

**과목명 | 디지털영상처리**

**실습 | 9**

**담당교수 | 정진우**

**학과 | 컴퓨터공학과**

**학년 | 3**

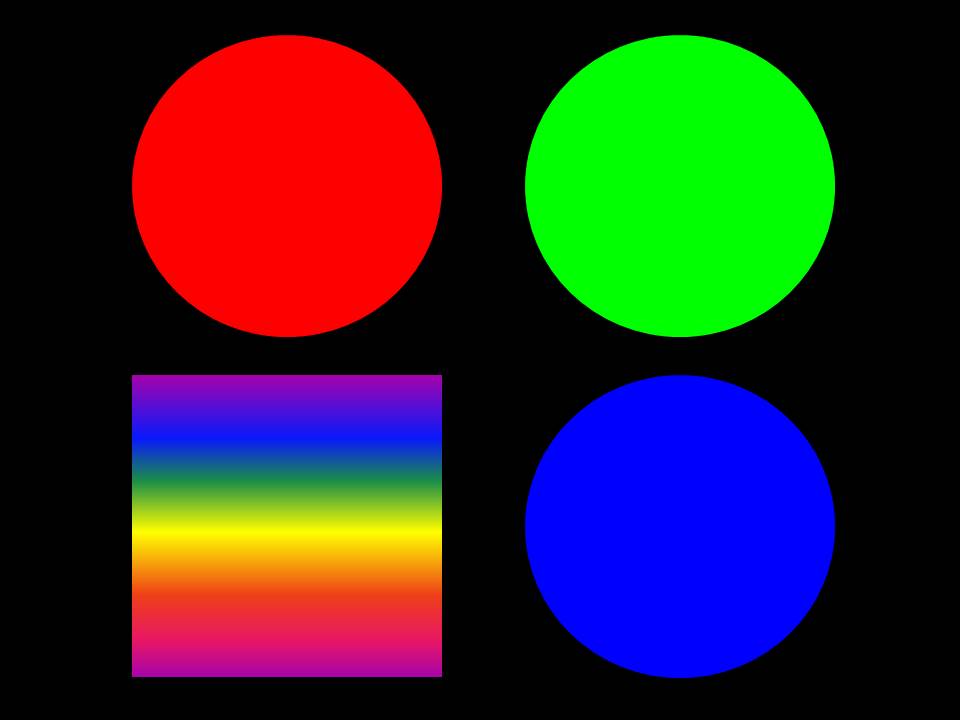
**학번 | 2010112406**

**제출일자 | 2016. 6. 2**

**이름 | 박도령**

**제출일 | 2016. 3. 31**

**실습에 사용한 이미지)**

** **

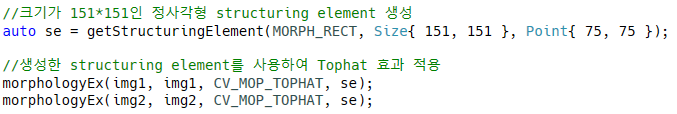
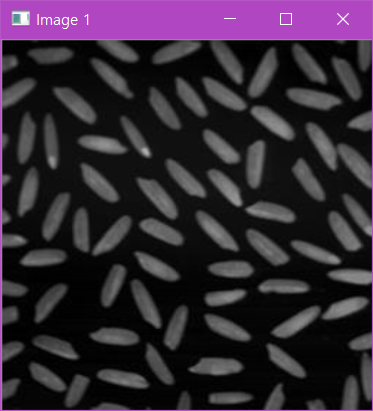
**tophat.PNG stuff\_color.jpg**

**실습 1)**

♋ **동작 순서 및 원리**

**1. Top-Hat operation using opencv functionfor input image to reduce the effect of non-uniform light intensity.**

**Top-hat 효과를 적용하기 위해 Structuring element가 필요하다. Opencv에서 지원하는 함수 getStructuringElement를 사용하여 사이즈가 151 \* 151이고 기준점의 좌표가 [75, 75]인 Structuring element를 생성할 수 있다. 이렇게 충분한 크기의 Structuring Element를 사용하는 이유는 관심 대상의 크기보다 크기 때문에 영상을 Binarization 하기 전에 밝기 값이 고르지 못한 배경을 0으로 단일하게 잡아줄 수 있기 때문이다. Top-hat 효과를 적용한 결과로써 단일한 밝기 값(0)을 가진 배경을 가지게 되었음을 확인할 수 있다. 코드는 다음과 같다.**



**2. Otzu binarization to make the binarized image for step1 output.**

**Opencv에서 제공하는 함수 threshold를 사용하여 Otsu method로 영상을 Binarization 한다. 코드는 다음과 같다.**

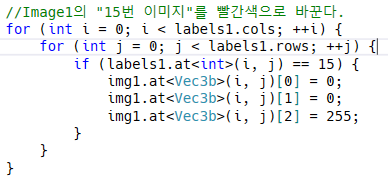
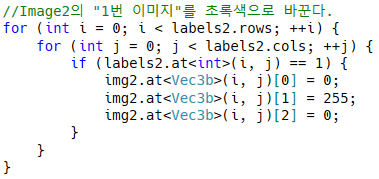


**3. Make Labeling for step 2 output.**

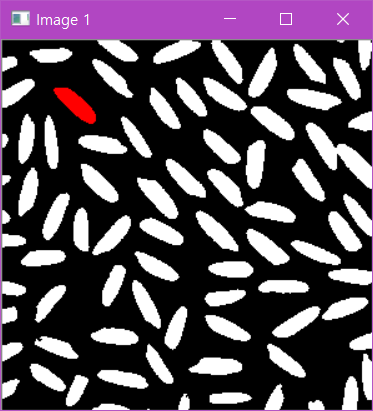
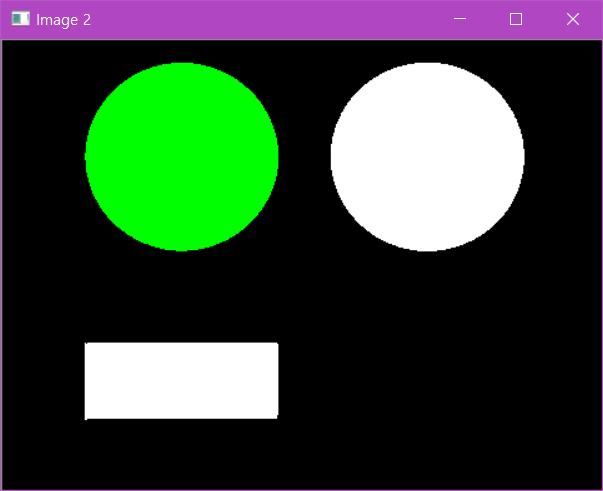
**Labeling은 이치화 영상의 나누어진 이미지들에 번호를 부여하는 것을 뜻한다. 픽셀 구역마다 해당하는 번호를 대입하기 위해서는 또 다른 매트릭스가 필요하다. 따라서 labels 라는 매트릭스 변수를 선언하고 Opencv에서 제공하는 함수 connectedComponents를 사용하여 번호가 매겨진 픽셀 위치들을 해당 변수에 저장하였다.**



**그 뒤에 Labeling을 적용했다는 것을 확인하기 위해 Labeling 된 하나의 이미지의 색깔을 변경해 보았다. 위의 이미지에서 번호 15가 매겨진 이미지를 빨간색으로 변경해 보았으며, 또 다른 이미지(stuff\_color.jpg)의 번호 1이 매겨진 이미지를 초록색으로 변경해 보았다. 코드는 다음과 같다.**

**♋ 결과 이미지 (ReportCode10-1(1).jpg, ReportCode10-1(2).jpg)**

**♋ 설명**

**ReportCode10-1(1).jpg 는 총 81개의 이미지 구역이 나누어졌으며 ReportCode10-2(2).jpg는 총 4개의 이미지 구역이 나누어져 Labeling 되었다.**

**실습 2)**

♋ **동작 순서 및 원리**

**Step 1, 2 are same with others of practice 1.**

**과정은 실습 1의 1,2와 같다.**

**4. Find edges with using Canny edge detection.**

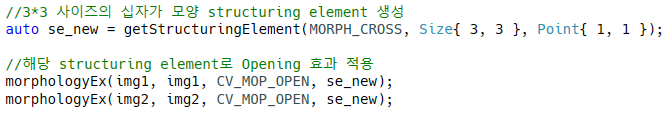
**Opencv에서 제공하는 함수 Canny를 사용하여 Otsu Binarization 한 영상의 Edge를 찾아낸다. 코드는 다음과 같다.**



**5. Adjust opening effect for step4 output.**

**Opening은 영상을 structuring element로 erode한 뒤 dilate 한 것이고, Closing은 영상을 dilate 한 뒤 erode한 것이다. 앞서 Canny detection으로 얻은 Canny edge가 너무 가늘어서 영상의 erosion으로부터 보호하기 위하여 3\*3 사이즈의 십자가 모양 structuring element를 사용하였으며, 영상을 4배로 확대시켜야만 했다. 그렇게 하지 않으면 영상이 erosion되었을 때 모두 지워져서 결국 아무것도 남지 않는 영상이 될 것이다. Opencv에서 제공하는 함수 morphologyEx를 사용하여 한번에 영상에 Opening 효과를 줄 수 있다. 코드는 다음과 같다.**



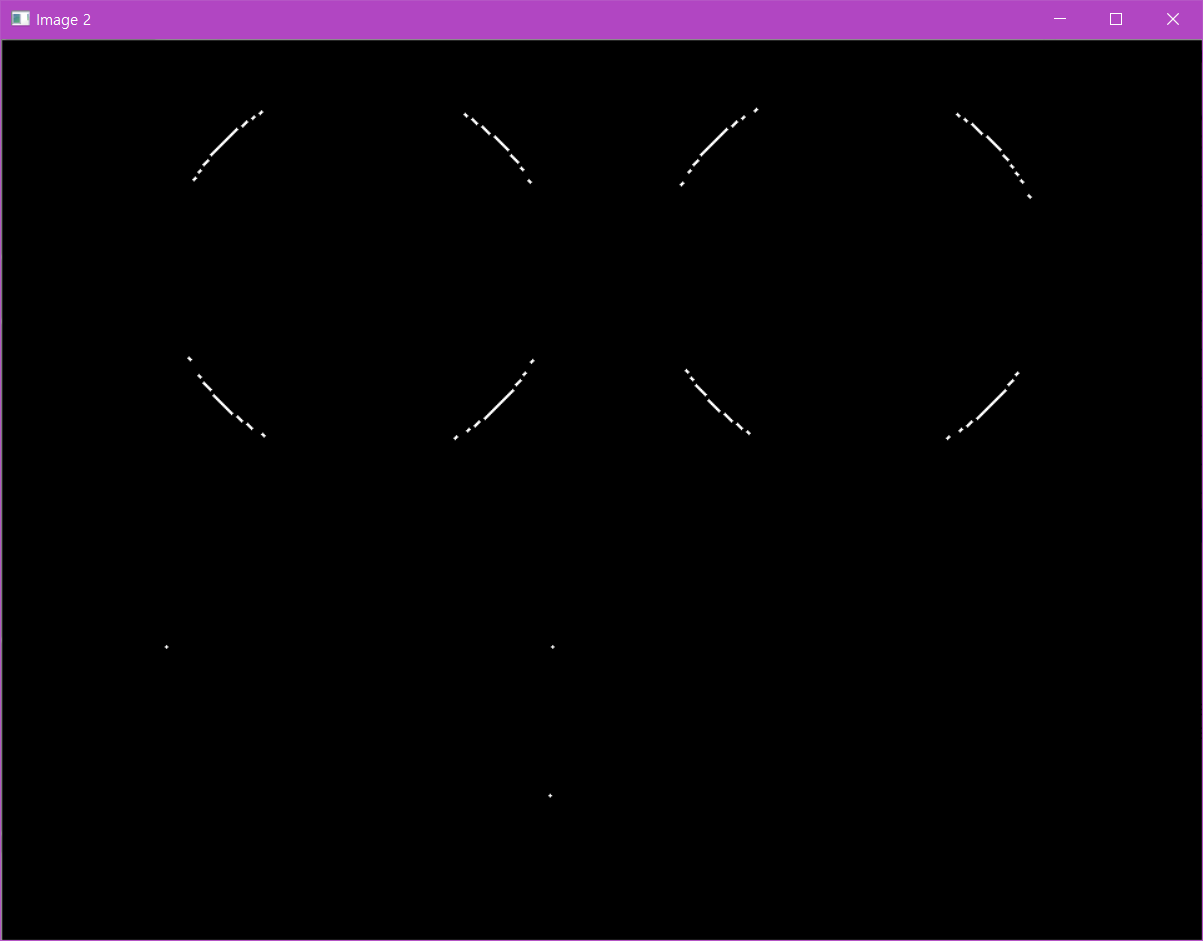


**6. Adjust closing effect for step5 output.**

**위와 마찬가지로 Opencv에서 제공하는 함수 morphologyEx를 사용하여 한번에 Closing 효과를 줄 수 있다. 코드는 다음과 같다.**



**♋ 결과 이미지(ReportCode10-2(1).jpg, ReportCode10-2(2).jpg)**

**실습 3)**

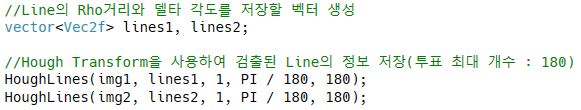
♋ **동작 순서 및 원리**

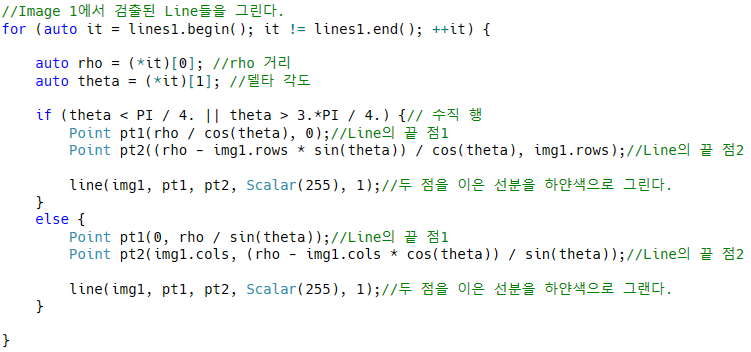
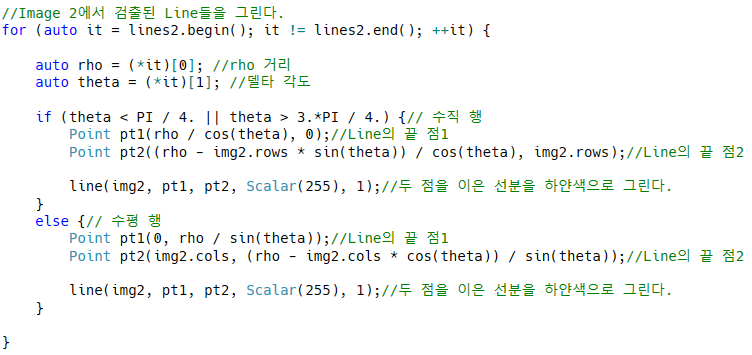
**Step 1, 2 are same with others of practice 1.**

**과정은 실습 1의 1,2와 같다.**

**3. Adjust Hough transform for step2 output to find lines.**

**영상의 선을 검출하기 위해서는 선의 정보(선의 rho 거리, Delta 각도)를 저장하기 위한 저장소 벡터가 필요하다. 따라서 벡터 lines를 정의하였다. 그 뒤에 Opencv에서 제공하는 함수 HoughLines를 사용하여 검출된 선의 정보를 해당 변수에 얻을 수 있다. rho 거리, Delta 각도를 이용하여 이미지 위에 하얀색 선을 그린다. 코드는 다음과 같다.**



♋ **결과 이미지(ReportCode10-3(1).jpg, ReportCode10-3(2).jpg)**

