Assignment: Part 2 - Report

Mingze Gao - 180221943

Content

1 Overview

1.1 Starting Code

As the assignment requires, I choose the outer loop parallelization which consumes about 12ms under CPU mode and 6ms under OpenMP mode as the starting code and make an improvement based on feedback.

1.2 Structure of Report

There are n implementations below which I have done in part 2.

- **Initial**

In the initial version, I have implemented the code by parallelising mosaic cell, storing pixel values in a vector and using shared memory.

- **Optimisation01**

2 Comparison

2.1 Initial

2.1.1 Overview

Within the initial version, I parallelise each mosaic cell by assigning the number of cells to the number of blocks and threads. In other words, in cell level, **each block represents a row** and **each thread represents a cell within the row**.

As for the data in the device, I use a vector to store a color value and use another vector to store the first element's address of each row in the previous vector. Because different threads process different cells and there will not be any conflict for reading and writing, I use the same vector as input and output. Moreover, I use a vector stored in shared memory to store the sum of each row's pixel value and sum the vector at the end in host.

2.1.2 Result

The consuming time of different approaches for the initial version is showed below.

Index	Mode	Width x Height	c	The number of Cells in a row	Time
1	CPU	2048 x 2048	4	512	0s, 19.22ms
2	OPENMP	2048 x 2048	4	512	0s, 18,38ms
3	**CUDA**	2048 x 2048	**4**	**512**	**0s, 1.27ms**
4	CPU	2048 x 2048	128	16	0s, 17.21ms
5	OPENMP	2048 x 2048	128	16	0s, 17.60ms
6	**CUDA**	2048 x 2048	**128**	**16**	**0s, 12.80ms**
7	CPU	2048 x 2048	256	8	0s, 15.87ms
8	OPENMP	2048 x 2048	256	8	0s, 15.47ms
9	**CUDA**	2048 x 2048	**256**	**8**	**0s, 60.46ms**

2.1.3 Discussion

Q1: The reason for consuming more time with the increase of c value.

A1: Obviously, GPU performs very well at parallel computing and the timing results also prove it. Since each thread process a cell, there will be less calculation within a thread if c value is small which means that it will fully use the advantages of GPU and consumes less time. [IMPROVEMENT - unrelated c]

Q2: The improvement of memory strategies

A2: Because I store all variables at global memory except the sum of each row's value stored in share memory, this may consume more time on addressing the location of an element. [IMPROVEMENT - using constant and texture]

3 Limitations

DELETE BEFORE SUBMIT

```
16 CPU -i 2048x2048PlainText.ppm -o 2048x2048PlainText_output.ppm -f PPM_PLAIN_TEXT
```