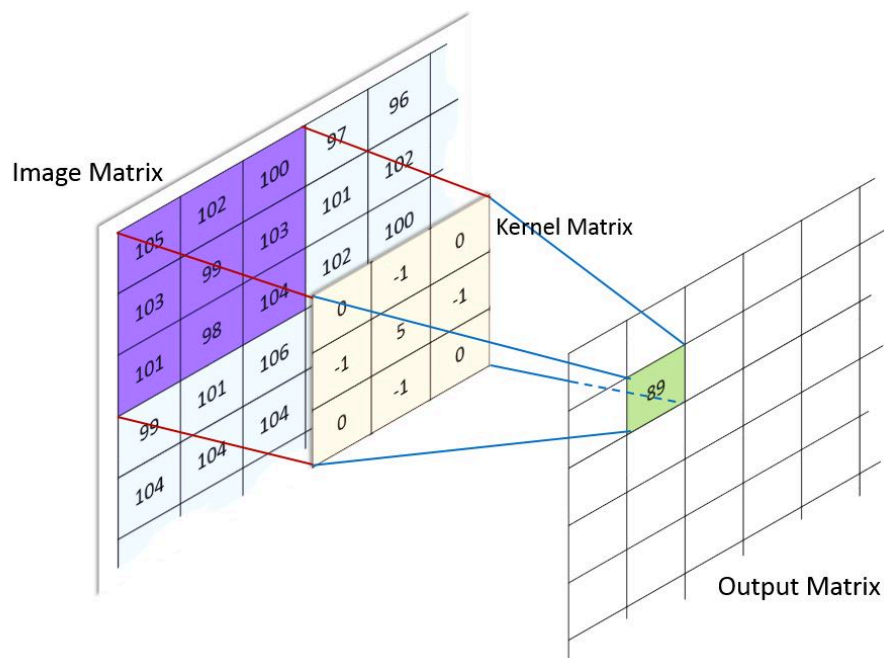


Documentación:

Análisis y Resultados de Image Blurring.

En esta práctica se implementaron tres diferentes formas de hacerle un “blur” a una imagen mediante el procesamiento con un filtro o núcleo (kernel). Este kernel se “sobrepone” a la matriz de la información de la imagen y mediante operaciones matemáticas (en este caso un promedio de multiplicaciones) y así modificar la información de la matriz de la imagen que resulta con alteraciones a la imagen como tal. Puede ser cambio en colores, saturación pero en este caso es el efecto de “blurring”.



Para el proyecto se implementó la teoría pasada en 3 principales ambientes:

1. Operaciones en CPU secuencialmente (sin threads).
2. Operaciones en CPU paralelo (con threads).
3. Operaciones en GPU paralelo.

Se llevaron a cabo 10 pruebas por cada configuración de threads y los resultados en ms se muestran a continuación (después de las especificaciones).

Especificaciones de la computadora:

Para las pruebas en CPU:

Macbook Pro 13' early 2011:

- 2.3GHz dual-core Intel Core i5 processor con 3MB shared L3 cache
- Intel HD Graphics 3000 with 384MB of DDR3 SDRAM shared with main memory.

Para las pruebas en GPU:

El GPU utilizado fue el del servidor proporcionado que cuenta con un GeForce GTX 670 (en la siguiente página):

GPU Engine Specs:	
CUDA Cores	1344
Graphics Clock (MHz)	915
Processor Clock (MHz)	980
Texture Fill Rate (billion/sec)	102.5
Memory Specs:	
Memory Clock	6.0 Gbps
Memory Interface	GDDR5
Memory Interface Width	256-bit GDDR5
Memory Bandwidth (GB/sec)	192.2

Los siguientes son los resultados de todas las pruebas con tiempos en ms.

Cálculo en CPU sin Threads:

Tamaño de imagen	480p	1080p	4k
Tiempos en ms.	123.2	1555.16	6532.59
Tiempos en ms.	140.166	1680.63	6249.14
Tiempos en ms.	140.282	1655.6	6279.96
Tiempos en ms.	185.658	1675.2	6795.48
Tiempos en ms.	239.996	1673.73	7461.34
Tiempos en ms.	135.636	1690.38	6326.38
Tiempos en ms.	130.775	1617.94	6126.66
Tiempos en ms.	152.518	1527.86	6131.58
Tiempos en ms.	135.861	1712.64	6120.47

Tamaño de imagen	480p	1080p	4k
Tiempos en ms.	130.344	1692.34	6144.68
Promedio en ms.	151.4436	1648.148	6416.828

Cálculo en CPU con Threads:

Utilizando 4 threads :

Tamaño de imagen	480p	1080p	4k
Tiempos en ms.	23.7614	479.615	1059.89
Tiempos en ms.	38.1021	266.073	1059.14
Tiempos en ms.	38.8762	265.576	1061.81
Tiempos en ms.	23.9406	266.097	1643.39
Tiempos en ms.	39.1544	448.65	1245.68
Tiempos en ms.	41.474	477.452	1206.18
Tiempos en ms.	39.3659	478.539	1468.51
Tiempos en ms.	41.3992	265.482	1058.89
Tiempos en ms.	23.9897	481.651	1099.88
Tiempos en ms.	41.7486	266.304	1059.41
Promedio en ms.	35.18121	369.5439	1196.278

Utilizando 6 threads :

Tamaño de imagen	480p	1080p	4k
Tiempos en ms.	28.6325	327.079	1190.57
Tiempos en ms.	28.6067	310.918	1297.85
Tiempos en ms.	28.5713	297.747	1261.49

Tamaño de imagen	480p	1080p	4k
Tiempos en ms.	28.699	325.396	1296.51
Tiempos en ms.	29.1796	324.39	1266.6
Tiempos en ms.	33.0685	304.183	1095.42
Tiempos en ms.	28.5785	324.178	1269.98
Tiempos en ms.	27.4099	351.265	1300.78
Tiempos en ms.	28.488	324.307	1297.62
Tiempos en ms.	35.4829	326.264	1293.43
Promedio en ms.	29.67169	321.5727	1257.025

Cálculo en GPU:

Utilizando 256 threads (16x16):

Tamaño de imagen	480p	1080p	4k
Tiempos en ms.	0.017994	0.014700	0.022127
Tiempos en ms.	0.013655	0.028960	0.030842
Tiempos en ms.	0.013781	0.015407	0.031471
Tiempos en ms.	0.012883	0.025772	0.022428
Tiempos en ms.	0.022143	0.016032	0.022415
Tiempos en ms.	0.013320	0.017067	0.034244
Tiempos en ms.	0.013157	0.016298	0.035892
Tiempos en ms.	0.015204	0.015348	0.021494
Tiempos en ms.	0.013552	0.014701	0.033591
Tiempos en ms.	0.035762	0.015591	0.032421
Promedio en ms.	0.0171451	0.0179876	0.0286925

Utilizando 512 threads (16x32):

Tamaño de imagen	480p	1080p	4k
Tiempos en ms.	0.012612	0.014667	0.025898
Tiempos en ms.	0.012960	0.014646	0.025045
Tiempos en ms.	0.013639	0.014619	0.024941
Tiempos en ms.	0.013579	0.014844	0.026159
Tiempos en ms.	0.013152	0.014508	0.048001
Tiempos en ms.	0.013608	0.016962	0.027639
Tiempos en ms.	0.012519	0.014423	0.022417
Tiempos en ms.	0.013885	0.016418	0.021966
Tiempos en ms.	0.013561	0.015060	0.032157
Tiempos en ms.	0.013518	0.028238	0.022260
Promedio en ms.	0.0133033	0.0164385	0.0276483

Utilizando 512 threads (2x256):

Tamaño de imagen	480p	1080p	4k
Tiempos en ms.	0.014066	0.015684	0.027555
Tiempos en ms.	0.013087	0.016193	0.021762
Tiempos en ms.	0.013122	0.015813	0.031260
Tiempos en ms.	0.014924	0.013798	0.030727
Tiempos en ms.	0.012911	0.014640	0.020572
Tiempos en ms.	0.013736	0.015770	0.021546
Tiempos en ms.	0.013186	0.016475	0.024176
Tiempos en ms.	0.013805	0.017177	0.027038
Tiempos en ms.	0.013500	0.015589	0.024293

Tamaño de imagen	480p	1080p	4k
Tiempos en ms.	0.013692	0.017457	0.025664
Promedio en ms.	0.0136029	0.0158596	0.0254593

Utilizando 1024 threads (32x32):

Tamaño de imagen	480p	1080p	4k
Tiempos en ms.	0.014677	0.015929	0.032451
Tiempos en ms.	0.014228	0.017713	0.030884
Tiempos en ms.	0.023433	0.015956	0.031547
Tiempos en ms.	0.013215	0.015303	0.024022
Tiempos en ms.	0.012261	0.029534	0.020580
Tiempos en ms.	0.013101	0.014894	0.028881
Tiempos en ms.	0.013727	0.014361	0.031298
Tiempos en ms.	0.013325	0.017416	0.032033
Tiempos en ms.	0.012830	0.016957	0.027386
Tiempos en ms.	0.014025	0.014821	0.032416
Promedio en ms.	0.0144822	0.0172884	0.0291498

Utilizando 128 threads (1x128):

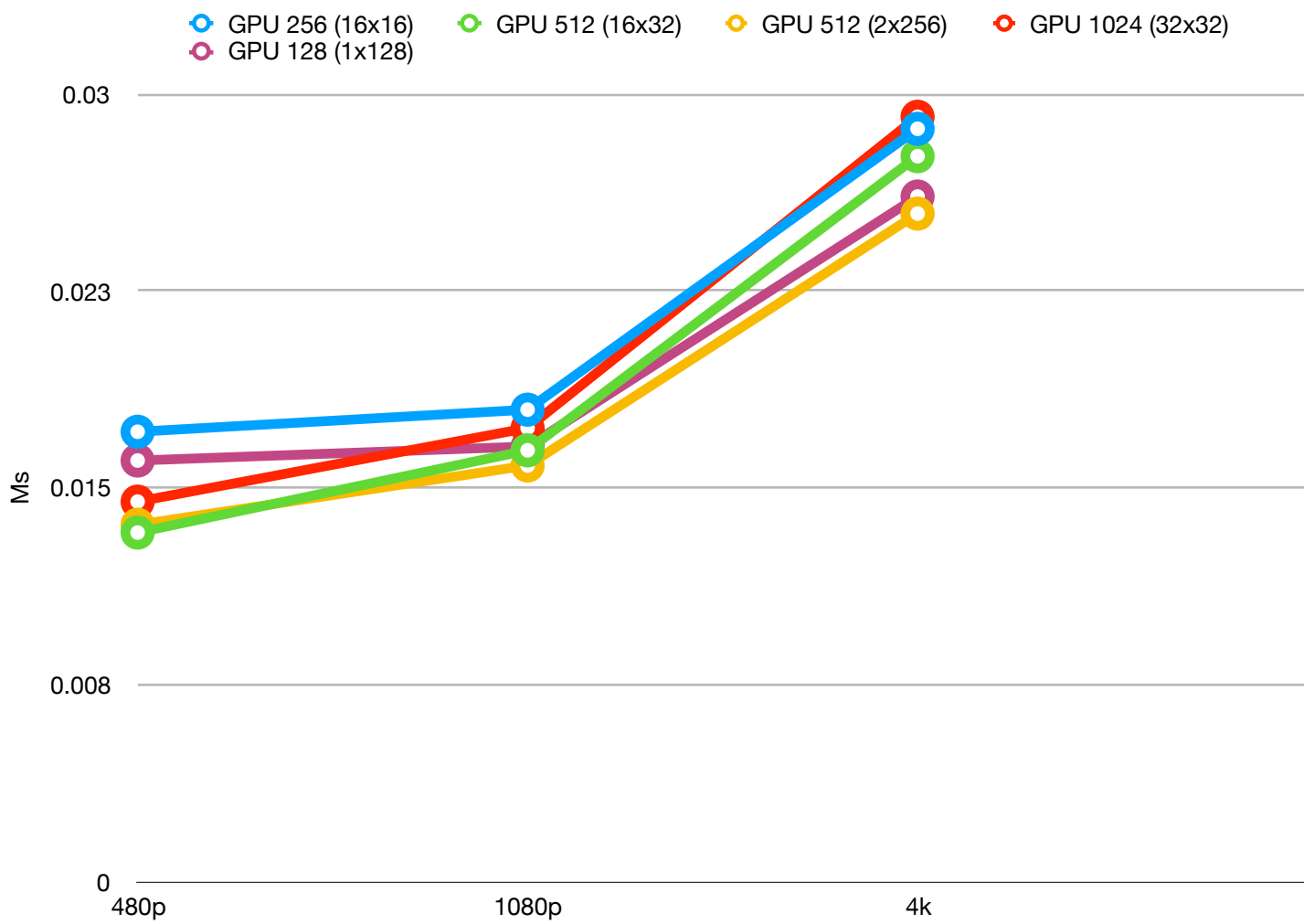
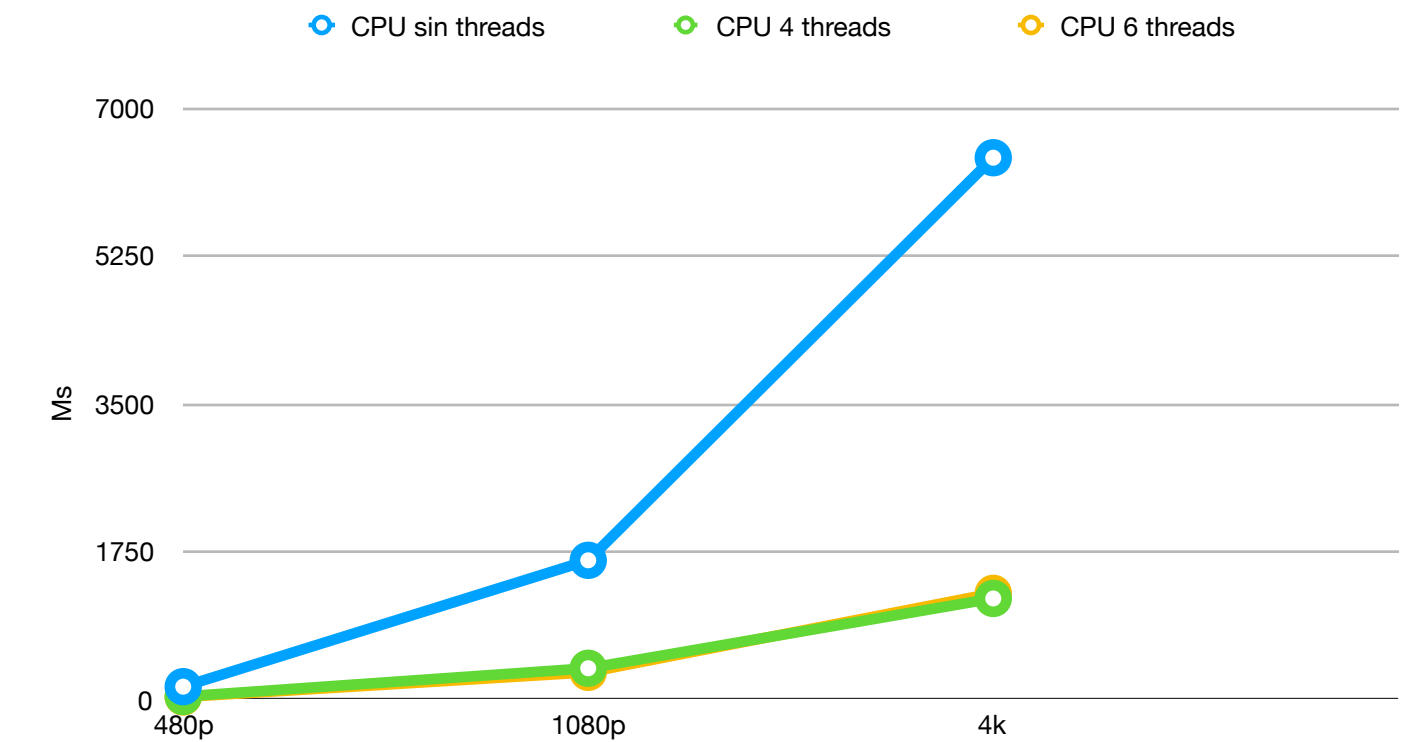
Tamaño de imagen	480p	1080p	4k
Tiempos en ms.	0.013342	0.015380	0.032058
Tiempos en ms.	0.015693	0.013773	0.031914
Tiempos en ms.	0.013183	0.016414	0.033675
Tiempos en ms.	0.013691	0.017492	0.021488
Tiempos en ms.	0.013271	0.015744	0.021100

Tamaño de imagen	480p	1080p	4k
Tiempos en ms.	0.015050	0.015709	0.020912
Tiempos en ms.	0.013217	0.014693	0.022008
Tiempos en ms.	0.013051	0.016171	0.024650
Tiempos en ms.	0.036242	0.024951	0.033120
Tiempos en ms.	0.013750	0.015513	0.020181
Promedio en ms.	0.016049	0.016584	0.0261106

Speedup:

Tamaño de imagen	480p	1080p	4k
CPU sin threads	151.4436	1648.148	6416.828
CPU 4 threads	35.18121	369.5439	1196.278
CPU 6 threads	29.67169	321.5727	1257.025
GPU 256 (16x16)	0.0171451	0.0179876	0.0286925
GPU 512 (16x32)	0.0133033	0.0164385	0.0276483
GPU 512 (2x256)	0.0136029	0.0158596	0.0254593
GPU 1024 (32x32)	0.0144822	0.0172884	0.0291498
GPU 128 (1x128)	0.016049	0.016584	0.0261106

Gráficas de la información promedio en la siguiente página.



Análisis y Conclusiones:

Como se puede observar de los resultados anteriores, tanto en la tabla marcado en amarillo como en las gráficas; el tiempo se beneficia de una manera bastante importante cuando existe más cantidad de threads que dividan el trabajo de procesar e ir cambiando la imagen para el output. A diferencia del proyecto anterior donde por la forma en la que se acomodaba el arreglo importaba más el peso que tuviera ya sea en columnas o en filas en el tiempo. En este caso lo importante es el número de threads ya que las imágenes representan matrices de un tamaño enorme por lo que el mayor número de operaciones en paralelo ayudan. Por supuesto que existe una necesidad para manejar todos estos threads pero este tiempo no afecta tanto el tiempo entre las operaciones más lentas y las más rápidas.

Por supuesto se puede observar que no existe comparación entre los tiempos en CPU y GPU, algo que los cuda cores proporcionan en cuanto a “paralelización”. Pasa de ser unos cuantos milisegundos a no llegar ni a un milisegundo. Otro elemento que prueba que mayor threads se reduce el tiempo es que fuera de las comparaciones “oficiales” que se llevaron a cabo, también se midió el tiempo del archivo 4k utilizando 1024x1024 threads y los resultados mantienen esta rapidez. Las pruebas fueron en su mayoría en el rango de **0.008 a 0.009** milisegundos, entonces mucho más rápido que cualquiera de los otros casos en cuanto a thread count.

Referencias:

Imagen principal de: http://machinelearningguru.com/computer_vision/basics/convolution/image_convolution_1.html

Información de hardware:

<https://www.geforce.com/hardware/desktop-gpus/geforce-gtx-670/specifications>

https://support.apple.com/kb/sp619?locale=en_US