

Lab 2. Edge Detection using Image Thresholding over 8-bit Gray Scale BMP images

EXPLANATION

Histogram stretching can be seen as a Window and Level contrast enhancement technique where the window ranges from the minimum to the maximum pixel values of the image. The image shown below has been enhanced via histogram stretching. Also, its new histogram is plotted. Note that the histogram now ranges from 0 to 255. This operation can be performed by normalization, which is a procedure that maps the minimum value of the image to 0 and its maximum value to 255.

Thresholding

In this technique, known as thresholding, the values of r below m are compressed by the transformation function into a narrow range of s , toward black. The opposite effect takes place for values of r above m . $T(r)$ produces a two-level (binary) image. A mapping of this form is called a thresholding function. Some fairly simple, yet powerful, processing approaches can be formulated with gray-level transformations. Because enhancement at any point in an image depends only on the gray level at that point, techniques in this category often are referred to as point processing.

```
if(pixel<th)
    pixel = 0;
else
    pixel = 255;
```

PROCEDURE

1. START
2. Open source file (say, in_vsatsat.bmp) and destination file (say out_vsatsat.bmp).
3. Read the header (1078 bytes) of the source file (in_vsatsat.bmp) and write it (1018 bytes) to destination file (out_vsatsat.bmp).
4. Read Image Data (say, $f(x)$) till end of the file.
5. Read appropriate threshold T .
6. Apply threshold on the input image.
7. Write S into destination file (out_vsatsat.bmp).
8. END

Program Code:

//Image Thresholding in C

```
#include<stdio.h>
#include<conio.h>
#include<math.h>
#include<string.h>
```

```
int main(void)
{
    FILE *sFile, *dFile;
```

```

int i, th;
unsigned char pixel;

sFile = fopen("bird.bmp", "rb");
if(sFile == NULL)
{
    printf("File not Found");
    getch();
    return 1;
}

dFile = fopen("birdThreshod.bmp", "wb+");
if(dFile == NULL)
{
    printf("File is not created");
    getch();
    return 1;
}

printf("Enter the value of threshold = ");
scanf("%d", &th);

for(i = 0; i < 1078; i++)
{
    fread(&pixel, sizeof(char), 1, sFile);
    fwrite(&pixel, sizeof(char), 1, dFile);
}
while(!feof(sFile))
{
    fread(&pixel, sizeof(char), 1, sFile);
    if(pixel < th)
    {
        pixel = 0;    //
        fwrite(&pixel, sizeof(char), 1, dFile);
    }
    else
    {
        pixel = 255;
        fwrite(&pixel, sizeof(char), 1, dFile);
    }
}
printf("Thresholded image generated successfully with Threshold = %d\n", th);
fclose(sFile);
fclose(dFile);
//getch();

return 0;

```

}

Program output:

```
D:\Lab2\Lab2.exe
Enter the value of threshold = 80
Thresholded image generated successfully with Threshold = 80
-----
Process exited after 6.624 seconds with return value 0
Press any key to continue . . .
```



bird.bmp



birdThreshod.bmp