Assignment 1

2017113547 이정근

소스코드 : rotation\_bw.py

import cv2

import numpy as np

def transform(img, angle): # forward transformation

height, width = img.shape

result = np.zeros((height, width), np.uint8) # result image

affine = np.array([[np.cos(np.radians(angle)), -np.sin(np.radians(angle)), 0],

[np.sin(np.radians(angle)), np.cos(np.radians(angle)), 0],

[0, 0, 1]]) # Affine transformation matrix

for x in range(width):

for y in range(height):

p = affine.dot(np.array([x, y, 1]))

xp = int(p[0])

yp = int(p[1])

if 0 <= yp < height and 0 <= xp < width:

result[yp, xp] = img[y, x]

return result

def backward\_map(in\_image, out\_image, angle):

height, width = out\_image.shape

result = np.zeros((height, width), np.uint8)

affine = np.array([[np.cos(np.radians(angle)), -np.sin(np.radians(angle)), 0],

[np.sin(np.radians(angle)), np.cos(np.radians(angle)), 0],

[0, 0, 1]]) # Affine transformation matrix

inverse\_affine = np.linalg.inv(affine)

for x in range(width):

for y in range(height):

if out\_image[y][x] == 0:

p = inverse\_affine.dot(np.array([x,y,1]))

xp = int(p[0])

yp = int(p[1])

if 0 <= yp < height and 0 <= xp < width:

result[y, x] = in\_image[yp][xp]

else:

result[y][x] = out\_image[y][x]

return result

in\_image = cv2.imread('dgu\_gray.png', 0) # img2numpy

out\_image = transform(in\_image, 20)

backward\_mapped\_image = backward\_map(in\_image, out\_image, 20)

cv2.imshow('Input Image', in\_image)

cv2.imshow('Result Image', out\_image)

cv2.imshow('Backward Mapped Image', backward\_mapped\_image)

cv2.imwrite('dgu\_gray\_rotate.png', out\_image) # save result img

cv2.imwrite('dgu\_gray\_backward\_mapped.png', backward\_mapped\_image)

cv2.waitKey()

결과 사진

dgu\_gray\_backward\_mapped.png



dgu\_gray.png



dgu\_gray\_rotate.png

