IMAGE PROCESSING MATLAB_2

PARMAR hardiksinh

2.1 Translation - Rotation and Interpolation

2.1.1 Translation:

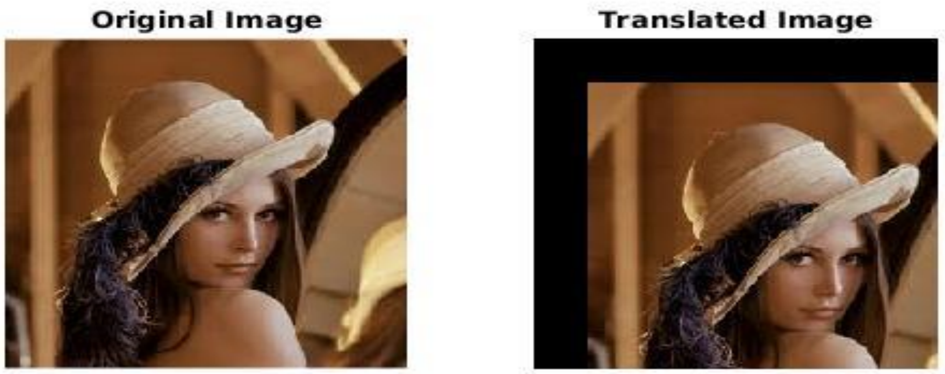
In a translation, every point of the object must be moved in the same direction and for the same distance.

shifts the image vertically by tx pixels and horizontally by ty pixels.

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \end{pmatrix}$$

My Function is: function [O] = myTranslation(I, tx, ty)
Here, I - Input Image
tx - Vertical Shift = 30
ty - Horizontal Shift = 30
Output(O) - Translated Image

Here the below image is the ('lena.jpeg') image and its output (translated image).



Original Image

Translated Image

2.1.2 Rotation :

A rotation is a circular movement of an object around a center (or point) of rotation.
rotate an image about its center (xc, yc) by an angle of theta.

IMAGE PROCESSING MATLAB_2

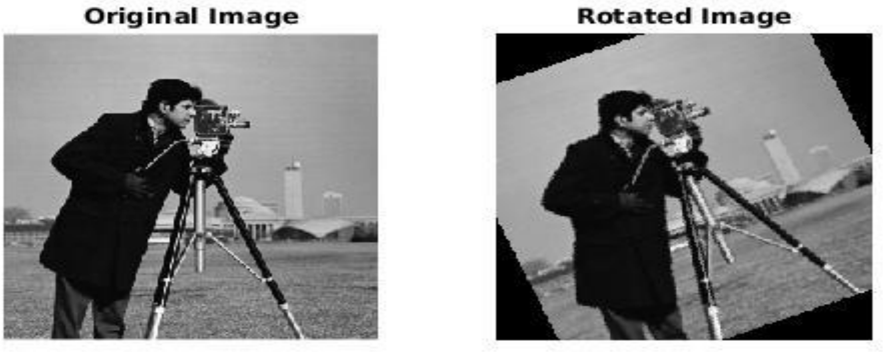
PARMAR hardiksinh

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} x - x_c \\ y - y_c \end{pmatrix} + \begin{pmatrix} x_c \\ y_c \end{pmatrix}$$

For rotation my function is: function [O] = myRotation(I, angle, method)

Here , I - Input Image
angle - Angle of Rotation = (-45)value
method - Interpolation method {"nearest", "bilinear"}

Here the below image is the ('cameraman.tif') image and its output (Rotatedimage image).



Original Image

Rotated Image

2.1.3 Transformation Nearest Neighbour:

Here first we have to take the pixels surrounding neighbour's intensity values and then calculate average of those intensity values of the pixels.

Here the below image is the ('cameraman.tif') image and its output (Nearest Neighbour image).

2.1.4 Bilinear Transformation :

Here first we have to take the pixels surrounding neighbour's intensity values (up, down, left , right) multiply the pixel value with the next pixel in the same row and then multiply next pixel with previous pixel intensity (For the top neighbour's) and repeat for bottom neighbour's then compute the same for newly obtain the neighbour's value.

Here the below image is the ('cameraman.tif') image and its output (Bilinear Transformation image).

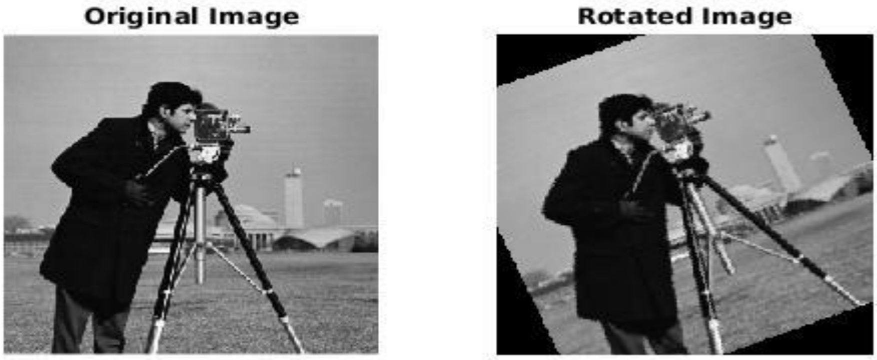


IMAGE PROCESSING MATLAB_2

PARMAR hardiksinh

3. Projective transformation :

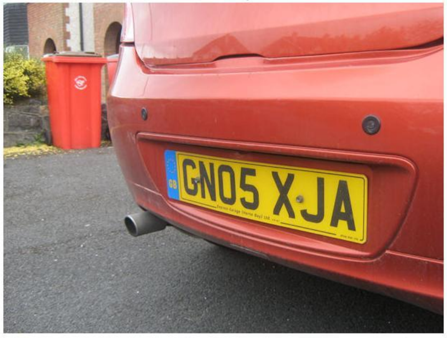
The idea of a projective plane can be applied to dimensional space in order to define projective n-space. Of particular interest is projective 3-space. Transformations within and between projective spaces are called *projectivities*and are the fundamental concern of projective geometry.

Projective transforms are useful for registering or aligning images or more generally scenes which can be approximated as flat fromdifferent viewpoints.

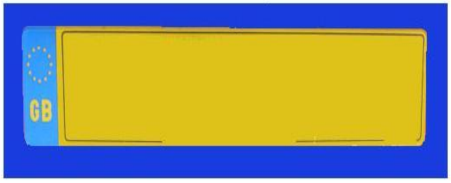
For Projective transformation my function is: function [Projected_image] = projectiveTransform(I , ref)
Here I - Input image
ref - Reference image

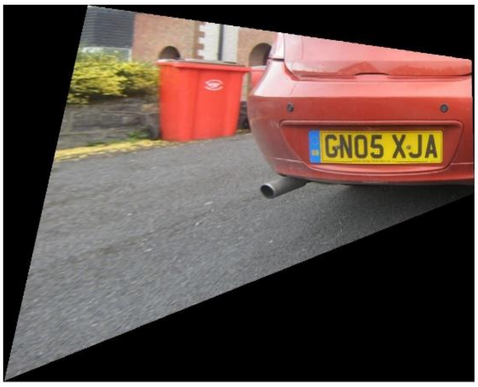
Here the below image is the 1. Input image , 2. Reference image, 3. Projected image 4. Cropped image after transformation.

Input Image



Reference Image





Croped Image (After Transformation)

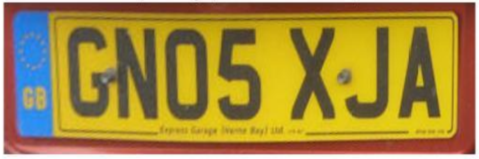


IMAGE PROCESSING MATLAB_2

PARMAR hardiksinh

4. Procrustes analysis :

Procrustes analysis is a form of statistical shape analysis used to analyse the distribution of a set of shapes. To compare the shapes of two or more objects, the objects must be first optimally "superimposed". Procrustes superimposition (PS) is performed by optimally translating, rotating and uniformly scaling the objects.

For Procrustes analysis my function is : function [O] = myProcrustes(X,Y)
Here X - Input points set
Y - Reference points set
O - Fitted points set

1.Translation :

In a translation, every point of the object must be moved in the same direction and for the same distance.

$$\mathrm{x}_i \to \mathrm{x}_i - \bar{\mathrm{x}},$$

2. Scaling :

Resize of the input image with respect with base Image.

$$X \to SI \text{ with } S = sI \text{ and } s = \frac{\sum_{i=1}^{N} \mathrm{x}_i'^{T} \mathrm{x}_i}{\sum_{i=1}^{N} \mathrm{x}_i^{T} \mathrm{x}_i}$$

3. Rotation:

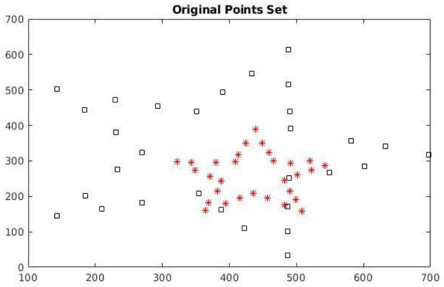
A rotation is a circular movement of an object around a center (or point) of rotation.

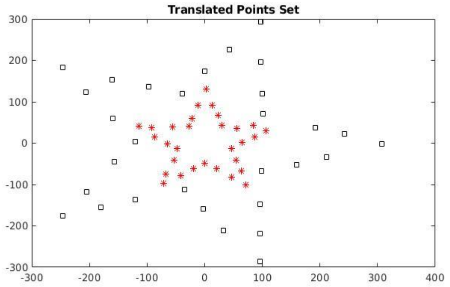
- form the product of the coordinate matrices XY^T
- calculate its SVD as $XY^T = USV^T$
- calculate the rotation matrix as $R = VU^T$

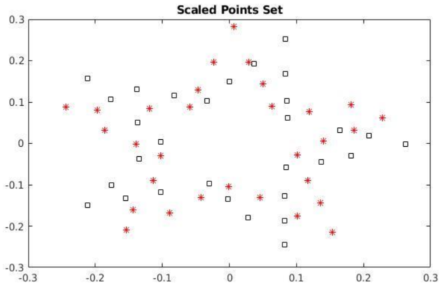
Here the below image is the output images of Procrustes analysis.

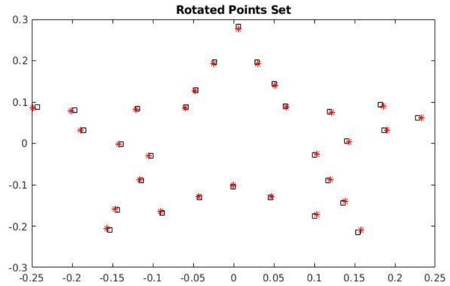
IMAGE PROCESSING MATLAB_2

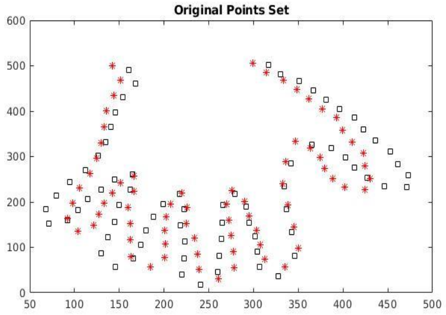
PARMAR hardiksinh

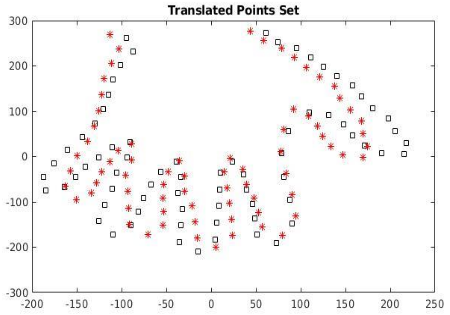

Original Points Set

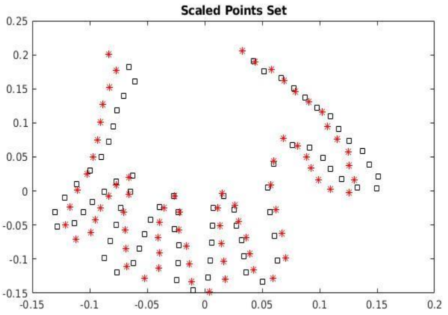

Translated Points Set

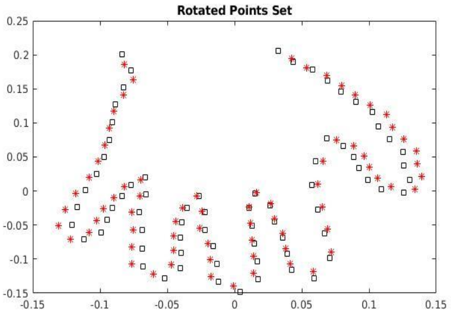

Scaled Points Set


Rotated Points Set


Original Points Set


Translated Points Set


Scaled Points Set


Rotated Points Set