

신호처리

비트맵 파일 영상처리

10

DeepThought | 2010.08.31. 02:28

앱으로 보기

영상처리를 하겠노라고 떠들어대다가 여기저기서 도움을 많이 받았습니다. 지식인에도 물어봤고 책도 많이 찾아봤고 다른사람 블로그에도 많이 다녔었죠. 제일 처음에 만들었던 코드와는 조금 달라졌어요. 구조체를 써서 좀 알아보기 쉽다고나 할까...? 방금 전까지도 하다하다 안되어서 '그냥 때려치울까'하는 생각도 했었습니다. 그런데 파일을 쓸때 wb를 w+로 바꾸니까 되더라구요. 프로그래밍은 어디서 작은것 하나 틀려서 전체가 돌아가지 않으면 고쳐도 짹짹해요. 내가 개념을 모르는것 같기는 한데 이건 운이 없어서 뺄것 같기도 하고...

뭐 여튼 설명하자면 이걸 Lenna.bmp파일을 읽어들여서 흑백변환, 역상변환, 밝게, 어둡게 변환하는 수행을 하는게 목적인 프로그램입니다. C언어를 배우긴 했는데 저학년이라 기본문법만 배워서 MFC같은것은 모릅니다. 처음에 참고한 영상처리 책들은 MFC를 쓰길래 저도 생각없이 따라 썼더니 자꾸 함수를 그냥 사용해서 원리도 모르고 그냥 사용하게 되더라구요. 그래서 재미도 없고 공부도 안될 것 같기에 그냥 알고리즘을 단순하게 C언어만으로 구현했습니다. 사실 영상처리라고 할것도 별로 없을만큼 단순합니다. BMP파일의 구조와 기타 다른 자료들은 동일 카테고리에 포스팅 해 놓았으니 참고하실 분들은 보십시오.

(다른 변환들도 있습니다. <http://cmg0404.blog.me/150097098318> , <http://cmg0404.blog.me/150097159381>)

사진 1. 프로그램 실행 모습.

화면에 나온대로 하면 됩니다. 만약 흑백변환을 하고싶다 하면 숫자 1을 키보드에서 누르고 엔터를 치면 되죠.

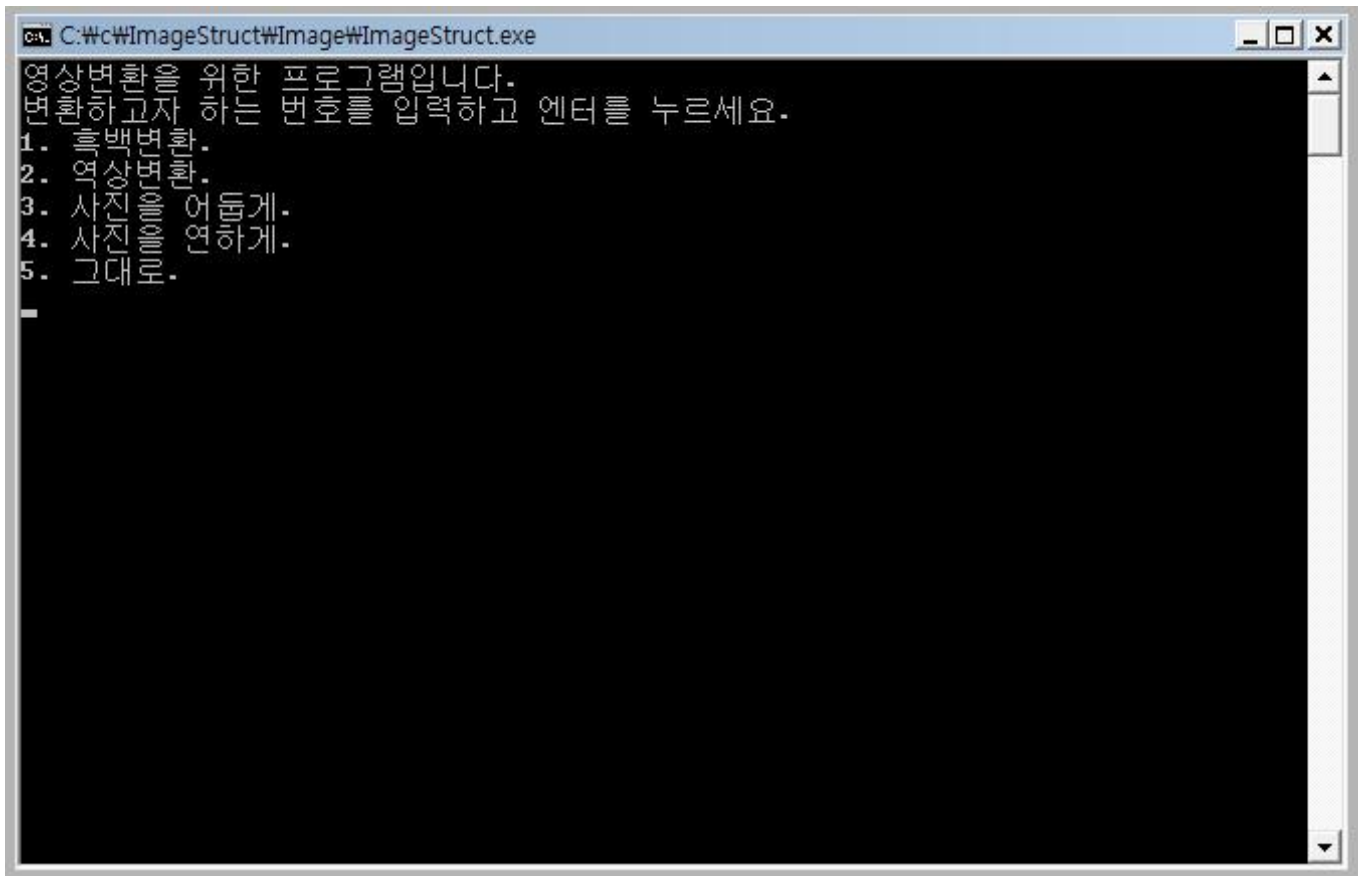


사진 2. 사랑스런 Lenna의 모습(-_-;;). 이 사진은 영상처리를 하는 모든 책에서 예제로 쓰고 있는 사진입니다. 이것저것 시험해 볼만한 특성이 골고루 녹아들어가 있어서 테스트 용으로 딱이라고 하더군요. 사진의 출처는 70년대 플레이보이(...)라네요. 원본은 누드사진인데 어떤 개발자(?)가 이사람 누드가 너무 좋았는지 영상처리 분야를 개척하면서 이 사진을 자기 논문에 써서 후대의 다른사람들도 이 사진을 쓰게 되었다는. 여튼 이 사진을 바탕으로 하여 변환을 수행합니다.



사진 2. 흑백변환한 사진. 아까 위의 시꺼먼 화면에서 숫자 1을 입력하면 결과물이 이렇게 나옵니다.



사진 3. 역상변환 사진. 아까 위의 시꺼먼 화면에서 숫자 2을 입력하면 결과물이 이렇게 나옵니다.



사진 4. 밝게 만든 사진. 아까 위의 시꺼먼 화면에서 숫자 3을 입력하면 결과물이 이렇게 나옵니다. 원본사진보다 약간 밝아졌죠? 그런데 사진에 아래의 그림과 같이 어색한 청색이 들어가 있음을 볼 수 있습니다. 이건 왜그러는고 하니... 비트맵 파일에서 한 픽셀의 색상을 저장하는 구조체에 보면 Red, Green, Blue 이렇게 세가지 정보가 저장이 되어있는데 각각의 정보가 처음에 파일에서 메모리로 읽어들여질 때 unsigned char형으로 저장되는데 저는 int형이라고 착각을 하고 코드를 짜서 그렇더라구요.

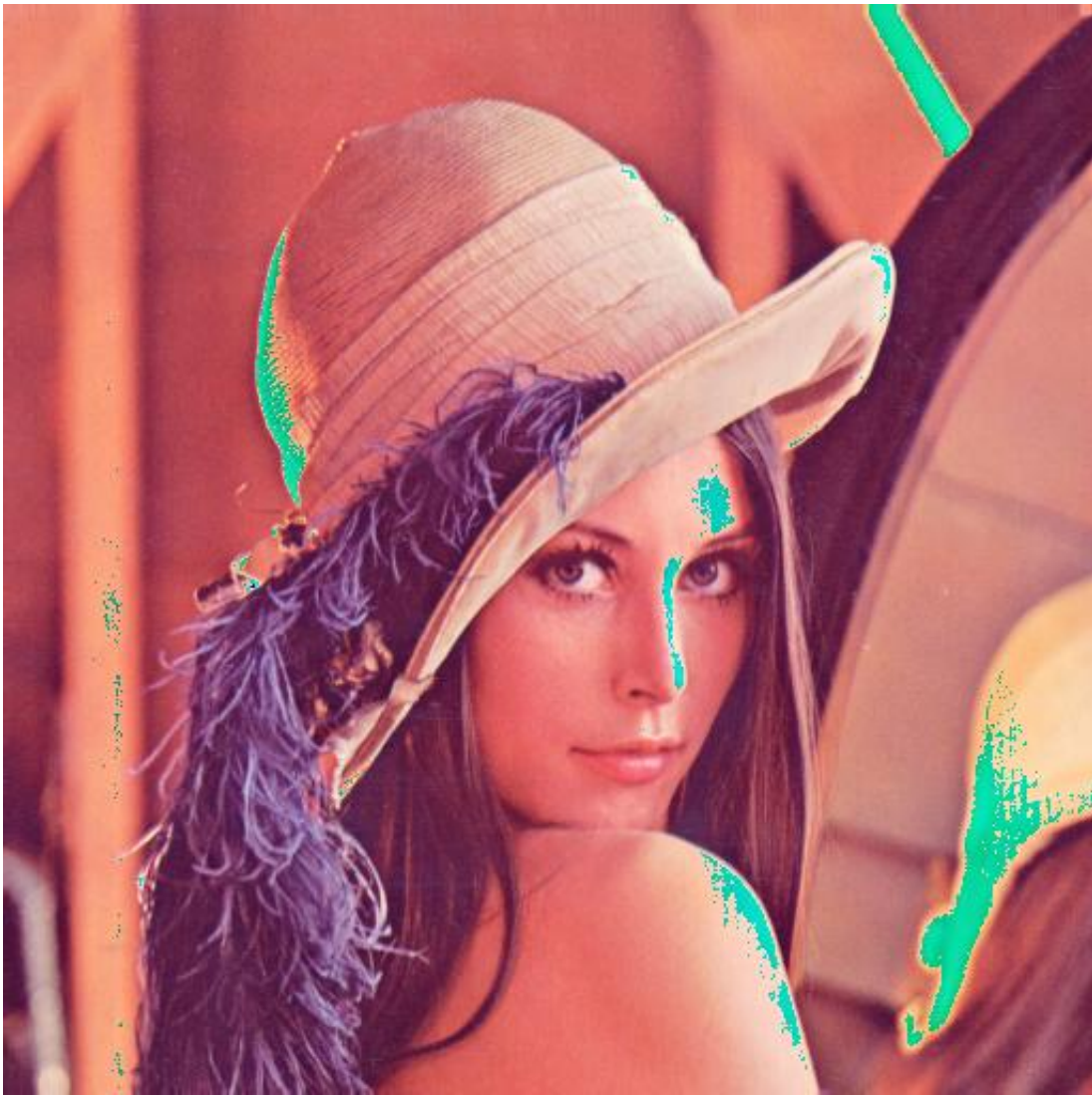


사진 5. 고친 코드로 만든 사진. 어색한 청색이 없어지고 정상적으로 밝게 되었습니다. 그리고 좀 더 밝게 만들도록 고쳤어요. 위에것은 밝은건지 구분이 잘 안가서.

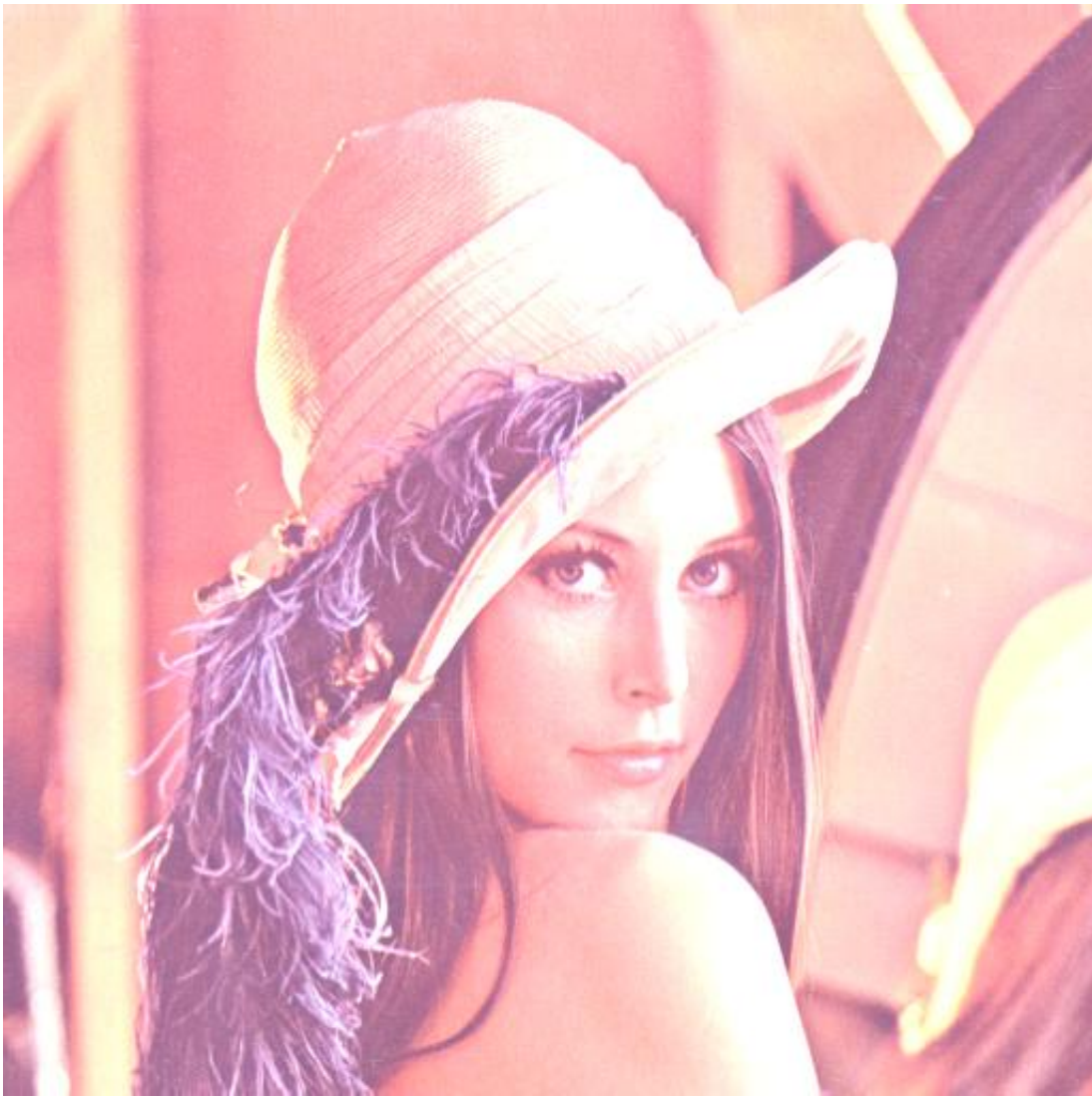


사진 6. 어둡게 만든 사진. 아까 위의 시꺼먼 화면에서 숫자 4을 입력하면 결과물이 이렇게 나옵니다. 원본사진보다 약간 어두워졌죠? 어두워진 정도가 미미해서 잘 안보일 수 있습니다. 마찬가지로 사진의 곳곳이 깨져있죠. 비트맵 파일에서 한 픽셀의 색상을 저장하는 구조체에 보면 Red, Green, Blue 이렇게 세가지 정보가 저장되어있는데 각각의 정보가 처음에 파일에서 메모리로 읽어들여질 때 unsigned char형으로 읽어져서 음수의 값을 가질수가 없더라구요. 이걸 생각안하고 계속 왜이렇게 어색한 청색이 나오는지 헤메었는데 이젠 알겠습니다.

예를들어 아래와 같은 코드에서 `pix[x].rgbBlue`는 int형이 아니라서 if문 안에 있는 `pix[x].rgbBlue<=0` 문장이 제 역할을 못하게 되는거죠.

```
pix[x].rgbBlue=pix[x].rgbBlue-10;
```

```
if(pix[x].rgbBlue<=0)    pix[x].rgbBlue=0;
```

이 코드를 고쳐서

```
v=(int)pix[x].rgbBlue-100;
```

```
if(v<=0) v=0;    pix[x].rgbBlue=v;
```

이렇게 고치면 됩니다.



사진 7. 고친 코드로 만든 사진. 위의 사진보다 조금 더 어둡게 하여 구분이 잘 가도록 했습니다.



첨부파일 사용법

1. 파일을 받고 압축을 풉니다.
2. ImageStruct.exe파일을 실행시킵니다.
3. 원하는 변환에 해당하는 숫자를 누르고 엔터!
4. 압축을 푼 폴더를 다시 보면 변환된 사진이 생긴걸 볼 수 있습니다.

```
#include
```

```
#include
```

```
#pragma pack(1)
```

```
typedef unsigned char  BYTE;
```

```
typedef unsigned short WORD;
```

```
typedef unsigned long  DWORD;
```

```
typedef long           LONG;
```

```
typedef struct _BITMAPFILEHEADER {
```

```
WORD    bfType;
DWORD   bfSize;
WORD    bfReserved1;
WORD    bfReserved2;
DWORD   bfOffBits;
} BITMAPFILEHEADER;
```

```
typedef struct _BITMAPINFOHEADER {
    DWORD   biSize;
    LONG    biWidth;
    LONG    biHeight;
    WORD    biPlanes;
    WORD    biBitCount;
    DWORD   biCompression;
    DWORD   biSizeImage;
    LONG    biXPelsPerMeter;
    LONG    biYPelsPerMeter;
    DWORD   biClrUsed;
    DWORD   biClrImportant;
} BITMAPINFOHEADER;
```

```
typedef struct _RGBTRIPLE {
    BYTE   rgbtBlue;
    BYTE   rgbtGreen;
    BYTE   rgbtRed;
} RGBTRIPLE;
```

```
void make_gray(RGBTRIPLE *pix, BITMAPINFOHEADER *bmih);
void make_reverse(RGBTRIPLE *pix, BITMAPINFOHEADER *bmih);
void make_thicker(RGBTRIPLE *pix, BITMAPINFOHEADER *bmih);
void make_thinner(RGBTRIPLE *pix, BITMAPINFOHEADER *bmih);
```

```
int main()
{
    BITMAPFILEHEADER bmfh;
    BITMAPINFOHEADER bmih;
    RGBTRIPLE *pix;
    int a, x;
    FILE *fp;
    FILE *ck;
    FILE *wt;
```

```
//-----파일 열기-----//
if((fp=fopen("Lenna.bmp", "rb"))==NULL)
```

```

{
    printf("file is not opened qtWn");
    return 1;
}
//printf("파일 열림WnWn");
//-----읽어들이기-----//
    fread(&bmfh, sizeof(BITMAPFILEHEADER), 1, fp); // 파일헤더 읽어들이
    fread(&bmih, sizeof(BITMAPINFOHEADER), 1, fp); // 정보헤더 읽어들이
    pix = (RGBTRIPLE*)malloc(bmih.biSizeImage); // 이미지 공간 할당
    fread(pix, 1, bmih.biSizeImage, fp); // 이미지 데이터 읽어들이
    fclose(fp);
//printf("Wnread completeWn");
//-----읽은 그대로 txt 쓰기-----//
    fp=fopen("checking.txt", "w+");
    fwrite(&bmfh, sizeof(BITMAPFILEHEADER), 1, fp); // 파일헤더 저장
    for(x=0;x<(bmih.biWidth*bmih.biHeight);x++)
        fprintf(fp, "[%d]: (%x)(%x)(%x)Wt", x, pix[x].rgbtRed, pix[x].rgbtGreen, pix[x].rgbtBlue); //헤더만 먼저 읽어들이기 때문에 헤더 다음에 다시 읽어들이는 pix[]는 헤더 다음부터
        //입력을 하는듯. 결국 pix[]배열은 배열 54개를 제외한 55번째부터 입력되는것.
    fclose(fp);
//printf("Wnwt completeWn");
//-----읽은 그대로 다시 bmp 쓰기-----//
/* ck=fopen("checking.bmp", "wb");
    fwrite(&bmfh, sizeof(BITMAPFILEHEADER), 1, fp); // 파일헤더 저장
    fwrite(&bmih, sizeof(BITMAPINFOHEADER), 1, fp); // 정보헤더 저장
    fwrite(pix, 1, bmih.biSizeImage, fp); // 이미지 데이터 저장
    free(pix);
    fclose(fp);
*/

//프로그램 화면
printf("영상변환을 위한 프로그램입니다.Wn변환하고자 하는 번호를 입력하고 엔터를 누르세요.Wn1. 흑백변환.Wn2. 역상변환.Wn3. 사진을 어둡게.Wn4. 사진을 연하게.Wn5. 그대로.Wn");
scanf("%d", &a);
switch(a)
{
case 1://흑백변환
{
    make_gray(pix, &bmih); //배열을 함수의 전달인자로 넘길시에는 (*pix)이게 아니라 그냥 (pix)로 써야한다
    wt=fopen("gray.bmp", "w+"); //아오씨 wb라고 쓰니 안되고 w+라 쓰니 되는구나. 왜 그런지는 잘 모르겠다만...

```

```

fwrite(&bmfh, sizeof(BITMAPFILEHEADER), 1, wt); // 파일헤더 저장
fwrite(&bmih, sizeof(BITMAPINFOHEADER), 1, wt); // 정보헤더 저장
fwrite(pix, 1, bmih.biSizeImage, wt);          // 이미지 데이터 저장
free(pix);
fclose(wt);
break;
}
case 2://역상변환
{
    make_reverse(pix, &bmih);
    wt=fopen("reverse.bmp", "wb");
    fwrite(&bmfh, sizeof(BITMAPFILEHEADER), 1, wt); // 파일헤더 저장
    fwrite(&bmih, sizeof(BITMAPINFOHEADER), 1, wt); // 정보헤더 저장
    fwrite(pix, 1, bmih.biSizeImage, wt);          // 이미지 데이터 저장
    free(pix);
    fclose(wt);
    break;
}
case 3://어둡게 변환
{
    make_thicker(pix, &bmih);
    wt=fopen("thicker.bmp", "wb");
    fwrite(&bmfh, sizeof(BITMAPFILEHEADER), 1, wt); // 파일헤더 저장
    fwrite(&bmih, sizeof(BITMAPINFOHEADER), 1, wt); // 정보헤더 저장
    fwrite(pix, 1, bmih.biSizeImage, wt);          // 이미지 데이터 저장
    free(pix);
    fclose(wt);
    break;
}
case 4://밝게 변환
{
    make_thiner(pix, &bmih);
    wt=fopen("thiner.bmp", "wb");
    fwrite(&bmfh, sizeof(BITMAPFILEHEADER), 1, wt); // 파일헤더 저장
    fwrite(&bmih, sizeof(BITMAPINFOHEADER), 1, wt); // 정보헤더 저장
    fwrite(pix, 1, bmih.biSizeImage, wt);          // 이미지 데이터 저장
    free(pix);
    fclose(wt);
    break;
}
case 5://그냥출력
{
    wt=fopen("non.bmp", "wb");
    fwrite(&bmfh, sizeof(BITMAPFILEHEADER), 1, wt); // 파일헤더 저장

```



```
fwrite(&bmih, sizeof(BITMAPINFOHEADER), 1, wt); // 정보헤더 저장
fwrite(pix, 1, bmih.biSizeImage, wt);          // 이미지 데이터 저장
fclose(wt);
```

[로그인](#)[전체 앱](#)[PC버전](#)[전체 서비스](#)

© NAVER Corp. | 블로그 고객센터 | 오류신고