ELL715 :- Assignment 1
Image Transformations

Saksham Jain(2017MT10747) Shivam Singla(2016MT10619)

Aim:

- Write a python/MATLAB code to perform affine transformations such as rotation, translation, shear, etc. and polar transformation of an image.

Transformations:

For all the experiments in the assignment, we used the following image provided along with the assignment. It's a colored image, so we get a 3-D matrix of the image. We perform all the transformations on each RGB channel differently. All the image files are attached along with the zip file and also shown in the report. The source code is also attached with the file.

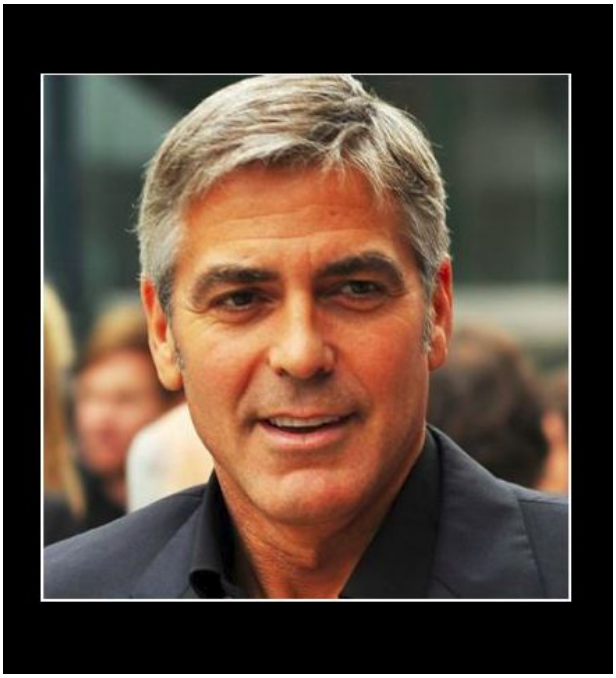

Fig. 1 - Test image

A. Polar transform :-

We transform the image using polar transform with origin at the center of the image. As for smaller values of the radius around the center, the distance between the points is very smaller, that's why some of the primary features like eyes and mouth are still distinctive in the transformed picture, and the effects of the transformations are more visible on head and neck. And at the bottom, we can see layers, as regions of blackout. Those are created because at higher values of the radii, the points go outside the image for certain theta values.

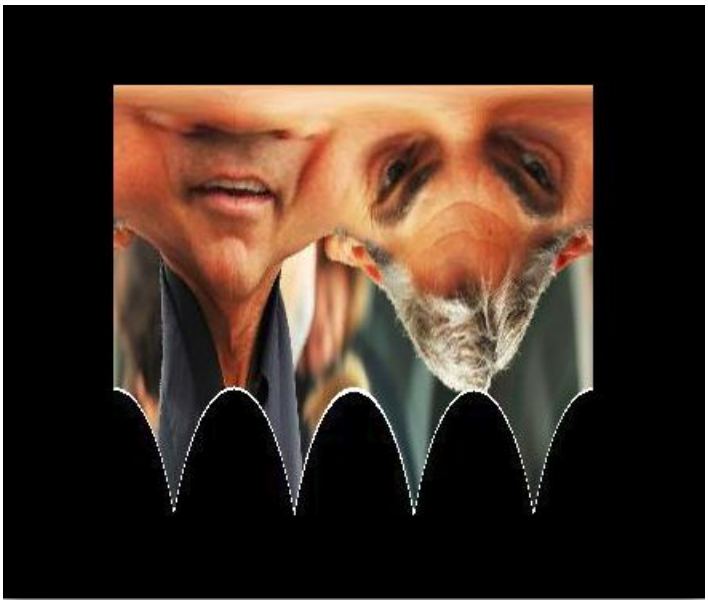

Fig. 2 - Polar transformed image

B. Rotation by 45 degrees and Flip :-

We rotate the image around an axis passing through the center and perpendicular to the image of the plane. This increases the size of the image to fit in all the pixels. The rest of the image is filled with black pixels, which are mostly around the corners of the new image. Then we flip the image, i.e., we take the mirror image of it around a plane passing through its center and perpendicular to the horizontal axis.

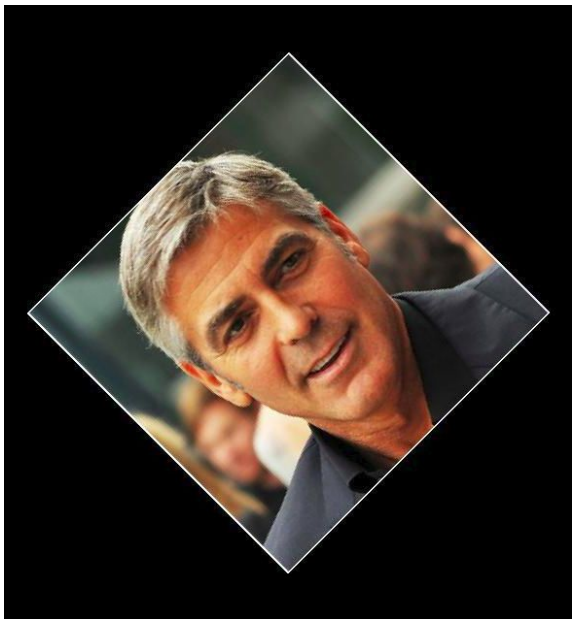

Fig. 3 - 45 degrees rotation

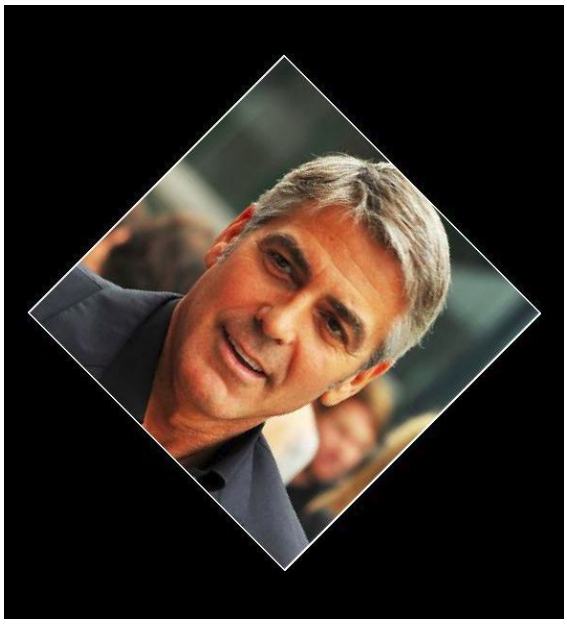

Fig. 4 - Flipped after rotation

C. Translate by an offset of 32 in both width and height :-

We transform the image by an offset of 32 in both the dimensions. This increases the size of the image by the amount of translation. We fill the remaining pixels of the image as black. There aren't any major changes in the image after it.

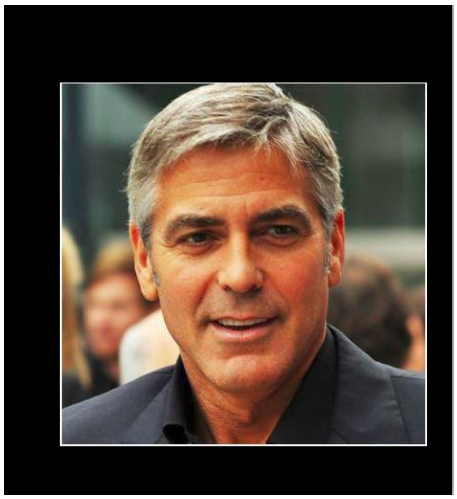

Fig. 5 - Translation of 32 in both the dimensions

D. Geometrically scaling the image by a factor of 3 :-

We scale the image by a factor of 3, by making a new image of reduced size and filling in every 3rd pixel in it from the original image. This reduces the size of the image by a factor of 3.

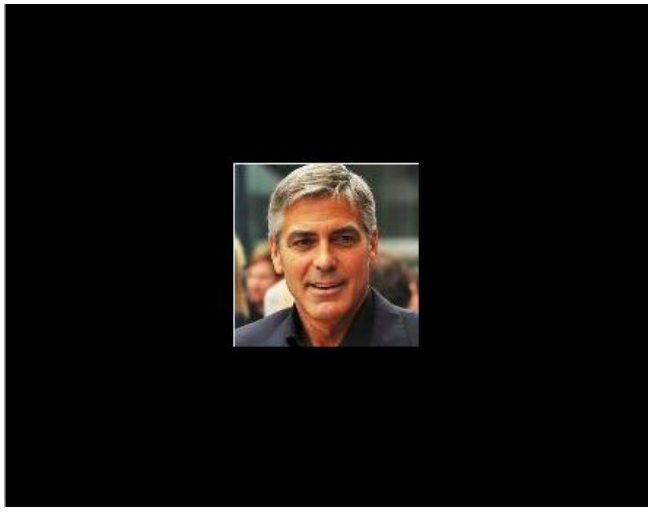

Fig. 6 - Scaling down by a factor of 3

E. Shear transform, rotation and scaling :-

We use the shear transform in the x-direction with a factor of 0.2x. Although the image is still quite distinctive after this, but with using larger shearing factors often destroys the contents of the image. After this, we use the rotation and scaling transform as performed earlier.

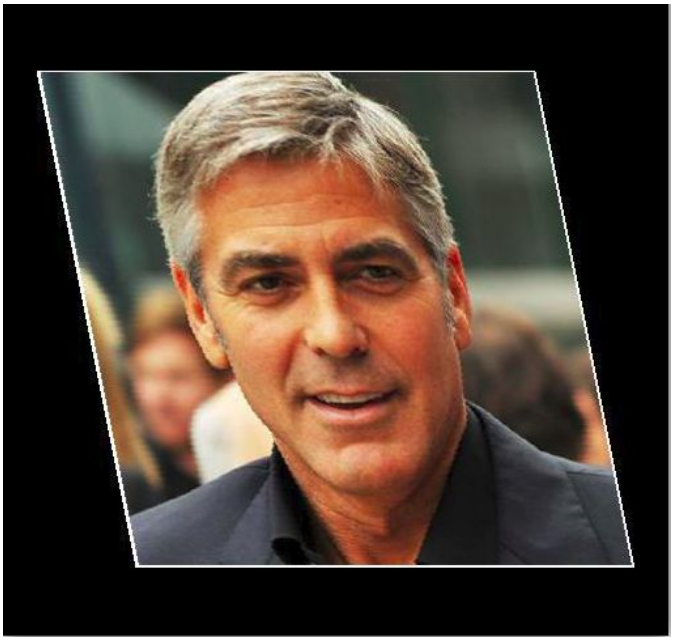

Fig. 7 - Shearing 0.2x in x - direction

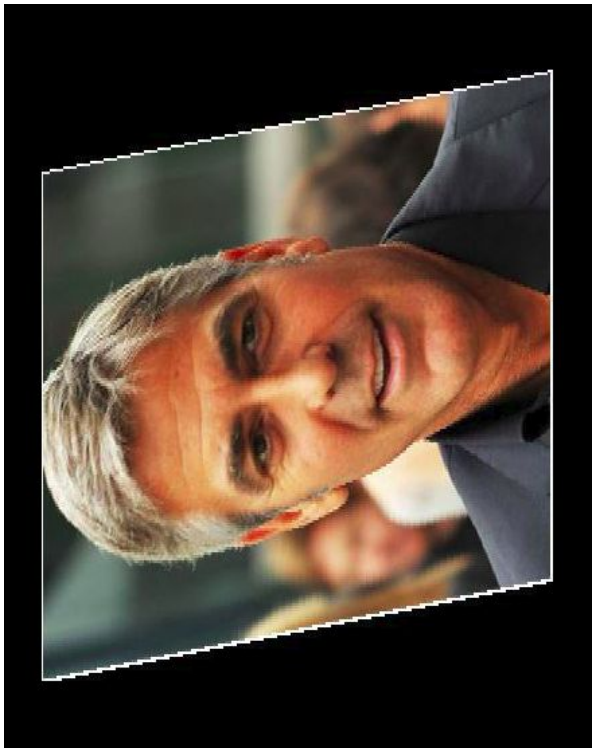

Fig. 8 - Rotating the image 90 degrees after shearing

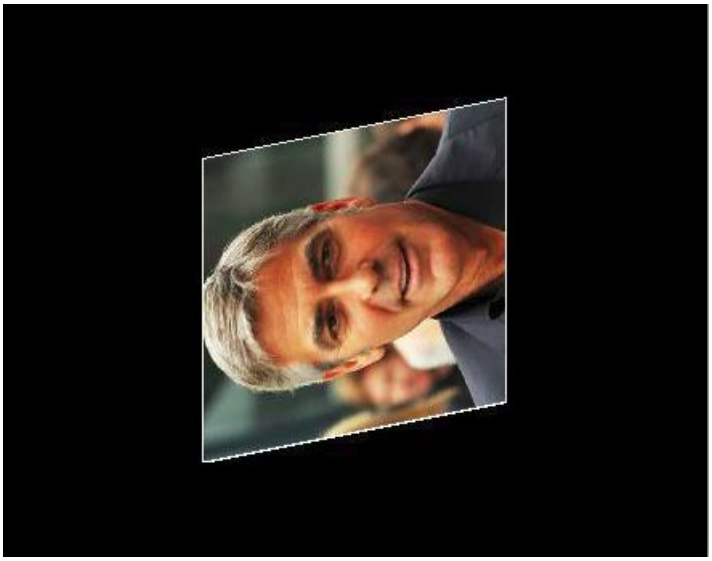

Fig. 10 - Finally scaling it down by a factor of 2

References:

1. Class lectures
2. https://docs.scipy.org/doc/numpy/reference/
3. https://matplotlib.org/3.1.1/users/index.html
4. https://docs.scipy.org/doc/scipy/reference/
5. https://docs.python.org/3.3/