Readme.txt 1716009 백지희

1. Image Rotation

- Purpose of this code

"rotate\_skeleton\_v2.cpp" is a code that implement image geometric transformation; specifically rigid (euclidean) transformation. The rotation does not always produce an integer pixel location. Thus, the interpolation is needed. In this code, there are two ways to implement an interpolation. One is a nearest neighbor interpolation and the other one is a bilinear interpolation.

-Environment

Window

OpenCV 2.4.13.6

Microsoft Visual studio 2019

- How to run this code

download skeleton code-Lec02.zip

cl rotate\_skeleton\_v2.cpp

- How to adjust parameters

to change input image: edit "input image" at line number 14

input = imread("input\_image");

to adjust the angle of rotation or an interpolation method: edit "angle", "interpolation method" at line number 26

rotated = myrotate<Vec3b>(input, angle, interpolation method);

Readme.txt 1716009 백지희

2. Image Stitching using Affine Transformation

- Purpose of this code

“stitching\_skeleton.cpp” is a code that implement image stitching using affine transformation. When two different images include same part in different angle, by stitching that two images we can get a wider range image. For getting affine matrix x, we can use Mx = b. After calculating affine matrix x, in this code, we apply it with right image to affine transform. Floating problem that occurs in transformation process is solved by bilinear interpolation.

-Environment

window

OpenCV 2.4.13.6

Microsoft Visual studio 2019

- How to run this code

download skeleton code-Lec02.zip

cl stitching\_skeleton.cpp

- How to adjust parameters

to change input image: edit "input image1"and “input image2” at line number 15, 16.

I1 = imread(“input image1”);

I2 = imread(“input image2”);

to adjust corresponding pixel to value that you calculate: edit ptl\_x, ptl\_y, ptr\_x, ptr\_y at line number 27~30

int ptl\_x[ ] = { , , , , };

int ptl\_y[ ] = { , , , , };

int ptr\_x[ ] = { , , , , };

int ptr\_y[ ] = { , , , , };