
GB-100 - Processamento Paralelo

ATIVIDADE PRÁTICA

Prof. Pedro Carlos da Silva Lara
10/11/2022

Nome: _____

1 Filtro de Sobel

O filtro Sobel é uma transformação usada em processamento de imagem, aplicada normalmente para detecção de contornos.

Inicialmente o filtro transforma a imagem RGB em escala de cinza aplicando a seguinte equação

$$a = 0.3 \cdot p_r + 0.59 \cdot p_g + 0.11 \cdot p_b$$

O filtro atua aplicando duas matrizes 3×3 (\mathbf{G}_x e \mathbf{G}_y) de forma convolucional na matriz da imagem de entrada \mathbf{A} .

$$\mathbf{G}_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} * \mathbf{A} \quad \text{e} \quad \mathbf{G}_y = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} * \mathbf{A}$$

O resultado do filtro \mathbf{S} , é dado por:

$$\mathbf{S} = \sqrt{\mathbf{G}_x^2 + \mathbf{G}_y^2}$$

O algoritmo pode ser visto abaixo. Neste caso, por simplicidade, estamos desconsiderando os pixels da borda.

```
/**
 * A: BW image input  matrix [0,255]
 * S: Sobel Filtered output matrix
 */
for( i = 1; i < height - 1; i++ ) {
    for( j = 1; j < width - 1; j++ ) {
        dy = 0; dx = 0;
        for( x = 0; x < 3; x++ ) {
            for( y = 0; y < 3; y++ ) {
                dx += A[i-1+x][j-1+y] * Gx[x][y];
                dy += A[i-1+x][j-1+y] * Gy[x][y];
            }
        }
        S[i][j] = sqrt( dx*dx + dy*dy );
    }
}
```

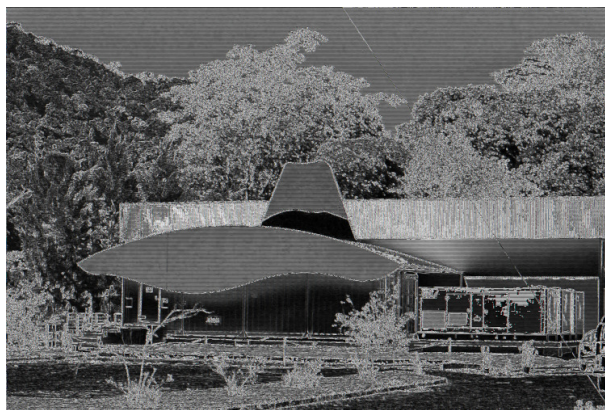


Figura 1: A esquerda a imagem original e a direita a imagem após o filtro de Sobel.

2 Atividade

A atividade proposta consiste em:

- Converter o código `sobel.c` para cuda, declarando os kernels e refatorando os índices.
- Otimizar o kernel (da aplicação do filtro) com o uso da memória shared.
- Coletar e comparar os tempos (usando cuda events).
- (Não precisa entregar hoje) Otimizar usando *streams* visando processamento de imagens grandes.

Deverá ser entregue uma versão do código que exibe o tempo para a versão sequencial, versão cuda e versão otimizada. Além disso, deve-se preencher a tabela abaixo com os resultados.

Os arquivos necessários para a realização desta tarefa podem ser baixados em <https://github.com/pcslara/gb100>.

Dispositivo (Xeon, Tesla K80, ...)	Tam. da Imagem	Tempo em <i>ms</i>	Versão (memória shared, seq., ...)