# Image processing miniproject – Rotation

## Algorithm



## Code

import cv2

import math

import numpy as np

class rotator:

angle = 40.0

x = 330

y = 330

radians = float(angle\*(math.pi/180))

img = cv2.imread('lena.jpg',0)

width,height = img.shape

def showImg(name, self):

cv2.imshow(name, self.img)

self.img = np.pad(self.img, (self.height) ,'constant', constant\_values=0)

self.width,self.height = self.img.shape

def printWH(self):

print(self.width)

print(self.height)

def rotate(self):

emptyF = np.zeros((self.width,self.height),dtype="uint8")

emptyB = np.zeros((self.width,self.height),dtype="uint8")

emptyBB = np.zeros((self.width,self.height),dtype="uint8")

for i in range(self.width):

for j in range(self.height):

#forward mapping

xf = int((i-self.x)\*math.cos(self.radians)-(j-self.y)\*math.sin(self.radians))+self.x

yf = int((i-self.x)\*math.sin(self.radians)+(j-self.y)\*math.cos(self.radians))+self.x

#backward map of the forward mapped picture

xfb = (xf-self.x)\*math.cos(self.radians)+(yf-self.y)\*math.sin(self.radians)+self.x

yfb = -(xf-self.x)\*math.sin(self.radians)+(yf-self.y)\*math.cos(self.radians)+self.x

#backward map of the original image

xb = int((i-self.x)\*math.cos(self.radians)+(j-self.y)\*math.sin(self.radians)+self.x)

yb = int(-((i-self.x)\*math.sin(self.radians))+((j-self.y)\*math.cos(self.radians))+self.x)

if xf < self.width and yf < self.height and xf>0 and yf > 0:

emptyF[i,j] = self.img[int(xf),int(yf)]

else:

pass

if xfb <= self.width and yfb <= self.height and xfb>=0 and yfb >= 0:

emptyB[i,j] = self.img[int(xfb),int(yfb)]

else:

pass

if xb < self.width and yb < self.height and xb>=0 and yb >= 0:

emptyBB[i,j] = self.img[int(xb),int(yb)]

else:

pass

cv2.imshow('Forward', emptyF)

cv2.imshow('Forward Backward', emptyB)

cv2.imshow('Backward', emptyBB)

def main():

rotator.showImg('normal', rotator)

rotator.printWH(rotator)

rotator.rotate(rotator)

cv2.waitKey(0)

cv2.destroyAllWindows

if \_\_name\_\_ == '\_\_main\_\_':

main()

## Explanation of code

## Documentation of program