

1. Line detection

- "img_building.tif" 파일을 이용하여, 아래의 문제에 맞도록 filter 를 설계하고, 그 결과영상을 출력하시오.
 - a. Vertical gradient component 를 추출하도록 3X3 Sobel filter 를 설계하고, 그 결과 영상을 출력하시오. 결과값의 음수값이 의미하는 바를 생각하시오.
 - b. Horizontal gradient component 를 추출하도록 3X3 Sobel filter 를 설계하고, 그 결과 영상을 출력하시오.
 - c. a)와 b)에서 얻어진 두 영상을 더하여, combined edge image 를 생성하시오.
 - d. 원본 영상(img_building.tif)에 5X5 average filter 를 적용한 영상을 이용하여, 다시 a, b, c 를 진행하고 위에서 얻어진 결과와 비교하시오.
 - e. d 에서 나온 결과에서 최대값의 10%인 값을 threshold 로 하여, binary image(1bit depth)를 만드시오.
 - f. d 에서 만든 average filter 가 적용된 영상을 가지고, LoG 를 이용하여, edge detection 을 실시하시오. 이때 edge() 함수를 사용하지 않고 fspecial() & conv2() 을 사용하고 threshold 를 통해 binary image 로 된 결과를 얻으시오.

2. Threshold (otsu method)

- "img_fingerprint.tif" 파일을 이용하여, otsu methode 를 이용한 thresholding 을 실행하시오.
 - a. img_fingerprint.tif 영상의 histogram 을 보이시오.imhist()
 - b. 다음 수식은 otsu methode 의 수식입니다. 이 수식을 이용하여 이미지를 입력변수로 받으면 k*값을 출력으로 내놓는 Result = otsuMethod(image)함수를 제작하시오. 제작한 함수를 이용하여 img_fingerprint.tif 의 k*값을 획득하시오.

p_i : 영상에서 i 의 값을 가지는 픽셀의 총수 / 영상의 총 픽셀수

$$P_1(k) = \sum_{i=0}^k p_i$$

$$m(k) = \sum_{i=0}^k ip_i$$

$$m_G = \sum_{i=0}^{L-1} ip_i$$

$$\sigma^2_B(k) = \frac{[m_G p_1(k) - m(k)]^2}{P_1(k)[1 - P_1(k)]}$$

$$\sigma^2_B(k^*) = \max_{0 \leq k \leq L-1} \sigma^2_B(k)$$

c. b 에서 획득한 k^* 값을 이용하여 Threshold 하여 화면에 출력하시오.

3. Moving average

- "img_shadedText.tif" 파일을 이용하여 아래의 결과를 출력하시오.

a. 2)에서 개발한 otsu method 방법을 통해 thresholding 을 실시하고, 결과 영상을 확인하여, otsu method 의 문제점에 대해 서술하시오.

b. 주어진 "movingaverage.m" 파일에 들어있는 함수를 이용하여, moving average 방법으로 segmentation 을 실시하고, 그 결과를 확인하시오.

hint) image_result = movingaverage(image , 20 , 0.5);

Tip. Matlab 에서 n/matrix

n : constant

가능한 방법 1 : n ./ matrix

*가능한 방법 2 : n * (matrix.^-1)*