

Processamento Digital de Imagens

Realce de Imagens

Eduardo A. B. da Silva

Programa de Engenharia Elétrica - COPPE/UFRJ

Laboratório de Sinais, Multimídia e Telecomunicações
eduardo@smt.ufrj.br

Sergio L. Netto

Programa de Engenharia Elétrica - COPPE/UFRJ

Laboratório de Sinais, Multimídia e Telecomunicações
sergioln@smt.ufrj.br

Abril de 2017





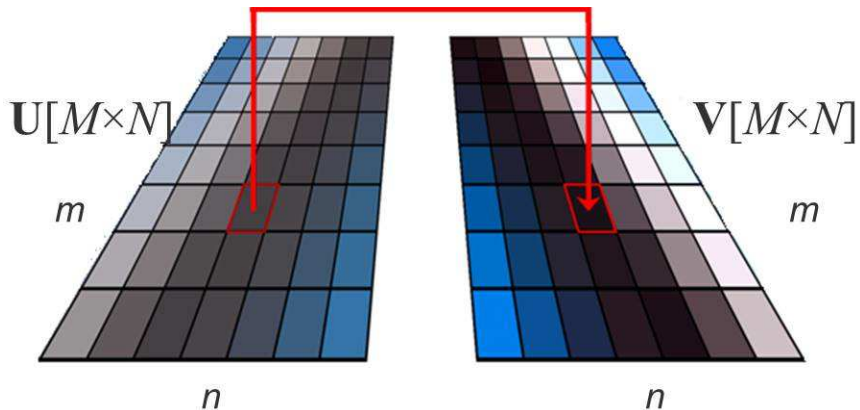
Sumário

① Realce de Imagens

- Operações Pontuais
- Operações Pontuais
- Modelagem de Histogramas
- Operações Espaciais
- Mapeamento Inverso de Contraste e Escalamento Estatístico
- Interpolação e Zoom
- Operações no Domínio das Transformadas
- Filtragem Homomórfica
- Realce de Imagens Multi-Espectrais
- Falsa Cor e Pseudo-Cor

Operações Pontuais

Não usam memória;

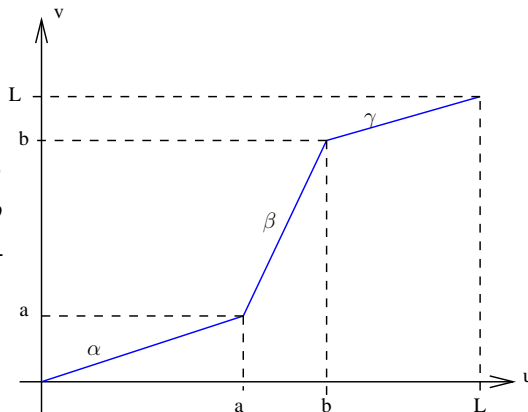


Operações Pontuais

Contrast Stretching

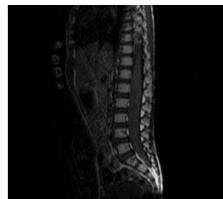
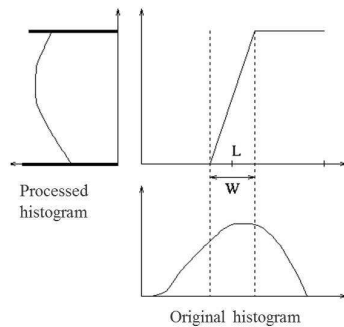
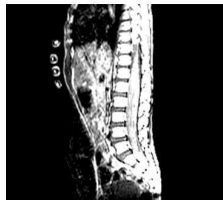
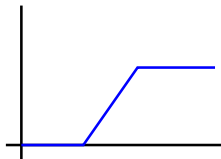
$$0 \leq x(m, n) \leq L$$

$$v = \begin{cases} \alpha u, & 0 \leq u < a \\ \beta(u - a) + va, & a \leq u < b \\ \gamma(u - b) + vb, & b \leq u \leq L \end{cases}$$

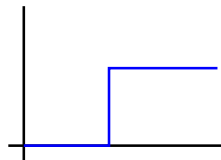




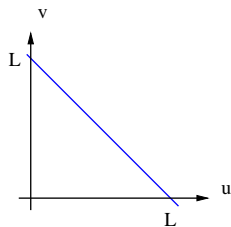
Clipping: $\alpha = \gamma = 0$



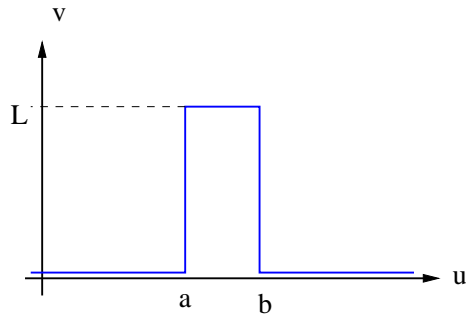
Thresholding: $\alpha = \gamma = 0; a = b = t$



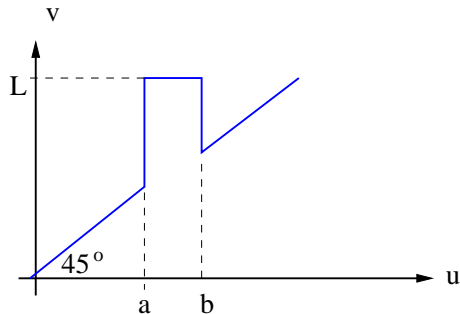
Negativo Digital:



Intensity Slicing:

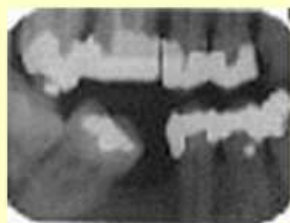


(sem background)



(com background)

Gray Level Slicing Effects - Examples



Original image

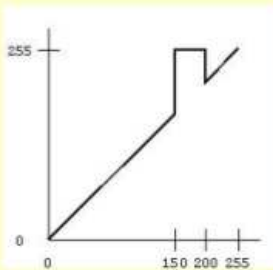
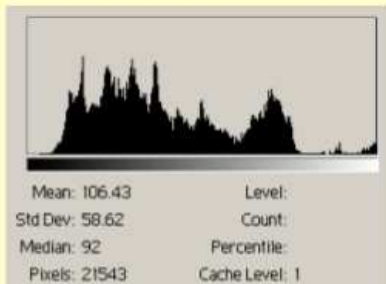
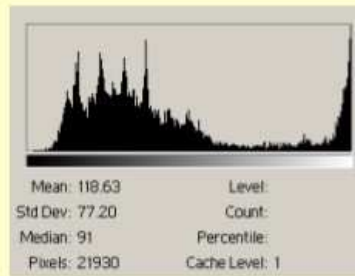


Image after slicing



Histogram Before



Histogram after slicing

Extração de Bits:

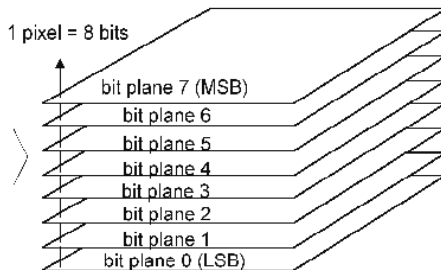
$$u = k_1 2^{B-1} + k_2 2^{B-2} + \dots + k_n 2^{B-n} + \dots + k_{B-1} 2 + k_B$$

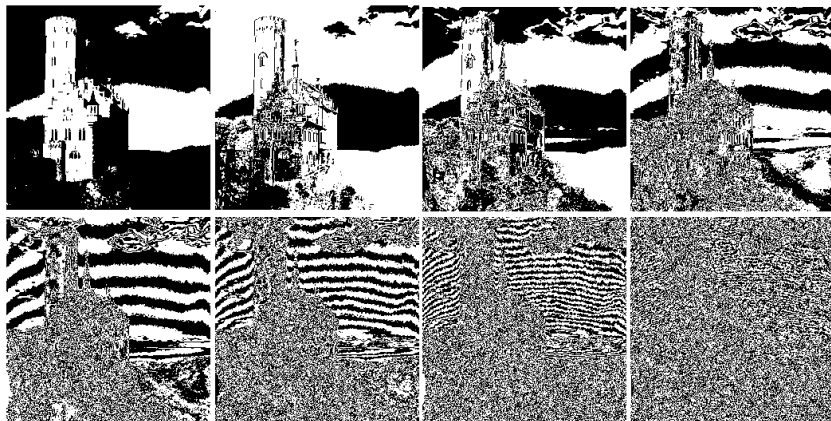
$$k_n = \left\lfloor \frac{u}{2^{B-n}} \right\rfloor - \left\lfloor \frac{u}{2^{B-n+1}} \right\rfloor \times 2, \quad v = \begin{cases} L, & k_n = 1 \\ 0, & n.d.p. \end{cases}$$

Ajuda a determinar o número de bits visualmente significativos.



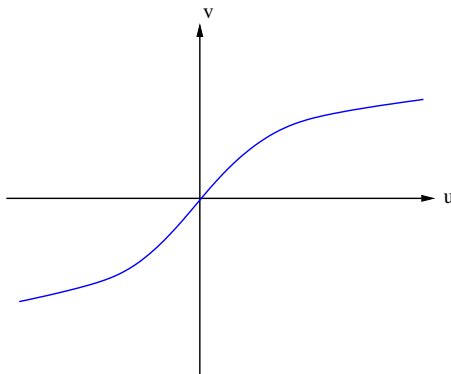
1 pixel = 256 grey level

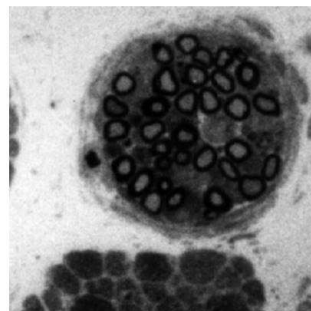
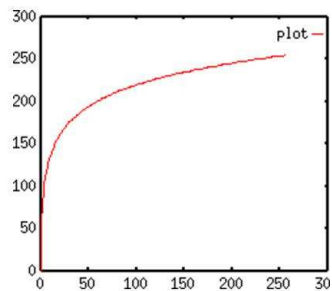
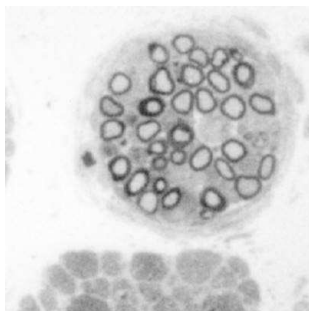




Compressão de Faixa Dinâmica:

$$v = c \log_{10}(1 + |u|)$$





Modelagem de Histogramas

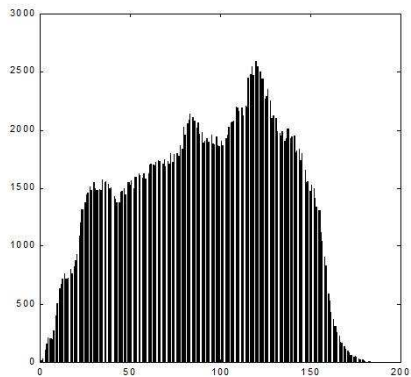


Imagem Subexposta:

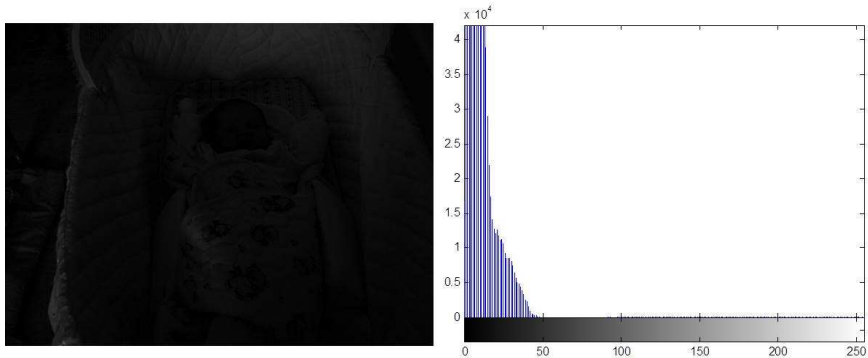
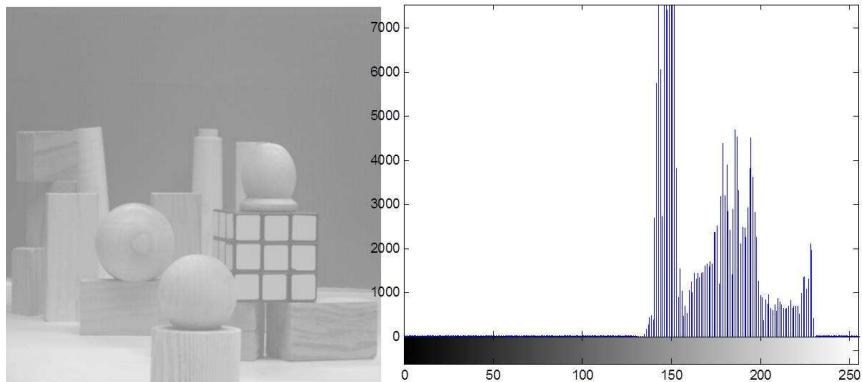


Imagem Superexposta:



Equalização de histogramas:

Caso Contínuo: u com pdf $p_u(u)$

$$v = F_u(u) = \int_0^u p_u(u) du \quad \text{possui densidade uniforme em } [0, 1]$$

$$P[v \leq V] = P[u \leq F_u^{-1}(V)] = \int_0^{F_u^{-1}(V)} p_u(u) du$$

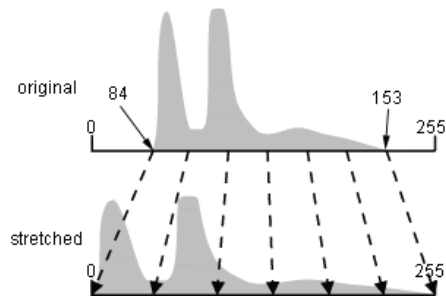
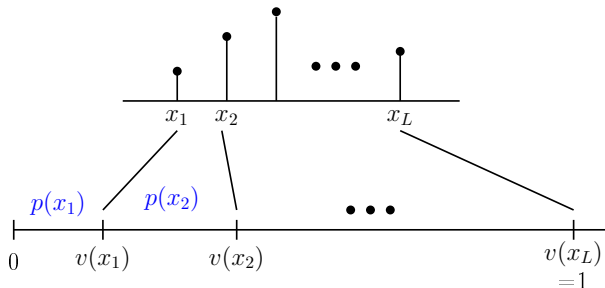
$$= F_u(F_u^{-1}(V)) = V$$

$$= V$$

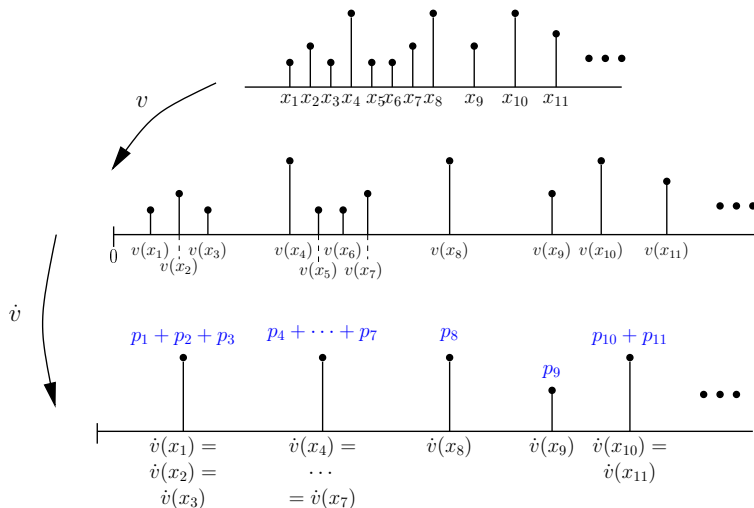
Caso Discreto: u com L níveis $x_i, i = 0, 1, \dots, L-1$, prob. $p_u(x_i)$

$$p_u(x_i) = \frac{n_i}{n} = \frac{h(x_i)}{\sum h(x_i)} \quad v(u) = \sum_{i=0}^k p_u(x_i), \quad x_k \leq u < x_{k+1}$$

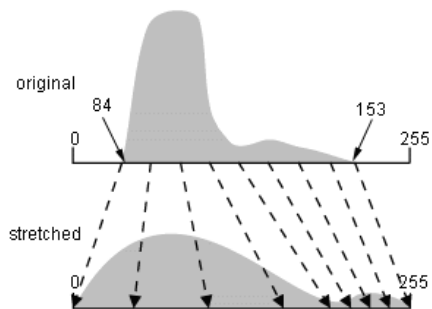
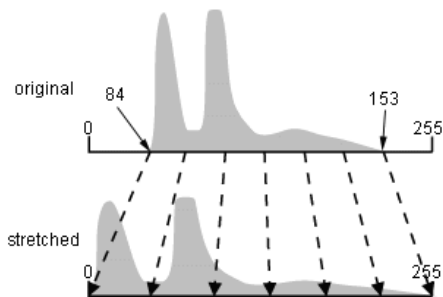
⇒ Se não requantizo V , faço só “histogram stretching”, e mantenho mais ou menos a forma:



Se, ao contrário, re-quantizo v , tenho:



$$\hat{v} = \left\lfloor \frac{v - v_{mn}}{1 - v_{mn}}(L - 1) + 0.5 \right\rfloor$$



Especificação de Histogramas:

$$p_u(u) \rightarrow p_v(v)$$

Se w uniforme: $u \rightarrow w$ $w(u) = \int_0^u p_u(u)du = F_u(u)$

$$v \rightarrow w \quad w(v) = \int_0^v p_v(v)dv = F_v(v)$$

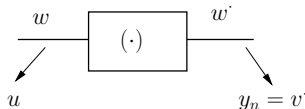
$$\Rightarrow v = F_v^{-1}(w) = F_v^{-1}(F_u(u))$$

u e v discretas: $u : x_i, i = 0, \dots, L-1, \quad p_u(x_i)$
 $v : y_i, i = 0, \dots, L-1, \quad p_v(y_i)$

Sejam $w = \sum_{i=0}^n p_u(x_i), \quad x_k \leq n < x_{k+1}$

$$\tilde{w}_k = \sum_{i=0}^k p_v(y_i), \quad k = 0, \dots, L-1$$

Seja $w' = \tilde{w}_n$ tal que $\begin{cases} \tilde{w}_n - w \geq 0, \\ \tilde{w}_{n-1} - w < 0 \end{cases}$

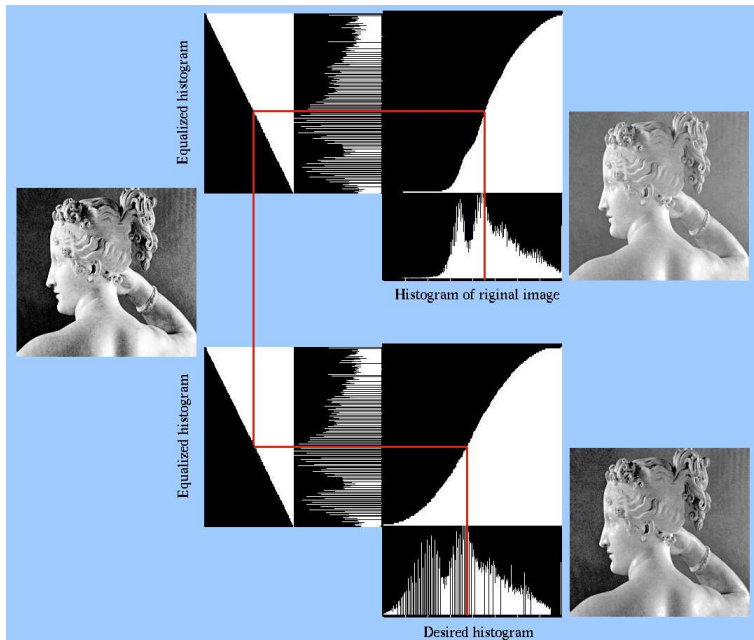


Ex: $x_i = y_i = 0, 1, 2, 3$;

$$p_u(x_i) = 0.25, i = 0, 1, 2, 3$$

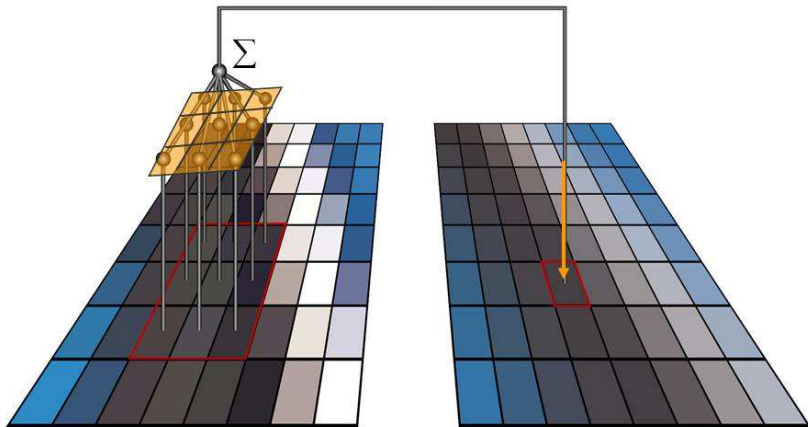
$$p_v(y_0) = p_v(y_3) = 0, p_v(y_1) = p_v(y_2) = 0.5$$

u	$p_u(x_i)$	w	$p_v(y_i)$	\tilde{w}_k	w'	n	v'
0	0.25	0.25	0.0	0.00	0.50	1	1(y_1)
1	0.25	0.50	0.5	0.50	0.50	1	1(y_1)
2	0.25	0.75	0.5	1.00	1.00	2	2(y_2)
3	0.25	1.00	0.0	1.00	1.00	2	2(y_2)



Operações Espaciais

Região é mapeada em um pixel



Médias Espaciais e Filtragem Passa-Baixas

$$v(m, n) = \sum_{(k,l) \in W} a(k, l) y(m-k, n-l)$$

Ex: $a(k, l)$

	0	1
0	1/4	1/4
1	1/4	1/4

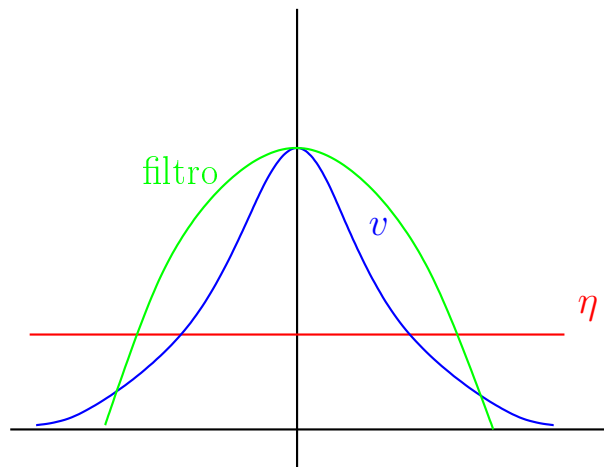
	-1	0	1
-1	1/9	1/9	1/9
0	1/9	1/9	1/9
1	1/9	1/9	1/9

	-1	0	1
-1	0	1/8	0
0	1/8	1/2	1/8
1	0	1/8	0

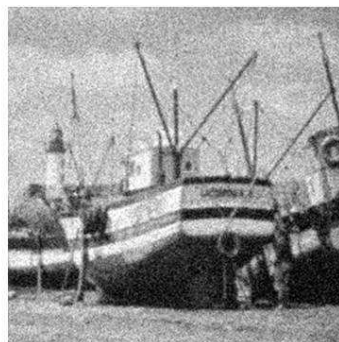
Se $y(m, n) = u(m, n) + \eta(m, n)$, $\eta(m, n) = \text{ruído}$

$$\Rightarrow v(m, n) = \sum \sum u(m-k, n-l) + \underbrace{\tilde{\eta}(m, n)}_{\text{tende a ser pequeno}}$$

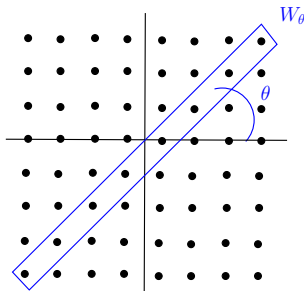
Ou, na frequência:



\Rightarrow aumenta a SNR



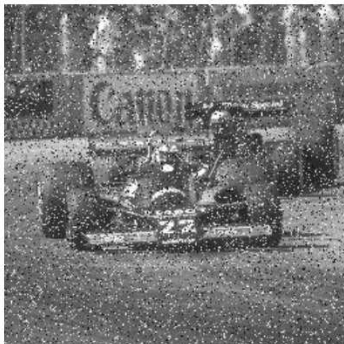
Directional Smoothing: (Quando tenho arestas predominando numa direção)



$$v(m, n, \theta) = \frac{1}{N\theta} \sum_{(k,l) \in W_\theta} y(m-k, n-l)$$

uso θ^* tal que $|y(m, n) - v(m, n, \theta)|$ é mínimo

$$\Rightarrow v(m, n) = v(m, n, \theta^*)$$



Filtragem por Medianas:

$$v(m, n) = \text{mediana}\{y(m - k, n - l), 9k, l) \in W\}$$

- ❶ É não linear;
- ❷ Bom para ruído impulsivo;
- ❸ Ruim quando há muito ruído na janela;

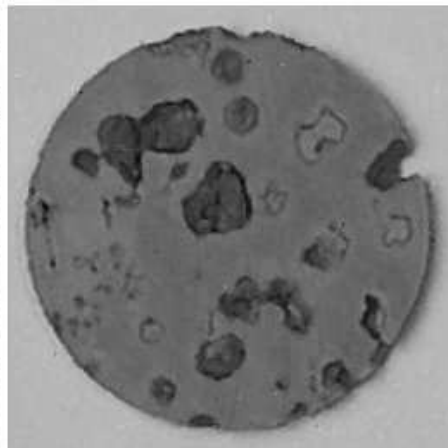
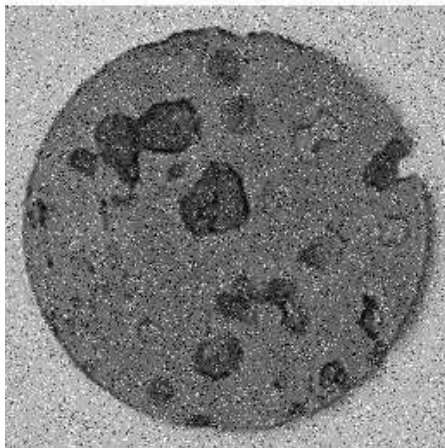
Se a janela tem N_W elementos, e a mediana é o $\frac{N_W + 1}{2}$ maior valor, sua busca requer

$$(N_W - 1) + (N_W - 2) + \dots + \frac{N_W - 1}{2} = \frac{3(N_W - 1)}{2}.$$

\Rightarrow posso reduzir para $\frac{1}{2} N_W \log_2 N_W$.

Se uma janela de tamanho k se move, a cada pixel k novos valores entram e k saem \Rightarrow a nova mediana pode ser achada em $k(N_W + 1)$ comparações.

MEDIAN FILTER



noisy lena



Gaussian filter



median filter



Outras Formas de Suavização

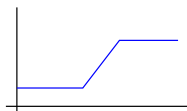
⇒ Só substituir o valor pela média quando o ruído é grande, isto é, $|v(m, n) - y(m, n)| \geq \text{threshold}$.

⇒ Mais algoritmos no Capítulo 8 (Image Restoration).

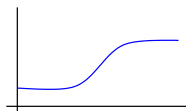
Unsharp Masking e Crispening

$$v(m, n) = u(m, n) + \lambda g(m, n), \quad 0 < \lambda < 1, \quad g(m, n) =$$

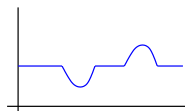
0	-1/4	0
-1/4	1	-1/4
0	-1/4	0



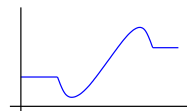
original



lowpass



original - lowpass



$1 + \lambda(\text{original} - \text{lowpass})$

A (original)



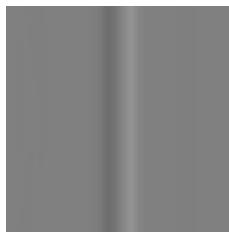
B (fora de foco)



C (B filtrado LP)

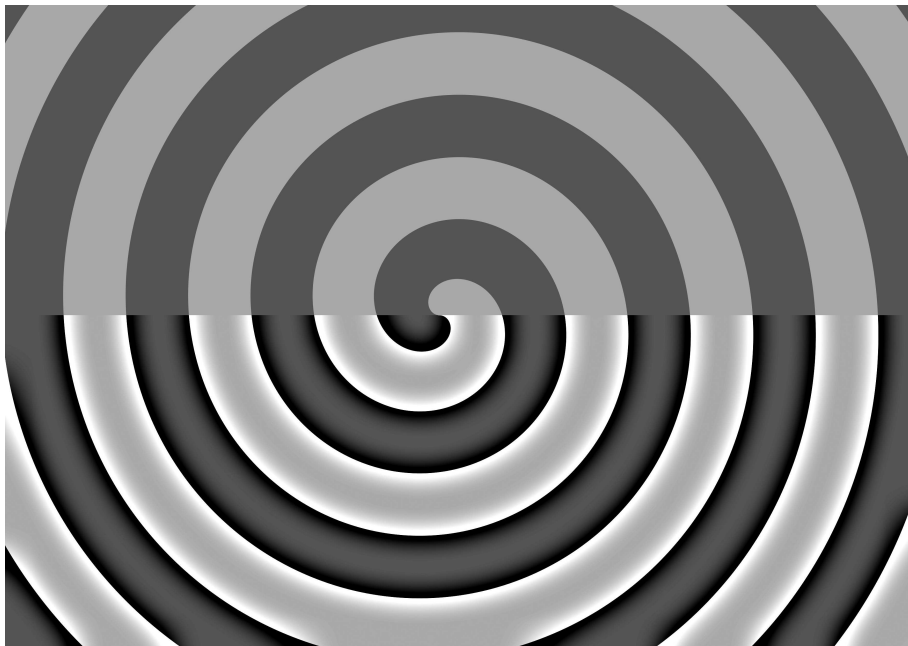


D (B-C)



B + D (unsharped)





Passa Baixas, Passa Altas e Passa Bandas

Passa Baixas: médias: $h_{LP}(m, n) \Rightarrow$ suavização de ruído e interpolação.

Passa Altas: $h_{HP}(m, n) = \delta(m, n) - h_{LP}(m, n) \Rightarrow$ realce de arestas.

Passa Bandas: $h_{BP}(m, n) = \underbrace{h_{L_1}(m, n)}_{\text{média com janela pequena}} - \underbrace{h_{L_2}(m, n)}_{\text{média com janela grande}}$
 \Rightarrow realce de arestas na presença de ruído.



Mapeamento Inverso de Contraste e Escalamento Estatístico

A habilidade para detectar um objeto em um background uniforme depende do tamanho do objeto e da razão de contraste:

$$\gamma = \frac{\sigma}{\mu} \begin{array}{l} \longrightarrow \text{variância do objeto e seu entorno} \\ \longrightarrow \text{luminância média do objeto} \end{array}$$

Seja: $v(m, n) = \frac{\mu(m, n)}{\sigma(m, n)} \Rightarrow \mu$ e σ calculadas dentro de uma janela W .

$$\mu(m, n) = \frac{1}{N_W} \sum_{(k, l) \in W} u(m - k, n - l)$$

$$\sigma(m, n) = \frac{1}{N_W} \sum_{(k, l) \in W} \{ [u(m - k, n - l) - \mu(m, n)]^2 \}^{\frac{1}{2}}$$

\Rightarrow gera uma imagem onde as arestas de baixo contraste são realçadas.

Caso especial: $v(m, n) = \frac{u(m, n)}{\sigma(m, n)} \Rightarrow$ escala cada pixel pelo desvio padrão da janela (escalamento estatístico)

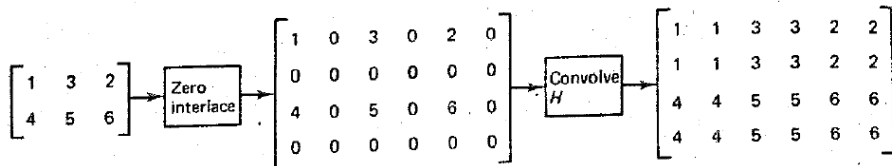
Interpolação e Zoom

Replicação:

$$H = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$$

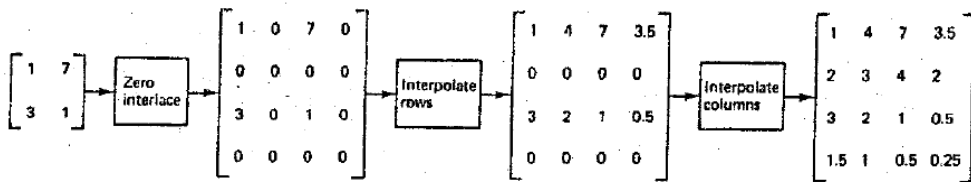
Esquema geral:

- Coloco zeros;
- Filtro com H " n " vezes $\Rightarrow (n-1)$ order hold.



Interpolação:

$$\left. \begin{aligned} v(2m, n) &= v_1(m, n) \\ v(2m+1, n) &= \frac{1}{2}[v_1(m, n) + v_1(m+1, n)], \\ 0 \leq m \leq M-1, 0 \leq n \leq 2N-1 \end{aligned} \right\}$$



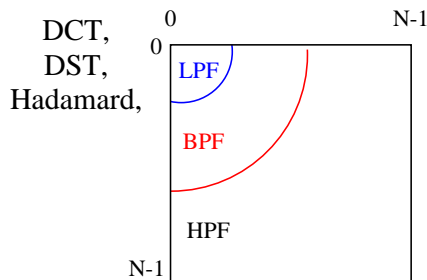
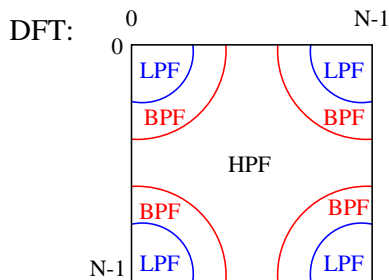
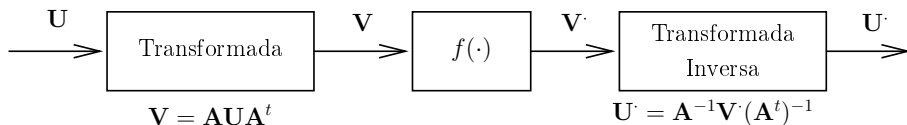
Equivale a filtrar a imagem com zeros entrelaçados com o filtro:

$$H = \begin{bmatrix} \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \\ \frac{1}{2} & \boxed{1} & \frac{1}{2} \\ \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \end{bmatrix}$$

Spline Interpolation: Encho com “ p ” zeros e filtro com H p vezes.

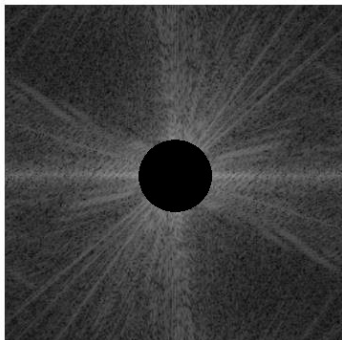
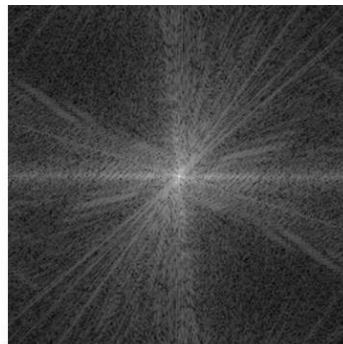
Operações no Domínio das Transformadas

Filtragem Linear Generalizada (Zonal Masking):





DFT

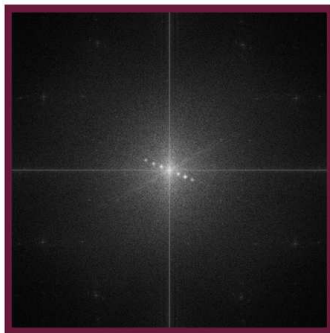


IDFT

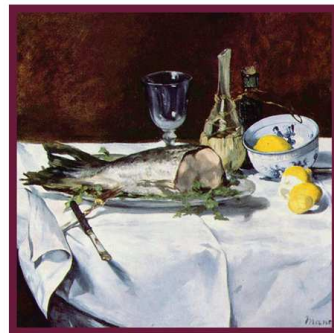




(a) Noisy source.



(b) Spectrum.



(c) Recovered image.

Figure 9.19. Periodic noise removal.

Filtro Gaussiano Inverso: máscara zonal $g(k, l) = e^{-\frac{k^2+l^2}{2\sigma^2}}$.

⇒ Observar que isto é válido para DCT, DST, Hadamard, etc. Para a DFT, modificar de acordo.

⇒ Ênfase de altas frequências de imagens borradas por fenômenos que podem ser modelados por PSF's Gaussianas (por exemplo, turbulência atmosférica).

Filtragem por Raiz

Seja \mathbf{V} a DFT de \mathbf{U} , $v(k, l) = |v(k, l)| e^{j\theta(k, l)}$.

⇒ $\dot{v}(k, l) = |v(k, l)|^\alpha e^{j\theta(k, l)}$, $0 \leq \alpha \leq 1$

Imagens normais: como $|v(k, l)|$ tende a cair em altas frequências, $\alpha \leq 1$ leva a enfatizar altas frequências.

Filtragem Homomórfica

Seja o modelo de imagem dado por $y(m, n) = \underbrace{x(m, n)}_{\text{reflectância}} \underbrace{i(m, n)}_{\text{iluminação}}$

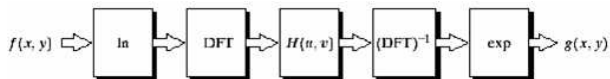
⇒ Problemas ocorrem em casos que a iluminação é ruim, ou está muito escuro em algumas áreas da cena e muito claro em outras.

⇒ $i(m, n)$ depende da fonte de iluminação (e varia dependendo de sua intensidade);

⇒ $x(m, n)$ depende das propriedades do objeto (e normalmente não varia);

⇒ Queremos reduzir a influência da iluminação (que tipicamente varia pouco dentro da imagem) enquanto realçamos o contraste do objeto.

$\log(y(m, n)) = \log(x(m, n)) + \log(i(m, n)) \Rightarrow$ Passo um filtro passa bandas em y , de modo que as variações de baixa frequência de $i(m, n)$ sejam anuladas, faço $e^{y(m, n)}$:





Realce de Imagens Multi-Espectrais

Sequências de \mathcal{I} imagens da mesma cena, uma em cada espectro de frequências (Ex: visível, infravermelho, etc.)

⇒ Tipicamente $1 \leq \mathcal{I} \leq 12$

⇒ Gero \mathcal{I} imagens a partir do conjunto que tenha as características principais bem representadas.

3 métodos principais:

- 1 Razões de Intensidade: $R_{ij}(m, n) = \frac{u_i(m, n)}{u_j(m, n)}$, $i \neq j \Rightarrow$ escolho os R_{ij} mais significativos.
- 2 Razões de Logaritmos: uso $\log R_{ij}$ quando sua faixa dinâmica é muito grande.

3 Componentes Principais:

$$u(m, n) = \begin{bmatrix} u_1(m, n) \\ u_2(m, n) \\ \vdots \\ u_{\mathcal{I}}(m, n) \end{bmatrix} \Rightarrow \text{Acho } \Phi, \text{ a KLT de } u(m, n)$$

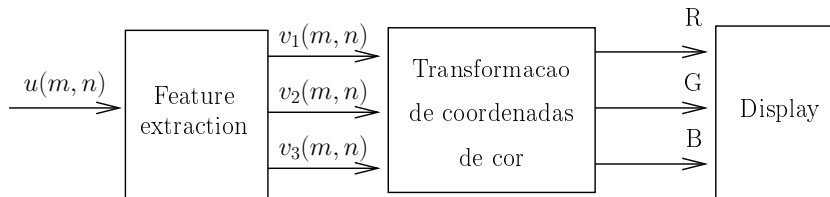
$\Rightarrow v(m, n) = \Phi u(m, n) \Rightarrow$ uso $\mathcal{I}_0 < \mathcal{I}$ imagens $v(m, n)$ mais significativas.

Ex: RGB \Rightarrow YIQ.

Falsa Cor e Pseudo-Cor

Posso distinguir mais cores que níveis de cinza.

⇒ Atribuo cores distintas a características distintas com base em um critério específico;



EX:

Níveis de cinza;

Médias espaciais;

⇒ Os mapeamentos bons de um modo geral são determinados experimentalmente.

Método mais simples: “intensity slicing”

