Digital Image Processing

LabWork #CV4

Professor : Jin-Woo Jung

* Using the given CFilter class, design a program that can process the following :

1.

1. Convert an image to a gray scale image (Input : jenny.jpg, rice.png)

|  |  |  |  |
| --- | --- | --- | --- |
| **Input** | | **Output** | |
|  |  |  |  |
| code | | | |
|  | | | |

1. Extract Sobel edge using GS\_sobel\_edge\_Sobel() or cv::Sobel() method

|  |  |  |  |
| --- | --- | --- | --- |
| **Input** | | **Output** | |
|  |  |  |  |
| code | | | |
|  | | | |

1. Extract Laplacian edge using GS\_laplacian\_edge\_Laplacian or cv::Laplacian() method

|  |  |  |  |
| --- | --- | --- | --- |
| **Input** | | **Output** | |
|  |  |  |  |
| code | | | |
|  | | | |

1. Extract Canny edge using GS\_canny\_edge\_Canny or cv::Canny() method

|  |  |  |  |
| --- | --- | --- | --- |
| **Input** | | **Output** | |
|  |  |  |  |
| code | | | |
|  | | | |

1. Extract Laplacian of Gaussian (sigma value = 2)

|  |  |  |  |
| --- | --- | --- | --- |
| **Input** | | **Output** | |
|  |  |  |  |
| code | | | |
|  | | | |

1. Discuss on the best edge detector. I.e., what is the best edge detector among Sobel, Laplacian, Canny and Laplacian of Gaussian? and why?

* Using the given CGeometry class, design a program that can process the following : Use only briefcase.jpg as Input image for all problems

1.

1. (Use the center of given image as center for rotation and scaling)
   1. Rotate the image by 45 degrees using GS\_affineRotate (clockwise)

|  |  |
| --- | --- |
| **Input** | **Output** |
|  |  |
| code | |
|  | |

* 1. Enlarge the image 1.5 times : A image

|  |  |
| --- | --- |
| **Input** | **Output** |
|  |  |

| code |
| --- |
|  |

1. (Use the center of given image as center for rotation and scaling)
   1. Enlarge the image 1.5 times

|  |  |
| --- | --- |
| **Input** | **Output** |
|  |  |
| code | |
|  | |

* 1. Rotate the image by 45 degrees using GS\_affineRotate (The ) : B image

|  |  |
| --- | --- |
| **Input** | **Output** |
|  |  |
| code | |
|  | |

1. Subtract B image from A image and check the result

(If two images are exactly same, the result would be black image)

|  |  |  |
| --- | --- | --- |
| **Input** |  | **Output** |
|  |  |  |
| code | | |
|  | | |

2.

1. (Use the center of given image as center for rotation and scaling)
   1. Rotate the image by 30 degrees (clockwise)

|  |  |
| --- | --- |
| **Input** | **Output** |
|  |  |
| code | |
|  | |

* 1. Repeat rotating the result image again 11 more times : A image

|  |  |
| --- | --- |
| **Input** | **Output** |
|  |  |
| code | |
|  | |

1. Subtract A image from the original image and check the result

(If two images are exactly same, the result would be black image)

|  |  |  |
| --- | --- | --- |
| **Input** | | **Output** |
|  |  |  |
| code | | |
|  | | |

1. Repeat the first step (1) of Prob.2 more than five times and then repeat (2) again.

(Input : briefcase.jpg)

|  |  |
| --- | --- |
| **Input** | **Output** |
|  |  |
| code | |
|  | |

1. Explain why the result of the step(3) B blurred.