컴퓨터 비전

HW #2. Filtering

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 1번 | 2번 | 3번 | 4번 |
| 구현 완료 여부 | 작성 | 성공 | 성공 | 성공 |

학번 2016603042

이름 최영찬

제출일 2021.10.28.

1. **RGB 영상 포멧 확인**

텍스트, 건물, 실외이(가) 표시된 사진

자동 생성된 설명



이미지를 불러와서 읽어드린 모습

텍스트이(가) 표시된 사진

자동 생성된 설명

그림과 같이 BGR 즉 blue,green,red 순으로 저장되어있다.

1. **YUV (YCbCr) color space**

**1)  YUV color space에 대해서 간략하게 조사**

YUV (YCbCr) color space란 디지털 비디오 스트림 전송 포멧이다.

영상에서 밝고 어두운 정도를 나타내는 성분 Y(Luminance) , 색상 정보를 가지는 U 와 V 성분(chrominance) 으로 구성되어있다.

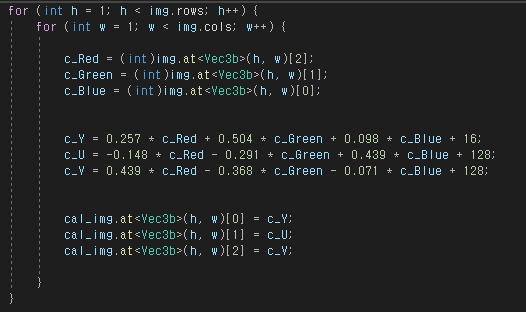
RGB영상에서 YUV영상으로 변환되는 원리

Y값은 RGV 의 각각의 값의 가중치를 부여하여 더한 값으로 결정된다.

U,V는 각각 Y를 빼서 얻는다.

즉 Y = K1\*R + K2\*G + K3\*B , U=B-Y , V= R-Y

**2) RGB를 YUV로 변환하는 코드를 작성 (실제 matrix 연산을 해서 구해야 함)**



이와 같이 R,G,B에 대한 정보를 img에서 추출해서 c\_Red, c\_Green , c\_Blue 에 저장한다.

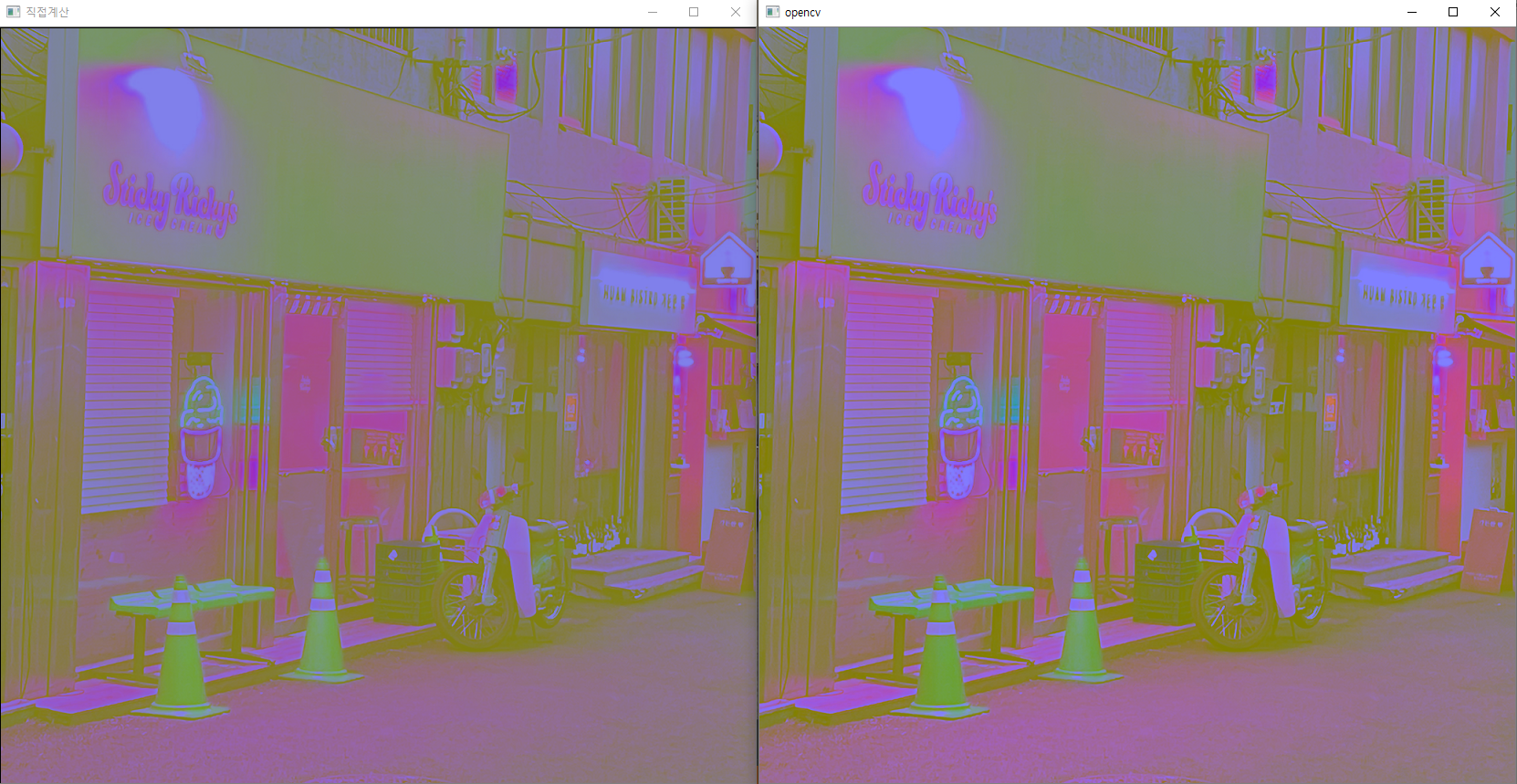
이후 R,G,B 에 각각 가중치를 곱하여 Y,U,V를 만든다

그리고 출력될 영상 cal\_img 의 BGR에 YUV값으로 대체한다.

**3) cvtcolor를 이용하여 YUV로 변환한 결과와 2)의 결과를 비교**

cvtColor 을 이용하여 입력영상 img를 opencv\_img로 BGR을 YUV로 만든다.





왼쪽이 직접 mat 데이터에 접근하여 계산한 방식이고 오른쪽은 cvtColor를 이용한 것이다.

결과적으로 두 영상이 거의 비슷하게 결과가 나왔다.

**3. Filtering 함수 작성**

**O Moving average filter**

텍스트이(가) 표시된 사진

자동 생성된 설명 3x3 ,5x5 필터를 만들었다.

모든 픽셀에 똑 같은 가중치를 주도록 만들었다.

텍스트이(가) 표시된 사진

자동 생성된 설명

경계값에 있는 것들은 그 전 픽셀의 값을 복사하여 repetition 한다.

sum값이 0보다 작은 것은 0, 255보다 큰 것은 255로 한다.

**O laplacian filter**

텍스트이(가) 표시된 사진

자동 생성된 설명3x3

배열의 합이 -1-1-1-1+8-1-1-1-1= 0 이 된다.

이 filter를 영상에 적용시키면 픽셀값이 일정한 부분에서는 0이되고 크게 변하는 edge가 있는 영역에서는 값이 커지게 된다.

텍스트이(가) 표시된 사진

자동 생성된 설명

흑백으로 바꾼 영상을 Laplacian filter를 통해 계산하는 과정이다

**4. 다양한 filter 적용 (위에서 구현한 filter 함수 사용)**

**O 3x3 크기의 moving average 수행**

텍스트, 실외, 하얀색, 도시이(가) 표시된 사진

자동 생성된 설명

왼쪽이 필터 적용 전, 오른쪽이 필터 적용 후

전체적으로 영상이 흐릿해지면서 Blurring 된것을 확인할 수 있다.

**O Laplacian 수행**

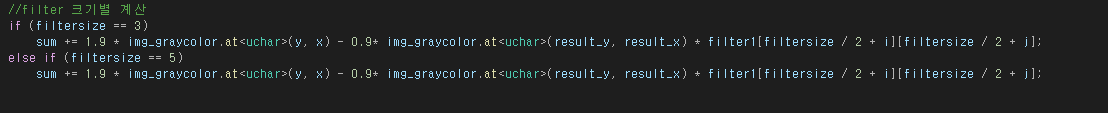
텍스트, 칠판이(가) 표시된 사진

자동 생성된 설명

Laplacian 필터를 적용하여 에지를 검출한 영상.

변화값이 크기 않은곳은 검정색이지만 변화가 있는 영역에서는 하얗게 테두리처럼 되어있는것을 확인 할 수 있다.

**O Sharpening filter 구현**



기존 원본 영상에서 moving average의 블러링된 영상을 빼서 더욱 선명해보이는 영상을 얻을 수 있다.

(1+a)\* Img\_graycolor(원본흑백영상) – a \* moving average filter 영상

위와 같은 공식을 이용했다.

아래 결과영상은 a=0.9 일경우로 구현했다.

텍스트, 실외, 도로이(가) 표시된 사진

자동 생성된 설명

텍스트, 실외이(가) 표시된 사진

자동 생성된 설명

위에가 원본영상, 아래가 sharpening을 한 영상이다

전체적으로 선이 굵어지고 진해졌다.

텍스트, 표지판, 실외이(가) 표시된 사진

자동 생성된 설명

특히 왼쪽 상단의 간판 글씨를 보면 sharpening을 한 영상이 테두리가 더 진하고 선명하게 보인다. 이로써 sharpening 구현에 성공했다.

이때 그냥 영상을 합치면 픽셀과의 연산에서 255가 넘거나 0 아래로 넘는 것이 생겨서

영상의 색이 반대로 나올 수 있다.

따라서 이때



를 써줘서 영상 두개를 합칠 때 오버플로우가 되는 현상을 막는 것이 중요하다.