

DISCIPLINA: BIOENGENHARIA.

Renata Coelho Borges
`renatacoelho@utfpr.edu.br`

Universidade Tecnológica Federal do Paraná

Filtragem, Melhoramento e Restauração de Imagens.

1 INTRODUÇÃO E VISÃO GERAL

2 PROCESSAMENTO POR PONTOS

3 *Bit-Level Slicing*

4 EQUALIZAÇÃO DE HISTOGRAMA

5 MASCARAMENTO: FILTRAGEM LINEAR NO DOMÍNIO ESPACIAL

6 FILTROS ESPACIAIS PARA NITIDEZ

- Imagens são influenciadas por ruídos de medição
- Perda na qualidade pode influenciar análise
- Imagens médica: destacar certas regiões

As técnicas de restauração e melhoramento de imagens podem ser consideradas como algoritmos computacionais que recebem uma imagem como entrada e geram uma versão aprimorada ou restaurada dessa imagem como saída.

- Restauração de imagens: domínio do **espaço** ou frequência (Fourier)
 - Filtragem no domínio espacial
- Melhoramento de imagens no domínio espacial:
 - Processamento por pontos
 - Cada pixel da imagem original nas coordenadas (x, y) é processado para criar o *pixel* correspondente nas coordenadas (x, y) na imagem processada.
 - Técnicas de mascaramento
 - não altera apenas o *pixel* nas coordenadas (x, y) da imagem original, mas também alguns *pixels* vizinhos desse ponto estão envolvidos na geração do *pixel* em coordenadas (x, y) na imagem aprimorada

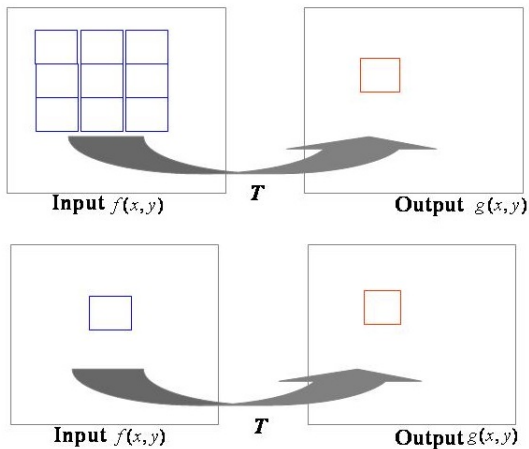


FIGURA: Diagrama esquemático de mascaramento (primeiro quadro) e processamento por pontos (segundo quadro).

- 1 INTRODUÇÃO E VISÃO GERAL
- 2 PROCESSAMENTO POR PONTOS
- 3 *Bit-Level Slicing*
- 4 EQUALIZAÇÃO DE HISTOGRAMA
- 5 MASCARAMENTO: FILTRAGEM LINEAR NO DOMÍNIO ESPACIAL
- 6 FILTROS ESPACIAIS PARA NITIDEZ

PROCESSAMENTO POR PONTOS

- Processamento por pontos envolve uma transformação matemática que modifica o valor dos *pixels* da imagem original

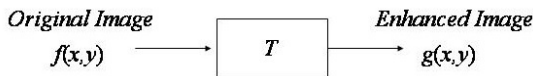
$$g(x, y) = T[f(x, y)]$$

em que

$f(x, y)$ é a imagem original

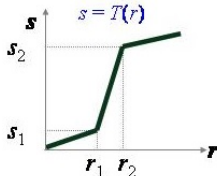
$g(x, y)$ é a imagem melhorada

T descreve a transformação entre essas duas imagens



MELHORAMENTO DO CONTRASTE

- Se o objeto de interesse ocupa apenas uma faixa específica da escala de cinza, então pode-se querer manipular a imagem de tal forma que o objeto ocupe uma faixa maior do nível de cinza e, portanto, aumente a visibilidade do objeto
- Análise de tumor cercado por órgãos mais escuros ou mais claros. Se outros órgãos e objetos não são o foco principal da análise, então, para visualizar melhor o tumor, é possível realizar o realce do contraste para “alongar” o nível de cinza do tumor
- O aprimoramento de contraste é um método para criar melhor visibilidade de um determinado



Exemplo (imagem de ultrassom do coração): aprimoramento e aumento do nível de cinza de [150,200] para [105,200].



(A)



(B)

```
I=imread('heart.tif');
S = size(I);
I = double(I);
J = zeros(S(1),S(2));
for i=1:S(1)
    for j=1:S(2)
        if I(i,j) <= 150
            J(i,j) = I(i,j);
        elseif (150 < I(i,j) ) & (I(i,j) < 200)
            J(i,j)=1.9*(I(i,j)-150)+ 105;
        elseif (I(i,j) >= 200)
            J(i,j) = .8*(I(i,j)-200) + 200;
        end
    end
end
imshow (I,[0,255]);
imshow (J,[0,255]);
```

- leitura da imagem
- operações numéricas:
conversão de *unsigned integer* para *double*

- 1 INTRODUÇÃO E VISÃO GERAL
- 2 PROCESSAMENTO POR PONTOS
- 3 *Bit-Level Slicing*
- 4 EQUALIZAÇÃO DE HISTOGRAMA
- 5 MASCARAMENTO: FILTRAGEM LINEAR NO DOMÍNIO ESPACIAL
- 6 FILTROS ESPACIAIS PARA NITIDEZ

- Em computadores digitais o nível de cinza de cada *pixel* de uma imagem representa um ou mais bytes
- Em imagens digitais, de acordo com o agrupamento de 0's e 1's, o nível de cinza é codificado como um número binário
- MSB e LSB
 - Uma alteração no LSB alteraria a escala de cinza em apenas 1 nível
- *Bit-Level Slicing* é um método de representar cada pixel de uma imagem com um ou mais bits do byte
 - resulta na perda de resolução, mas ao mesmo tempo reduz o tamanho do armazenamento necessário para salvar a imagem, já que cada pixel é agora representado por um número menor de bits

Objetivos do *Bit-Level Slicing*:

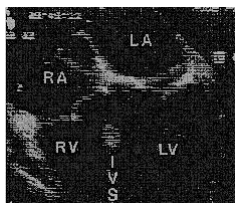
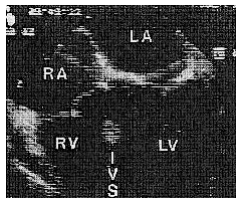
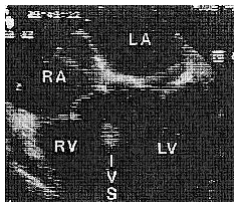
- 1 representar a imagem com menos bits e ao mesmo tempo manter um nível mínimo de qualidade
- 2 converter a imagem de nível de cinza para uma escala binária
- 3 melhorar a imagem, concentrando-se nos níveis de cinza que são mais importantes.

Exemplo: Neste exemplo é mostrado que preservando os quatro MSBs e descartando os bits restantes da imagem original a qualidade da imagem é mantida e o tamanho do espaço necessário para armazenar a imagem é reduzido.

```
I=imread('2.jpg');
I=rgb2gray(I);
imshow(I,[0 255]);
S=size(I);
I=double(I);
I=dec2base(I,2);
newS=size(I);
J= zeros(S(1),S(2) );
    for i = 1:newS(1)
        k=char(I(i,:));
        k(5)='0';
        k(6)='0';
        k(7)='0';
        k(8)='0';
        k=base2dec(k,2);
        a=fix(i/S(1))+1;
        b=mod(i,S(1));
            if b==0
                b=S(1);
                a=a-1;
            end
        J(b,a)=k;
    end
figure,
imshow(J,[0 255]);
```

- Leitura de cada pixel da imagem original em nível de cinza em 8 bits
- Loop “for” onde apenas os quatro MSBs são preservados e o restante dos bits são definidos como 0

- Imagem original
- Imagem após o fatiamento no nível de bits
- Imagem após descartar os bits 2, 5, 6, 7 e 8



- 1 INTRODUÇÃO E VISÃO GERAL
- 2 PROCESSAMENTO POR PONTOS
- 3 *Bit-Level Slicing*
- 4 EQUALIZAÇÃO DE HISTOGRAMA
- 5 MASCARAMENTO: FILTRAGEM LINEAR NO DOMÍNIO ESPACIAL
- 6 FILTROS ESPACIAIS PARA NITIDEZ

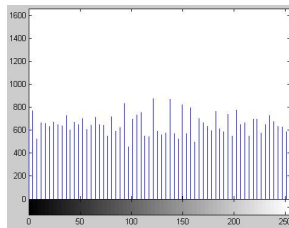
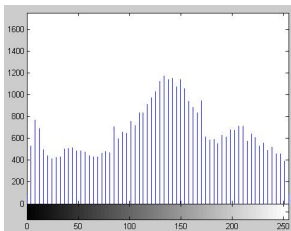
EQUALIZAÇÃO DE HISTOGRAMA

- Técnica mais popular para melhoramento de imagens
 - Baseada na manipulação de imagens manipulando o histograma respectivo
- Necessidade de equalização: Por que?
 - “Equalizar” a distribuição dos níveis de cinza na imagem antes de outras etapas de processamento

A tarefa de equalização do histograma é criar um equilíbrio entre todos os níveis de cinza e criar uma imagem cujo histograma esteja próximo da distribuição uniforme

Exemplo: Utilização da função pronta disponibilizada pelo Matlab:

```
I=imread('7.jpg');  
I=rgb2gray(I);  
J=histeq(I)  
imshow(I),  
figure  
imshow(J)  
figure,  
Imhist(I,64)  
figure,  
Imhist(J,64)
```



- 1 INTRODUÇÃO E VISÃO GERAL
- 2 PROCESSAMENTO POR PONTOS
- 3 *Bit-Level Slicing*
- 4 EQUALIZAÇÃO DE HISTOGRAMA
- 5 MASCARAMENTO: FILTRAGEM LINEAR NO DOMÍNIO ESPACIAL
- 6 FILTROS ESPACIAIS PARA NITIDEZ

MASCARAMENTO: FILTRAGEM LINEAR NO DOMÍNIO ESPACIAL

- Filtragem espacial utilizada para aplicações com imagens
- Descrição no domínio da frequência de filtros bidimensionais (2-D) é geralmente mais complexa do que os filtros unidimensionais (1-D)
- Filtragem espacial um vetor de pesos é usado para expressar o efeito do filtro em cada pixel da imagem
- O valor do pixel (x, y) de uma imagem processada, $g(x, y)$, é calculado como a soma dos produtos dos coeficientes do filtro e da imagem original

$$g(x, y) = \sum_{s=-a}^a \sum_{t=-b}^b w(s, t) f(x + s, y + t)$$

- As máscaras podem ser consideradas como a resposta truncada no domínio espacial dos filtros lineares 1-D

- Os filtros passa-baixa atenuam ou eliminam componentes de alta frequência de uma imagem, como bordas, textura e outros detalhes nítidos
 - Suavização, desfoque e redução de ruído
- Etapa de pré-processamento para remover detalhes sem importância
- Utilizados para unir pequenas lacunas em linhas e curvas (como uma técnica de interpolação)

Desvantagem: atenuam as bordas e alguns detalhes nítidos que geralmente são importantes em muitas aplicações.

- **Primeira máscara:**
- Cálculo da média dos valores de *pixel* em torno de um *pixel* para gerar o pixel correspondente
- Quanto maior a máscara se torna, maior a atenuação em componentes de alta frequência
- **Segunda máscara:**
- Neste filtro atribuímos um valor maior ao coeficiente central da máscara para enfatizar e acentuar a importância do pixel central
- Os valores atribuídos a outros coeficientes são muitas vezes inversamente proporcionais à sua distância do coeficiente central
- Essa máscara evitará parcialmente o efeito indesejado de desfoque

$$(1/9)^*$$

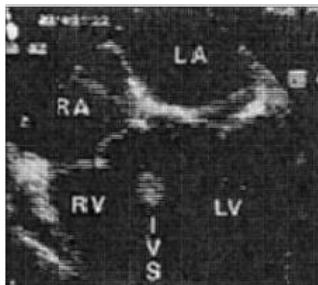
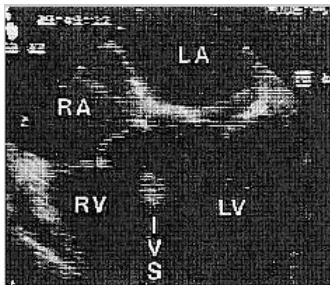
1	1	1
1	1	1
1	1	1

$$(1/16)^*$$

1	2	1
2	4	2
1	2	1

Exemplo: Implementação de um filtro passa-baixa no Matlab

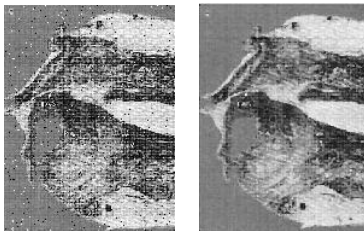
```
l=imread('image.jpg');  
l=rgb2gray(l);  
H=(1/9)*[1 1 1; 1 1 1; 1 1 1];  
smooth_image = imfilter(l,H);  
imshow(l);  
figure ,  
imshow(smooth_image);
```



- Esses são filtros estatísticos não-lineares que são frequentemente descritos no domínio do espaço
- Reduzem o ruído sem eliminar as bordas e outros conteúdos de alta frequência
- Executam as seguintes operações para encontrar cada valor de pixel na imagem processada:
 - 1) Todos os pixels na vizinhança do pixel na imagem original (identificados pela máscara) são inseridos em uma lista
 - 2) Essa lista é classificada em ordem crescente (ou decrescente).
 - 3) A mediana da lista ordenada (isto é, o pixel no meio da lista) é escolhida como o valor de pixel para a imagem processada.

Exemplo: Implementação de um filtro de média no Matlab

```
l=imread('no_noisy_image.jpg');  
l=rgb2gray(l);  
J=imnoise(l,'salt&pepper',.2);  
k=medfilt2(J);  
imshow(J);  
figure,  
imshow(k);
```



Vantagem sobre o filtro passa-baixa: reduz ruído sem reduzir o contraste de bordas.

COMPARAÇÃO MF x LP



- 1 INTRODUÇÃO E VISÃO GERAL
- 2 PROCESSAMENTO POR PONTOS
- 3 *Bit-Level Slicing*
- 4 EQUALIZAÇÃO DE HISTOGRAMA
- 5 MASCARAMENTO: FILTRAGEM LINEAR NO DOMÍNIO ESPACIAL
- 6 FILTROS ESPACIAIS PARA NITIDEZ

- Filtros de nitidez são usados para extrair e destacar detalhes finos de uma imagem e também para realçar alguns detalhes borrados
- Um uso típico de tais filtros é o desfocagem de uma imagem para fornecer informações de borda mais nítidas e visíveis
- Existem muitas aplicações para filtros de nitidez, incluindo imagens médicas, impressão eletrônica, inspeção industrial e orientação autônoma em sistemas militares

Tipos importantes de filtros para nitidez: filtros high-pass, filtros high-boost e filtros derivados.

FILTRO PASSA-ALTA

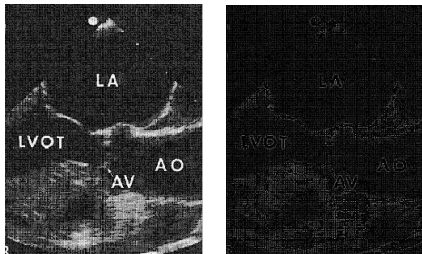
- Como nos filtros passa-baixas lineares, as máscaras usadas para filtragem de passagem alta não são nada além das aproximações truncadas da representação no domínio do espaço dos filtros passa-altas ideais típicos
- Como tal, nos filtros de passa-alta, a forma da resposta ao impulso deve ter coeficientes (+) próximos do centro e coeficientes (-) na extremidades.

$(1/9)^*$

-1	-1	-1
-1	8	-1
-1	-1	-1

Exemplo: Implementação de um filtro passa-alta no Matlab

```
l=imread('med_image.jpg');  
l=rgb2gray(l);  
H=(1/9)*[1 1 1; 1 8 1; 1 1 1];  
sharpened = imfilter(l,H);  
imshow(l);  
figure,  
imshow(sharpened);
```



Os filtros passa-alta eliminam todas as componentes importantes de baixa frequência que são necessários para uma imagem bem definida.

FILTROS *High-Boost* (HB)

- São uma extensão de filtros passa-alta, enquanto destacam as altas frequências, preservam alguns componentes de baixa frequência e evitam valores negativos de *pixels*
- São também chamados de filtros de ênfaseção
- Um filtro HB pode ser simplesmente definido como uma combinação ponderada da imagem original e da versão filtrada HP da imagem
- Nesta combinação, as componentes de alta frequência são destacadas mais que as de baixa frequência, ou seja

$$HB = (A - 1)original_image + HP_filtered_image$$

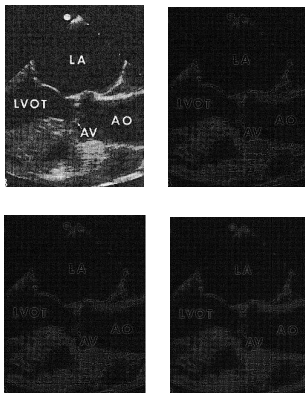
$$(A-1) \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} + 1/9 \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix} = 1/9 \begin{bmatrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

$$HB = (A - 1)original_image + HP_filtered_image$$

- Definindo $A > 1$, uma versão ponderada da imagem original é adicionada de volta às componentes de alta frequência
- Essas escolhas de A mantêm as componentes de baixa frequência perdidas na filtragem passa-alta pura e, portanto, produzem imagens aprimoradas que não apenas enfatizam as altas frequências, mas também mostram quase todos os componentes de baixa frequência que são frequentemente perdidos quando utilizados filtros HP padrão.

Exemplo: Implementação de um filtro HB no Matlab. Como resultado: imagem original, imagem filtrada com $A = 1.1$, imagem filtrada com $A = 1.15$ e imagem filtrada com $A = 1.2$.

```
l=imread('test.jpg');  
l=rgb2gray(l);  
A=1.2;  
H=(1/9)*[1 1 1;1 -8 1; 1 1 1];  
K=[0 0 0;0 1 0;0 0 0];  
HB=( (A-1).*K) + H;  
sharpened = imfilter(l,HB);  
imshow(l);  
figure ,  
imshow(sharpened);
```



Para a detecção de contornos e bordas é necessária a utilização de operadores que sejam sensíveis à mudança e consigam suprimir áreas de níveis de cinzento constante. Enquanto os filtros de média suprimem estruturas com valores altos, a detecção de contornos requer uma operação de filtragem que dê realce a mudanças nos valores de cada pixel e que anulem áreas com valores constante, sendo os operadores derivativos adequados para esta função.

Operadores de Roberts, Prewitt e Sobel.