

## Trabalho de Sistemas Operacionais Primeira Unidade

**Disciplina:** DCA0108 – Sistemas Operacionais

**Turma:** 01, 2020.6, 24M12

**Professor:** Diogo Pinheiro Fernandes Pedrosa

**E-mail:** diogo@dca.ufrn.br

### Problema – Realce de Imagens em Nível de Cinza

O problema que deverá ser tratado neste primeiro trabalho é a criação de um programa que lance *threads* para realizar tarefas de pré-processamento em uma imagem em nível de cinza, mas especificamente proceder com técnicas de melhoria dessa imagem.

A figura a ser tratada é:



Para obtê-la, basta acessar a página:

<https://people.sc.fsu.edu/~jburkardt/data/pgma/balloons.ascii.pgm>

ou

<https://people.sc.fsu.edu/~jburkardt/data/pgma/balloons.png>

para baixar a imagem. Nestes *links*, a imagem é apresentada em dois formatos diferentes. O primeiro é o PGM no formato *ascii* (*balloons.ascii.pgm*), bastante simples de ser compreendido. O segundo *link* é para a mesma imagem, mas no formato PNG.

Uma imagem em nível de cinza é formada por uma estrutura em *array*, onde cada elemento é um *pixel* e possui seu valor definido entre 0 e 255, onde 0 indica o preto e 255 indica branco.

A tarefa consiste em escrever um programa que abra a imagem e lance duas *threads*:

- (a) A primeira *thread* deve realizar o realce dessa imagem por meio de uma técnica de *alargamento de contraste*, onde se busca aumentar a escala dinâmica dos níveis de cinza da imagem.
- (b) A segunda *thread* deve realizar o realce por meio de técnica de *equalização de histograma*. Esta técnica consiste na checagem da probabilidade de ocorrência de uma determinada intensidade de cinza e, pela aplicação de uma função contínua de transformação, a qual é baseada nessa distribuição das intensidades.

Uma boa imagem em nível de cinza tende a ter uma distribuição equalizada de seus *pixels* em cada um dos 256 níveis de cinza diferentes. Quando há uma grande concentração de *pixels* em uma mesma região do nível de cinza (como na região de *pixels* mais claros, como a imagem dos balões que foi apresentada), torna-se importante aplicar um algoritmo para que ocorra uma redistribuição de *pixels* nos vários valores dos níveis na imagem. Para fazer este tipo de tarefa, há alguns algoritmos bastante conhecidos e muitas referências que podem ser pesquisadas.

## Alargamento de Contraste

Toda transformação ocorrida no contraste de uma imagem busca a melhoria de sua qualidade, de acordo com critérios subjetivos de quem vê a imagem. Isso vai facilitar a percepção de informações que estão presentes na imagem.

Uma imagem digital  $I$ , com dimensões  $M \times N$  e em nível de cinza, é formada por *pixels* cujo valor variam entre 0 (menor possível) e 255 (maior possível). Quando se refere à intervalo de contraste, diz-se que é a diferença entre os valores de intensidade máxima e mínima que a imagem  $I$  possui. A partir desse intervalo de contraste, é possível aplicar uma função de transformação que modifique a intensidade de cada um dos *pixels* da imagem  $I$ , gerando uma imagem melhorada  $G$  com intervalo de contraste variando no intervalo  $[0, 255]$ .

Por simplicidade, essa função de transformação pode ser a seguinte função linear:

$$G(i, j) = \frac{255}{I_{max} - I_{min}} \cdot (I(i, j) - I_{min})$$

onde:

- $I_{max}$  é o maior valor de intensidade de *pixel* da imagem  $I$ .
- $I_{min}$  é o menor valor de intensidade de *pixel* da imagem  $I$ .

Assim sendo, a *thread* que irá proceder esse realce deverá ter o seguinte algoritmo:

Declarar uma imagem (matriz)  $G$ , com mesma dimensão da imagem  $I$ .

Obter o valor  $I_{\max}$  de  $I$ .

Obter o valor  $I_{\min}$  de  $I$ .

Para  $i$  variando de 0 até  $M-1$

Para  $j$  variando de 0 até  $N-1$

$G(i, j) = [255 / (I_{\max} - I_{\min})] * (I(i, j) - I_{\min})$

Fim-para

Fim-para

Salvar  $G$  em um arquivo.

Uma observação: deve-se notar que a operação de transformação pode resultar em valores de ponto flutuante. Assim sendo, deve-se ter o cuidado para converter o resultado final para inteiro.

## Equalização de Histograma

Em uma imagem digital, chama-se de histograma a contagem de *pixels* para cada um dos 256 níveis de cinza que a imagem possui. Isso é representado por um gráfico que mostra essa distribuição de níveis de cinza, ou por um vetor de 256 posições, onde cada elemento contém a contagem de *pixels* naquela intensidade (as posições 0, 1, 2, ..., 255 representam as intensidades de cinza). Por meio dessa distribuição, é possível determinar a probabilidade de ocorrência de um nível de cinza em uma imagem.

Um histograma de uma imagem  $I$  pode ser facilmente calculado através do seguinte procedimento:

Declarar um vetor  $H$  com 256 posições, com todos os valores iguais a zero (0).

Para  $i$  variando de 0 até  $M$

Para  $j$  variando de 0 até  $N$

$H(I(i, j))++$

Fim-para

Fim-para

Observe, neste procedimento, que o elemento  $(i, j)$  da imagem  $I$  serve como índice para o elemento do vetor de histograma  $H$ .

Para calcular a probabilidade de ocorrência de cada nível de cinza, basta dividir cada elemento de  $H$  pelo total  $M \times N$  de *pixels* da imagem.

Declarar um vetor  $P$ , com 256 posições, com todos os valores iguais a zero (0).

Para  $i$  variando de 0 até 255

```
        P(i) = H(i) / ( M * N )  
Fim-para
```

Uma função de equalização de histograma bastante utilizada é a função *probabilidade acumulada*. Ou seja, a nova imagem *G* terá a intensidade de seus *pixels* calculados pelo acúmulo da probabilidade de ocorrência da intensidade do *pixel* correspondente da imagem *I*.

Essa probabilidade acumulada pode ser obtida em um vetor, também de 256 elementos, da seguinte forma:

```
Declarar um vetor p de 256 elementos e todos iguais a zero (0).
```

```
Declarar uma variavel cont, como ponto flutuante.
```

```
Para i variando de 0 até 255  
    cont = 0  
    Para j variando de 0 até i  
        cont = cont + P(j)  
    Fim-para  
    p(i) = cont  
Fim-para
```

Por fim, para gerar a imagem de saída, faz-se:

```
Declarar uma imagem G, de dimensões M por N, e com todos os elementos iguais a zero (0).
```

```
Para i variando de 0 até M-1  
    Para j variando de 0 até N-1  
        G(i, j) = round( 255 * p( I(i, j) ) )  
    Fim-para  
Fim-para  
  
Salvar G em arquivo
```

## Observações

1. Este é um trabalho em grupo (até 4 componentes);
2. A imagem a ser manipulada pode ser no formato PGM tipo *ascii*, no PNG ou em outro formato qualquer de preferência. Utilize bibliotecas adequadas da linguagem de programação escolhida pelo grupo;
3. Caso desejem abrir a imagem no formato *ascii*, a página <http://netpbm.sourceforge.net/doc/pgm.html> apresenta como é esse formato, o que permitirá com que o grupo elabore um código para abrir e manipular a imagem;
4. Para salvar as imagens de saída, o grupo pode escolher qualquer formato de imagem que desejar (PGM ou outro qualquer);

5. A linguagem com a qual vão desenvolver o programa e qual biblioteca de *threads* vão usar é de livre escolha do grupo. Mas um detalhe importante: a imagem original (ou a estrutura que a recebe) não pode ser modificada, já que as *threads* lançadas realizam a leitura dos *pixels* para fazer as tarefas determinadas;
6. Os resultados deverão ser apresentados em um texto no formato de artigo. Utilizem o *template* da Sociedade Brasileira de Computação para escrita do trabalho. Este *template* está disponível em:  
<https://www.sbc.org.br/documentos-da-sbc/summary/169-templates-para-artigos-e-capitulos-de-livros/878-modelosparapublicaodeartigos>
7. O trabalho é relativo a *threads*, e não sobre processamento de imagens. Assim sendo, o texto deve conter pelo menos:
  - (a) Uma introdução, que apresente o problema de programação concorrente e que justifique a utilização de *threads* ao invés de criação de processos;
  - (b) Uma seção que explique a biblioteca de *threads* escolhida, abrangendo também o desenvolvimento do código que foi construído; e
  - (c) Uma seção que apresente os resultados obtidos (as imagens realçadas)
8. O trabalho deve estar no formato PDF e enviado para a tarefa aberta no SIGAA, até o prazo final combinado entre a turma e apresentado no sistema.