

Cyan



//color2gray.c 파일을 이용하였습니다.

//for(i=0;i<(number_of_pixels);i++) 안의 내용을 수정하였으며, 수정된 부분만 첨부하였습니다.

//첨부된 내용에는 없지만 red, green, blue, c, m, y 를 float 형태로 선언하여 사용했습니다.

```
for (i = 0; i < (number_of_pixels); i++) {
    red = (float)buffer[3 * i] / 255;
    green = (float)buffer[3 * i + 1] / 255;
    blue = (float)buffer[3 * i + 2] / 255;
    //RGB' = RGB 0<=RGB<=1 만들기
    c = 1 - red;
    m = 1 - green;
    y = 1 - blue;
    //rgb -> cymk 변환공식 이용

    temp = 255 * c; //(256 * 3 * c) / 3
```

```
    CLIP(temp, 0, 255);  
  
    obuffer[i] = temp; // Generates output buffer for Brightness component  
  
}
```

Magenta



//color2gray.c 파일을 이용하였습니다.

//for(i=0;i<(number_of_pixels);i++) 안의 내용을 수정하였으며, 수정된 부분만 첨부하였습니다.

//첨부된 내용에는 없지만 red, green, blue, c, m, y 를 float 형태로 선언하여 사용했습니다.

```
for (i = 0; i < (number_of_pixels); i++) {
    red = (float)buffer[3 * i] / 255;
    green = (float)buffer[3 * i + 1] / 255;
    blue = (float)buffer[3 * i + 2] / 255;
//RGB' = RGB 0<=RGB<=1 값으로 만들기
    c = 1 - red; //cyan
    m = 1 - green; //magenta
    y = 1 - blue; //yellow

    temp = 255 * m; //(256 * 3 * m) / 3
    CLIP(temp, 0, 255);

    obuffer[i] = temp; // Generates output buffer for Brightness component
}
```

Yellow



//color2gray.c 파일을 이용하였습니다.

//for(i=0;i<(number_of_pixels);i++) 안의 내용을 수정하였으며, 수정된 부분만 첨부하였습니다.

//첨부된 내용에는 없지만 red, green, blue, c, m, y 를 float 형태로 선언하여 사용했습니다.

```
for ( i = 0; i < (number_of_pixels); i++) {
    red = (float)buffer[3 * i] / 255;
    green = (float)buffer[3 * i + 1] / 255;
    blue = (float)buffer[3 * i + 2] / 255;
//RGB' = RGB 0<=RGB<=1
    c = 1 - red;
    m = 1 - green;
    y = 1 - blue;
//RGB -> CMY 공식 이용
    temp = 255 * y; //(256 * 3 * y) / 3
    CLIP(temp, 0, 255);

    obuffer[i] = temp; // Generates output buffer for Brightness component
}
```

Black



//color2gray.c 파일을 이용하였습니다.

//for(i=0;i<(number_of_pixels);i++) 안의 내용을 수정하였으며, 수정된 부분만 첨부하였습니다.

//첨부된 내용에는 없지만 red, green, blue, c, m, y 를 float 형태로 선언하여 사용했습니다.

```
for (i = 0; i < (number_of_pixels); i++) {
    red = (float)buffer[3 * i] / 255;
    green = (float)buffer[3 * i + 1] / 255;
    blue = (float)buffer[3 * i + 2] / 255;
//RGB' = RGB 0<=RGB'<=1
    c = 1 - red;
    m = 1 - green;
    y = 1 - blue;
//RGB -> CYM 공식 이용
    //find black = min(c,m,y)
    k = 1; //k = black (초기화)
    if (c < m)
        k = c;
    else k = m;
    if (y < k)
```

```
        k = y;

        temp = 255 * k; //(256 * 3 * k)/ 3
        CLIP(temp, 0, 255);

        obuffer[i] = temp; // Generates output buffer for Brightness component
    }
```

Red



//color2gray.c 파일을 이용하였습니다.

//for(i=0;i<(number_of_pixels);i++) 안의 내용을 수정하였으며, 수정된 부분만 첨부하였습니다.

```
for (i = 0; i < (number_of_pixels); i++) {  
  
    buffer[3 * i + 1] = buffer[3 * i]; //green = red  
    buffer[3 * i + 2] = buffer[3 * i]; //blue = red  
  
    temp = (buffer[3 * i] + buffer[3 * i + 1] + buffer[3 * i + 2]) / 3; //  
Generates Brightness component ->make gray  
    CLIP(temp, 0, 255);  
  
    obuffer[i] = temp; // Generates output buffer for Brightness component  
  
}
```

Green



//color2gray.c 파일을 이용하였습니다.

//for(i=0;i<(number_of_pixels);i++) 안의 내용을 수정하였으며, 수정된 부분만 첨부하였습니다

```
for (i = 0; i < (number_of_pixels); i++) {
```

```
    buffer[3 * i] = buffer[3 * i+1]; // red = green  
    buffer[3 * i + 2] = buffer[3 * i+1]; //blue = green
```

```
    temp = (buffer[3 * i] + buffer[3 * i + 1] + buffer[3 * i + 2]) / 3; //  
Generates Brightness component ->make gray  
    CLIP(temp, 0, 255);
```

```
    obuffer[i] = temp; // Generates output buffer for Brightness component
```

```
}
```


Blue



//color2gray.c 파일을 이용하였습니다.

//for(i=0;i<(number_of_pixels);i++) 안의 내용을 수정하였으며, 수정된 부분만 첨부하였습니다.

```
for (i = 0; i < (number_of_pixels); i++) {
```

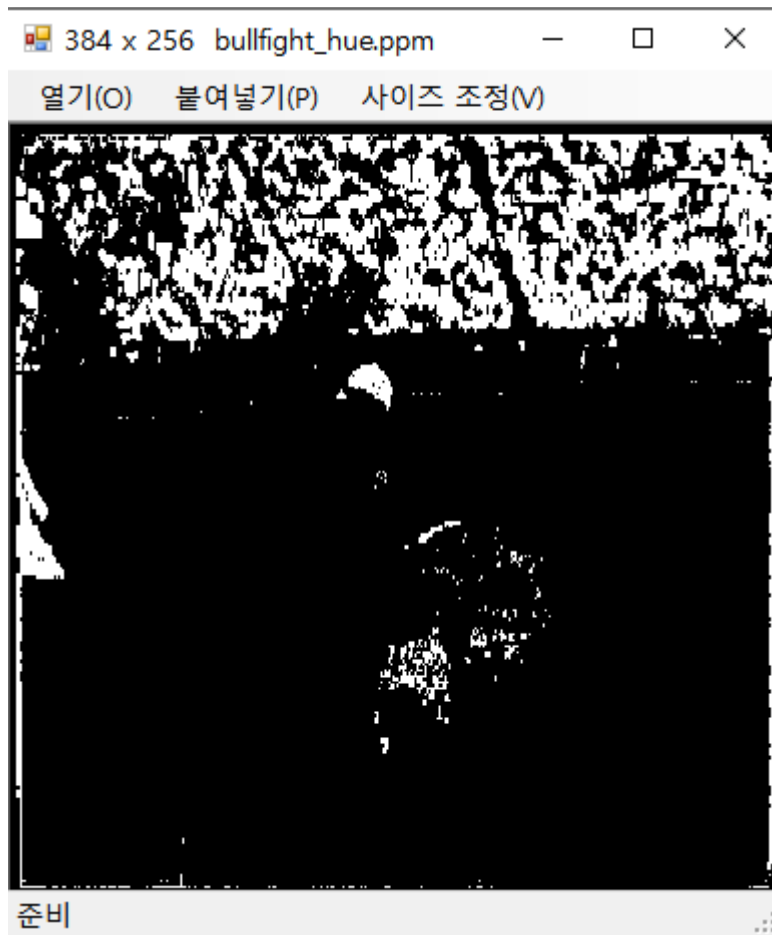
```
    buffer[3 * i] = buffer[3 * i+2]; //red = blue  
    buffer[3 * i + 1] = buffer[3 * i+2]; // green = blue
```

```
    temp = (buffer[3 * i] + buffer[3 * i + 1] + buffer[3 * i + 2]) / 3; //  
    Generates Brightness component ->make gray  
    CLIP(temp, 0, 255);
```

```
    obuffer[i] = temp; // Generates output buffer for Brightness component
```

```
}
```

Hue



//color2gray.c 파일을 이용하였습니다.

//for(i=0;i<(number_of_pixels);i++) 안의 내용을 수정하였으며, 수정된 부분만 첨부하였습니다.

```
float red, green, blue; //for cymk
float c, m, y;
float hue;
float angle = 0;
for (i = 0; i < (number_of_pixels); i++) {

    red = (float)buffer[3 * i] / 255;
    green = (float)buffer[3 * i + 1] / 255;
    blue = (float)buffer[3 * i + 2] / 255;
    min = 1;

    //각도 찾기
    angle = 0; //초기화
    angle = 0.5*(red - green + red - blue) / sqrt((red - green)*(red - green) +
(red - blue)*(green - blue));

    hue = acos(angle);
    if (blue < green) //green값이 blue보다 큰 경우
        hue = 360.0 - hue;
```

```
temp = hue;

CLIP(temp, 0, 255);

obuffer[i] = temp; // Generates output buffer for Brightness component
}
```

Saturation



//color2gray.c 파일을 이용하였습니다.

//for(i=0;i<(number_of_pixels);i++) 안의 내용을 수정하였으며, 수정된 부분만 첨부하였습니다.

```
float min; //find cmyk
float k; //black
float red, green, blue; //for cymk
float c, m, y;
float s; //saturation
for (i = 0; i < (number_of_pixels); i++) {

    red = (float)buffer[3 * i] / 255;
    green = (float)buffer[3 * i + 1] / 255;
    blue = (float)buffer[3 * i + 2] / 255;
    min = 1;

    //find min 최솟값 찾기
    if (red <= green)
        min = red;
    else //gree < red
        min = green;
    if (blue < min)
        min = blue;
```

```
s = 1.0 - (3.0 / (red + green + blue))*min; //공식이용  
temp = s * 255;
```

```
CLIP(temp, 0, 255);
```

```
obuffer[i] = temp; // Generates output buffer for Brightness component
```

```
}
```

Intensity



//color2gray.c 파일을 이용하였습니다.

//for(i=0;i<(number_of_pixels);i++) 안의 내용을 수정하였으며, 수정된 부분만 첨부하였습니다.

```
for (i = 0; i < (number_of_pixels); i++) {  
  
    temp = (buffer[3 * i] + buffer[3 * i + 1] + buffer[3 * i + 2]) / 3; //  
    Generates Brightness component ->make gray  
    //color2gray가 intensity 공식과 같아서(R+G+B/3) 추가할 코드가 없었습니다.  
    CLIP(temp, 0, 255);  
  
    obuffer[i] = temp; // Generates output buffer for Brightness component  
  
}
```