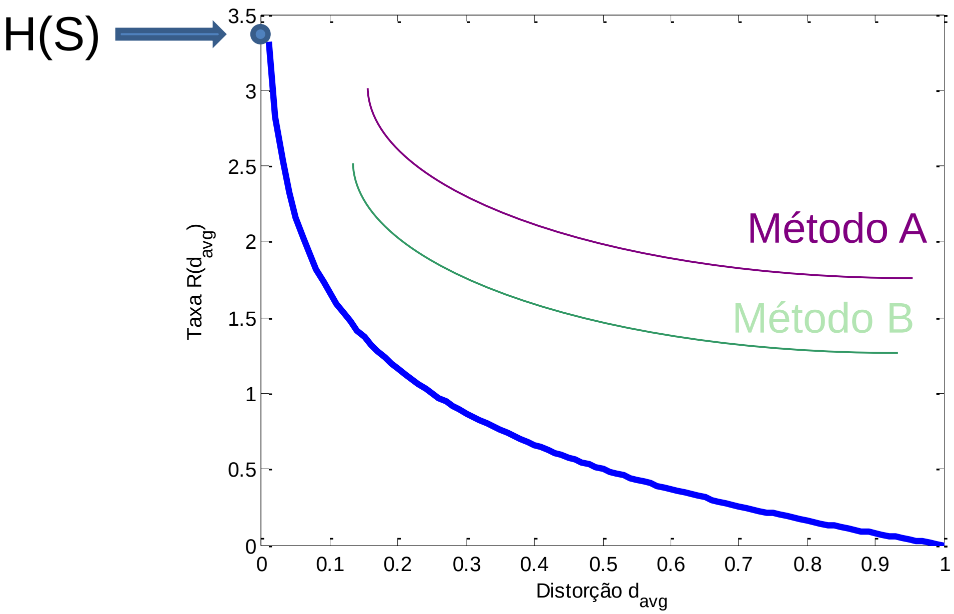
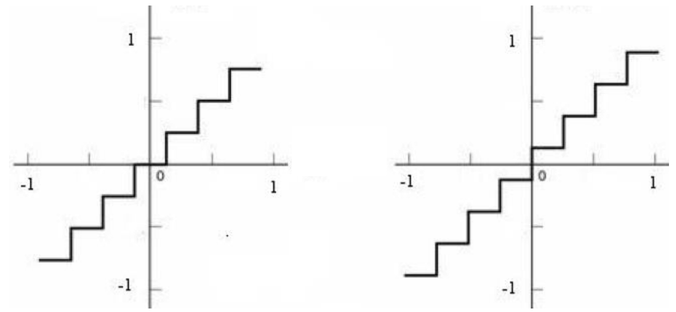
Teoria da Informação: função Taxa/Distorção

- Rate distortion function
  - Nenhum método de compressão está abaixo do limite inferior (assumindo fontes sem memória)
  - Para uma dada distorção $d_{avg} = \mathbf{E}[d(S,\tilde{S})]$, existe uma taxa $R(d_{avg})$, a qual corresponde ao número mínimo de bits necessários para reconstruir a fonte S.
  - Ex:

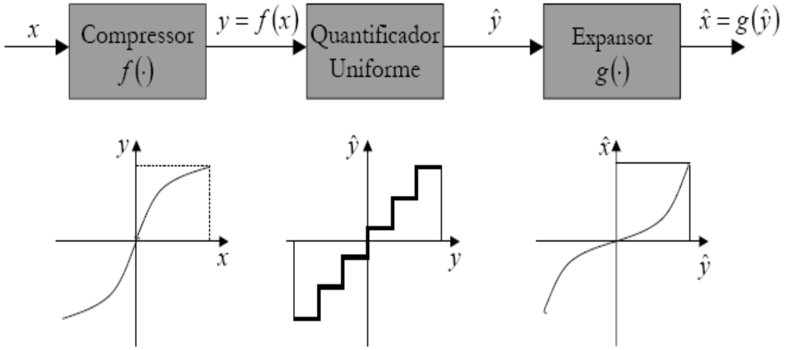


Métodos de Quantificação

- Quantificação escalar
  - Uniforme
    - Midtread e Midrise

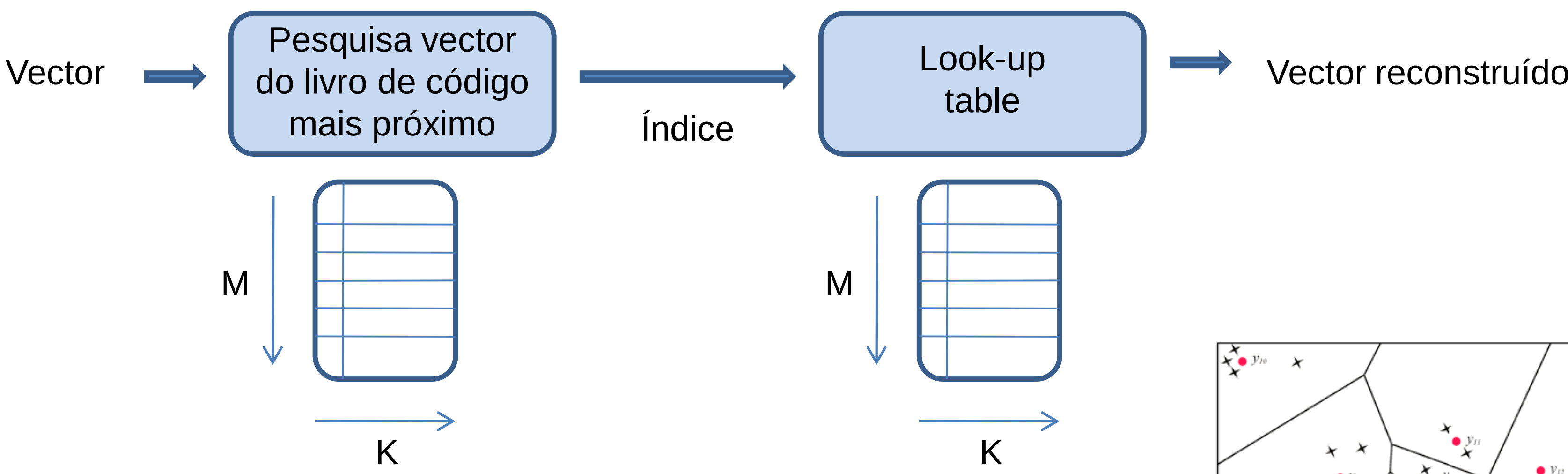


  - Não uniforme
    - Quantificação óptima (Lloyd-Max)
    - Companding / Expanding
      ex: lei A e lei μ

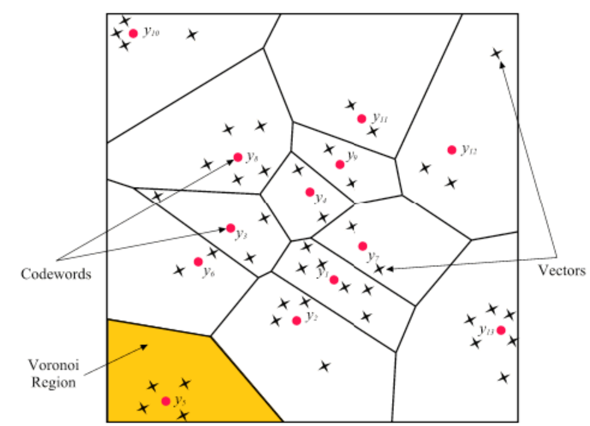


Métodos de Quantificação

- Quantificação Vectorial
  - Quantificar vectores (em vez de escalares);
  - Mapeamento de N vectores a K dimensões num conjunto de M vectores (Livro de Código);
  - Não há nenhuma ordem implicita como caso 1D.



- Problemas a resolver:
  - Qual o livro de código
  - Qual a métrica
  - Algoritmo de pesquisa



Métodos de compressão com perdas

- Codificação por transformada
  - Discrete Fourier Transform (DFT);
  - Discrete Cosine Transform (DCT);

- Motivação:
  - Tanto o sistema auditivo humano como o sistema visual tem limitações na frequência, que podem ser explorados.

- Principio básico:
  - Codificar vectores é mais eficiente que codificar escalares;
  - Se da transformação linear de um vector **x**, resultar um vector **y** com menor correlação, então a codificação deste pode ser mais eficiente que a codificação do vector inicial.

$$\mathrm{y} = \mathrm{T}\mathrm{x}$$

  - A transformação **T**, não faz compressão dos dados, esta é efectuada no processo de quantificação do vector **y**.

Discrete Fourier Transform - DFT

- DFT – unidimensional

$$
\begin{aligned}
X(k) &= \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} e^{-j\frac{2\pi k}{N}n} x(n) \qquad 0 \leq k \leq N-1 \\
x(n) &= \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{j\frac{2\pi k}{N}n} X(k) \qquad 0 \leq n \leq N-1
\end{aligned}
$$

ou seja

$$
\begin{bmatrix} X(0) \\ X(1) \\ \vdots \\ X(N-1) \end{bmatrix} = \frac{1}{\sqrt{N}} \begin{bmatrix} e^{-j\frac{2\pi 0}{N}0} & e^{-j\frac{2\pi 0}{N}1} & \cdots & e^{-j\frac{2\pi 0}{N}N-1} \\ e^{-j\frac{2\pi 1}{N}0} & e^{-j\frac{2\pi 1}{N}1} & \cdots & e^{-j\frac{2\pi 1}{N}N-1} \\ \vdots & & \ddots & \\ e^{-j\frac{2\pi(N-1)}{N}0} & e^{-j\frac{2\pi(N-1)}{N}1} & \cdots & e^{-j\frac{2\pi(N-1)}{N}N-1} \end{bmatrix} \begin{bmatrix} x(0) \\ x(1) \\ \vdots \\ x(N-1) \end{bmatrix}
$$

matricialmente,

$$\mathbf{X} = \mathbf{T}\mathbf{x}$$

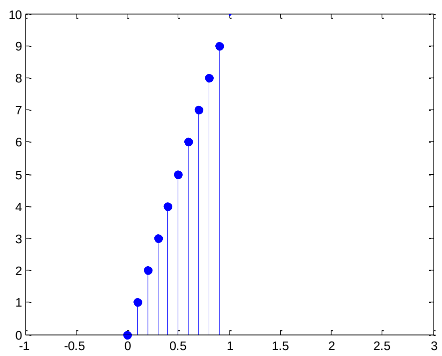
onde, $$\mathbf{T}_{k,n} = \frac{1}{\sqrt{N}} e^{-j\frac{2\pi k}{N}n}$$

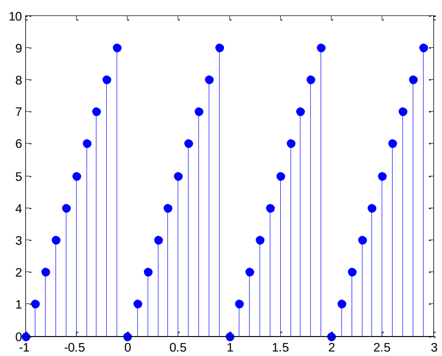
- Notar que a IDFT é: $\mathbf{x} = \mathbf{T}^{*T}\mathbf{X}$, dado que $\mathbf{T}^{-1} = \mathbf{T}^{*T}$

Discrete Fourier Transform - DFT

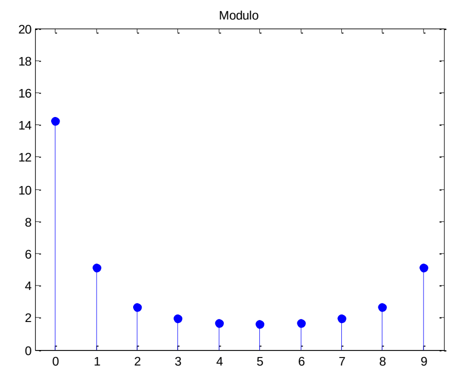
- Assume periodicidade que não é verdade para a grande maioria dos casos.
- Obriga a que haja componentes de alta frequência com valores elevados.
- FFT - Implementação com complexidade muito baixa.

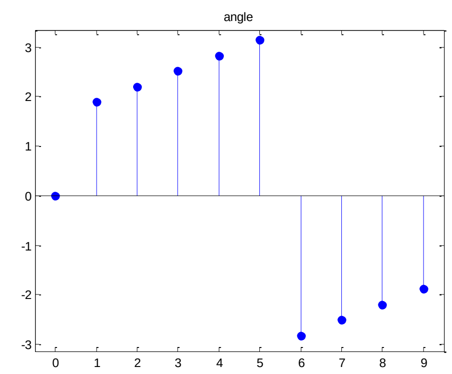
x(n)





X(k)





Discrete Cosine Transform - DCT

- DCT – unidimensional

$$
\begin{aligned}
y(k) &= \sqrt{\frac{2}{N}}\alpha(k)\sum_{n=0}^{N-1} cos\left(\frac{2n+1}{2N}\pi k\right)x(n) \qquad 0 \le k \le N-1 \\
x(n) &= \sqrt{\frac{2}{N}}\sum_{k=0}^{N-1} \alpha(k) cos\left(\frac{2n+1}{2N}\pi k\right)y(k) \qquad 0 \le n \le N-1 \\
\alpha(0) &= 1/\sqrt{2}, \;\; \alpha(k) = 1, \; k \neq 0
\end{aligned}
$$

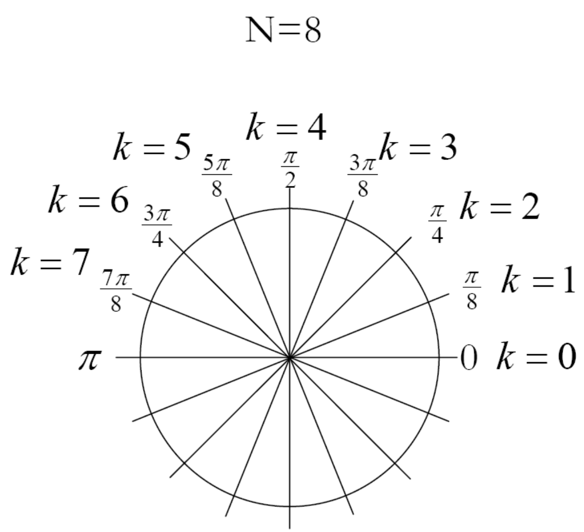
Ou seja $\mathbf{y} \;=\; \mathbf{Cx}$

$$
\begin{bmatrix} y(0) \\ y(1) \\ \vdots \\ y(N-1) \end{bmatrix} = \begin{bmatrix} \sqrt{\frac{1}{N}} & \sqrt{\frac{1}{N}} & \sqrt{\frac{1}{N}} & \cdots \\ \sqrt{\frac{2}{N}} & \sqrt{\frac{2}{N}} cos\left(\frac{3}{2N}\pi 1\right) & \sqrt{\frac{2}{N}} cos\left(\frac{5}{2N}\pi 1\right) & \cdots \\ \vdots & \vdots & \vdots & \cdots \\ \sqrt{\frac{2}{N}} & \sqrt{\frac{2}{N}} cos\left(\frac{3}{2N}\pi (N-1)\right) & \sqrt{\frac{2}{N}} cos\left(\frac{5}{2N}\pi (N-1)\right) & \cdots \end{bmatrix} \begin{bmatrix} x(0) \\ x(1) \\ \vdots \\ x(N-1) \end{bmatrix}
$$

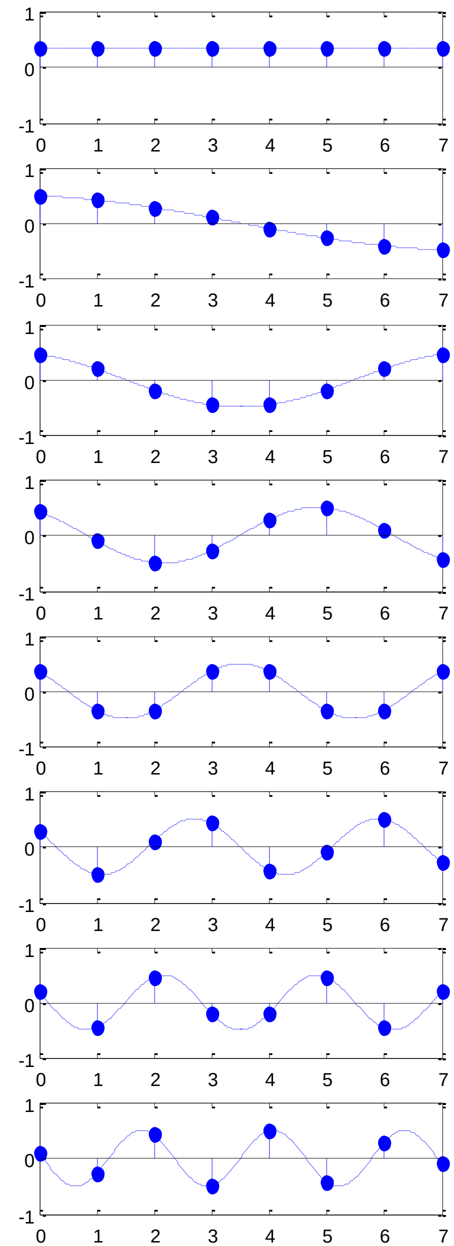
- Notar que: $\mathbf{C}^{-1} = \mathbf{C}^{*T}$

Discrete Cosine Transform - DCT

- Os coeficientes da DCT são todos reais (não precisa de fase)
- A DCT não é a parte real da DFT!

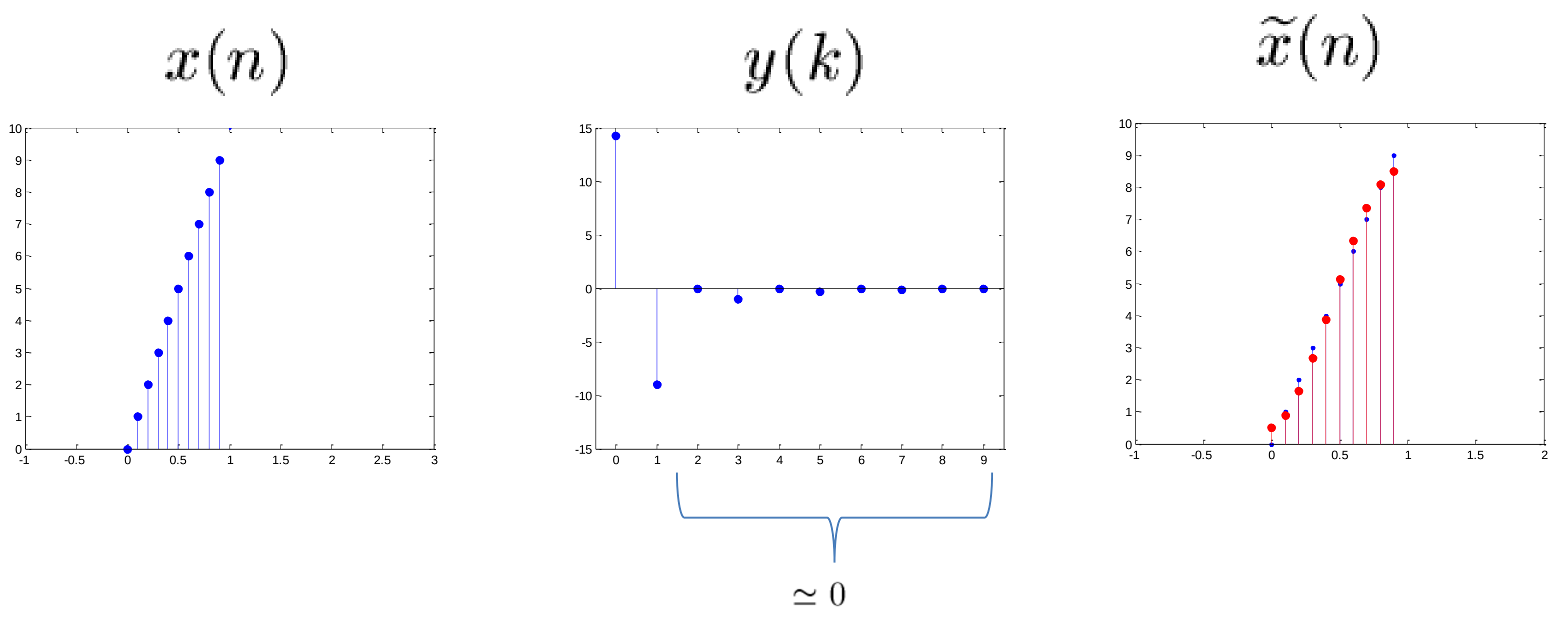


- Qual a relação entre os coeficientes da DCT e da DFT ?



Discrete Cosine Transform - DCT

- Não assume periodicidade logo as componentes de alta frequência têm valores muito menores que a DFT.
- A compactação de energia nos primeiros coeficientes é superior à DFT.



Transformações Ortonormadas

- A DCT e a DFT são casos particulares de transformações ortonormadas:

- As transformações ortonormadas têm as seguintes propriedades:

  - $\mathbf{T}^{-1} = \mathbf{T}^{*T}$

  - A transformação $\mathbf{y} = \mathbf{T}\mathbf{x}$ é a representação do sinal na base constituída pelos vectores coluna da matriz $\mathbf{T}^{*T} = [\mathbf{t}_1 \ldots \mathbf{t}_N]$

  - Os vectores coluna têm norma 1 e são ortogonais entre si:

$$
\begin{aligned}
\mathbf{t}_i^{*T}\mathbf{t}_j &= 0, \;\; i \neq j \\
&= 1, \;\; i = j
\end{aligned}
$$

Transformações Ortonormadas

- Propriedades das transformações ortonormadas (cont.) :

  - Conservação da energia:

$$\|\mathbf{y}(k)\|^2 = \sum_{k=0}^{N-1} |\mathbf{y}(k)|^2 = \sum_{n=0}^{N-1} |\mathbf{x}(n)|^2 = \|\mathbf{x}(n)\|^2$$

  - Erro quadrático :
    (Admitindo que houve erro na transmissão, ou na compressão, ou que simplesmentre houve coeficientes que foram alterados)

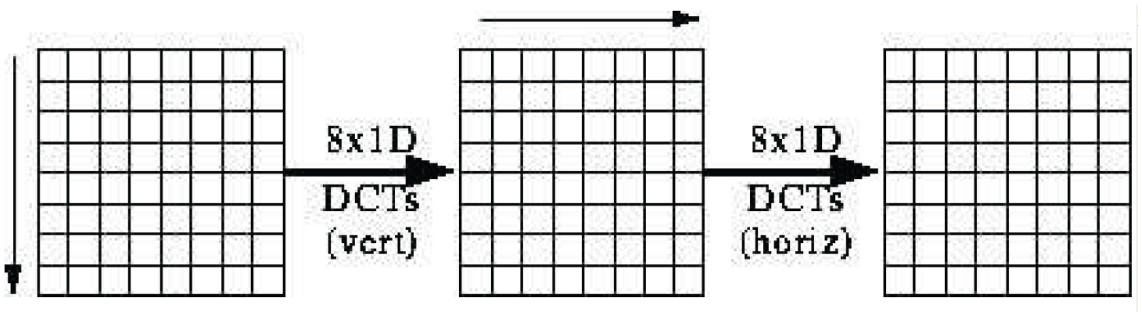
$$\sum_{k=0}^{N-1} |\mathbf{y}(k) - \tilde{\mathbf{y}}(k)|^2 = \sum_{n=0}^{N-1} |\mathbf{x}(n) - \tilde{\mathbf{x}}(n)|^2$$

Propriedades das transformações 2D ortonormadas

- Para sinais 2D (imagens) pode-se extender o conceito desta transformação.

$$\mathbf{Y}(k,l) = \mathbf{T}(k,l,m,n)\mathbf{X}(m,n)$$

- A transformação 2D é separável em duas operações
  uma transformação aplicada a todas as colunas
  seguida de outra transformação aplicada a todas as linhas



$$\begin{aligned}\mathbf{Y}(k,l) &= \mathbf{T}^*(k,m)\mathbf{X}(m,n)\mathbf{T}^T(l,n)\\ \mathbf{X}(m,n) &= \mathbf{T}^T(k,m)\mathbf{Y}(k,l)\mathbf{T}^*(l,n)\end{aligned}$$

Propriedades das transformações 2D ortonormadas

- As propriedasdes das transformações em 1D são extendidas para 2D (convolução, correlação, conservação de energia, erro quadrático, etç)

$$||\mathbf{Y}(k,l)||^2 = \sum_{k=0}^{M-1}\sum_{l=0}^{N-1} |\mathbf{Y}(k,l)|^2 = \sum_{m=0}^{M-1}\sum_{n=0}^{N-1} |\mathbf{X}(n,m)|^2 = ||\mathbf{X}(n,m)||^2$$

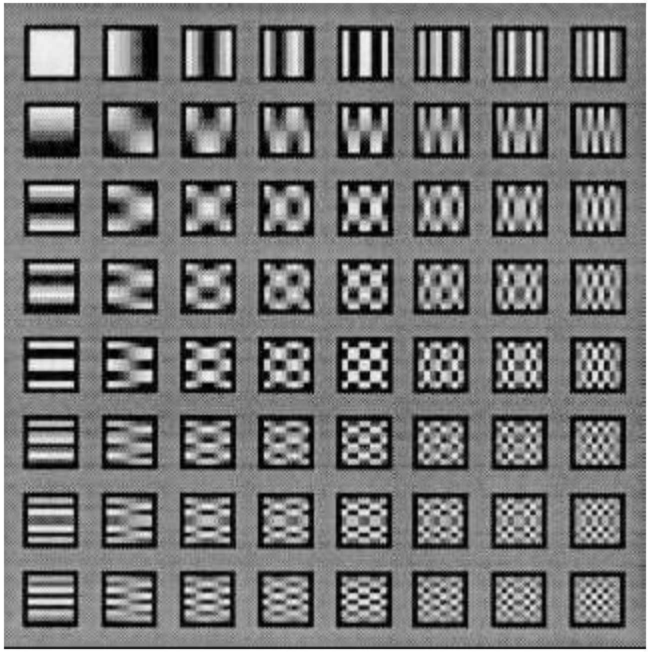
$$\sum_{k=0}^{M-1}\sum_{l=0}^{N-1} |\mathbf{Y}(k,l) - \tilde{\mathbf{Y}}(k,l)|^2 = \sum_{m=0}^{M-1}\sum_{n=0}^{N-1} |\mathbf{X}(n,m) - \tilde{\mathbf{X}}(n,m)|^2$$

DCT - 2D

- Para cada pixel x(m,n):

$$
\begin{aligned}
y(k,l) &= \frac{2}{\sqrt{MN}}\alpha(k)\alpha(l)\sum_{m=0}^{M-1}\sum_{n=0}^{N-1} cos\left(\frac{2m+1}{2M}\pi k\right) cos\left(\frac{2n+1}{2N}\pi l\right) x(m,n) \\
x(m,n) &= \frac{2}{\sqrt{MN}}\sum_{k=0}^{M-1}\sum_{l=0}^{N-1} \alpha(k)\alpha(l) cos\left(\frac{2m+1}{2M}\pi k\right) cos\left(\frac{2n+1}{2N}\pi l\right) y(k,l) \\
\alpha(0) &= \sqrt{2}/2,\ \ \alpha(k) = 1,\ k \neq 0
\end{aligned}
$$

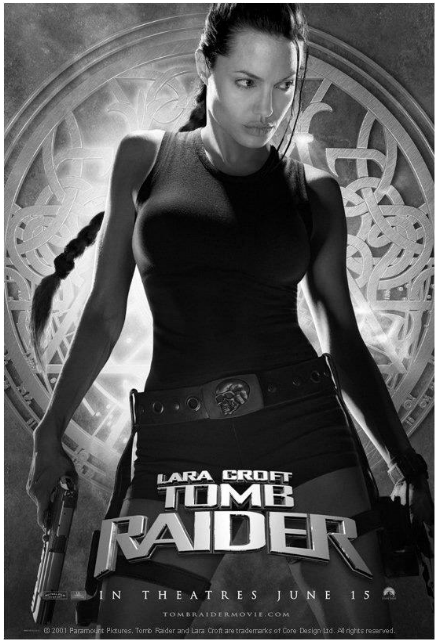
- As funções de base da DCT 2D para o caso M=N=8 são:



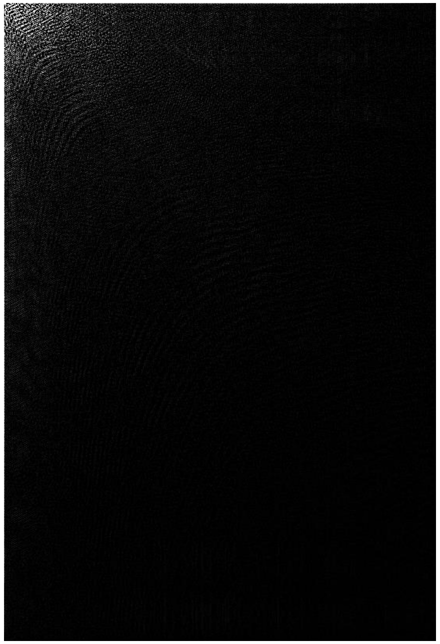
DCT(2D) versus DFT(2D)

- A DCT consegue concentrar melhor a energia da imagem:

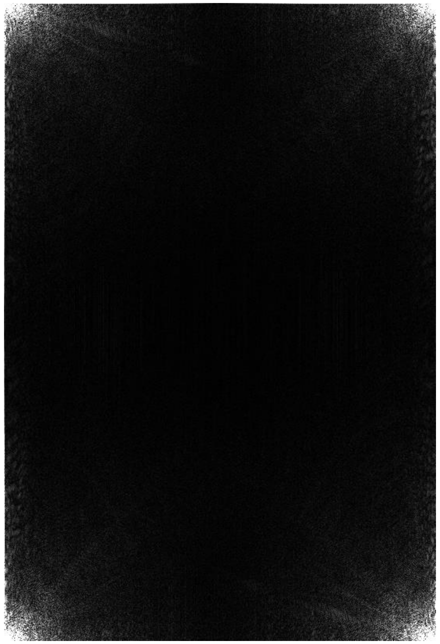
I(m,n)



DCT(I)



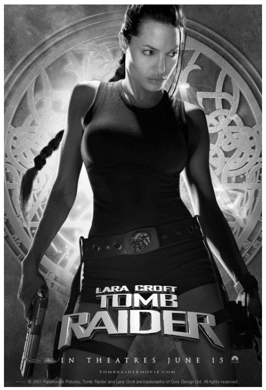
FFT(I)

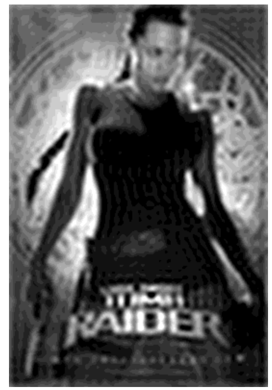


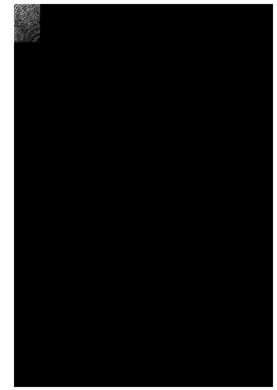
Reconstrução de imagem

- Reconstruir a imagem apenas 1% dos coeficientes da DCT

I(m,n) iDCT DCT(I)







- Reconstruir a imagem apenas 4% dos coeficientes da DCT

Quantificação e Codificação

- A quantificação e a atribuição de bits têm de ser optimizadas de forma a que a compressão seja mais eficiente.

- O critério de optimização mais comum é minimizar o erro quadrático médio (mean square error - mse)

- Usar um quantificador não uniforme óptimo
  - Pode ser complicado o seu projecto
  - É dependente da imagem

- Usar um quantificador uniforme

- Notar que se a transformada for ortonormada,
  o mse da imagem é igual ao mse dos coeficientes.

Vantagens/Desvantagens da DCT e DFT

- Descorrelaciona as amostras do sinal
  (remove a redundânica dos dados)

- Concentra a energia em poucos coeficientes
  (selecção de coeficientes)
  Melhora a codificação entrópica

- As funções de base são de grande dimensão;
  Ex: para um sinal com 1024 amostras tem-se 1024 funções de base e cada tem dimensão 1024.

  - Na prática, para resolver este problema, a imagem divide-se em sub-blocos reduzindo assim a complexidade computacional. (problema: occorrem efeitos de bloco e perda de contraste)

- Apresenta problemas para sinais não estacionários;

- É dificil estimar as características temporais (sinais 1D) ou espaciais (sinais 2D) apartir dos coeficientes espectrais.

JPEG

- Standard desde 1992
- Objectivo: compressão de imagens fotográficas (estáticas)
- Trabalha com:
    - Imagens a cor (24 bit) ou cizento
    - Imagens com 65535 x 65535
    - Precisão de 8 ou 12 bits
- Compromisso entre qualidade e factor de compressão;
- Pode ser usado com ou sem perdas
- Modos de operação
    - Sequencial
    - Progressivo
    - Hierárquico
    - Reversível (sem perdas)

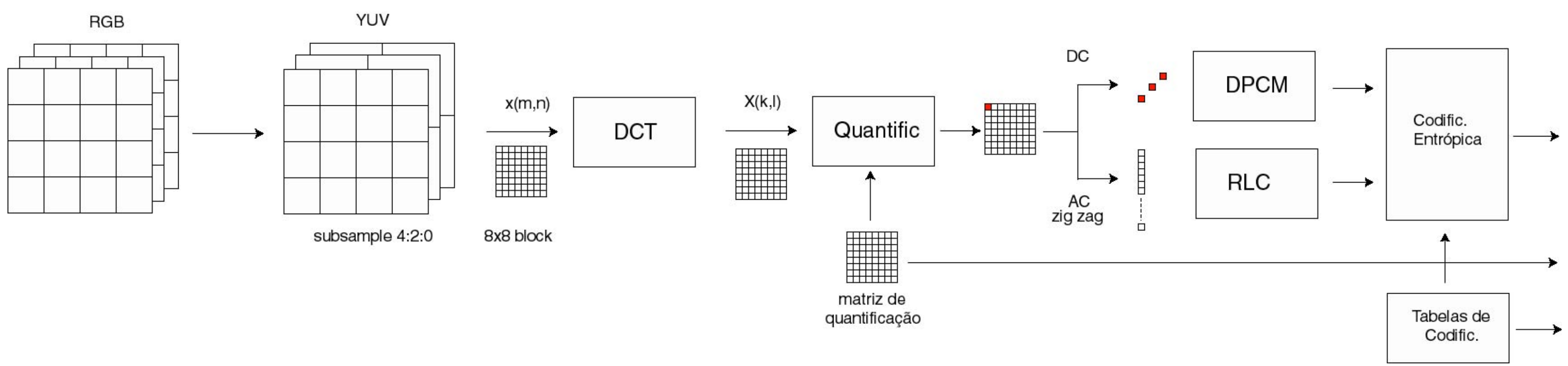
JPEG - Baseline

- A imagem a cores é transformada para o espaço de côr YUV, e faz-se uma sub-amostragem da crominância:

  A acuidade visual é maior para a luminância do que para a crominância.
- Tratamento da imagem em sub-blocos de 8x8:

  A variação em cada bloco não é muito grande.
- A quantificação dos coeficientes da DCT perde componentes de mais alta frequência:

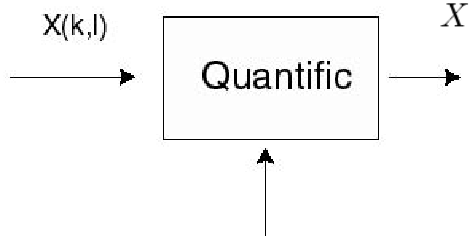
  A visão humana não é muito sensível a variações espaciais de alta frequência.



JPEG - Quantificação

- As tabelas de quantifiacação resultam de estudos psico-visuais, com o objectivo de maximizar a taxa de compressão sem perda perceptiveis de qualidade.

1337	56	-27	18	78	-60	27	-27
-38	-27	13	44	32	-1	-24	-10
-20	-17	10	33	21	-6	-16	-9
-10	-8	9	17	9	-10	-13	1
-6	1	6	4	-3	-7	-5	5
2	3	0	-3	-7	-4	0	3
4	4	-1	-2	-9	0	2	4
3	1	0	-4	-2	-1	3	1



$$\widetilde{X}(k,l) = \text{round}\left(\frac{X(k,l)}{Q(k,l)}\right)$$

84	5	-3	1	3	-1	1	0
-3	-2	1	2	1	0	0	0
-1	-1	1	1	1	0	0	0
-1	0	0	1	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
0	0	0	0	0	0	0	0

- Luminance

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

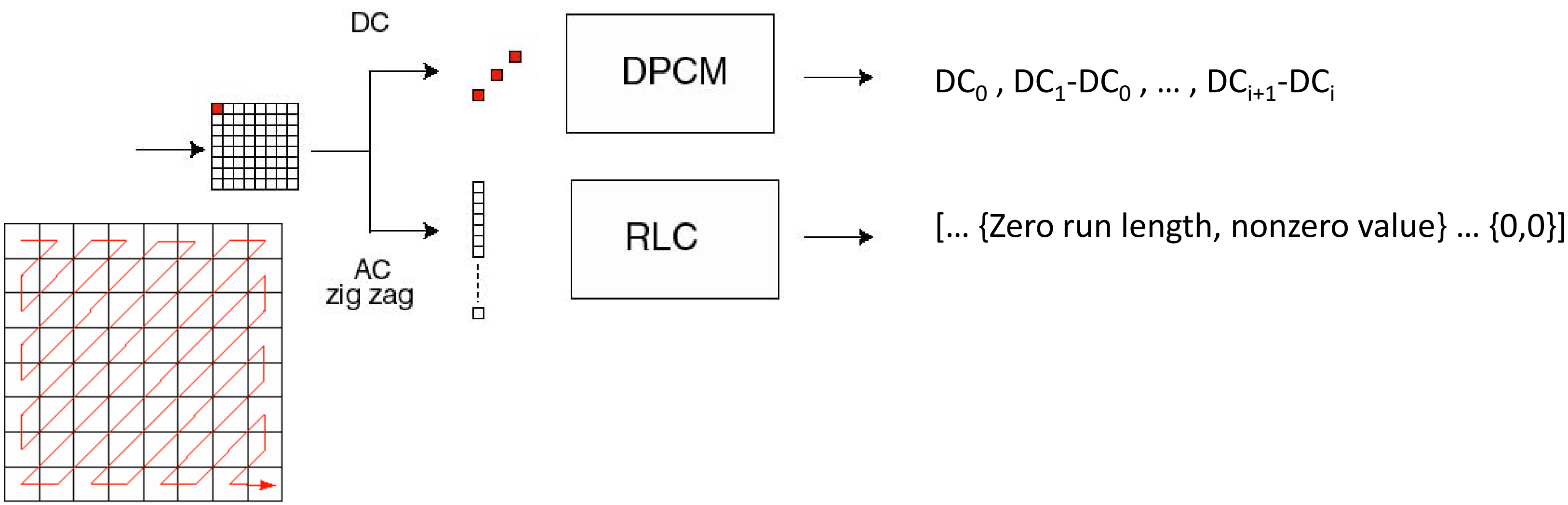
- Chrominance

17	18	24	47	99	99	99	99
18	21	26	66	99	99	99	99
24	26	56	99	99	99	99	99
47	66	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99

- O factor de qualidade (JPEG) tem uma relação linear com a matriz Q.
- Os valores da DCT são arredondados.
- Os valores no canto inferior direito da matriz Q são maiores, logo no resultado vêm zeros.

JPEG - compressão

- As tabelas de quantifiacação resultam de estudos psico-visuais, com o objectivo de maximizar a taxa de compressão sem perda perceptiveis de qualidade.



84	5	-3	-1	-2	-3	1	1	-1	-1	0	0	1	2	3	-1	1	1	0	0	0	0	0	0	1	1	0	1

5	{0,5}	{0,-3}	{0,-1}	{0,-2}	{0,-3}	{0,1}	{0,1}	{0,-1}	{0,-1}	{2,1}	{0,2}	{0,3}	{0,-1}	{0,1}	{0,1}	{6,1}	{0,1}	{1,1}	{0,0}

- Nota: Para reduzir o número de bits que representam o DC, subtrai-se ao x(m,n) - bloco 8x8, o valor de 128. Esta operação apenas afecta o DC e não os coeficientes AC.

JPEG - compressão

- Cada código DPCM (coeficiente DC) é representado pelo par {Size, Amplitude}, de acordo com a tabela. O campo Size indica quantos bits codificam o campo Amplitude.
  O campo Size é codificado pelo algoritmo de Huffman.
  (para valores negativos usa-se o complento de 1)

DPCM → {Size,Amplitude}

SIZE	AMPLITUDE
1	-1,1
2	-3,-2,2,3
3	-7..-4,4..7
4	-15..-8,8..15
5	-31..-16,16..31
6	-63..-32,32..63
7	-127..-64,64..127
8	-255..-128,128..255
9	-511..-256,256..511
10	-1023..-512,512..1023

JPEG - compressão

- Cada par {Zero run length,non zero value} - (coeficiente AC), o campo *non-zero-value* é codificado como o DC com {size, amplitude}

  Os campos Zero-run-length e Size ocupam 4 bits cada e são codificados pelo algoritmo de Huffman.

  O campo amplitude não é codificado (ocupa 8bits)
- A codificação aritmética é também suportada na norma JPEG em alternativa ao código de Huffman.

RLC → {Zero run length, nonzero value}

5	{0,5}	{0,-3}	{0,-1}	{0,-2}	{0,-3}	{0,1}	{0,1}	{0,-1}	{0,-1}
{3,5}	{0,3}{5}	{0,2}{-3}	{0,1}{-1}	{0,2}{-2}	{0,2}{-3}	{0,1}{1}	{0,1}{1}	{0,1}{-1}	{0,1}{-1}

{2,1}	{0,2}	{0,3}	{0,-1}	{0,1}	{0,1}	{6,1}	{0,1}	{1,1}	{0,0}
{2,1}{1}	{0,2}{2}	{0,2}{3}	{0,1}{-1}	{0,1}{1}	{0,1}{1}	{2,1}{1}	{0,1}{1}	{1,1}{1}	{0,0}

JPEG – compressão (blocos Lena)

```
[[ 162.  162.  162.  161.  162.  157.  163.  161.]
 [ 162.  162.  162.  161.  162.  157.  163.  161.]
 [ 162.  162.  162.  161.  162.  157.  163.  161.]
 [ 162.  162.  162.  161.  162.  157.  163.  161.]
 [ 162.  162.  162.  161.  162.  157.  163.  161.]
 [ 164.  164.  158.  155.  161.  159.  159.  160.]
 [ 160.  160.  163.  158.  160.  162.  159.  156.]
 [ 159.  159.  155.  157.  158.  159.  156.  157.]]
```

```
[[ 166.  162.  162.  160.  155.  163.  160.  155.]
 [ 166.  162.  162.  160.  155.  163.  160.  155.]
 [ 166.  162.  162.  160.  155.  163.  160.  155.]
 [ 166.  162.  162.  160.  155.  163.  160.  155.]
 [ 166.  162.  162.  160.  155.  163.  160.  155.]
 [ 161.  160.  155.  159.  154.  154.  156.  154.]
 [ 159.  163.  158.  163.  155.  155.  156.  152.]
 [ 159.  162.  162.  160.  153.  153.  153.  151.]]
```

```
[[ 157.  156.  161.  161.  154.  156.  154.  157.]
 [ 157.  156.  161.  161.  154.  156.  154.  157.]
 [ 157.  156.  161.  161.  154.  156.  154.  157.]
 [ 157.  156.  161.  161.  154.  156.  154.  157.]
 [ 157.  156.  161.  161.  154.  156.  154.  157.]
 [ 156.  156.  153.  157.  152.  153.  153.  150.]
 [ 153.  160.  154.  154.  158.  150.  155.  152.]
 [ 155.  154.  156.  153.  155.  153.  152.  152.]]
```

```
[[   1283.5   4.7   3.2  -0.1   0.2  -0.5  -0.4   0.5]
 [      7.9  -0.7   0.5  -4.9   1.9   2.9  -3.7   3.3]
 [     -5.0  -0.2  -1.5   1.7  -0.6  -0.4   1.8  -2.2]
 [      2.2   1.1   1.7   0.9  -0.7  -1.3   0.2   1.1]
 [     -1.0  -1.1  -3.4  -1.3   1.7   1.1  -1.4  -0.6]
 [      1.2   0.4  -1.8  -0.1  -2.0   0.7   1.6   0.7]
 [     -1.7   0.2   3.0   1.6   1.6  -2.2  -1.1  -0.9]
 [      1.3  -0.4  -2.3  -1.5  -0.8   1.9   0.5   0.6]]
```

```
[[  80.   0.   0.   0.   0.   0.   0.   0.]
 [   1.   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.   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.]]
```

```
[[  80.   2.   0.   0.   0.   0.   0.   0.]
 [   1.   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.   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.]]
```

```
[[  78.   1.   0.  -1.   0.   0.   0.   0.]
 [   1.   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.   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.]]
```

```
DC  = [ 80.
AC  = [[(1, 1), (0, 0)]

DC  = {7bit, 1010000}
DC  = {11110, 1010000}

AC  = {1,1} 1    {0,0}
AC  = 1100  1    1010
```

```
0.
[(0, 2), (0, 1), (0, 0)]

{0bit}
{00}

{0,2} 2   {0,1} 1 {0,0}
01    10  00    1 1010
```

```
 -2.]
[(0, 1), (0, 1), (3, -1), (0, 0)]

{2bit, - 10}
{011, 01}

{0,1} 1 {0,1} 1 {3,1}  0 {0,0}
00    1 00    1 111010 0 1010
```

DC AC AC ... EOB DC AC AC ... EOB ...

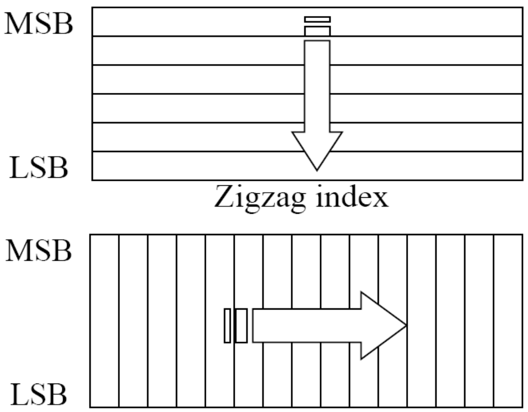
11110 1010000 1100 1 1010 00 01 10 00 1 1010 011 01 00 1 00 1 111010 0 1010

57 bit para codificar 3 blocos de 8x8 pixeis (com 8bit por pixels)
Taxa de Compressão de 27x

Modos JPEG

- Sequencial:
  Cada componente é analisada da esquerda para a direita e de cima para baixo.

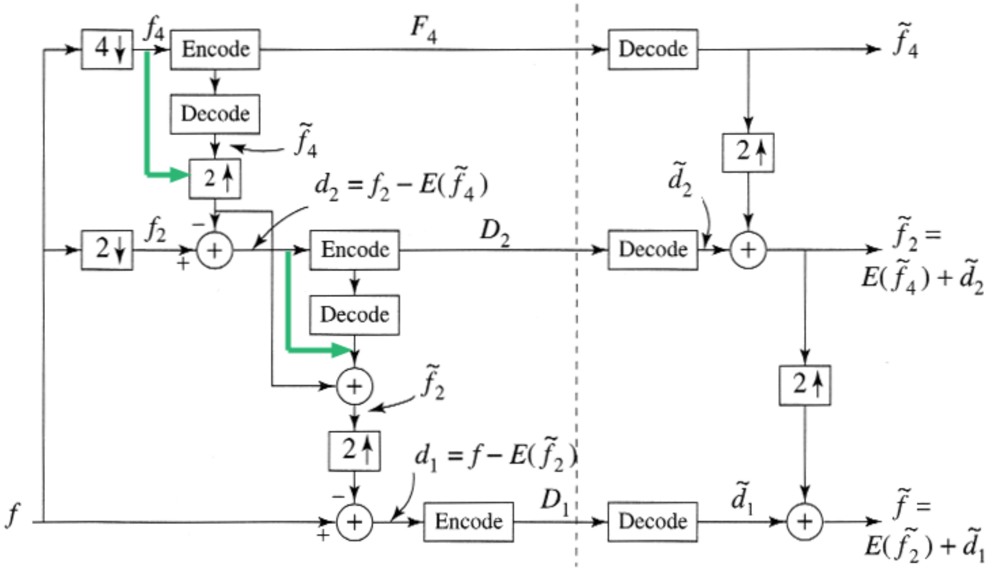
- Progressivo:
  Neste modo dá-se relevância à ordem como se envia os dados:
  - A proximações sucessivas: do MSB para o LSB
  - Selecção espectral: Do DC e primeiros AC Para os últimos AC.



Modos JPEG

- Hierarquico:
  - Cada nível tem uma resolução espacial maior permitindo a codificação de mais detalhes.

  Exemplo com 3 níveis:

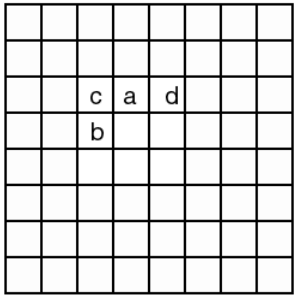


Qualidade/taxa de compressão JPEG

- 0.25 – 0.5 bits/pixel : qualidade moderada a boa
- 0.5 – 0.75 bits /pixel: qualidade boa a muito boa
- 0.75 – 1.5 bits/pixel: Qualidade excelente
- 1,5 – 2 bits/pixel: Indistinguível do original

Modos JPEG

- Reversível (Lossless)
  - Usado em imagens médicas ou de dificil aquisição
  - Baixa taxa de compressão
  - Algoritmo de baixa complexidade
  - Não usa a DCT, usa um modelo preditivo
    - Para cada pixel são aplicados oito modos, sendo selecionada o que dá menor erro.



selection-value	prediction
0	no prediction
1	A
2	B
3	C
4	A+B-C
5	A+((B-C)/2)
6	B+((A-C)/2)
7	(A+B)/2