멀티미디어 프로그래밍 과제3 보고서

22011831 김형규

목차

1. 개요

이미지 경로와 필터 사이즈를 입력 받아, 해당 필터 사이즈만큼 Mean Filtering을 빠르게 수행하되, 필터의 크기에 상관없이 동일한 시간이 걸리도록 해야 한다. 이 작업을 수행하는 함수를 만들어 주어진 입출력 조건을 만족하는 프로그램을 짜야 한다.

2. Mean Filtering이란?

평균 필터링은 이미지 필터링의 한 종류로, 임의의 한 픽셀의 밝기를 주변 픽셀의 밝기 값의 평균으로 조정하는 기술이다. 이 기술을 사용하면 이미지의 노이즈를 줄일 수 있다. 원리는 다음과 같다.

먼저 평균 필터링을 수행하는 필터(커널)의 크기 k를 입력받는다. 임의의 한 픽셀을 중심으로 주변 k개까지 픽셀의 밝기들을 검사하여, 그 밝기들의 평균을 계산한다. 이 평균값이 중심 픽셀의 밝기 값이 되는 것이다. 예를 들어, k가 1이라면 주변 1개의 픽셀들의 밝기를 검사하므로 필터의 크기는 3x3이 되는 것이다. 이 작업을 이미지의 모든 픽셀에 수행하면 이미지가 뿌옇게(blur 처리)된다.

수식으로 표현하면, 중심 픽셀의 좌표가 (x, y)라고 할 때, (x-k, y-k)부터 (x+k, y+k)위치에 있는 모든 픽셀, 총 $(2k+1)^2$개 픽셀의 밝기 값을 모두 더한다. 그 값을 $(2k+1)^2$로 나누면 중심 픽셀의 밝기 값이 된다. 반복문으로 x를 k~w-k까지, y를 k~h-k까지 돌리면 이미지의 테두리 부분을 제외한 영역에 평균 필터링 연산을 수행할 수 있다.

다만 이 방법은 연산횟수가 많아서 시간이 굉장히 오래 걸린다. 필터의 크기가 $(2k+1)^2$이기 때문에 k값이 증가함에 따라 필터의 크기는 변화량의 제곱만큼 커진다. 즉, k가 n만큼 커지면 연산에 걸리는 시간은 n^2만큼 느려지는 것이다.

3. Summed-Area-Table에 대하여

Summed-Area-Table(이하 SAT)은 2차원 배열의 원소들의 합을 계산할 때, 연산 횟수를 줄여 시간을 단축할 수 있는 방법 중 하나이다. $w \times h$크기의 2차원 배열 arr이 있다고 할 때, 이 2차원 배열에 대한 같은 크기의 SAT를 정의할 수 있다. SAT[y][x]의 값은 arr[0][0]부터 arr[y][x]까지 사각형 영역 안에 있는 모든 원소를 더한 값으로 정의한다. 이를 식으로 표현하면 다음과 같다.

두 변수 x와 y에 대해 arr과 SAT는 다음 식을 만족한다. (단, x<w, y<h이다.) ->식❶
1. x=0, y=0 일 때
SAT[0][0] = arr[0][0]

2. x>0, y=0 일 때
SAT[0][x] = SAT[0][x-1] + arr[0][x]

3. x=0, y>0 일 때
SAT[y][0] = SAT[y-1][0] + arr[y][0]

4. x>0, y>0 일 때
SAT[y][x] = SAT[y-1][x] + SAT[y][x-1] - SAT[y-1][x-1] + arr[y][x]

arr배열

2	7	1	4	5
3	2	8	6	2
1	0	6	9	2
8	5	3	2	4
6	4	5	3	1

SAT배열

2	9	10	14	19
5	14	23	33	40
6	15	30	49	58
14	28	46	67	80
20	38	61	85	99

SAT를 이용하면, Mean Filtering의 연산 횟수를 줄일 수 있다.
이미지는 CvScalar형 2차원 배열이라고 볼 수 있다. Mean Filtering에서 평균을 낼 때 필터 안에 있는 픽셀의 밝기 값을 모두 더하는 과정이 필요한데, 여기서 SAT를 이용할 수 있다.

imgArr

2	7	1	4	5
3	2	8	6	2
1	0	6	9	2
8	5	3	2	4
6	4	5	3	1

imgSAT

2	9	10	14	19
5	14	23	33	40
6	15	30	49	58
14	28	46	67	80
20	38	61	85	99

imgArr은 이미지에서 각 픽셀의 밝기 값을 2차원 배열 상에 대입한 것이고, imgSAT은 imgArr에 대한 SAT이다. imgArr의 중앙에 있는 픽셀에 대한 평균 필터링을 진행하려고 한다. imgArr의 노란색 영역 안에 있는 값들을 모두 더해야 하는데, imgSAT의 주황색 칸에는 imgArr의 노란색 영역, 하늘색 영역, 분홍색 영역, 연두색 영역 안의 값들을 모두 더한 값이 저장되어있다. 즉, (노란색) = 주황색(67) - (하늘색) - (분홍색) - (연두색)인 것이다.
그런데, (하늘색) = 파란색(14) - 초록색(2), (분홍색) = 빨간색(14) - 초록색(2)이고, (연두색) = 초록색(2) 이다. 따라서, (노란색) = 주황색(67) - 파란색(14) - 빨간색(14) + 초록색(2) 이다.
이것을 일반화하면 중심이 (x,y)이고 필터의 크기가 $(2k+1)^2$라고 할 때, 필터 안에 있는 숫자의 합은 **SAT[y+k][x+k] - SAT[y+k][x-k-1] - SAT[y-k-1][x+k] + SAT[y-k-1][x-k-1] -> 식❷** 라고 할 수 있다. 이 수식의 값을 필터의 크기로 나눠 (x, y)위치의 밝기 값에 대입하면 평균 필터링이 된다. 필터 안의 값들의 평균을 단 네 번의 연산만으로 구할 수 있는 것이다.

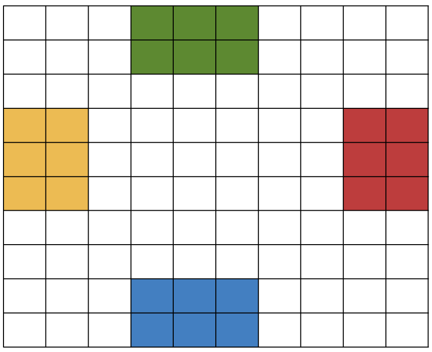
4. 핵심 아이디어 & 시행착오

위에서 가장 중요하게 생각해야하는 부분은 2차원 배열 두 개를 먼저 정의하는 것이다.
2차원 배열은 이미지의 크기와 똑같지만, 입력으로 어떤 이미지가 들어올지 모르기 때문에 2차원 배열을 동적으로 할당해줘야 한다. 입력으로 들어올 이미지를 src라고 하고, src의 가로 길이를 w, 세로 길이를 h라고 하자..
src는 CvScalar형 변수가 2차원 배열을 이루고 있는 것이라고 해석할 수 있다. CvScalar형 배열의 시작 위치를 가리키는 포인터를 h개 할당 받은 뒤, 반복문을 돌려 각 포인터마다 CvScalar형 변수를 저장할 수 있는 공간 w개를 동적으로 할당 받으면 CvScalar형 2차원 배열을 동적으로 할당 받을 수 있다. 2차원 배열 동적할당을 코드로 표현하면 다음과 같다.

```
CvScalar** srcArr = (CvScalar**)malloc(sizeof(CvScalar*) * h);
for (int y = 0; y < h; y++) {
        srcArr[y] = (CvScalar*)malloc(sizeof(CvScalar) * w);
}
```

이 배열의 각 원소에는 이미지의 같은 위치에서의 CvScalar형 변수가 들어있는 것이다. 이 배열을 srcArr라고 하자.

그리고 같은 방법으로 CvScalar형 2차원 배열을 할당 받아 srcArr에 대한 SAT를 만들어 준다. **식❶**을 활용하여 srcArr에 대한 SAT배열의 값들을 초기화 해준다. 이 배열을 srcSAT라고 해준다. 그리고 최종 결과 값을 저장할 IplImage형 변수 dst를 src와 같은 크기로 만들어주면, srcArr과 srcSAT를 이용하여 dst를 그리는 작업을 시작할 수 있다.
식❷를 이용하면 이미지의 대부분 영역에서 평균 필터링을 진행할 수 있다. 하지만 필터가 이미지의 테두리 쪽에 붙어있을 때는 따로 예외처리를 해주어야 한다.



이 10x10크기의 2차원 배열을 이미지라고 생각하고, 3x3 필터가 위의 4가지 색깔의 영역에 위치했을 때를 생각해 보자. 원래대로 계산하면 필터의 크기는 9가 되지만, 필터의 테두리 부분이 이미지 밖을 넘어가는 바람에 필터의 크기가 줄어들게 된다.
또한, 필터가 색칠된 영역에 있을 때는 **식❷**에서 정의되지 않는 항이 생긴다. 예를 들어, 필터가 초록색 영역에 있을 때는 y-k-1이 0보다 작기 때문에 **식❷**에서 SAT[y-k-1][x+k]와 SAT[y-k-1][x-k-1]가 정의되지 않는다. 또, 필터가 빨간색 영역에 있을 때는 **식❷**에서 SAT[y+k][x+k]와 SAT[y-k-1][x+k]가 정의되지 않는다. 즉, 평균 필터링에서 주의해야하는 부분은 두 가지이다.

1. 필터의 크기가 바뀌는가?

2. 식❷에서 정의되지 않는 항이 있는가?

필터는 중심이 (x, y)일 때 왼쪽 위 꼭짓점이 (x-k, y-k)이고 오른쪽 아래 꼭짓점이 (x+k, y+k)이다. 필터의 가로길이를 kw, 세로길이를 kh라고 하면, 필터의 크기는 kw*kh이다. 만약 x+k>=w 이거나 y+k>=h 라면, 필터의 크기는 바깥쪽으로 삐져나가는 만큼, 즉 kw는 (x+k)-(w-1)만큼, kh는 (y+k)-(h-1)만큼 줄어든다. 줄어든 필터의 크기를 수식으로 표현하면 [kw-{x+k-(w-1)}] * [kh-{y+k-(h-1)}]가 된다. 또한, **식❷**에서 인덱스 값에 x+k나 y+k가 들어가 있는 항은 정의되지 않는다. 이런 경우, x+k 대신 w-1(오른쪽 끝의 x좌표)를 대입하고, y+k 대신 h-1(아래쪽 끝의 y좌표)를 대입한 뒤에 **식❷**를 계산하면 정의되지 않는 항 없이 필터 안의 밝기 값들의 합을 계산할 수 있다.

만약, x-k<0 이거나 y-k<0라면, kw는 (k-x)만큼 줄고, kh는 (k-y)만큼 줄어든다. 즉, 줄어든 필터의 크기는 (kw-(k-x)) * (kh-(k-y)) 가 된다. 또한, x-k-1<0 이거나 y-k-1<0 이면 **식❷**에서 정의되지 않는 항이 생긴다. 이 경우에는 그냥 정의되지 않는 항을 빼고 계산하면 된다.

이 예외처리를 통해 필터 안에 있는 숫자들의 합을 조정하고, 이 값을 조정된 필터의 크기로 나눠주면 이미지의 모든 영역에 대해 평균 필터링을 수행할 수 있다.