

Photo Processing

Edward Bolles
Cs 610
June 26, 2007

A s s i g n m e n t S p e c i f i c a t i o n s

Scope: Input a picture from file. Modify the picture using negative and contrast formulas. Save modified pictures to file.

Inputs and Source of Inputs:

- Picture: file.

Outputs:

- Modified pictures using negative and contrast formulas.

Formulas:

- Negative = $255 - x$
- Stretch = $256 / (256 - (\text{uplimit} - \text{lowlimit}))$
- NewValue = $(\text{inputValue} - \text{lowLimit}) * \text{stretch}$

Limitations and needs:

- Convert images to jpg for viewing

Source Code

```
/* **** */
/* photo.c */
/* Image manipulation */
/* Edward Bolles */
/* **** */

#include "tools.h"
#define VIEW 20

typedef unsigned char pixel;

typedef struct {
    char filename[80];
    char filetype[3];
    int width;
    int height;
    int maxgrey;
    unsigned char** pixel;// why **? does this pixel clash with my typedef pixel above?
} pgm_type;

//-----prototypes-----
void read_header ( pgm_type header[] );
void read_image( pixel* pixel_array, pgm_type header[] );
void write_image( pixel* pixel_array, pgm_type header[] );
void negative( pixel* pixel_array, pgm_type header[] );
void contrast( pixel* pixel_array, pgm_type header[] );
void see_image( pixel* pixel_array, pgm_type header[] );
//-----main-----
void main (void)
{
    pixel* pixel_array;// make pointer for image array
    int decide;
    pgm_type header[1] = {0};//I put a subscript so I can access struct elements lines below.
    banner();
    puts( "\nImage processing\n\n" );
    read_header( header );
    pixel_array = malloc((header[0].height * header[0].width) * sizeof(pixel));
    if (pixel_array==NULL) fatal("Not enough memory\n");
    read_image( pixel_array, header );
    see_image( pixel_array, header);

    for(;;)
    {
        printf("Enter \n1.Negative \n2.Contrast \n3.Open \n4.Save As \n5.Quit\n>");
        scanf("%i", &decide);
        if (decide == 1)
        {
            negative(pixel_array, header);
            see_image( pixel_array, header);
        }
        else if (decide == 2)
        {
            contrast(pixel_array, header);
            see_image( pixel_array, header);
        }
        else if (decide == 3)
        {
            read_header( header );
            pixel_array = malloc((header[0].height * header[0].width) * sizeof(pixel));
            if (pixel_array==NULL) fatal("Not enough memory\n");
            read_image( pixel_array, header );
            see_image( pixel_array, header);
        }
    }
}
```

```

        else if (decide == 4) write_image( pixel_array, header );
        else if (decide == 5) break;
        else printf ("Try again.\n");
    }
    bye();
}

//-----
//
void read_header( pgm_type header[] )
{
    stream infile;
    printf ("Name of file you want to open:");
    scanf(" %79[^\n]", header[0].filename );
    infile = fopen ( header[0].filename, "rb" );
    if ( infile == NULL) fatal ( " Error: can not open input file %s.", header[0].filename );
    printf("Read Image\n\n");
    fscanf( infile, " %2s%i%i%i", header[0].filetype, &header[0].width, &header[0].height,
&header[0].maxgrey);
    printf("HEADER: filetype %2s, width %i, height %i, maxgrey %i\n", header[0].filetype, header[0].width,
header[0].height, header[0].maxgrey);
}

//-----
//
void read_image( pixel* pixel_array, pgm_type header[] )
{
    int num_bytes;
    stream infile;
    infile = fopen ( header[0].filename, "rb" );
    num_bytes = fread( pixel_array, sizeof(pixel), header[0].height * header[0].width, infile);

    if (num_bytes != header[0].height * header[0].width)
        fatal( "Error, input file is not size %i X %i\n", header[0].height * header[0].width );
    fclose( infile );
}

//-----
//
void write_image( pixel* pixel_array, pgm_type header[] )
{
    int num_bytes;
    stream outfile;
    printf ("Name of file for result image:");
    scanf(" %79[^\n]", header[0].filename );
    outfile = fopen ( header[0].filename, "wb" );
    if ( outfile == NULL) fatal ( " Error: cannot open output file %s.", header[0].filename );
    fprintf( outfile, "%2s\n%i %i\n%i\n", header[0].filetype, header[0].width, header[0].height,
header[0].maxgrey);
    num_bytes = fwrite( pixel_array, sizeof(pixel), header[0].height * header[0].width, outfile);
    if (num_bytes != header[0].height * header[0].width)
        fatal( "Error, unable to write entire output file\n", header[0].height * header[0].width);
    fclose( outfile );
}

//-----
//
void negative( pixel* pixel_array, pgm_type header[] )
{
    int slot;
    printf("\nNEGATIVE\n");
    for (slot = 0; slot < (header[0].width * header[0].height); ++slot)
    {
        pixel_array[slot] = 255 - pixel_array[slot];
    }
}

//-----
//

```

```

void contrast( pixel* pixel_array, pgm_type header[] )
{
    int slot, upper, lower, stretch;
    printf("\nCONTRAST\nChoose an upper limit from 200 to 254\n");
    scanf("%i", &upper);
    printf("Choose a lower limit from 100 to 1\n");
    scanf("%i", &lower);
    stretch = 256/(256 - (upper - lower));
    for (slot = 0; slot < (header[0].width * header[0].height); ++slot)
    {
        if (pixel_array[slot] <= upper && pixel_array[slot] >= lower) pixel_array[slot] =
(pixel_array[slot] - lower) * stretch; //if pixel within threshold, stretch
        else if (pixel_array[slot] > upper) pixel_array[slot] = 255; //else make pixel white
        else pixel_array[slot] = 0; //else make pixel black
    }
}
//-----
//
void see_image( pixel* pixel_array, pgm_type header[] )
{
    int slot;
    printf("\nFile: %s\n", header[0].filename);
    printf( "%2s\n%i %i\n%i\n", header[0].filetype, header[0].width, header[0].height, header[0].maxgrey);
    for(slot = 0; slot < VIEW; slot++)
    {
        printf("%02x ", pixel_array[slot]);
    }
    printf("\n\n");
}

```

Test Plan

Case	Input: original	Output: negative	Output: contrast
Typical cases	50 35 0a 36 37 36 20 35 30 37 0a 32 35 35 0a 21 28 2c 2a 26	aƒ ca f5 c9 c8 c9 df ca cf c8 f5 cd ca ca f5 de d7 d3 d5 d9	3c 06 00 08 0a 08 00 06 00 0a 00 00 06 06 00 00 00 00 00 00

Test Data Output

♀

```
-----  
      Edward Bolles  
      CS 604  
      Fri Aug 17 2007  11:49:44  
-----
```

Image processing

Name of file you want to open:suns_raw.pgm
Read Image

HEADER: filetype P5, width 676, height 507, maxgrey 255

File: suns_raw.pgm
P5
676 507
255
50 35 0a 36 37 36 20 35 30 37 0a 32 35 35 0a 21 28 2c 2a 26

Enter
1.Negative
2.Contrast
3.Open
4.Save As
5.Quit
>1

NEGATIVE

File: suns_raw.pgm
P5
676 507
255
af ca f5 c9 c8 c9 df ca cf c8 f5 cd ca ca f5 de d7 d3 d5 d9

Enter
1.Negative
2.Contrast
3.Open
4.Save As
5.Quit
>4

Name of file for result image:suns_negative.pgm

Enter
1.Negative
2.Contrast
3.Open
4.Save As
5.Quit
>3

Name of file you want to open:suns_raw.pgm
Read Image

HEADER: filetype P5, width 676, height 507, maxgrey 255

File: suns_raw.pgm
P5
676 507
255
50 35 0a 36 37 36 20 35 30 37 0a 32 35 35 0a 21 28 2c 2a 26

Enter
1.Negative
2.Contrast
3.Open

4.Save As
5.Quit
>2

CONTRAST
Choose an upper limit from 200 to 254
220
Choose a lower limit from 100 to 1
50

File: suns_raw.pgm
P5
676 507
255
3c 06 00 08 0a 08 00 06 00 0a 00 00 06 06 00 00 00 00 00

Enter
1.Negative
2.Contrast
3.Open
4.Save As
5.Quit
>4
Name of file for result image:suns_contrast.pgm
Enter
1.Negative
2.Contrast
3.Open
4.Save As
5.Quit
>3
Name of file you want to open:suns_negative.pgm
Read Image

HEADER: filetype P5, width 676, height 507, maxgrey 255

File: suns_negative.pgm
P5
676 507
255
50 35 0a 36 37 36 20 35 30 37 0a 32 35 35 0a af ca f5 c9 c8

Enter
1.Negative
2.Contrast
3.Open
4.Save As
5.Quit
>1

NEGATIVE

File: suns_negative.pgm
P5
676 507
255
af ca f5 c9 c8 c9 df ca cf c8 f5 cd ca ca f5 50 35 0a 36 37

Enter
1.Negative
2.Contrast
3.Open
4.Save As
5.Quit
>

suns_raw



suns_negative



suns_contrast



Notes:

The program seems fine. The only strange thing is that when I opened `suns_negative`, it opens showing the pixel array of `suns_raw`. Yet the picture is indeed negative as shown above.

Also, I noticed that the pointer `pixel` is declared twice in the typedef. Is there any problems with this?