INFORME TERCER PARCIAL

SEBASTIAN DUQUE RESTREPO

1112783873

ARQUITECTURA DE COMPUTADORES

UNIVERSIDAD TECNOLÓGICA DE PEREIRA

FACULTAD DE INGENIERÍAS
INGENIERÍA DE SISTEMAS Y COMPUTACIÓN

2015

INFORME

Imágenes de entrada/Imágenes de salida (secuencial y paralelo):

Imagen 1:

Secuencial:

Paralelo:

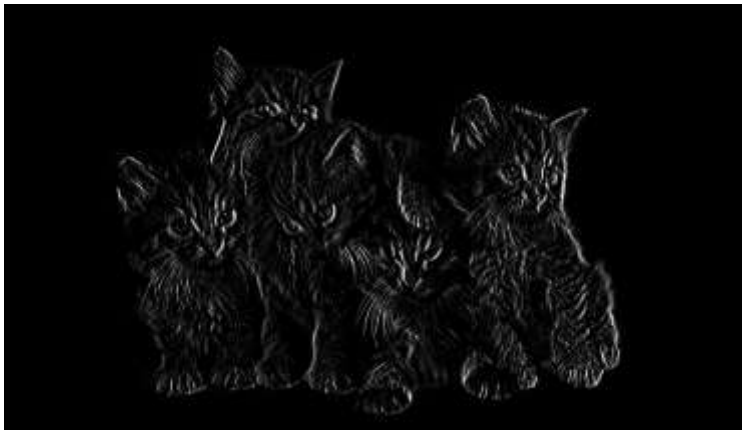
Imagen 2:

Secuencial:

Paralelo:

Imagen 3:

Secuencial:



Paralelo:

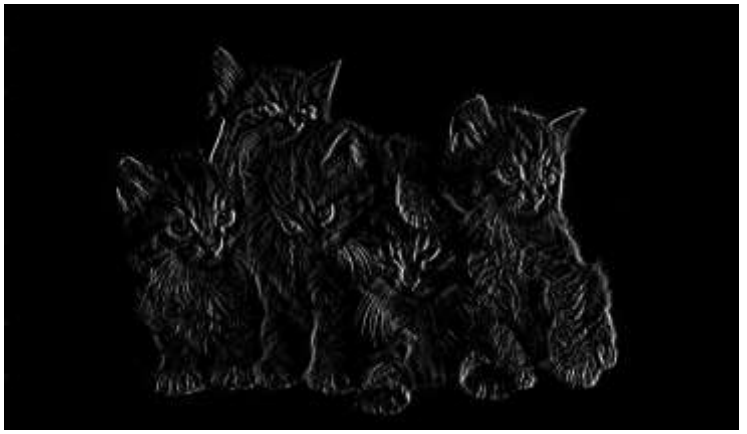


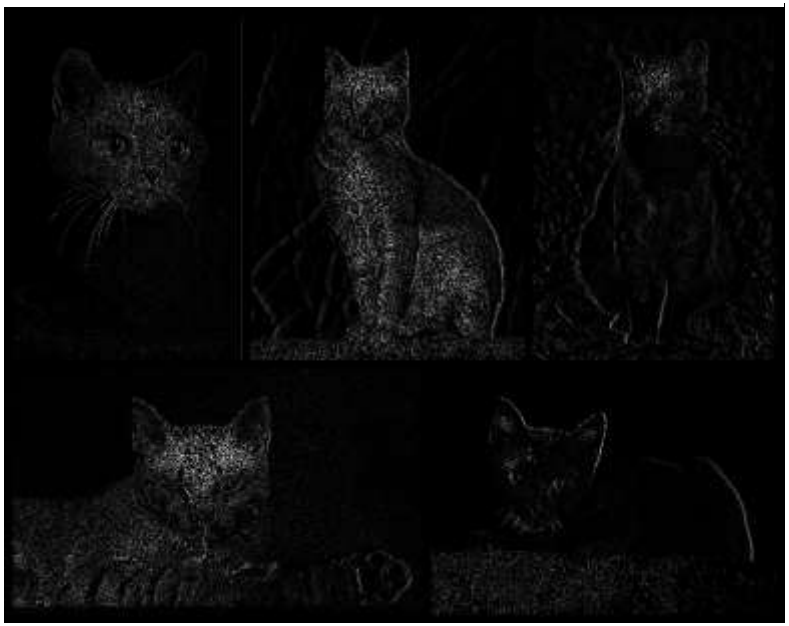
Imagen 4:

Secuencial:

Paralelo:

Imagen 5:

Secuencial:



Paralelo:

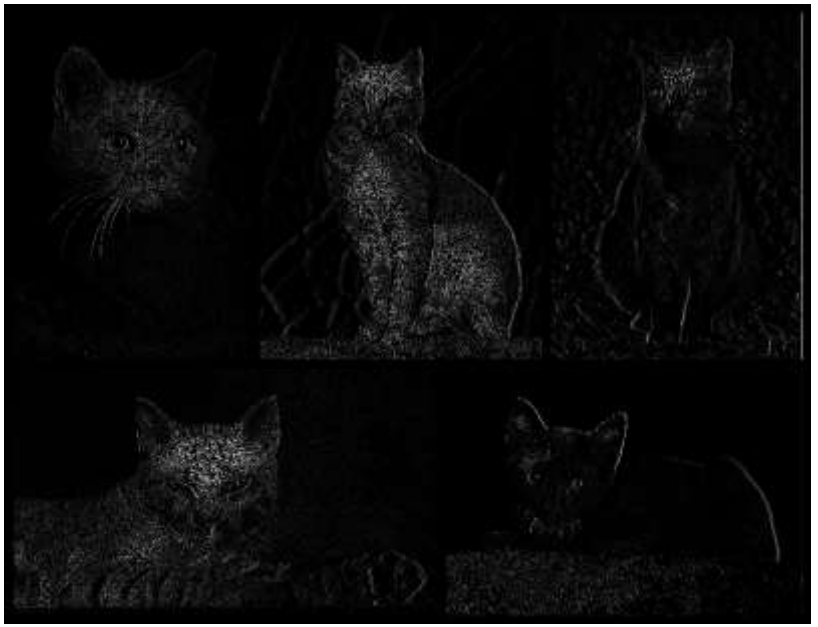


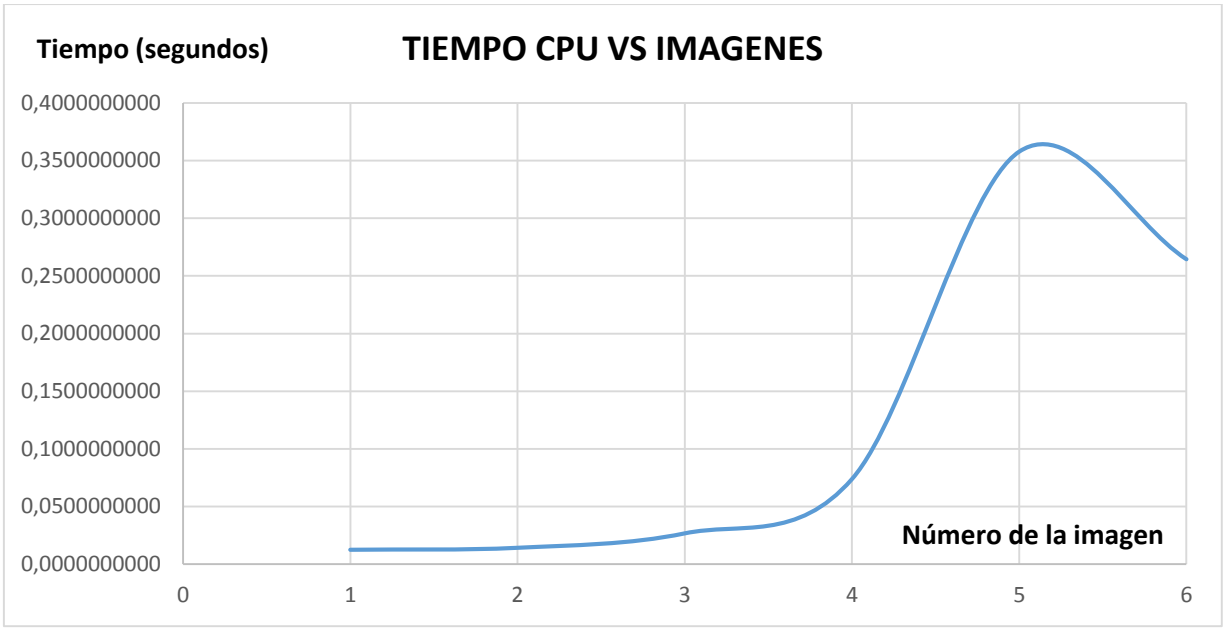
Imagen 6:

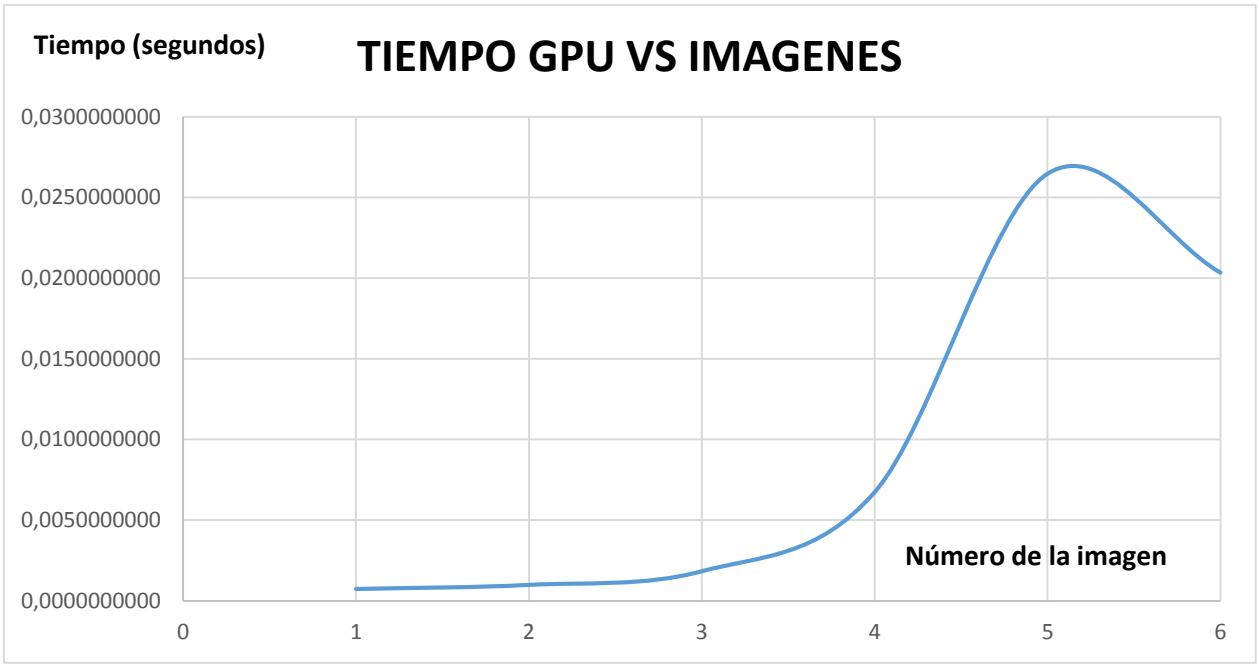
Secuencial:

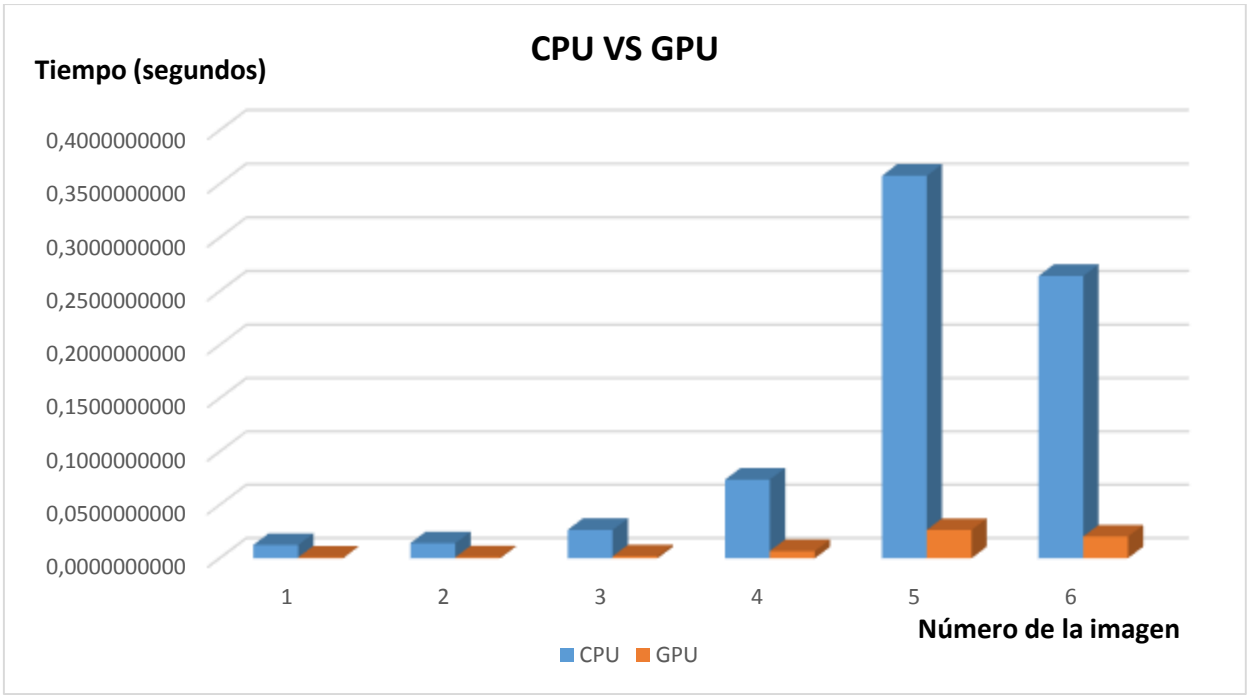
Paralelo:

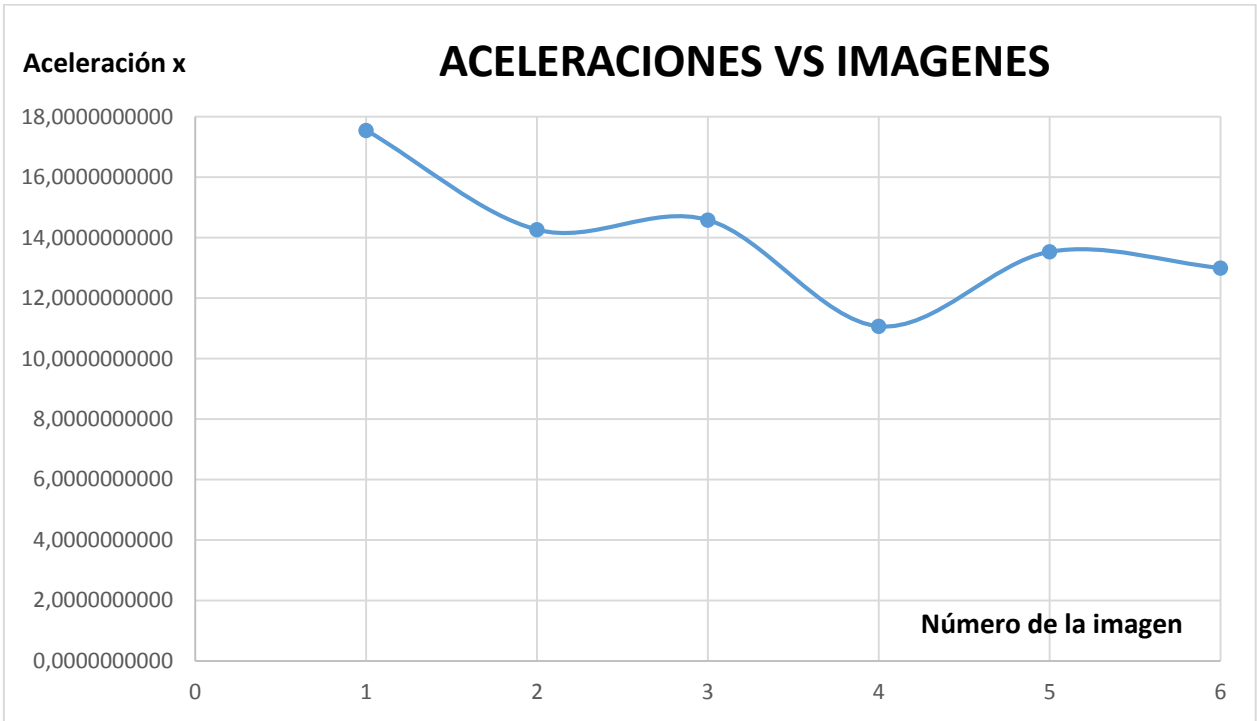
TABLAS DE TIEMPO DE EJECUCIÓN

Nombre de la imagen	CPU (segundos)	GPU (segundos)	Aceleración
img1	0,0129200000	0,0005800000	22,2758620690 X
	0,0115170000	0,0008110000	14,2009864365 X
	0,0116730000	0,0008090000	14,4289245983 X
	0,0135740000	0,0008730000	15,5486827033 X
	0,0123620000	0,0005810000	21,2771084337 X
Promedio	0,0124092000	0,0007308000	17,5463128482 X
img2	0,0115560000	0,0008520000	13,5633802817 X
	0,0111050000	0,0009300000	11,9408602151 X
	0,0130690000	0,0010430000	12,5302013423 X
	0,0186380000	0,0010630000	17,5333960489 X
	0,0162840000	0,0010350000	15,7333333333 X
Promedio	0,0141304000	0,0009846000	14,2602342443 X
img3	0,0216750000	0,0014390000	15,0625434329 X
	0,0272340000	0,0019520000	13,9518442623 X
	0,0265400000	0,0018990000	13,9757767246 X
	0,0276800000	0,0019310000	14,3345416882 X
	0,0300190000	0,0019280000	15,5700207469 X
Promedio	0,0266296000	0,0018298000	14,5789453710 X
img4	0,0699860000	0,0063770000	10,9747530187 X
	0,0719060000	0,0067790000	10,6071691990 X
	0,0699110000	0,0073790000	9,4743190134 X
	0,0811130000	0,0058890000	13,7736457803 X
	0,0758270000	0,0072220000	10,4994461368 X
Promedio	0,0737486000	0,0067292000	11,0658666296 X
img5	0,3555730000	0,0256750000	13,8489970789 X
	0,3401060000	0,0263000000	12,9317870722 X
	0,3859900000	0,0269630000	14,3155435226 X
	0,3597460000	0,0255660000	14,0712665259 X
	0,3471520000	0,0278310000	12,4735726348 X
Promedio	0,3577134000	0,0264670000	13,5282333669 X
img6	0,2540100000	0,0203260000	12,4968021254 X
	0,2578570000	0,0200760000	12,8440426380 X
	0,2866870000	0,0204020000	14,0519066758 X
	0,2652700000	0,0205170000	12,9292781596 X
	0,2580210000	0,0203870000	12,6561534311 X
Promedio	0,2643690000	0,0203416000	12,9956366060 X


Tiempo (segundos)
TIEMPO CPU VS IMAGENES
0,4000000000
0,3500000000
0,3000000000
0,2500000000
0,2000000000
0,1500000000
0,1000000000
0,0500000000
0,0000000000
0
1
2
3
4
5
6
Número de la imagen


Tiempo (segundos)
TIEMPO GPU VS IMAGENES
0,0300000000
0,0250000000
0,0200000000
0,0150000000
0,0100000000
0,0050000000
0,0000000000
0
1
2
3
4
5
6
Número de la imagen


CPU VS GPU
Tiempo (segundos)
0,4000000000
0,3500000000
0,3000000000
0,2500000000
0,2000000000
0,1500000000
0,1000000000
0,0500000000
0,0000000000
1
2
3
4
5
6
Número de la imagen
CPU
GPU


Aceleración x
ACELERACIONES VS IMAGENES
18,0000000000
16,0000000000
14,0000000000
12,0000000000
10,0000000000
8,0000000000
6,0000000000
4,0000000000
2,0000000000
0,0000000000
0
1
2
3
4
5
6
Número de la imagen

CONCLUSIONES

- En el código elaborado para este trabajo se pudo apreciar que la parte secuencial fue un poco más sencilla debido a que se utilizo la librería OpenCV, la cual ya tenia definida funciones como la de pasar una imagen a escala de grises y otra como la de aplicar un filtro de Sobel, mientras que para CUDA se nos hizo necesario implementar funciones que realizaran lo anterior, pero obviamente que se realicen de manera paralela o sea utilizando los hilos y bloque de la GPU.

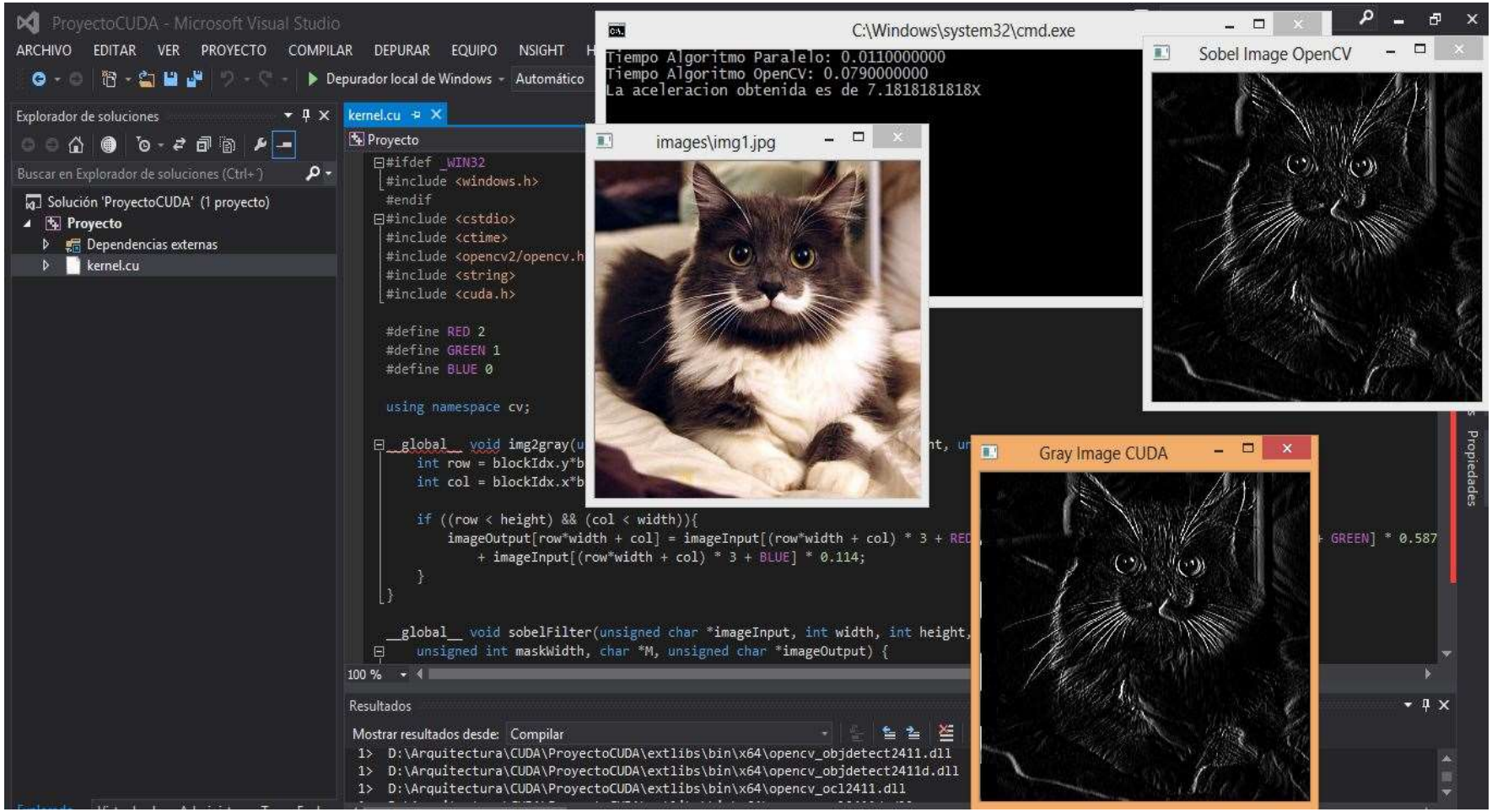
- Aunque en la parte secuencial se utilizo una librería estándar, fue mucho más rapido con CUDA.

- Se pudo observar que para las últimas dos imágenes se tardaron mucho más que en las imágenes anteriores tanto de manera paralela como secuencial.

- Al comparar la imágenes obtenidas para la realización de este informe, se pudo identificar que la imagen que es generada con la librería OpenCV (secuencial) tiene significativamente mejor calidad en los bordes externos, ya que la imagen que fue generada con CUDA (paralela) tenía una pequeñas interferencias en algunos bordes, pero es muy poco y en general las imágenes eran muy similares.

- Se pudo observar el gran poder que tiene hacer un algoritmo de forma paralela, ya que esto fue muy notorio en las imágenes 5 y 6 en donde se vio un cambio bastante significativo en los tiempos (secuencial VS paralelo) y en general se pudieron ver que hubieron unas aceleraciones bastante interesantes.

- Una observación importante es que aprovechando que mi computador posee una tarjeta NVIDIA obviamente no tan potente como la que esta conectada al servidor, se hicieron pruebas del proyecto, y aún asi se pudo apreciar que las aceleraciones en la GPU fueron significativas, en general realizarlo de manera paralela siempre se obtiene un buena aceleración, a continuación se puede apreciar una imagen de una de las pruebas que realice en mi computador:


ProyectoCUDA - Microsoft Visual Studio
ARCHIVO EDITAR VER PROYECTO COMPILAR DEPURAR EQUIPO NSIGHT
Depurador local de Windows
Automático
Explorador de soluciones
Buscar en Explorador de soluciones (Ctrl+')
Solución 'ProyectoCUDA' (1 proyecto)
Proyecto
Dependencias externas
kernel.cu
kernel.cu
Proyecto
#ifdef _WIN32
#include <windows.h>
#endif
#include <cstdio>
#include <ctime>
#include <opencv2/opencv.h
#include <string>
#include <cuda.h>
#define RED 2
#define GREEN 1
#define BLUE 0
using namespace cv;
__global__ void img2gray(u
int row = blockIdx.y*b
int col = blockIdx.x*b
if ((row < height) && (col < width)){
imageOutput[row*width + col] = imageInput[(row*width + col) * 3 + RED
+ imageInput[(row*width + col) * 3 + BLUE] * 0.114;
}
}
__global__ void sobelFilter(unsigned char *imageInput, int width, int height,
unsigned int maskWidth, char *M, unsigned char *imageOutput) {
100 %
Resultados
Mostrar resultados desde: Compilar
1> D:\Arquitectura\CUDA\ProyectoCUDA\extlibs\bin\x64\opencv_objdetect2411.dll
1> D:\Arquitectura\CUDA\ProyectoCUDA\extlibs\bin\x64\opencv_objdetect2411d.dll
1> D:\Arquitectura\CUDA\ProyectoCUDA\extlibs\bin\x64\opencv_ocl2411.dll
C:\Windows\system32\cmd.exe
Tiempo Algoritmo Paralelo: 0.0110000000
Tiempo Algoritmo OpenCV: 0.0790000000
La aceleracion obtenida es de 7.1818181818X
images\img1.jpg
Sobel Image OpenCV
Gray Image CUDA
Propiedades