



JPEG Image Compression

André Damas Mora
FCT–UNL
2025–2026

Compressão JPEG (Joint Photographic Experts Group)

- ▶ A compressão JPEG funciona bem em imagens naturais, mas mal em letras, cartoons ou desenhos com linhas.



Compressão JPEG (Joint Photographic Experts Group)

- ▶ Publicado a primeira vez em 1992
- ▶ Atinge com facilidade rácios de compressão de 20:1 em imagens a cores
- ▶ Uma desvantagem real é a acumulação de perdas de cada vez que se descomprime, altera e volta a comprimir (a perda é pequena mas existe)
- ▶ A norma JPEG é composta por três partes:
 - CODIFICADOR – Recebe como input a imagem digital a codificar e os parâmetros da codificação a efetuar e gera a informação sobre a imagem comprimida.
 - DESCODIFICADOR – Recebe como input a informação sobre a imagem comprimida e os parâmetros de descodificação e gera a imagem descodificada.
 - FORMATO DE TRANSPORTE – especifica o formato dos ficheiros de suporte às imagens comprimidas. Este formato permite a transferência de informação comprimida entre aplicações em plataformas heterogéneas.

Compressão JPEG – II

► Contempla quatro formatos de JPEG:

- **Baseline JPEG** – 24 bits/pixel
 - mais básico e comum
- **Extended JPEG** – 36 bits/pixel,
 - opção codificação Aritmética/Huffman,
 - quantificação adaptativa,
 - formato SPIFF (Still Picture Interchange File Format)
 - refinamento seleccionável
- **Lossless JPEG** – 24 / 36 bits
 - obtido efectuando codificação DPCM + Aritmética/Huffman.
- **Progressive**
 - Detalhe progressivo

Progressive JPEG

- ▶ O “baseline” JPEG efectua uma única passagem top-down sobre a imagem.
- ▶ O “progressive” JPEG divide o ficheiro numa série de passagens.
 - A primeira passagem mostra a imagem com baixa qualidade, ocupando pouco espaço.
 - As passagens posteriores vão aumentando a qualidade até ser atingida a mesma qualidade obtida pela codificação “baseline”.
- ▶ A vantagem é permitir visualizar uma imagem ao mesmo tempo que é recebida e ir ganhando detalhe ao longo do tempo.
 - A difusão da Internet de transmissão lenta através de modems e o aumento de capacidade computacional dos computadores fizeram com que a sua utilização aumentasse significativamente em browsers.

Baseline vs Progressive

Baseline JPEG

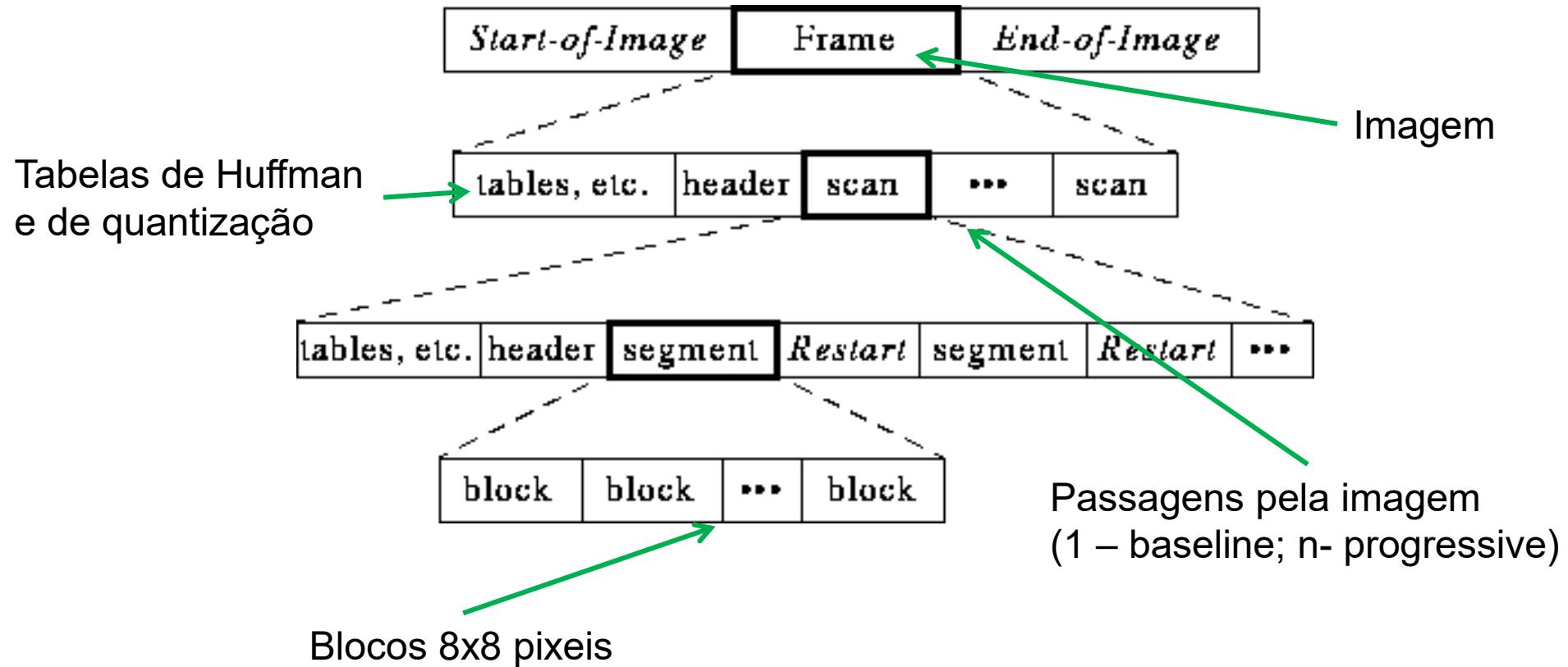


Progressive JPEG



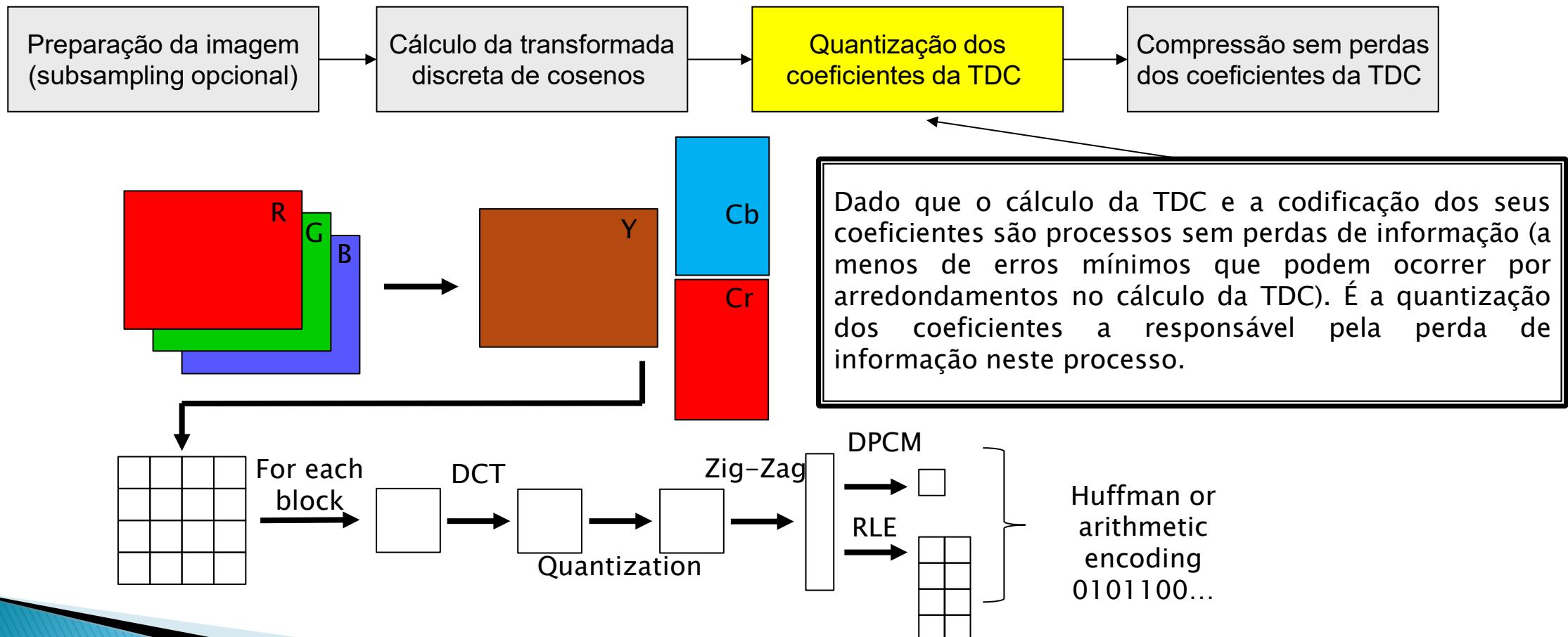
IONOS

Organização do Ficheiro



Compressão JPEG – O algoritmo

- A compressão JPEG com perdas é composta por quatro passos principais:



1 – O espaço de cor utilizado

- ▶ A codificação JPEG utiliza o espaço de cor YCbCr tal como definido pela recomendação CCIR 601 (256 níveis).
Y – Luminância (brightness) Cb e Cr – crominância blue e red
- ▶ Os níveis YCbCr (256 níveis) podem ser calculados diretamente dos valores RGB da seguinte forma:

$$Y = 0.299 R + 0.587 G + 0.114 B$$

$$Cb = -0.1687 R - 0.3313 G + 0.5 B + 128$$

$$Cr = 0.5 R - 0.4187 G - 0.0813 B + 128$$

- ▶ A conversão inversa para RGB é efetuada com base em:

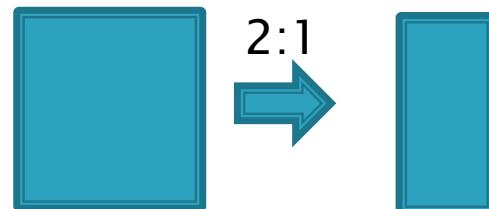
$$R = Y + 1.402 (Cr-128)$$

$$G = Y - 0.34414 (Cb-128) - 0.71414 (Cr-128)$$

$$B = Y + 1.772 (Cb-128)$$

2 – Processo de “subsampling”

- ▶ O processo de “subsampling” é opcional.
- ▶ O subsampling efetua a redução 2:1 do número de pixels nas componentes Cb e Cr fazendo a média de cada dois pixels adjacentes. A componente de luminosidade Y é mantida inalterada.
- ▶ Dado que na imagem original cada componente representa 1/3 da informação total, ao reduzir-se duas delas para metade teremos $1/3 + 1/6 + 1/6 = 2/3$, sem que seja notória degradação significativa na qualidade da imagem.
- ▶ Este processo é efetuado opcionalmente como pré-processamento da informação a codificar não influenciando de qualquer forma a compressão efetuada pelo processo JPEG.



3–Transformada discreta de cosenos

- ▶ Dá uma representação da imagem no domínio da frequência
 - boa para poucos coeficientes

- ▶ Transformada directa

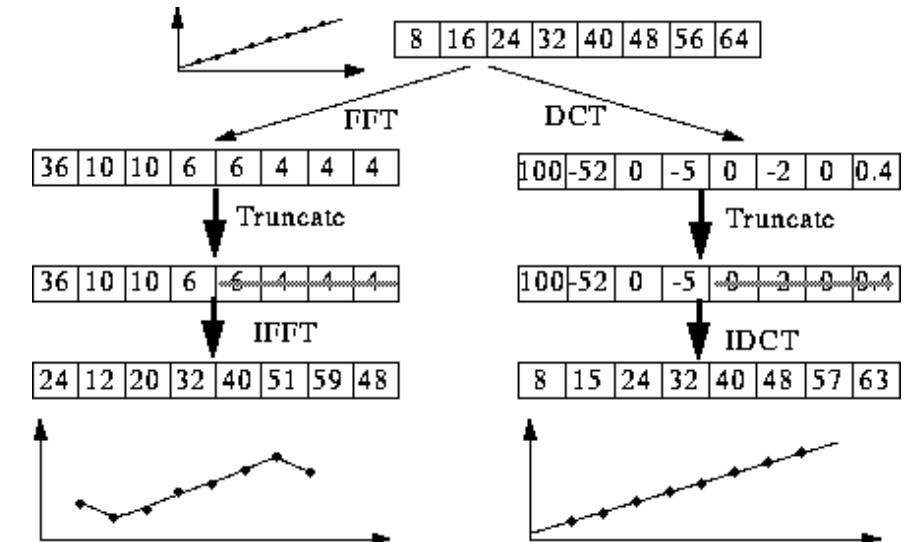
$$\mathfrak{I}(u, v) = \frac{2}{N} C(u) C(v) \sum_{j=0}^{N-1} \sum_{k=0}^{N-1} F(j, k) \cos\left(\frac{\pi}{N} u \left(j + \frac{1}{2}\right)\right) \cos\left(\frac{\pi}{N} v \left(k + \frac{1}{2}\right)\right)$$

- ▶ Transformada inversa

$$F(u, v) = \frac{2}{N} \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} C(u) C(v) \mathfrak{I}(u, v) \cos\left(\frac{\pi}{N} u \left(j + \frac{1}{2}\right)\right) \cos\left(\frac{\pi}{N} v \left(k + \frac{1}{2}\right)\right)$$

onde

$$C(w) = \begin{cases} \frac{1}{\sqrt{2}} & \text{se } w = 0 \\ 1 & \text{se } w \neq 0 \end{cases}$$



3–Transformada discreta de cosenos

10	10	10	10	10	11	12	13
10	10	10	10	10	11	12	13
10	10	10	10	10	11	12	13
10	10	10	10	10	11	12	13
11	11	11	11	11	11	12	13
11	12	12	12	12	12	13	14
13	13	13	13	13	13	14	15
14	14	14	14	14	14	15	16

DCT

```
function C = dct(F)
for u=1:8
for v=1:8
aux=0
for j=1:8
for k=1:8
aux=aux+F(j,k)*cos(pi/8*(u-1)*(j-1+0.5))*cos(pi/8*(v-1)*(k-1+0.5))
end
end
if (u==1) cu=1/sqrt(2); else cu=1; end
if (v==1) cv=1/sqrt(2); else cv=1; end
C(u,v)=cu*cv*aux/4;
end
end
```

Transformada
Directa de cosenos

Claro que este código
pode ser optimizado!

Valor DC
Transformada
inversa

```
function R = idct(C,TabCos,cof)
for j=1:8
for k=1:8
aux=0
for u=1:8
for v=1:8
if (u==1) cu=1/sqrt(2); else cu=1; end
if (v==1) cv=1/sqrt(2); else cv=1; end
aux=aux+cu*cv*C(u,v)*cos(pi/8*(u-1)*(j-1+0.5))*cos(pi/8*(v-1)*(k-1+0.5))
end
end
R(j,k)=aux/4
end
end
```

94.50	-5.62	3.81	-1.83	0.50	-0.02	0.05	-0.01
-9.56	-1.52	0.59	0.22	-0.45	0.15	0.25	-0.30
4.46	0.00	0.00	0.00	0.00	0.00	0.00	0.00
-0.79	0.53	-0.21	-0.08	0.16	-0.05	-0.09	0.11
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
-0.39	-0.36	0.14	0.05	-0.11	0.03	0.06	-0.07
0.32	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.02	0.30	-0.12	-0.04	0.09	-0.03	-0.05	0.06

Valores AC

Claro que este código
pode ser optimizado!

IDCT

10	10	10	10	10	11	12	13
10	10	10	10	10	11	12	13
10	10	10	10	10	11	12	13
10	10	10	10	10	11	12	13
11	11	11	11	11	11	12	13
12	12	12	12	12	12	13	14
13	13	13	13	13	13	14	15
14	14	14	14	14	14	15	16

4–Quantização dos coeficientes

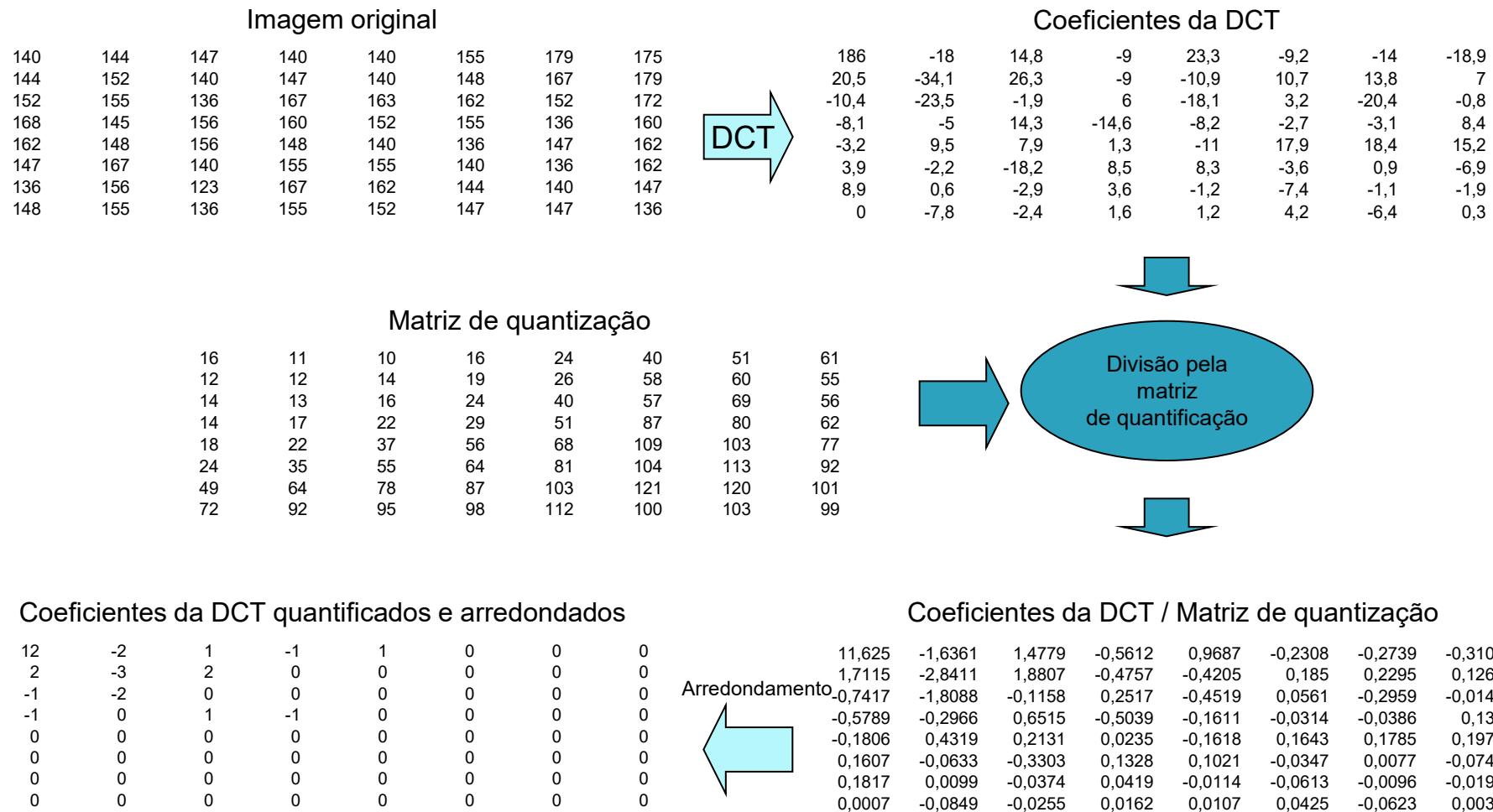
- ▶ A quantização dos coeficientes é efetuada dividindo os coeficientes da DCT pela matriz de quantização e arredondando os valores resultantes para valores inteiros.
 → É aqui que as perdas se vão notar.
- ▶ A matriz de quantização normalmente utilizada pela compressão JPEG para a luminância é a seguinte:

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



- ▶ Esta tabela é guardada nos ficheiros JPEG de forma a que o codificador a possa ajustar a imagens particulares. É também frequente utilizar-se uma tabela para a componente Y distinta das utilizadas para as componentes Cb e Cr.

Consequências da quantificação – I



Consequências da quantificação – II

Coeficientes da DCT quantificados e arredondados

12	-2	1	-1	1	0	0	0
2	-3	2	0	0	0	0	0
-1	-2	0	0	0	0	0	0
-1	0	1	-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

multiplicar pela
matriz
de
quantificação

Matriz de quantização

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



Coeficientes da DCT “restaurados”

192	-22	10	-16	24	0	0	0
24	-36	28	0	0	0	0	0
-14	-26	0	0	0	0	0	0
-14	0	22	-29	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

IDCT

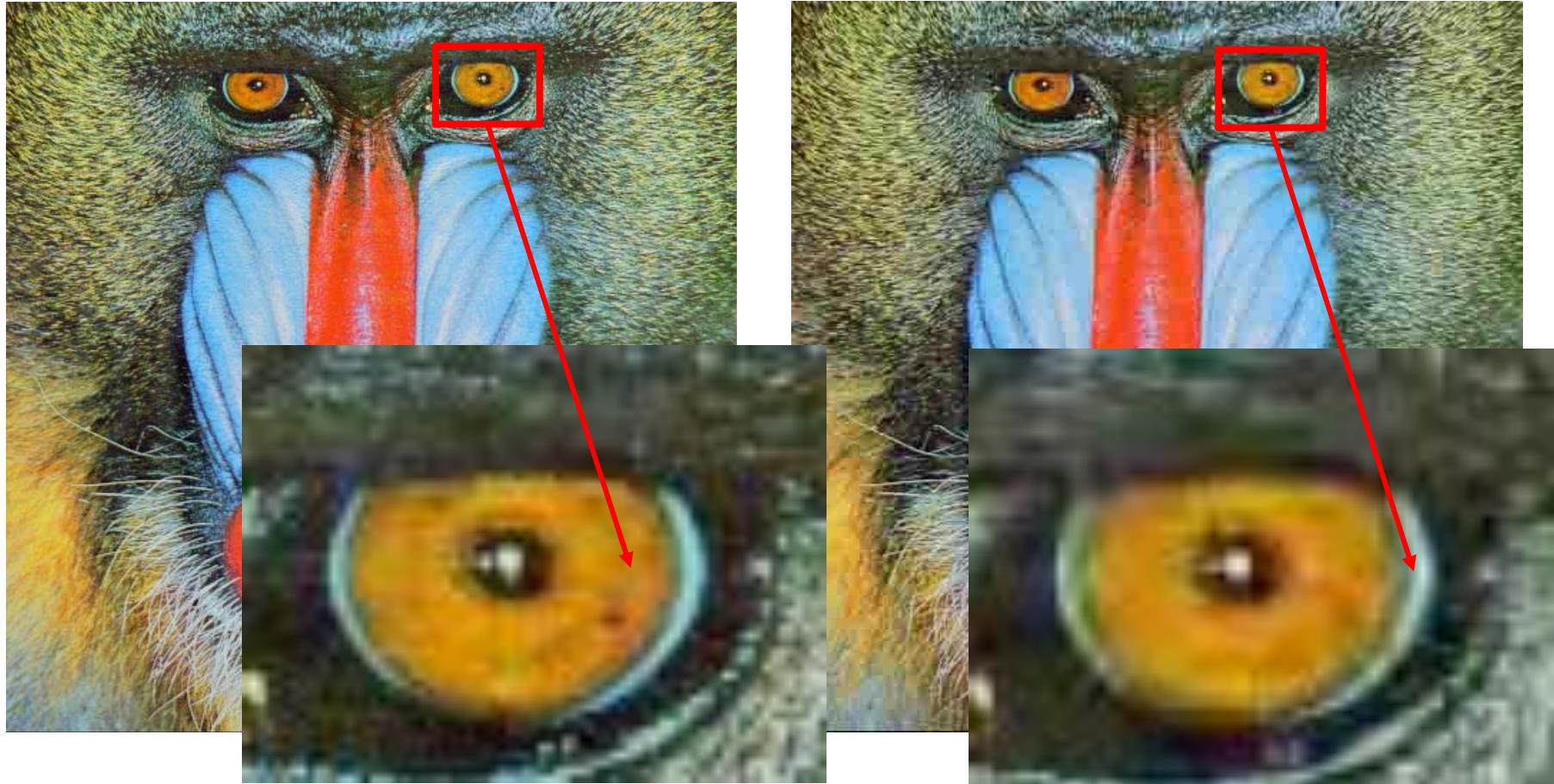
Imagen resultante

141	140	142	144	141	145	168	193
149	143	143	150	154	156	166	179
158	149	147	157	164	162	161	164
161	154	152	159	160	154	153	158
156	155	157	156	147	139	146	160
149	151	156	154	142	132	141	159
145	145	150	155	150	140	141	150
143	139	145	158	162	152	142	140

Imagen original

140	144	147	140	140	155	179	175
144	152	140	147	140	148	167	179
152	155	136	167	163	162	152	172
168	145	156	160	152	155	136	160
162	148	156	148	140	136	147	162
147	167	140	155	155	140	136	162
136	156	123	167	162	144	140	147
148	155	136	155	152	147	147	136

Artefactos

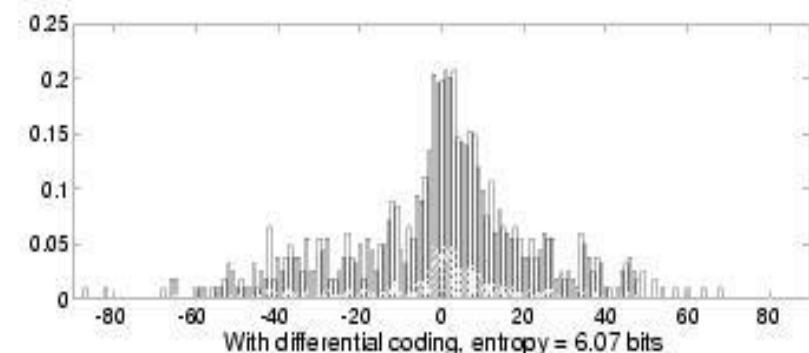
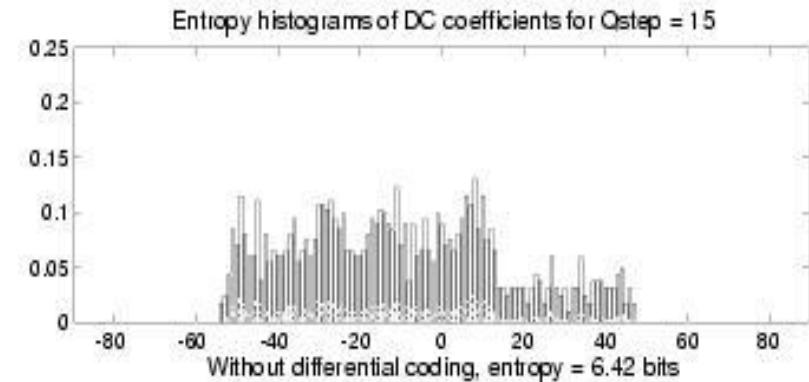


Original

JPEG Compressed

5–Codificação dos coeficientes DC da DCT

- ▶ A codificação dos coeficientes DC da DCT é efectuada seguindo os seguintes passos:
 - Dado que existe normalmente uma grande relação entre blocos consecutivos o valor DC é codificado como a diferença para o último valor DC transmitido – DPCM
 - Os blocos são codificados da esquerda para a direita, de cima para baixo
 - O primeiro bloco é codificado relativamente a zero
- ▶ A codificação diferencial DPCM permite concentrar o histograma de valores DC em torno do zero baixando a sua entropia como se mostra nas figuras ao lado



Codificar blocos consecutivos

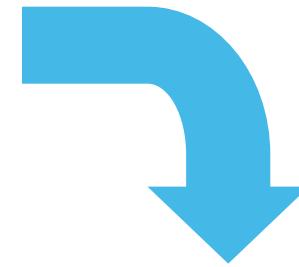
72	22	-5	-3	1	0	0	0
2	1	-3	1	0	0	0	0
3	-5	0	2	-2	1	0	0
3	-3	0	0	0	0	0	0
-3	1	1	-1	0	0	0	0
1	-1	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Bloco (0,0)

63	-10	1	1	0	0	0	0
6	1	-2	1	0	0	0	0
-9	-3	3	1	0	0	0	0
6	2	0	1	0	0	0	0
-2	1	0	0	0	0	0	0
1	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Bloco (1,0)

Subtrair
Componente DC
do bloco (0,0)
 $63 - 72 = -9$



-9	-10	1	1	0	0	0	0
6	1	-2	1	0	0	0	0
-9	-3	3	1	0	0	0	0
6	2	0	1	0	0	0	0
-2	1	0	0	0	0	0	0
1	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Tabela de codificação de valores DC

- ▶ Nesta codificação é enviado o número de bits do valor utilizando a codificação de Huffman e o código do valor como additional bits
- ▶ Exemplo:
 - Valor = -8 size = 4 additional bits = 4, ou seja, código = 101 0111

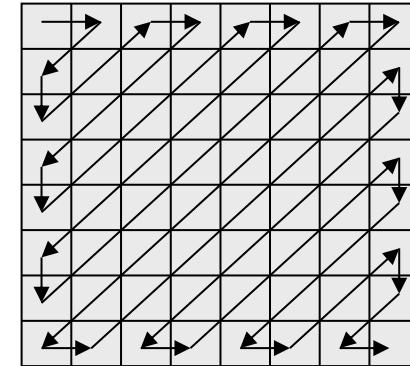
Size	Code length	Code word
0	2	00
1	3	010
2	3	011
3	3	100
4	3	101
5	3	110
6	4	1110
7	5	11110
8	6	111110
9	7	1111110
10	8	11111110
11	9	111111110

DC Value	Additional Bits
0	
-1	0 1
-3,-2	00,01
2,3	10,11
-7,-6,-5,-4	000,001,010,011
4,5,6,7	100,101,110,111
-15,...,-8	0000,...,0111
8,...,15	1000,...,1111
-31,...,-16	0 0000,...,1 1111
16,...,31	
-63,...,-32	00 0000,...,11 1111
32,...,63	
-127,...,-64	000 0000,...,111 1111
64,...,127	
-255,...,-128	0000 0000,...,1111 1111
128,...,255	
-511,...,-256	0 0000 0000,...,1 1111 1111
256,...,511	
-1023,...,-512	00 0000 0000,...,11 1111 1111
512,...,1023	
-2047,...,-1024	000 0000 0000,...,111 1111 1111
1024,...,2047	

5–Codificação dos restantes 63 coeficientes AC

► A codificação dos restantes 63 coeficientes da DCT é efetuada seguindo os passos:

- Reordenação dos coeficientes em zig-zag
 - Permite aumentar as sequências de zeros.
 - As frequências mais baixas (normalmente mais importantes) são portanto enviadas primeiro.
 - Adequada para a codificação RLE (Run-Length Encoding)



Esta mesma matriz é utilizada na codificação de vídeo H263

- Codificação dos coeficientes utilizando uma técnica baseada na entropia. O resultado desta codificação é uma sequência de blocos de três valores que contêm:
 - Run – o número de zeros consecutivos que antecedem o coeficiente não nulo transmitido;
 - Size – o número de bits utilizados para codificar a amplitude do valor, tal como determinado pela codificação de Huffman;
 - Value – a amplitude do coeficiente.

(Run, size) Value

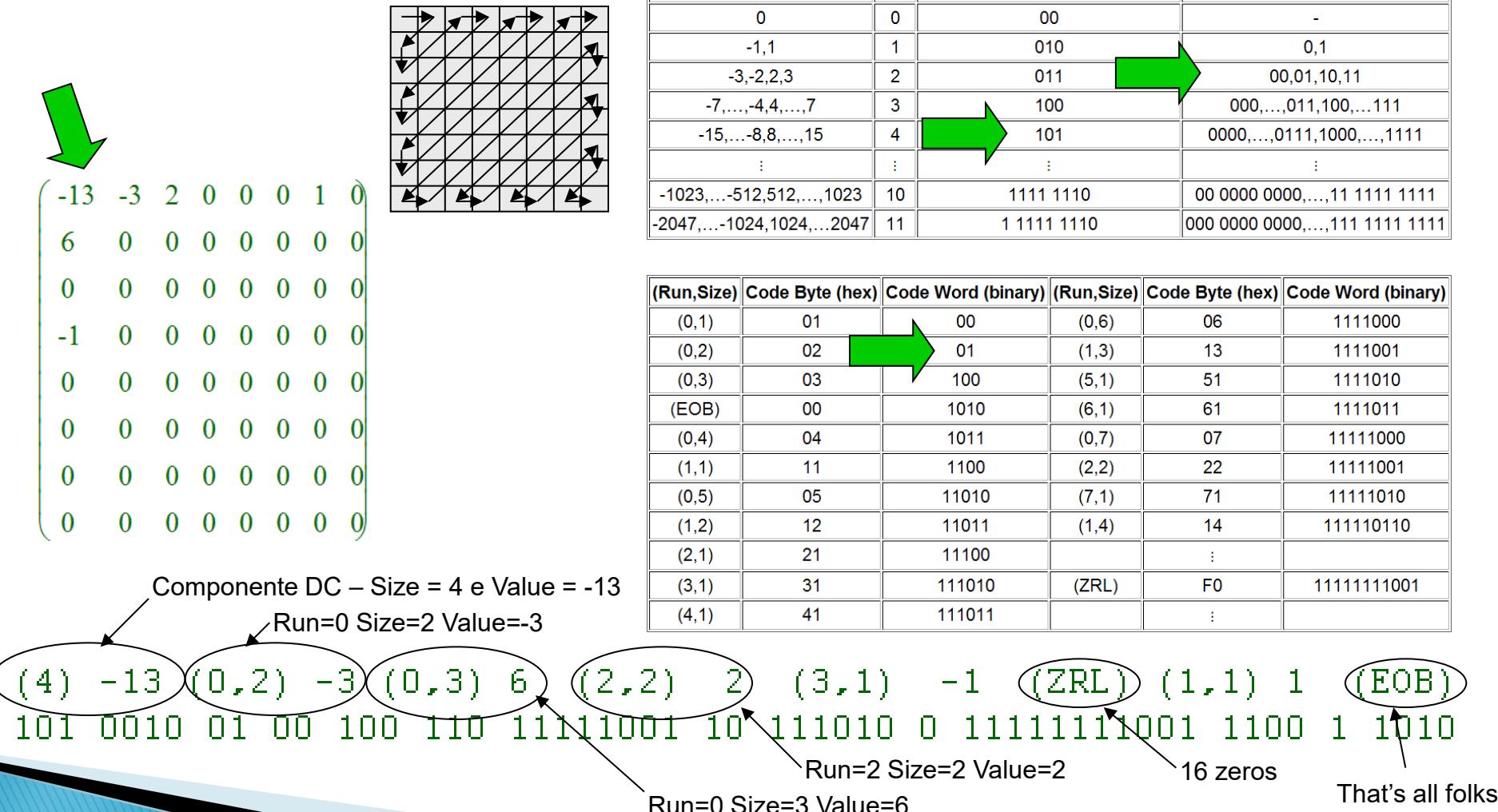
Codificação AC – run lenght encoding

- ▶ Utilizada para os restantes 63 coeficientes AC
- ▶ É composta por sequências run-amplitude pois existem normalmente muitas sequências de zeros
- ▶ Só os valores de Run=1 → 15 e Size=1 → 10 são codificados
- ▶ São usados dois códigos extra:
 - (Run,Size)=(0,0) – EOB (End Of Block) – fim de codificação
 - (Run, Size) = (15,0) – ZRL (Zero Run Length) – utilizado sempre que surgem mais de 15 zeros (15 zeros seguidos de zero = 16 zeros)

Códigos JPEG para codificar componente AC (Run,Size) dos coeficientes DCT da luminância

(Run,Size)	Code Byte (hex)	Code Word (binary)	(Run,Size)	Code Byte (hex)	Code Word (binary)
(0,1)	01	00	(0,6)	06	1111000
(0,2)	02	01	(1,3)	13	1111001
(0,3)	03	100	(5,1)	51	1111010
(EOB)	00	1010	(6,1)	61	1111011
(0,4)	04	1011	(0,7)	07	11111000
(1,1)	11	1100	(2,2)	22	11111001
(0,5)	05	11010	(7,1)	71	11111010
(1,2)	12	11011	(1,4)	14	111110110
(2,1)	21	11100		:	
(3,1)	31	111010	(ZRL)	F0	11111111001
(4,1)	41	111011		:	

Codificação de um bloco- exemplo



Como ajustar a Compressão?

- ▶ Decimação das componentes Cr e Cb
 - usar / não usar
- ▶ Alterar a matriz de quantização
 - Outras matrizes
 - Aumentar ou diminuir a amplitude da matriz de quantização com um factor multiplicativo:

$$Q' = Q * \frac{100}{factor}$$

100 – menor compressão
1 – maior compressão

Diferentes níveis de compressão



Compressão mínima



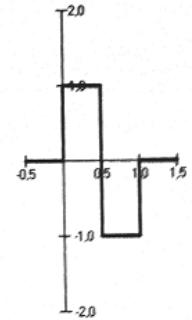
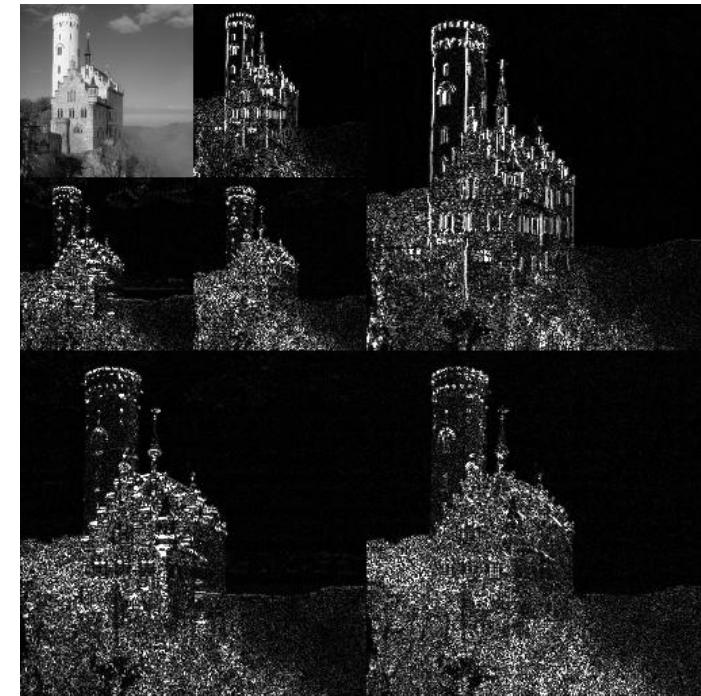
Compressão máxima

Lossless JPEG

- ▶ A codificação Lossless JPEG foi criada depois do JPEG, em 1993
- ▶ Depois de avaliados diversos algoritmos, o algoritmo LOCO dos HP Labs foi o escolhido (<http://www.hpl.hp.com/loco/>)
- ▶ Nunca obteve um êxito muito significativo embora seja usado com alguma frequência na área médica
- ▶ As técnicas em que se baseia são diferentes do JPEG
- ▶ Usa codificação DPCM a partir dos vizinhos de cada pixel em vez da DCT

JPEG 2000

- ▶ Foi lançado em 2000 para evoluir o standard JPEG (1992)
- ▶ Em vez da DCT usa Transformada Wavelet
- ▶ Extensão é .jp2 ou .jpx
- ▶ Vantagens:
 - Melhor compressão
 - Representação em múltiplas resoluções
 - Progressive
 - Com ou sem perdas
 - Robustez a erros de transmissão
 - Suporta transparência



Comparação JPEG/JPEG2000



Exemplos

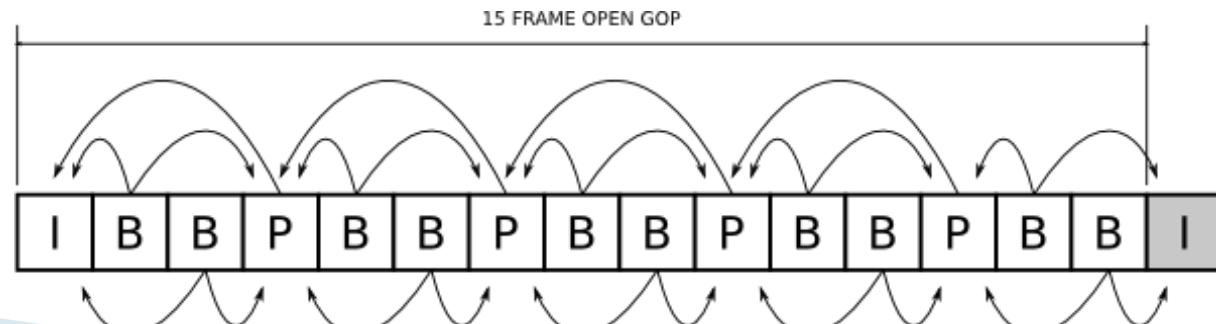
	Size	JPG	JPEG 2000	JPEG2000 Lossless	PNG
	2.8MPixels	632KB	219KB	2.41MB	2.81MB
	25MPixels	16.MB	9MB	28.6MB	16.3MB

MPEG – Compressão de vídeo

- ▶ No MPEG 1 e 2 a compressão é conseguida à custa de 5 técnicas de compressão:
 - Transformada Discreta de Cossenos
 - Quantização
 - Codificação de Huffman
 - Previsão por Compensação de Movimento – são codificadas as diferenças entre frames
 - Codificação bi-direcional – algumas frames são obtidas com informação de frames anteriores e seguintes
- ▶ MPEG-3 é uma extensão dos anteriores
- ▶ MPEG-4 O algoritmo descreve objetos em vez de pixéis

MPEG-2 – Tipos de Frames

- ▶ Um GOP é uma sequência de frames comprimidos que começa com uma frame I.
- ▶ Frame I – contém toda a informação requerida para reconstituir a frame original. Basicamente o formato JPEG –
- ▶ Frame P – codificam o movimento de macro-blocos de 16x16 em relação à frame I ou frame P anterior – compressão 1:2
- ▶ Frame B – possibilitam bidirecionalidade e são obtidas por frames I e P anteriores e posteriores – compressão 1:4



Exemplo de MPEG-4 com falhas

1% average packet loss

Vidyo

H.264/AVC



3Mbps, 5% packet loss in 1 sec bursts every 5 sec