

# 2020-2학기 임베디드신호처리실습

## 결과보고서

Lab5. DCT



한국산업기술대학교  
KOREA POLYTECHNIC UNIVERSITY

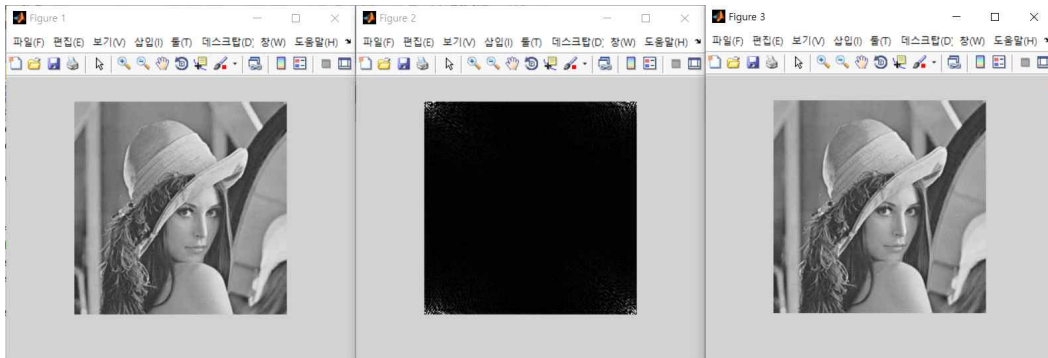
2016146041 정성균

2014146012 박동훈

### 3.1 2차원 DFT (FFT) 변환을 통한 이미지 처리

- **실습** 'lena.raw' 이미지의 2차원 DFT (FFT) 변환 크기값 (magnitude)을 출력하시오.  
(fft2(), myfun\_LoadImage(), imshow() 함수 이용. 주의: fft 결과값이 0 ~ 255 범위를 가지도록 적절한 상수 인자로 정규화 (normalization) 해주어야 함.)
- **실습** 위 2차원 DFT (FFT)된 이미지의 역변환 이미지를 출력하시오.  
(ifft2(), myfun\_LoadImage(), imshow() 함수 이용)
- **실습 DEMO** 위 실습 결과들을 그래프로 그리시오. (그림 3 참고)

```
1 - clear;
2 - clc;
3
4 - row =256;
5 - col =256;
6
7 - Z = myfun_LoadImage('lena.raw',row,col);
8 % LoadImage 함수를 이용하여 이미지 불러오기
9 - figure(1)
10 - imshow(Z)
11
12 - figure(2)
13 |
14 - Z1 = (1/(256*256))*fft2(Z); %1 / (256*256) 정규화 후 2차원 FFT
15
16 - imshow(real(Z1))
17
18 - figure(3)
19
20 - Z2 = uint8(256*256*(ifft2(Z1))); %역으로 IFFT, 256*256을 곱해줌
21 %원소값을 unsigned 8 bits 정수로 변환 출력
22 %0 ~ 255의 정수로 변환시키고 출력
23
24 - imshow(Z2)
25
26
```



DFT/FFT의 경우 실수부, 허수부 부분을 연산한다. Row, Col 부분이 각각  $N$ 이라서  $1/N^2$ 로 정규화를 시킨다. 그리고 DCT/FFT의 경우에는 주파수 도메인으로 변환한 것을 보았을 때 4개의 모서리 부분에 흰색 부분(저주파 성분)이 약간 보이는 것을 알 수 있다. 이것은 DFT/FFT가 실수, 허수 부분을 연산하게 되어 위상이 있기 때문에, 원점 대칭을 통해 4개의 모서리에서 저주파성분이 띄게 되는 것이다.

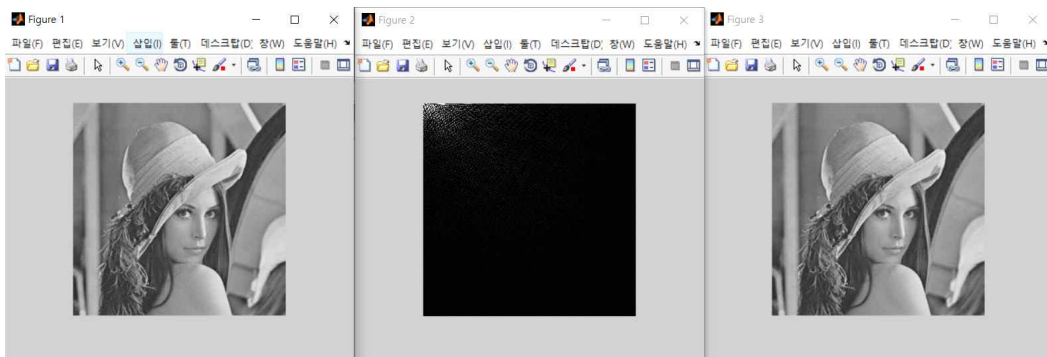
### 3.2 2차원 DCT 변환을 통한 이미지 처리

- 실습** 'lena.raw' 이미지의 2차원 DCT 변환 크기값 (magnitude)을 출력하십시오.  
 (dct2(), myfun\_LoadImage(), imshow() 함수 이용. 주의: dct 결과값이 0 ~ 255 범위를 가지도록 적절한 상수 인자로 정규화 (normalization) 해주어야 함.)
- 실습** 위 2차원 DCT된 이미지의 역변환 이미지를 출력하십시오.  
 (dct2(), myfun\_LoadImage(), imshow() 함수 이용)
- 실습 DEMO** 위 실습 3.2 결과들을 그래프로 그리고, 실습 3.1 DFT 결과와 비교하여 설명하십시오. (그림 4 참고)

```

1 - clear;
2 - clc;
3
4 - row =256;
5 - col =256;
6
7 - Z = myfun_LoadImage('lena.raw',row,col);
8 % LoadImage 함수를 이용하여 이미지 불러오기
9 - figure(1)
10 - imshow(Z)
11
12 - figure(2)
13
14 - Z1 = dct2(Z);
15 - Z2 = Z1/256; % 1 / 256 정규화 후 2차원 DCT
16 - imshow(Z2)
17
18 - figure(3)
19 - Z3 = uint8(256*(idct2(Z2))); %역으로 IDCT, 256을 곱해준다
20 %원소값을 unsigned 8 bits 정수로 변환 출력
21 %0 ~ 255의 정수로 변환시키고 출력
22 - imshow(Z3)
23

```

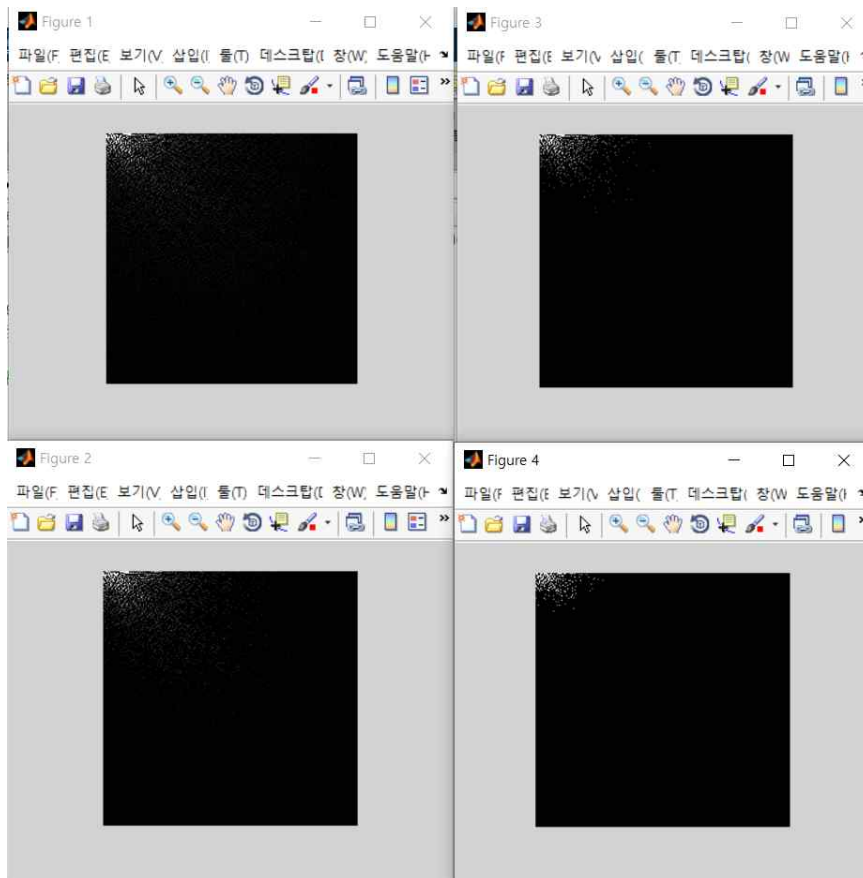


DCT의 경우 DFT/FFT와 다르게 실수부만 연산을 진행한다. DCT로 변환을 한 결과에서는 영상의 원점 부분에만 흰색 부분(저주파 성분)이 약간 보이는 것을 알 수 있는데 이것은 DCT가 DFT/FFT와 다르게 실수값만 가지고 변환을 해주기 때문에 영상에서의 원점 부분인 top - left 부분에서만 흰색 부분(저주파 성분)이 보이는 것이다.

### 3.3 DCT 변환을 이용한 이미지 압축

- (1) **실습** DCT 변환 크기값 (0 ~ 255 으로 정규화)에 대해서 문턱값 (threshold) [0.01, 0.1, 0.25, 0.5] 보다 작은 값은 0으로 처리하는 압축을 수행하고, 압축처리된 DCT 결과값을 IDCT 역변환을 수행해서 복원한 이미지를 출력하여 문턱값에 따라 비교하시오.

```
1 - clear;
2 - clc;
3
4 - row =256;
5 - col =256;
6
7 - Z = myfun_LoadImage('lena.raw',row,col);
8 % LoadImage 함수를 이용하여 이미지 불러오기
9 - Z1 = dct2(Z);
10 - Z2 = Z1/255; %정규화
11
12 - x = 0.01;
13 - x2 = 0.1;
14 - x3 =0.25;
15 - x4 = 0.5;
16
17 - figure(1)
18 - Z2(abs(Z2) < x) = 0; %문턱값에 의하여 압축
19 - imshow(Z2)
20
21 - figure(2)
22 - Z2(abs(Z2) < x2) = 0;
23 - imshow(Z2)
24
25 - figure(3)
26 - Z2(abs(Z2) < x3) = 0;
27 - imshow(Z2)
28
29 - figure(4)
30 - Z2(abs(Z2) < x4) = 0;
31 - imshow(Z2)
32
```



```

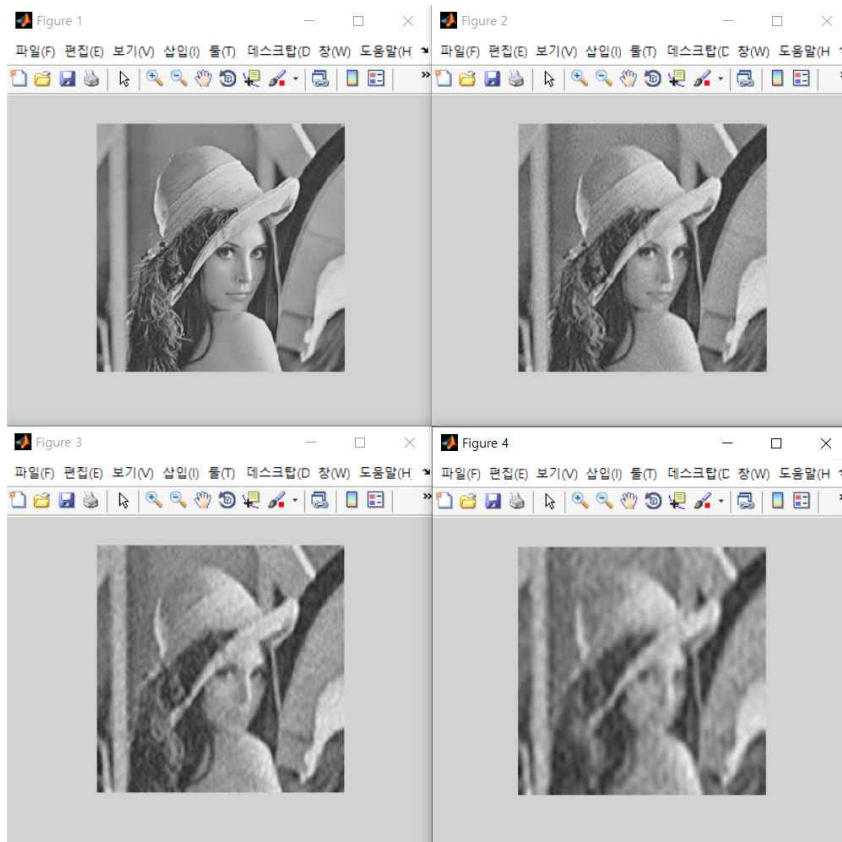
1 - clear;
2 - clc;
3
4 - row =256;
5 - col =256;
6
7 - Z = myfun_LoadImage('lena.raw',row,col);
8 - % LoadImage 함수를 이용하여 이미지 불러오기
9 - Z1 = dct2(Z);
10 - Z2 = Z1/255; % 정규화
11
12 - x = 0.01;
13 - x2 = 0.1;
14 - x3 =0.25;
15 - x4 = 0.5;
16
17 - figure(1)
18 - Z2(abs(Z2) < x) = 0; %문턱값에 의하여 압축
19 - K = idct2(Z2); %복원
20 - imshow(K)
21
22 - figure(2)
23 - Z2(abs(Z2) < x2) = 0;
24 - K = idct2(Z2);
25 - imshow(K)
26
27 - figure(3)
28 - Z2(abs(Z2) < x3) = 0;

```

```

29 - K = idct2(Z2);
30 - imshow(K)
31
32 - figure(4)
33 - Z2(abs(Z2) < x4) = 0;
34 - K = idct2(Z2);
35 - imshow(K)
36

```



문턱값이 커질수록 주파수 영역에서는 저주파 성분이 줄어든다. 그래서 표시할 수 있는 저주파의 영역이 줄어드는 것인데, 이 변화를 흰색 부분이 점점 없어지는 것을 보면 알 수 있을 것이다. 이에 따라서 시간 영역에서는 문턱값이 커질수록 0값이 많아지므로 해상도가 떨어진다.



- (2) **실습** 2차원 DCT 결과값에 대해서 문턱값 (threshold) 대비 압축률 관계를 구하시오. (x축: 문턱값, y축: 압축률)

- 문턱값 [0.001, 0.01, 0.05, 0.1, 0.25, 0.5] 에 대해서 압축률을 구하시오.
- 압축률은 다음으로 정의하시오.

$$\text{압축률} = \frac{\text{압축 처리 후 이미지에서의 DCT 값이 0이 아닌 픽셀의 개수}}{\text{원본 이미지에서의 DCT 값이 0이 아닌 픽셀의 개수}}$$

단, 원본 이미지에서의 DCT 값은 0.001 미만이면 0으로 간주한다.

- 압축률 값은 log 스케일로 그리시오.

- (2) **실습** 2차원 DCT 결과값에 대해서 문턱값 (threshold) 대비 정확도 관계를 구하시오. (x축: 문턱값, y축: RMSE 값)

- 문턱값 [0.001, 0.01, 0.05, 0.1, 0.25, 0.5] 에 대해서 압축률을 구하시오.
- 정확도는 RMSE (root mean square error)로 나타내시오.

$$\text{RMSE} = \sqrt{\frac{1}{N_1 N_2} \sum_{n_1=1}^{N_1} \sum_{n_2=1}^{N_2} |x(n_1, n_2) - \hat{x}(n_1, n_2)|^2},$$

여기서  $x(n_1, n_2)$ 는 원본 이미지의  $(n_1, n_2)$  픽셀에 해당하 gray-scale 값 ( $0 \sim 255$ ).  $\hat{x}(n_1, n_2)$ 는 압축 복원된 이미지의  $(n_1, n_2)$  픽셀에 해당하는 gray-scale 값( $0 \sim 255$ )이다.

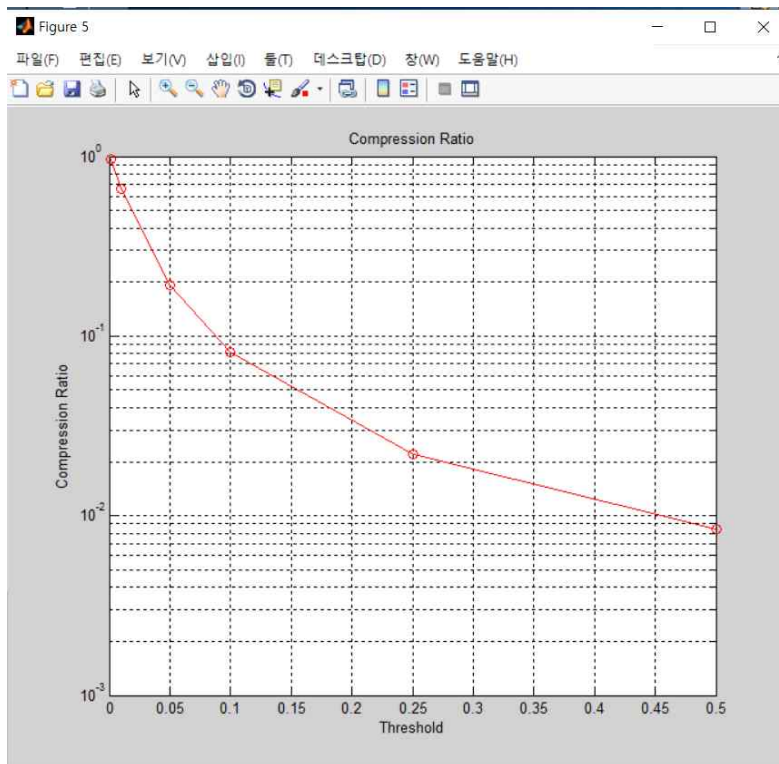
- (4) **실습 DEMO** 위 실습 결과들을 그래프로 나타내고 문턱값 (threshold)와 압축률 및 정확도와의 관계에 대해서 논하시오. (그림 5, 6, 7, 8 참고)

```

1 - clear;
2 - clc;
3
4 - X= myfun_LoadImage('lena.raw',256, 256);
5 - % LoadImage 함수를 이용하여 이미지 불러오기
6 - Z = (1/256)*dct2(X); % 1 / 256 정규화 후 2차원 DCT
7 - pix_num_OG = numel(Z); % Original 사진의 픽셀값 개수
8 -
9 - th = [0.001, 0.01, 0.05, 0.1, 0.25, 0.5]; % 문턱값 계산을 위한 배열
10 - pix_num = zeros(1,6); % pix_num 값을 담아주기 위하여 배열을 생성한다.
11 - Ratio = zeros(1,6); % Ratio 값을 담아주기 위하여 배열을 생성한다.
12 - for i = 1:6
13 -     Z(abs(Z) <= th(i)) = 0; % 문턱값에 의하여 압축
14 -     pix_num(i) = numel(find(abs(Z))); % Z의 절대값에서 0이 아닌
15 -     % 행렬의 원소를 찾아 그 개수를 세어 pix_num(i)에 저장하고
16 -     Ratio(i) = pix_num(i)/pix_num_OG; % 기본 픽셀값 개수를 나눈다.
17 - end
18 - figure(5);
19 - semilogy(th,Ratio,'o-r'); % 로그 Scale 그래프를 그려준다.
20 - axis([0, 0.5, 10^-3, 10^0]);
21 - xlabel('Threshold');
22 - ylabel('Compression Ratio');
23 - title('Compression Ratio');
24 - grid on
25

```

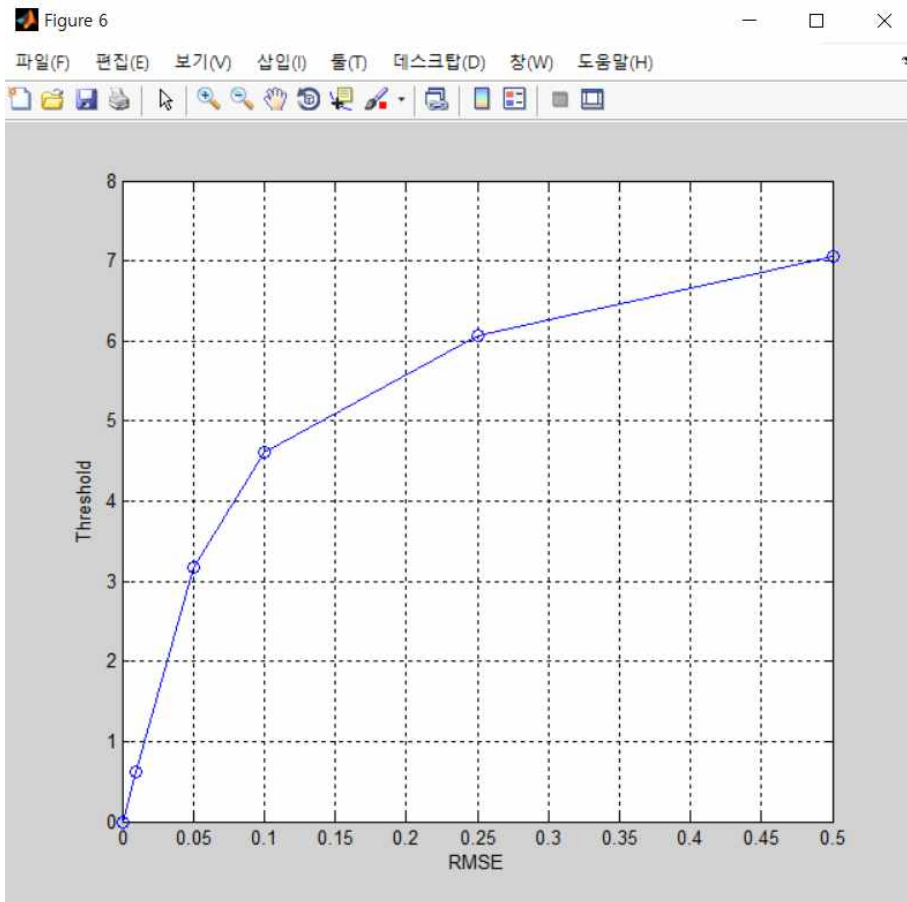




```

1
2 - clear; clc;
3 - X= myfun_LoadImage('lena.raw',256, 256);
4 - % LoadImage 함수를 이용하여 이미지 불러오기
5
6 - Z = (1/256)*dct2(X); % 1 / 256 정규화 후 2차원 DCT
7 - Z_inv = uint8(256*idct2(Z));
8 - % 역으로 IDCT하고 unsigned 8 bits 정수로 Z_inv를 만들어준다.
9 - th = [0.001, 0.01, 0.05, 0.1, 0.25, 0.5]; % 문턱값 계산을 위한 배열
10 - RMSE = zeros(1,6); % RMSE 값을 담아주기 위하여 배열을 생성한다.
11
12 - for i = 1:6
13 -     Z(abs(Z) <= th(i)) = 0; % 문턱값에 의하여 압축
14 -     Z_in(:, :, i) = uint8(256*idct2(Z)); % Z_in, Z_sub 라는 변수를 만들어서
15 -     % 3차원 배열로 잡아준다. 마지막 인수는 th와 맞춰주기 위함이다.
16 -     Z_sub(:, :, i) = (Z_inv - Z_in(:, :, i)).^2;
17 -     RMSE(i) = sqrt(1/(256*256)*(sum(sum(Z_sub(:, :, i)))))); % RMSE 연산
18 - end
19
20 - figure(6)
21 - plot (th,RMSE, '-ob')
22 - xlabel('RMSE');
23 - ylabel('Threshold');
24 - grid on;
25

```



### 실습 3.3 - (2) 문턱값 대비 압축률 관계

2차원 DCT 결과값에 대하여 문턱값 대비 압축률 관계는 문턱값(threshold)이 커질 수록 압축 처리 후 0이 아닌 픽셀 수는 점점 줄어든다. 압축률은 X 100을 해서 100%에 가까워야 원본과 가까운 것이고 점점 떨어질수록 압축이 더 많이 되고 있다는 뜻이다. 압축이 된 만큼 픽셀의 수는 점점 줄어들 것이고 이로 인해 저주파 성분이 많이 사라지게 되므로 그림의 해상도도 점점 떨어질 것이다.

압축률의 식을 보면 분모는 고정이고 분자만 바뀌고 있는데, 문턱값이 높아질수록 분자의 값(압축 처리 후 이미지에서의 DCT 값이 0이 아닌 픽셀의 개수)은 점점 줄어들어서 점점 감소하고 있는 그래프가 도출되는 것을 알 수 있다. 값을 담아주기 위해 pix\_num과 Ratio에 Zero padding을 하여 그릇을 만들고 for문을 돌려서 numel과 find 함수를 이용해서 0이 아닌 픽셀의 개수를 세어서 pix\_num에 넣어주고, Ratio에 하나씩 넣어준 pix\_num을 기본 픽셀 개수로 나누어 총 6개의 압축률 값을 만들어냈다.

### 실습 3.3 - (3) 문턱값 대비 정확도 관계

RMSE는 참값에서 측정값과의 거리를 나타내는 것으로, 정확도를 측정해주는 수치이다. 여기서  $x(n1,n2)$ 는 원본 이미지의 픽셀에 해당하는 gray-scale 값이고(참값),  $x'(n1,n2)$ 는 압축 복원된 이미지의  $(n1,n2)$  픽셀에 해당하는 gray-scale 값이다.(측정값) 이 둘의 오차제곱의 평균을 구하고 루트를 씌우는 방식으로 산출한다.

RMSE가 0이면 기본 이미지와 측정 이미지가 같다는 뜻이 되겠고, 문턱값이 증가할수록 이 수치는 점점 증가한다. 두 이미지의 차이가 문턱값 때문에 점점 증가하여 픽셀의 개수 차이가 커진다면(압축이 더 많이 된다면) RMSE 값은 점점 증가할 것이다. RMSE 값이 크다는 것은 즉 이미지의 해상도가 많이 떨어져 뭉개졌음을 뜻한다.

RMSE의 식은 for문 안에 있다. 문턱값에 의하여 압축을 한 후,  $Z_{in}$ 과  $Z_{sub}$  라는 변수를 만들어서 공식에 넣어주었다. 배열을 통해서 총 6개의 정확도 값을 만들어내었고 이를 그래프에 표시했다.