INF01046 – Fundamentos de processamento de imagens

Aula 14 – DFT Rápida

Horacio E. Fortunato

Instituto de Informática
Universidade Federal do Rio Grande do Sul
Porto Alegre – RS

hefortunato@inf.ufrgs.br

Link do curso: http://www.inf.ufrgs.br/~hefortunato/cursos/INF01046

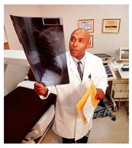
2° semestre de 2009

Processamento Digital de Imagens - Nesta disciplina

Sensores e Aquisição de Imagens

- Sistema visual Humano
- Modalidade de Imagens
- Câmeras Digitais

Processamento para a interpretação humana



- Realce de Imagens:
  - Processamento de histograma
  - Filtragem espacial
  - Filtragem no domínio da frequência
- Restauração de Imagens:
  - Remoção de ruído
  - Remoção de borramento
- Espaços de Cores
- Imagens em Alta Faixa Dinâmica

Percepção por máquina

- Detecção de linhas e bordas
- Limiarização
- Segmentação

Armazenamento e Comunicação

- Compressão de imagens

Transformada de Fourier bidimensional discreta

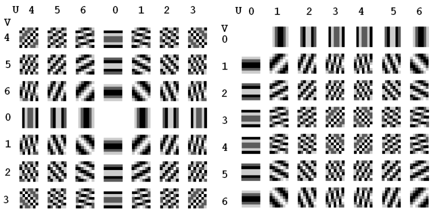
$$f(x,y)=\sum_{u=0}^{M-1}\sum_{v=0}^{N-1}F(u,v)\cdot e^{2\pi i(\frac{ux}{M}+\frac{vy}{N})} \rightarrow F(u,v)=\rho(u,v)e^{i\phi} \rightarrow f(x,y)=\sum_{u=0}^{M-1}\sum_{v=0}^{N-1}|F(u,v)|e^{i\phi}\cdot e^{2\pi i(\frac{ux}{M}+\frac{vy}{N})}$$

$$f(x,y)=\sum_{u=0}^{M-1}\sum_{v=0}^{N-1}|F(u,v)|\cdot e^{i[\phi+2\pi(\frac{ux}{M}+\frac{vy}{N})]}=\sum_{u=0}^{M-1}\sum_{v=0}^{N-1}|F(u,v)|\cdot[\cos(\phi+2\pi(\frac{ux}{M}+\frac{vy}{N}))+i\cdot\sin(\phi+2\pi(\frac{ux}{M}+\frac{vy}{N}))]$$

Se f(x,y) é uma função real, então: $f(x,y)=\sum_{u=0}^{M-1}\sum_{v=0}^{N-1}|F(u,v)|\cdot\cos(\phi+2\pi(\frac{ux}{M}+\frac{vy}{N}))$

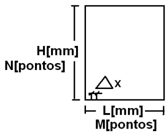
$\cos(\phi+2\pi(\frac{ux}{M}+\frac{vy}{N}))$

para $M=N=7$ *e* $\phi=0\rightarrow$



Origem desplazado N = M = 7

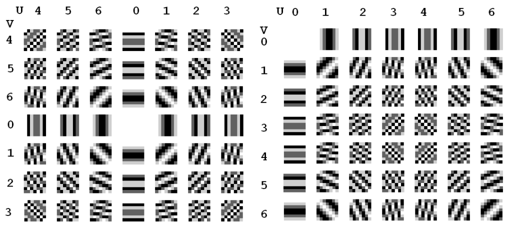
Transformada de Fourier bidimensional discreta



Δx = L / M [mm]
Δy = H / N [mm]

Wx-max = M / 2 * L = 1 / 2 * Δx
Wy-max = N / 2 * H = 1 / 2 * Δy

$$f(x,y)=\sum_{u=0}^{M-1}\sum_{v=0}^{N-1}F(u,v)\cdot e^{2\pi i(\frac{ux}{M}+\frac{vy}{N})}$$

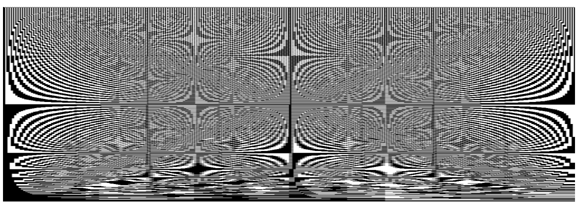


Origem desplazado N = M = 7

Aliasing

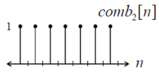
Padrões de interferência que aparecem quando uma imagem que possuen componentes de alta frequência é amostrada utilizando poucos pontos. Exemplo, imagem original com 512 pontos de largura, uma coluna branca e a seguinte preta.

A mesma imagem com diferentes resoluções de amostragem.
Cada linha amostrada com um ponto a menos: 512, 511, … 1



Função de amostragem

Definimos uma função de amostragem como

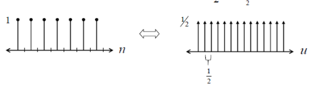


$$comb_{M,N}(x,y)=\sum_{k=-\infty}^{\infty}\sum_{l=-\infty}^{\infty}\delta(x-kM,y-lN)$$

A sua transformada de Fourier também é uma função de amostragem

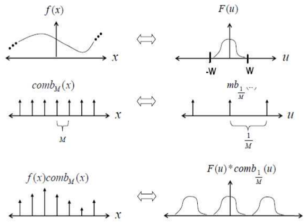
$$\underbrace{\sum_{k=-\infty}^{\infty}\sum_{l=-\infty}^{\infty}\delta[m-kM,n-lN]}_{comb_{M,N}[m,n]}\Leftrightarrow\underbrace{\frac{1}{MN}\sum_{k=-\infty}^{\infty}\sum_{l=-\infty}^{\infty}\delta\left(u-\frac{k}{M},v-\frac{l}{N}\right)}_{comb_{\frac{1}{M},\frac{1}{N}}(u,v)}$$

$comb_2[n]$ $\Leftrightarrow$ $\frac{1}{2}comb_{\frac{1}{2}}(u)$



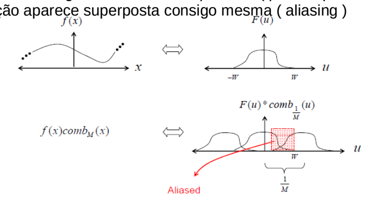
Amostragem

Se f(x) é uma função continua, f(X) é de “banda limitada” se sua transformada de Fourier é zero fora de um intervalo [-W, W] com W finito.
Podemos obter uma versão discreta de f(x) multiplicando pela função de amostragem, que no domínio da frequência, é equivalente à convolução.

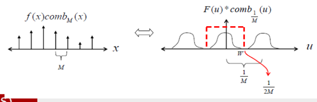


Amostragem

Se os pontos de amostragem estão muito separados (poucos pontos) a transformada de Fourier da Função aparece superposta consigo mesma (aliasing)



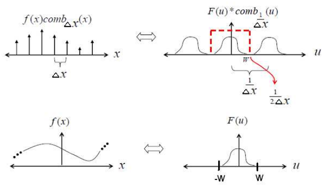
Se os pontos de amostragem estão juntos (muitos pontos) a transformada de Fourier da Função continua, aparece separada para cada ponto (não aparece aliasing)



Teorema da amostragem de Whittaker - Shannon

Se f(x) é uma função de banda limitada com freq. max. W então é possível recuperar a função f(x) completa, partindo de uma amostragem, sempre que o intervalo de amostragem seja menor que 1/2*W.

$$\Delta x \leq 1/2 \cdot W$$



Para esto, multiplicamos a transformada de Fourier da função amostrada, por uma função do tipo 1 se -W<u<W e 0 senão, e calculamos a transformada inversa de Fourier.

Para evitar o efeito de aliasing, deve se aumentar o número de amostras (pontos de amostragem mais juntos) ou filtrar a imagem com um filtro passa baixas antes de fazer a amostragem.

Exemplo de Aliasing: Imagem original

Exemplo de Aliasing: Imagem com poucos pontos

Exemplo de Aliasing: Imagem + passa baixas + poucos pontos

Implementação de um algoritmo para calcular a DFT

Ao estudar a propriedade de separabilidade da DFT observamos que é possível calcular uma DFT bidimensional como duas DFT unidimensionais, uma após a outra:

$$F(u,v)=\frac{1}{(M\cdot N)}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x,y)\cdot e^{-2\pi i\left(\frac{ux}{M}+\frac{vy}{N}\right)}$$

$$F(u,v)=\frac{1}{N}\sum_{y=0}^{N-1} e^{-2\pi i\left(\frac{vy}{N}\right)}\cdot\frac{1}{M}\sum_{x=0}^{M-1} f(x,y)\cdot e^{-2\pi i\left(\frac{ux}{M}\right)}$$

$$F(u,v)=\frac{1}{N}\sum_{y=0}^{N-1} e^{-2\pi i\left(\frac{vy}{N}\right)}\cdot F(u,y)$$

F(u, y) deve ser computada N vezes (uma para cada valor de y).
Então começaremos por implementar um algoritmo para o computo da DFT unidimensional

$$F(u)=\frac{1}{M}\sum_{x=0}^{M-1} f(x)\cdot e^{-2\pi i\frac{ux}{M}}$$

$$Formula\,de\,Euler\rightarrow e^{-2\pi i\frac{u\cdot x}{M}}=\cos(-2\pi\frac{u\cdot x}{M})\ +\ i\cdot\sin(-2\pi\frac{u\cdot x}{M})=\cos(2\pi\frac{u\cdot x}{M})\ -\ i\cdot\sin(2\pi\frac{u\cdot x}{M})$$

$$F(u)=\frac{1}{M}\sum_{x=0}^{M-1} f(x)\cdot[\cos(2\pi\frac{u\cdot x}{M})\ -\ i\cdot\sin(2\pi\frac{u\cdot x}{M})]$$

13

Implementação de um algoritmo para calcular a DFT

$$F(u)=\frac{1}{M}\sum_{x=0}^{M-1} f(x)\cdot[\cos(2\pi\frac{u\cdot x}{M})\ -\ i\cdot\sin(2\pi\frac{u\cdot x}{M})]$$

$$f(x)=f_R(x)\ +\ i\cdot f_I(x)$$

$$F(u)=\frac{1}{M}\sum_{x=0}^{M-1}[f_R(x)\ +\ i\cdot f_I(x)]\cdot[\cos(2\pi\frac{u\cdot x}{M})\ -\ i\cdot\sin(2\pi\frac{u\cdot x}{M})]$$

$$F(u)=\frac{1}{M}\sum_{x=0}^{M-1}[\ f_R(x)\cdot\cos(2\pi\frac{u\cdot x}{M})+f_I(x)\cdot\sin(2\pi\frac{u\cdot x}{M})\ ]\ +\ i\ [\ f_I(x)\cdot\cos(2\pi\frac{u\cdot x}{M})\ -f_R(x)\cdot\sin(2\pi\frac{u\cdot x}{M})\ ]$$

$$F(u)=\ F_R(u)+i\cdot F_I(u)$$

$$F_R(u)=\frac{1}{M}\sum_{x=0}^{M-1}[\ f_R(x)\cdot\cos(2\pi\frac{u\cdot x}{M})+f_I(x)\cdot\sin(2\pi\frac{u\cdot x}{M})\ ]$$

$$F_I(u)=\frac{1}{M}\sum_{x=0}^{M-1}[\ f_I(x)\cdot\cos(2\pi\frac{u\cdot x}{M})\ -f_R(x)\cdot\sin(2\pi\frac{u\cdot x}{M})\ ]$$

14

Implementação de um algoritmo para calcular a DFT

$$C_M[u][x]=\cos(2\pi\frac{u\cdot x}{M})\quad e\quad S_M[u][x]=\sin(2\pi\frac{u\cdot x}{M})$$

$$F_R(u)=\frac{1}{M}\sum_{x=0}^{M-1}[\ f_R(x)\cdot C_M[u][x]\ +\ f_I(x)\cdot S_M[u][x]\ ]$$

$$F_I(u)=\frac{1}{M}\sum_{x=0}^{M-1}[\ f_I(x)\cdot C_M[u][x]\ -\ f_R(x)\cdot S_M[u][x]\ ]$$

A função dft aceita vetores unidimensionais de dimensão K, com a componente real e imaginária de uma linha ou coluna e retorna vetores unidimensionais com as componentes real e imaginaria da DFT

```
void dft( double *fr,  double *fi, double *dftr, double *dfti, int K )
{
int u,x;
double alfa,**cm, **sm;

  Alocar buffers cm, sm.....

  for ( x = 0; x < K; x++ ) {
    for ( u = 0; u < K; u++ ) {
        alfa = 2 * PI * (double) x * (double) u / (double) K;
        cm[x][u] = cos ( alfa );   sm[x][u] = sin ( alfa );
    }
  }

  for ( u = 0; u < K; u++ ) {
     dftr[u] = 0.0;  dfti[u] = 0.0;
     for ( x = 0; x < K; x++ ) {
       dftr[u] += fr[x] * cm[x][u] - fr[x] * sm[x][u];
       dfti[u] += fi[x] * cm[x][u] + fi[x] * sm[x][u];
     }
  }

  for ( u = 0; u < K; u++ ) {
     dftr[u] /= double(K);    dfti[u] /= double(K);
  }

  Free buffers cm, sm ....
}
```

15

Implementação de um algoritmo para calcular a DFT

$$F(u,v)=\frac{1}{(M\cdot N)}\sum_{y=0}^{N-1}\sum_{x=0}^{M-1} f(x,y)\cdot e^{-2\pi i\left(\frac{ux}{M}+\frac{vy}{N}\right)}$$

A função dft2 calcula a DFT bidimensional de uma matriz chamando repetidas veces a função dft.
Primeiro, calcula uma matriz intermédia com as DFT das linhas da imagem.
Depois calcula as DFT das colunas desta matriz.

```
void dft2( double **fr,  double **fi, double **dftr, double **dfti, int M, int N )
{
int x,y;
double **tempr, **tempi;     //imagem temporaria
double *vr, *vi, *vlr, *vli; //buffers para copiar as colunas da matriz temporaria

   Allocar buffers: tempr  tempi, vr, vi, vlr, vli....

   // calcular DFT das linhas e armazenar em matriz temp
   for ( y = 0; y < N; y++ ) {
      dft ( fr[y], fi[y], tempr[y], tempi[y], N );
   }

   // calcular DFT das colunas
   for ( x = 0; x < M; x++ ) {
      //copiar coluna a buffer
      for ( y = 0; y < N; y++ ) {
         vr[y] = tempr[y][x]; vi[y] = tempi[y][x];
       }

      //calcular DFT do buffer
      dft ( vr, vi, vlr, vli, M );

      //copiar DFT de buffer a coluna
      for ( y = 0; y < N; y++ ) {
          dftr[y][x]= vlr[y]; dfti[y][x] = vli[y];
      }
   }

   free buffers: tempr  tempi, vr, vi, vlr, vli....

}
```

16

Transformada rápida de Fourier - FFT

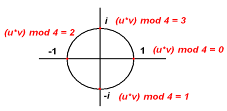
$$F(u)=\frac{1}{M}\sum_{x=0}^{M-1} f(x)\cdot e^{-2\pi i\frac{ux}{M}}$$

$$para\,M=4\ \rightarrow\ F(u)=\frac{1}{4}\sum_{x=0}^{3} f(x)\cdot e^{-2\pi i\frac{ux}{4}}\ \rightarrow\ F_u=\frac{1}{4}\sum_{x=0}^{3} f_x\cdot A_{xu}$$

$$com\quad A_{xu}=e^{-2\pi i\frac{ux}{4}}=e^{-2\pi i\frac{(u\cdot x)\,mod\ 4}{4}}$$

$$[F_0,F_1,F_2,F_3]=\frac{1}{4}\cdot[f_0,f_1,f_2,f_3]\cdot\begin{bmatrix}1&1&1&1\\1&-i&-1&i\\1&-1&1&-1\\1&i&-1&-i\end{bmatrix}$$

$$[F_0,F_1,F_2,F_3]=\frac{1}{4}\cdot[f_0,f_2,f_1,f_3]\cdot\begin{bmatrix}1&1&1&1\\1&-1&1&-1\\1&-i&-1&i\\1&i&-1&-i\end{bmatrix}$$



$$(u\cdot x)\,mod\ 4=\begin{bmatrix}0&0&0&0\\0&1&2&3\\0&2&0&2\\0&3&2&1\end{bmatrix}$$

17

Transformada rápida de Fourier - FFT

$$[F_0,F_1,F_2,F_3]=\frac{1}{4}\cdot[f_0,f_2,f_1,f_3]\cdot\begin{bmatrix}1&1&1&1\\1&-1&1&-1\\1&-i&-1&i\\1&i&-1&-i\end{bmatrix}\rightarrow\begin{bmatrix}A&A\\B&-B\end{bmatrix}\ com\ \ A=\begin{bmatrix}1&1\\1&-1\end{bmatrix}\ e\ \ B=\begin{bmatrix}1&-i\\1&i\end{bmatrix}$$

$$F_0=\frac{1}{4}*(f_0+f_2+f_1+f_3)=\frac{1}{4}*(f_0+f_2)+\frac{1}{4}*(f_1+f_3)=F_{0p}+F_{0i}$$

$$F_1=\frac{1}{4}*(f_0-f_2-i\cdot f_1+i\cdot f_3)=\frac{1}{4}*(f_0-f_2)+\frac{1}{4}*(-i\cdot f_1+i\cdot f_3)=F_{1p}+F_{1i}$$

Vemos que calculando F0 e F1, obtemos F2 e F3 re-utilizando F_{0p}, F_{0i}, F_{1p} e F_{1i} mediante as expressões:

$$F_2=\frac{1}{4}*(f_0+f_2-f_1-f_3)=\frac{1}{4}*(f_0+f_2)-\frac{1}{4}*(f_1+f_3)=F_{0p}-F_{0i}$$

$$F_3=\frac{1}{4}*(f_0-f_2+i\cdot f_1-i\cdot f_3)=\frac{1}{4}*(f_0-f_2)-\frac{1}{4}*(-i\cdot f_1+i\cdot f_3)=F_{1p}-F_{1i}$$

18

Transformada rápida de Fourier - FFT

A transformada rápida de Fourier, é um método para optimizar o computo da DFT, que tira proveito das propriedades de redundância e simetria dos coeficientes da transformação.
Vamos ver um método, que é válido para dimensões de imagens pares.

$$F(u)=\frac{1}{N}\sum_{x=0}^{N-1} f(x)\cdot e^{-2\pi i\frac{ux}{M}} \quad com \quad N=2\cdot M \qquad \rightarrow \; F(u)=\frac{1}{2\cdot M}\sum_{x=0}^{2\cdot M-1} f(x)\cdot e^{-2\pi i\frac{ux}{2\cdot M}} \quad com \quad u=1,2,\ldots,2\cdot M-1$$

Podemos separar os termos com x par dos termos com x impar:

$$F(u)=\frac{1}{2}\cdot\left[\ \frac{1}{M}\sum_{x=0}^{M-1} f(2\cdot x)\cdot e^{-2\pi i\frac{u(2\cdot x)}{2\cdot M}} + \frac{1}{M}\sum_{x=0}^{M-1} f(2\cdot x+1)\cdot e^{-2\pi i\frac{u(2\cdot x+1)}{2\cdot M}}\ \right]$$

$$F(u)=\frac{1}{2}\cdot\left[\ \frac{1}{M}\sum_{x=0}^{M-1} f(2\cdot x)\cdot e^{-2\pi i\frac{u\cdot x}{M}} + \frac{1}{M}\sum_{x=0}^{M-1} f(2\cdot x+1)\cdot e^{-2\pi i\frac{u\cdot x}{M}}\cdot e^{-\pi i\frac{u}{M}}\ \right]$$

$$F(u)=\frac{1}{2}\cdot\left[\ F_{par}(u)+F_{impar}(u)\cdot e^{-\pi i\frac{u}{M}}\ \right] \quad para \quad u=1,2,\ldots,2\cdot M-1$$

$$F_{par}(u)=\frac{1}{M}\sum_{x=0}^{M-1} f(2\cdot x)\cdot e^{-2\pi i\frac{u\cdot x}{M}} \qquad F_{impar}(u)=\frac{1}{M}\sum_{x=0}^{M-1} f(2\cdot x+1)\cdot e^{-2\pi i\frac{u\cdot x}{M}}\cdot e^{-\pi i\frac{u}{M}}$$

Vemos que uma transformada de 2.M pontos pode ser expressada como a suma de duas de M pontos.

Transformada rápida de Fourier - (Cont)

$$F(u)=\frac{1}{2}\cdot\left[\ F_{par}(u)+F_{impar}(u)\cdot e^{-\pi i\frac{u}{M}}\ \right] \quad para \quad u=1,2,\ldots,2\cdot M-1$$

$$F_{par}(u)=\frac{1}{M}\sum_{x=0}^{M-1} f(2\cdot x)\cdot e^{-2\pi i\frac{u\cdot x}{M}} \qquad F_{impar}(u)=\frac{1}{M}\sum_{x=0}^{M-1} f(2\cdot x+1)\cdot e^{-2\pi i\frac{u\cdot x}{M}}\cdot e^{-\pi i\frac{u}{M}}$$

As seguintes relações, permitem calcular F(u) para u entre M e N-1 utilizando os cálculos intermédios efetuados para computar F(u) para u de 1 a N-1:

$$F_{par}(u+M)=\frac{1}{M}\sum_{x=0}^{M-1} f(2\cdot x)\cdot e^{-2\pi i\frac{(u+M)\cdot x}{M}}=\frac{1}{M}\sum_{x=0}^{M-1} f(2\cdot x)\cdot e^{-2\pi i\frac{u\cdot x}{M}}=F_{par}(u)$$

$$F_{impar}(u+M)=\frac{1}{M}\sum_{x=0}^{M-1} f(2\cdot x+1)\cdot e^{-2\pi i\frac{(u+M)\cdot x}{M}}\cdot e^{-\pi i\frac{u+M}{M}}=\frac{1}{M}\sum_{x=0}^{M-1} f(2\cdot x+1)\cdot e^{-2\pi i\frac{u\cdot x}{M}}\cdot e^{\frac{-\pi i u}{M}}\cdot(-1)=-F_{impar}(u)$$

$$F(u)=\frac{1}{2}\cdot\left[\ F_{par}(u)+F_{impar}(u)\cdot e^{-\pi i\frac{u}{M}}\ \right] \quad para \quad u=1,2\ldots M-1$$

$$F(u)=\frac{1}{2}\cdot\left[\ F_{par}(u)-F_{impar}(u)\cdot e^{-\pi i\frac{u}{M}}\ \right] \quad para \quad u=M,M+1,\ldots,2\cdot M-1$$

Transformada rápida de Fourier - Custo da DFT

O número de operações de adição e multiplicação necessários para calcular a DFT sem optimizações é proporcional a M* M

Utilizando o algoritmo FFT para uma imagem de dimensão M = 2 ^ n obtemos

$$se\, M=2^n \;\Rightarrow\; Número\ de\ operações \;\propto\; M\log_2 M$$

$$C(M)=\frac{M^2}{M\cdot\log_2(M)} \;\rightarrow\; C(n)=\frac{(2^n)^2}{2^n\cdot\log_2(2^n)}=\frac{2^n}{n}$$

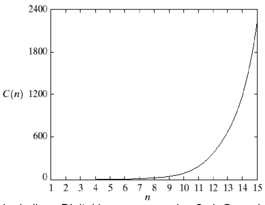


FIGURE 4.42 Computational advantage of the FFT over a direct implementation of the 1-D DFT. Note that the advantage increases rapidly as a function of *n*.

Imagem extraída do livro: Digital image processing 2ed, Gonzales e woods.

Processamento Digital de Imagens - Tarefas

Tarefas Acumuladas:

- •Leia os Capítulo 1, 2, e 3 (aulas 01 a 09) do livro Gonzalez, R. & Woods 2da Ed. (em Inglês)
- •Faça os exercicios do Capítulos 1 a 4 do livro Gonzalez, R. & Woods 2da Ed. (em Inglês)
- •Leia as seções 4.1 a 4.6.5 do Capítulo 4 do livro Gonzalez, R. & Woods 2da Ed. (em Inglês)
- •Faça os exercícios do Capítulo 4 do livro Gonzalez, R. & Woods 2da Ed. (em Inglês)
- •Estude as seções 1, 2 e 3 do tutorial do MATLAB: http://www.mathworks.com/access/helpdesk/help/pdf_doc/matlab/getstart.pdf

Tarefas Novas:

- •Leia as seções 4.6.6 a 4.6.7 do Capítulo 4 (aula 14) do livro Gonzalez, R. & Woods 2da Ed. (em Inglês)

Nota Importante: No livro Gonzalez, R.& Woods em português os capítulos possuem número diferente

Livro Gonzalez, R. & Woods 2ª Ed. (em Inglês):
Gonzalez, R. & Woods, R. Digital Image Processing 2ª Ed. Prentice Hall, 2002.
Link do curso: http://www.inf.ufrgs.br/~hefortunato/cursos/INF01046