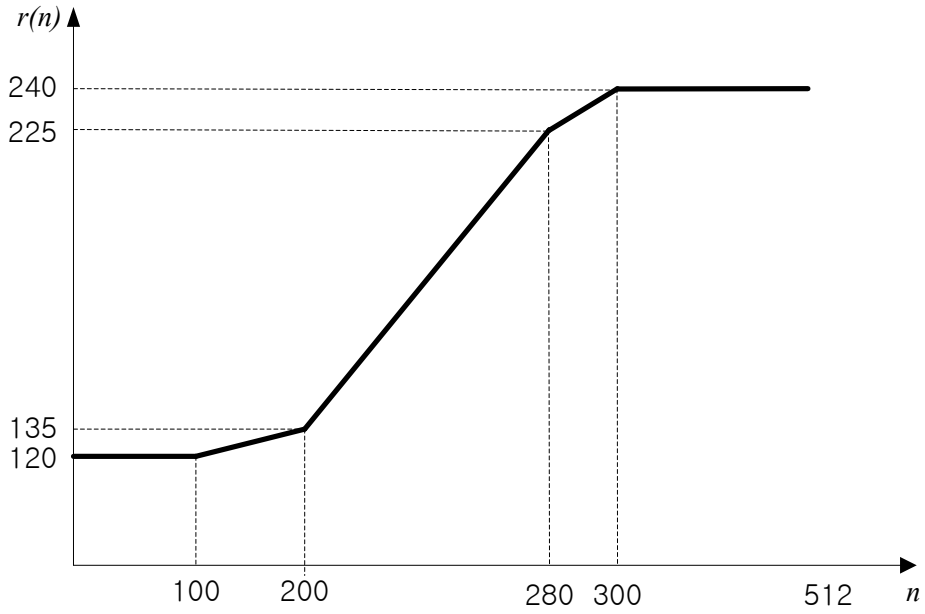
영상처리

HW 01

2014253005 박세현

1. Generate a 512x512 raw image, each row of which is a smooth ramp r(n) as shown in the figure below. Display the raw image on a PC monitor using Paint Shop Pro and designate the location of two Mach bands.



512 * 512 크기를 가진 unsigned char 타입의 일차원 포인터 배열을 만들고, 구간에 따라 위 그래프의 값을 위치에 맞게 대입한다. Unsigned char 타입이므로 변화하는 구간이 실제로는 그래프처럼 선형으로 증가하지 않고 계단형으로 증가한다. 해당 포인터 배열을 완성하고 raw 파일로 저장한다.

이렇게 생성된 파일을 뷰어로 열어보면 왼쪽에서 어두운 회색에서 오른쪽으로 갈수록 점점 밝아지는 흰색에 가까운 회색의 모습을 하는 파일을 확인할 수 있다. 여기서 경계선 부근에서 색이 더 진하게 보이는데 이를 Mach band effect라고 한다.

Raw 파일은 가공을 하지 않은 '날것의' 파일이므로 이러한 방식으로 저장될 수 있다. 동시에 이렇게 생성된 파일은 다른 파일들과 달리 헤더가 없어 일반적인 방법으로는 파일을 열어볼 수 없

고 뷰어를 통하더라도 이미지의 크기나 컬러 형식을 알고 있어야 제대로 열어볼 수 있다.

```
void generate_raw_data(BYTE *image, bool flag) {
  BYTE val;
  for (int i = 0; i < MAX; i++) {
    for (int j = 0; j < MAX; j++) {
      if (0 <= j && j < 100) {
        val = 120;
      } else if (100 <= j && j < 200) {
        val = flag ? 120 + (15.0 / 100) * (j - 100) : 135;
      } else if (200 <= j && j < 280) {
        val = flag ? 135 + (90.0 / 80) * (j - 200) : 225;
      } else if (280 <= j && j < 300) {
        val = flag ? 225 + (15.0 / 20) * (j - 280) : 240;
      } else if (300 <= j && j < MAX) {
        val = 240;
      }
      image[i * MAX + j] = val;
    }
  }
}
```

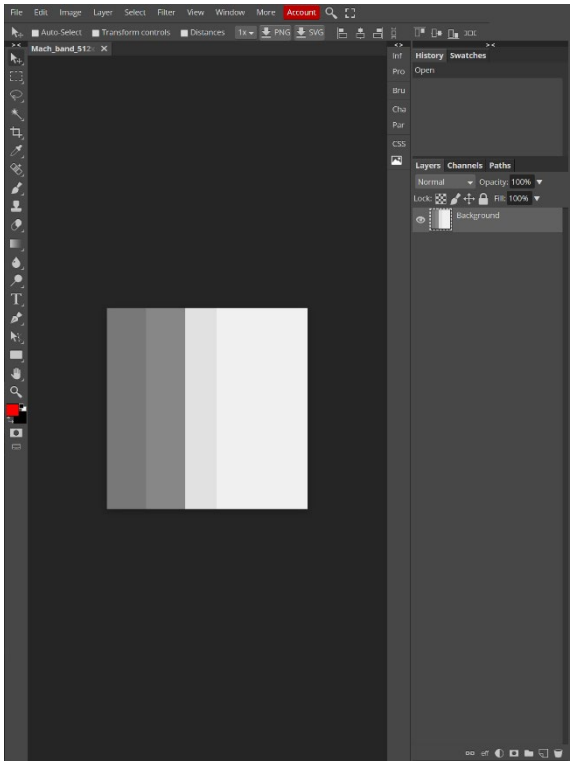
generate_raw_data()는 RAW 파일을 생성하는 부분이다. flag 값이 true이면 그래프와 이미지처럼 생성된 이미지에서 색이 서서히 바뀌게 된다. 반면에 flag값이 false이면 이미지에서 색이 설정된 값으로 급격하게 바뀌게 된다. Mach band를 극적으로 느끼게 하기 위해 기본적으로 false 상태이다.

```
int main() {
  bool image_style_flag = false;
  BYTE *image = (BYTE *)malloc(sizeof(BYTE) * MAX * MAX);
  FILE *fp;
  fp = fopen("outputs/Mach_band_512x512.raw", "wb");
  if (fp == NULL) {
    printf("File Open Error\n");
    return 0;
  }

  generate_raw_data(image, image_style_flag);
  fwrite(image, sizeof(BYTE), MAX * MAX, fp);
  fclose(fp);
}
```

generate_raw_data()를 통해 생성된 이미지를 저장한다.

그 결과 다음과 같이 출력되었다.



2. Using the raw image generated in the problem 1, generate a BMP (bitmap) image file which can be directly displayed on your PC monitor without executing Paint Shop Pro or Photoshop software.

BMP 파일은 압축을 하지 않은 상태의 이미지 파일로 압축을 하지 않아 상대적으로 파일의 크기가 크다. 그리고 파일의 아래쪽부터 이미지의 정보를 담고 있다는 특징이 있다.

RAW 파일로 BMP 파일을 만들기 위해서는 먼저 생성하려는 파일 앞에 BMP 헤더들을 추가해야 한다. 추가해야 하는 헤더로는 BITMAPFILEHEADER, BITMAPINFOHEADER, RGBQUAD가 있다.

BITMAPFILEHEADER	
변수 이름	설명
bfType	BMP 파일인지 구별하는 부분. 0x4D42
bfSize	BMP 파일의 크기

bfReserved1	예약된 위치로 0 이어야 한다.
bfReserved2	예약된 위치로 0 이어야 한다.
bfOffBits	BITMAPFILEHEADER의 처음부터 이미지 정보가 담긴 곳 전까지의 크기. 비트맵 데이터의 시작 위치를 알 수 있다.

BITMAPINFOHEADER	
변수 이름	설명
biSize	이 헤더의 크기
biWidth	이미지의 가로 길이
biHeight	이미지의 세로 길이
biPlanes	Planes의 수로 1로 설정된다.
biBitCount	픽셀 당 비트. 이번 과제에서는 8비트(256색)를 이용한다.
biCompression	압축방식으로 0으로 설정한다.
biSizeImage	그림의 크기. 여기서는 512 * 512가 된다.
biXPelsPerMeter	미터 당 픽셀로 가로 해상도를 나타낸다. 여기서는 0을 갖는다.
biYPelsPerMeter	미터 당 픽셀로 세로 해상도를 나타낸다. 여기서는 0을 갖는다.
biClrUsed	비트맵에서 사용한 컬러 테이블의 컬러 인덱스 수.
biClrImportant	비트맵을 표현하는데 중요한 컬러 인덱스의 수. 0을 입력하면 모두 중요한 것으로 처리된다.

RGBQUAD	
변수 이름	설명
rgbBlue	파란색의 세기
rgbGreen	초록색의 세기

rgbRed	빨간색의 세기
rgbReserved	예약된 위치로 0이 되어야 한다.

이 헤더를 새로 만들어질 파일의 앞부분에 추가한다. 그 뒤에 문제 1번에서 만든 RAW 파일에서 이미지 정보를 읽어오고 위아래를 뒤집어 새로 만들어질 파일의 헤더 다음에 추가한다. 다음은 그 코드이다.

```
typedef struct headers {
  BITMAPFILEHEADER hFile;
  BITMAPINFOHEADER hInfo;
  RGBQUAD hRGB[256];
} BITMAPHEADERS;

void reverse_raw_data(BYTE *image) {
  for (int i = 0; i < MAX; i++) {
    for (int j = 0; j < MAX; j++) {
      if (i < MAX / 2) {
        swap(image[i * MAX + j], image[(MAX - i - 1) * MAX + j]);
      }
    }
  }
}
```

위에서 설명한 것처럼 BMP 파일 생성시에는 RAW 파일에서 볼 수 있는 이미지를 뒤집어서 저장해야한다. 이 함수는 RAW 그림을 뒤집어 주는 부분이다.

```
void generate_headers(BITMAPHEADERS &bh) {
  bh.hFile.bfType = 0x4D42;
  bh.hFile.bfReserved1 = 0;
  bh.hFile.bfReserved2 = 0;
  bh.hFile.bfOffBits = sizeof(BITMAPFILEHEADER) + sizeof(BITMAPINFOHEADER) +
                       sizeof(RGBQUAD) * 256;
  bh.hFile.bfSize = bh.hFile.bfOffBits + MAX * MAX;

  bh.hInfo.biSize = 40;
  bh.hInfo.biWidth = MAX;
  bh.hInfo.biHeight = MAX;
  bh.hInfo.biPlanes = 1;
  bh.hInfo.biBitCount = 8;
  bh.hInfo.biCompression = 0;
  bh.hInfo.biSizeImage = MAX * MAX;
  bh.hInfo.biXPelsPerMeter = 0;
  bh.hInfo.biYPelsPerMeter = 0;
  bh.hInfo.biClrUsed = 0;
  bh.hInfo.biClrImportant = 0;

  for (int i = 0; i < 256; i++) {
    bh.hRGB[i].rgbBlue = i;
    bh.hRGB[i].rgbGreen = i;
    bh.hRGB[i].rgbRed = i;
    bh.hRGB[i].rgbReserved = 0;
  }
}
```

위의 코드는 BMP 파일의 헤더를 생성하는 부분이다.

```
int main() {
  BITMAPHEADERS bh;
  generate_headers(bh);

  BYTE *image;
  FILE *output_file, *input_file;
  input_file = fopen("outputs/Mach_band_512x512.raw", "rb");
  image = (BYTE *)malloc(sizeof(BYTE) * MAX * MAX);
  fread(image, sizeof(BYTE), MAX * MAX, input_file);
  fclose(input_file);

  reverse_raw_data(image);

  output_file = fopen("outputs/New_Mach_band_512x512.bmp", "wb");
  fwrite(&bh.hFile, sizeof(BITMAPFILEHEADER), 1, output_file);
  fwrite(&bh.hInfo, sizeof(BITMAPINFOHEADER), 1, output_file);
  fwrite(bh.hRGB, sizeof(RGBQUAD), 256, output_file);

  fwrite(image, sizeof(BYTE), MAX * MAX, output_file);
  fclose(output_file);
  free(image);
}
```

메인 부분이다. 1번에서 생성한 RAW 파일을 불러오고, 새로 만들 BMP 파일에 헤더를 기록한 다음 불러온 RAW 데이터를 뒤집어서 기록한다.

2번의 결과로 생성된 BMP 파일로 Mach band effect를 느낄 수 있다.

3. Rotate the image obtained in number 1 90 degrees to the counter-clockwise and store it properly in BMP format as you did in 2. Display the stored BMP image on your PC monitor.

```
BYTE *rotate_raw_data(BYTE *image) {
  BYTE *newimage = (BYTE *)malloc(sizeof(BYTE) * MAX * MAX);
  for (int i = 0; i < MAX; i++) {
    for (int j = 0; j < MAX; j++) {
      newimage[(MAX - j - 1) * MAX + i] = image[i * MAX + j];
    }
  }
  return newimage;
}
```

문제 3번의 경우 코드의 대부분은 문제 2번에서 사용한 코드를 재사용한다. 문제 3번의 조건에서 반시계 방향으로 90도 회전을 해야 하므로 rotate_raw_data()를 통해 RAW 데이터

를 회전시키는 부분이다.

```c
int main() {
  BITMAPHEADERS bh;
  generate_headers(bh);

  BYTE *image;
  FILE *output_file, *input_file;
  input_file = fopen("outputs/mach_band_512x512.raw", "rb");
  image = (BYTE *)malloc(sizeof(BYTE) * MAX * MAX);
  fread(image, sizeof(BYTE), MAX * MAX, input_file);
  fclose(input_file);

  image = rotate_raw_data(image);
  reverse_raw_data(image);

  output_file = fopen("outputs/new_rotated_mach_band_512x512.bmp", "wb");
  fwrite(&bh.hFile, sizeof(BITMAPFILEHEADER), 1, output_file);
  fwrite(&bh.hInfo, sizeof(BITMAPINFOHEADER), 1, output_file);
  fwrite(bh.hRGB, sizeof(RGBQUAD), 256, output_file);

  fwrite(image, sizeof(BYTE), MAX * MAX, output_file);
  fclose(output_file);
  free(image);
}
```

메인 부분도 문제 2와 같고 BMP 파일에 저장하기 위해 이미지를 뒤집기 전에 회전시킨다.

그 결과 다음과 같이 출력되었다.

Discussion

문제를 어떤 걸 알았는지, 어떻게 해결했는지

문제 1번의 경우에는 특별한 문제가 없었다. 다만, RAW 파일을 읽을 수 있는 뷰어들이 실제로는 읽을 수 없었고, Photopea에서 온라인 상에서 이미지를 확인할 수 있었다.

문제 2번에서 처음에는 제공된 BMP 파일의 헤더를 추출하였고, Lena의 이미지가 담긴 RAW 파일로 테스트한 결과 예상과 다른 모습으로 출력되었다. 생성된 이미지의 상단에는 이미지가 비어 있었고, 하단에는 동일한 이미지가 가로로 3개 표현되었다. 파일의 정보를 확인해 보니 해당 파일은 24bit 형식으로 색을 표현하는데 이 과제에서 생성한 이미지는 8bit이기 때문에 문제가 발생한 것으로 확인하였다. 해결방법으로는 해당 파일을 8bit 형식으로 바꾸었고 헤더에 RGBQUAD 정보가 담긴 부분도 재정의 하니 문제없이 출력되었다. 그러나, 결과적으로 제출한 코드에서는 헤더의 값을 처음부터 입력하는 방식으로 구현하였다.

문제 3번에서는 반시계방향으로 90도 회전하는 것을 빼면 문제 2번과 다른 것이 없기 때문에 특별한 문제는 없었다.

Reference

비트맵 헤더 정보

https://docs.microsoft.com/en-us/windows/win32/api/wingdi/ns-wingdi-bitmapfileheader

https://docs.microsoft.com/en-us/previous-versions//dd183376(v=vs.85)

https://docs.microsoft.com/en-us/windows/win32/api/wingdi/ns-wingdi-rgbquad

아스키 코드

https://ko.wikipedia.org/wiki/ASCII

※ Please make sure to enclose programming source codes when submitting the report.