INSTITUTO FEDERAL
São Paulo

Exercícios #02

Instituto Federal de São Paulo (IFSP)
Campus Caraguatatuba
Tópicos Avançados (TPA A6)

03 de Novembro de 2021

Objetivos A segunda lista de exercícios da disciplina de "Tópicos Avançados" tem como objetivo implementar e testar alguns recursos disponíveis na biblioteca OpenCV. Você irá trabalha com conversão de espaço de cores (i.e., RGB e HSV), separação e manipulação de canais de cores, e processamento de imagens e vídeos utilizando o OpenCV.

Exercício 2.01. Nesse exercício, você irá criar images coloridas com fundo preto, e irá desenhar polígonos de diferentes cores e tabanhos em posições aleatórias na imagem.

(a) **Criando uma image:** Utilize a função `np.zeros()` para criar uma imagem de tamanho 512×512. Essa imagem deve ter o fundo totalmente preto (i.e., $RGB = (0, 0, 0)$). Para representar uma imagem digital, é importante definir o tipo da estrutura de dados como `np.uint8` (i.e., uma estrutura que representa dados inteiros entre 0 e 255).

(b) **Desenhando um retângulo:** Utilize a função `cv2.rectangle()` para desenhar um retângulo verde no centro da image. Você deve utilizar dois *trackbars*[1] para definir a altura e a largura do seu retângulo. Utilize a função `cv2.createTrackbar()` para criar o *trackbar*.

(c) **Calculando a área do retângulo:** Crie uma função para calcular a área do retângulo. Você deve calcular a área todas as vezes que o usuário mudar o formato do retângulo. Utilize a função `cv2.putText()` [2] para escrever o tamanho do retângulo no canto superior esquerdo da image. Utilize qualquer cor de sua preferência.

(d) **Desenhando um círculo:** Utilize a função `cv2.circle()` para desenhar um círculo no centro da imagem sobre o retângulo verde. Você deve utilizar um *trackbar* para definir o raio do círculo. Crie uma função para calcular e exibir a área do círculo logo abaixo da área do retângulo.

[1] `https://docs.opencv.org/3.4.15/d9/dc8/tutorial_py_trackbar.html`
[2] `https://docs.opencv.org/4.5.3/d6/d6e/group__imgproc__draw.html`

(e) **Desenhando linhas:** Utilize a função `cv2.lines()` para desenhar duas linhas diagonais que interligam os cantos do retângulo. Essas linhas devem ser de cores distintas e devem atualizar de acordo com o tamanho atual do retângulo.

Exercício 2.02. Neste exercício, você irá utilizar transformações geométricas em cada *frame* capturado de uma *webcam* ou de um arquivo de vídeo (o mesmo que você utilizou no exercício passado). Você deve criar um *script* Python para capturar um *stream* de vídeo, coletar os dados de entrada de *trackbars* e alterar as propriedades da imagem usando transformação geométricas. utilize o arquivo `Ex202_image_transformations.py` para responder esse exercício.

(a) ***Stream* de vídeo:** Crie um *script* Python para capturar uma sequência de imagnes de um arquivo de vídeo ou câmera utilizando o OpenCV.

(b) **Crie uma janela com *trackbars*:** Utilize a funcão `cv2.namedWindow()`[3] para criar uma janela do OpenCV chamada `Transformations`. Adicione quatro *trackbars*, a saber: `Rotation`, `Scale`, `Translation (X)`, e `Translation (Y)`. Lembre-se de criar as funções para serem executadas quando o usuário mudar os valores dos *trackbars* e para definir o intervalo correto de cada transformação (e.g. rotação de 0° a 360°).

(c) **Função afim**: Desenvolva uma função chamada `affineTransformation()` que dada uma imagem de entrada, o ângulo θ, as duas variáveis bidimensionais de translação $t(x, y)$ e a escala s, a função irá retornar uma imagem modificada com uma função afim[4]. Utilize a função `cv2.warpAffine()`[5] para aplicar a função afim na imagem de entrada. Essa função do OpenCV utiliza uma transformação geométrica na imagem de entrada utilizando a seguinte operação de matrizes:

$$g(x, y) = f(M_{11}x + M_{12}y + M_{13}, M_{21}x + M_{22}y + M_{23}).$$

Dica: Crie uma matrix de transformação M de tamanho 2×3 combinando as matrizes de rotação, translação e escala.

(d) **Mostre a imagem processada:** Mostre a imagem original de entrada e a imagem processada em duas janelas distintas do OpenCV.

(e) **Combinando múltiplas imagens:**$^{(Extra:)}$ Tente mostrar as duas imagens na mesma janela `Transformations`. Você deve utilizar as funções `np.hstack()`[6] e `np.vstack()`[7] para combinar duas imagens na horizontal e vertical.

[3] `https://goo.gl/XbeKQ6`
[4] `https://en.wikipedia.org/wiki/Affine_transformation`
[5] `https://goo.gl/DHCFfG`
[6] `https://goo.gl/sxeqvx`
[7] `https://goo.gl/7eMMN2`

A Figura 1 mostra um exemplo de uma janela do OpenCV com os *trackbars* utilizados para modificar as propriedades das transformações geométricas, e ambas imagens combinadas horizontalmente.

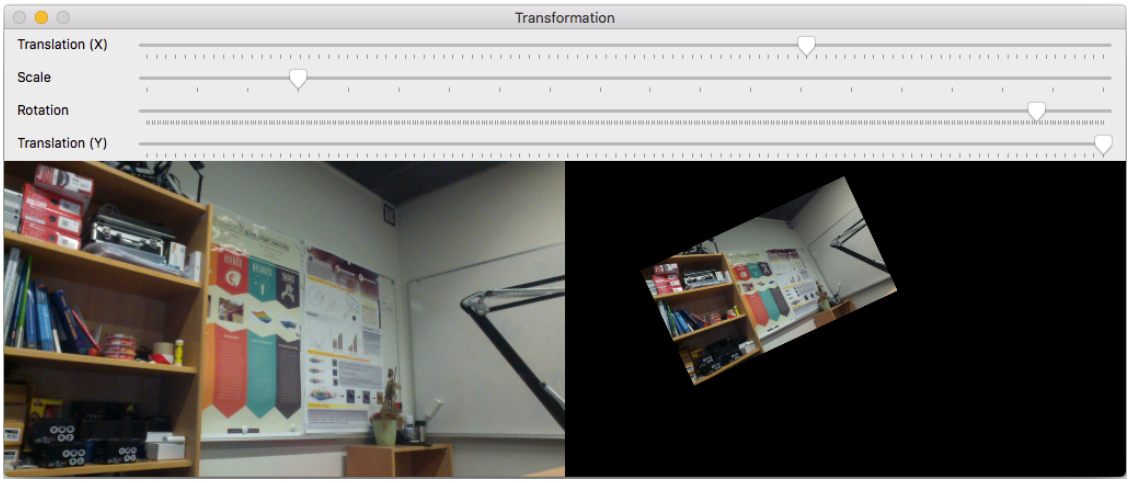

Figure 1: Exemplo de mútiplas transformações geométricas aplicadas em uma imagem.