

영상처리

과제 HW#6

학과 : 소프트웨어학과

학번 : 2016039016

이름 : 유수형



HW#6-1. 지문영상 개선하기

❖ 강의슬라이드 p.29에 있는 절차에 따라 모폴로지 기법을 적용하고, 그 결과 확인해보기



```
Mat erodeImage;  
erode(srcImage, erodeImage, crossKernel, anchor, iterations); // 침식 연산  
imshow("침식", erodeImage);
```

```
Mat erodil_Image;  
dilate(erodeImage, erodil_Image, crossKernel, anchor, iterations); // 침식 후  
imshow("침식 - 팽창 Image", erodil_Image);
```

```
Mat opendil_Image;  
dilate(erodil_Image, opendil_Image, crossKernel, anchor, iterations); // 열림  
imshow("침식 - 팽창 - 팽창 Image", opendil_Image);
```

```
Mat open_close_Image;  
erode(opendil_Image, open_close_Image, crossKernel, anchor, iterations); // 침식  
imshow("침식 - 팽창 - 팽창 - 침식 Image", open_close_Image);
```

순차적 연산

```
Mat openImage;  
morphologyEx(srcImage, openImage, MORPH_OPEN, crossKernel, anchor, iterations); // 열림 연산  
imshow("열림 image", openImage);
```

```
Mat openclose_Image;  
morphologyEx(openImage, openclose_Image, MORPH_CLOSE, crossKernel, anchor, iterations); // 닫힘 연산  
imshow("열림 - 닫힘 Image", openclose_Image);
```

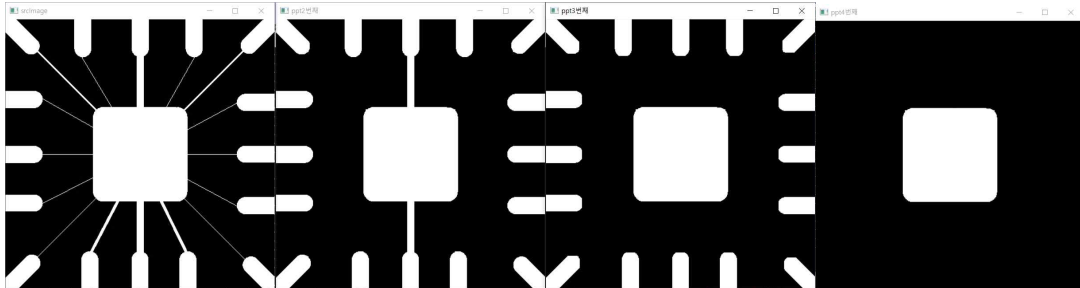
한번에 연산된 결과 도출

erode 함수 적용 후 dilate함수를 적용하여 열림 연산을 한 결과와 morphologyEx함수를 통해 열림 연산한 결과가 같았다.

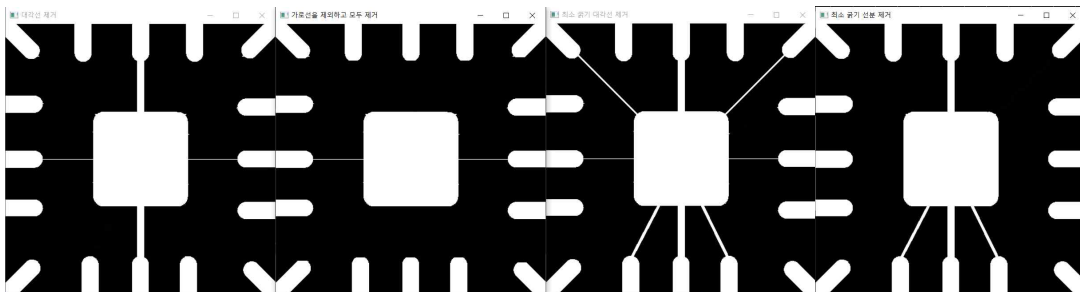
열림 이후 닫힘 연산에 대해 큰 효과가 보이진 않았지만 객체 내 빈 공간이 채워졌다.

HW#6-2. 원하는 선분 없애기

- 1) 강의슬라이드 p.10에 있는 절차에 따라 모폴로지 기법을 적용하고, 그 결과 확인해보기
- 2) 1의 것 외에 특정한 선분 없애보기



기존의 ppt에 있는 내용들을 실행한 결과



새로 추가한 내용들을 실행한 결과

1번부터 4번까지 순서대로 대각선 제거, 가로선을 제외하고 모두 제거, 최소 굵기 대각선 제거, 최소 굵기 선분 제거를 실행.

```
Size size1(3, 3); // 커널의 크기는 제거하려는 잡음보다 커야한다.  
Size size2(3, 1);  
Size size4(7, 7);  
Size size5(7, 1);  
Size size7(15, 15);  
Size size8(15, 1);  
Size size6(45, 45);
```

모폴로지 기법 실행에 들어가는 커널 변수 Size설정

HW#6-3. 노이즈라 생각되는 영역 지우기

강의슬라이드 p.46에 있는 절차에 따라 모폴로지 기법을 적용하고, 그 결과 확인해보기

```
Mat Kernel1 = getStructuringElement(MORPH_ELLIPSE, size); // 3 x 3 사이즈의 타원형 구조요소 생성 (반지름 1인 디스크 구조요소와 유사)
cout << "Kernel1=" << Kernel1 << endl;

Mat Kernel2 = getStructuringElement(MORPH_ELLIPSE, size2); // 5 x 5 사이즈의 타원형 구조요소 생성 (반지름 2인 디스크 구조요소와 유사)
cout << "Kernel2=" << Kernel2 << endl;

Mat Kernel3 = getStructuringElement(MORPH_ELLIPSE, size3); // 7 x 7 사이즈의 타원형 구조요소 생성 (반지름 3인 디스크 구조요소와 유사)
cout << "Kernel3=" << Kernel3 << endl;
```

타원형 구조 요소를 생성(size는 3x3, 5x5, 7x7의 디스크 구성요소로 구성)

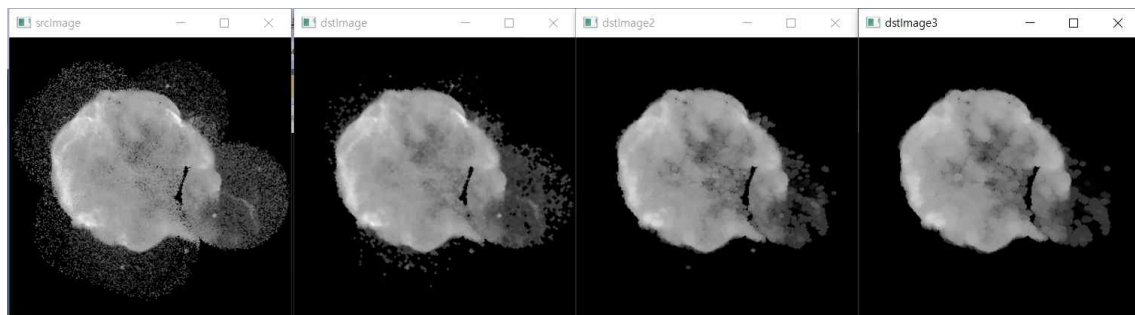
```
Mat dstImage;
morphologyEx(srcImage, dstImage, MORPH_OPEN, Kernel1, anchor, iterations);
morphologyEx(dstImage, dstImage, MORPH_CLOSE, Kernel1, anchor, iterations);
imshow("dstImage", dstImage);
```

```
Mat dstImage2;
morphologyEx(srcImage, dstImage2, MORPH_OPEN, Kernel2, anchor, iterations);
morphologyEx(dstImage2, dstImage2, MORPH_OPEN, Kernel2, anchor, iterations);
imshow("dstImage2", dstImage2);
```

```
Mat dstImage3;
morphologyEx(srcImage, dstImage3, MORPH_OPEN, Kernel3, anchor, iterations);
morphologyEx(dstImage3, dstImage3, MORPH_OPEN, Kernel3, anchor, iterations);
imshow("dstImage3", dstImage3);
```

이 후 각 커널에 대해 열림-닫힘 연산을 수행

실행 결과



HW#6-4. Edge 검출하기

- 1) 강의슬라이드 p.47에 있는 절차에 따라 모폴로지 기법을 적용하고, 그 결과 확인해보기
- 2) 1차 미분연산자와 2차 미분연산자를 통해 edge를 검출하고, 모폴로지 연산자를 사용했을 경우와 어떻게 다른지 비교해보기

```
Mat dstImage;  
morphologyEx(srcImage, dstImage, MORPH_GRADIENT, Kernel, anchor, iterations); // 평창 연산과 침식 연산 수행한 결과를 차 연산  
imshow("dstImage", dstImage);
```

MORPH_GRADIENT를 통한 차 연산을 수행

실행 결과

