

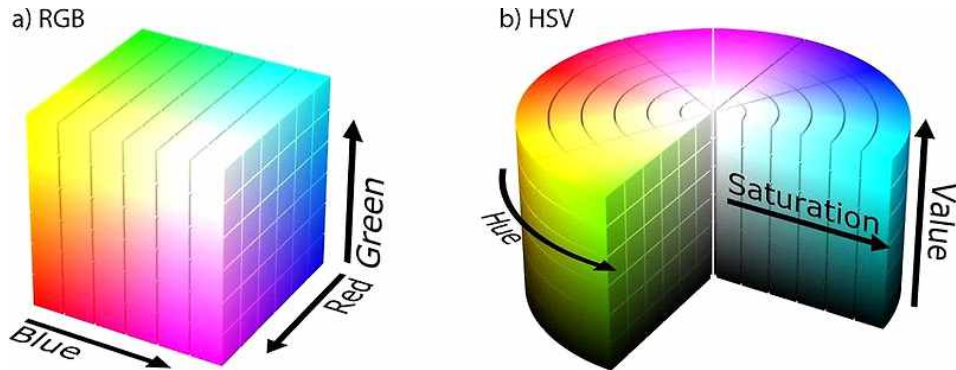
Vision & openCV day1 hw1

hsv & GaussianBlurs2 구현 과제 보고서

목차

- HSV 란?
- Gaussian 필터란?
- HSV 구현 코드 설명
- Gaussian 필터 구현 코드 설명
- 가우시안 필터 적용 후 변경점

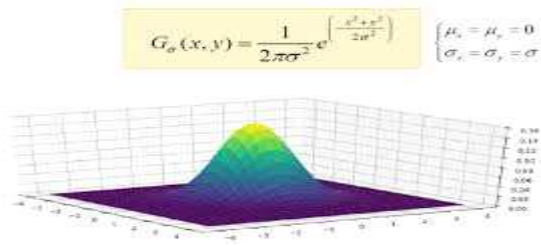
- HSV 란 ?



- HSV

- RGB와는 다르게 색을 나타내는 방법
- 색도, 채도, 명도의 3가지 요소로 구성되어 있는 색을 표현하는 하나의 방법이다.
- 색도란 색상이라고도 불리며 일반적으로 주파장과 순도를 결합시켜서 표시하거나 그 색에 포함된 3원색 즉 RGB의 비율에 의해 표시된다.
- 색상을 원처럼 배열한 것을 색상환이라고 부르며 밝기를 무시한 색의 성질을 색도라고 부른다. 색상환에서 바로 옆에 위치한 색들을 유사색이라고 부르며 반대편에 있는 색을 보색이라고 부른다. 보색의 양 옆에 이웃해있는 색의 경우 반대색이라고 부른다.
- 명도는 밝기의 정도를 나타낸다. 더욱 자세히 설명하면 명도는 물체의 색이나 빛의 색이 지니는 밝기의 정도를 나타낸다.
- 명도가 높은 컬러를 톤트, 명도가 낮은 컬러를 셰이드라고 한다.
- 명도는 먼셀의 색체계를 적용하면 수직축의 높이로 표현된다.
- HSV 색공간에서는 명도를 0에서 100사이의 값을 가지는 명도값 b로 표현한다.
- 채도란 색의 선명함 정도를 말한다.
- 선명도의 높낮이에 따라 선명함이 달라지며 만약 채도가 0일 경우 무채색이라고 부른다.
- 채도가 높은 색을 흔히 짙다 라고 표현하며 채도가 낮은 색을 흔히 희리다 라고 칭한다.
- 위의 모든 개념을 이용해 OPEN CV에서 기존 이미지를 HSV 색상 공간으로 변환하여 특정 색상의 물체를 추적하거나 분리하는 등의 작업을 수행할 수 있다.

- Gaussian 필터란?



- 가우시안 필터란 가우시안 분포, 즉 정규 분포를 이용해 이미지의 노이즈를 제거하고 이미지를 부드럽게 만드는 영상 처리 기법이다.
- 필터링을 대상으로하는 픽셀에 가장 높은 가중치를 설정하고 주변 픽셀로 갈 수록 가중치를 점점 줄여간다.
- 단순히 평균 필터와 다르게 중앙 픽셀의 영향을 더 중요하게 생각하여 더 부드럽고 자연스러운 블러링 효과를 얻을 수 있다는 장점이 있다.
- 원리 : 가우시안 분포 함수를 사용하는 필터 마스크를 생성한다. 이를 커널이라고 부르기도 한다, 중앙에서 멀어질수록 0에 가까워진다.
-> 필터 마스크를 겹쳐 픽셀 값과 마스크 값을 곱하여 모두 더하는 컨볼루션 계산을 한다. -> 원본 픽셀의 가중 평균값을 대체한다.
- 용도 : 이미지의 노이즈를 제거하여 깨끗한 이미지를 얻을 때
 , 이미지의 외곽을 전체적으로 부드럽게 만들어주고 싶을 때

-HSV 구현 코드 설명

먼저 헤더파일 부분을 살펴보며 사용할 변수에 대해 설명하겠다.

```
● #include <iostream>
   #include "opencv2/opencv.hpp"
```

- C++ 의 기본적인 기능을 이용하기 위해 iostream을 선언
- hsv를 구현하는 함수를 opencv2에서 불러오기 위해 opencv2/opencv.hpp를 인클루드.

- class OpenCV

- 이 프로그램에서 주로 사용할 OpenCV 클래스 선언

- private:

```
cv::Mat hsv;
cv::Mat dst;
cv::Mat red_s1;
cv::Mat red_e1;
cv::Mat red_s2;
cv::Mat red_e2;
cv::Mat blue_e;
cv::Mat blue_s;
cv::Mat red;
```

- private 란에 이미지 정보를 저장하는 cv의 자료형 Mat을 이용하여 적용할 이미지, 결과 이미지에 해당하는 변수를 선언
- 위에서부터 그린, 레드, 블루

- cv::Scalar lower_g;
cv::Scalar upper_g;
cv::Scalar low_r1;
cv::Scalar high_r1;
cv::Scalar low_s2;
cv::Scalar high_s2;
cv::Scalar low_b;
cv::Scalar high_b;

- 특정 색만 추출하도록 하기 위해 범위를 정해주어야 하는데 그 범위와 최대와 최소를 정하는 Scalar 변수들을 선언
- 순서대로 녹색 : 최소, 최대 / 빨강1 : 최소, 최대 / 빨강2 : 최소, 최대 / 파랑 : 최소, 최대

- public :

```
void hsvv();
```

- hsvv로 변환해주는 hsvv함수를 선언한다. 함수 구현과 관련한 부분은 cpp에서 설명하겠다.

● `std::string path;`
`cv::Mat src;`

- 사진의 파일 주소를 문자열로 저장하여 불러오기 위한 path 변수
- src 변수는 이 path 경로를 읽어 이미지를 불러오는 역할을 한다.

● `show_window()`
- 해당 함수는 변환한 이미지를 윈도우 창에 출력하는 함수이다.

다음은 본격적으로 cpp 파일을 살펴보겠다,
위에서 설명한 hw1.hpp와 같이

● `#include <iostream>`
`#include "opencv2/opencv.hpp"`
`#include "../include/hw1_package/hw1.hpp"`

C++, opencv프로그램을 실행하는데 필요한 헤더파일과 hw1.hpp 파일을 인클루드한다.

● `void OpenCV::show_window() {`

`cv::imshow("Row",src);`
`cv::imshow("Green",dst);`
`cv::imshow("Red",red);`
`cv::imshow("Blue",blue_e);`

`}`

- hpp에서 언급했듯 해당 함수는 처리한 이미지를 윈도우 창에 출력하는 기능을 하는 함수이다.
- ROW라는 이름의 창에 src <원본 이미지>를 띄우도록 하였다.
- GREEN이라는 이름의 창에 dst <녹색 추출적용 이미지>를 띄우도록 하였다.
- RED라는 이름의 창에 red <빨강 추출 적용이미지>를 띄우도록 하였다.
- BLUE라는 이름의 창에 blue_e <파랑추출 적용 이미지>를 띄우도록 하였다.

● void OpenCV::hsvv()

- hsv를 구현하는 함수에 대해 설명하겠다.

```
● cv::cvtColor(src, hsv, cv::COLOR_BGR2HSV);
  lower_g = cv::Scalar (35, 50, 50);
  upper_g = cv::Scalar (85, 255, 255); // hsv 범위
```

```
  cv::inRange(hsv, lower_g, upper_g, dst);
```

- src를 cvtColor 라는 cv 전용 함수를 이용해 RGB 색 공간에서 HSV를 뜻하는 Hue, Saturation, Value 색 공간으로 변환 후 결과 이미지를 hsv에 저장한다.
- lower_g 과 upper_g의 범위를 지정하여 HSV 범위 또한 지정해준다, SCalar 괄호 뒤 숫자들은 blue, green, red의 순서로 각 채널의 값을 나타내며 변환 사이트를 찾아보며 옆에 얹은 동기와 함께 최적의 값을 찾아내었다.
- cv 전용함수 inRange로 hsv의 각 픽셀이 지정된 lower g과 upper g 범위 내에 있는지 확인하여, 해당 픽셀을 255->흰색, 범위를 벗어나면 0->검은색으로 변환하여 결과 이미지 dst에 저장한다.

● (위의 코드와 구조가 동일)

```
.
```

```
red = red_e1 + red_e2;
```

빨강의 경우에도 이와 비슷하지만 한 가지 차이점이 있다.

과제로 주어진 사진에서 빨간공의 값이 일정하지 않다는 것을 BGR 값을 찾으며 알게 되었고 이를 해결하기 위해 두 개로 나누어 hsv로 각각 이미지를 만든 후 이 두 개를 합치는 과정을 추가해주었다.

- (green 코드와 구조가 동일)

.

.

```
cv::inRange(blue_s, low_b, high_b, blue_e);
```

- 파랑 hsv 적용 코드는 red와 달리 green과 구조가 같다.

Gaussian 필터 구현 코드 설명

- void OpenCV::Gaussianblur() {

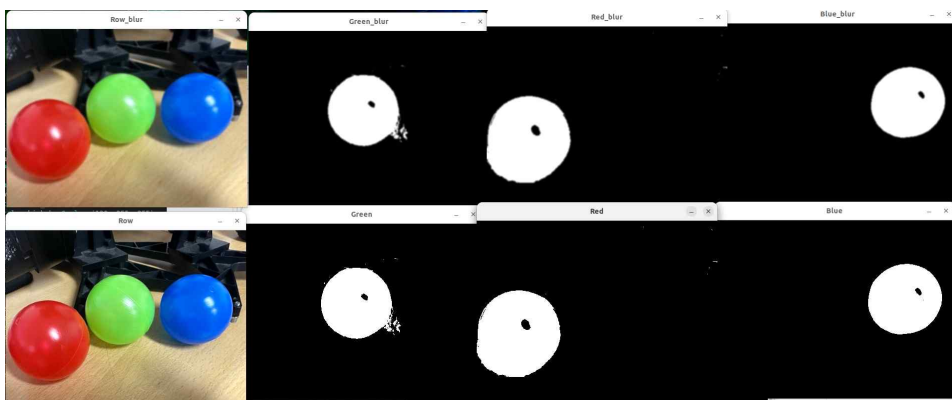
```
    cv::GaussianBlur(src, src_blur, cv::Size(7, 7), 0);
    cv::GaussianBlur(dst, dst_blur, cv::Size(7, 7), 0);
    cv::GaussianBlur(red, red_blur, cv::Size(7, 7), 0);
    cv::GaussianBlur(blue_e, blue_e_blur, cv::Size(7, 7), 0);
```

}

- 위 함수는 hsv 변환이미지에 가우시안함수를 적용하는 기능의 함수이다.
- opencv에서 제공하는 GaussianBlur 함수를 사용하여 입력 이미지 즉 src에 블러를 적용한 결과 이미지 즉 , src_blur에 저장한다.
- size(7,7)로 선언해서 이미지의 픽셀에 7x7크기로 블러를 적용한다. 마지막 0은 x 축 방향의 가우시안 표준 편차라고 한다.

실행 결과는 다음과 같다.

- 결과 사진



가우시안 필터 적용 후 변경점

- 가우시안 필터를 적용 후 기존 hsv 변환 이미지 보다 한층 더 노이즈가 줄어들었고 물체의 색을 추출한 하얀색 부분의 외곽이 보다 더 부드러게 되었다는 것을 확인할 수 있었다.
- 다만 필터 적용 과정에서 경계선이 뚜렷하게 드러나야 하는 부분이 오히려 모호해지는 상황이 발생하기도 하였다.

감사합니다.