	Result	Size	Time	Cycles	GPU	SM Frequency	Process	Attributes
Current	580 - gradient_kernel	(120, 68, 1)x(16, 16, 1)	417.44 us	600,430	0 - NVIDIA GeForce GTX 1660 Ti	1.43 Ghz	[82909] image	

Summary | Details | Source | Context | Comments | Raw | Session

Compare | Tools | View | Export

This table shows all results in the report. Use the column headers to sort the results in this report. Double-click a result to see detailed metrics. Double-click on demangled names to rename it.

ID	Estimated Speedup [%]	Function Name	Demangled Name	Duration [us] (2059.65 us)	Runtime Improvement [us] (1260.53 us)	Compute Throughput [%]	Memory Throughput [%]	# Reg
3	50.57	gradient_kernel	gradient_kernel(flo...	417.44	211.11	44.25	49.43	
4	46.96	edge_kernel	edge_kernel..char..	178.40	83.77	68.06	68.06	
5	48.09	thresholding_kernel	thresholding_kernel..	88.51	42.56	31.90	51.91	
6	63.13	hough_kernel	hough_kernel..char..	881.54	556.56	81.89	19.05	

Note: Speedup estimates provide upper bounds for the optimization potential of a kernel assuming its overall algorithmic structure is kept unchanged.

Lg Throttle Stalls
Est. Speedup: 50.57%

On average, each warp of this workload spends 18.4 cycles being stalled waiting for the L1 instruction queue for local and global (LG) memory operations to be not full. Typically, this stall occurs only when executing local or global memory instructions extremely frequently. Avoid redundant global memory accesses. Try to avoid using thread-local memory by checking if dynamically indexed arrays are declared in local scope, of if the kernel has excessive register pressure causing by spills. If applicable, consider combining multiple lower-width memory operations into fewer wider memory operations and try interleaving memory operations and math instructions. This stall type represents about 63.0% of the total average of 29.3 cycles between issuing two instructions.

L1TEX Global Load Access Pattern
Est. Speedup: 29.14%

The memory access pattern for global loads from L1TEX might not be optimal. On average, only 22.6 of the 32 bytes transmitted per sector are utilized by each thread. This could possibly be caused by a stride between threads. Check the Source Counters section for uncoalesced global loads.

Uncoalesced Global Accesses
Est. Speedup: 27.27%

This kernel has uncoalesced global accesses resulting in a total of 4609584 excessive sectors (28% of the total 16746864 sectors). Check the L2 Theoretical Sectors Global Excessive table for the primary source locations. The CUDA Programming Guide has additional information on reducing uncoalesced device memory accesses.