Digital Image Processing

LabWork #CV6

Professor : Jin-Woo Jung

* Using the given CMorph, CEdge, CImageAnalysis classes
* Design a program that can perform the followings

|  |
| --- |
| 1. Binarization and Labeling : (1) + (2) + (3) + (4) + (5) 2. Finding Edge: [ (1) + (2) + (3) + (4) + (6) ] compare to [ (1) + (2) + (3) + (4) + (7) + (8) + (9) ] 3. Hough Transform: (1) + (2) + (3) + (4) + (10) 4. Compare the outputs of Finding Edge and Hough Transform |

<Steps for the implementation>

Use stuff\_color.jpg and rice.png after convert to grayscale image.

1. Erosion using erode function (kernel size 11x11 and MORPH\_ELLIPSE)
2. Dilation using dilate function step1 output
3. subtract the step2 output from the original image (gray)
4. Otzu binarization to make the binarized image for step3 output (threshold value = 0, THRESH\_BINARY | THRESH\_OTSU)
5. Labeling using GS\_findContours function for step4 output
6. Find edges of step4 output using GS\_canny\_edge\_Canny
7. Find edges of step4 output using GS\_sobel\_edge
8. Closing using GS\_closing function for step7 output (MORPH\_RECT)
9. Opening using GS\_opening function for step8 output (MORPH\_RECT)
10. Hough transform of step4 output using GS\_basicHoughTransformGray to find lines
11. Binarization and Labeling : (1) + (2) + (3) + (4) + (5)
    1. Erosion using erode function (kernel size 11x11 and MORPH\_ELLIPSE)

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

* 1. Dilation using dilate function step1 output

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

* 1. subtract the step2 output from the original image (gray)

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

* 1. Otzu binarization to make the binarized image for step3 output (threshold value = 0, THRESH\_BINARY | THRESH\_OTSU)

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

* 1. Labeling using GS\_findContours function for step4 output

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

| code |
| --- |
|  |

1. Finding Edge: [ (1) + (2) + (3) + (4) + (6) ] compare to [ (1) + (2) + (3) + (4) + (7) + (8) + (9) ]
   1. Find edges of step4 output using GS\_canny\_edge\_Canny

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

* 1. Find edges of step4 output using GS\_sobel\_edge

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

* 1. Closing using GS\_closing function for step7 output (MORPH\_RECT)

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

* 1. Opening using GS\_opening function for step8 output (MORPH\_RECT)

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

| code |
| --- |
|  |

* 비교내용 작성

1. Hough Transform: (1) + (2) + (3) + (4) + (10)
   1. Hough transform of step4 output using GS\_basicHoughTransformGray to find lines

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

| code |
| --- |
|  |

1. Compare the outputs of Finding Edge and Hough Transform