CSC8014 Topics: GPU Programming **Spring 2017**

Homework #4

1. What's the purpose of texture memory? In what circumstance does texture memory increase the performance of GPU computing?

Texture memory is a type of digital storage that makes texture data readily available to GPUs, typically 3D graphics hardware. It is most often implemented as specialized TRAM that is designed for rapid reading and writing, enabling the graphics hardware increased performance in rendering 3D imagery. In some situations it will provide higher effective bandwidth by reducing memory requests to off-chip DRAM thus increasing the performance of GPU. Texture caches are designed for graphics applications where memory access patterns exhibit a great deal of *spatial locality*. In a computing application, this roughly implies that a thread is likely to read from an address "near" the address that nearby threads read.

2. Read the attached code `heat.cu` and answer the following questions:
   a. What's the purpose of calling `copy_const_kernel()` in the for loop in the method `anim_cpu()`? What expect to happen if we comment this method call out?

This makes sure when the surrounding temperature is calculated, the central heat source of the code does not change. Given some grid of input temperatures, copy the temperature of cells with heaters to this grid. This will overwrite any previously computed temperatures in these cells, thereby enforcing our restriction that "heating cells" remain at a constant temperature.

When we comment this section out the heat source in the center disappears causing the whole thing to collapse in a circular manner. The rest of the area has lesser temperature dissipated to it and everything cools down much faster.

   b. In method, the for loop iterates 90 times. Can we change the number? What difference does it make if we decrease the number? What about if we increase the number to 180?

If we decrease the loop iteration the process of heat dissipation is much slower. And upon increasing i the process is much faster. If it is increased to 180 the process of heat transfer is faster than when i was 90. This is because 90 provides a nice trade off between having to download a bitmap image for every time step and computing time steps per frame. If the i is increased by a lot it results in a jerky animation which looks laggy. If we decrease i it looks much more smoother but a lot of computation is done. 90 is a much more suitable number for i

   c. What's the purpose of calling `swap()` in the for loop in the method `anim_cpu()`?

Swap is used to swap between the old values with new values so that another new value for the grid can be computed. It swaps the input and output buffers in preparation of the next time step.The output temperature grid computed in step 2 will become the input temperature grid that we start with in step 1 when simulating the next time step.

3. Read the attached code `heat_2d.cu` where texture memory is used. Please answer the following questions:
    a. The implementation of anim_gpu() is different from the one in `heat.cu`, why is the method `swap()` no longer called?

Here we are using global memory which can not be changed like we changed the old values with new values in heat.cu. Instead we use flag to set the old values to new values where the boolean can be changed to 0 or 1 depending on which value to be used. Since the signature of blend_kernel() changed to accept a flag that switches the buffers between input and output, we need a corresponding change to the anim_gpu() routine.Rather than swapping buffers, we set dstOut = !dstOut to toggle the flag after each series of calls.

    b. How do we bind the GPU memory constant buffer of heaters, input, and output temperature to texture memory.

We use cudaBindTexture() to bind the buffers of heaters, input and output temperatures to texture memory and use cudaUnbindTexture() to unbind these.