1. Average 필터링을 효율적으로 구현할 수 있는 Moving Average에 대하여 공식을 이용하여 설명한다.

만일 하나의 줄줄이 늘어선 픽셀의 값을 표현한 열이 있다고 가정한다. 이 문자열 P는 { p1, p2, p3 … pn } 이다. 이 때, 이의 평균 T1은

1/n\* 이 된다고 할 수 있다. 이 때, 이 열에서 { p1, p2, …. Pj } 까지가 삭제되고, 그 다음으로 { pn+1, pn+2.. pn+k } 가 추가되었다고 가정하자. 그럼 이의 평균 T2은

1/(k-j) \* 할 수 있다.

Average 필터링에서는 한 번 필터를 옮길 때마다 같은 수의 픽셀을 제거하고, 추가하므로 여기에서 삭제되고 추가되는 픽셀은 j로 같다. 즉 아무리 계속 진행해도 k-j는 n으로 고정되며, 이를 이용하면 T2는

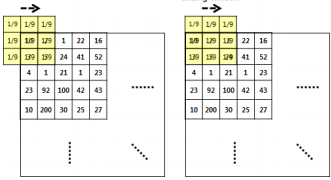
T2=1/(k-j) \* = 1/n\*(pj+1 + pj+2 …. + pk)

=(p1 + p2 … + pn) / n + p(n+1 + pn+2 … + pk) / n – ( p1 + p2… + pj) / n

=T1 + \* 1/n – 1/n \*

라고 표현할 수 있다. 따라서, Tn은 위의 식의 T1을 Tn-1로 바꾸고, 시그마 인자를 바꾼 것임을 알 수 있다.

1. Median 필터링을 효율적으로 구현할 수 있는 방법에 관하여 도식적으로 설명해 보자. 3x3 윈도우 사이즈를 사용하는 경우를 가정하여 구체적으로 설명한다.

만일 3\*3 윈도우 사이즈를 이용해 Median 필터링을 구현하는 경우, 

위와 같이 맨 왼쪽 위에서 시작한다 가정할 때, 만일 위의 (1,1)부터 (2,2)까지를 계산했다고 하면 그 다음의 (1,2) 위치의 픽셀에서는 굳이 필터가 겹치는 부분을 다시 계산할 필요가 없다. 이 경우에는 그저 겹치지 않는 부분인 1과 24만 계산하면 될 것이다.

다음 역시 겹치는 부분은 그대로 냅두고, 새로운 필터 영역인 22과 41을 계산하며, 새 필터에서 기존의 필터를 뺀 여집합인 (1,1)과 (1,2)의 값을 제거하면 다음 영역을 계산할 수가 있다.

이를 도식적으로 설명하기 위해, 한 필터의 맨 위를 p11, 맨 아래를 p33으로 둔다고 가정한다. 다음 sorting알고리즘을 처리하여, 아래와 같은 형태가 되었다고 하자.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| P11 | P31 | P22 | P33 | P21 | P12 | P13 | P32 | P23 |

그럼 가운데에 해당하는 p21이 median이 되며, 이 픽셀에 들어갈 값이 될 것이다.

다음은, p11과 p21, p31을 제거하고 크기가 3인 임시배열에 p14, p24, p34를 넣고, 이를 다시 한 번 해당 배열에 추가한다. 다시 한번 sorting알고리즘을 사용한다. 이 때, 아래와 같이 되었다고 하자.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| P14 | P22 | P33 | P34 | P24 | P12 | P13 | P32 | P23 |

그럼 가운데에 해당하는 p24가 median이 되며, 이 픽셀에 들어갈 값이 될 것이다.

이를 한 행이 끝이 날 때까지 반복하며, 새로운 행이 된다면 이 배열을 초기화하고 다시 한 번 과정을 반복하면 곧 moving median 필터링 기법이 완성된다.