

Otimizando o tempo de execução no processamento de imagens

Henrique Miyamoto e Thiago Benites

I. CONTEXTUALIZAÇÃO

Threads, assim como processos, são mecanismos que permitem que um programa execute ações de forma que elas pareçam estar sendo executadas ao mesmo tempo. A diferença entre eles é que *threads* possuem área de memória compartilhada, o que não ocorre com processos [1]. Apresentamos uma comparação de desempenho entre diferentes métodos para aplicação de brilho em uma imagem. A mesma funcionalidade foi implementada de quatro maneiras diferentes: usando múltiplas *threadas*, usando multiprocessos, em uma única linha de execução, varrendo a matriz por linhas e por colunas. O objetivo é comparar e discutir os desempenhos de cada método a partir do tempo de execução de sistema de cada um deles.

A tabela I apresenta as sintaxes para os diferentes métodos de aplicação de brilho usados em nosso programa.

TABLE I
SINTAXE PARA DIFERENTES MÉTODOS DE APLICAÇÃO DE BRILHO

Método de implementação	Sintaxe
<i>Multithreadas</i>	destino.jpg = origem.jpg *float THR
Multiprocessos	destino.jpg = origem.jpg *float PRC
Varredura por linhas	destino.jpg = origem.jpg *float LIN
Varredura por colunas	destino.jpg = origem.jpg *float COL

II. DEMONSTRAÇÃO

Para medição dos tempos de execução foi usada a função `gettimeofday` da biblioteca `time.h`. As tabelas a seguir apresentam os tempos de execução dos diferentes métodos, para imagens de diferentes tamanhos: pequena 32x32 pixels (*dora.jpg*), média 640x640 (*ssa.jpg*) e grande 2592x1944 (*demo.jpg*).

TABLE II
TEMPOS DE EXECUÇÃO DOS MÉTODOS PARA IMAGEM PEQUENA.

Implementação	Tamanho da imagem	Tempo de sistema (ms)
Multithreads	32x32	3,7660
Multiprocessos	32x32	53,8670
Varredura por colunas	32x32	0,092000
Varredura por linhas	32x32	0,113000

III. ANÁLISE

Inicialmente, o número de *threads* e processos

Observa-se que o método com menor tempo de execução depende do tamanho da imagem utilizada. Para as imagens

TABLE III

TEMPOS DE EXECUÇÃO DOS MÉTODOS PARA IMAGEM MÉDIA.

Implementação	Tamanho da imagem	Tempo de sistema (ms)
Multithreads	640x640	31,2480
Multiprocessos	640x640	825,2130
Varredura por colunas	640x640	9,732000
Varredura por linhas	640x640	9,556000

TABLE IV

TEMPOS DE EXECUÇÃO DOS MÉTODOS PARA IMAGEM GRANDE.

Implementação	Tamanho da imagem	Tempo de sistema (ms)
Multithreads	2592x1944	101,4530
Multiprocessos	2592x1944	7188,9360
Varredura por colunas	2592x1944	169,008000
Varredura por linhas	2592x1944	147,995000

pequena e média, as varreduras em uma única linha de execução (por linha e por coluna) são as que apresentam melhor desempenho. No entanto, ao aplicar a funcionalidade à imagem grande, o tempo de execução do método *multithread* mostra-se o menor dentre os comparados. Isso se deve provavelmente à solução de compromisso do método de múltiplas *threads*: há um gasto computacional relacionado à criação delas, que, para imagens pequenas, acaba prejudicando o desempenho; para imagens grandes, no entanto, esse gasto é compensado pelo ganho de tempo da execução paralela das *threads*. No caso da implementação com múltiplos processos, seu desempenho é sempre inferior ao dos demais processos. Uma possível explicação para isso é o gasto necessário para criação da área de memória compartilhada.

REFERÊNCIAS

- [1] MITCHELL, Mark; OLDHAM, Jeffrey e SAMUEL, Alex. *Advanced Linux Programming*. Indianapolis: New Riders, 2001.

* Os arquivos do projeto estão disponíveis em <https://github.com/miyamotohk/linguagem-processamento-imagem>.