**Lab 2. Image Filtering**

고급소프트웨어실습 3분반

20151591 이지현

**숙제: Average 필터와 Median 필터의 효율적 구현**

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

→ Window Size가 w인 Average 필터링에서, Moving Average 방식을 사용하고자 한다면, 현재 위치 (m, n)에서의 평균값은 이전 위치 (m-1, n)의 평균값에서 w-1이전 라인에 위치한 값의 평균을 뺀 후, w이후 라인에 위치한 값의 평균을 더해주면 된다. 이를 수식으로 나타내면 아래와 같다.

이를 실제 구현을 위해 공식으로 나타내면 다음과 같이 나타낼 수 있다.

다만, Kernel이 위치하는 값이 이미지 상 범위를 벗어나는 경우(예를 들면 이미지 시작지상 행이 시작되는 근방이나 끝나는 근방에서)에는 일반적인 Average 필터링 계산을 통해 구해야 한다.

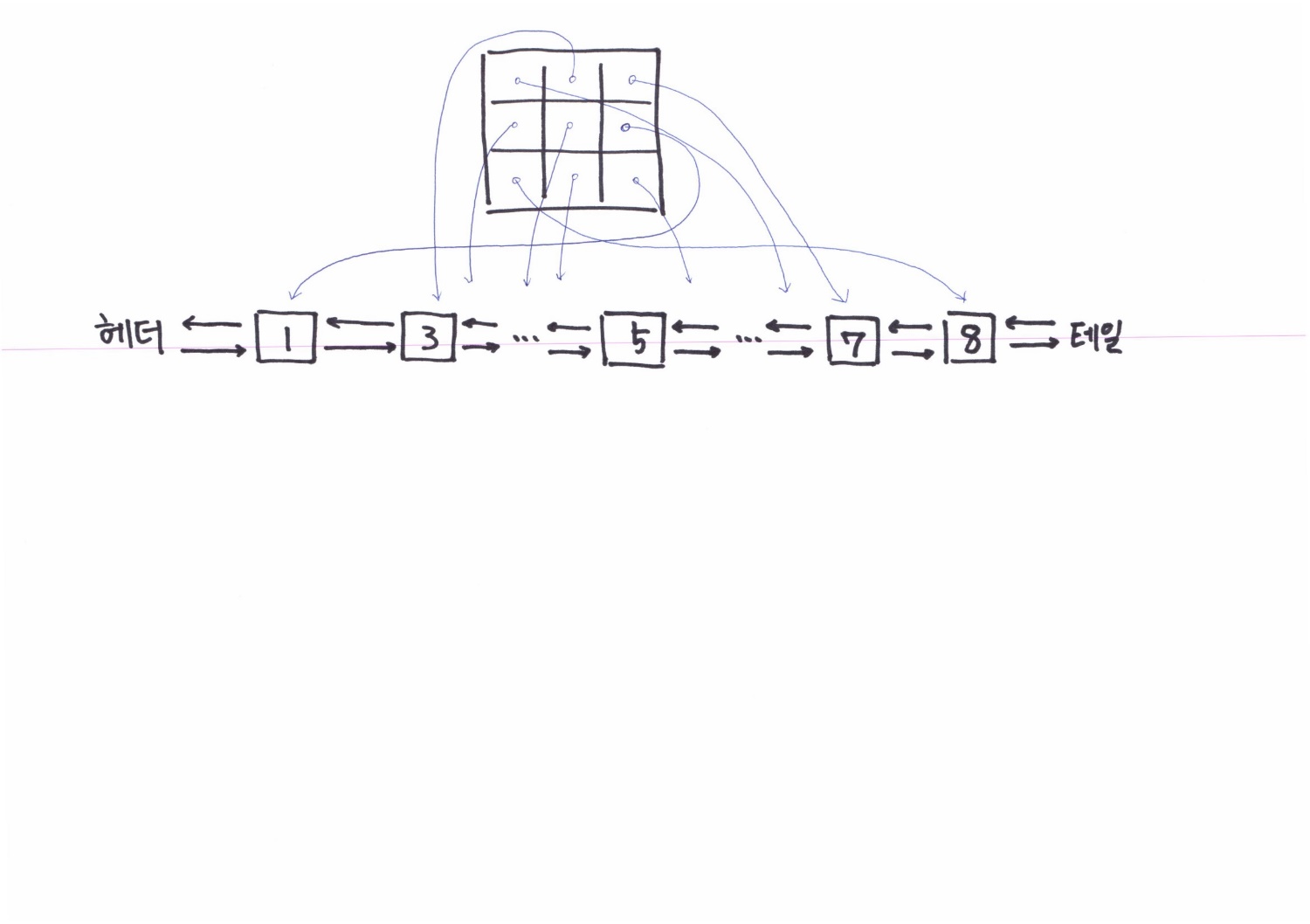
이 방식을 사용하면 모든 값을 일일이 계산하는 Average 필터링은 번의 계산이 필요한 반면, Moving Average 방식은 한 위치에서 w만큼만 연산을 진행하기 때문에 번의 계산이 필요하여, w가 커지면 커 질수록 연산 속도가 w에 비례하여 차이가 난다. 실습 결과 또한 이론적 결과와 비슷하게 나왔다.

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

→ 우리가 위의 Average 필터의 알고리즘을 향상시키기 위해 Moving Average를 사용했던 것과 유사하게 Median 필터링 또한 Moving하며 움직이는 Moving Median이라는 방식을 사용하면 속도를 향상시킬 수 있을 것이다. 예를 들면 아래와 같은 3×4 크기의 이미지가 있다고 가정하고, 빨간색이 칠해진 구역이 이전의 윈도우에 해당하는 위치, 노란색으로 칠해진 구역이 현재의 윈도우에 해당하는 위치라 하자.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 5 | 3 | 7 | 9 |  | 5 | 3 | 7 | 9 |  | 5 | 3 | 7 | 9 |
| 7 | 4 | 1 | 2 | 7 | 4 | 1 | 2 | 7 | 4 | 1 | 2 |
| 8 | 3 | 6 | 3 | 8 | 3 | 6 | 3 | 8 | 3 | 6 | 3 |

이 때, 3×3 윈도우에 해당하는 값을 저장할 더블 linked list를 구성하고, 각각의 노드를 가리키는 배열을 만든다. 또한 더블 linked list는 항상 정렬된 상태로 집어넣으며, Median 값을 가지는 링크 노드의 위치를 따로 저장한다. 이를 나타내면 다음과 같다.



Median 값

현재: 5

이 때, 다음 윈도우로 넘어가면 링크 노드를 가리키는 배열에서 맨 왼쪽에 해당하는 배열들의 노드를 찾아가 링크들을 모두 끊어주고 새로 들어온 9, 2, 3값을 가지는 노드들을 생성하여 연결한다. 또한 이 노드들도 linked list에 삽입해야 하는데, 현재의 Median 값을 가지는 위치와의 대소비교를 하여 삽입 위치를 설정한다. 마치 insert sort의 모습과 비슷하게 진행될 것이다. 이렇게 하여 새로운 median 값을 새 값들을 추가하며 자연스레 재설정 할 수 있다.