

1. 실제 밝기보다 더 어둡거나 밝게 느끼게 하려면 Sharpening 을 하여야 한다.

1차원 필터 기호로 설계를 할때 (필터크기 3)

$\begin{bmatrix} -1 & 3 & -1 \end{bmatrix}$ 으로 설계가 가능하다.

필터의 Value를 모두 합했을 때 1 이 되고

필터를 가한 이미지는 Sharp 해진다.

2. 자연 영상에서는 컬러, 음영이 연속적으로 변한다.

이 아날로그 영상을 기계어를 통해 영상으로 얻기 위해 디지털 영상으로 변화해야 하는데, Sampling 을 하고 Quantization 을 하게 된다.

Sampling : 아날로그 영상에서 일정한 간격으로 데이터를 얻는다.

샘플링의 크기는 가로, 세로 방향으로 각각 몇 번 데이터를 취득하는냐를 나타낸다.

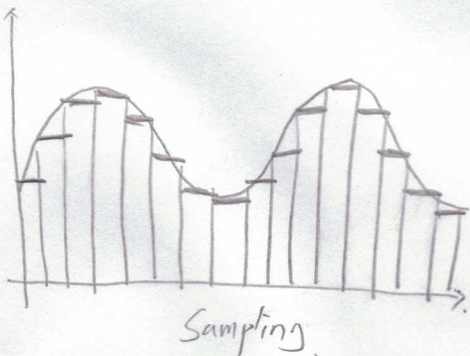
각 샘플은 pixel 이라 부른다.

Quantization : Sampling 으로 얻은 데이터가 숫자로 된 값을 할당한다.

각각의 샘플이 숫자로 된 값을 할당하는

Quantization 수준은 몇 비트로 사용하여 그 숫자를 나타낼 것인가에 따라 결정된다.

Quantization



* Sampling & quantization 을 더욱 많이 하면 메모리가 부족하고 더욱 작게하면 왜곡현상이 발생한다.

3. Mask 가

0	8	2
3	6	4
5	7	1

이다.

원 영상에서 0 ~ 26이 범위이다

27 Level 을 9 level로 변환해 주면 아래 표와 같다.

27 Level	9 Level
0 ~ 2	0
3 ~ 5	1
6 ~ 8	2
9 ~ 11	3
12 ~ 14	4
15 ~ 17	5
18 ~ 20	6
21 ~ 23	7
24 ~ 26	8

1 pixel 이 9 pixel

이 되므로

6x6 영상 사이즈가

18x18로 바뀐다.

즉 높은 해상도가 요구되어

결과를 6x6 으로 원소화하면

2x2 으로 원 영상을 변환

후 ordered dithering 을 해야한다.

1	0	0	1	0	0	1	0	0	1	0	1	1	0	1	1	0	0
0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0
0	0	0	0	0	1	0	0	1	1	0	1	0	0	1	0	0	0
1	0	0	1	0	0	1	0	1	1	0	1	1	0	1	1	0	0
0	0	0	0	0	0	1	0	0	1	0	1	1	0	0	0	0	0
0	0	0	0	0	1	0	0	1	1	0	1	0	0	1	0	0	1
1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1
1	0	0	1	0	1	1	0	1	1	1	1	0	0	0	0	0	0
0	0	1	0	0	1	0	0	1	1	0	1	0	0	1	0	0	1
1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1
0	0	0	1	0	1	1	0	0	1	0	0	1	1	1	1	0	1
0	0	1	0	0	1	0	0	1	0	0	1	1	1	1	1	0	1
1	0	1	1	0	1	1	0	1	1	1	1	1	1	1	1	0	1
1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1
1	0	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1
1	0	0	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1
0	0	0	1	0	1	1	0	1	1	1	1	1	0	1	1	1	1
0	0	1	1	0	1	1	0	1	1	1	1	1	0	0	1	1	1

<ordered dithering>

2	3	5	20	6	2
1	3	10	15	10	4
10	13	12	20	7	6
7	12	11	9	21	16
20	15	23	26	25	20
3	15	15	21	12	22

<원본>

=> active 된 부분은 13 표현해준다.

2	3	5
10	13	12
7	12	11

10	13	12
7	12	11
20	15	23

4. Median cut 알고리즘을 하기위해서 원 배열에서 0~26이 몇개씩 있는지 확인이 필요하다.

빈기	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
개수	1	2	3	1	1	2	2	0	1	3	1	3	1	0	4	1	0	0	0	4

↓
median

빈기	21	22	23	24	25	26	0
개수	2	1	1	0	1	1	0

총 36개 이므로 18개를 기준으로 median cut을 하된다.

즉 median은 12 이므로

0~12는 0, 13~26은 1로 두라.

⇒ 1 bit Quantization.

빈기	0	1	2	3	4	5	6	7	8	9	10	11	12
개수	6	1	2	3	1	1	2	2	0	1	3	1	3

⇒ 0

빈기	13	14	15	16	17	18	19	20	21	22	23	24	25	26
개수	1	0	4	1	0	0	0	4	2	1	1	0	1	1

⇒ 1

0인 빈기정보를 가진 table의 median은 총 26개의 절반 10을 기준으로 cut한다.

즉 median은 6 이므로

0~6은 00, 7~12는 01로 두라.

1인 빈기정보를 가진 table의 median은 총 18개의 절반 8을 기준으로 cut한다.

즉 median은 20 이므로

13~20은 10, 21~26은 11로 두라.

⇒ 2 bit Quantization.

빈기	0	1	2	3	4	5	6
개수	0	1	2	3	1	1	2

⇒ 00

빈기	7	8	9	10	11	12
개수	2	0	1	3	1	3

⇒ 01

빈기	13	14	15	16	17	18	19	20
개수	1	0	4	1	0	0	0	4

⇒ 10

빈기	21	22	23	24	25	26
개수	2	1	1	0	1	1

⇒ 11

00인 빈기정보를 가진 table의 value 수는 10 이므로 median은 5를 기준으로 cut한다.

즉 0~3은 000, 4~6은 001

이인 빈기정보를 가진 table의 value 수는 10 이므로 median은 5를 기준으로 cut한다.

즉 7~10은 010, 11~12는 011

10인 빈기정보를 가진 table의 value 수는 10 이므로 median은 5를 기준으로 cut한다.

즉 13~15는 100, 16~20은 101

11인 빈기정보를 가진 table의 value 수는 6 이므로 median은 3을 기준으로 cut한다.

즉 21~22는 110, 23~26은 111

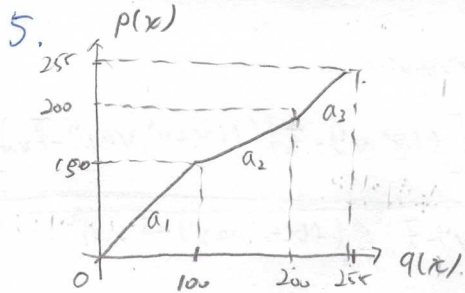
⇒ 3 bit Quantization.

10진수	2진수	빈기
0	000	0~3
1	001	4~6
2	010	7~10
3	011	11~12
4	100	13~15
5	101	16~20
6	110	21~22
7	111	23~26

⇒ Lookup table.

0	0	1	5	1	0
0	0	2	4	2	1
2	4	3	5	2	1
2	3	3	2	6	5
5	4	7	7	7	5
0	4	4	6	3	6

< 결과 영상 >



$$a_1 = 1.5$$

$$a_2 = 0.5$$

$$a_3 = 1$$

$$p(x) = 1.5 q(x) \quad (0 < q(x) \leq 100)$$

$$p(x) = 0.5(q(x) - 100) + 150$$

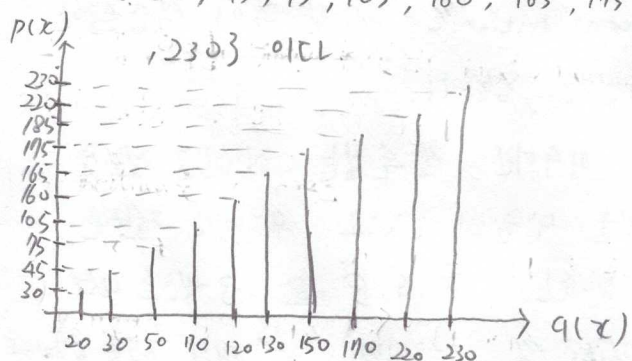
$$= 0.5 q(x) + 100 \quad (100 < q(x) \leq 200)$$

$$p(x) = 1(q(x) - 200) + 200$$

$$= q(x) \quad (200 \leq q(x) \leq 255)$$

$$q(x) = \{20, 30, 50, 70, 120, 130, 150, 170, 220, 230\}$$

$$p(x) = \{30, 45, 75, 105, 160, 165, 175, 185, 220, 230\}$$



6.

2	3	5	20	6
1	3	10	15	10
10	13	12	20	7
7	12	11	9	21
20	15	23	26	25

Image

-1	0	1
-2	0	2
-1	0	1

mask

* 가장자리 pixel 사용 제외.

$$\frac{\partial I}{\partial x}$$

2	3	5
1	3	10
10	13	12

$$-2 - 2 - 10$$

$$\times \text{mask} = +5 + 20 + 12 = 23$$

3	5	20
3	10	15
13	12	20

$$-3 - 6 - 13$$

$$\times \text{mask} = +20 + 30 + 20 = 48$$

5	20	6
10	15	10
12	20	7

$$-5 - 20 - 12$$

$$\times \text{mask} = +6 + 20 + 14 = 3$$

1	3	10
10	13	12
7	12	11

$$-1 - 20 - 7$$

$$\times \text{mask} = +10 + 24 + 11 = 17$$

3	10	15
13	12	20
12	11	9

$$-3 - 26 - 12$$

$$\times \text{mask} = +15 + 40 + 18 = 32$$

10	15	10
12	20	7
11	9	21

$$-10 - 24 - 11$$

$$\times \text{mask} = +10 + 14 + 21 = 0$$

10	13	12
7	12	11
20	15	23

$$-10 - 14 - 20$$

$$\times \text{mask} = +12 + 24 + 23 = 15$$

13	12	20
12	11	9
15	23	26

$$-13 - 24 - 15$$

$$\times \text{mask} = +20 + 18 + 26 = 12$$

12	20	7
11	9	21
23	26	25

$$-12 - 22 - 23$$

$$\times \text{mask} = +7 + 42 + 25 = 17$$

2	3	5	20	6
1	23	48	3	10
10	17	32	0	7
7	15	12	17	21
20	15	23	26	25

-3- $\frac{\partial I}{\partial x}$ 결과 (가장자리도 그대로 사용)

7. Template matching 방법은 어디에 8이 있는지를 찾는 데 사용할 수 있다.

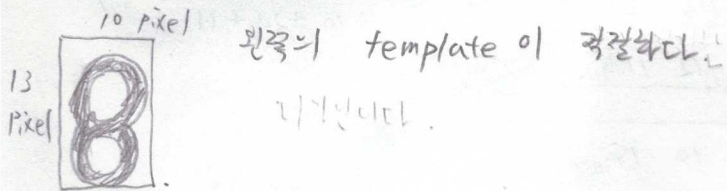
image 와 template을 correlation 했을 때.

상대적으로 아주 큰 값이 나오는 부분이 있다면 그 부분이 일치하는 부분이다.

Correlation은 Vector dot product와 유사하다.

변환된 이미지가 70×13 pixel 이고 총 7개 pixel과 밀 숫자가 들어 있다.

아래 보고 1개의 문자 혹은 숫자는 10×13 pixel 을 가진다고 볼 수 있다.



Image와 template을 matching 시켜주는 여러 방법이 있다.

① Cross - Correlation.

$$d_{cc}(x,y) = \sum_{x',y'} [t(x',y') - f(x+x',y+y')]^2$$

픽셀값의 제곱차를 이용해 matching 한다.

Template t 를 탐색 영역 f 에서 이동시키며 픽셀의 제곱의 합계를 계산한다. Matching 되는 위치에서 가장 작은 값을 가지고, 만약에 일치하면 0, 일치하지 않으면 작을수록 값이 커진다.

② Correlation Response

$$C(x,y) = f * t = \sum_{x',y'} [t(x',y') \cdot f(x+x',y+y')]$$

template과 입력 영상의 곱을 하여 모두 더한다.

Matching 되는 위치에서 큰 값을 갖고

일치하지 않으면 작은 값이 나오거나 0이 나온다.

③ Normalize Correlation

$$r(x,y) = \frac{\sum_{x',y'} [(t(x',y') - \bar{t})(f(x+x',y+y') - \bar{f}_{x,y})]}{\sqrt{\sum_{x',y'} (t(x',y') - \bar{t})^2 \sum_{x',y'} (f(x+x',y+y') - \bar{f}_{x,y})^2}}$$

즉 $\frac{Con(\vec{a}, \vec{b})}{\sqrt{Var(\vec{a})Var(\vec{b})}}$ 을 구하는 것이다. (이때 \vec{a} = template, \vec{b} = image)

Normalized 값은 -1 ~ 1 구간으로

나오게 되며 $\vec{a} = \vec{b}$ 면 1이 나와

matching 이 되고, $\vec{a} = -\vec{b}$ 이 나오면 inverstion 된 이미지만 같은 문자를 나타내 matching 된다.

결과가 0 이 나오면 일치하지 않는다.

8. a) Color Quantization 의 필요성

Color Quantization은 제한된 수의 색상으로만 이미지를 표현하는 방법이다.

무한대로 이루어진 색을 줄이는 방법을 색을 줄이는 만큼의 간격으로 유한한 정보만 사용하는 것이다. R, G, B 로 3색으로 나눠

각각 8bits 씩 조합하면 2^{24} 개의 색을 표현할 수 있다. Quantization을 하지 않으면 Memory가 이를 감당할 수 없어서 표현이 불가능할 것이다.

그렇다고 양자화 Level을 낮게 하면 많은 색상을 표시 못해 실제 이미지와는 차이가 생길 것이다.

따라서 적당한 Quantization이 필요하다.

b) Image Correlation 과 Convolution 의 차이점.

보통 신호처리를 할 때 Convolution은 $\int_{-\infty}^{\infty} g(u)h(t-u)du$ 로 뒤집어서 사용되고, Correlation은 $\int_{-\infty}^{\infty} g(u)h(u)du$ 로 그대로 곱해서 더하는 것이다.

하지만 영상 처리에서는 filter가 대칭인 경우가 많다. 즉, 굳이 대칭하지 않아도 그대로 사용할 수 있어서

-4- Convolution이나 Correlation이나 같이 사용한다.