<보고서>

17011703 소프트웨어학과 홍예지

이 프로그램을 구성하기 위해서 메인함수와 void형 함수 4개를 구현해서 실행할 수 있도록 했다. Void형 함수에는 감마 코렉션 그래프를 구현하는 함수와 그림을 찍어내는 함수, 싸인함수 그래프를 구현하는 함수와 그림을 찍어내는 함수, 4개로 구상했다. 우선 main함수에서 이미지를 포인터로 받아줄 준비를 해준 뒤에 교수님께서 주신 실행파일로 내부 구조를 다듬었다. printf함수를 이용해서 프로그램의 기틀을 잡아준 뒤에 문자열로 받을 그림의 위치(주소)를 받아내었다. 이 때 주소를 잘못 쳤을 경우를 대비하여 확인해주는 if문을 작성해주었다. 그림이 그 위치에 없다면 "File not Found!"가 출력될 수 있도록 설계해주고 만약 있다면 while(반복문)문을 나갈 수 있게 해주었다. 무한루프를 돌게해서 그림의 주소를 정확히 입력할 때까지 계속 문자열로 받아줄 수 있게 해주었다. 그리고 그림을 로드해준다. 변수 num을 이용해서 함수를 고르게 하는데 1을 골랐을 때는 감마코렉션으로, 2를 골랐을 때는 sin함수로 갈 수 있게 if문을 이용해서 작성해준다.

먼저 1을 골랐을 때의 감마함수에 대해서 알아보기로 한다. 복사본(dst)를 생성해서 원본으로부터 크기를 맞춰주고 gamma값은 1.0으로 시작한다. Gamma correction의 공식은 g=f^r(감마) 이다. 원본에서 색을 뽑아서 이 색에 255를 나눠준후에 gamma의 값을 곱해준뒤, 다시 255를 곱해줘야하는데 이는 감마곡선이 0에서 1로 정의되기에, 입력할 때는 255로 나누어주고 모니터에 출력해줄 때는 다시 255를 곱해서 출력해주어야 한다. 이를 void형 함수로 구현해주었는데 자세히 설명해보자면, 원본, 복사본과 변화하는 gamma의 값을 인자로 받고 원본의 사이즈를 잰 후 x와y의 값에 제한을 줘서 색을 뽑아내서 부여하는 for문을 작성한다. for문의 시작은 y의 좌표로 시작하고 변수 f에서 x와 y의 지점에서의 원본 색을 f에 부여한 후에 복사본에 부여할 색 g를 선언한 후 색 3원소에 일일이 감마코렉션을 적용해준다. 식은 g.val[k]=pow(f.val[k]/255,gamma)*255;이다. 여기서 k는 0,1,2 색의 3원소를 의미하며 위에 설명 해주었다싶이 감마곡선이 0에서1까지 정의되기에 입력할 때는 255로 나누어주고 출력할 때는 다시 255를 곱해서 출력해준다. 그리곤 함수 안에서 출력해준다. 다시 main함수로 돌아와서 graph를 그려주는데 우선 그래프는 흰 창을 생성해서 거기에다가 검은색점을 출력해놓는 것이기 때문에 먼저 사이즈(250,250)에 맞춰서 창을 하나 생성한다. 사이즈는 실행파일에서 생성된 그래프의 크기에 맞추었다. 그리고 void형 인자가 graph그림과 gamma인 함수를 선언해서 내부를 구현한다. 우선 흰 창을 생성해야되기 때문에 그래프에 cvScalar함수를 이용해 흰색을 부여한다. (255,255,255)면 그래프 전체에 흰색을 부어준다. 실행파일을 살펴보면 gamma가 0보다 작아지면 그래프의 y좌표가 0이 된다. 그 외에는 그래프를 그려주어야하는데 x의 값의 변화에 따라 y의 좌표가 달라지게끔 해서 좌표마다 검은색 점을 찍어서 그래프를 구현했다. 우리가 구현하고 싶은 그래프의 모습과 opencv에서의 x,y좌표는 다르기 때문에 그래프를 구현하기 위해서는 좌표를 잘 계산해야 한다. 전체 세로길이는 250이기에 여기서 1을 뺀 수에서 x가 감마코렉션 수식에 들어간 값을 빼주면 y의 값을 완성해줄 수 있다. 여기에

cvScalar(0,0,0)인 검은색 점을 좌표마다 찍어서 그래프를 완성해준다. y좌표는 실수로 선언해주고 x의 좌표도 실수로 강제형변환을 해주어 계산해주어야 한다. 이렇게 첫 두 개의 창을 띄우게 된다. 다시 메인 함수로 돌아와서 키보드 조작을 해주는데 무한루프를 돌게 해서 키를 계속 입력받을 수 있게 해준다. 1을 눌렀을 때 gamma의 값을 증가시켜준뒤 다시 Gamma사진변환함수와 그래프함수를 호출해서 그림과 그래프를 갱신시켜준다. 그리곤 현재 상태를 출력해준다. 2를 눌렀을 경우에는 gamma의 값을 감소시켜서 다시 사진변환함수와 그래프함수를 호출해서 창에 변화를 준다. Gamma가 0보다 작아지는 경우 또한 생기는데 이 때는 실수 0에서 멈추게 한다. 그리고 'Q'를 눌렀을 때는 나갈 수 있게 식을 구현해준다. 이 때 exit함수를 이용해서 탈출하게 해준다.

sin함수를 구현해주기 위해서 우선 <math.h>헤더파일을 인클루드 시켜준 뒤 활용해주었다. sin함수를 구현하는 else문으로 넘어가면 gamma함수를 구현해줄 때와의 형식은 유사하다. 처음의 sine값을 정해준 뒤에 사인값이 들어간 그림과 그래프를 띄우는 형식이다. Void형 함수를 깊숙이 살펴보자면 gamma함수 구현과 형식은 같지만 그 원본의 색을 따서 복사본에 색을 변환해서 넣어줄 때의 공식이 다르다. 나는 이 sin함수에 대해서만 알기 때문에 구현했을 때 내가 생각해왔던 것과 사진이 다르게 나와서 놀랬다. 하지만 주어진 실행프로그램을 실행 시키면서 정말 수많은 시도 끝에 식을 구현했다. 입력값에 변화하는 변수의 값 sine에 (2*3.14/249)를 곱해주었다. 그리고 출력하기 위해서 255를 곱해서 출력해주었다. 그래프를 그릴 때 가로의 크기가 250인데 이를 구현하기 위해서(주기가 250이다)(2*3.14/249)로 가로 주기를 맞추어 주었고 이에 변화하는 sine값과 원본의 삼원소의 값을 곱해주고 출력을 위해 255을 곱해서 g에 넣어주어 복사본을 출력해주었다. 사인함수 그래프를 생성해줄 때는 새로운 하얀색의 창을 생성해준 뒤에 gamma그래프와 마찬가지로 변화하는 x의 값에 대해서의 y좌표의 식을 세워서 해당하는 점을 검은 점으로 출력해주었다. 처음의 sine이 1일 때의 값이 때를 생각해서 주기는 가로의 길이인 250으로 주기를 맞추어서 계산해주었다. 또한 이 값의 최댓값은 세로길이의 절반이 되어야 하므로 124로 정해서 사인함수에 곱해주었고 이의 값을 그대로 대칭이동 시켜서 y의 좌표의 값을 결정해주었다. 사인 함수의 함수 2개를 구현해준 후 main함수로 돌아와서 무한루프를 돌려서 키보드에 따른 세부사항을 구현해준다. 1을 눌렀을 때 sine값이 증가하게 해준 후 두 개의 함수를 호출시켜주고 현재 상태 출력해준다. 2를 눌렀을 때는 제한사항이 있다. 주어진 실행프로그램을 돌려보았을 때 sine의 값이 0.5에서 내려가지 않는 것을 확인할 수 있는데 이에 맞춰서 2를 계속 눌러도 0.5로 값을 유지할 수 있게 값을 설정해준다.

이로써 변환함수에 대해서 그래프도 그려보고 함수에 따른 그림의 변화모습을 알아보았는데 직접 그래프를 식을 세워서 그려보니 어떠한 원리로 프로그램이 돌아가는 지 정확히 알게 되었다.