) Envie os arquivos na pasta disponibilizada.

ESCREVA AQUI A SEU PROGRAMA

In []:

```
### INSIRA AQUI O CÓDIGO ###
```

Referências

MINICHINO, J., HOWSE, J.. Learning OpenCV 3 Computer Vision with Python, 2nd Edition, Packt Publishing, 2015.

Tutorial: OpenCV-Python Tutorials https://docs.opencv.org/3.4.1/d6/d00/tutorial_py_root.html
(https://docs.opencv.org/3.4.1/d6/d00/tutorial_py_root.html)

Tutorial: Getting Started with Images https://docs.opencv.org/3.4.1/dc/d2e/tutorial_py_image_display.html
(https://docs.opencv.org/3.4.1/dc/d2e/tutorial_py_image_display.html)

- Insira novas referencia. caso tenha utilizado.