CONVOLUCIÓN EN DOS DIMENSIONES:
FILTRO DE SOBEL
APLICADO A UNA IMAGEN
SECUENCIAL, BÁSICA, CON CATCHING Y CON TILING

HIGH PERFORMANCE COMPUTING (HPC)
PARCIAL #2

ESTUDIANTES:
Diego Alejandro Ramirez Ramirez
Carlos Eduardo Zuleta Uribe

PROFESOR:
John H. Osorio

UNIVERSIDAD TECNOLÓGICA DE PEREIRA
FACULTAD DE INGENIERÍAS
INGENIERÍA DE SISTEMAS Y COMPUTACIÓN

21/10/2015

ESPECIFICACIONES DEL SISTEMA

La máquina en la cual se corrió el algoritmo tiene las siguientes características:

CPU:

processor : 1
vendor_id : GenuineIntel
cpu family : 6
model : 58
model name : Intel® Core™ i7-3770K CPU @ 3.50GHz
stepping : 9
cpu MHz : 1600.000
cache size : 8192 KB
cpu cores : 4

GPU:

Tesla K40c: 3.5
Global memory: 11519mb
Shared memory: 48kb
Constant memory: 64kb
Block registers: 65536
Warp size: 32
Threads per block: 1024
Max block dimensions: [1024, 1024, 64]
Max grid dimensions: [2147483647, 65535, 65535]

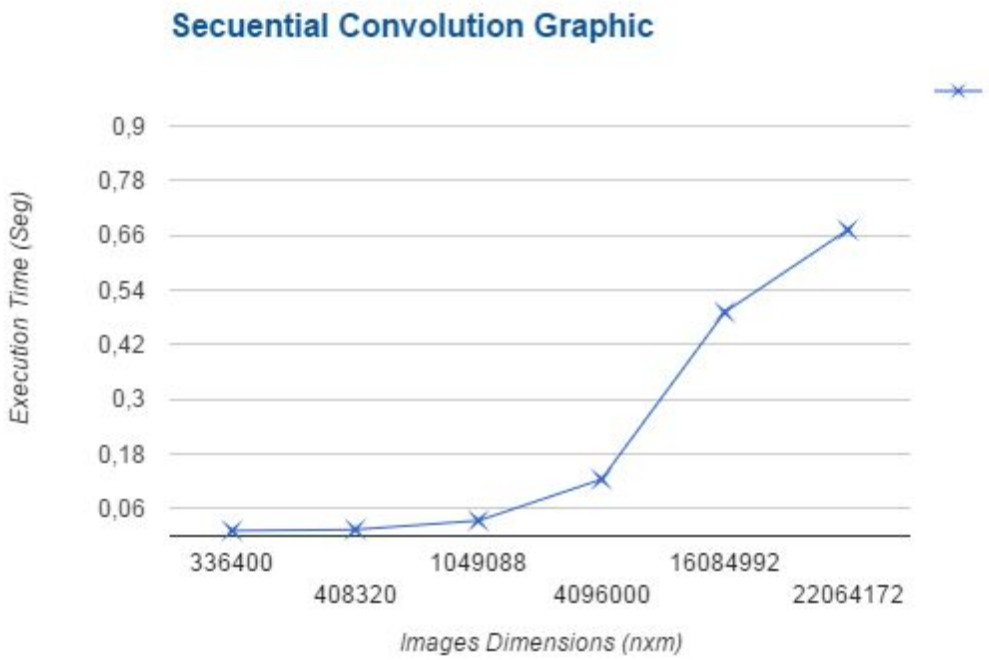
PROCESO APLICADO

Se tomaron 20 datos para cada uno de las imágenes con las cuatro diferentes implementaciones del algoritmo: Convolución 2D secuencial aplicando el filtro de sobel tanto en el eje X como en el eje Y, utilizando la librería opencv. para la convolución 2D con memoria global, convolución 2D con memoria constante y convolución 2D con memoria compartida también se realizó el trabajo en los dos ejes; todo esto dejando un valor constante para la Mask, para este caso un valor de 3, luego se calculó una media más precisa. Finalmente se halló la aceleración entre Convolución 2D secuencial respecto a cada una de las demás y la aceleración de la implementación con memoria constante con respecto a la implementación con memoria compartida, tomando el valor medio de cada tiempo Secuencial y dividiéndolo entre cada valor de tiempo medio de los otros tres respectivamente, del mismo modo para la Aceleración de convolución con memoria compartida en este caso, tiempo de Convolución con constante dividido entre tiempo de Convolución con compartida.

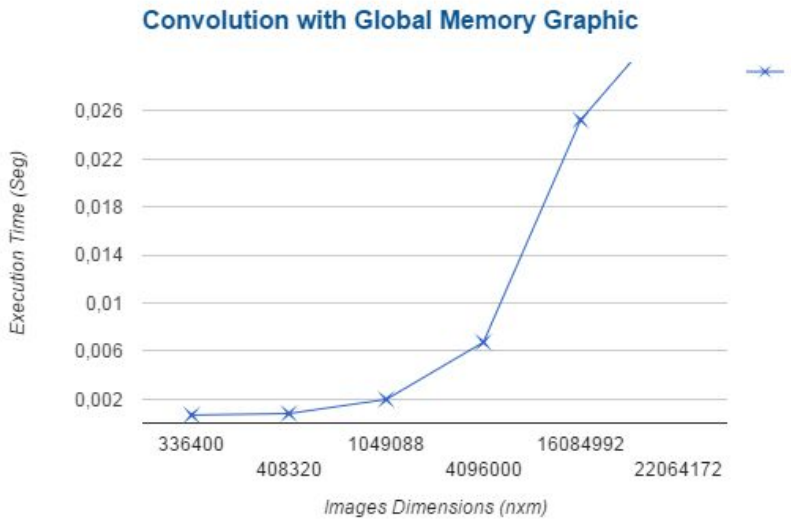
En la siguiente tabla se muestra con más detalle lo mencionado anteriormente.

Images Dimensions			**Time 2D Convolution**				**Accelerations**			
n	**m**	**mxn**	**Secuencial**	**Memoria Global**	**Memoria Constante**	**Memoria Compartida**	**CPU /GPU**	**CPU/GPUC**	**CPU/GPUT**	**GPUC/GPUT**
580	580	336400	0,01164620	0,00066790	0,00048530	0,00042675	17,43704147	23,99793942	27,29045108	1,137199766
638	640	408320	0,01402465	0,00079050	0,00056470	0,00048935	17,74149273	24,83557641	28,65975273	1,153979769
1366	768	1049088	0,03357095	0,00196815	0,00142980	0,00118100	17,05710947	23,47947265	28,42586791	1,210668925
2560	1600	4096000	0,12425716	0,00673045	0,00437665	0,00388580	18,46193939	28,39092914	31,97724021	1,126318905
4928	3264	16084992	0,49057445	0,02524620	0,01653010	0,01407660	19,43161545	29,67764563	34,85035094	1,17429635
5226	4222	22064172	0,67120183	0,03459575	0,02229240	0,01943275	19,40127993	30,10899813	34,53972443	1,147156218

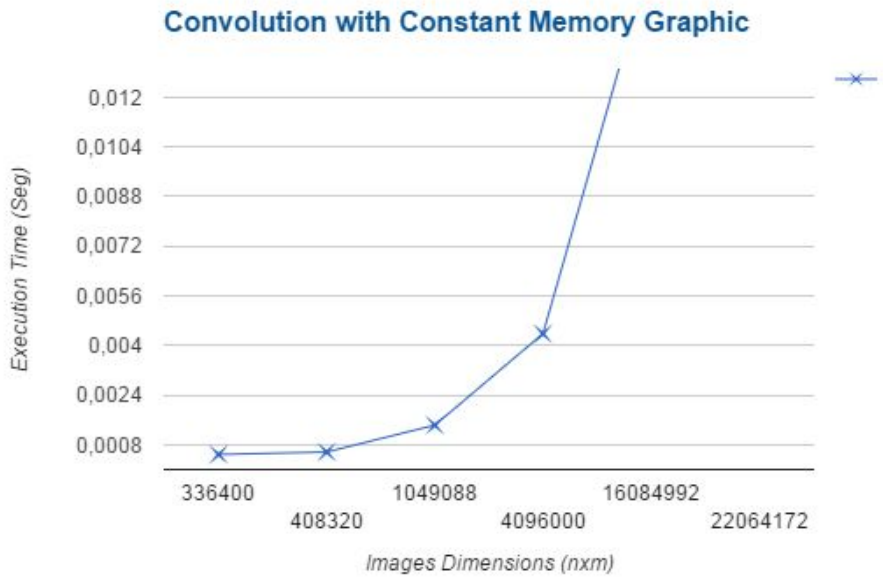
Gráficos de tiempos con cada una de las implementaciones.



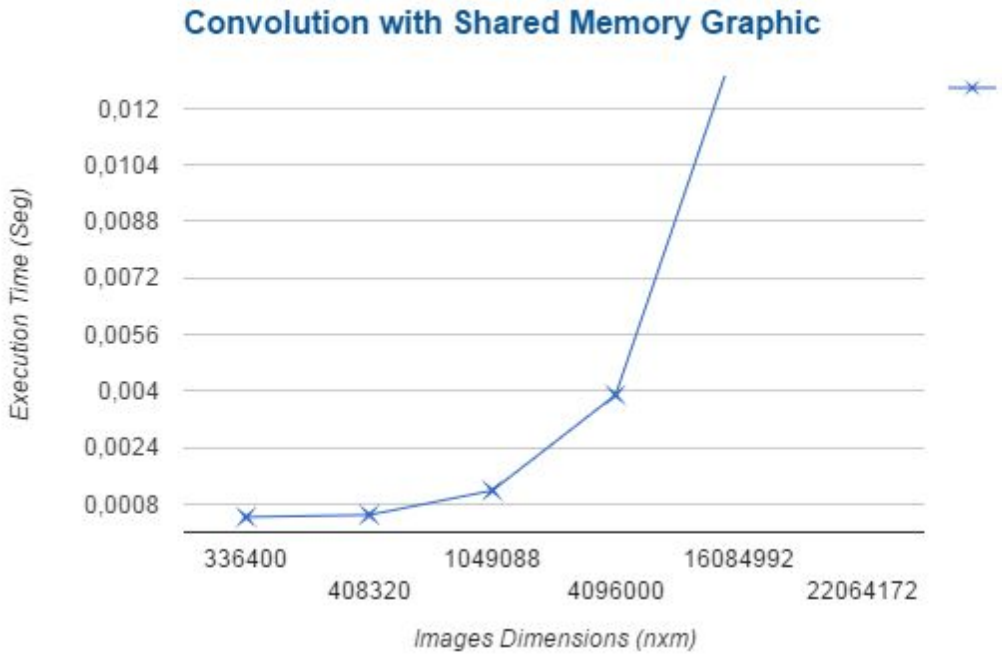
Gráfica que representa la ejecución de la implementación secuencial para cada imagen a la que se le aplicó el algoritmo; tiempos de ejecución por imagen contra las dimensiones de cada imagen.



Gráfica que representa la ejecución de la implementación con Memoria Global para cada imagen a la que se le aplicó el algoritmo; tiempos de ejecución por imagen contra las dimensiones de cada imagen.

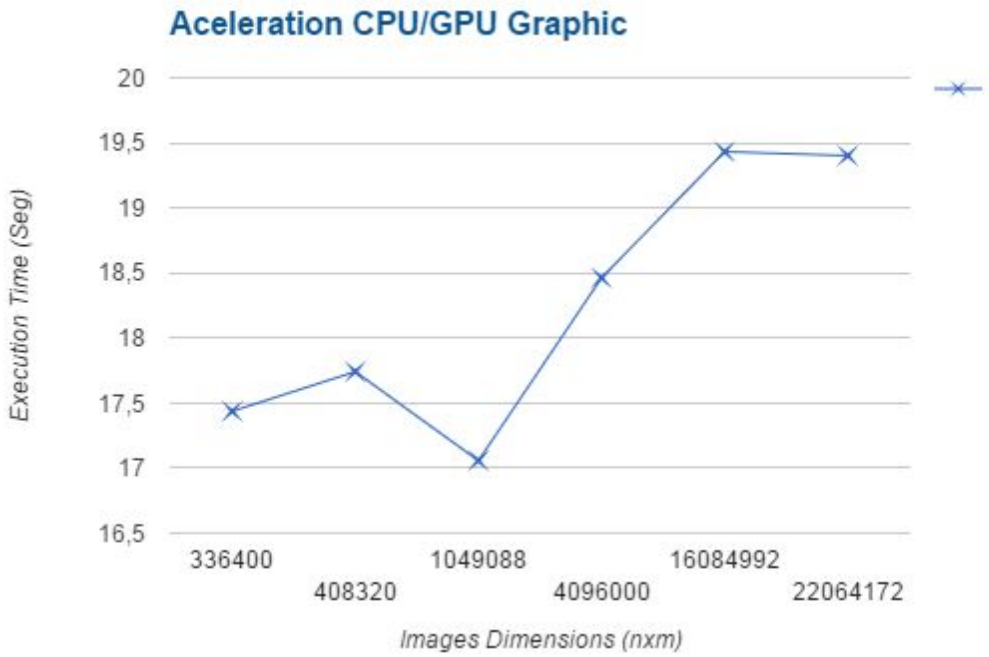


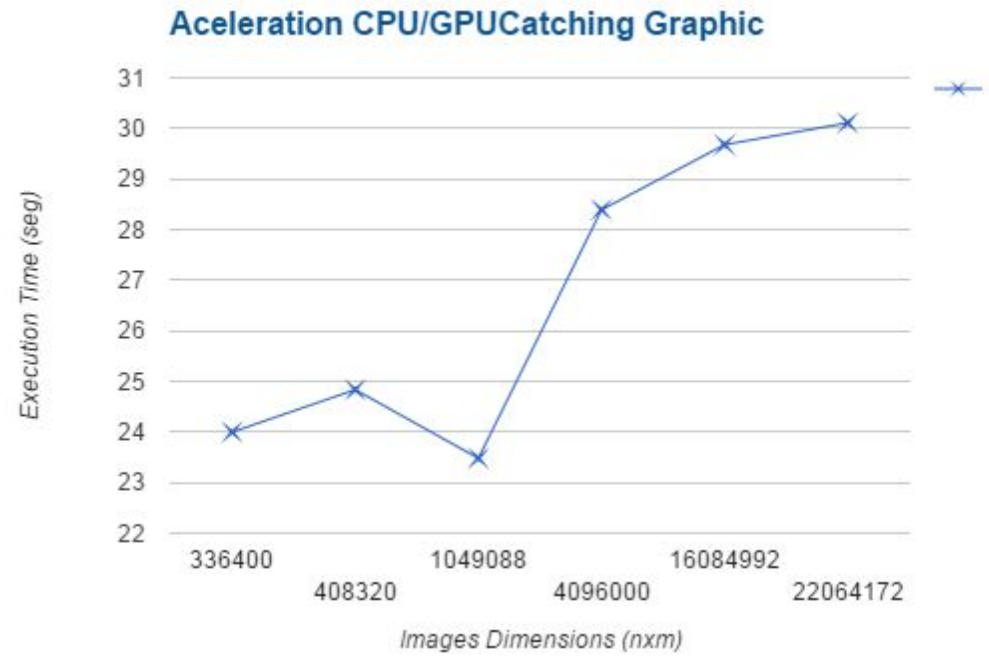
Gráfica que representa la ejecución de la implementación con Memoria Constante para cada imagen a la que se le aplicó el algoritmo; tiempos de ejecución por imagen contra las dimensiones de cada imagen.

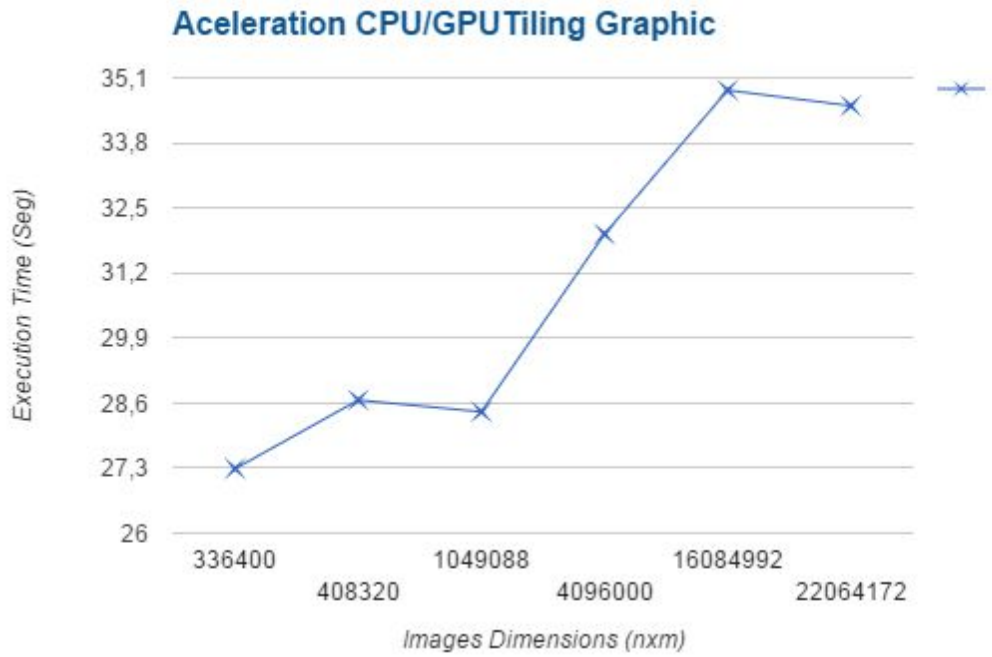


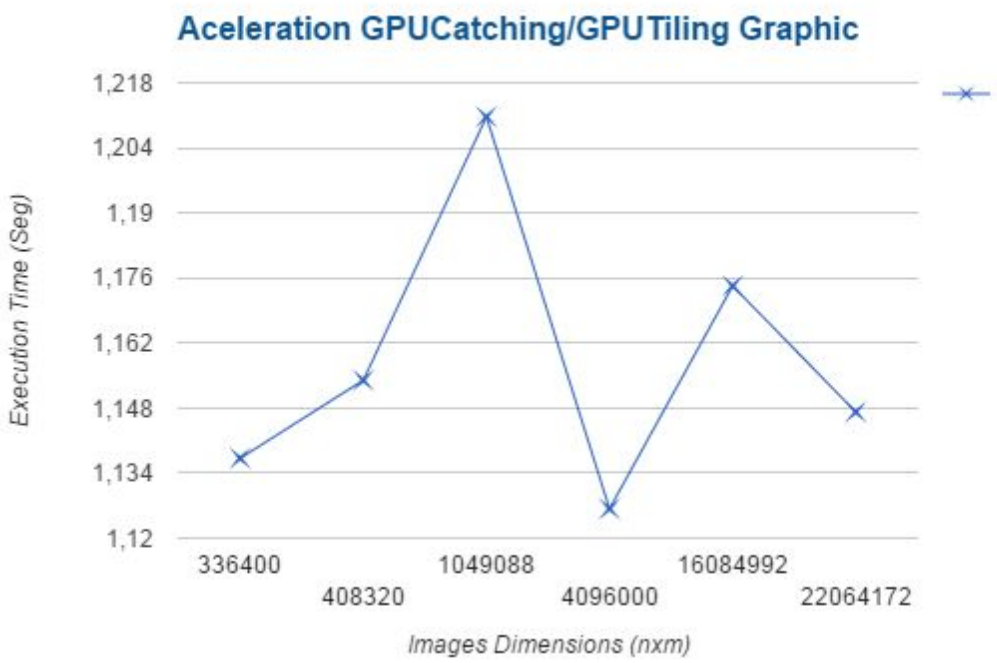
Gráfica que representa la ejecución de la implementación con Memoria Compartida para cada imagen a la que se le aplicó el algoritmo; tiempos de ejecución por imagen contra las dimensiones de cada imagen.

A continuación se mostrarán las gráficas para cada una de las aceleraciones obtenidas en las cuatro diferentes implementaciones de Convolución 2D.

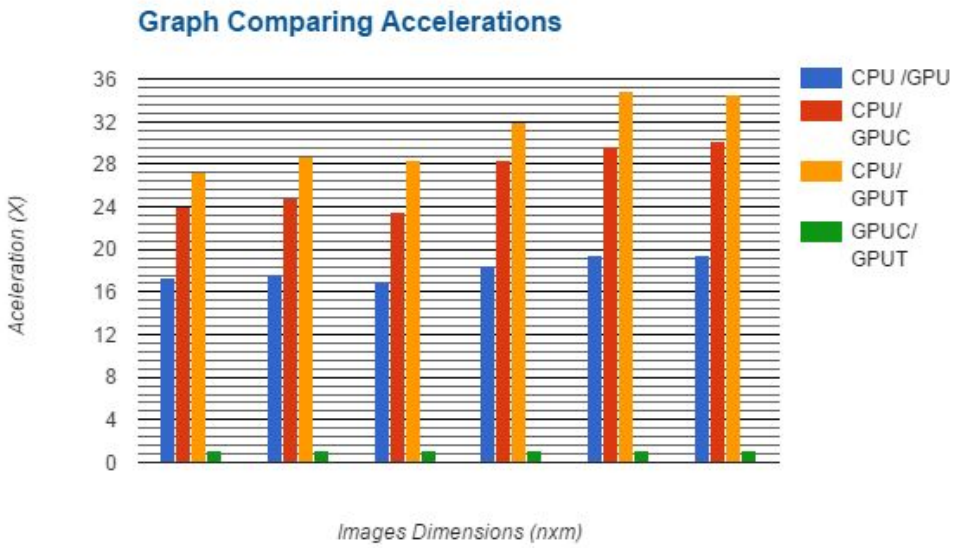





Aceleration CPU/GPUTiling Graphic
35,1
33,8
32,5
31,2
29,9
28,6
27,3
26
Execution Time (Seg)
336400
408320
1049088
4096000
16084992
22064172
Images Dimensions (nxm)


Aceleration GPUCatching/GPUTiling Graphic
1,218
1,204
1,19
1,176
1,162
1,148
1,134
1,12
Execution Time (Seg)
336400
408320
1049088
4096000
16084992
22064172
Images Dimensions (nxm)

Aquí se muestra la gráfica comparativa de las aceleraciones respectivas:



Comparación de imágenes de entrada con la salida después de pasar por el filtro de sobel en el eje X y en el eje Y. La imagen a la izquierda es la original, y obviamente la imagen de la derecha es la imagen resultante después de haberle aplicado mediante el algoritmo de convolución el filtro de Sobel.

Imagen 1

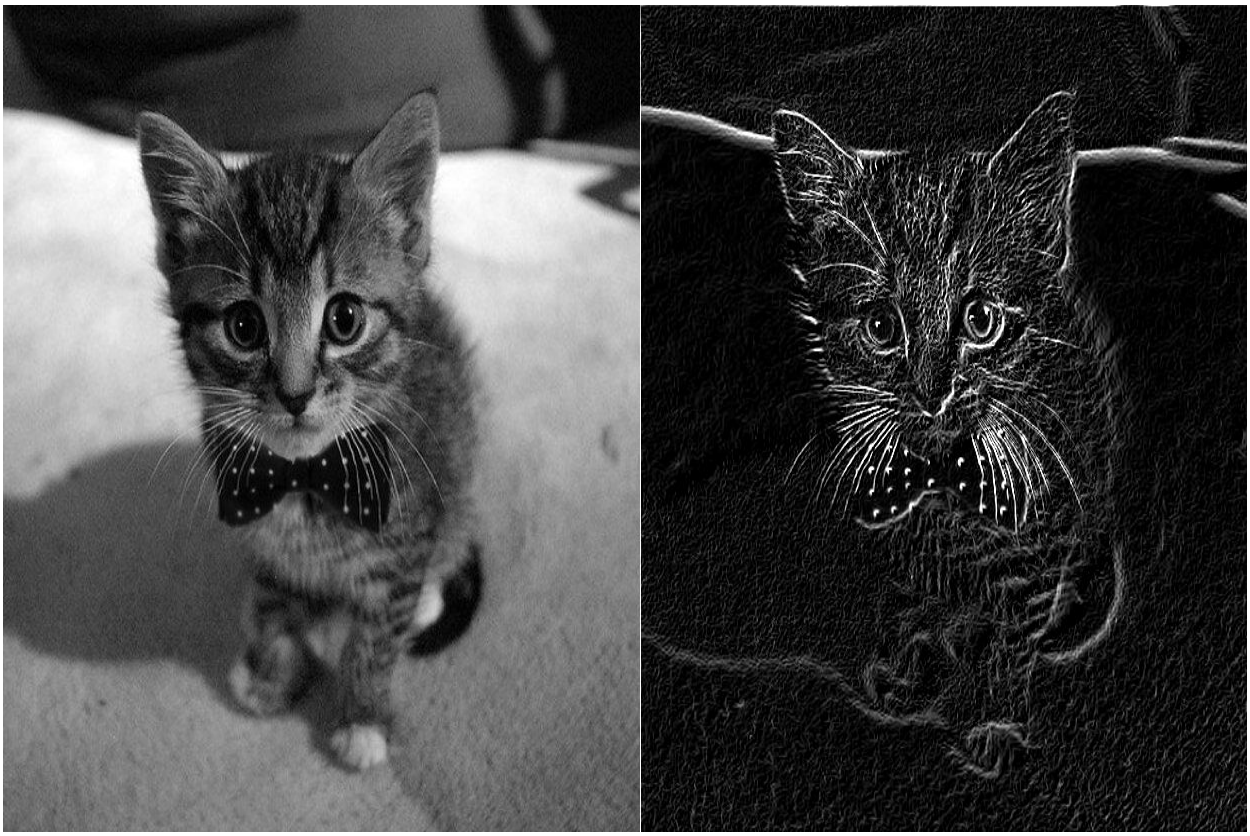


Imagen 2

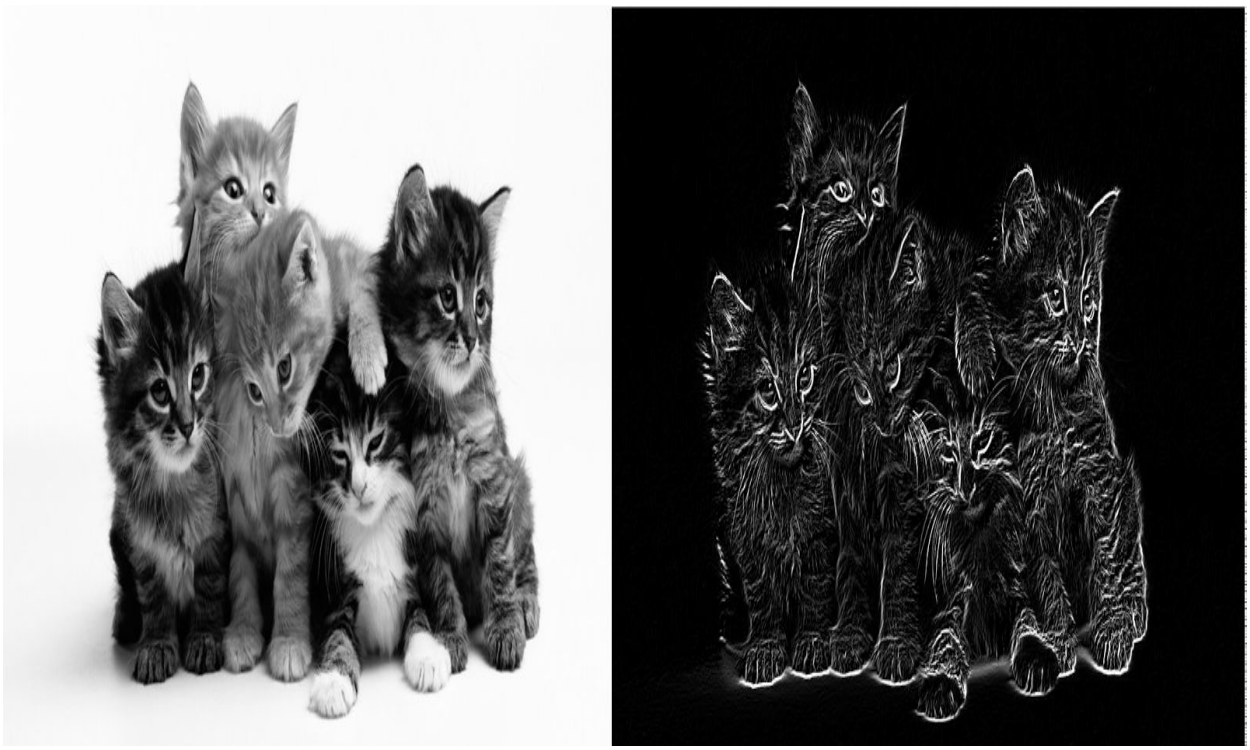


Imagen 3

CONCLUSIONES

Analizando el trabajo realizado se encontró que:

- Al comprobar la aplicación del filtro de Sobel a cada una de las imágenes pudimos observar que con la implementación con Memoria Global queda mejor resaltada la imagen resultado.
- Podemos ver claramente en la gráfica comparativa de las aceleraciones, que la aceleración de Memoria Constante con respecto a Memoria Compartida es bastante parecida en todas las ejecuciones con las diferentes imágenes con respecto a las demás aceleraciones.
- Se evidenció que la implementación de convolución 2D con memoria compartida presenta un mejor rendimiento y una mayor aceleración comparado con las demás implementaciones.
- En la aceleración de la implementación de Convolución 2D con Memoria Constante con respecto a la de Memoria Compartida no es muy grande dicha aceleración, se mantiene muy constante; demasiado pequeña.