

# Criação e Gerenciamento de *Threads*

## Tarefa da Unidade I

**Disciplina:** DCA0108 - Sistemas Operacionais  
**Turma:** 01, 2023.2  
**Horário:** 24M12  
**Local:** Setor 4, sala C2  
**Professor:** Diogo Pinheiro Fernandes Pedrosa  
diogo.pedrosa@ufrn.br

## Contextualização

A área de processamento e análise de imagens digitais teve um desenvolvimento significativo nos anos iniciais do século XXI. Este crescimento foi derivado do interesse de pesquisa e desenvolvimento de tecnologias pertinentes a áreas como medicina, biologia, automação industrial, sensoriamento remoto, entre outros.

O processamento digital de imagens consiste em um conjunto de técnicas para capturar, representar e transformar imagens com o auxílio do computador. Com isso, pode-se extrair e identificar informações das imagens e melhorar a qualidade visual de alguns elementos estruturais, o que facilita a percepção humana e, também, a interpretação automática por meio de programas computacionais.

Existem várias etapas, que são aplicadas a uma imagem, para se ter um processamento completo. Dentre elas, destaca-se a *segmentação*, que realiza a extração e identificação de áreas de interesse contidas na imagem como, por exemplo, detecção de descontinuidades (bordas) ou similaridades (regiões) na imagem processada.

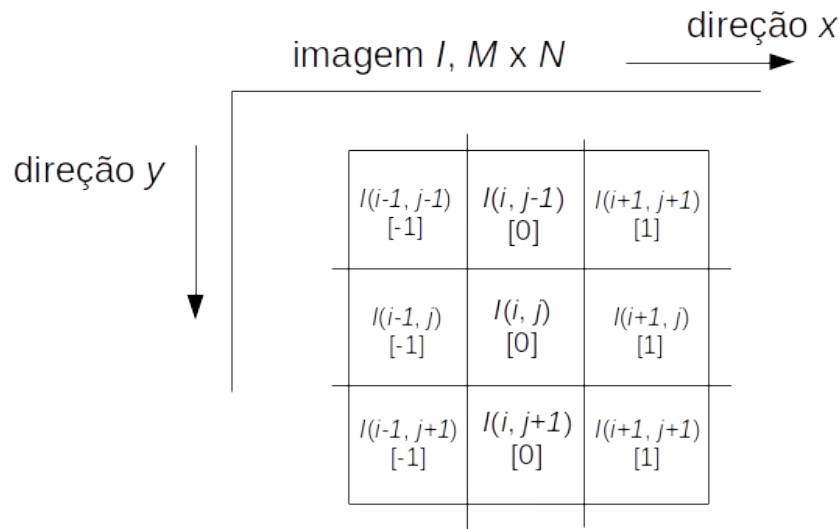
A tarefa de detecção de bordas consiste em determinar o limite, ou fronteira, entre duas regiões com propriedades e características distintas das intensidades dos *pixels* da imagem. Considerando que a imagem consiste em um *array* M por N, no qual podem-se definir as direções x e y, a detecção de bordas poderá ser obtida através de uma operação de convolução entre uma *máscara* (que configura-se como um operador) que é aplicado sobre um *pixel* de interesse da imagem e também em seus *pixels* vizinhos. Há diversos operadores para realizar essa tarefa. Um exemplo são os operadores de Prewitt, dados pelas matrizes a seguir:

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$$

$$G_y = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

onde  $G_x$  e  $G_y$  são as máscaras para detecção de bordas nas direções  $x$  e  $y$  da imagem digital. A imagem final segmentada é dada pela junção das imagens das duas bordas obtidas.

Por exemplo, a figura a seguir mostra como é a aplicação do operador de Prewitt na direção  $x$  para detectar existência de borda no *pixel*  $(i, j)$  da imagem  $I$ . No caso, considera-se que a imagem  $I$  é em nível de cinza. Ou seja, seus *pixels* têm valores que variam no intervalo  $[0, 255]$ , onde 0 é atribuído ao nível preto e 255 ao nível branco (os valores inteiros intermediários correspondem aos diferentes níveis de cinza).



O algoritmo para a resolução desta tarefa é definido a seguir.

Dada uma imagem  $I$ , em nível de cinza, de dimensões  $M \times N$ , fazer:

- Criar uma imagem (array)  $M \times N$ ,  $G_x$ , com todos os seus elementos iguais a 0;
- Criar uma imagem (array)  $M \times N$ ,  $G_y$ , com todos os seus elementos iguais a 0;
- Criar uma imagem (array)  $M \times N$ ,  $G$  (pode ser com todos os elementos iguais a zero).

*# Calculando as informações de bordas*

Para  $i$  variando de 1 até  $M-2$

Para  $j$  variando de 1 até  $N-2$

*# Cálculo de  $G_x$ ...*

$$G_x(i, j) = [ I(i+1, j-1) + I(i+1, j) + I(i+1, j+1) ] - [ I(i-1, j-1) + I(i-1, j) + I(i-1, j+1) ]$$

*# Saturando...*

Se  $G_x(i, j) < 0$ ,  $G_x(i, j) = 0$ ;

Se  $G_x(i, j) > 255$ ,  $G_x(i, j) = 255$ ;

*# Cálculo de  $G_y$ ...*

$$G_y(i, j) = [ I(i-1, j+1) + I(i, j+1) + I(i+1, j+1) ] - [ I(i-1, j-1) + I(i, j-1) + I(i+1, j-1) ]$$

*# Saturando...*

Se  $G_y(i, j) < 0$ ,  $G_y(i, j) = 0$ ;

Se  $G_y(i, j) > 255$ ,  $G_y(i, j) = 255$ ;

fim-para

fim-para

# Gerando imagem de saída...

Para  $i$  variando de 0 até  $M-1$

Para  $j$  variando de 0 até  $N-1$

$G(i, j) = G_x(i, j) + G_y(i, j)$

Se  $G(i, j) > 255$ ,  $G(i, j) = 255$

fim-para

fim-para

## Problema

Elabore um programa, na linguagem de programação mais conveniente, que possa, através de uma biblioteca de *threads*, criar duas *threads* para que uma possa obter a imagem de borda na direção  $x$  e, a outra, na direção  $y$ , a partir de uma imagem em nível de cinza existente, segundo o algoritmo apresentado anteriormente.

Neste programa, a *thread* principal (ou *thread* mãe) deverá abrir a imagem do problema e representá-la através de um *array* apropriado. Esta *thread* principal deverá, então, criar duas *threads*, em que serão calculadas as imagens de bordas  $G_x$  e  $G_y$ , respectivamente. Ao terminar suas tarefas, a *thread* mãe deverá ler as duas imagens (*arrays*)  $G_x$  e  $G_y$  resultantes e, assim, calcular a imagem de saída  $G$ . Note que será necessário sincronizar a *thread* mãe com a execução das duas *threads* filhas, invocando a função *join* correspondente da biblioteca utilizada.

Pelo fato das *threads* serem fluxos de execução distintos de um mesmo espaço de endereçamento, elas conseguem ter acesso aos *arrays* associados às imagens, caso eles tenham sido definidos como estruturas globais.

## Para análise do trabalho

A análise do resultado obtido será feita por meio de um relatório sucinto, submetido à turma virtual no SIGAA. Neste relatório, devem constar:

- A identificação do aluno ou aluna (nome completo e matrícula);
- Uma breve descrição sobre qual linguagem/biblioteca de *threads* foi utilizada, apresentando as funções que foram usadas (um parágrafo);
- Uma explicação sobre o programa desenvolvido, destacando os passos da resolução do problema proposto e as imagens resultantes (alguns parágrafos);
- Uma análise sobre a biblioteca de *threads* escolhida, informado se houve, ou não, complexidade do estudo e desenvolvimento de um programa *multithread* com a biblioteca em questão, além de outras análises que considerar pertinente (um parágrafo);
- Um anexo com o código do programa (ou *link* para GitHub, caso seja preferível); e
- Referências consultadas.

## Observações

- Embora possa parecer complexo, esta tarefa é relativamente simples, uma vez que o objeto de análise é a programação *multithread* (e não o processamento de imagens). Sendo assim, recomenda-se que ela possa ser realizada individualmente ou em duplas;
- Os exemplos apresentados pelas referências dão destaque à biblioteca Pthreads (em linguagem C/C++). Contudo, outras linguagem/bibliotecas preferenciais podem ser usadas;
- O relatório deve ser submetido em um arquivo no formato PDF, de modo a evitar eventuais problemas de formatação ao ser aberto em outros dispositivos/computadores;
- A imagem para análise está disponível para *download* na turma virtual (SIGAA). Ela estará em dois formatos (PNG, binário, e PGM, formato texto). Caso prefiram, podem usar os recursos que acharem necessários (bibliotecas, funções) para abrir uma imagem no formato PNG e ter acesso aos seus pixels. Caso não queiram usar bibliotecas para arquivos imagem, o arquivo no formato texto (*Portable Gray Map, PGM*) é bastante simples de ser interpretado e lido, de tal forma que é possível criar suas próprias rotinas para isso;



- Quaisquer dúvidas e/ou questões relacionadas ao trabalho podem ser tratadas pelo email (diogo.pedrosa@ufrn.br).