

# MC970/MO644 - Programação Paralela

## Laboratório 10

Professor: Guido Araújo  
Monitor: Rafael Cardoso Fernandes Sousa

# Histograma GPU

Neste laboratório, iremos paralelizar Histograma de Imagens - usando CUDA.

## Enunciado

Para paralelizar o Histograma, deve-se utilizar CUDA C. O histograma deve ser feito nos 3 canais: Red, Green e Blue (RGB).

A computação que deve ser movida para a GPU é a seguinte:

```
for (j = 0; j <= 3; j++) {  
    for (k = 0; k <= 3; k++) {  
        for (l = 0; l <= 3; l++) {  
            for (i = 0; i < n; i++) {  
                if (image->data[i].red == j  
                    && image->data[i].green == k  
                    && image->data[i].blue == l) {  
                    count++;  
                }  
            }  
            h[x] = count / n; //Histograma normalizado  
            count = 0;  
            x++;  
        }  
    }  
}
```

Use o grupo para tirar dúvidas =).

## Testes e Resultado

Para compilar o seu programa, basta entrar no servidor mo644, a partir do serviço ssh do IC, e digitar o comando **/usr/local/cuda-7.5/bin/nvcc histogram.cu -o histogram**. Para executá-lo, ainda no servidor mo644, basta digitar **./histogram arq\$.in**.

Os inputs consistem em 3 imagens com as seguintes resoluções: 720p, 1080p e 4k, todas no formato PPM.

Não haverá comparação de Speedup a partir do ParSuSy, peço que você faça isso e inclua as informações no arquivo fonte em formato de comentário. Monte uma tabela contendo as seguintes colunas: entrada, tempo\_serial, tempo\_GPU\_criar\_buffer, tempo\_GPU\_offload\_enviar, tempo\_kernel, tempo\_GPU\_offload\_receber, GPU\_total, speedup (tempo\_serial / GPU\_total). Exemplo:

arq1.ppm, 100, 5, 10, 25, 10, 50, 2.

arq2.ppm, 600, 30, 50, 20, 50, 150, 4

Dêem uma olhada em : <https://devblogs.nvidia.com/parallelforall/how-implement-performance-metrics-cuda-cc/> para identificar como capturar o tempo da execução do Kernel, uma vez que sua execução é não bloqueante.

OBS: GPU\_total = tempo\_GPU\_criar\_buffer + tempo\_GPU\_offload\_enviar + tempo\_kernel + tempo\_GPU\_offload\_receber

PLEASE, COMPARTILHEM ESTAS INFORMAÇÕES NO GRUPO!!

## Submissões

O número máximo de submissões é de 10.

## Compilação e Execução

O ParSuSy irá compilar o seu programa através do compilador nvcc.

## Links Úteis

[http://developer.download.nvidia.com/compute/cuda/1.1-Beta/x86\\_website/projects/histogram64/doc/histogram.pdf](http://developer.download.nvidia.com/compute/cuda/1.1-Beta/x86_website/projects/histogram64/doc/histogram.pdf)  
[https://www.vivaolinux.com.br/dica/Utilizando-o-comando-scp.](https://www.vivaolinux.com.br/dica/Utilizando-o-comando-scp)