

# Informe Convolución CUDA

Comparación entre modelos de Convolución para aplicar  
Filtro de Sobel

Rafael Pinzón Rivera  
Christian Patiño Grisales



Universidad Tecnológica de Pereira  
Pereira  
Octubre 2015

# Índice general

<b>1. Introducción</b>	<b>4</b>
1.1. Filtro de Sobel . . . . .	4
1.2. Convolución Básica . . . . .	5
1.3. Convolución con Constante . . . . .	5
1.4. Convolución con Tiles y Constante . . . . .	5
<b>2. Pruebas y Gráficas</b>	<b>6</b>
2.1. Image 1 . . . . .	6
2.2. Image 2 . . . . .	9
2.3. Image 3 . . . . .	12
2.4. Image 4 . . . . .	13
2.5. Image 5 . . . . .	15
2.6. Image 6 . . . . .	18
<b>3. Conclusiones</b>	<b>22</b>

# Índice de figuras

2.1. Img1.jpg Original . . . . .	7
2.2. Img1.jpg Filtered . . . . .	8
2.3. Img2.jpg Original . . . . .	10
2.4. Img2.jpg Filtered . . . . .	11
2.5. Img3.jpg Original . . . . .	12
2.6. Img3.jpg Filtered . . . . .	13
2.7. Img4.jpg Original . . . . .	14
2.8. Img4.jpg Filtered . . . . .	14
2.9. Img5.jpg Original . . . . .	16
2.10. Img5.jpg Filtered . . . . .	17
2.11. Img6.jpg Original . . . . .	19
2.12. Img6.jpg Filtered . . . . .	19
2.13. Gráfico de comparación de tiempos . . . . .	20
2.14. Gráfico de Barras de aceleración . . . . .	20
2.15. Gráfico de Aceleración . . . . .	21

# Índice de cuadros

2.1.	Convolución Img1.jpg. tiempo: microsegundos. . . . .	6
2.2.	Convolución Img2.jpg. tiempo: microsegundos. . . . .	9
2.3.	Convolución Img3.jpg. tiempo: microsegundos. . . . .	12
2.4.	Convolución Img4.jpg. tiempo: microsegundos. . . . .	13
2.5.	Convolución Img5.jpg. tiempo: microsegundos. . . . .	15
2.6.	Convolución Img6.jpg. tiempo: microsegundos. . . . .	18

# Capítulo 1

## Introducción

La convolución es un operador matemático que transforma dos funciones  $f$  y  $g$  en una tercera función que en cierto sentido representa la magnitud en la que se superponen  $f$  y una versión trasladada e invertida de  $g$ . La mayoría de los filtros usan una matriz de convolución. En esta investigación se analizó el filtro de Sobel, cuyo objetivo es detectar los cambios de una imagen (su derivada).

### 1.1. Filtro de Sobel

Esta técnica fue creada por Irwin Sobel y Gary Feldman en Stanford Artificial Intelligence Laboratory (SAIL). Presentan lo que es la matriz gradiente 3x3 en 1968, la cual es técnicamente un operador operador de diferenciación discreto, generando una aproximación de la gradiente de la función de la intensidad de la imagen.

$$G_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} \times I \quad (1.1)$$

Filtro Horizontal,  $I$ : Imagen.

$$G_y = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} \times I \quad (1.2)$$

Filtro Vertical,  $I$ : Imagen.

## **1.2. Convolución Básica**

Se usan los hilos para que accedan a cada posición del vector y multipliquen por los elementos de la máscara. La matriz de entrada y la máscara vienen del host, desde la memoria global.

## **1.3. Convolución con Constante**

Se usa una variable tipo `__constant__` que es como si fuera una variable compartida, utilizada como memoria caché con mucho más rápido acceso que la memoria global, dónde se guarda la máscara (la matriz de filtro). Acelerando de manera notable con respecto a la forma básica. Lo malo de este método es que el “Ratio” Sigue siendo de uno, ya que solo se está realizando una operación por cada hilo.

## **1.4. Convolución con Tiles y Constante**

Se combina la máscara en memoria constante y el uso de memoria compartida para leer datos del vector de entrada. Para aprovechar así el uso del acceso a memoria, subiendo el “Ratio” a 3, desarrollando tres operaciones por hilo. Se utiliza el concepto de “Halo Elements” los cuales cargarán los datos de la izquierda y de la derecha del índice del hilo en cuestión.

# Capítulo 2

## Pruebas y Gráficas

### 2.1. Image 1

Img1.jpg Total Size = 336400				
Test	Sequential	Basic	Constant	Tiled
1	3624	333	224	223
2	3575	328	224	225
3	3184	315	216	214
4	3190	309	214	213
5	3185	309	226	215
6	3184	318	217	215
7	3181	313	216	214
8	3580	329	224	223
9	3187	314	213	213
10	3182	307	212	214
11	3197	315	216	216
12	3183	311	215	213
13	3187	316	217	215
14	3184	317	216	214
15	3190	308	214	210
16	3188	311	218	216
17	3186	308	217	212
18	3187	316	214	215
19	3575	323	222	219
20	3184	311	216	217
Average	3266,65	315,55	217,55	215,8

Cuadro 2.1: Convolución Img1.jpg. tiempo: microsegundos.



Figura 2.1: Img1.jpg Original



Figura 2.2: Img1.jpg Filtered

## 2.2. Image 2

Img2.jpg Total Size = 408320				
Test	Sequential	Basic	Constant	Tiled
1	4760	487	255	254
2	4632	365	246	246
3	4113	362	245	244
4	4120	359	245	244
5	4116	355	241	247
6	4115	356	245	241
7	4118	356	246	246
8	4635	371	251	255
9	4199	359	247	247
10	4137	362	245	247
11	4112	362	246	248
12	4124	365	247	244
13	4117	361	244	242
14	4116	358	246	246
15	4117	359	245	245
16	4124	360	245	244
17	4115	355	255	243
18	4122	363	243	243
19	4623	362	247	246
20	4115	359	244	244
Average	4231,5	366,8	246,4	245,8

Cuadro 2.2: Convolución Img2.jpg. tiempo: microsegundos.



Figura 2.3: Img2.jpg Original



Figura 2.4: Img2.jpg Filtered

## 2.3. Image 3

Img3.jpg Total Size = 1049088				
Test	Sequential	Basic	Constant	Tiled
1	10874	1190	588	583
2	10516	847	573	571
3	9357	842	571	569
4	9366	843	572	551
5	9354	838	570	567
6	9348	850	572	574
7	9346	844	572	572
8	10569	862	588	587
9	9407	866	575	576
10	9356	841	570	571
11	9438	857	576	573
12	9354	842	567	566
13	9354	846	572	571
14	9363	842	568	569
15	9361	846	573	575
16	9352	841	567	566
17	9358	847	575	575
18	9358	844	569	570
19	10536	853	578	577
20	9363	846	571	571
Average	9616,5	864,35	573,35	571,7

Cuadro 2.3: Convolución Img3.jpg. tiempo: microsegundos.



Figura 2.5: Img3.jpg Original

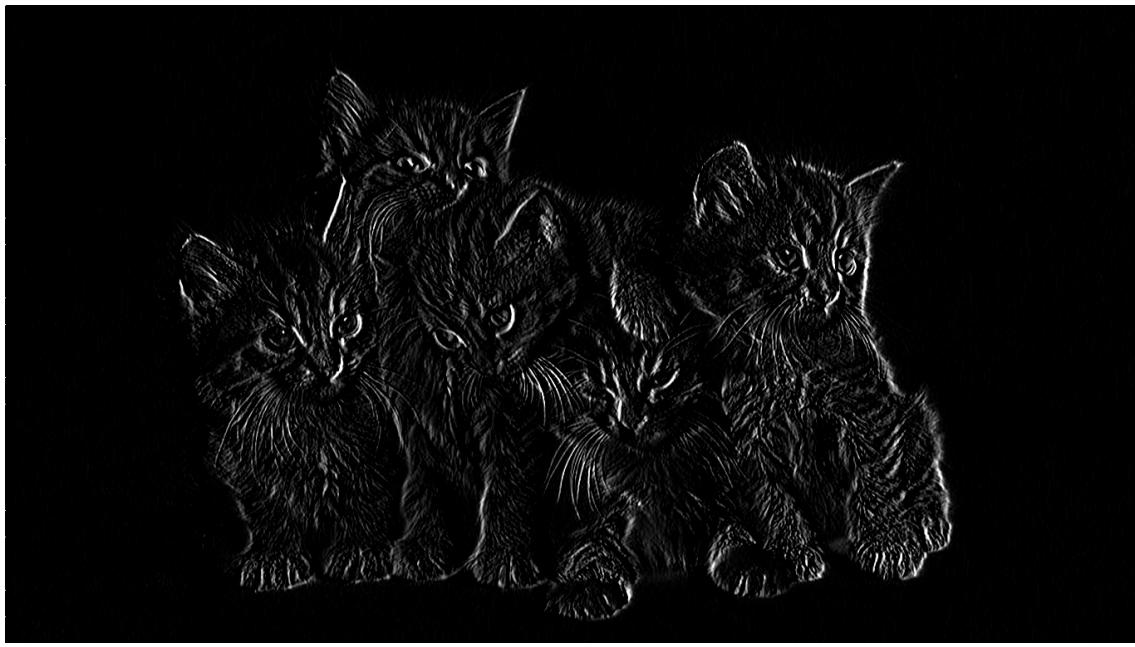


Figura 2.6: Img3.jpg Filtered

## 2.4. Image 4

Img4.jpg Total Size = 4096000				
Test	Sequential	Basic	Constant	Tiled
1	38254	3298	1635	1638
2	35271	2694	1634	1622
3	35285	2749	1643	1648
4	35254	2733	1646	1659
5	37710	2735	1638	1652
6	35508	2728	1641	1638
7	35382	2752	1654	1638
8	39450	2735	1643	1644
9	35398	2760	1687	1676
10	35296	2736	1638	1644
11	35267	2734	1646	1649
12	35345	2732	1629	1631
13	35345	2750	1642	1645
14	35416	2726	1627	1625
15	35314	2747	1645	1653
16	37246	2742	1649	1644
17	35282	2746	1645	1645
18	35389	2721	1636	1630
19	39808	2748	1651	1646
20	35286	2740	1643	1642
Average	9616,5	864,35	573,35	571,7

Cuadro 2.4: Convolución Img4.jpg. tiempo: microsegundos.



Figura 2.7: Img4.jpg Original



Figura 2.8: Img4.jpg Filtered

## 2.5. Image 5

Img5.jpg Total Size = 22064172				
Test	Sequential	Basic	Constant	Tiled
1	202281	15041	8019	8025
2	201912	15552	7991	7990
3	204844	15627	8015	8002
4	202158	15327	8037	7997
5	212209	15292	8004	8005
6	201889	15458	8049	8109
7	202188	15325	8045	8057
8	202807	15652	8040	8050
9	203495	14943	7958	7952
10	218755	16140	8173	8073
11	201913	15594	8020	8033
12	202113	15534	7956	7979
13	203782	16037	8238	8234
14	202507	15155	8008	7956
15	202218	15332	8028	8015
16	202274	15354	8018	8016
17	204213	15398	8035	8035
18	220255	15627	8013	8039
19	223218	16153	8186	8183
20	225040	15440	8165	8176
Average	207003,6	15499,05	8049,9	8046,3

Cuadro 2.5: Convolución Img5.jpg. tiempo: microsegundos.



Figura 2.9: Img5.jpg Original



Figura 2.10: Img5.jpg Filtered

## 2.6. Image 6

Img6.jpg Total Size = 16084992				
Test	Sequential	Basic	Constant	Tiled
1	148669	11228	5885	5969
2	149134	10833	5849	5860
3	149106	10764	5870	5868
4	150440	10732	5819	5804
5	148752	10890	5888	5884
6	148558	9879	5866	5863
7	167099	10960	5873	5868
8	148763	11159	5877	5882
9	149392	10847	5869	5873
10	167907	11135	5988	5991
11	150995	10791	5888	5873
12	148571	9830	5801	5810
13	170552	10861	5862	5869
14	148812	10747	5853	5852
15	148715	10736	5836	5831
16	148412	9887	5876	5887
17	165187	10798	5881	5882
18	151738	11127	5987	6090
19	148791	11157	5874	5877
20	164498	10965	5887	5868
Average	153704,6	10766,3	5876,45	5885,05

Cuadro 2.6: Convolución Img6.jpg. tiempo: microsegundos.



Figura 2.11: Img6.jpg Original

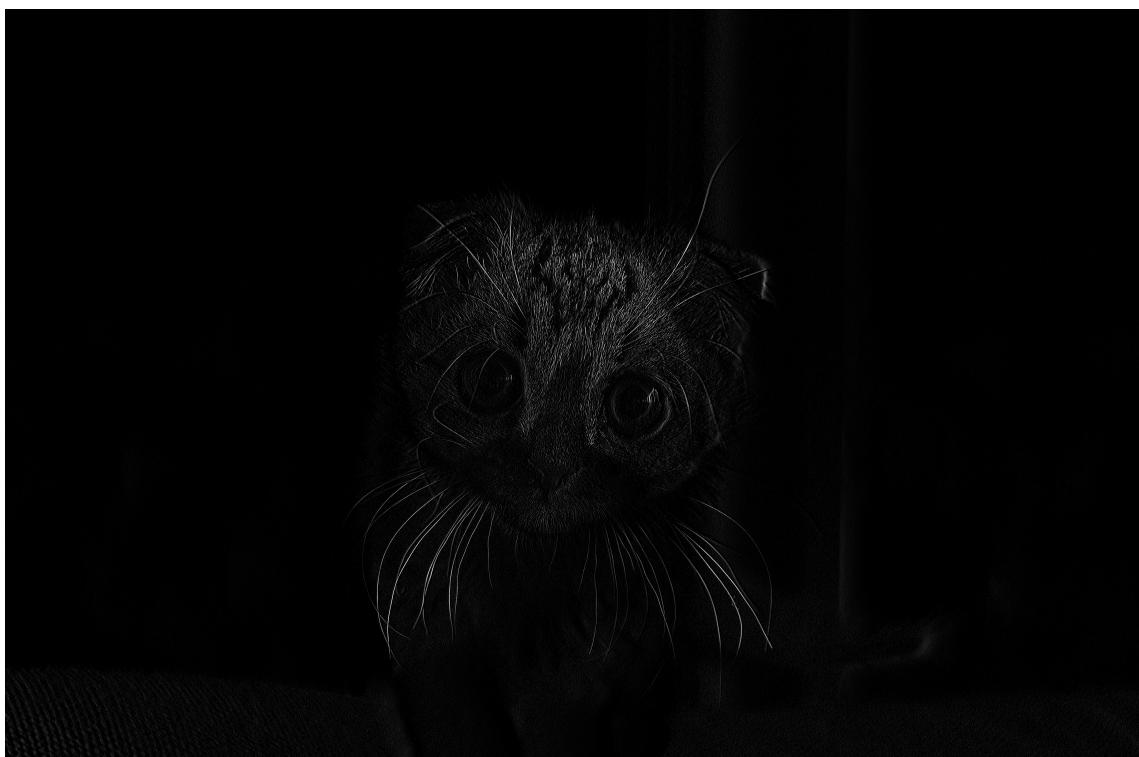


Figura 2.12: Img6.jpg Filtered

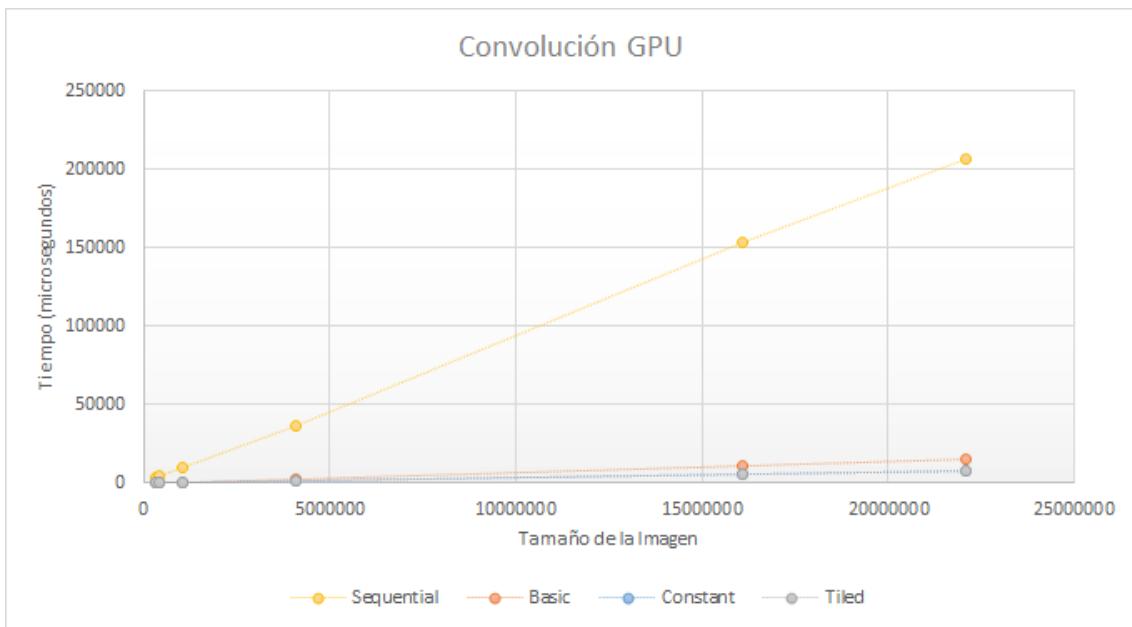


Figura 2.13: Gráfico de comparación de tiempos

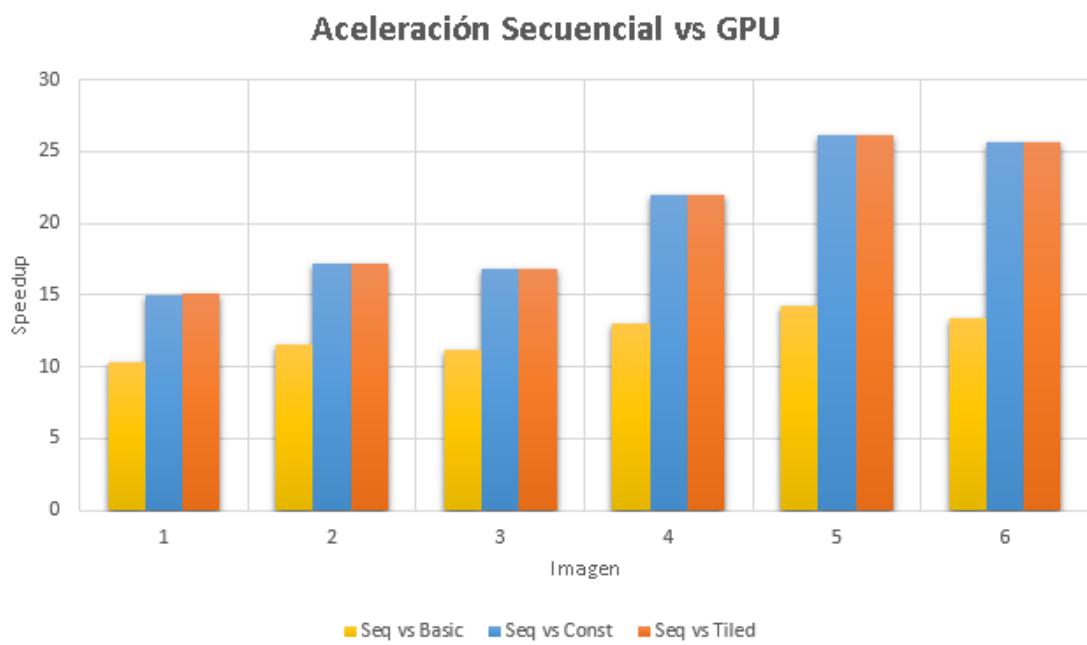


Figura 2.14: Gráfico de Barras de aceleración

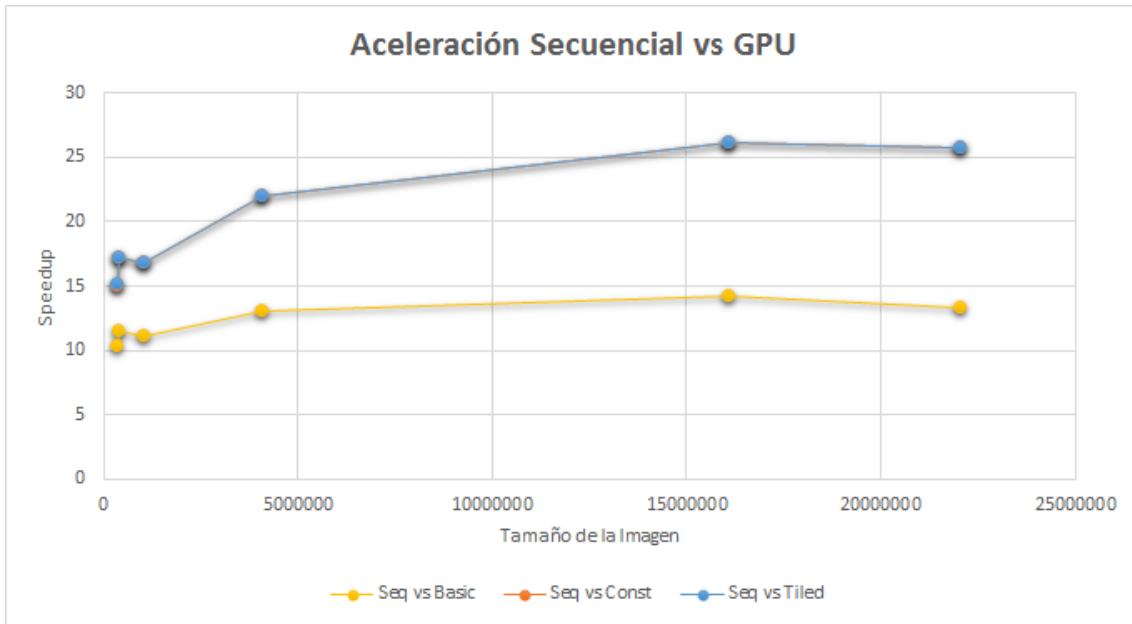


Figura 2.15: Gráfico de Aceleración

# Capítulo 3

## Conclusiones

1. Se pudo observar en la gráfica 2.13 que la convolución desarrollada en la GPU es mucho más veloz que la secuencial, esto debido a la parallelización optimiza mucho estas operaciones a través del uso de múltiples hilos.
2. Se puede ver que en las convoluciones realizadas por GPU, la versión con memoria compartida (Tiles) es la más veloz de las tres versiones, como se observa en las tablas, siempre es la que obtiene el menor tiempo.
3. El uso de memoria global no es una buena forma de implementar paralelización, ya que las implementaciones con memoria caché y memoria compartida son considerablemente mas eficientes.
4. Aunque no hayan imágenes mas pequeñas en las pruebas realizadas, el comportamiento de los tiempos de ejecución hace que el desempeño de los algoritmos paralelos sea menor que el del algoritmo secuencial cuando se manejan pocos datos, en este caso pixeles.

# Bibliografía

- [1] Victor Podlozhnyuk, *Image Convolution with CUDA*, 2007.