Homework MP 6 Answer

1. Initial Implementation:
   The elements in memory will be stored in rowmajor format, so taking this as input directly (by removing the zero_padding), this was used in the kernel for computation.
   The speed up observed was 3x. This is with the histo_bins in shared memory.
   Challenge was in clearing the output memory. This was the first time resetting memory was necessary. This was done by making each thread clear up one memory space. This was the performance hit.

   INPUT
   WIDTH 9000
   HEIGHT 9000

   RESULT:
   Timing 'ref_2dhisto' started
   GetTimeOfDay Time (for 50 iterations) = 8.62
   Clock Time (for 50 iterations) = 8.62
   Timing 'ref_2dhisto' ended
   Timing 'opt_2dhisto' started
   GetTimeOfDay Time (for 50 iterations) = 2.211
   Clock Time (for 50 iterations) = 2.22
   Timing 'opt_2dhisto' ended
   Test PASSED

2. Changes made
   Challenge was declaring the histogram_bin as uint8_t data type (since 255 is the saturation limit).
   So, both of the above was taken care by calling a atomicCAS function (taken from NVIDIA website. No extra header file was needed).
   This made lot of the execution in series since all threads had to pass through atomicCAS function. The performance dropped and could not correct result.

3. Improvement:

   Step 1: Divide each row into different groups so that each thread can concurrently work on a portion of the row.

   Acheived by dividing the row into 'T' groups each having (width/T) elements.

   Step 2: Calculate the row id and column id according to this new configuration. Row id = (idx / T) * height of the input data.

   Column id = (idx % T) * (width / T) , as each thread accesses data strided in the columns by (width / T)

Homework MP 6 Answer

Step 3: Each thread performs work only in its assigned group - The loop runs from 0 to (width / T)

We obtained improvement of only 3X.

INPUT
WIDTH 996
HEIGHT 124

RESULT:

For 500 iterations:

Timing 'ref_2dhisto' started

GetTimeOfDay Time (for 500 iterations) = 1.098

Clock Time (for 500 iterations) = 1.1

Timing 'ref_2dhisto' ended

Timing 'opt_2dhisto' started

GetTimeOfDay Time (for 500 iterations) = 0.502

Clock Time (for 500 iterations) = 0.5

Timing 'opt_2dhisto' ended

Test PASSED

4. 3rd Improvement:

   Observation was that lot of time was taken up in clearing the shared memory. So, tried not to use the shared memory and a performance gain was observed.

   The cuda specific function to clear the global memory was used (cudaMemset from NVIDIA website).

   The performance now was 10x.

5. 4th improvement:

   Initially all the threads were updating the histo_bins irrespective of the value. Since the problem statement defines the saturation value, once the histo_bin reaches this value, no updates are necessary. So, a condition was placed. A performance gain of extra 10x was observed! This was due to avoiding many serial operations (atomicAdd). The performance gain for large inputs is 20x.

Homework MP 6 Answer

6. A second kernel was written to make sure all the histo_bins are saturated properly. Since this is executed in parallel, a further performance gain was observed. Time spent half an hour

   Performance gain was 25x for large inputs.

7. Finally,

   The optimum block_size was found to be 512 after trial and error and this put up a restriction on the number of grids that can be launched in a kernel at once.

   So, to cater to large input data elements, kernel is called in a loop. Even though no performance gain can be attributed to this implementation, large data sets can be handled now

RESULTS:

1. INPUT SIZE

   Input Height = 1024

   Input Width = 990

   Timing 'ref_2dhisto' started

   GetTimeOfDay Time (for 50 iterations) = 0.109

   Clock Time (for 50 iterations) = 0.11

   Timing 'ref_2dhisto' ended

   Timing 'opt_2dhisto' started

   GetTimeOfDay Time (for 50 iterations) = 0.007

   Clock Time (for 50 iterations) = 0

   Timing 'opt_2dhisto' ended

   Test PASSED

2. INPUT SIZE

   Input Height = 5000

   Input Width = 5000

Homework MP 6 Answer

Timing 'ref_2dhisto' started

GetTimeOfDay Time (for 50 iterations) = 2.692

Clock Time (for 50 iterations) = 2.7

Timing 'ref_2dhisto' ended

Timing 'opt_2dhisto' started

GetTimeOfDay Time (for 50 iterations) = 0.068

Clock Time (for 50 iterations) = 0.08

Timing 'opt_2dhisto' ended

Test PASSED

3. INPUT SIZE:

Input Height = 9000

Input Width = 9000

Timing 'ref_2dhisto' started

GetTimeOfDay Time (for 50 iterations) = 8.588

Clock Time (for 50 iterations) = 8.59

Timing 'ref_2dhisto' ended

Timing 'opt_2dhisto' started

GetTimeOfDay Time (for 50 iterations) = 0.211

Clock Time (for 50 iterations) = 0.22

Timing 'opt_2dhisto' ended

Test PASSED

Statistics:

Input Size	**CPU execution time (Secs)**	**GPU execution time (Secs)**	**Speed-up**
990x1024	0.109	0.007	16X
5000x5000	2.7	0.068	40X
9000x9000	8.59	0.211	41X