Parallel Computing

Lab 4 - Convolution

TEST	Input	Mask	Execution Time(ms)			
			Kernel 1 Without Tilling	**Kernel 2** Input Tilling	**Kernel 3** Output Tilling	**PyTorch**
1	8 images (1280 x 720)	global	**4.958**	**5.7909**	**5.247**	**2.19**
2	8 images (1280 x 720)	constant	**4.2639**	**4.73**	**3.974**	
3	16 images (1280 x 720)	global	**9.9118**	**11.577**	**10.513**	**4.1785**
4	16 images (1280 x 720)	constant	**8.5256**	**9.4609**	**7.959**	
5	32 images (1280 x 720)	global	**19.83**	**23.16**	**21.078**	**8.322**
6	32 images (1280 x 720)	constant	**17.049**	**18.925**	**15.937**	

Conclusions

1. PyTorch consistently exhibits the fastest execution time, typically completing tasks in nearly half the time compared to our implementation.
2. Across all scenarios, Kernel 2 (input tiling) consistently demonstrates the slowest performance, regardless of whether the mask is stored in constant or global memory (which seems theoretically not okay)
3. Kernel 1 (no tiling) achieves the highest speed when the mask is stored in global memory.
4. When the mask resides in constant memory, Kernel 3 (output tiling) consistently achieves the highest speed.
5. All kernels using constant mask are faster than kernels not using constant mask respectively, since constant memory in CUDA effectively magnifies memory bandwidth without consuming shared memory.