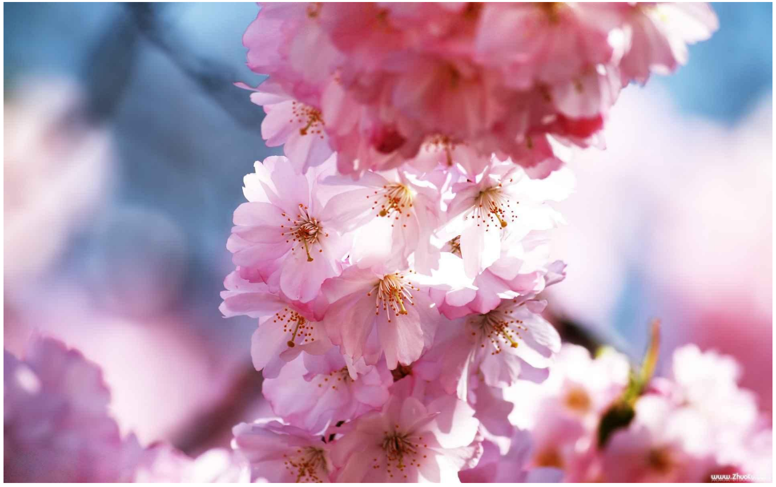
HPC Homework 4

Yuwei Jiang, N11031694

1 Iteratively blurred images

To do convolution iteratively, for CPU work, instead of convolution from the array *gray* every time, we define a new array *congray2*. We first read data from *gray* to *congray*. Then for each iteration, we store data after convolution in *congray2*, and then copy *congray2* into *congray*.
For Opencl work, we convolute from *buf_gray* to emphbuf_congray in even loops, and convolute from emphbuf_congray to emphbuf_gray in odd loops. By doing this, we don't need a new pointer. In fact, this idea can also be applied for CPU work.
As a simple test, we used a picture of cherry flower of size 1920 × 1200. The original picture is shown in Figure 1. 100 times of convolution result into Figure 2, 3, and 1000 times result into Figure 4, 5.
There might be a problem with OpenCL version of convolution for boundary points, since we can that in Figure 3 and 5, as the number of iteration increases, the boundary gets darker.

Figure 1: CPU, iter=100



2 Performance for different group size

We fixed the number of iteration to be 100, and tested on the GPU [0] Cypress at *opencl1.cims.nyu.edu*. Taking different local work group size, we obtain the following results:

w_g size	Elapsed time(s)	MPixels/s	Bandwidth(GB/s)	GFlop/s
4	0.052337	44.022322	1.408714	8.593356
8	0.012674	181.785325	5.817130	35.485316
12	0.014504	158.858204	5.083463	31.009839
16	0.010860	212.153444	6.788910	41.413310

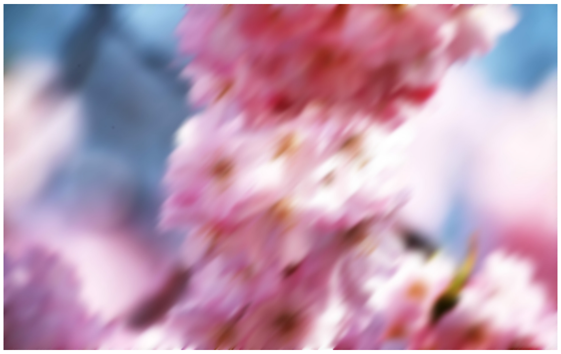


Figure 2: CPU, iter=100

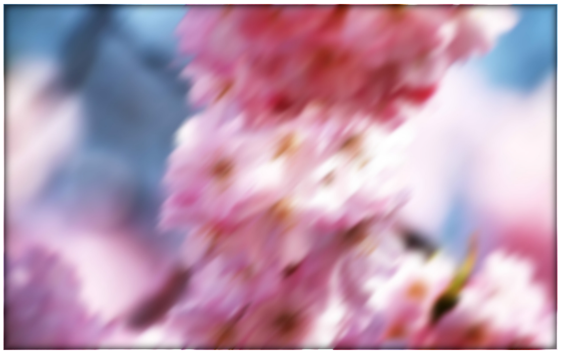


Figure 3: OpenCL, iter=100

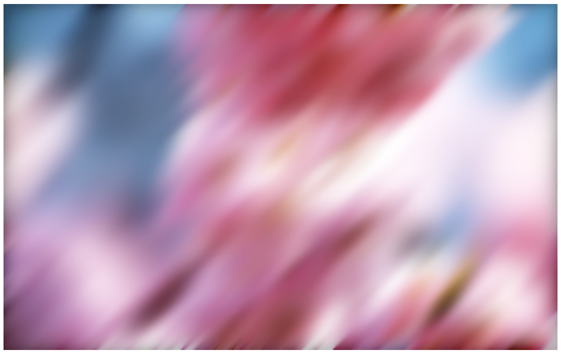


Figure 4: CPU, iter=1000

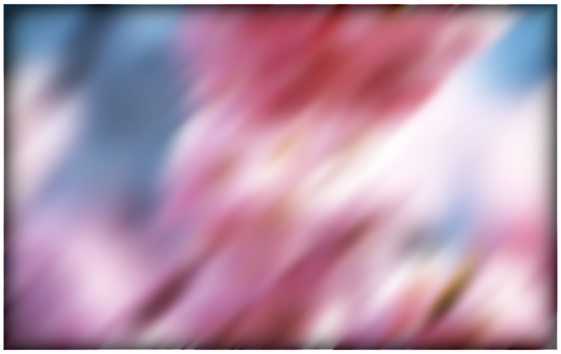


Figure 5: OpenCL, iter=1000

Further increasing the work_group_size will get the error ”invalid work group size”. The strange thing here is that the performance of size 12 is worse than both 8 and 16.