멀티미디어정보처리

HW2

컴퓨터공학과

2013043401

**정지성**

제출일:17.04.12

과제에 들어가기 앞서서

코드의 가독성을 높이기 위해

파일위치를 String 변수로 선언하였고,(변수명은 대문자로 선언하였습니다.)

반복적인 코드와 기능별로 함수를 만들었습니다.

void m\_show(Mat img, String saveName){

winNum++; ->전역변수 입니다.

**namedWindow**("Window"+winNum,CV\_WINDOW\_AUTOSIZE);

**imshow**("Window"+winNum,img);

String locate = saveName+".jpg";

**imwrite**(locate, img);

}

void m\_show(Mat,String);

void m\_Blur(Mat,float);

void m\_Shar(Mat,float,float,float);

void m\_Nois(Mat);

void m\_Edge(Mat);

**예시)**

String LYALC\_D = "D:\\실습자료\\hw1\\lya-lc-dark.bmp";

String LYALC\_X = "D:\\실습자료\\hw1\\lya-lc-안되는예.bmp";

Mat img = imread(LYALC,CV\_32FC3);

Blurring, Sharpening, Noise reduction, Edge detection

그러므로 아래 내용부터는 함수만 보여드리겠습니다.

**Blurring**

**Code**

void m\_Blur(Mat img,float A){

Mat st = cvCreateMat(img.rows,img.cols,CV\_8UC3);

float mask[] = {A,A,A, A,A,A, A,A,A};

Vec3b color;

int sum[3];

int cnt;

for(int y=1;y<img.rows-1;y++){ ->가장자리에 있는 픽셀 때문에

for(int x=1;x<img.cols-1;x++){ 1부터 시작한다.

for(int c=0;c<3;c++)

sum[c]=0;

cnt =0;

for(int h=y-1;h<=y+1;h++){ ->배열의 수 만큼 반복한다.

for(int w=x-1;w<=x+1;w++){

color = img.at<Vec3b>(h,w); ->영상의 픽셀을 가져온다.

sum[0]+= color[0]\*mask[cnt]; ->각 픽셀의 RGB 값에 설정한 값을

sum[1]+= color[1]\*mask[cnt]; 곱한다.

sum[2]+= color[2]\*mask[cnt];

cnt++;

}

}

for(int c=0;c<3;c++){

if(sum[c]<0) ->예외처리

sum[c]=0;

else if(sum[c]>255)

sum[c]=255;

st.at<Vec3b>(y,x)[c]=sum[c]; ->Mat 변수 st에 값을 대입한다.

}

}

}

m\_show(st,SAVE+"\_BLUR");

}

**Blurring**

**Image**

|  |  |  |
| --- | --- | --- |
| **원본** | **3X3** | **5X5** |
| **C:\Users\jisung\AppData\Local\Microsoft\Windows\INetCache\Content.Word\save(0).jpg** | **C:\Users\jisung\AppData\Local\Microsoft\Windows\INetCache\Content.Word\save_BLUR_0(1).jpg** | **C:\Users\jisung\AppData\Local\Microsoft\Windows\INetCache\Content.Word\save_BLUR5_0(2).jpg** |
| **C:\Users\jisung\AppData\Local\Microsoft\Windows\INetCache\Content.Word\save(0).jpg** | **C:\Users\jisung\AppData\Local\Microsoft\Windows\INetCache\Content.Word\save_BLUR5_0(1).jpg** | **C:\Users\jisung\AppData\Local\Microsoft\Windows\INetCache\Content.Word\save_BLUR_0(2).jpg** |
| **C:\Users\jisung\AppData\Local\Microsoft\Windows\INetCache\Content.Word\save(0).jpg** | **C:\Users\jisung\AppData\Local\Microsoft\Windows\INetCache\Content.Word\save_BLUR5_0(1).jpg** | **C:\Users\jisung\AppData\Local\Microsoft\Windows\INetCache\Content.Word\save_BLUR_0(2).jpg** |
| **C:\Users\jisung\AppData\Local\Microsoft\Windows\INetCache\Content.Word\save(0).jpg** | **C:\Users\jisung\AppData\Local\Microsoft\Windows\INetCache\Content.Word\save_BLUR_0(1).jpg** | **C:\Users\jisung\AppData\Local\Microsoft\Windows\INetCache\Content.Word\save_BLUR5_0(2).jpg** |
| **C:\Users\jisung\AppData\Local\Microsoft\Windows\INetCache\Content.Word\save(0).jpg** | **C:\Users\jisung\AppData\Local\Microsoft\Windows\INetCache\Content.Word\save_BLUR(1).jpg** | **C:\Users\jisung\AppData\Local\Microsoft\Windows\INetCache\Content.Word\save_BLUR5_0(2).jpg** |

**Blurring**

**Discussion**

Blurring은 정지영상을 흐릿하게 만드는 효과가 있다.

홀수 N 으로 N X N 행렬을 만들어 convolution한다.

이웃한 픽셀끼리 평균을 내어 동일한 값을 갖게 만들어 픽셀 간 값 차이를 줄인다.(경계선의 줄인다.) 근접 픽셀간 값이 동일하기 때문에 전체적인 영상이 흐릿하게 보이는 것이다.

Convolution을 할 때 로 한다. 예를 들어 3X3 인 경우 1/9 이고 5X5 인 경우 1/25를 곱하여 Blurring 효과를 얻는다. 즉 마스크의 총 합이 1이 되도록 만드는 것이다.

이 곱셈 값을 바꿔도 근접 픽셀 값은 같기 때문에 흐릿한 효과는 갖을 수 있지만 영상의 명도가 달라진다. 값이 기준보다 높으면 전체적으로 값을 올리는 것이기 때문에 영상이 밝아지고 낮으면 어두워진다.

(5X5에 1/5 와 1/125를 곱함)

Blurring은 convolution 크기가 클수록 더 강하게 나타난다.

더 많은 픽셀끼리 값이 비슷해 지기 때문에 전체적으로 더 흐릿해 지는 것이다. 위의 비교 영상에서 보이는 것처럼 3X3보다 5X5가 더 흐릿하다.

추가적으로 구현 시 관련하여 N x N 의 mask를 사용하기 위해서는 for문에 2/N 을 시작과 종료 조건으로 사용하여야 한다.

Blurring 은 smoothing 이라고도 부르고 위의 개념을 응용하여 아래 나올 noise 제거를 효과적으로 할 수 있다. 이에 관련된 내용은 아래에서 언급하겠다.

**sharpening**

**Code**

void m\_Shar(Mat img,float A,float B,float C){

Mat st = cvCreateMat(img.rows,img.cols,CV\_8UC3);

float mask[] = {A,B,A,

B,C,B,

A,B,A};

Vec3b color;

int sum[3];

int cnt;

for(int y=1;y<img.rows-1;y++){

for(int x=1;x<img.cols-1;x++){

for(int c=0;c<3;c++)

sum[c]=0;

cnt =0;

for(int h=y-1;h<=y+1;h++){

for(int w=x-1;w<=x+1;w++){

color = img.at<Vec3b>(h,w);

sum[0]+= color[0]\*mask[cnt];

sum[1]+= color[1]\*mask[cnt];

sum[2]+= color[2]\*mask[cnt];

cnt++;

}

}

for(int c=0;c<3;c++){

if(sum[c]<0)

sum[c]=0;

else if(sum[c]>255)

sum[c]=255;

st.at<Vec3b>(y,x)[c]=sum[c];

}

}

}

m\_show(st,SAVE+"\_Shar");

}

**sharpening**

**Image**

|  |  |  |
| --- | --- | --- |
| **원본** | **Mask1** | **Mask2** |
| **C:\Users\jisung\AppData\Local\Microsoft\Windows\INetCache\Content.Word\save(0).jpg** | **C:\Users\jisung\AppData\Local\Microsoft\Windows\INetCache\Content.Word\save_Shar-1(2).jpg** | **C:\Users\jisung\AppData\Local\Microsoft\Windows\INetCache\Content.Word\save_Shar0(1).jpg** |
| **C:\Users\jisung\AppData\Local\Microsoft\Windows\INetCache\Content.Word\save(0).jpg** | **C:\Users\jisung\AppData\Local\Microsoft\Windows\INetCache\Content.Word\save_Shar0(1).jpg** | **C:\Users\jisung\AppData\Local\Microsoft\Windows\INetCache\Content.Word\save_Shar-1(2).jpg** |
| **C:\Users\jisung\AppData\Local\Microsoft\Windows\INetCache\Content.Word\save(0).jpg** | **C:\Users\jisung\AppData\Local\Microsoft\Windows\INetCache\Content.Word\save_Shar0(1).jpg** | **C:\Users\jisung\AppData\Local\Microsoft\Windows\INetCache\Content.Word\save_Shar-1(2).jpg** |
| **C:\Users\jisung\AppData\Local\Microsoft\Windows\INetCache\Content.Word\save(0).jpg** | **C:\Users\jisung\AppData\Local\Microsoft\Windows\INetCache\Content.Word\save_Shar0(1).jpg** | **C:\Users\jisung\AppData\Local\Microsoft\Windows\INetCache\Content.Word\save_Shar-1(2).jpg** |
| **C:\Users\jisung\AppData\Local\Microsoft\Windows\INetCache\Content.Word\save(0).jpg** | **C:\Users\jisung\AppData\Local\Microsoft\Windows\INetCache\Content.Word\save_Shar0(1).jpg** | **C:\Users\jisung\AppData\Local\Microsoft\Windows\INetCache\Content.Word\save_Shar-1(2).jpg** |

**sharpening**

**Discussion**

Sharpening 은 영상을 선명하게 보여주는 효과가 있다.

Blurring 과는 반대로 영상의 경계선 값을 더 높여 선을 날카롭게 만든다.

영상에 mask convolution을 통해 구현할 수 있다. Mask의 가운데에 있는 픽셀에 주변 픽셀을 수치만큼 곱하여 대입한다. 곱셈 연산 후 덧셈연산을 수행하기 때문에 근접 픽셀 간 수치의 차가 높을수록 결과값이 더 높아져서 경계 값, 윤곽선이 도드라져 보이는 것이다.

경계값을 높이기 위해서는 mask 값을 설정하는 것이 중요하다.

위의 영상에서 mask는 이렇게 설정하였다.

|  |  |  |
| --- | --- | --- |
| 0 | -1 | 0 |
| -1 | 5 | -1 |
| 0 | -1 | 0 |

|  |  |  |
| --- | --- | --- |
| -1 | -1 | -1 |
| -1 | 9 | -1 |
| -1 | -1 | -1 |

Mask1 Mask2

Mask 의 수치를 보면, 단순히 가운데 값이 높은 것이 아니라 전체의 합이 1이 된다.

Mask의 총 합이 1이 아닌 경우 영상의 결과가 크게 차이가 난다.

합이 0인 경우를 보면(5번 영상)

|  |  |  |
| --- | --- | --- |
| -1 | -1 | -1 |
| -1 | 8 | -1 |
| -1 | -1 | -1 |



이렇게 윤곽선만 찾기 위해서는 합을 0으로 설정하면 된다.

합을 음수로 설정하면 이렇게 윤곽선이 잘 보이지 않는다. (더 내리면 검은색 화면)

|  |  |  |
| --- | --- | --- |
| -1 | -1 | -1 |
| -1 | 8 | -1 |
| -1 | -1 | -1 |



총 합을 3으로 해 보았다.

|  |  |  |
| --- | --- | --- |
| -1 | -1 | -1 |
| -1 | 11 | -1 |
| -1 | -1 | -1 |



이렇게 윤곽은 보이지만 기존 영상보다 명도가 더 높아진다.

그러므로 기존 영상에 선명한 효과를 주기 위해서는 합이 1이 되도록 설정하여야 한다.

픽셀값이 선명한 효과를 얻기 위해 전체합이 1이상이여야 한다.

**Noise reduction**

**Code**

중간값

void m\_Nois\_M(Mat img){

Mat st = cvCreateMat(img.rows,img.cols,CV\_8UC1);

int sort\_array[9];

int cnt=0;

for(int y=1;y<img.rows-1;y++){

for(int x=1;x<img.cols-1;x++) {

cnt = 0;

for(int h=y-1;h<=y+1;h++){

for(int w=x-1;w<=x+1;w++){

sort\_array[cnt]= img.at<uchar>(h,w);

cnt++;

}

}

vector<int>myvector(sort\_array,sort\_array+9);

sort(myvector.begin(),myvector.end());

st.at<uchar>(y,x) = myvector.at(4);

}

}

m\_show(st,SAVE+"\_Med");

}

void m\_Nois\_A(Mat img, float A){

Mat st = cvCreateMat(img.rows,img.cols,CV\_8UC1);

float mask[]={A,A,A,A,A,A,A,A,A};

Vec3b color;

int sum,cnt;

for(int y=1;y<img.rows-1;y++){

for(int x=1;x<img.cols-1;x++){

sum = 0;

cnt = 0;

for(int h=y-1;h<=y+1;h++){

for(int w=x-1;w<=x+1;w++){

sum += img.at<uchar>(h,w)\*mask[cnt];

cnt++;

}

}

st.at<uchar>(y,x) = sum;

}

}

m\_show(st,SAVE+"\_Ave");

}

**Noise reduction**

**Image**

**Noise reduction**

**Discussion**

노이즈에 대한 분석

마스크

기타 알고리즘

**Edge detection**

**Code**

**Edge detection**

**Image**

**Edge detection**

**Discussion**

<http://webnautes.tistory.com/1044>

<http://blog.naver.com/PostView.nhn?blogId=wlsgk123123&logNo=10093294588&redirect=Dlog&widgetTypeCall=true>