

ELEC 5593 High Performance Computer Architecture (HPCA)

HW: GPU (CUDA) Image Filters

- Rutvij Girish Karkhanis

Different types of image filters such as sobel 3x3 and 5x5 matrix filter, average filter, high boost filter were implemented by using cuda on GPU. CPU and GPU execution time were calculated and compared for analysis.

1. Sobel Filter (3x3 matrix).

Sobel filter is used for edge detection in the image. The output image from the sobel filter shows the sudden changes around the edges in the image. The sobel matrix of size 3x3 is used for finding out the edges in the image. The changes were calculated in both horizontal as well as vertical direction and their absolute values are calculated for the resultant change. The sobel filter is used on the lena.bmp and dublin.bmp image. The findings are as below:

Input Size	GPU Execution Time (ms)	CPU Execution Time (ms)	Speedup (%)
lena.bmp	0.068000	2.641000	97.42521
dublin.bmp	0.640000	63.541000	98.99277

The resulting images namely lena_sobel_out.bmp and dublin_sobel_out.bmp have been added to the .tar file.

2. Average Filter.

Similar to the Sobel filter, except each pixel generates the sum of eight neighbors and the center pixel and divides by 9. In the case of pixels as characters, it is best to cast the values as float, add a sum, and then cast the resulting float back to an unsigned character.

The CPU and GPU (gpu+mem) time are calculated for the average filter execution. The values are tabulated as below:

Input Size	GPU Execution Time (ms)	CPU Execution Time (ms)	Speedup (%)
lena.bmp	0.065000	1.824000	96.43000
dublin.bmp	0.711000	42.217999	98.31588

3. Boost Filter.

Similar to the average filter, the Boost filter sums the 9 values, and then subtracts the difference between the center pixel and the average. The difference is then

multiplied by a boost factor and added to the original value, finally the result is guaranteed to be between 0 and 255.

The CPU and GPU (gpu+mem) time are calculated for the high boost filter execution.

The values are tabulated as below:

Input Size	GPU Execution Time (ms)	CPU Execution Time (ms)	Speedup (%)
lena.bmp	0.067000	2.371000	97.1740
dublin.bmp	0.737000	66.551003	98.8925

4. Sobel Filter with 5x5 array

$$\begin{bmatrix} 1 & 2 & 0 & -2 & -1 \\ 4 & 8 & 0 & -8 & -4 \\ 6 & 12 & 0 & -12 & -6 \\ 4 & 8 & 0 & -8 & -4 \\ 1 & 2 & 0 & -2 & -1 \end{bmatrix} \begin{bmatrix} -1 & -4 & -6 & -4 & -1 \\ -2 & -8 & -12 & -8 & -2 \\ 0 & 0 & 0 & 0 & 0 \\ 2 & 8 & 12 & 8 & 2 \\ 1 & 4 & 6 & 4 & 1 \end{bmatrix}$$

The new sobel filter used the 5x5 array matrix.

Input Size	3x3 GPU Execution Time (GPU + memory) (ms)	5x5 GPU Execution Time (GPU + memory) (ms)	Speedup (%)
lena.bmp	0.068000	0.128000	-88.235294
dublin.bmp	0.640000	2.195000	-242.96875

From the table it is seen that the sobel filter using 5x5 matrix takes more time than sobel filter with 3x3. This is due to the more computation required for the larger sobel matrix. The output files for the filter have been added in the attachment.