UNIVERSITÉ DE GENÈVE
FACULTÉ DES SCIENCES
Département d'informatique

Université de Genève

Départment d'informatique

Imagerie Numérique

Geometric transformations

TP Class N° 5

November 20, 2020

Exercise 1. Affine transformations *(1 point)*

Programm a function that performs the affine transformations via matrix multiplication. For more details please see Theme 5, pp. 12-23.

You are given a triangle with the coordinates:

$$\begin{aligned}(x_1, y_1) &= (0, 0)\\ (x_2, y_2) &= (3, 2)\\ (x_3, y_3) &= (2, 4)\end{aligned}$$

Perform the following transformation to the original triangle:

(a) scaling in 2 and $\frac{1}{2}$ times;

(b) translation:

- by 2 along the horizontal axe;
- by 4 along the vertical axe;
- by 1.5 along the both axes;

(c) rotation:

- left by 55°;
- right by 180°;

(d) flipping across horizontal and vertical axes;

(e) bending via horizontal (with $b = 2$), vertical (with $c = 1.5$) and joint shearing (with $b = 1.5$ and $c = 2.5$).

For each transformation visualise the orignal and transformed figures in one plot. A visualisation example is given in Fig. 1.

Hint:

- to plot the legend use *plt.legend(.)*
- to plot the grid use *plt.grid("on")*

Important: The figures coordinates should be readable from your visulalisation. For this choose the proper scale and set *plt.axis('on')*. Otherwise, please, report the figure coordinates.

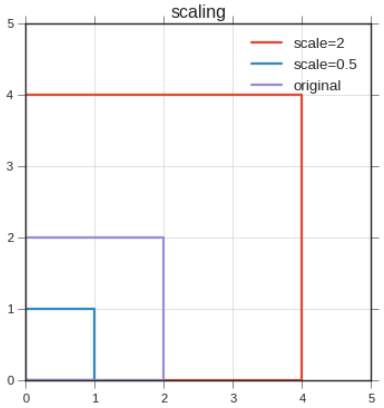

Figure 1: Visualisation example.

Exercise 2. *(1 point)*

(a) Read a grayscale image *img_1.png*.

- rotate it left by 30 degrees
- rotate it right by 45 degrees

  Hint: use the *skimage.transform.rotate(.)* function. Pay attention to the rotation direction.

Display the original and rotated images. Conclude about the effects of rotation (**hint:** resulting image size, border effects and image quality, etc.). Report what kind of interpolation was used at the rotation.

(b) Apply the subexercise (a) to the color image *lena.png*. Visualise the original and rotated images.

Exercise 3. Projective transformation *(1 point)*

Programm a function that performs the projective transformations via matrix multiplications. For more details, please, see the Theme 5 pp. 38-40.

You are given a rectangular with the coordinates:

$$\begin{aligned}(x_1, y_1) &= (0, 0)\\ (x_2, y_2) &= (2, 0)\\ (x_3, y_3) &= (3, 2)\\ (x_4, y_4) &= (1, 2)\end{aligned}$$

Perform the following projective transformation to the original figure:

$$\text{(a) } T_1 = \begin{bmatrix} 1 & 0.5 & 1 \\ 0.5 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \text{ and (b) } T_2 = \begin{bmatrix} 1.5 & -0.5 & 0 \\ 0 & 0.5 & 0 \\ 0 & -0.002 & 1 \end{bmatrix}$$

Visualise the orignal and transformed figures in one plot. Explain the obtained results.

Important: The figures coordinates should be readable from your visulalisation. For this choose the proper scale and set *plt.axis('on')*. Otherwise, please, report the figure coordinates.

Exercise 4. *(1 point)*

You are given an image *lena.png*.

(a) Convert this image to the grayscale. Apply the projective transformation with T_1 and T_2 from the exercise 3.
Hint: The projective transformation T for an image I in python with *skimage* package:

- *transf = skimage.transform.ProjectiveTransform(T)*
- *result_image = skimage.transform.warp(I, transf.inverse)*

(b) Perform the exercise (a) to the color image.

Display the original and transformed images. Conclude about the effects of transformations.

Exercise 5. Nearest interpolation *(1 point)*

Programm a function that performs the nearest interpolation.
Important: try to programm without loops.

(a) Read a grayscale image *img_1.png*. Downscale this image in 3 times by keeping only each third element. Upsale the downscaled image back to the original scale by using your function. Display the original and the scaled images. Explain the interpolation quality based on the MSE.

(b) Perform the subexercise (a) by using the *resize(.)* function from the *OpenCV* package with a parameter *interpolation = cv2.INTER_NEAREST*. Compare the obtained results with the results of exercise (a). Explain the difference between the results of your and Python functions, if any.

(c) Perform the subexercies (b) to the grayscale *lena.png* image. Explain the difference between the results obtained for the image *img_1.png* and *lena.png*.

Exercise 6. Bilinear interpolation *(1 point)*

Programm a function that performs the bilinear interpolation.

(a) Perform the exercise 5 (a)-(b) for a bilinear interpolation.

Hint: use the parameter *interpolation = cv2.INTER_LINEAR*

(b) Explain the advantages and disadvantages of the bilinear interpolation compared with the nearest interpolations.

Submission

Please archive your report and codes in "Name_Surname.zip" (replace "Name" and "Surname" with your real name), and upload to "`Assignments/TP5: Geometric transformations`" on `https://moodle.unige.ch` before Thursday, December 3 2020, 23:59 PM. Note, **the assessment is based not only on your code, but also on your report, which should include your answers to all questions and the experimental results**.