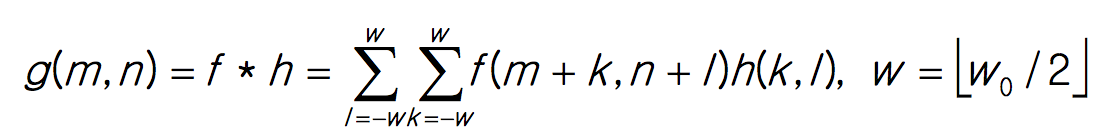
고급소프트웨어실습1 3반

Average 필터와 Median 필터의 효율적 구현

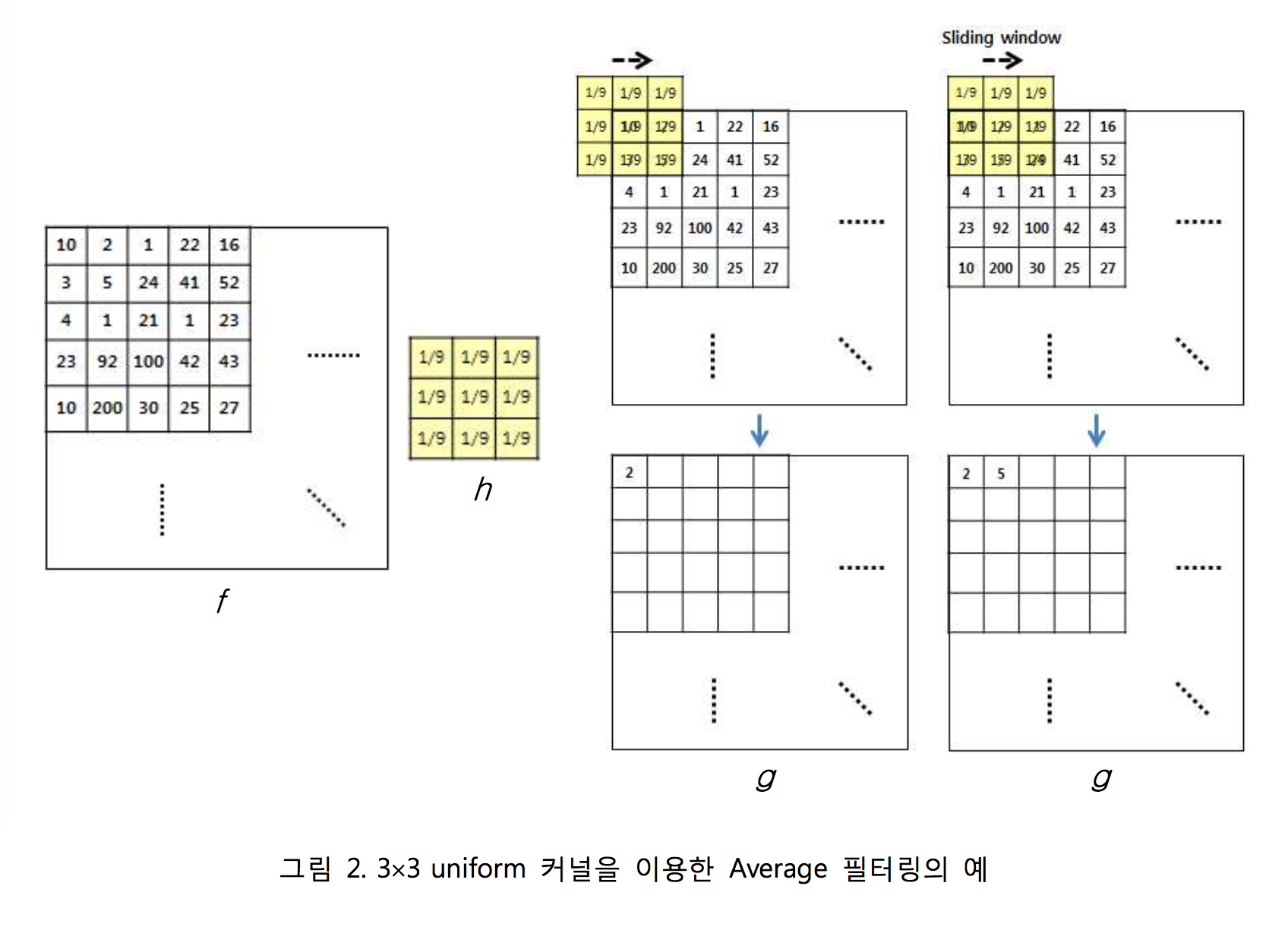
20121635 장종석

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

Average 필터는 가장 기본적인 필터링 기법으로, 각 픽셀 값에 대하여 주변의 픽셀 값들과의 평균 값을 구하여 결과이미지에 할당하는 방법으로, 아래와 같은 공식을 따르게 된다.



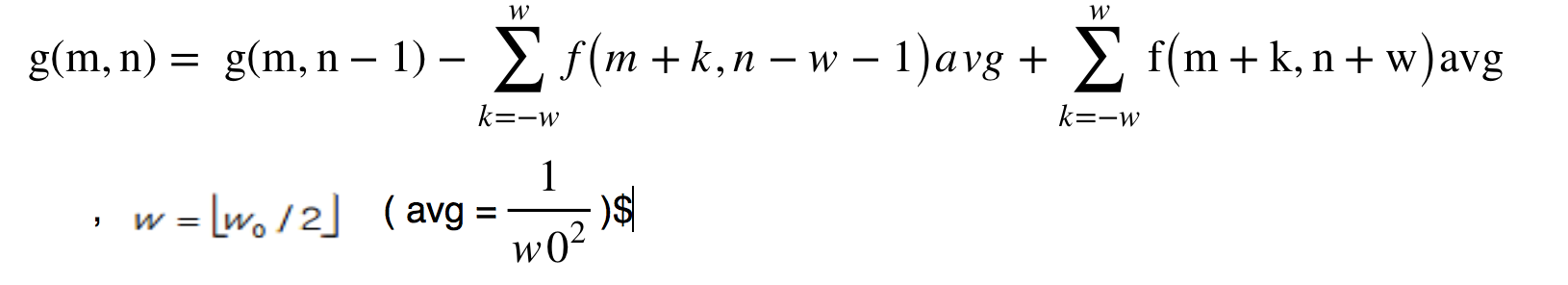
Average 필터의 경우 아래 그림과 같이 계산이 진행되어 결과값으로 저장이 된다.



이처럼, 계산을 할경우에는, 중복된 연산이 발생한다. 예를 들경우, (1,1)과 (1,2) 의 값을 계산하기 위해서(1,1)의 경우 (1,1),(1,2),(2,1),(2,2) 값을 계산하며, (1,2)의 경우 (1,1),(1,2),(1,3),(2,1),(2,2),(2,3) 값을 계산한다. 위의 연산의 경우 (1,1),(1,2),(2,1),(2,2) 의 연산이 중복되는 것을 확인 할 수 있다.

이를 개선하기 위해 Moving Average 연산법을 사용할 수 있다. Moving Average 의 경우, 기존에 계산 값을 기준으로 새롭게 연산하여 더해야 할 부분, 뺄 부분만 추가적으로 계산을 진행하면 된다. 예를들어, (1,1)과 (1,2)의 값을 계산하기 위해 (1,1)의 경우 (1,1),(1,2),(2,1),(2,2) 값을 계산하며, (1,2)의 경우 기존에 구해진 값에 (1,3),(2,3) 의 값을 추가해주면 된다. 이 경우, Average 필터링을 사용하는 경우보다 연산 횟수가 줄어든 것을 확인 할 수 있다.

이를 공식화 시키게 되면 아래와 같다.



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

Median 필터링은 Average 필터링과 비슷한 방법으로 필터 윈도우(박스)를 이동시켜 가며, 윈도우 안의 픽셀 값들의 중간 값을 결과 이미지(g)에 할당한다. (예 : 3 x 3 윈도우를 사용한다면, sorting 후 5번째 값에 해당.)

Median 필터링의 경우, 윈도우 안에 있는 픽셀값을 오름차순 혹은 내림차순으로 정렬하여 중간 값을 찾아야 한다. 즉, 기존의 방법을 사용할 경우, 픽셀이 움직일때마다, 정렬을 다시 해야한다. 만약, Moving Average 와 비슷하게 중복된 부분을 재활용한다면, 정렬횟수가 줄어드는 효과가 나타날 수 있다.

예를 들어보면, (2,2)과 (2,3)의 값을 median 필터링을 활용하여 구한다고 했을시,

(2,2) 는 (1,1), (1,2)(1,3), (2,1),(2,2),(2,3), (3,1),(3,2),(3,3) 의 값을 정렬시켜야 하며,

(2,3) 은 (2,1),(2,2),(2,3), (3,1),(3,2),(3,3), (4,1),(4,2),(4,3) 의 값을 정렬시켜야 한다.

즉, (2,1),(2,2),(2,3), (3,1),(3,2),(3,3) 부분이 중복됨을 알 수 있다. 이 부분을 새롭게 정렬하는게 아니라, 기존에 정렬되어 있는 부분을 활용한다면, 기존의 median 필터링보다 빠른 연산 결과를 나타낼 것이다.