Laboratório de Redes de Computadores

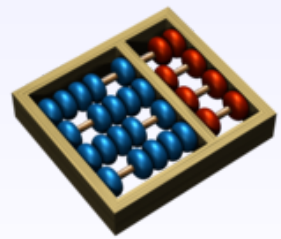
MC102 - Algoritmos e Programação de Computadores

Laboratório 4

Carlos A. Astudillo Trujillo

Carlos Alberto Astudillo Trujillo
Instituto de Computação
Universidade Estadual de Campinas

Campinas, 10 de maio de 2012



Laboratório de Redes de Computadores

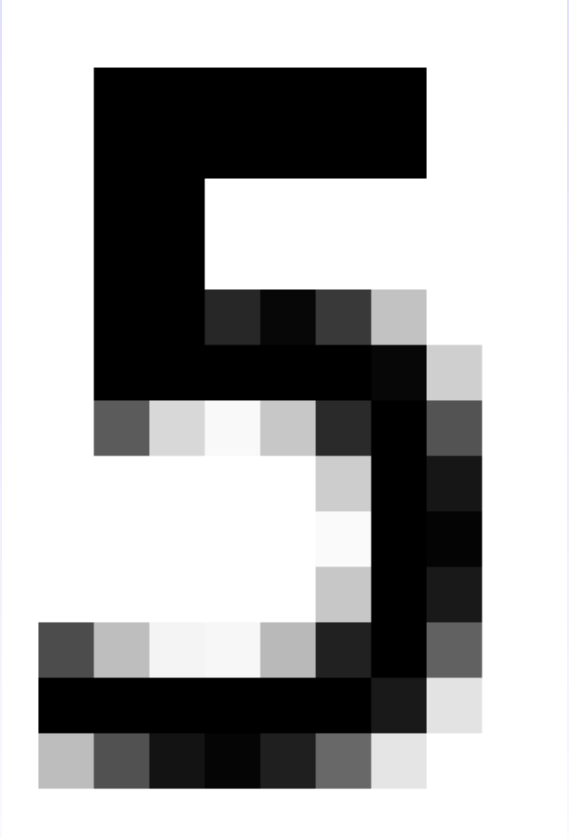
Roteiro I

Laboratório de Redes de Computadores

Roteiro

Laboratório de Redes de Computadores

Imagem



Laboratório de Redes de Computadores

Representação Matricial de Uma Imagem

255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255
255	255	0	0	0	0	0	0	255	255	255
255	255	0	0	0	0	0	0	255	255	255
255	255	0	0	255	255	255	255	255	255	255
255	255	0	0	255	255	255	255	255	255	255
255	255	0	0	39	7	57	194	255	255	255
255	255	0	0	0	0	0	7	207	255	255
255	255	91	216	249	199	42	0	83	255	255
255	255	255	255	255	255	205	0	22	255	255
255	255	255	255	255	255	250	0	4	255	255
255	255	255	255	255	255	199	0	25	255	255
255	76	190	244	247	185	33	0	97	255	255
255	0	0	0	0	0	0	25	227	255	255
255	189	81	19	5	31	104	229	255	255	255
255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255

Laboratório de Redes de Computadores

Significado dos Píxeis

Definição de Pixel

Convencionalmente, cada uma das células desta matriz é chamada de pixel (picture element) e seu valor representa o tom de cinza da imagem.

Valor dos Píxeis e Cor

Normalmente, o valor de uma célula é representado por um número inteiro no intervalo [0, 255] e os valores mais próximos de zero são mais escuros.

Laboratório de Redes de Computadores

Operações sobre Imagem

Neste laboratório, serão consideradas três operações clássicas da área de processamento de imagem denominadas:

- Limiarização.
- Dilatação.
- Erosão.

Laboratório de Redes de Computadores

Limiarização

A Limiarização é uma técnica bastante simples que classifica as células de uma determinada imagem de acordo com um número inteiro t, chamado de limiar. No caso mais simples, os valores de cada uma das células da imagem são comparados com o valor de t, aplicando a seguinte regra:

$$J(x,y) = \begin{cases} 0, \text{ se } I(x,y) < t, \\ 1, \text{ caso contrário.} \end{cases} \quad (1)$$

em que I(x,y) é o valor da célula analisada na imagem de entrada e J(x,y) é o valor desta mesma célula na imagem de saída.

Laboratório de Redes de Computadores

Erosão e Dilatação I

Erosão e dilatação são as duas operações fundamentais de uma metodologia para análise de imagens chamada morfologia matemática. Nestas operações, o valor resultante em cada célula depende dos valores de algumas células vizinhas.
Em cada célula, o resultado da erosão é o mínimo entre os valores das células vizinhas, similarmente, o resultado da dilatação é o máximo entre estes valores.

Erosão e Dilatação II

Neste laboratório, somente as 8 células mais próximas (além do valor da própria célula analisada) serão consideradas e, desta maneira, o valor resultante da aplicação da erosão é obtido por:
$J(x, y) = \min\{I(x-1, y-1), I(x, y-1), I(x+1, y-1), I(x-1, y), I(x, y), I(x+1, y), I(x-1, y+1), I(x, y+1), I(x+1, y+1)\}$
Enquanto a dilatação é calculada por:
$J(x, y) = \max\{I(x-1, y-1), I(x, y-1), I(x+1, y-1), I(x-1, y), I(x, y), I(x+1, y), I(x-1, y+1), I(x, y+1), I(x+1, y+1)\}$

Erosão e Dilatação III

$I(x-1,y-1)$	$I(x, y-1)$	$I(x+1, y-1)$
$I(x-1,y)$	$I(x, y)$	$I(x+1, y)$
$I(x-1,y+1)$	$I(x, y+1)$	$I(x+1, y+1)$

A Figura ilustra a disposição dos 8 vizinhos mais próximos do pixel (x,y) em uma imagem representada por uma matriz I.
Deve-se observar que nas bordas da imagem (quando se avalia a primeira ou a última coluna/linha da imagem, por exemplo) alguns vizinhos não existem e, neste caso, seus valores devem ser desconsiderados do cálculo.

Laboratório de Redes de Computadores

Objetivos do Laboratório

O objetivo deste laboratório é exercitar a manipulação de funções utilizando matrizes. Portanto, seis funções devem ser criadas obrigatoriamente, implementando os seguintes protótipos:

- **int** *ler_imagem*(**int** imagem[][30], **int*** n_cols, **int*** n_linhas, **int*** maximo);
- **void** *limiarizacao*(**int** imagem_entrada[][30], **int** imagem_saida[][30], **int** n_cols, **int** n_linhas, **int** limiar);
- **void** *erosao*(**int** imagem_entrada[][30], **int** imagem_saida[][30], **int** n_cols, **int** n_linhas);
- **void** *dilatacao*(**int** imagem_entrada[][30], **int** imagem_saida[][30], **int** n_cols, **int** n_linhas);
- **void** *imprimir_imagem*(**int** imagem[][30], **int** n_cols, **int** n_linhas, **int** maximo);
- **int** *valor_maximo*(**int** imagem[][30], **int** n_cols, **int** n_linhas);

Laboratório de Redes de Computadores

Objetivos do Laboratório

É importante observar que:

- Os protótipos das funções pressupõem o uso de matrizes, que podem ser pré-definidas com tamanho 30x30 embora nem sempre todas as posições sejam utilizadas.
- O valor máximo da imagem deve ser recalculado após uma operação ser aplicada.
- A função ler imagem deve devolver um inteiro informando se houve ou não erro na leitura da imagem. Alternativamente, pode-se especificar o tipo de erro utilizando um número para cada tipo.

Laboratório de Redes de Computadores

Entradas e Saídas

O programa deverá receber primeiro a operação a ser efetuada (l, d ou e) e subsequentemente os dados de entrada, conforme a seguinte Tabela.

Operação	Descrição	Dados de Entrada	Dados de Saída
l	Limiarização	valor para limiar imagem de entrada	imagem de saída
d	Dilatação	imagem de entrada	imagem de saída
e	Erosão	imagem de entrada	imagem de saída

Laboratório de Redes de Computadores

Exemplos de Entrada e Saída

Exemplo de Limiarização

Entrada

l 15
P2
7 3
21
0 1 2 3 4 5 6
11 12 13 14 15 16 17
10 10 10 21 21 21 21

Saída

P2
7 3
1
0 0 0 0 0 0 0

Laboratório de Redes de Computadores

Formato de Imagens PGM

1. A primeira linha contém a string P2 e pode ser desconsiderada na leitura.
2. A segunda linha, são informados dois inteiros indicando o tamanho horizontal e vertical da imagem, respectivamente.
3. A terceira linha contém um inteiro com o maior valor da imagem.
4. Em seguida, são informados os inteiros com os valores das células da imagem.

Um exemplo de imagem no formato PGM é apresentado a seguir:

```
P2
24 7
15
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 3 3 3 3 0 0 7 7 7 7 0 0 11 11 11 11 0 0 15 15 15 15 0
0 3 0 0 0 0 0 7 0 0 0 0 0 11 0 0 0 0 0 15 0 0 15 0
0 3 3 3 0 0 0 7 7 7 0 0 0 11 11 11 0 0 0 15 15 15 15 0
0 3 0 0 0 0 0 7 0 0 0 0 0 11 0 0 0 0 0 15 0 0 0 0
0 3 0 0 0 0 0 7 7 7 7 0 0 11 11 11 11 0 0 15 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

Laboratório de Redes de Computadores

Entradas inválidas

Os seguintes casos serão testados e as respectivas mensagens deverão ser informadas:

- Se o caractere informado não corresponder a nenhuma das operações citadas, deve-se imprimir: Erro: Opcao invalida.
- Se pelo menos uma das dimensões da imagem não forem maiores que zero, deve-se interromper a leitura e imprimir: Erro: Dimensoes devem ser maiores que zero.
- Se o valor de algum pixel for menor que zero e maior do que 255, deve-se interromper a leitura e imprimir: Erro: Valor invalido.

Exemplos de Entradas inválidas I

Operação Invalida

Entrada

o **Saída**

Erro: Opcao invalida.

Dimensões Invalidas

Entrada

I 10

P2

0 30

Saída

Erro: Dimensoes devem ser maiores que zero.

Exemplos de Entradas inválidas II

Valor de Celula Invalido

Entrada

I 15
P2
30 30
0 10 10 0 5 -1

Saída

Erro: Valor invalido.

Laboratório de Redes de Computadores

Vizualizando a imagem

Ao executar os testes, pode-se redirecionar a saída do programa para um arquivo e visualizar a imagem resultante.
Como você deve saber, para redirecionar a saída de seu programa basta executar o comando:
$./programa < entrada.in > nomeimagem.pgm$

Laboratório de Redes de Computadores

Função Main

```
int main(){
int imagemEntrada[30][30], imagemSaida[30][30], numeroLinhas, numeroColunas, limiar,
     valorMaximo;
int flag;
char opcao;
    scanf("%c",&opcao);
    switch(opcao){
    case '1':
      break;
    case'd':
        flag = ler_imagem(imagemEntrada, &numeroColunas, &numeroLinhas, &valorMaximo)
        if(flag==1){Tratar Erro!}
        else if(flag==2){Tratar Erro!}
        break;
    case'e':
        break;
    default:
         printf("Tratar Erro!");
        break;
    }
return 0;
}
```

Laboratório de Redes de Computadores

Função Ler Imagem

```
int ler_imagem (int imagem[][30], int *n_cols, int *n_linhas, int *maximo){
int i,j;
 scanf ("%d%d",n_cols,n_linhas);

if (*n_cols<=0||*n_linhas<=0){
 return 1;
 }

scanf ("%d",maximo);

for (i=0; i < *n_linhas; i++){
    for (j=0; j < *n_cols; j++){
    scanf ("%d",&imagem[i][j]);

    if (imagem[i][j]<0 || imagem[i][j]>255)
    return 2;

    }
}
return 0;
}
```

Laboratório de Redes de Computadores

Função Dilatacão (Ou Erosão)

```
void dilatacao(int imagem_entrada[][30],int imagem_saida[][30],int n_cols,int n_linhas){
    int i,j

    for(i=0; i < n_linhas; i++){
        for(j=0; j< n_cols; j++){
            if(i==0&&j==0){
                ...
            //imagem_saida[i][j]= menor(...);
            imagem_saida[i][j]= maior(...);
            }
            else if(i==0&&j == (n_cols-1)){...}
            else if(i==(n_linhas-1) && j==0){...}
            else if(i==(n_linhas-1) && j==(n_cols-1)){...}
            else if(i==0 && j!=0 && j!=(n_cols-1)){...}
            else if(j==(n_cols-1) && i !=0 && i!= (n_linhas-1)){...}
            else if(i==(n_linhas-1) && j !=0 && j != (n_cols-1)){...}
            else if(j==0 && i!=0 && i!=(n_linhas-1)){...}
            //if (i!=0&&i!=(n_linhas-1)&&j!=0&&j!=(n_cols-1))
            else{
           ...

            }
        }
    }
}
```

Laboratório de Redes de Computadores

Muito Obrigado

Carlos A. Astudillo Trujillo

website: http://www.lrc.ic.unicamp.br/~castudillo/mc102
e-mail: castudillo@lrc.ic.unicamp.br