ITE4052 Computer Vision (Spring 2025)

Programming Assignment 1

- Deadline: April 21st 11:59 PM.
- Submit the Python code and a report summarizing the results (pdf) on the LMS.
- You can use either English or Korean for the report.
- Late submission will get the half score.
- If you use ChatGPT, you can be caught for a plagiarism (immediate F). Don't use it.

* Please install OpenCV, Matplotlib, Numpy, skimage with "*pip install ~*" and import them:
  import cv2, skimage.data
  import numpy as np
  import matplotlib.pyplot as plt

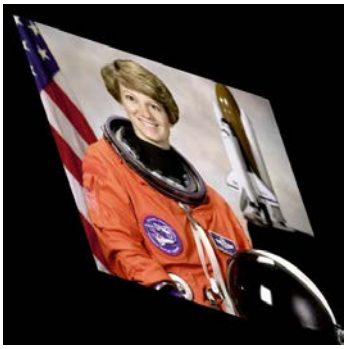
* **skimage** module is just for sample image.
  (Do not use this module after loading the sample image)

1. Projective Image Transformation [2pt]
Load an image with the *skimage.data.astronaut()*. Implement the 2D projective transform given a 3X3 transformation matrix using *cv2.warpAffine()* and *cv2.warpPerspective()* functions. Make the transformation matrix with Numpy using *np.array()* and find the transformation parameters that result in images similar to the ones provided below. Perform the following transformations: **(1) Rotation**, **(2) Similarity transform**, **(3) Affine transform**, and **(4) Projective transform**. While the transformed images do not need to match the provided ones exactly, compare various parameters and find the set of parameters that produce the most similar results. Cropping of the image during certain transformations is a natural outcome of the process and does not require any adjustment to the image size. Use *plt.show()* function to show the results on your own image. Indicate what transformation matrices (with parameters) you used in your report. *Finally, for your own image, repeat the entire process using a different set of transformation parameters.* And briefly analyze how different parameters within the same transformation affect the output image.

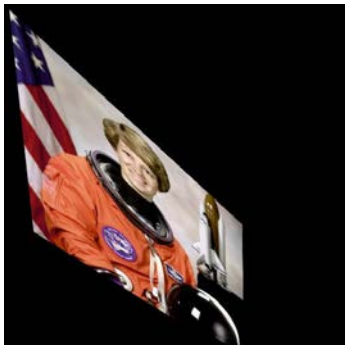

Original


Rotation


Similarity Transform



Affine Transform



Projective Transform

2. Color Space [2pt]

For an example image from *skimage.data.astronaut()*, **(1)** display color components R, G, and B, respectively. Also, transform color coordinates RGB to YCbCr and HSI, respectively. Display each component of the **(2)** YCbCr coordinates and each component of the **(3)** HSI coordinates, respectively. You can get RGB by using these codes: *R, G, B = image[:, :, 0], image[:, :, 1], image[:, :, 2]* . Instead of using Python functions, implement the color transformation by yourself using the formulations in the lecture slides (Also, refer to the formulations in [1, 2]). From the displayed results, compare the characteristics of the coordinates HSI, YCbCr, and RGB. Moreover, show **(4)** the modified images in the RGB space by increasing the value of the 'Blue' component in the RGB coordinates and the 'Saturation' component in the HSI coordinates. *Finally, for your own image, repeat the whole process again.*

[1] https://en.wikipedia.org/wiki/HSL_and_HSV
[2] https://en.wikipedia.org/wiki/YCbCr