

Atividade de Laboratório 6b

Objetivos

O objetivo desta atividade é exercitar o uso de instruções para manipulação de vetores e matrizes utilizando o conjunto de instruções da arquitetura RISC-V.

Descrição

Neste laboratório, você deve fazer um programa em linguagem de montagem do RISC-V que leia uma imagem em formato [pgm](#), aplique um filtro de detecção de bordas e exiba o resultado na tela.

Aplicando um filtro a uma imagem

O primeiro passo desse laboratório é ler a imagem em formato pgm e armazenar seu conteúdo em uma matriz (assim como você fez no laboratório 6a). A seguir, você deve aplicar o seguinte filtro na imagem:

$$w = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

Seja w a matriz de filtro acima, M_{in} a matriz representando a imagem de entrada e M_{out} a matriz representando a imagem de saída. A ideia básica para a aplicação desse filtro é que cada *pixel* $[i,j]$ de M_{out} seja definido como:

$$M_{out}[i][j] = \sum_{k=0}^2 \sum_{q=0}^2 w[k][q] * M_{in}[i+k-1][j+q-1]$$

Note que isso pode levar a matriz M_{in} a ser indexada com índices inválidos (negativos ou fora de sua dimensão). Para evitar esses casos, os *pixels* da borda da imagem M_{out} devem ser inicializados com a cor preta e não é necessário calcular o valor do filtro para eles.

Você pode visualizar melhor como este filtro funciona em (selecione o filtro “Outline”): [Image Kernels explained visually](#). Este filtro também é conhecido como operador Laplaciano para detecção de bordas.

Note também que os *pixels* da imagem devem ter valores entre 0 (preto) e 255 (branco). Caso o resultado da fórmula não esteja neste intervalo, você deve utilizar o valor mais próximo no

intervalo (isto é, valores menores que 0 se tornam 0, enquanto valores maiores que 255 se tornam 255).

Você pode obter exemplos de imagens pgm neste endereço:

[PGMB Files - Binary Portable Gray Map Graphics Files](#)

Entrada

Seu programa deve ler um arquivo chamado “imagem.pgm”, que estará no mesmo diretório do executável, conforme explicado no Lab 06a.

Saída

Seu programa deve exibir a imagem na tela, utilizando o periférico canvas, conforme explicado no Lab 06a.