

1. 제공된 'roundImage.png' 파일을 읽어와서 변수 ri에 저장하시오.

2. low pass filter

a) average filter를 15X15의 크기로 생성하여 변수 af에 저장하시오.

b) a)에서 생성한 필터를 이용하여 ri를 filtering 하거나 convolution 하여 변수 afr에 저장하시오.

Matlab에서 제공하는 filter2와 conv2의 차이

conv2는 수학적인 convolution을 수행함에 따라 필터와 이미지의 개념이 없고 $Y[n] = H[n]*X[n]$ 의 계산을 수행한 결과를 보여 주게 됩니다. filter2는 필터와 영상이 이미 정해져 있으며 영상의 각 픽셀에 대한 filter에 대응되는 연산을 수행합니다.

두 결과물을 비교 할 경우

1. conv2는 두 입력의 위치가 달라도 항상 같은 크기의 출력을 가지고 filter2는 이미지로 입력된 크기에 맞추어진 결과를 내놓습니다.

2. 방향성을 가진 필터를 사용할 경우 두 결과물의 방향은 180도 다르게 나오게 됩니다.

c) 변수 ri와 afr에 저장된 이미지를 하나의 figure에 출력하시오.

Tip. afr은 0과 255만 있는 영상이 아닙니다. histogram equalization을 해주세요.

d) 다음 명령어를 이용하여 ri와 afr의 profile을 확인하시오.

```
figure,plot(ri(128,:),'linewidth',3);
hold all;
plot(afr(129,:), 'linewidth',3, 'linestyle','--');
```

e) c)와 d)에서 이미지와 profile이 다른 이유가 무엇인가요?

3. filter generation

다음 명령어들을 실행 하여 각 필터를 생성하고 필터의 값을 확인하시오.

```
gf = fspecial('gaussian',3, .5) % ('gaussian',hsize,sigma)
lf = fspecial('laplacian', .5) % ('laplacian',alpha)
pf = fspecial('prewitt')
sf = fspecial('sobel')
```

4. 방향성 필터

'sobel'과 'prewitt'은 가장 기본적인 high pass filter 이며 방향성을 가지고 있습니다.

- a) 변수 ri에 저장된 이미지를 각각 sf와 pf로 필터링 이나 컨벌루션하여 sfr1과 pfr1으로 저장하시오.
- b) 같은 figure에 sfr1과 pfr1를 출력하시오.(histogram equalization을 해주세요.)
- c) 방향성을 가지지 않은 high pass filter 결과물을 얻어서 sfr2와 pfr2에 저장하시오.
(*'sobel'과 'prewitt'필터가 아닌 필터는 사용하지 않습니다.*)
- d) sfr2와 pfr2와의 차를 구하고 절대값을 취하여 변수 df 에 저장하시오.
- e) 같은 figure에 sfr2,pfr2와 df를 출력하시오.(histogram equalization을 해주세요.)
- f) e)에서 제시된 결과물을 바탕으로 'sobel'과 'prewitt'의 차이를 확인하시오.

Tip. LUT을 조정하며 차이를 쉽게 확인할 수 있습니다.

5. high pass filter 응용

- a) 제공된 'chest.png'파일을 열어 변수 chest에 저장하시오.
- b) 변수 chest에 저장된 이미지를 lf를 이용하여 필터링이나 컨벌루션 하여 변수 lfChest에 저장하시오.
- c) lfChest lf new_chest . .
- d) 하나의 figure에 chest과 lfChest, new_chest을 출력하고 비교하시오.