NATIONAL INSTITUTE OF TECHNOLOGY KARNATAKA SURATHKAL

DEPARTMENT OF INFORMATION TECHNOLOGY

IT 301 Parallel Computing LAB 6

16th September 2020

Faculty: Dr. Geetha V and Mrs. Thanmayee

Akshith Nettar Mahalinga

181IT104

Write a parallel program (using Openmp) to convert a color image to grayscale and YIQ. The RGB values (in decimal) are already extracted and stored in "KittenRGB.txt" file. Read the input values from the file.

a. Compute the grayscale conversion using luminosity method, that is,

G=R*0.21+G*0.72+B*0.07.

b. Here is the RGB -> YIQ conversion:

Y = 0.299*R + 0.587*G + 0.114*B

I = 0.596*R – 0.275*G – 0.321*B

Q = 0.212*R – 0.523*G + 0.311*B

Analysis: Compare the time taken for the computation with Single thread and Multiple threads(2,4,8,16) . Prove that parallel computation is faster than serial. NOTE: The RGB.txt file has RGB values in single line and not matrix format. Ex: R G B R G B R G B …

The image used for extracting RGB values is 300 * 300 pixels jpg

CODE

```
#include<stdio.h>
#include<omp.h>
#define ROWS 300*300

void rgbToGrayscale(int RGB[ROWS][3]) {
   double grayscale = 0.0;
   for(int i=0; i<ROWS; i++) {
       grayscale = 0.21*RGB[i][0] +
                0.72*RGB[i][1] + 0.07*RGB[i][2];
  }
}

void rgbToYIQ(int RGB[ROWS][3]) {
   double Y = 0.0, I = 0.0, Q = 0.0;
   for(int i=0; i<ROWS; i++)  {
       Y = 0.299*RGB[i][0] + 0.587*RGB[i][1] +
                0.114*RGB[i][2];
       I = 0.596*RGB[i][0] - 0.275*RGB[i][1] -
                0.321*RGB[i][2];
```

```
        Q = 0.212*RGB[i][0] - 0.523*RGB[i][1] +
                0.311*RGB[i][2];
    }
}

void rgbToGrayscale_parallel(int RGB[ROWS][3],
                int n_threads) {
    double grayscale = 0.0;
    #pragma omp parallel for private(grayscale)
                num_threads(n_threads)
                schedule(guided, 3)
    for(int i=0; i<ROWS; i++) {
        grayscale = 0.21*RGB[i][0] +
                0.72*RGB[i][1] + 0.07*RGB[i][2];
    }
}

void rgbToYIQ_parallel(int RGB[ROWS][3], int
                n_threads) {
    double Y=0,I=0,Q=0.0;
    #pragma omp parallel for private(Y,I,Q)
                num_threads(n_threads)
                schedule(guided, 3)
    for(int i=0; i<ROWS; i++) {
        Y = 0.299*RGB[i][0] + 0.587*RGB[i][1] +
                0.114*RGB[i][2];
        I = 0.596*RGB[i][0] - 0.275*RGB[i][1] -
                0.321*RGB[i][2];
        Q = 0.212*RGB[i][0] - 0.523*RGB[i][1] +
                0.311*RGB[i][2];
    }
}

int main() {
```

```
    char ch;
    FILE* fp = fopen("KittenRGB.txt", "r");
    int RGB[ROWS][3];
    int row = 0, col = 0;
    do {
        fscanf(fp, "%d", &RGB[row][col]);
        fscanf(fp, "%d", &RGB[row][col+1]);
        fscanf(fp, "%d", &RGB[row][col+2]);
        row += 1;
    } while((ch = fgetc(fp))!=EOF);

    double et = 0.0, st = 0.0;

    for(int i=2; i<=16; i=i*2) {
        st = omp_get_wtime();
        rgbToYIQ_parallel(RGB, i);
        et = omp_get_wtime();
        printf("RGB --> YIQ PARALLEL TIME
                TAKEN=%f\n", et-st);
    }
    printf("\n");

    for(int i=2; i<=16; i=i*2) {
        st = omp_get_wtime();
        rgbToGrayscale_parallel(RGB, i);
        et = omp_get_wtime();
        printf("RGB --> GRAYSCALE PARALLEL TIME
                TAKEN=%f\n", et-st);
    }
    printf("\n");

    st = omp_get_wtime();
    rgbToYIQ(RGB);
```

```
    et = omp_get_wtime();
    printf("RGB-->YIQ SERIAL TIME TAKEN=%f\n", et
                - st);
    printf("\n");

    st = omp_get_wtime();
    rgbToGrayscale(RGB);
    et = omp_get_wtime();
    printf("RGB-->GRAYSCALE SERIAL TIME
                TAKEN=%f\n", et - st);
    printf("\n");

}
```

SCREENSHOTS

```
RGB --> YIQ PARALLEL TIME TAKEN=0.000510
RGB --> YIQ PARALLEL TIME TAKEN=0.000285
RGB --> YIQ PARALLEL TIME TAKEN=0.000969
RGB --> YIQ PARALLEL TIME TAKEN=0.002821

RGB --> GRAYSCALE PARALLEL TIME TAKEN=0.000494
RGB --> GRAYSCALE PARALLEL TIME TAKEN=0.000161
RGB --> GRAYSCALE PARALLEL TIME TAKEN=0.002576
RGB --> GRAYSCALE PARALLEL TIME TAKEN=0.001005

RGB-->YIQ SERIAL TIME TAKEN=0.000880

RGB-->GRAYSCALE SERIAL TIME TAKEN=0.000376
```