



Filtragem Espacial

Guillermo Cámara-Chávez

Filtragem com Preservação de Bordas



- Filtros estatísticos passa-baixas podem suprimir detalhes importantes da imagem
 - linhas finas,
 - cantos de objetos.
- Isso acontece porque não é levado em consideração se o pixel se encontra sobre uma borda

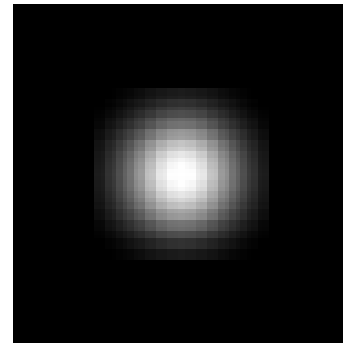
Filtragem com Preservação de Bordas



$$GB[I]_p = \sum_{q \in S} G_{\sigma}(\| \mathbf{p} - \mathbf{q} \|) I_q$$



normalized
Gaussian function

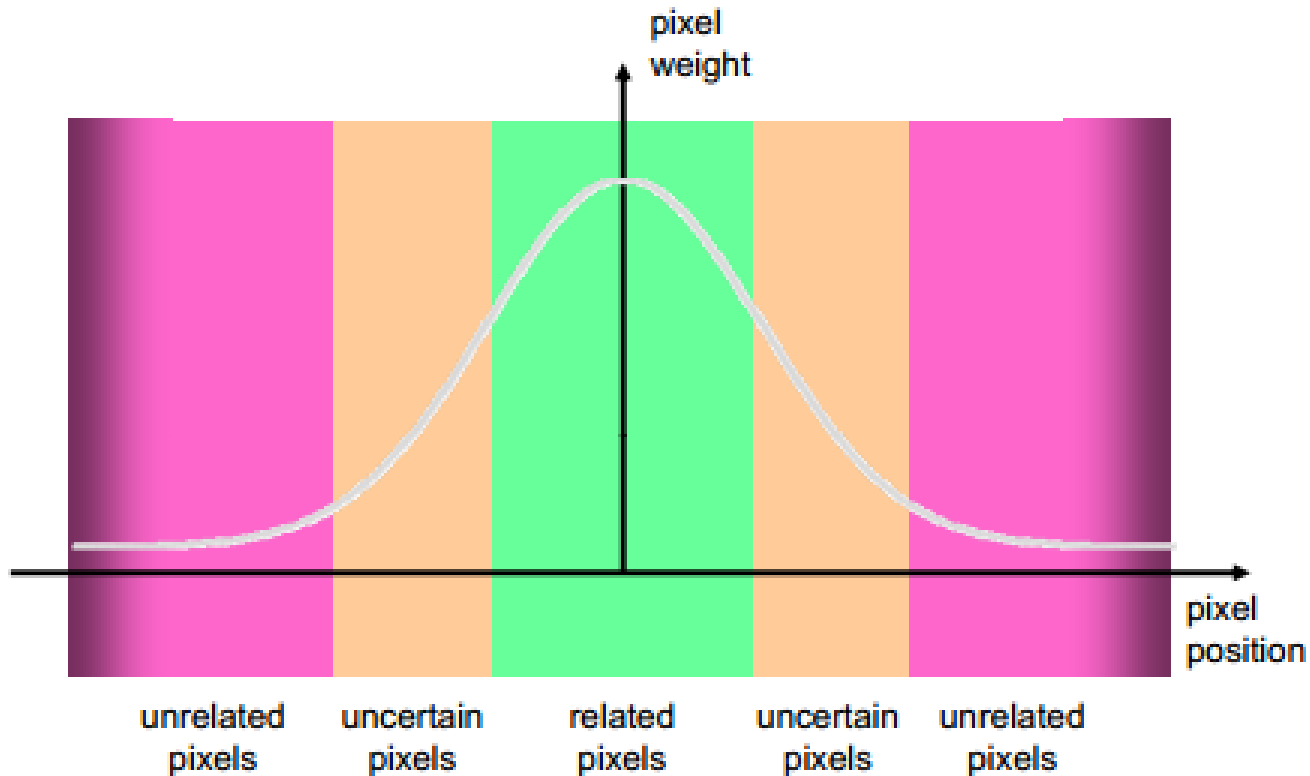


Filtragem com Preservação de Bordas



Gaussian Profile

$$G_{\sigma}(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{x^2}{2\sigma^2}\right)$$



Filtragem com Preservação de Bordas



Spatial Parameter

$$GB[I]_p = \sum_{q \in S} G_{\sigma}(\| \mathbf{p} - \mathbf{q} \|) I_q$$

σ
size of the window



input



small σ



limited smoothing



large σ



strong smoothing

Filtragem com Preservação de Bordas



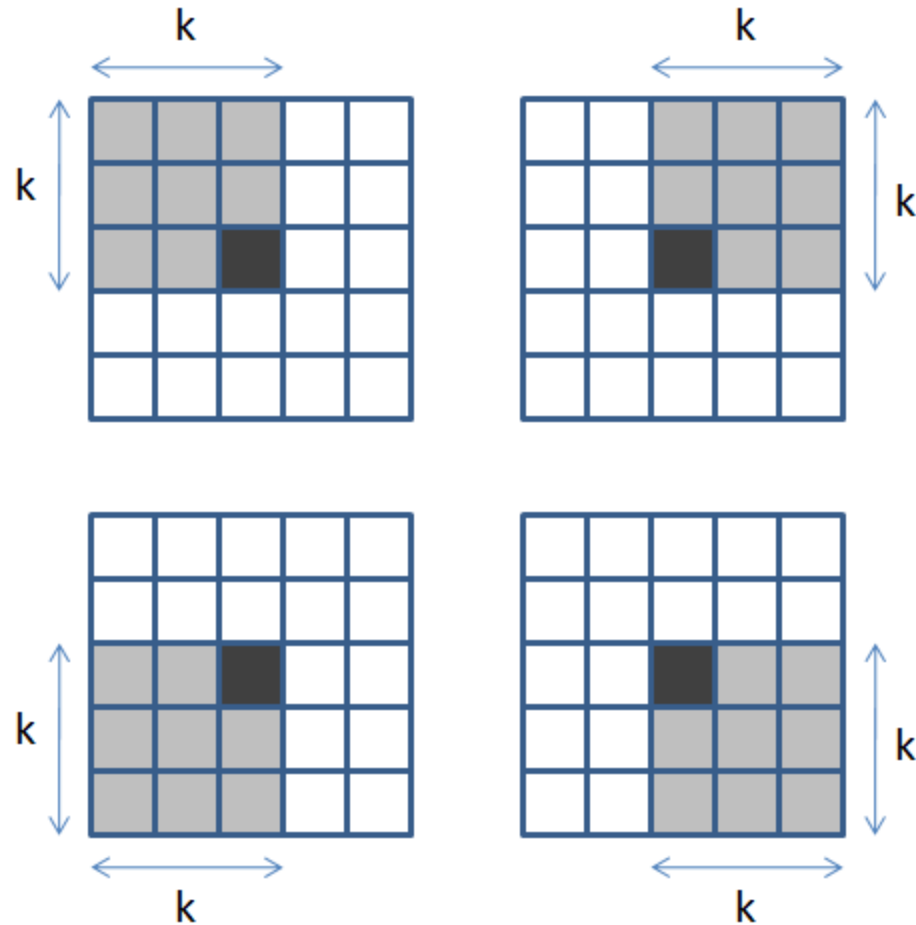
- Como definir o valor de σ
 - Depende da aplicação
 - Estratégia comum: proporcional ao tamanho da imagem
 - Exemplo, 2% da diagonal da imagem

Filtragem com Preservação de Bordas



- Máscaras de Kuwahara
 - Uma das primeira técnicas com preservação de bordas
 - Foi proposto em 1976
 - O filtro considera uma região quadrada de dimensão $(2k-1) \times (2k-1)$

Filtragem com Preservação de Bordas



Filtragem com Preservação de Bordas



- Máscaras de Kuwahara
 - A variância dos níveis de cinza para cada janela
 - O valor do pixel (x,y) é substituído pela média dos níveis de cinza da janela com menor variância

Filtragem com Preservação de Bordas



$K = 2$



$K = 3$

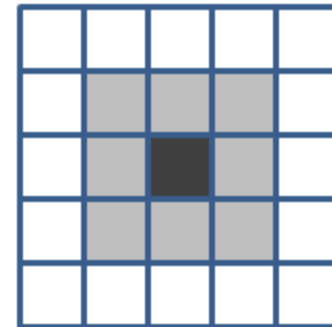
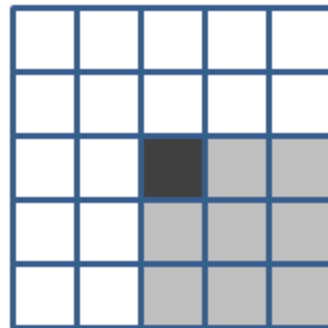
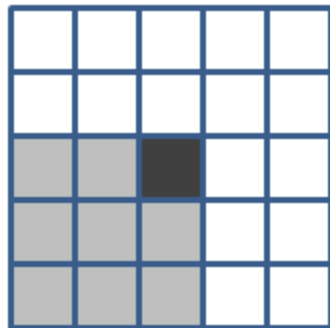
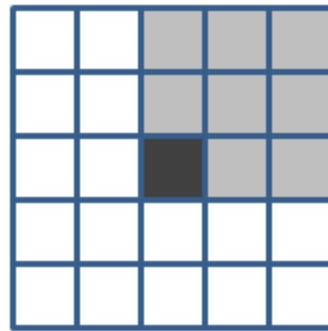
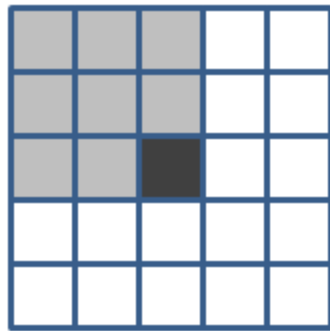


$K = 4$

Filtragem com Preservação de Bordas



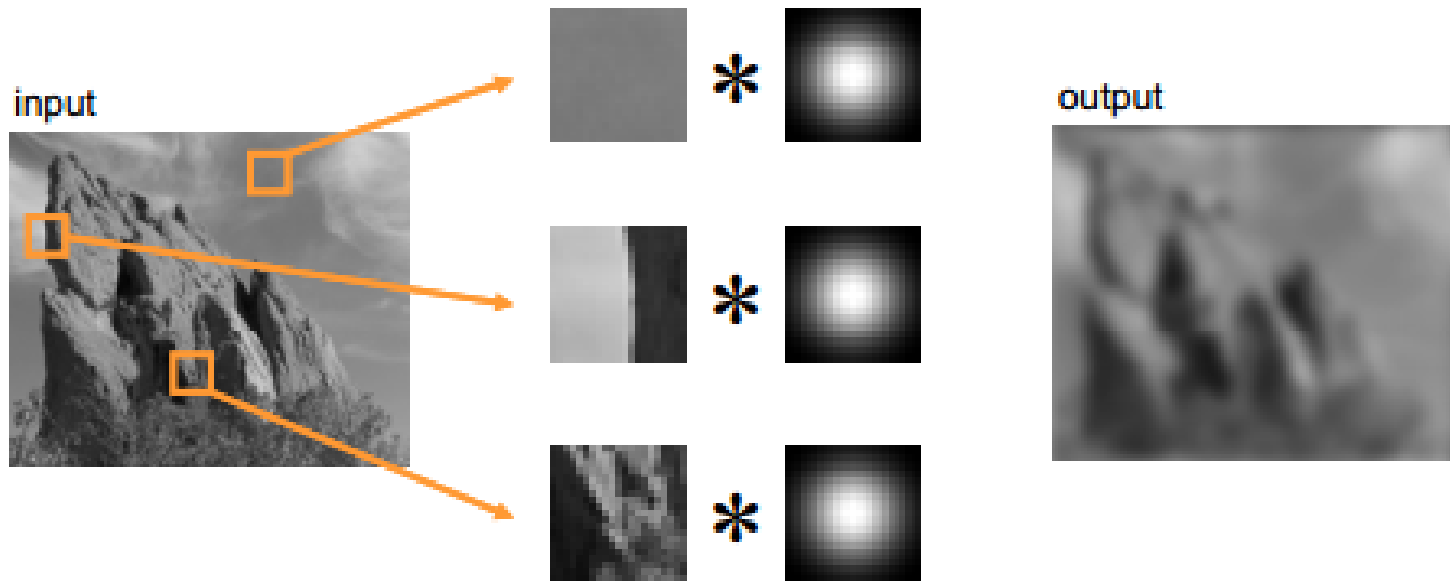
- Máscaras de Tomati e Tsuji



Filtragem com Preservação de Bordas



- O borramento vem da média dos pixels de regiões próximas (bordas).

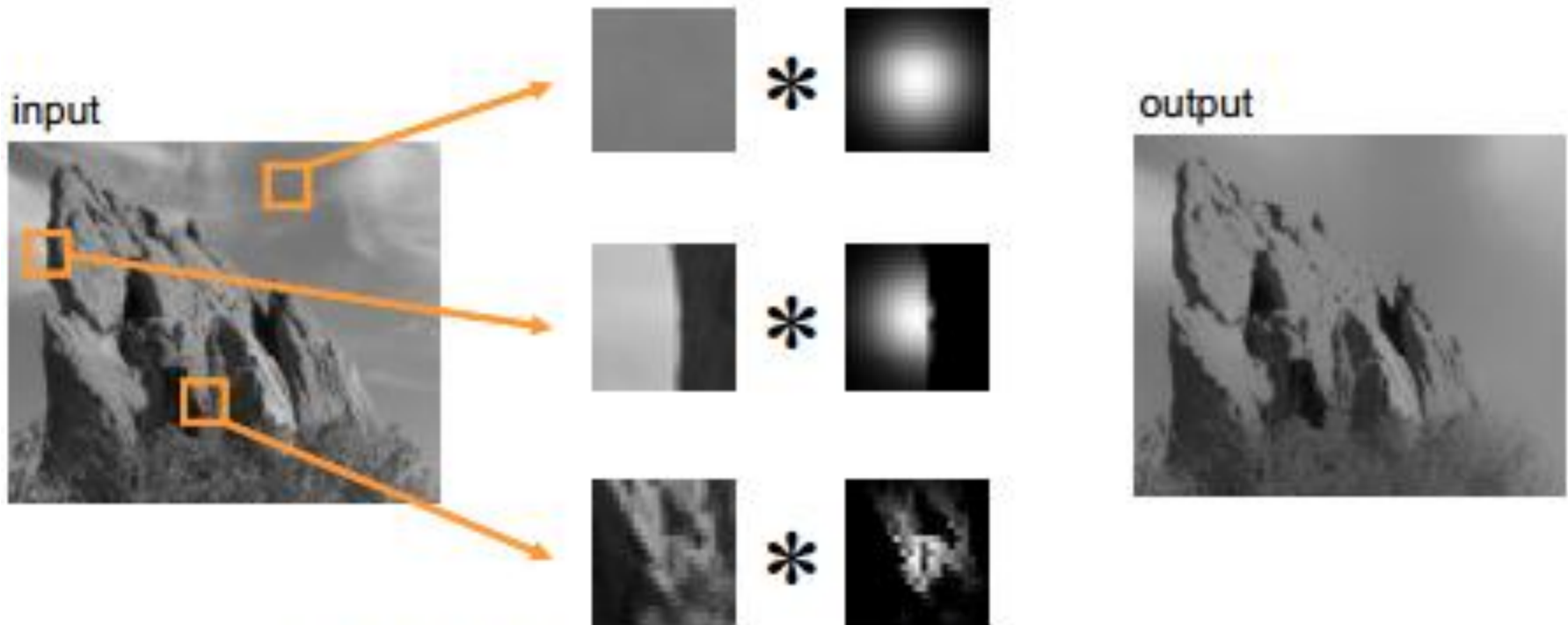


Same Gaussian kernel everywhere.

Filtragem com Preservação de Bordas



- Filtragem bilateral



The kernel shape depends on the image content.

Filtragem Bilateral



- A mesma idéia: média ponderada dos pixels

$$BF[I]_p = \overset{\text{new}}{\frac{1}{W_p}} \sum_{q \in S} \overset{\text{not new}}{G_{\sigma_s}(\|p - q\|)} \overset{\text{new}}{G_{\sigma_r}(|I_p - I_q|)} I_q$$

The diagram illustrates the components of the bilateral filter equation:

- normalization factor**: Indicated by a pink arrow pointing from the pink box $\frac{1}{W_p}$ to the text.
- space weight**: Indicated by an orange arrow pointing from the orange box $G_{\sigma_s}(\|p - q\|)$ to the text. Below the text is a 2D Gaussian kernel visualization.
- range weight**: Indicated by a blue arrow pointing from the blue box $G_{\sigma_r}(|I_p - I_q|)$ to the text. Below the text is a 1D Gaussian kernel visualization along the intensity axis I .

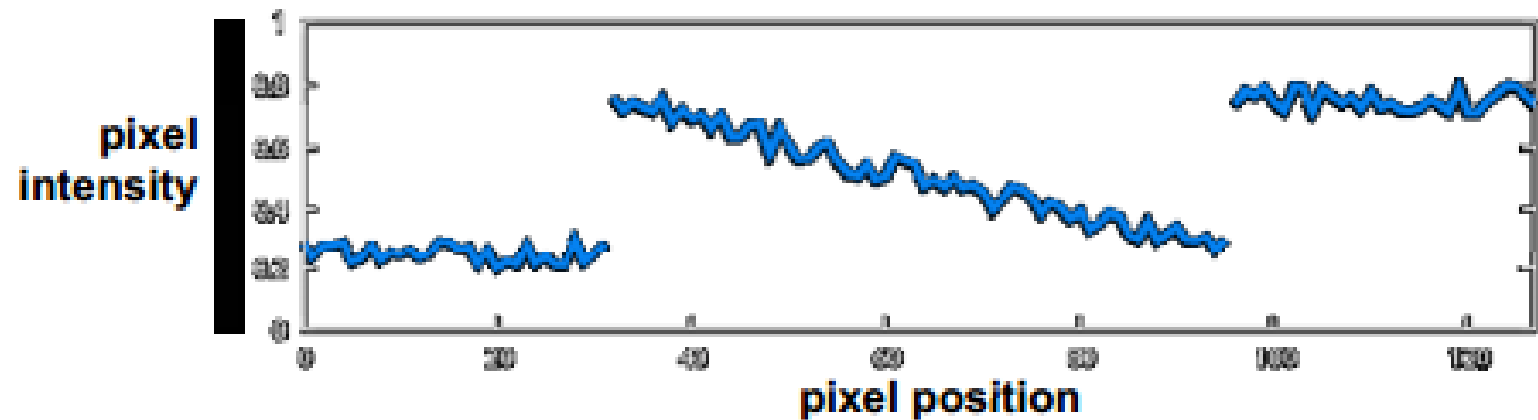
Filtragem Bilateral



- 1D image = line of pixels



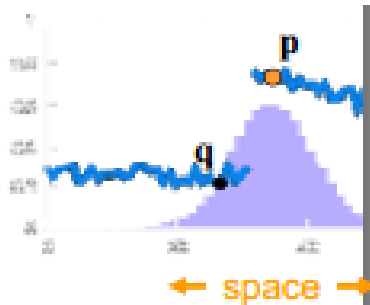
- Better visualized as a plot



Filtragem Bilateral

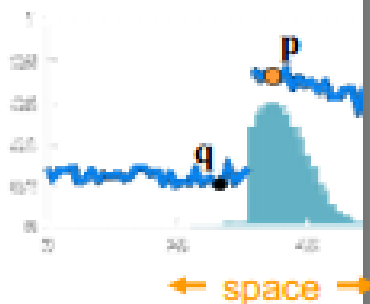


Gaussian blur



Bilateral filter

[Aurich 95, Smith 97]



[Tomasi 98]

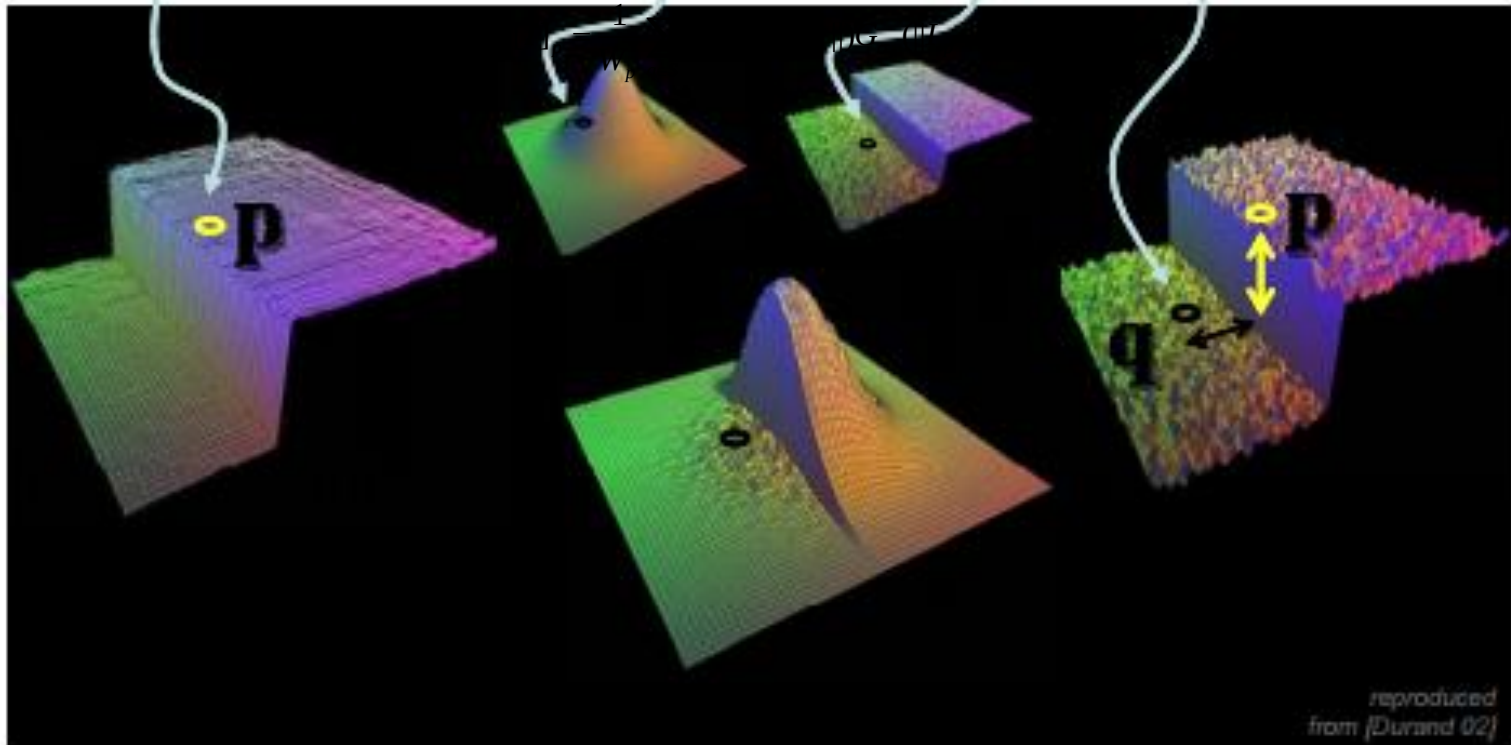
$$GB[I]_p = \sum_{q \in S} \underbrace{G_{\sigma}(\|p - q\|)}_{\text{space}} I_q$$

$$BF[I]_p = \underbrace{\frac{1}{W_p}}_{\text{normalization}} \sum_{q \in S} \underbrace{G_{\sigma_s}(\|p - q\|)}_{\text{space}} \underbrace{G_{\sigma_r}(\|I_p - I_q\|)}_{\text{range}} I_q$$

Filtragem Bilateral



$$BF[I]_p = \frac{1}{W_p} \sum_{q \in S} G_{\sigma_r}(\|p - q\|) G_{\sigma_i}(\|I_p - I_q\|) I_q$$



Filtragem Bilateral



$$BF[I]_p = \frac{1}{W_p} \sum_{q \in S} G_{\sigma_s}(\|p - q\|) G_{\sigma_r}(\|I_p - I_q\|) I_q$$

$$W_p = \sum_{q \in S} G_{\sigma_s}(\|p - q\|) G_{\sigma_r}(\|I_p - I_q\|)$$

Filtragem Bilateral



- Como definir os parâmetros
 - Parâmetro espacial: proporcional ao tamanho da imagem
 - Exemplo, 2% da diagonal da imagem
 - Parâmetro de intervalo de intensidade (*range parameter*): proporcional à amplitude das bordas
 - Exemplo, média ou mediana dos gradientes da imagem

Filtragem Bilateral



$\text{Sigma}_p = 7$

$\text{Sigma}_q = 9$

window = 3

The Minimal Mean Square Error Filter



$$g(i, j) = \left(1 - \frac{\sigma_n^2}{\sigma_L^2}\right) f(i, j) + \frac{\sigma_n^2}{\sigma_L^2} \mu_L$$

- $f(i, j)$: imagem observada
- L : vizinhança de $N_h \times N_h$
- μ_L : média local
- σ_L^2 : variância local
- σ_n^2 : variância ruído

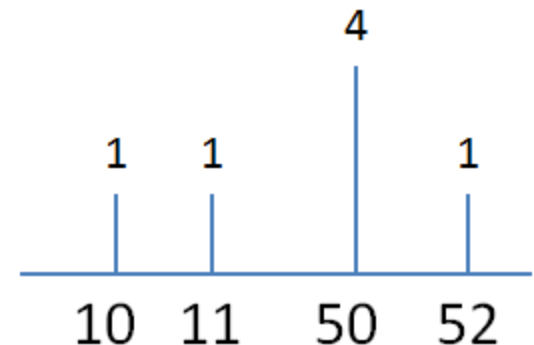
Filtros Não Lineares



- Filtro da Moda

- Consiste em substituir o pixel central P de $w \times w$ pelo NC mais freqüente na janela $w \times w$: a “moda”

$$W_{3 \times 3} = \begin{bmatrix} 11 & 50 & 50 \\ 10 & 52 & 50 \\ 10 & 11 & 50 \end{bmatrix} \rightarrow \boxed{50}$$



A moda = 50 $\Rightarrow p(5) = 50$

Filtros Não Lineares



- Filtro da ordem k
 - Associam o k -ésimo valor dos NC's de uma janela $w \times w$ arranjados em ordem crescente.
 - Sejam :
 $p(1), p(2), \dots, p(E)$ os NC's dos elementos de w
 $z(1), z(2), \dots, z(E)$ os mesmos NC's arranjados em ordem crescente

Filtros Não Lineares



- Casos especiais:
 - $k = 1$ -> filtro min
 - $k = E$ -> filtro max
 - $k = (E+1)/2$ -> filtro da mediana
- Resultados apresentados na literatura mostram que os filtros de ordem maior que o filtro da mediana alargam regiões claras da imagem

Filtros Não Lineares



- Filtros MIN MAX
 - Generalizações de operadores de compressão e expansão para aplicação em imagens
 - São também indicados para remoção de ruído do tipo sal ou pimenta

Filtros Não Lineares



```
function nimg = PepperNoise(img, p3)
    if nargin < 2
        p3 = 0.05;
    end
    nimg = img;
    x = rand(size(img));
    nimg(x < p3/2) = 0; % valor mínimo
```

Filtros Não Lineares



```
function nimg = MAX(img, tam)
    [fil, col] = size(img);
    nimg = img;
    tam_janela = floor(tam/2);
    for i = tam_janela + 1 : fil - tam_janela
        for j = tam_janela + 1 : col - tam_janela
            janela = img(i-tam_janela : i+tam_janela, ...
                          j-tam_janela:j+tam_janela);
            nimg(i,j) = max(janela(:));
        end
    end
end
```

Filtros Não Lineares



Filtros Não Lineares



- Filtros não lineares em matlab:
 - `nlfilter(f, [m n], 'sliding', fun)`
 - `colfilt(f, [m n], 'sliding', fun)`
 - `f`: imagem
 - `[m n]`: tamanho da janela
 - `fun`: função usada na janela

Filtros Não Lineares



- Padding (preenchimento para tratar problemas das fronteiras da imagem)
 - `padarray(f, [a b], tipo, pos)`
 - `f`: imagem
 - `[a b]`: numero de linhas e colunas a ser acrescentadas
 - `tipo`: valores preenchidos (`'symmetric'`, `'replicate'`, `'circular'`)
 - `pos`: a partir de onde são acrescentado os valores (`'pre'`, `'post'`, `'both'`)

Filtros Não Lineares



```
function nimg = MAX2(img, tam)
nlin = floor(tam/2);
ncol = floor(tam/2);
nimg = colfilt(img, [tam tam], 'sliding', @max);
```


Filtros Não Lineares



- Filtro da mediana
 - Consiste em substituir o pixel central da janela $w \times w$ pelo valor de cinza média de da janela, cujos elementos foram ordenados.
 - Preserva, sem borrar, os contornos de regiões maiores que a janela usada.
 - Elimina descontinuidades que são menores que o tamanho da janela.

Filtros Não Lineares



$$W_{3 \times 3} = \begin{matrix} 11 & 50 & 49 \\ 10 & 80 & 52 \\ 10 & 51 & 50 \end{matrix}$$

em ordem crescente:

10 10 11 49 50 50 51 52 80

↑

Mediano = 50 $\rightarrow$ $p(5) = 50$

Filtros Não Lineares



- *ordfilt2(f, n, vizinhança]*) substitui o pixel avaliado pelo n-ésimo elemento ordenado dentro de cada janela

```
img = imread('lenna.png');  
img_ruído = imnoise(img, 'salt & pepper');  
nimg = ordfilt2(img, 1, ones(3,3)); % filtro min  
nimg = ordfilt2(img, 9, ones(3,3)); % filtro max  
nimg = ordfilt2(img, 5, ones(3,3)); % filtro mediana
```

Filtros Não Lineares



- *medfilt2*(*A*, [*m n*]) filtro da médiana em 2-D, onde *A* é a imagem e [*m n*] é a dimensão da máscara. Por *default* o tamanho da máscara é de 3x3

```
img = imread('lenna.png');
```

```
img_ruido = imnoise(img, 'salt & pepper');
```

```
nimg = medfilt2(img);
```

Filtros Não Lineares



Filtros Não Lineares



- Filtro sigma
 - A suavização é alcançada através da média com apenas aqueles elementos que têm seus NC's dentro da faixa dos 2-sigmas.
 - O valor é fixado pelo pixel central P da janela $w \times w$
 - Para uma distribuição gaussiana 95.5% das amostras estão dentro da faixa dos 2σ

Filtros Não Lineares



- Procedimento:
- Considere uma janela de imagem $w \times w$ cujo pixel central P tem $NC = p(i)$
 1. Calcule a faixa dos
 $2 \sigma \ (p(i) - T, p(i) + T)$, onde $T = 2\sigma$
 2. Calcule a média dos elementos dentro da faixa do
 2σ
 3. Faça $p(i) = \text{média dos elementos}$

Filtros Não Lineares



$$W_{5 \times 5} = \begin{matrix} 10 & 10 & 50 & 50 & 50 \\ 10 & 10 & 50 & 50 & 50 \\ 10 & 10 & 80 & 50 & 50 \\ 10 & 10 & 50 & 50 & 50 \\ 10 & 10 & 50 & 50 & 50 \end{matrix}$$

$$P = p(i) = 80$$

$$\text{Considerando } \sigma = 20 \rightarrow T = 2 \times 20 = 40$$

Filtros Não Lineares



- 1) Faixa dos $2\sigma \rightarrow (80-40, 80+40) = (40, 120)$
- 2) Cálculo da média dos 2σ : $M = (14 \times 50 + 80) / 15 = 52$
- 3) $p(i) = 52$

Filtros Não Lineares



Suponha que $P = 200$ (ruído sal e pimenta) para

$$\sigma = 20 \rightarrow T = 2 \times 20 = 40$$

1) Faixa dos $2\sigma \rightarrow (200-40, 200+40) = (160, 240)$

2) $M = 200$

3) $p(i) = 200!$

Como contornar o problema?

Filtros Não Lineares



Substitua o item 3) por:

$$3) p(i) = \begin{cases} \text{média dos } 2\sigma, & \text{se } R > k \\ \text{média com 8 vizinhos imediatos,} & \text{se } R \leq k \end{cases}$$

onde:

R = número de pixels dentro da faixa 2σ

k = constante pré-definida para a janela:

$$W_{7 \times 7} \rightarrow k < 4 \quad w_{5 \times 5} \rightarrow k < 3$$

Filtros Não Lineares



- Filtro sigma polarizado
 - Variação do filtro sigma
 - Aguça as bordas tipo rampa
 - Aumenta o contraste de detalhes tênues.
 - A polarização é introduzida pelo cálculo da média (faixa superior e faixa inferior do 2σ)
 - Faixa superior: $(p(i), p(i) + T)$
 - Faixa inferior: $(p(i), p(i) - T)$

Filtros Não Lineares



- As diferenças absolutas entre $p(i)$ e média superior e entre $p(i)$ e a média inferior são calculadas.
- O pixel central P tem seu NC substituído pela média que tenha a menor diferença absoluta

$$W_{5 \times 5} = \begin{matrix} 10 & 10 & 40 & 50 & 50 \\ 10 & 10 & 40 & 50 & 50 \\ 10 & 10 & 40 & 50 & 50 \\ 10 & 10 & 40 & 50 & 50 \\ 10 & 10 & 40 & 50 & 50 \end{matrix} \quad \text{Para } \sigma = 20 \rightarrow T = 40$$

Filtros Não Lineares



Faixa superior: (40, 80)

Faixa inferior: (0, 40)

Média superior: $p(i) = (5 \times 40 + 10 \times 50) / 15$

Média inferior: $p(i) = (5 \times 40 + 10 \times 10) / 15$

$MS = 47; MI = 20; \rightarrow p(i) = 47$

Realce (Sharpening)



- Tende a realçar mudanças abruptas de níveis de cinza na imagem.
- Esta operação é conhecida como “*edge enhancement*”, “*edge crispening*” ou “*unsharp masking*”.

Realce (Sharpening)



- Alguns exemplos de máscaras:

$$M_1 = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

$$M_2 = \begin{bmatrix} 1 & -2 & 1 \\ -2 & 5 & -2 \\ 1 & -2 & 1 \end{bmatrix}$$

$$M_3 = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

Realce (Sharpening)



Conv. com a
máscara M_1



Conv. com a
máscara M_2

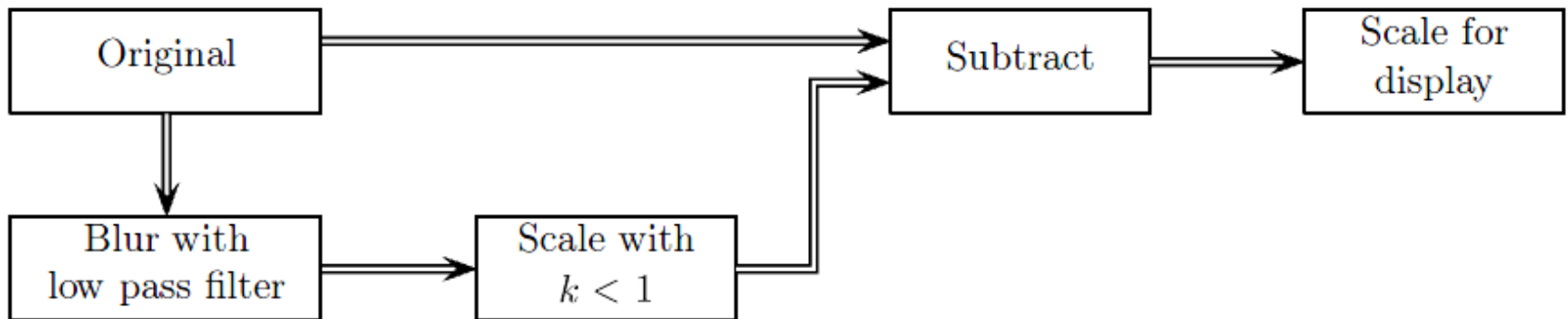


Conv. com a
máscara M_3

Realce (Sharpening)



- Outra forma de realçar uma imagem é através do seguinte método



Realce (Sharpening)



```
img = double(imread('lenna.png'));  
f = fspecial('average');  
fl = imfilter(img, f, 'same', 'replicate');  
fu = img - fl / 1.5  
imshow(fu/70);
```

Realce (Sharpening)



- A aplicação de um operador de “unsharp masking”, está constituído por uma seqüência de duas operações
 1. Obter uma versão suavizada da imagem
 2. Combinação entra a imagem original e o resultado da primeira operação

Realce (Sharpening)



$$f(x,y) = f_{\text{lowpass}}(x,y) + f_{\text{highpass}}(x,y)$$

O filtro da média é um filtro passa-baixas

$$f(x,y) = f_{\text{mean}}(x,y) + f_{\text{highpass}}(x,y)$$

$$f_{\text{highpass}}(x,y) = f(x,y) - f_{\text{mean}}(x,y)$$

Realce (Sharpening)



```
f= imread('moon.png');  
w = [0 1 0; 1 -4 1; 0 1 0];  
f2 = im2double(f);  
g2=imfilter(f2,w,'replicate');  
g=f2-g2;
```

Realce (Sharpening)



Realce (Sharpening)



- Em alguns casos é interessante realçar as altas freqüências sem atenuar as baixas.
- Nesta situação é interessante adicionar uma quantidade proporcional de altas freqüências à imagem original adicionando um termo de ganho G

$$f_{\text{highpass}}(x,y) = (1+G).f(x,y) - G.f_{\text{mean}}(x,y), \text{ onde } 0 \leq G \leq 2$$

Realce (Sharpening)



- A máscara que implementa o método anterior é:

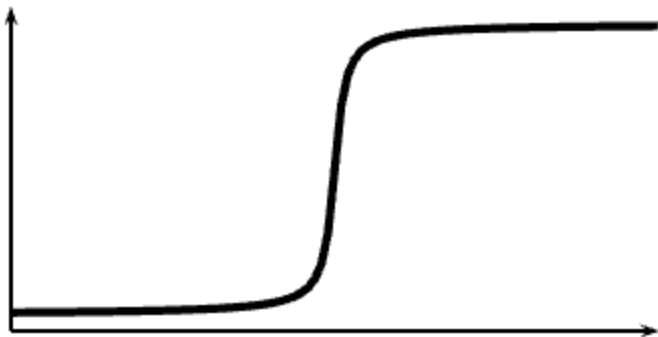
$$M = \begin{matrix} -G/9 & -G/9 & -G/9 \\ -G/9 & (9+8G)/9 & -G/9 \\ -G/9 & -G/9 & -G/9 \end{matrix}$$

Realce (Sharpening)

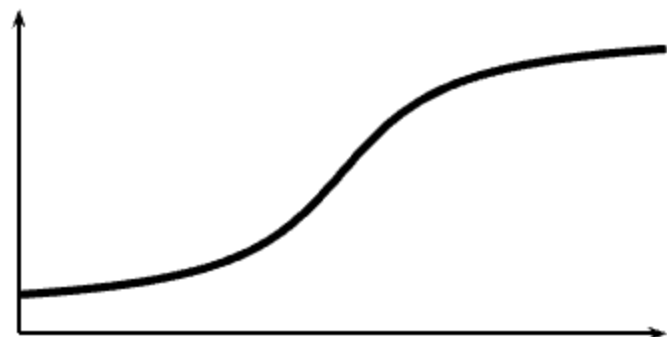


$$G = 2$$

Realce (Sharpening)

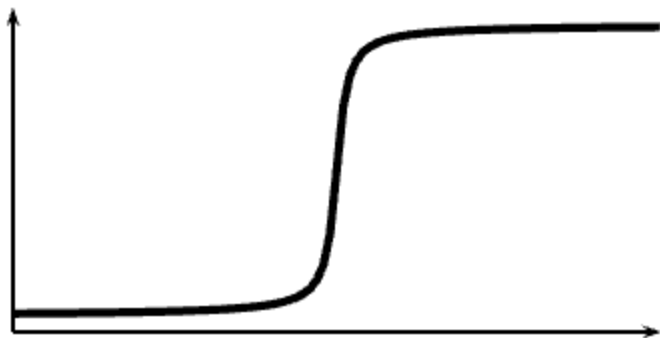


(a) Pixel values over an edge



(b) The edge blurred

Realce (Sharpening)



(a) Pixel values over an edge



(c) $(a) - k(b)$

Sinal realçado

Realce (Sharpening)



```
img = imread('lenna.png');  
mask = fspecial('unsharp');  
nimg = imfilter2(img, mask, 'same');  
imshow(nimg);
```

Convolução / Correlação



CORRELAÇÃO E CONVOLUÇÃO ESPACIAIS

- Correlação é o processo de mover uma máscara de filtro sobre uma imagem e computar a soma de produtos em cada posição, exatamente como explicado anteriormente.

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

- A convolução difere da correlação pela rotação do filtro de 180°.

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

- A fig. ilustra a diferença entre essas duas operações para uma função 1D.

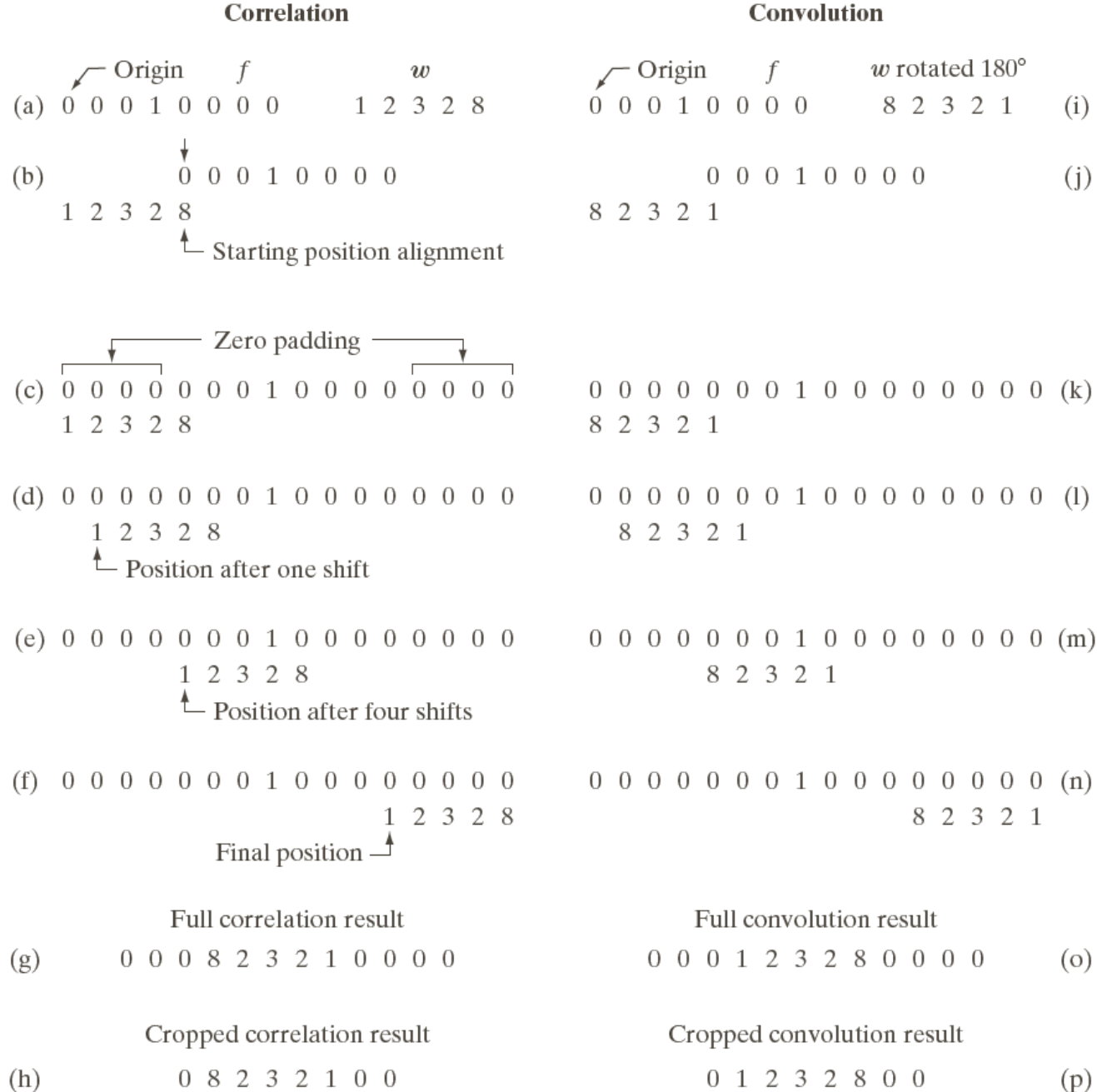


FIGURE 3.29 Illustration of 1-D correlation and convolution of a filter with a discrete unit impulse. Note that correlation and convolution are functions of *displacement*.

Convolução / Correlação

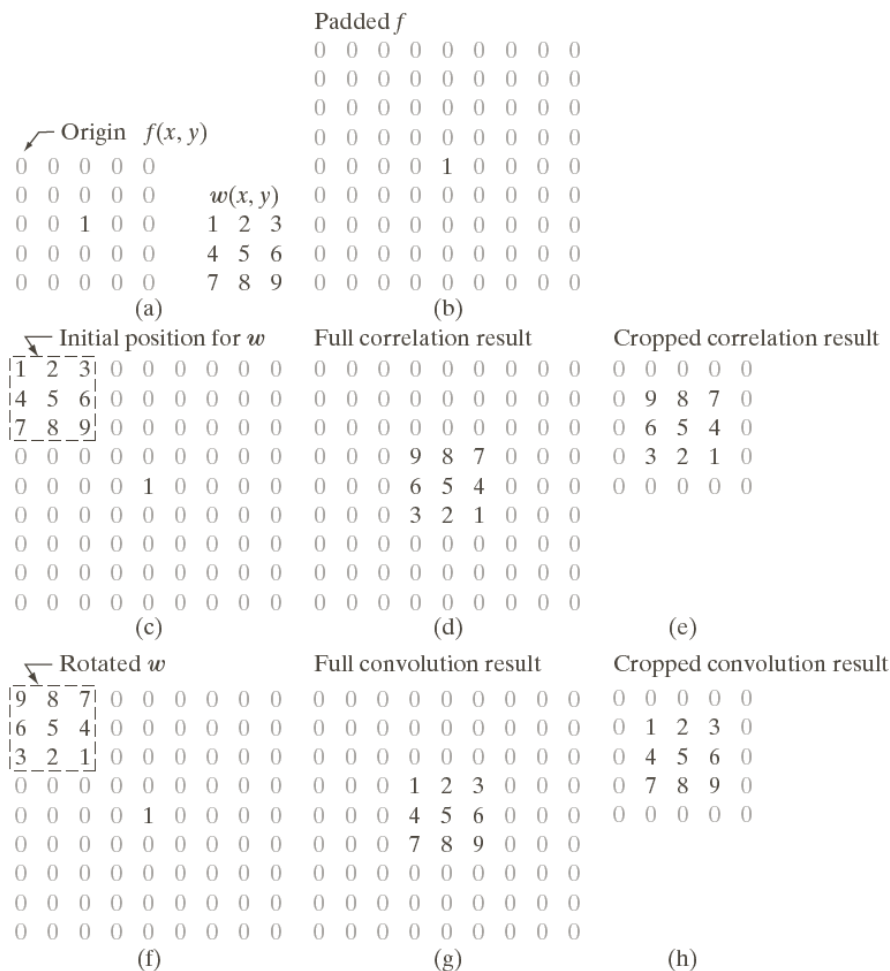


FIGURE 3.30
Correlation (middle row) and convolution (last row) of a 2-D filter with a 2-D discrete, unit impulse. The 0s are shown in gray to simplify visual analysis.

Correlação (linha do meio) e convolução (última linha) de um filtro 2D com um impulso unitário discreto 2D. Os 0s são mostrados em cinza para facilitar a análise visual.

Convolução / Correlação



- Observações:
 - Usar operações de correlação ou convolução para a filtragem espacial é uma questão de preferência.
 - O importante é escolher uma máscara com os coeficientes adequados para que o resultado esperado seja obtido.
 - Para uma máscara com os valores de coeficientes simétricos os resultados da correlação e convolução coincidem (filtros isotrópicos, ou invariantes a rotação).

Convolução / Correlação



- Observações:
 - Finalmente, é comum encontrar termos como filtro de convolução, máscara de convolução ou kernel de convolução na literatura de processamento de imagens, denotando um filtro espacial, sem necessariamente significar que o filtro seja usado para uma verdadeira convolução.
 - A expressão “convolver uma máscara com uma imagem” é comumente usada para denotar um processo de correlação.

Métricas de Qualidade em Imagens



- Uma imagem pode sofrer degradações durante o processo de aquisição, transmissão ou processamento.
- Métricas de qualidade podem ser utilizadas para avaliar a similaridade entre duas imagens
- Considerando duas imagens, ***f*** e ***g***, ambas com dimensões $M \times N$

Métricas de Qualidade em Imagens



- Erro máximo (ME, do inglês, *Maximum Error*)

$$ME = \max | f(x, y) - g(x, y) |$$

- Quanto menor essa métrica, melhor a nova imagem se aproxima da original
- É bastante sensível ao ruído ou a variações locais nas imagens

Métricas de Qualidade em Imagens



- Erro médio absoluto (MAE, do inglês, *Mean Absolut Error*)

$$MAE = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} |f(x, y) - g(x, y)|$$

— Quanto menor essa métrica, melhor a nova imagem

Métricas de Qualidade em Imagens



- Erro médio quadrático (MSE, do inglês, *Mean Square Error*)

$$MSE = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} (f(x, y) - g(x, y))^2$$

— Quanto menor essa métrica, melhor a nova imagem

Métricas de Qualidade em Imagens



- Raiz do erro médio quadrático (RMSE, do inglês, Root *Mean Square Error*)

$$MSE = \sqrt{\frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} (f(x, y) - g(x, y))^2}$$

— Uma variação do erro mínimo quadrático

Métricas de Qualidade em Imagens



- Erro médio quadrático normalizado (NMSE, do inglês, *Normalized Mean Square Error*)

$$NMSE = \frac{\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} (f(x, y) - g(x, y))^2}{\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} (f(x, y))^2}$$

- Valores variam entre 0 e 1, o que torna possível a avaliação de imagens com dimensões diferentes

Métricas de Qualidade em Imagens



- Relação Sinal-Ruído de Pico (PSNR, do inglês, *Peak Signal to Noise Ratio*)

$$PSNR = 10 \log_{10} \frac{(L_{\max})^2}{RMSE} = 10 \log_{10} \frac{MNL_{\max}^2}{\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} (f(x, y) - g(x, y))^2}$$

— Avalia a diferença global entre duas imagens

Métricas de Qualidade em Imagens



- L_{max} é o valor máximo de intensidade cinza, tipicamente $L_{max} = 255$
- Quanto maior esse métrica, melhor a nova imagem
- Relação sinal-ruído (SNR, do inglês, Signal to Noise Ratio)

$$SNR = 10 \log_{10} \frac{\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} (f(x, y))^2}{\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} (f(x, y) - g(x, y))^2}$$

Métricas de Qualidade em Imagens



- Covariância

$$\sigma_{fg} = \frac{\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} (f(x, y) - \mu_f)(g(x, y) - \mu_g)}{MN}$$

onde μ_f e μ_g representam o nível de cinza média nas imagens ***f*** e ***g***

— está relacionada com a presença, ou não, de redundância de informação entre duas imagens

Métricas de Qualidade em Imagens



- Coeficiente de correlação
 - O valor varia entre -1 a 1, valores próximos de 0 representam um relacionamento linear mais fraco

$$\sigma_{fg} = \frac{\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} (f(x, y) - \mu_f)(g(x, y) - \mu_g)}{\sqrt{\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} (f(x, y) - \mu_f)^2 \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} (g(x, y) - \mu_g)^2}}$$

Exercícios Propostos



1. Implementar o método de realce de imagens, onde $f_{\text{highpass}}(x,y) = (1+G).f(x,y) - G.f_{\text{mean}}(x,y)$. Não utilizar a máscara apresentada em aula