CS6023: GPU Programming (Jan 2025)

Assignment 1

Due Feb 9, 23:59

1 Problem Statement

Thomas loves photography and is fascinated by the various filter presets available in his camera. He loves to capture the moments and see them from a different perspective. And decided to make his own preset and decided to apply the transformation to entire image collection he has.

He stumbles upon the recent Nvidia CES conference and being a geek, decides to use parallel computation to complete the work faster.

1. **Inverted Gray Scale:**
   Given a file of RGB pixel values in 3 different matrices of the same dimensions, use the image representation given above to make it a grayscale image and the final image should be inverted.

   Note:
   RGB to Gray Scale is the average value of the three components Red, Green, Blue.

   $$\text{gray(i,j)} = \left\lfloor \frac{red(i,j)+green(i,j)+blue(i,j)}{3} \right\rfloor$$

   gray(i,j) defines the j^{th} value in i^{th} row of the matrix.

   Final result matrix should be of form final$_matrix(i, j) = gray(n - i, j)$
   where **n** is total number of rows in image matrix.

2. **Thomas Transformation:**
   Given a file of RGB pixel values in 3 different matrices of the same dimensions, use the below function to compute the final image matrix

   $$\text{res(i,j)} = \lfloor 0.5 \times red(i,j) \rfloor + \lfloor \sqrt{green(i,j)} \rfloor + \text{blue(i,j)}$$

   res(i,j) defines the j^{th} value in i^{th} row of the matrix.

   Note:
   Please adhere to the exact function definition mentioned above.
   Apply the floor function to round it to the largest integer less than or equal to the value, to maintain consistency in terms of data types for all the arithmetic operations.

Your Task is to generate **both** the transformed matrices and copy the matrix information from device to host. Refer guidelines to know more about the starter code.

2 General Guidelines:

- Do not change any other existing pieces of code that are given in the starter code. Please add necessary memory allocation and memcpy operations.
- We will time each of the kernel calls to ensure that your implementation is parallel.
- This assignment is an individual effort, and it is expected that all work submitted is your own. Plagiarism, the act of using someone else's work, is a serious academic offence and will not be tolerated. Any instances of plagiarism will result in severe consequences and would be facing academic disciplinary actions.

3 Input Output Format:

The input folder consists of input testcases where each file represents the a testcase.
Input:

- The first line would be the 2 integers which are space separated which represents the Dimension of the Image given in the file.
- And is followed by 3 matrices of the same dimension, which represent 3 different channels of the image i.e Red, Green and Blue. Note that the order followed by the matrices is the same as mentioned above.

Output:

- There would be 1 output matrix corresponding to each transformation whose dimension would be same as to that of the input provided as a testcase.
- So it is expected to have 2 matrices one followed by another in the output which is handled in the starter code. You are expected to populate the corresponding matrices in the corresponding variables correctly to achieve marks for each testcase.
- The final output from the device should be stored in the arrays . This array will be copied to the host and written to a file. This is present in the starter code.
- This will be compared against the corresponding output file in the output folder for each testcase.

3.1 Sample Input

2	3		Dimension of matrix
1	2	4	Red channel of matrix of dim 2 × 3
12	11	4	
2	4	4	Green channel of matrix of dim 2 × 3
2	4	4	
6	8	4	Blue channel of matrix of dim 2 × 3
6	8	4	

3.2 Sample Output

```
 6   7  4
 3   4  4
 7  11  8
13  15  8
```

Explanation:

$$\text{T1} = \begin{bmatrix} 6 & 7 & 4 \\ 3 & 4 & 4 \end{bmatrix} \text{ and T2} = \begin{bmatrix} 7 & 11 & 8 \\ 13 & 15 & 8 \end{bmatrix}$$

T1 is Inverted Gray Scale and **T2** is Thomas Transformation

$$\text{T1}[0][0] = \lfloor \tfrac{red[1][0]+green[1][0]+blue[1][0]}{3} \rfloor$$

$$\text{T2}[0][0] = \lfloor 0.5 \times red[0][0] \rfloor + \lfloor \sqrt{green[0][0]} \rfloor + \text{blue}[0][0]$$

4 Testing

- You can use the tester program given along with the starter code to test your program.
- Delete the existing *.cu files and place your RollNumber.cu file in **submit** folder
- Run bash test.sh to evaluate your program against the available sample testcase or make your own testcases and evaluate against your custom testcases.

5 Important instructions

- If you have any general questions, please post on moodle.
- Fill in your code in the starter.cu file. Rename it as your rollno.cu. For example, if your roll number is cs23m028, your file should be named as cs23m028.cu
- Submit a single .cu file with your roll number (e.g., cs24m001.cu). The output must match the expected format. A deviation from these will incur a 25% penalty.
- Questions specific to you can be emailed to Hrudai (CS23M028) and Joel (CS23M031), copying Rupesh (rupesh).
- Sequential codes will not fetch marks.
- After submitting, please download your file and test it for correctness.