This document includes some experiments on the speed of siftgpu. The new version refers to V311 and the old version refers to V293. **The test image is an image that has a large number of features (http://cs.unc.edu/~ccwu/siftgpu/frac.pgm), and processing normal images should have a higher speed.**

Experiment #1: Image size 1024x768, multi orientation, sub-pixel localization					
8800 GTX, 768GB GPU memory, Win XP, Intel 3G Hz CPU	Old Version (Unpacked)	New Version Unpacked		New Version Packed	
Pyramid Construction	16.1ms	16.2ms	-----	5.6ms	2.9x
Keypoint Detection	12.4ms	9.5ms	1.3x	12ms	------
Keypoint List	23.8ms	12.8ms	1.9x	11.6ms	2.0x
Orientation	13.4ms	9.5ms	1.4x	7.8ms	1.7x
Multi-O Keypoint List	11.8ms	3.9ms	3.0x	3.9ms	3.0x
Descriptor	44ms	27.8ms	1.6x	28.5ms	1.5x
Overall Speed	8.5hz	12.3hz	1.4x	14.3hz	1.7x
# Features	3082	3082		3078	

Experiment #2: Image size 1024x768, multi orientation, sub-pixel localization, up-sampling					
8800 GTX, 768GB GPU memory, Win XP , Intel 3G Hz CPU	Old Version (Unpacked)	New Version Unpacked		New Version Packed	
Pyramid Construction	51.4ms	52ms	-----	15.4ms	3.3x
Keypoint Detection	38.6ms	33.3ms	1.2x	39ms	------
Keypoint List	30.8ms	19.4	1.6x	15.7ms	2.0x
Orientation	25.4ms	13.2	1.9x	14.2ms	1.8x
Multi-O Keypoint List	15.0ms	5.5	2.7x	5.1ms	2.9x
Descriptor	106ms	68.5	1.6x	73.6ms	1.4x
Overall Speed	3.8hz	5.2hz	1.5x	6.1hz	1.6x
# Features	10769	10771		10768	

What the new version has done to make every step faster

1. Pyramid construction:
    a. the new unpacked version is exactly the same as the old version;
    b. the packed version has a better utilization of GPU texture lookup bandwidth by using all of the four channels to store intensities (compared with 1 in the unpacked version)
2. Keypoint detection:
    a. The new unpacked version skips some unnecessary gradient computation
    b. The packed version is a little bit slower than previous version, because the DOG computation is now done in this stage (in pyramid construction stage for unpacked)
3. Keypoint List

    Previous timing is using a pure-GPU implementation. Now I have a GPU/CPU mixed implementation, which use GPU to do reduction first and process on CPU when reduction size is small. The combination of them is faster than both pure-GPU and pure-CPU version.
4. Orientation

    Previous version uses dynamic vector indexing by default, which turned out to be slower than a set of unrolled conditional writes. Although current GPU supports dynamic vector indexing, we should avoid using them.
5. Multi-Orientation Keypoint List

    Previous version uses GPU, while new versions use only CPU for this part. The same reason as the keypoint list generation, when the data size is small, GPU-reduction is slower than CPU.
6. Descriptor Computation

    Previous version uses 8 shaders / 4 render targets to write one descriptor, while new version uses 16 shaders /2 render targets. I also tried 1 shader/8 render targets, which is even slower. It must be better to use more shaders together to finish a complicated job.

The next page gives the experiments on GTX280

GTX 280 Experiments

Experiment #1: Image size 1024x768, multi orientation, sub-pixel localization						
	New Version Unpacked			New Version Packed		
	8800 GTX	GTX 280		8800 GTX	GTX 280	
Pyramid Construction	16.2ms	13.2	1.2x	5.6ms	4.8	1.2x
Keypoint Detection	9.5ms	2.9	3.3x	12ms	4.8	2.5x
Keypoint List	12.8ms	6.5	1.9x	11.6ms	5.8	2x
Orientation	9.5ms	11.3	-----	7.8ms	9.5	------
Multi-O Keypoint List	3.9ms	2.3	1.7x	3.9ms	2.2	1.8x
Descriptor	27.8ms	18.5	1.5x	28.5ms	14.8	1.9x
Overall Speed	12.3hz	18.0hz	1.5x	14.3hz	23hz	1.6x

Experiment #2: Image size 1024x768, multi orientation, sub-pixel localization, up-sampling						
	New Version Unpacked			New Version Packed		
	8800 GTX	GTX 280		8800 GTX	GTX 280	
Pyramid Construction	52ms	41.3	1.3x	15.4ms	12.5	1.2x
Keypoint Detection	33.3ms	10.3	3.2x	39ms	16.2	2.4x
Keypoint List	19.4	10.2	1.9x	15.7ms	8.1	1.9x
Orientation	13.2	20.3	------	14.2ms	16.1	---------
Multi-O Keypoint List	5.5	3.2	1.7x	5.1ms	3.2	1.6x
Descriptor	68.5	45.5	1.5x	73.6ms	32.6	2.3x
Overall Speed	5.2hz	7.6hz	1.5x	6.1hz	11.2	1.8x

It can be seen that GTX 280 gives a 50%-80% speedup over 8800GTX.

Below are some additional experiments on other image sizes.

Experiment #1: multi orientation, sub-pixel localization, packed, up-sampling (-m –s –pack –fo -1) **System**: 8800 GTX, 768GB GPU memory, Win XP , Intel 3G Hz CPU					
	320x240	640x480	800x600	1024x768	1280x960
Pyramid Construction	7.9ms	10.0ms	13.4ms	15.4ms	29.4ms
Keypoint Detection	6.8ms	16.8ms	25.3ms	39.0ms	59.0ms
Keypoint List	10.0ms	13.8ms	15.1ms	15.7ms	20.1ms
Orientation	6.1ms	8.3ms	10.5ms	14.2ms	18.8ms
Multi-O Keypoint List	3.1ms	4.6ms	5.1ms	5.1ms	6.4ms
Descriptor	17.5ms	37.5ms	50.5ms	73.6ms	99.6ms
Overall Speed	20.0hz	10.9hz	8.4hz	6.1hz	4.3hz
# Features	1198	4473	6832	10768	16615

Experiment #2: multi orientation, sub-pixel localization, packed (-m –s –pack) **System**: 8800 GTX, 768GB GPU memory, Win XP , Intel 3G Hz CPU							
	320x240	640x480	800x600	1024x768	1280x960	1600x1200	2048x1536
Pyramid	6.5ms	7.7ms	8.8ms	8.8ms	9.9ms	13.9ms	20.7ms
Detection	3.5ms	6.7ms	9.3ms	12.5ms	16.7ms	24.7ms	39.2ms
Keypoint List	6.6ms	9.7ms	10.8ms	12.1ms	14.1ms	16.2ms	17.4ms
Orientation	4.4ms	6.3ms	6.8ms	7.9ms	9.1ms	12.0ms	14.6ms
MOK List	2.3ms	3.5ms	3.4ms	4.3ms	4.7ms	5.4ms	5.7ms
Descriptor	10.7ms	17.7ms	22.1ms	28.5ms	38.2ms	52.3ms	75.5ms
Overall Speed	29.0hz	19.6hz	16.4hz	13.5hz	10.9hz	8.0hz	5.8hz
Features	348	1282	1937	3078	4758	7269	11588