

# 중간 - 5주차 정리

## 잡음의 종류와 제거

잡음 : Noise, 원래 있어야 하는 화소 X, 화소 값이 특별하게 바뀌어 있는 것

### 잡음 종류

- 1. 가우시안 잡음 : 영상 화소 값이랑 비슷한데 불규칙 한 잡음, 차이가 크지 않음

→ 잡음에 해당하는 화소 값들만 모아두고 히스토그램을 그리면 가우시안 분포를 이룸

- 2. 임펄스 (Impulse: 충격, 충동) 잡음 : 영상 화소 값과 뚜렷하게 다른 잡음

→ Salt (255, 백) & Pepper Noise (0, 흑) : 하얗거나 검은 잡음들



사운드 : 시간의 흐름에 따른 진폭 변화 신호 , 영상 : 가로/세로 위치 변화에 따른 신호

### 잡음 제거 방법

: 노이즈의 종류에 따라서 제거하는 방법이 다름 (가우시안→ 평균 마스크, 임펄스→중간 값 필터링)

1. 가우시안 잡음 : 평균 마스크(평활화 박스) 를 활용하여 잡음을 제거

→ 주변 화소 값이 비슷하기 때문에 영상의 대비 약화 시킴



⇒ 잡음은 흐려졌지만 퍼져 보이고 완벽히 제거되지 않는 단점이 발생

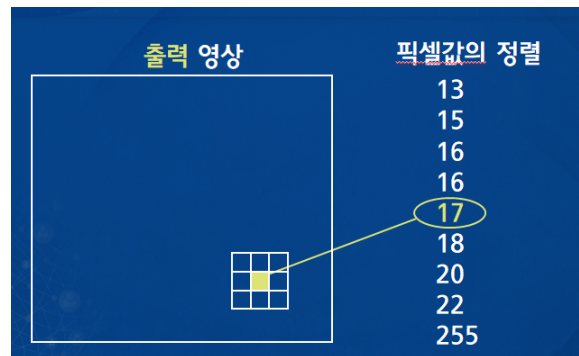
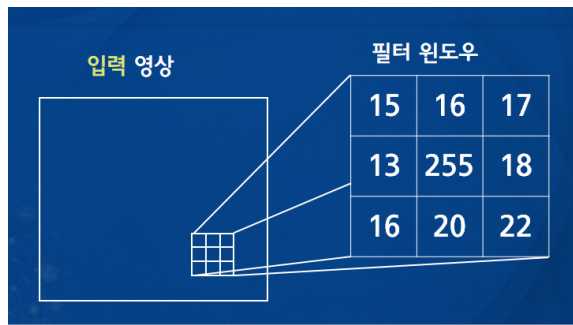
▲ 만약, **임펄스 잡음** 을 **평균 마스크** 를 활용하여 노이즈를 제거한다면?

⇒ 0 (Pepper) 혹은 255 (Salt) 에 해당하는 노이즈를 뭉갸다고 해도 오히려 퍼지기 때문에 비 효과적

ex ) 얼굴의 조그마한 점을 평균마스크를 활용하여 컨볼루션 연산을 진행하면 점의 밝기는 밝아지는 반면에 점의 크기는 커지는 문제가 발생



2. **임펄스 잡음** : **중간 값 필터링 (Median Filter)** 을 통해서 잡음을 제거 **By. Bubble Sorting**



: 9개의 화소 값들을 오름차순으로 정렬 시켜서 중간에 위치한 값을 Center Position에 위치 시킨다.

⇒ 컨볼루션 연산과 마찬가지로 센터 값을 중간 값으로 대체하는 연산

이 때, 노이즈가 많으면 해당 중간 값 필터링이 의미가 없다.

노이즈란, 3 x 3 영역에서 하나 정도의 빈도수를 의미

▲ 만약, 가우시안 잡음 을 중간값 필터링 을 활용하여 노이즈를 제거한다면?

⇒ 노이즈가 다른 화소와 큰 차이를 보이지 않기 때문에 Sorting 이후 중간 값을 찾아서 대체한다고 해도 잡음 제거에 큰 의미를 보이지 못한다.

## 임펄스 노이즈 제거 코드 → Median Filtering

: 가우시안 노이즈는 평균 마스크를 활용하면 되므로 생략



pepper(0)& salt(255)

```
void swap(BYTE * a, BYTE *b) {
    BYTE temp = *a;
```

```

    *a = *b;
    *b = temp;
}

void Sorting(BYTE * arr, int Size){ // 버블소팅
    for (int i=0; i<Size-1; i++){ //피벗 인덱스
        for (int j=i+1; j<Size; j++){ //비교대상 인덱스
            if (arr[i]>arr[j]) swap(&arr[i], &arr[j]); // call by reference
        }
    }
}

int main(){
    BYTE Temp[9];
    for (int i=1; i<H-1; i++){
        for (int j=1; j<W-1; j++){
            for (int m=-1; m<=1;m++){
                for (int n=-1;n<=1;n++){
                    Temp[(m+i)*3+(n+j)]=Image[(m+i)*3+(n+j)];
                }
            }
        }
        Sorting(Temp,9); // 정렬을 위한 임시배열 , 마스크 크기
        Output[i*W+j]=Temp[4]; // 중간 값을 센터 값으로 대체
    }
    return 0;
}

```

## 버블 소팅

3 5 1 2 7

