

3. 영상의 산술 및 논리연산

영상간의 산술 및 논리연산은 두 영상의 픽셀들의 산술 및 논리 연산을 의미한다. 직관적이고 간단하지만 다양한 분야의 영상처리에 널리 활용되고 있다. 산술연산은 덧셈, 뺄셈, 곱셈, 나눗셈의 4칙 연산을 포함하며, 논리 연산은 논리곱, 논리합, 부정, exclusive-OR 등의 논리 연산을 포함하게 된다.

가. 덧셈 연산

두 영상의 정보를 결합하는데 사용되며, 영상의 합성이나 모핑 등 특수한 효과를 만들 수 있다. 아래의 식(3-1)처럼 단순한 합으로 표현할 수도 있고, 식(3-2)처럼 스케일링 효과를 반영하여 표현할 수도 있다. 식(3-2)에서 $\alpha=0.5$ 이면 두 영상간의 평균이 된다.

$$h(x,y) = f(x,y) + g(x,y) \quad (3-1)$$

$$h(x,y) = \alpha f(x,y) + (1 - \alpha)g(x,y) \quad (3-2)$$

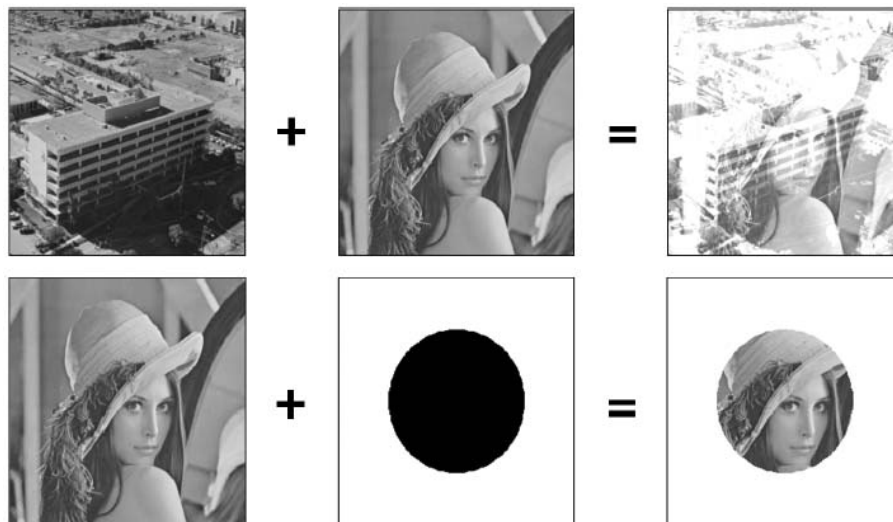


그림 3-1 영상의 덧셈

식(3-2)에서 $\alpha=0.5$ 이면 두 영상간의 평균이 되는데, 이를 응용하여 잡음제거 효과를 볼 수가 있다. 일반적으로 잡음이 컬러 노이즈(color noise)가 아니라 백색 가우시안 잡음이라면, 각각 다른 가우시안 노이즈가 첨가된 영상을 여러 장 더하면 잡음이 제거되는 효과를 볼 수가 있다.

$$\frac{1}{2} \left(\begin{array}{c} \text{Image 1} \\ \text{Image 2} \end{array} \right) + \begin{array}{c} \text{Image 3} \\ \text{Image 4} \end{array} = \begin{array}{c} \text{Image 5} \\ \text{Image 6} \end{array}$$

$$\frac{1}{8} \sum_{i=1}^8 \begin{array}{c} \text{Image 7} \\ \text{Image 8} \end{array} = \begin{array}{c} \text{Image 9} \\ \text{Image 10} \end{array}$$

그림 3-2 영상의 평균연산 및 이를 이용한 잡음제거

나. 뺄셈 연산

아래의 식(3-3)으로 표현되는 두 영상의 차는 두 영상에서 서로 대응되는 같은 위치의 해당 화소끼리의 차를 계산하여 구한다.

$$h(x,y) = f(x,y) - g(x,y) \quad (3-3)$$

또한, 식(3-4)와 같이 뺄셈 연산의 결과에 절대값을 취하여, 두 영상의 차이(difference)를 구할 수 있다. 차이 연산은 움직이지 않는 배경을 제거함으로써 영상내의 물체의 움직임을 검출하는데 주로 사용된다.

$$h(x,y) = |f(x,y) - g(x,y)| \quad (3-4)$$

$$\begin{array}{ccc} \text{Image 11} & - & \text{Image 12} \\ \text{Image 13} & - & \text{Image 14} \end{array} = \begin{array}{c} \text{Image 15} \\ \text{Image 16} \end{array}$$

그림 3-3 영상의 뺄셈



그림 3-4 영상의 차이(difference)

다. 영상의 AND 연산(논리곱) 및 OR 연산(논리합)

논리연산은 이진영상에서 주로 적용하며, 이진 영상처리의 기본 처리도구이다. 주요 연산은 AND, OR, NOT이며, 이들을 이용하여 다른 논리연산을 유도 할 수 있다.

128값을 가지는 영상과 AND 연산을 취하면 128보다 큰 값은 128로 바뀌게 되고, 128보다 작은 값은 0으로 바뀌게 된다. 또한, 127값을 가지는 영상과 OR 연산을 취하면 127보다 큰 값은 255로 바뀌게 되고, 127보다 작은 값은 127로 바뀌게 된다.

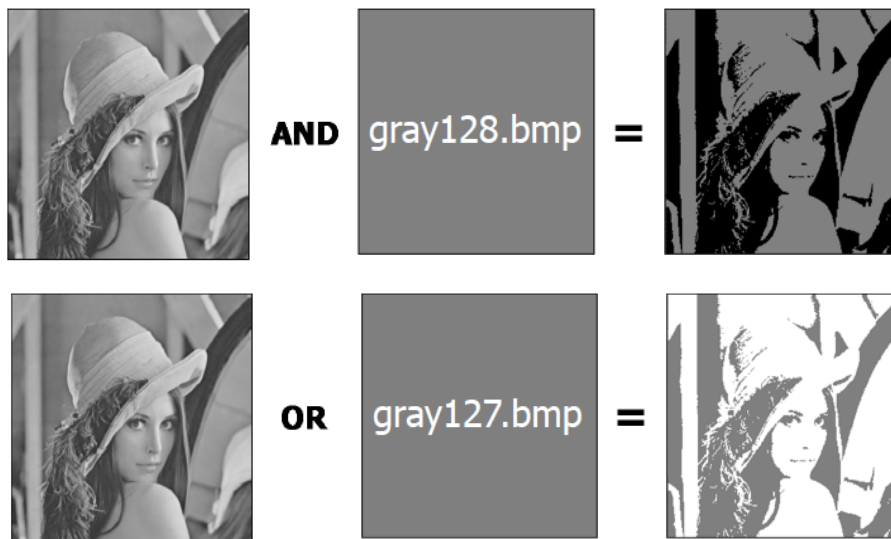


그림 3-5 영상의 논리연산의 예

< 실습 3-1 >

1. aero.bmp, hole.bmp, lena영상을 사용하여 그림 3-1 결과를 화면에 출력하시오.
2. 그림 3-2에서 잡음이 포함된 8장의 영상을 평균하여 잡음이 제거된 영상을 출력하시오. noise1.bmp - noise8.bmp를 사용하시오.
3. hole2.bmp 영상을 사용하여 그림 3-3 결과를 화면에 출력하시오.
4. diff1.bmp와 diff2.bmp를 사용하여 그림 3-4 결과를 화면에 출력하시오.
5. gray127.bmp와 gray128.bmp 영상을 이용하여, 그림 3-5 결과를 화면에 출력하시오.

라. 비트평면(Bit-plane) 분할

흑백영상의 경우 픽셀당 8bits를 할당한다. 이때, 8bits를 구성하는 각각의 비트들로만 영상을 구성하면 8개의 독립적인 영상을 획득할 수 있게 된다. 즉, MSB의 경우 1일 경우 128의 밝기 정보를 가지고, 0일 경우 0의 밝기 정보를 가지는 이진 영상이 만들어진다. 또한, LSB의 경우 1일 경우 1의 밝기 정보를 가지고, 0일 경우 0의 밝기 정보를 가지는 이진 영상이 만들어진다.

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

MSB

LSB

그림 3-6 픽셀값에 대응되는 8bits 예

이처럼 각 비트에 대해 그 값이 0인지 1인지 검사하여 이를 하나의 영상형태로 만드는 것을 비트평면(bit-plane) 이라하고, 흑백영상은 8개의 비트평면을 가지게 되며, 이를 분리하면 비트평면 분할이라고 한다. 비트평면 분할은 영상을 구성하는 각 픽셀의 특정 비트의 중요도를 분석하는데 중요한 정보를 제공한다. 그림 3-6은 lena 영상을 비트평면으로 분할한 결과를 보여주고 있다. 최상위 MSB 비트부터 최하위 LSB 비트까지 순서대로 비트평면을 분할한 결과를 차례대로 보여주고 있다. 최상위비트가 가장 큰 값을 가지므로 당연히 영상의 정보를 가장 많이 포함하고 있으며, 내려갈수록 영상의 정보가 사라지게 되며 최하위 비트의 경우 1과 0값으로만 구성되게 되므로 영상의 정보를 알아볼 수 없는 잡음영상처럼 보이게 된다.

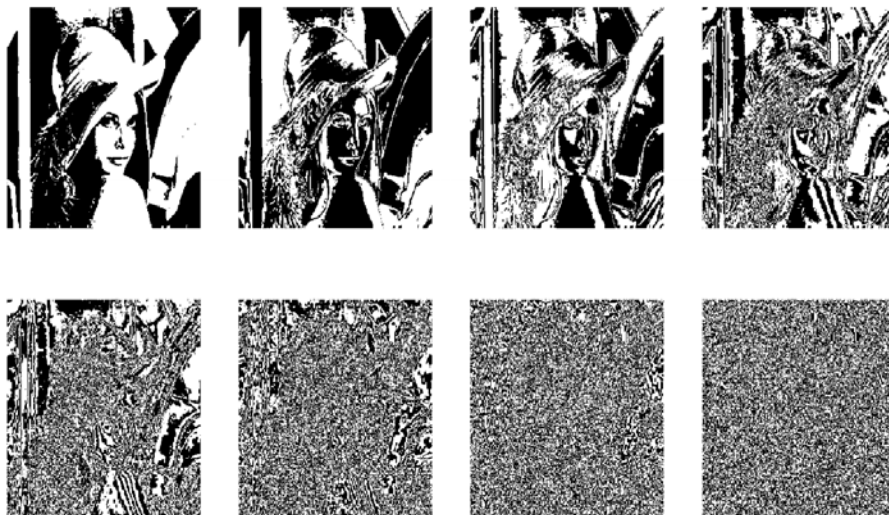


그림 3-7 lena 이미지의 비트평면 분할 결과



그림 3-8 lena 이미지의 비트평면 분할로부터 복원한 영상

* lambda 함수 란?

- 익명 함수 또는 함수의 축약형이다..
- 표현식 : lambda 인수 : <구문>

ex) lambda x : x + 1

ex) g = lambda x, y : x * y

g(2,3)

6

< Example 3-1 > Number to bitstream & BitPlane Slice Example

```
bin8 = lambda x : ''.join(reversed( [str((x >> i) & 1) for i in range(8)] ) )

def BitPlane_Slice(value):
    c = np.zeros(8)
    n = 7
    bits = bin8(value)

    for i in range(8):
        p = pow(2, n)
        c[i] = p * int(bits[i])
        n = n - 1
    return c

value = 100
Bits_Value = BitPlane_Slice(value)
print(Bits_Value)
```

마. 비트평면 분할을 이용한 영상의 워터마킹

영상의 최하위 비트는 1과 0으로만 이루어지므로, 실제 영상의 정보를 거의 가지고 있지 않으며, 잡음영상처럼 보이게 된다. 최하위 비트평면을 다른 정보로 교체하여도 실제 영상의 픽셀 값은 1만큼 변하거나 아니면 변하지 않게 된다. 즉, LSB 비트평면을 다른 비트평면 정보로 교체하여도 영상의 품질에는 거의 영향을 미치지 않게 된다. 이 점을 이용하면 영상의 화질저하를 최소화 시키면서 부가정보를 삽입하는 워터마킹이 가능하게 된다.

그림 3-9는 lena 영상의 비트평면 중에서 최하위 비트평면을 저작권 정보로 교체한 것을 보여주고 있다. 그리고, 그림 3-10은 최하위 비트평면에 저작권 정보를 삽입하여, 영상을 복원한 결과를 보여주고 있다. 그림 3-10에서 확인 할 수 있듯이, 최하위 비트평면이 다른 정보로 바뀌어도 영상의 품질에는 거의 영향을 미치지 않는다.

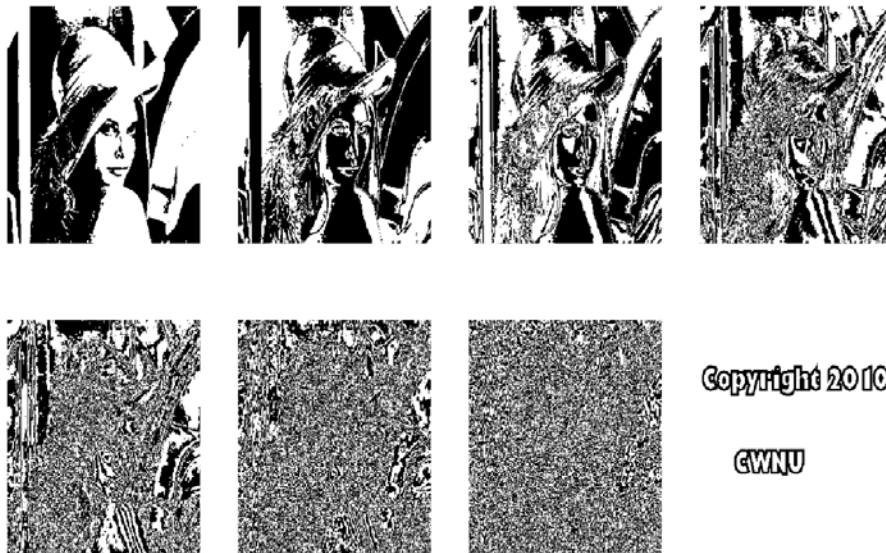


그림 3-9 lena 이미지에서 최하위 비트평면을 저작권 정보로 교체한 예



그림 3-9 최하위 비트평면을 저작권 정보로 교체하여 복원한 영상

< 실습 3-2 >

1. lena영상을 사용하여 그림 3-7의 결과를 화면에 출력하시오.
2. 비트평면으로 분할된 그림 3-7의 결과를 이용하여, 원본 이미지를 복구하고, 그림 3-8의 결과를 출력하시오.
3. copyright.bmp 영상을 lena 영상의 최하위 비트평면에 삽입하여, 비트평면들을 합하여 이미지를 복구하고 lena_w.bmp 파일로 출력하시오.
4. lena_w.bmp 파일을 읽어서, 워터마킹된 저작권 정보를 추출하시오.