**Image processing using openCV: Rotation**

 Rotation is a process of rotating an image with respect to an angle.

|  |  |
| --- | --- |
| cos A | -sin A |
| sin A | cos A |

Transformation matrix for rotation in 2D is given by, R =

where A is the angle of rotation.

To obtain transformation matrix for rotation, we use openCV function

cv2.getRotationMatrix2D(center, angle, scale): the function returns the transformation matrix for scaled rotation with specified center of rotation. It has three arguments:

1. center : it specifies the center of rotation i.e., the point about which rotation takes place.
2. angle: it specifies the angle of rotation.
3. scale: it specifies the scaling factor of the resultant figure.

Transformation matrix that openCV use to transform an image for scaled rotation with specified center of rotation is given by:

\begin{bmatrix} \alpha &  \beta & (1- \alpha )  \cdot center.x -  \beta \cdot center.y \\ - \beta &  \alpha &  \beta \cdot center.x + (1- \alpha )  \cdot center.y \end{bmatrix}

Where,

\begin{array}{l} \alpha =  scale \cdot \cos \theta , \\ \beta =  scale \cdot \sin \theta \end{array}

After obtaining the transformation matrix, we can use the cv2.warpAffine() function to obtain the required transformed image.

Python code:

import cv2

img = cv2.imread('picture.jpg',1);

rows,cols,channels = img.shape;

print('shape:',img.shape);

#shape: (174, 290, 3)

M = cv2.getRotationMatrix2D((cols/2,rows/2), 45, 0.75);

#center = (145, 87)

#angle of rotation: 45 degree

#scale= 0.75

output\_img = cv2.warpAffine(img, M, (cols,rows));

cv2.imshow('Original', img);

cv2.imshow('Output Image',output\_img);

#wait for 10 seconds

cv2.waitKey(10000);

cv2.destroyAllWindows();

Output:

