

# Assignment Pthreads

## Parallel and distributed computing

### Dithering a 2D Image File

Digital images are stored as 2D array of pixels. Each pixel in a color image is represented by 3 bytes (called the color depth). One byte for each of the RGB colors (red, blue, and green). An alternative to the 24-bit color depth is the indexed color mode. In indexed color mode, fewer number of bits are used to index through a color map. In grayscale (also called black and white) images, 8-bits are stored per pixels to represent many shades of gray.

Sometimes, the bit-depth of an image is needed to be reduced for some reason. For example, the display or printer don't support that many colors, or to make the image file smaller. In these cases, we need to compensate for the lost colors due to this reduction. This compensation for the lost colors by the available colors is called image dithering. In this technique a simulated color by using the available colors is used to trick the eye seeing the original color. This is achieved by placing dots of colors close to each other in a way that they are blended by the eye to look like the desired color.

There are many algorithms used for image dithering, in this assignment we will use pattern dithering.

### Pattern Dithering

Pattern dither uses a mask to dither the image. A pattern that gives good results is

8 3 4

6 1 2

7 5 9

In this assignment you will use the above mask to dither a grayscale image (8-bit grayscale image) to generate a black and white 1-bit image. Follows the following steps to do that:

- 1) Normalize the original image where each pixel is represented by 1 byte (0-255) to numbers between 0 and 9. You can do this by dividing the pixel values by 28.
- 2) Start in the upper left corner of the image, place the 3 x 3 mask over the top leftmost 3 x 3 block of pixels. In each pixel position, if the normalized pixel value is greater than the mask value, then this pixel is given a value of 1 in the dithered image. Otherwise it is a 0.
- 3) Then move the mask to the right by three pixels.

4) When you finish a row, move the mask down by three rows and back over to the left.

## Matlab

Here is the original grayscale image before and after dithering



To see the image in matlab you need to do the following at matlab prompt

1) read the data from the file oimage.txt

```
>> A = dlmread('oimage.txt');
```

2) remove first row since it contains the size

```
>> A(1, :) = [];
```

3) See the image

```
>> figure; imshow(A, [0, 255]);
```

**Q1: Serial program.** Write a serial program that accept as input from the line command three parameters, the first parameter is the name of the file that contains the original image, the second parameter is the name of the file that contains the mask (given above), and the third parameter is just the name of the file to be used to store the dithered image.

For simplicity I will give you the image file as a two dimensional array of chars (1-byte integer).

The first line of the file will contain two numbers, (number of rows and number of columns).

Your output file should have similar format as the input image but the values of the pixels should be either 0 or 1. Time the run time of the execution of the program.

**Q2:** Parallelize the serial program using pthread. Time the execution of the parallel program when using 1, 2, ..., p cores, where p is the maximum number of cores that your machine has.

**Q3:** Write a 1 page summary of what you did and include the timing data, and the speedup

**What to turn in?**

A compressed tar ball that contains Five files:

- 1) the serial code file
- 2) the pthread parallel code file
- 3) the original image file that I provided you
- 4) the mask file
- 5) a text file that contain your writeup
- 6) makefile that allow to make the serial and the parallel version of the program