

**Exercício de Programação**

**Descrição:**

Deseja-se processar uma imagem digital I (de  $250 \times 250$  pixels – 256 níveis de cinza), pixel a pixel, armazenada em um arquivo texto. Uma imagem digital, em nível de cinza, é representada por uma matriz de pontos, e cada ponto pode assumir valores discretos na faixa  $[0, 255]$ . Para tanto, você deve desenvolver um sistema com interface gráfica cujo *layout* é mostrado na Figura 1.

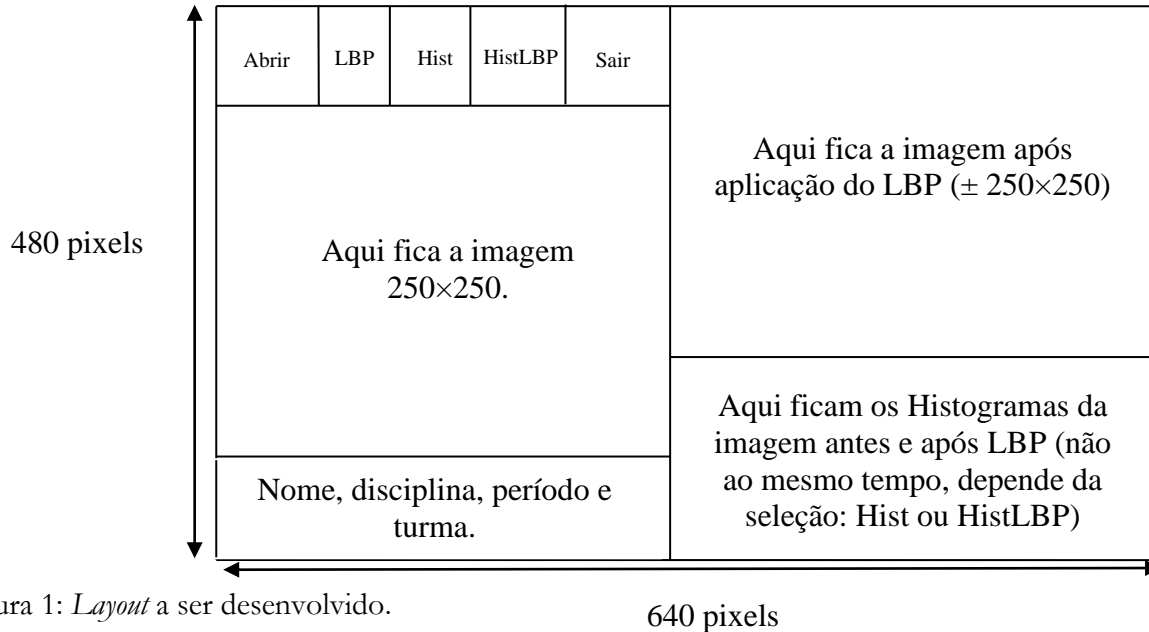
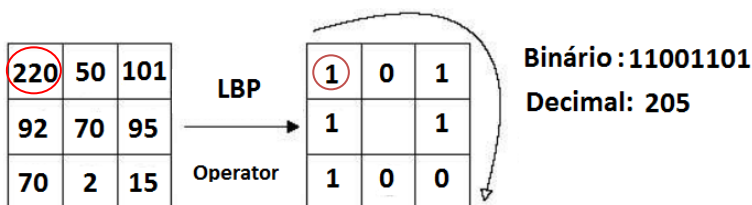


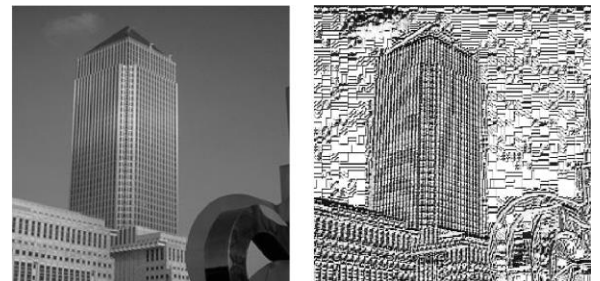
Figura 1: *Layout* a ser desenvolvido.

Todo o controle da interface deve ser feito usando o *mouse* para realizar a ação desejada. O objetivo é tratar uma imagem, gerando como resultado, além da própria imagem, tratada, o histograma da imagem tratada. A imagem deve ser tratada com a técnica conhecida como LBP – *Local Binary Pattern*.

A técnica LBP funciona da seguinte forma: supondo uma janela de  $3 \times 3$  pixels, o LBP consiste em rotular o valor do pixel central fazendo comparações com cada um dos seus 8 pixels vizinhos. A janela  $3 \times 3$  deve deslizar completamente pela imagem, da esquerda para a direita, de cima para baixo, de forma a operar sobre cada pixel da imagem (excluindo-se os pixels de borda da imagem). Assim, o pixel central da janela coincidirá com um dado pixel da imagem. A partir daí são realizadas as comparações do pixel central com seus 8 pixels vizinhos. Cada comparação produz um número binário em que o valor 0 será atribuído quando o pixel vizinho for menor que o pixel central e 1 caso contrário. Ao final de todas as comparações, os 8 valores são concatenados e um número binário de 8 dígitos é formado. Este número, convertido para a forma decimal, gera um valor de 0 a 255 que será atribuído ao pixel da imagem que coincide com o pixel central da janela. A Figura 2 ilustra esse processo, perceba que a ordem de comparação com os pixels vizinhos não é feita de forma aleatória sendo, neste caso, em sentido horário.



a)



b)

Figura 2: Em a) observa-se a aplicação do LBP e em b) observa-se a figura de um prédio e o resultado do LBP. Em a), o pixel circundado com um círculo vermelho é o ponto de partida e corresponderá ao bit LSB (*least significant bit*).

Todas as linhas da interface devem ser brancas e o fundo deve ser preto. Já para o menu de opções (menu superior), uma vez que uma opção seja selecionada, deve-se trocar a cor da legenda da opção de branco para amarelo e assim permanecer até que uma outra opção seja selecionada. A seleção deverá ser exclusivamente feita pelo uso do mouse. Para tanto, deve-se posicionar o cursor do mouse em qualquer parte do retângulo que contiver a função desejada e clicar com o botão esquerdo do mouse para executar a operação. No menu, tem-se as opções:

Opção	Funcionalidade
<b>Abrir</b>	Abrir o arquivo contendo a imagem, aqui chamado de 'imagem.txt' e a desenha na região reservada a ela conforme mostrado na Figura 1;
<b>LBP</b>	Gera a imagem tratada pelo LBP.
<b>Hist</b>	Calcula e desenha o histograma da imagem original
<b>HistLBP</b>	Calcula e desenha o histograma da imagem tratada;
<b>Sair</b>	Sair do programa, retornando a janela DOS do dosbox para o modo original;

A qualquer instante, você pode aplicar qualquer uma das funções do menu superior, desde que a imagem já esteja carregada, e a ação deve ser executada. Ao clicar em 'Abrir' com um arquivo já aberto, seu código deve fechar o arquivo aberto para, novamente, abri-lo, recarregando a imagem na tela. As cores na tela são de acordo com a Tabela 1. Observe que você vai ter que requantizar os níveis de cinza da imagem, pois no modo gráfico usado tem-se 640×480 pixels, cada pixel com um máximo de 16 valores, conforme a Tabela 1 (gerando falso coloreamento).

Tabela 1: Tabela de Cores modo VGA 640×480

Preto	0	Cinza	8
Azul	1	Azul_claro	9
Verde	2	Verde_claro	10
Cyan	3	Cyan_claro	11
Vermelho	4	Rosa	12
Magenta	5	Magenta_claro	13
Marrom	6	Amarelo	14
Branco	7	Branco_intenso	15

Para facilitar a programação, use as funções gráficas do arquivo LINEC.ASM, que contém o básico para a mudança do modo de vídeo (gráfico, 640×480 com 16 cores), funções de plotar ponto, desenhar uma linha, posicionar o cursor e escrever um caractere. Este programa deve ser usado como referência para a programação do exercício. Em processamento de imagens o ponto (0,0) deve ser o canto superior esquerdo.

Procedimento:

O programa deverá abrir o arquivo, ler o seu conteúdo e processá-lo de acordo com o que foi selecionado. Para ler um arquivo, utilize as seguintes funções da INT 21H: 3DH, 3EH, 3FH, e 4CH e as interrupções da BIOS das funções de vídeo (INT 10H). Para a utilização do mouse, utilize a INT 33H, cuja forma de utilização é semelhante à INT 21H (veja uma descrição detalhada em [http://stanislavs.org/helppc/int\\_33.html](http://stanislavs.org/helppc/int_33.html)).

Cada amostra de valor de um pixel varia na faixa inteira de 0 a 255 (0 é convencionado como nível de cinza preto e 255 como nível de cinza branco). Seu programa deve abrir o arquivo, converter cada amostra de string, composta por até 3 dígitos ASCII, para um valor inteiro de 8 bits. No arquivo TXT, cada valor de pixel (representada pela string de até 3 dígitos) está separado do seu sucessor por um espaço em branco (20H na Tabela ASCII). Por exemplo:

- a string de ASCII 29, composta por 2 bytes, representa o número 29 (1 byte),
- a string de ASCII 123, composta por 3 bytes, representa o número 123 (1 byte),
- a string de ASCII 7, composta por 1 byte, representa o número 7 (1 byte).

No Anexo é possível encontrar um código em Matlab para cálculo de histograma e figuras retratando a relação entre imagens e histogramas.

### **Importante: Procedimento para entrega**

Enviar para o email: [evandro.salles@ufes.br](mailto:evandro.salles@ufes.br) com o assunto: “EP\_sistemb1\_2023-1”. O nome do arquivo ASM deve ser as iniciais de seu nome e deve conter, no máximo, 8 caracteres. No arquivo, não se esqueça de adicionar logo nas primeiras linhas, comentadas, o seu nome completo e a turma. O nome do arquivo a ser aberto por seu programa é “imagem.txt”. Enviar até 19/06/2023, às 23:59h.

### **Atenção:**

O exercício é individual. Qualquer cópia parcial ou total acarretará na atribuição da nota 0 (zero) para todos os envolvidos.

### **Bibliografia:**

Gonzalez, R. C.; Woods, R. E. Digital Image Processing (3rd Edition). Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 2006. ISBN 013168728X.

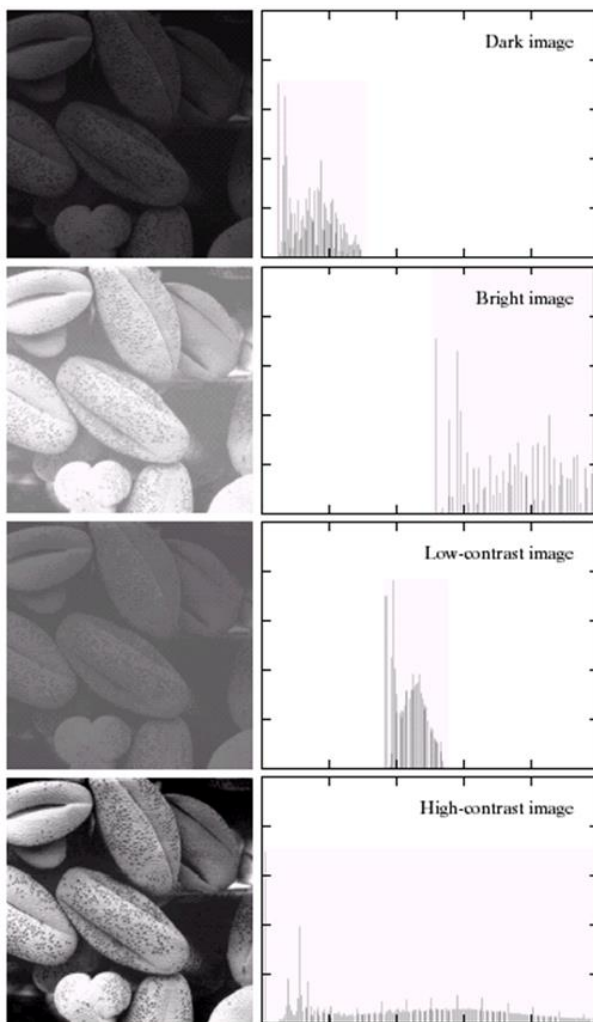
Ribeiro, M. V. L. , Proposta de Local Binary Pattern Coerente e Incoerente na Categorização de Cenas, Dissertação de Mestrado, PPGEE – UFES, outubro de 2017.

## Anexo

Quadro 1: No Matlab/Octave, o cálculo de um histograma pode ser descrito da seguinte forma (script):

```
% Método de equalização de histogramas.  
% Aqui carrega-se a imagem original  
x = imread('imagem.jpg');  
% Aqui, a imagem foi redimensionada de 500x500 pixels para 250x250 pixels.  
f = imresize(x, 0.5, 'bicubic');  
% No seu caso, em assembly, a imagem de 250x250 pontos será fornecida no formato raw.  
% Aqui calcula-se o histograma da imagem  
histograma=zeros(256,1);  
for i=1:250,  
    for j=1:250,  
        % Aqui, observe que o Matlab não indexa posição 0 e uma imagem pode ter nível de cinza=0.  
        histograma(f(i,j)+1)=histograma(f(i,j)+1)+1;  
    end;  
end;  
colormap(gray);  
subplot(211); imshow(uint8(f));  
subplot(212); plot(histograma);
```

Na Figura 3 é possível observar alguns exemplos de imagens e seus histogramas.



Caso 1: imagem com baixo contraste e brilho.

Caso 2: imagem com baixo contraste e muito brilho.

Caso 3: : imagem com baixo contraste e brilho médio.

Caso 4: : imagem com bom contraste e brilho adequado.

Figura 3: Exemplos de imagens e seus histogramas. Observe os fenômenos de brilho e contraste e histograma.