Benchmark entre la implementación por software y la de shaders de máscara de convolución blur-effect

Presentado por: Juan Sebastián Herrera Maldonado
Presentado al Profesor: Jean Pierre Charalambos Hernandez

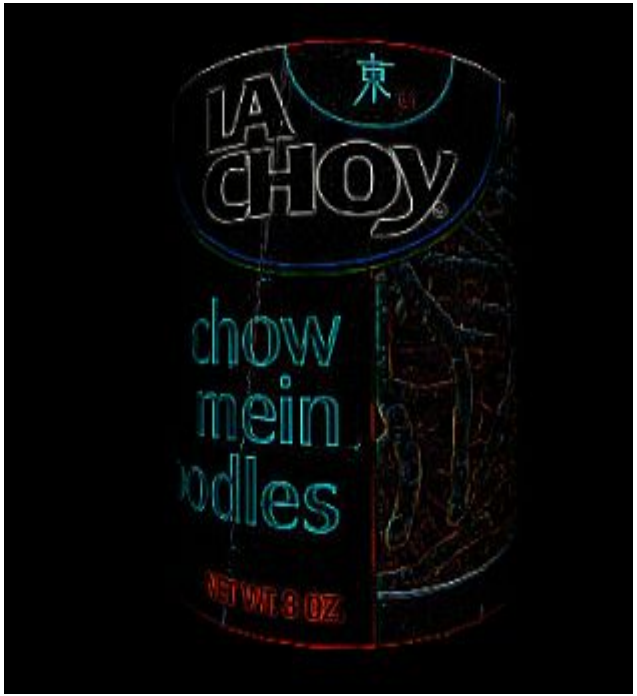
Filtrado de imágenes

Conjunto de técnicas englobadas dentro del procesamiento de imágenes cuyo objetivo fundamental es obtener, a partir de una imagen origen, otra final cuyo resultado sea más adecuado para una aplicación específica mejorando ciertas características de la misma.

Se consideran los filtros como operaciones que se aplican a los píxeles de una imagen digital para optimizarla, enfatizar cierta información o conseguir un efecto especial en ella.

Objetivos Filtrado de imágenes

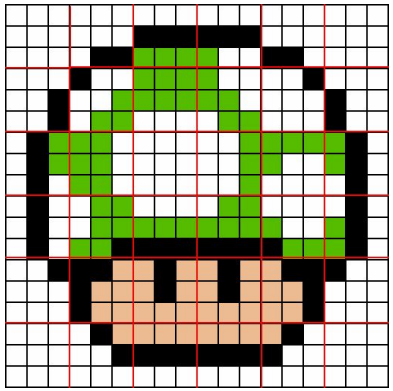
- Suavizar la imagen
- Eliminar ruido
- Realzar bordes
- Detectar bordes



Edge detection filter

Filtrado de imágenes

Para el desarrollo del del Benchmark se tendrá en cuenta solamente él el filtrado lineal (filtros basados en núcleos o máscaras de convolución) en el dominio del espacio, en este tipo de filtrado Las operaciones de filtrado se llevan a cabo directamente sobre los píxeles de la imagen.



35	40	41	45	50
40	40	42	46	52
42	46	50	55	55
48	52	56	58	60
56	60	65	70	75

×

	0	1	0
	0	0	0
	0	0	0

=

		42		

Blur effect

Dada una imagen de n × m píxeles, para cada pıxel se hallará el promedio entre sus vecinos cercanos definido por un kernel impar k, dada la condición $3 \leq k \leq 15$.

$$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

K = 3

Blur effect

K = 3

	-1	0	1
-1	1	2	3
0	4	5	6
1	7	8	9

K = 5

	-2	-1	0	1	2
-2	1	2	3	4	5
-1	6	7	8	9	10
0	11	12	13	14	15
1	16	17	18	19	20
2	21	22	23	24	25

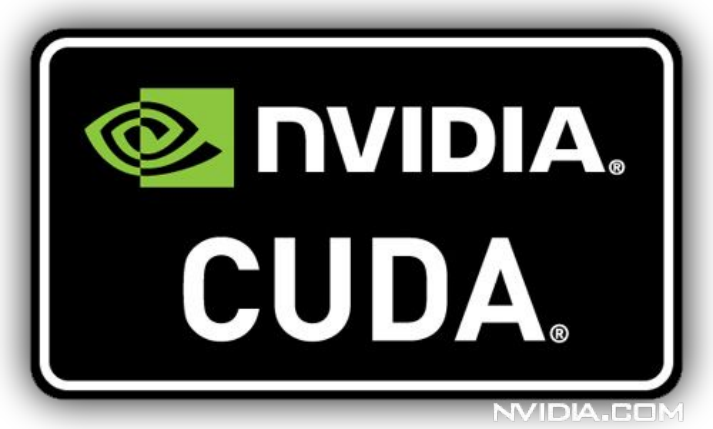
K = 7

	-3	-2	-1	0	1	2	3
-3	1	2	3	4	5	6	7
-2	8	9	10	11	12	13	14
-1	15	16	17	18	19	20	21
0	22	23	24	25	26	27	28
1	29	30	31	32	33	34	35
2	36	37	38	39	40	41	42
3	43	44	45	46	47	48	49

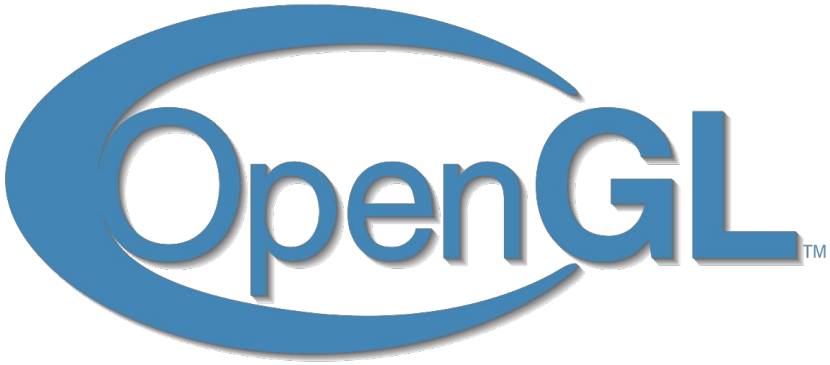
Herramientas implementación por software

OpenMP

C POSIX library

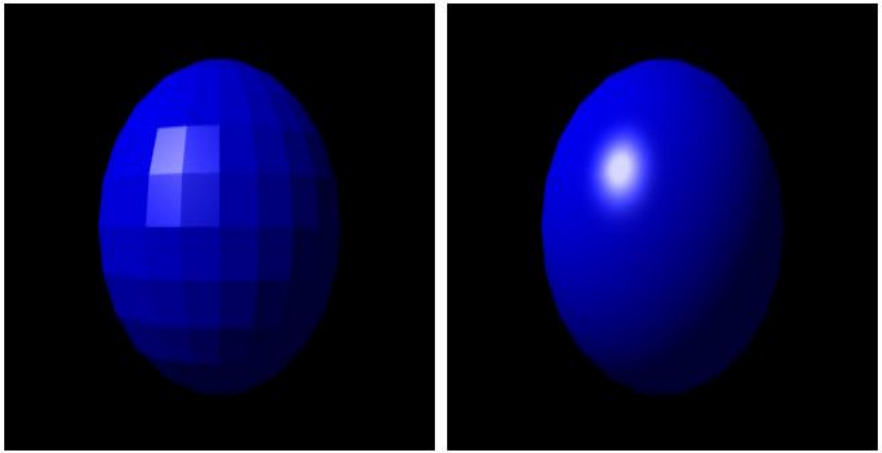


Herramientas implementación por Shaders



Shader

Los shaders son pequeños programas que se basan en la GPU. Es un tipo de programa de computadora que originalmente se usa para sombrear (la producción de niveles apropiados de luz , oscuridad y color dentro de una imagen) pero que ahora realiza una variedad de funciones especializadas en varios campos de gráficos de computadora. Este procedimiento consta de tres etapas:



FLAT SHADING PHONG SHADING

- Una primera etapa en la que se procesan los vértices de los modelos (transformaciones e iluminación). **(vertex shader)**
- Una segunda etapa en la que se analiza cada primitiva para detectar qué porción de la misma queda dentro del área visual o viewport y se procesa cada triángulo para obtener los píxeles que lo forman (rasterization). **(geometry shader)**
- Por último, una etapa en la que se analiza cada píxel y se determina su color y si dicho píxel debe ser visible o no dependiendo de su profundidad. **(pixel/fragment shader)**

Implementación

Aplicación del algoritmo blur effect a una imagen de 720 pixeles con kernel 15

Implementación

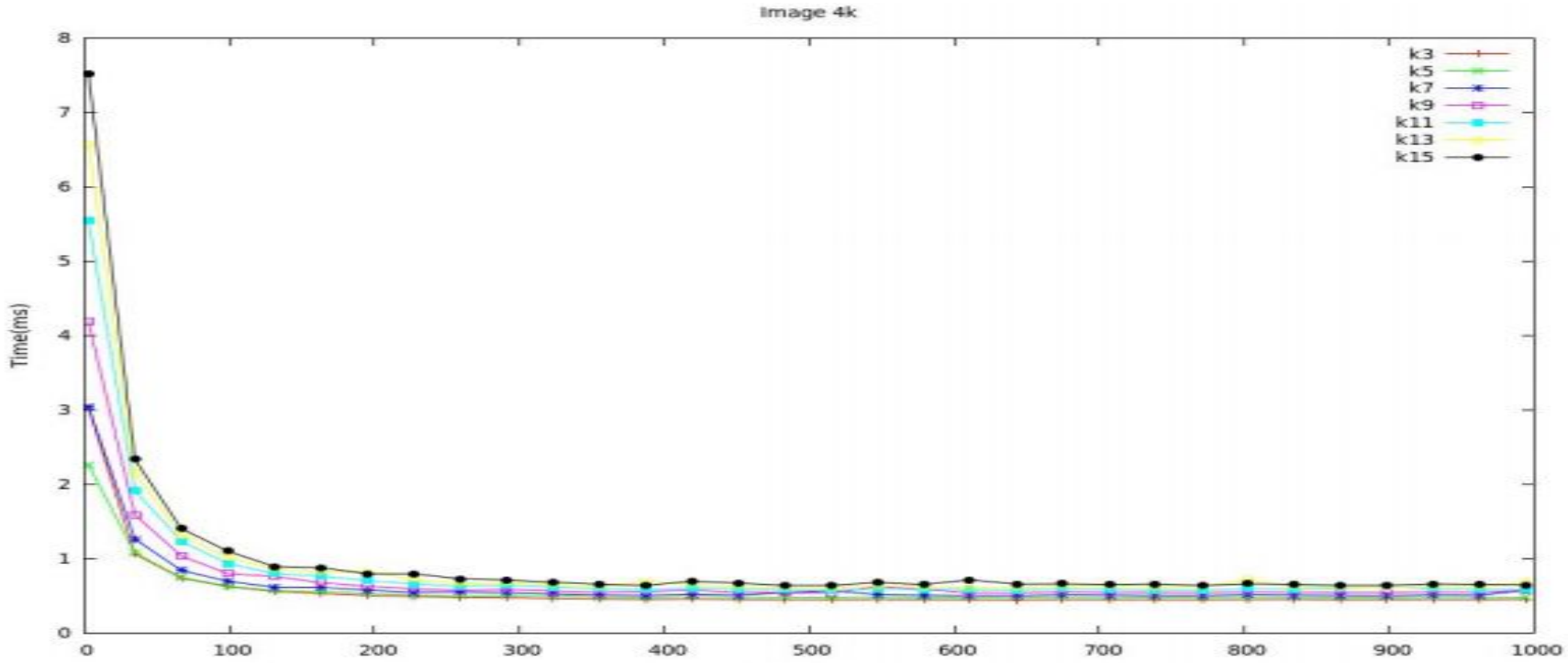
Aplicación del algoritmo blur effect a una imagen de 1080 pixeles con kernel 15

Implementación

Aplicación del algoritmo blur effect a una imagen de 4k pixeles con kernel 15

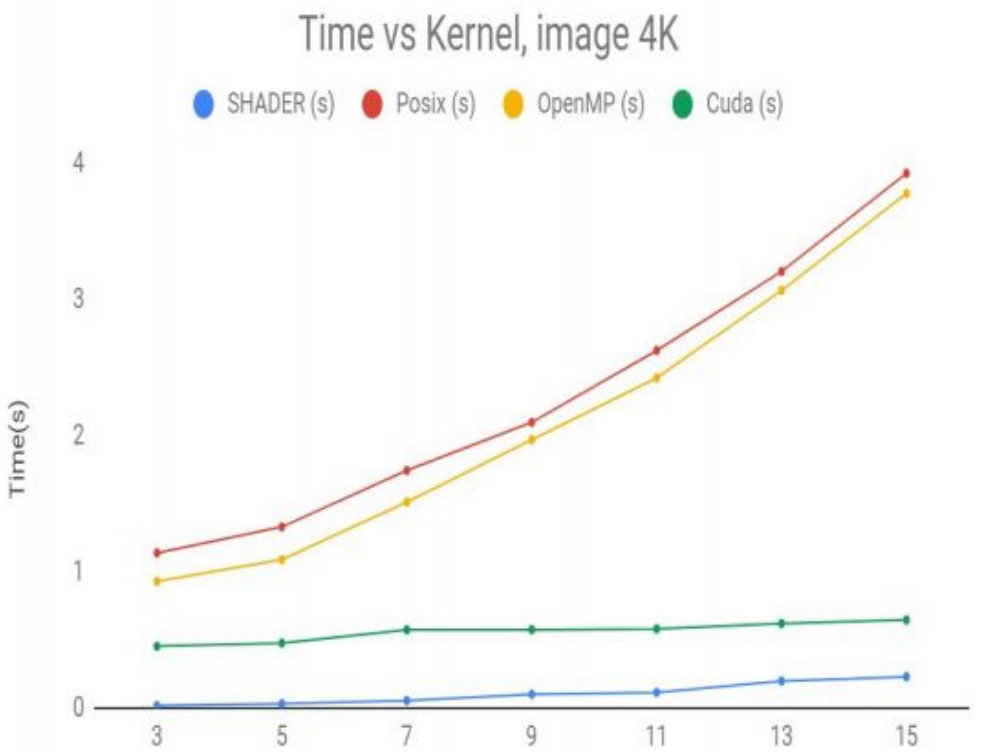
Implementación

Gráfica de la implementación por software de imagen 4k variando los hilos y el kernel , utilizando CUDA.



Implementación

Para esta gráfica se tomaron los mejores tiempos de cada herramienta de la implementación por software usando la imagen en 4K y se compararon con los tiempos de la implementación por shaders usando la imagen en 4K.



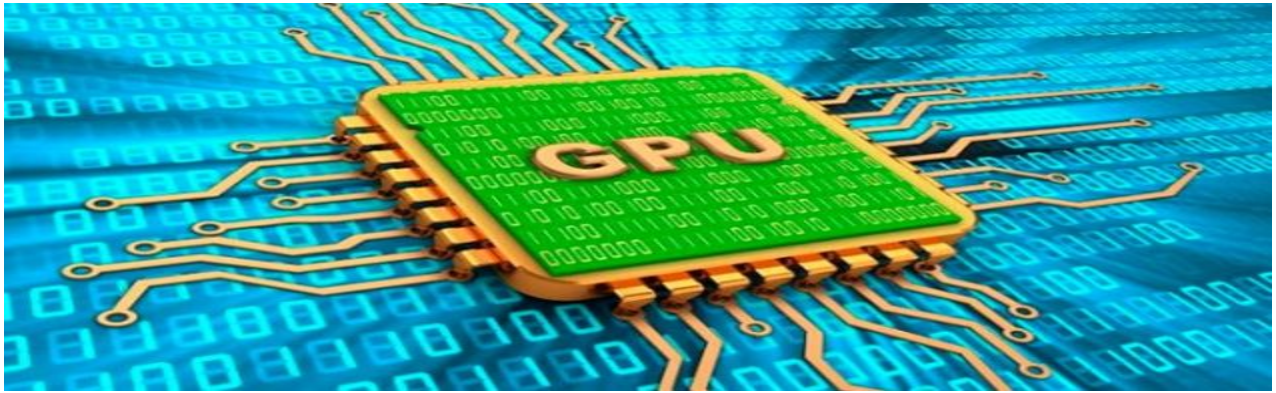
Análisis y conclusiones

Siendo así y analizando la gráfica 13 se puede ver claramente como la implementación por shaders mejora notablemente los tiempos de ejecución comparado con las herramientas (implementación por software) de OpenMP y Posix

La implementación por software utilizando CUDA y la implementación por shaders tienen en común que ambas buscan optimizar la ejecución del algoritmo blur-effect por medio del uso de la GPU (unidad de procesamiento gráfico). Es por esta razón que estas dos herramientas tienen los mejores tiempos de ejecución.

Análisis y conclusiones

El desarrollo del hardware gráfico ha ido haciendo cada vez más programables las tarjetas gráficas, de modo que los programadores puedan utilizar mejor sus funcionalidades. Con la introducción de los lenguajes de shaders se han podido implementar numerosos algoritmos (iluminación, técnicas de mapeo de texturas, efectos atmosféricos, entre otros) para que sean procesados al nivel de la tarjeta gráfica.




GRACIAS!