Convolution

Questions

1. Three applications of convolution:
   (a) Image processing: e.g. Filters like blurring which can also be used for edge detection via Difference of Gaussians.
   (b) Audio processing: noise reduction, and EQ filtering.
   (c) Real-time rendering: e.g. Irradiance environment maps (GPU Gems 2 Ch 10) which precomputes diffuse reflections from a scene (real or virtual).

2. One thread does one multiplication and addition for an element on the mask filter ($2 \times M \times M$), and we have to do this for the color channels $C = 3$ ($C \times 2M^2$). So for an image of width and height $W \times H$, we have $W \times H \times C \times (2M^2)$ float operations.

3. The shared memory requires $W \times H \times C$ global reads, but there is an additional of up to

$$\begin{aligned} 4 \times \left(\frac{M-1}{2}\right)^2 + 4 \times \frac{M-1}{2} \times T &= (M-1)^2 + 2(M-1) \times T \\ &= (M-1) \times (M-1+2T) \end{aligned}$$

   global reads per tile (where T is the tile size $T \times T$) because if a halo cell is not a ghost cell, it has to be read from global memory from the else statement in the nested for loop.

4. The global writes is equal to the number of output pixels we need to write to, so $W \times H \times C$ global writes.

5. Minimum: At the corner pixel we would have only 9 out of the 25 mask filter elements performing real operation. So we have $9 \times 2 = 18$ additions and multiplications and $18 \times 3 = 54$ float operations for the three color channels.

   Maximum: At the center pixel we would have all mask elements performing real operations, so $25 \times 2 \times 3 = 150$ float operations.

   Average: We have alot of pixels to deal with...

   - 4 corner pixels performing 54float operations
   - 2×4 pixels horizontally and vertically adjecent to the corner pixels performing $(3 \times 4) \times 6 = 72$float operations
   - $4 \times (N-4)$ edge pixels performing $(3 \times 5) \times 6 = 90$float operations
   - 4 pixels diagonally adjecent to the corner pixels performing $(4 \times 4) \times 6 = 96$float operations
   - $4 \times (N-4)$ inner edge pixels performing $(4 \times 5) \times 6 = 120$ float operations.
   - $(N-4)^2$ inner pixels performing $(5 \times 5) \times 6 = 150$ float operations.

   Double checking if we counted the number of pixels correctly:

$$4 + 2 \times 4 + 4 \times (N-4) + 4 + 4 \times (N-4) + (N-4)^2 = 16 + 8N - 32 + N^2 - 8N + 16 = N^2$$

   So the average float operations per pixel is

$$\frac{4 \times 54 + 8 \times 72 + 4 \times (N-4) \times 90 + 4 \times 96 + 4 \times (N-4) \times 120 + (N-4)^2 \times 150}{N^2}$$

6. For a $W \times H = 64 \times 64$ with a computation time of $0.066\,622\,\mathrm{ms}$, the throughput is roughly

$$\frac{64 \times 64 \times 3 \times 50 \text{ FLO}}{0.066\,622\,\mathrm{ms}} = 9.2\,\mathrm{GFLOPS}$$

   This would scale linearly with the size of the input image or $O(W \times H)$

7. The asymptotic time complexity of the sequential convolution is still $O(W \times H)$ because the parallel convolution is only changing the constant factor by adding parallel execution threads (P) that will roughly decrease the computation time by the $1/P$.

8. For the 64 input image, it take 0.242 ms to allocate the memory on the GPU, 0.402 ms to copy the input image to the GPU, and 0.058 ms to copy the output image back to the CPU. This overhead will scale linearly with the output size, since the host-device memory transfer have a limit to the bandwidth given by the PCIe bus.

9. If the mask size was increased to $M = 1024$ with the block dimension 16×16, the shared memory would be ineffective because most of the time would be spend doing global reads. Futhermore, the constant memory capacity of 64 KB would be exceeded ($1024^2 \times 4 \approx 4\,\text{MB}$), so there would be a CUDA runtime error or compilation error that would prevent the kernel from running.

10. There must be a separate output memory buffer, because the convolution needs to access the original input values, so doing the convolution in place would overwrite the original input values and threads would try to read modified values and run into race conditions.

11. An identity mask does not change the input image once the mask is applied through convolution, and it has the value of 1 at the center and 0 elsewhere. So for the $M = 5$ mask, the identity mask is

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

OUTPUT

```
[junseo@r40a-09.sif Convolution]$ ./convolution.sh
[TIME][GPU][Doing GPU memory allocation][/home/warehouse/junseo/cuda-code-repo-joonsuuh/Module8/Convolution/solution.cu: 99-105] Elapsed time: 0.241623 ms
[TIME][Copy][Copying data to the GPU][/home/warehouse/junseo/cuda-code-repo-joonsuuh/Module8/Convolution/solution.cu: 107-115] Elapsed time: 0.402116 ms
[TIME][Compute][Doing the computation on the GPU][/home/warehouse/junseo/cuda-code-repo-joonsuuh/Module8/Convolution/solution.cu: 117-128] Elapsed time: 0.066622 ms
[TIME][Copy][Copying data from the GPU][/home/warehouse/junseo/cuda-code-repo-joonsuuh/Module8/Convolution/solution.cu: 130-135] Elapsed time: 0.058161 ms
[TIME][GPU][Doing GPU Computation (memory + compute)][/home/warehouse/junseo/cuda-code-repo-joonsuuh/Module8/Convolution/solution.cu: 97-137] Elapsed time: 0.915401 ms
The solution is correct
[TIME][GPU][Doing GPU memory allocation][/home/warehouse/junseo/cuda-code-repo-joonsuuh/Module8/Convolution/solution.cu: 99-105] Elapsed time: 0.288811 ms
[TIME][Copy][Copying data to the GPU][/home/warehouse/junseo/cuda-code-repo-joonsuuh/Module8/Convolution/solution.cu: 107-115] Elapsed time: 0.411354 ms
[TIME][Compute][Doing the computation on the GPU][/home/warehouse/junseo/cuda-code-repo-joonsuuh/Module8/Convolution/solution.cu: 117-128] Elapsed time: 0.065845 ms
[TIME][Copy][Copying data from the GPU][/home/warehouse/junseo/cuda-code-repo-joonsuuh/Module8/Convolution/solution.cu: 130-135] Elapsed time: 0.112055 ms
[TIME][GPU][Doing GPU Computation (memory + compute)][/home/warehouse/junseo/cuda-code-repo-joonsuuh/Module8/Convolution/solution.cu: 97-137] Elapsed time: 1.03922 ms
The solution is correct
[TIME][GPU][Doing GPU memory allocation][/home/warehouse/junseo/cuda-code-repo-joonsuuh/Module8/Convolution/solution.cu: 99-105] Elapsed time: 0.260005 ms
[TIME][Copy][Copying data to the GPU][/home/warehouse/junseo/cuda-code-repo-joonsuuh/Module8/Convolution/solution.cu: 107-115] Elapsed time: 0.366302 ms
[TIME][Compute][Doing the computation on the GPU][/home/warehouse/junseo/cuda-code-repo-joonsuuh/Module8/Convolution/solution.cu: 117-128] Elapsed time: 0.058778 ms
[TIME][Copy][Copying data from the GPU][/home/warehouse/junseo/cuda-code-repo-joonsuuh/Module8/Convolution/solution.cu: 130-135] Elapsed time: 0.149658 ms
[TIME][GPU][Doing GPU Computation (memory + compute)][/home/warehouse/junseo/cuda-code-repo-joonsuuh/Module8/Convolution/solution.cu: 97-137] Elapsed time: 0.965906 ms
The solution is correct
[TIME][GPU][Doing GPU memory allocation][/home/warehouse/junseo/cuda-code-repo-joonsuuh/Module8/Convolution/solution.cu: 99-105] Elapsed time: 0.401631 ms
[TIME][Copy][Copying data to the GPU][/home/warehouse/junseo/cuda-code-repo-joonsuuh/Module8/Convolution/solution.cu: 107-115] Elapsed time: 0.395446 ms
[TIME][Compute][Doing the computation on the GPU][/home/warehouse/junseo/cuda-code-repo-joonsuuh/Module8/Convolution/solution.cu: 117-128] Elapsed time: 0.092652 ms
[TIME][Copy][Copying data from the GPU][/home/warehouse/junseo/cuda-code-repo-joonsuuh/Module8/Convolution/solution.cu: 130-135] Elapsed time: 0.020975 ms
[TIME][GPU][Doing GPU Computation (memory + compute)][/home/warehouse/junseo/cuda-code-repo-joonsuuh/Module8/Convolution/solution.cu: 97-137] Elapsed time: 1.08277 ms
The solution is correct
[TIME][GPU][Doing GPU memory allocation][/home/warehouse/junseo/cuda-code-repo-joonsuuh/Module8/Convolution/solution.cu: 99-105] Elapsed time: 0.231705 ms
[TIME][Copy][Copying data to the GPU][/home/warehouse/junseo/cuda-code-repo-joonsuuh/Module8/Convolution/solution.cu: 107-115] Elapsed time: 0.359831 ms
[TIME][Compute][Doing the computation on the GPU][/home/warehouse/junseo/cuda-code-repo-joonsuuh/Module8/Convolution/solution.cu: 117-128] Elapsed time: 0.074439 ms
[TIME][Copy][Copying data from the GPU][/home/warehouse/junseo/cuda-code-repo-joonsuuh/Module8/Convolution/solution.cu: 130-135] Elapsed time: 0.023703 ms
[TIME][GPU][Doing GPU Computation (memory + compute)][/home/warehouse/junseo/cuda-code-repo-joonsuuh/Module8/Convolution/solution.cu: 97-137] Elapsed time: 0.852796 ms
The solution is correct
[TIME][GPU][Doing GPU memory allocation][/home/warehouse/junseo/cuda-code-repo-joonsuuh/Module8/Convolution/solution.cu: 99-105] Elapsed time: 0.35084 ms
[TIME][Copy][Copying data to the GPU][/home/warehouse/junseo/cuda-code-repo-joonsuuh/Module8/Convolution/solution.cu: 107-115] Elapsed time: 0.403491 ms
[TIME][Compute][Doing the computation on the GPU][/home/warehouse/junseo/cuda-code-repo-joonsuuh/Module8/Convolution/solution.cu: 117-128] Elapsed time: 0.074383 ms
[TIME][Copy][Copying data from the GPU][/home/warehouse/junseo/cuda-code-repo-joonsuuh/Module8/Convolution/solution.cu: 130-135] Elapsed time: 0.346616 ms
[TIME][GPU][Doing GPU Computation (memory + compute)][/home/warehouse/junseo/cuda-code-repo-joonsuuh/Module8/Convolution/solution.cu: 97-137] Elapsed time: 1.32942 ms
The solution is correct
[TIME][GPU][Doing GPU memory allocation][/home/warehouse/junseo/cuda-code-repo-joonsuuh/Module8/Convolution/solution.cu: 99-105] Elapsed time: 0.212496 ms
[TIME][Copy][Copying data to the GPU][/home/warehouse/junseo/cuda-code-repo-joonsuuh/Module8/Convolution/solution.cu: 107-115] Elapsed time: 0.361462 ms
[TIME][Compute][Doing the computation on the GPU][/home/warehouse/junseo/cuda-code-repo-joonsuuh/Module8/Convolution/solution.cu: 117-128] Elapsed time: 0.056755 ms
[TIME][Copy][Copying data from the GPU][/home/warehouse/junseo/cuda-code-repo-joonsuuh/Module8/Convolution/solution.cu: 130-135] Elapsed time: 0.021183 ms
[TIME][GPU][Doing GPU Computation (memory + compute)][/home/warehouse/junseo/cuda-code-repo-joonsuuh/Module8/Convolution/solution.cu: 97-137] Elapsed time: 0.820656 ms
The solution is correct
```

Figure 1: Convolution