

Compressão com perdas

Codificação por transformadas e
o padrão JPEG



Anderson L Menezes
Leonardo G Tampelini
Maxiwell S Garcia



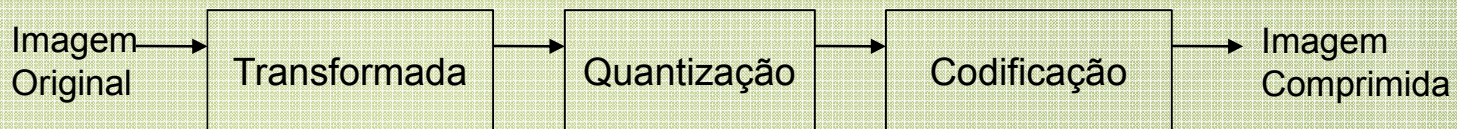
Introdução

- Um método é dito “com perdas” (*lossy data*) quando a informação obtida após a descompressão é diferente da informação original (antes da compressão)
- O resultado é um arquivo comprimido bem menor do que os gerados pelas técnicas sem perdas
- Muitos métodos recorrem às limitações da anatomia humana, explorando determinadas características das imagens
- Necessidade de trocar informações com uma largura de banda limitada e o armazenamento de mais informações em um espaço menor
- Aplica-se em vários tipos de mídia (música, vídeo e **imagens**)



Codificação por Transformadas

Em geral, possuí três partes distintas:



- Transformadas: *os dados são descorrelacionados*
- Quantização: *são eliminados os coeficientes não essenciais para recriar a imagem*
- Codificação: *elimina todos os zeros resultantes e compacta os coeficientes quantizados*



Padrão JPEG

- JPEG (*Joint Photographic Expert Group*) é um padrão altamente utilizado para compressão de imagens, especialmente em imagens fotográficas
- O arquivo final apresenta perdas em relação ao original proporcionais a um coeficiente variável de compressão (armazenado no *header* do arquivo)
- Possui 4 modos de operações definidos:
 - **Sequential DCT-based encoding**
 - Progressive DCT-based encoding
 - Hierarchical coding
 - Lossless encoding
- A implementação do primeiro modo é a mais simples



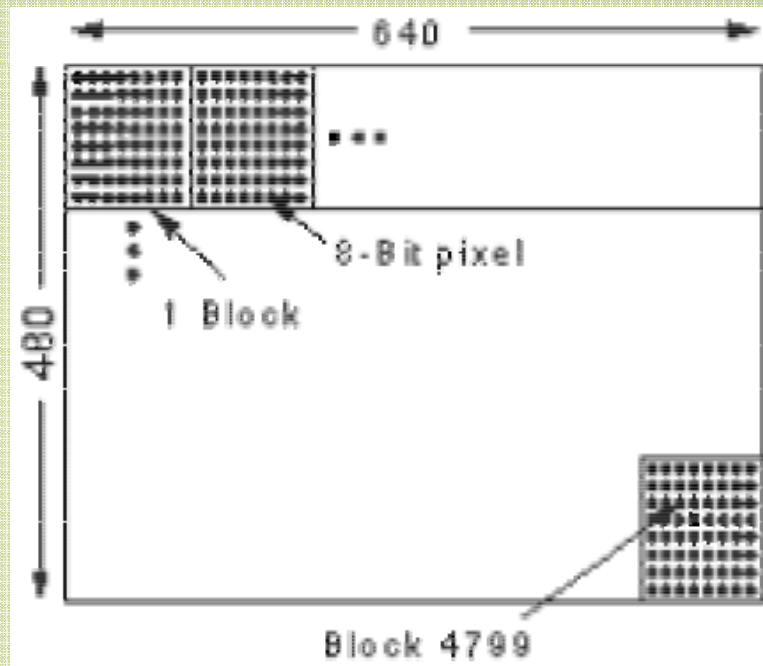
Padrão JPEG

- É dividido em 5 etapas:
 - Transformação de RGB para YCbCr
 - Downsampling
 - Transformada discreta dos Cossenos em 2D (DCT2)
 - Quantização
 - Codificação por entropia
- Toda a imagem a ser comprimida para JPEG é visto, pelo software responsável, como um conjunto de *data-units*, de 8x8 pixel.



Padrão JPEG

- Imagem 640x480, dividida em *data-units*





RGB para YCbCr

- Os componentes RGB apresentam uma relação muito íntima entre si, o que torna inviável o processamento de cada uma das partes isoladas.
- RGB é convertido para um espaço de cores formado por **luminancia** e **crominancias** (as partes não estão intimamente ligadas)
- Luminancia (Y): é a imagem em tons de cinza;
- Crominancias: apresentam as informações de cores de azul (Cb) e de vermelho (Cr).



RGB para YCbCr

- YCbCr (ITU-R BT.601)

$$Y = 0,299R + 0,587G + 0,114B$$

$$Cb = 0,564B - 0,564Y$$

$$Cr = 0,713R - 0,713Y$$



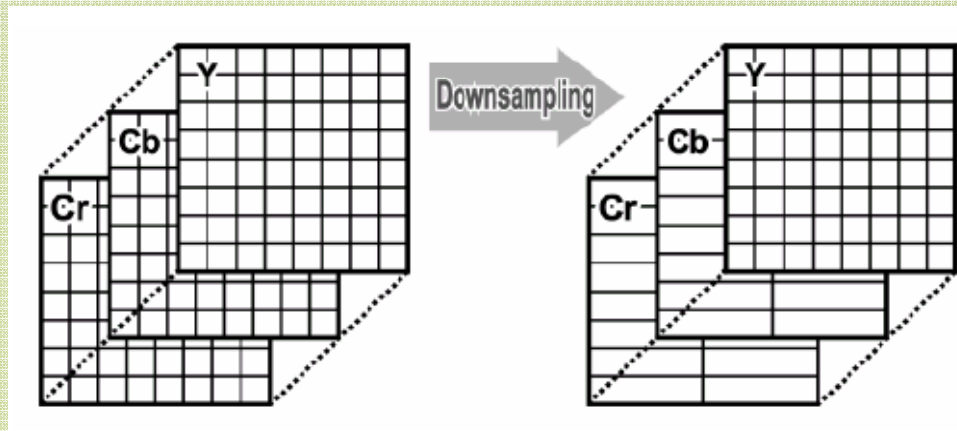
Downsampling

- Inicia-se o processo de redução de dados (perdas irreversíveis)
- O olho humano é mais sensível às informações de luminância do que as de cores.
- Com isso, parte das informações de Cb e Cr podem ser descartadas sem prejuízo aparente
- Especifica-se a proporção:
 - $4:1:1$ = para cada 4 informações de luminância tem-se uma de Cb (azul) e uma de Cr (vermelho)



Downsampling

- Um *data-unit* sofrendo *downsampling*



- Se um *data-unit* tinha o tamanho de $64+64+64 = 192$ bytes agora possui $64+16+16 = 96$ bytes
- Diminuição de 50% com uma perda pouco significativa



DCT

(*Discrete Cosine Transform*)

- Cada bloco é transformado pelo DCT de duas dimensões (DCT2) de maneira independente

$$G_{u,v} = \alpha(u)\alpha(v) \sum_{x=0}^7 \sum_{y=0}^7 g_{x,y} \cos \left[\frac{\pi}{8} \left(x + \frac{1}{2} \right) u \right] \cos \left[\frac{\pi}{8} \left(y + \frac{1}{2} \right) v \right]$$

where

- u is the horizontal spatial frequency, for the integers $0 \leq u < 8$.
- v is the vertical spatial frequency, for the integers $0 \leq v < 8$.

$$\alpha_p(n) = \begin{cases} \sqrt{\frac{1}{8}}, & \text{if } n = 0 \\ \sqrt{\frac{2}{8}}, & \text{otherwise} \end{cases} \quad \text{is a normalizing function}$$

- $g_{x,y}$ is the pixel value at coordinates (x, y)
- $G_{u,v}$ is the DCT coefficient at coordinates (u, v)



DCT

(*Discrete Cosine Transform*)

- Identifica redundâncias presente nos blocos da imagem
- Assim, converte a amplitude espacial dos dados para frequência espacial (*frequency-domain*)
- Esta etapa não tras perda para a imagem
- O resultado desta transformação é um conjunto chamado coeficiente DCT com a **maioria dos valores próximos de zero**



DCT Exemplo

- Um bloco de uma imagem de 8-bit (*grayscale*) a ser compactada



52	55	61	66	70	61	64	73
63	59	55	90	109	85	69	72
62	59	68	113	144	104	66	73
63	58	71	122	154	106	70	69
67	61	68	104	126	88	68	70
79	65	60	70	77	68	58	75
85	71	64	59	55	61	65	83
87	79	69	68	65	76	78	94



DCT Exemplo

- A imagem precisa ser normalizada de -128 à 127 (por ser 8-bit)

52	55	61	66	70	61	64	73
63	59	55	90	109	85	69	72
62	59	68	113	144	104	66	73
63	58	71	122	154	106	70	69
67	61	68	104	126	88	68	70
79	65	60	70	77	68	58	75
85	71	64	59	55	61	65	83
87	79	69	68	65	76	78	94



-76	-73	-67	-62	-58	-67	-64	-55
-65	-69	-73	-38	-19	-43	-59	-56
-66	-69	-60	-15	16	-24	-62	-55
-65	-70	-57	-6	26	-22	-58	-59
-61	-67	-60	-24	-2	-40	-60	-58
-49	-63	-68	-58	-51	-60	-70	-53
-43	-57	-64	-69	-73	-67	-63	-45
-41	-49	-59	-60	-63	-52	-50	-34



DCT Exemplo

- Após a aplicação da transformada DCT2

$$\begin{bmatrix} -415 & -30 & -61 & 27 & 56 & -20 & -2 & 0 \\ 4 & -22 & -61 & 10 & 13 & -7 & -9 & 5 \\ -47 & 7 & 77 & -25 & -29 & 10 & 5 & -6 \\ -49 & 12 & 34 & -15 & -10 & 6 & 2 & 2 \\ 12 & -7 & -13 & -4 & -2 & 2 & -3 & 3 \\ -8 & 3 & 2 & -6 & -2 & 1 & 4 & 2 \\ -1 & 0 & 0 & -2 & -1 & -3 & 4 & -1 \\ 0 & 0 & -1 & -4 & -1 & 0 & 1 & 2 \end{bmatrix}$$



Quantização

- É nesta fase que o tamanho do arquivo diminui drasticamente
- A partir de um coeficiente de compactação, os coeficientes DCT (achados na etapa anterior) são truncados
- São mantidos apenas os mais relevantes
- A grande quantidade de informação perdida nesta fase é irreversível e por isso, uma imagem JPEG possui menos detalhes que a original em TIFF, por exemplo

$$B_{j,k} = \text{round} \left(\frac{G_{j,k}}{Q_{j,k}} \right)$$

G são os coeficientes DCT
Q é a matriz de quantização



Quantização

- O padrão JPEG adota uma matriz de quantização típica:

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

- Porém, outras podem ser utilizadas, posto que no cabeçalho JPEG a matriz de quantização deve ser especificada



Quantização

Exemplo

- Aplicando a quantização no exemplo do DCT

$$G = \begin{bmatrix} -415 & -30 & -61 & 27 & 56 & -20 & -2 & 0 \\ 4 & -22 & -61 & 10 & 13 & -7 & -9 & 5 \\ -47 & 7 & 77 & -25 & -29 & 10 & 5 & -6 \\ -49 & 12 & 34 & -15 & -10 & 6 & 2 & 2 \\ 12 & -7 & -13 & -4 & -2 & 2 & -3 & 3 \\ -8 & 3 & 2 & -6 & -2 & 1 & 4 & 2 \\ -1 & 0 & 0 & -2 & -1 & -3 & 4 & -1 \\ 0 & 0 & -1 & -4 & -1 & 0 & 1 & 2 \end{bmatrix}$$

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

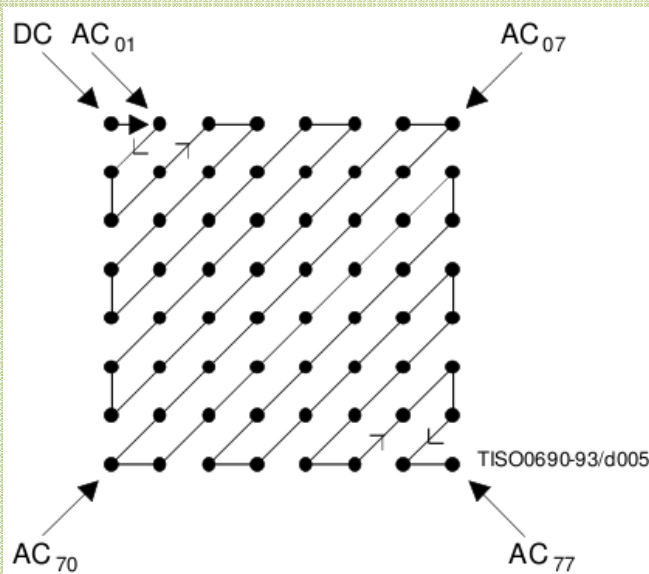
$$B_{j,k} = \text{round} \left(\frac{G_{j,k}}{Q_{j,k}} \right)$$

$$B = \begin{bmatrix} -26 & -3 & -6 & 2 & 2 & -1 & 0 & 0 \\ 0 & -2 & -4 & 1 & 1 & 0 & 0 & 0 \\ -3 & 1 & 5 & -1 & -1 & 0 & 0 & 0 \\ -4 & 1 & 2 & -1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$



Codificação

- Nesta fase, os dados são reordenados seguindo um zig-zag para que os coeficientes de baixa frequência sejam colocados na frente dos de alta frequência



DC

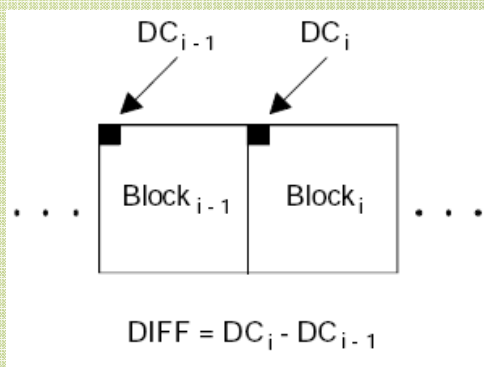
-26	-3	-6	2	2	-1	0	0
0	-2	-4	1	1	0	0	0
-3	1	5	-1	-1	0	0	0
-4	1	2	-1	0	0	0	0
1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

-26								
-3	0							
-3	-2	-6						
2	-4	1	-4					
1	1	5	1	2				
-1	1	-1	2	0	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	0	0	0			
0	0	0	0	0				
0	0	0	0					
0	0	0						
0	0							
0								
0								



Codificação

- O primeiro desses 64 valores é o coeficiente DC: coeficiente para o qual a frequência é zero em ambas as dimensões
- Os outros 63 são os coeficientes AC: componentes de frequência espacial do bloco
- O coeficiente DC do bloco anterior é usado como estimativa para o coeficiente DC atual e apenas a diferença é codificada (codificação diferencial ou DPCM)





Codificação

- Os possíveis zeros consecutivos dos coeficientes AC são agrupados pelo algoritmo de codificação por comprimento de cadeia, o *RUN-Lenght Encoding* (RLE)
- **Até aqui:**
 - o bloco foi organizado linearmente (zig-zag) retirando o DC da primeira posição;
 - a diferença entre o DC do bloco atual e o DC do bloco anterior foi armazenado em uma variável temporária
 - os possíveis zeros dos AC foram agrupados.
- Aplica-se agora a **codificação por entropia**, utilizando a codificação de Huffman ou a codificação aritmética (proprietária).



Codificação por entropia

- A **codificação de Huffman** é a mais utilizada: usa as probabilidades de ocorrência dos símbolos no conjunto de dados a ser comprimido para determinar códigos de tamanho variável para cada símbolo
- Funcionamento deste método:
 - Suponha uma string de dados que possua:
 - 10 A
 - 5 B
 - 5 C



Codificação por entropia

- Fila de prioridades $Q = [c:5, b:5, a:10]$

- Passo 1:

$Q \Rightarrow (10) \quad a:10$

0/ \1

b:5 c:5

- Passo 2:

$Q \Rightarrow (20)$

0/ \1

(10) a:10

0/ \1

b:5 c:5

Tabela de Huffman resultante:

a	1
b	00
c	01



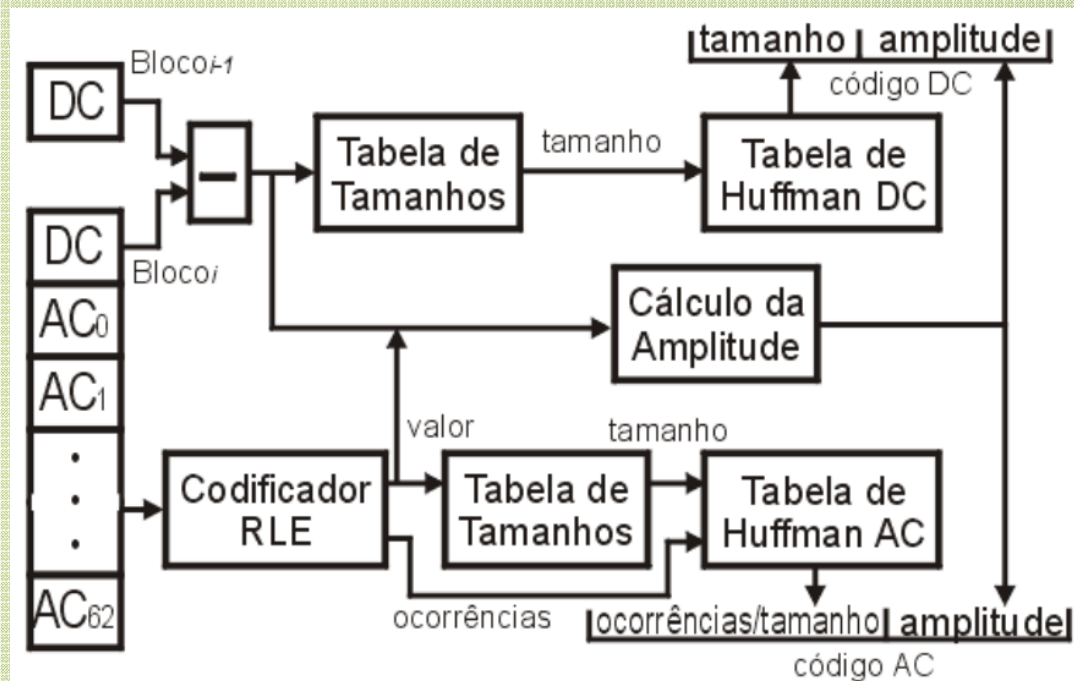
Codificação por entropia

- Por isso, deve-se armazenar no cabeçalho do arquivo JPEG as tabelas de Huffman resultante da codificação:
 - Y/DC
 - CbCr/DC
 - Y/AC
 - CbCr/AC
- Pelo padrão JPEG, os AC e os DC são codificados com tabelas diferentes, e para cada componente (Y, CrCb) existe uma tabela de Huffman apropriada e apenas essa deve ser usada



Codificação por entropia

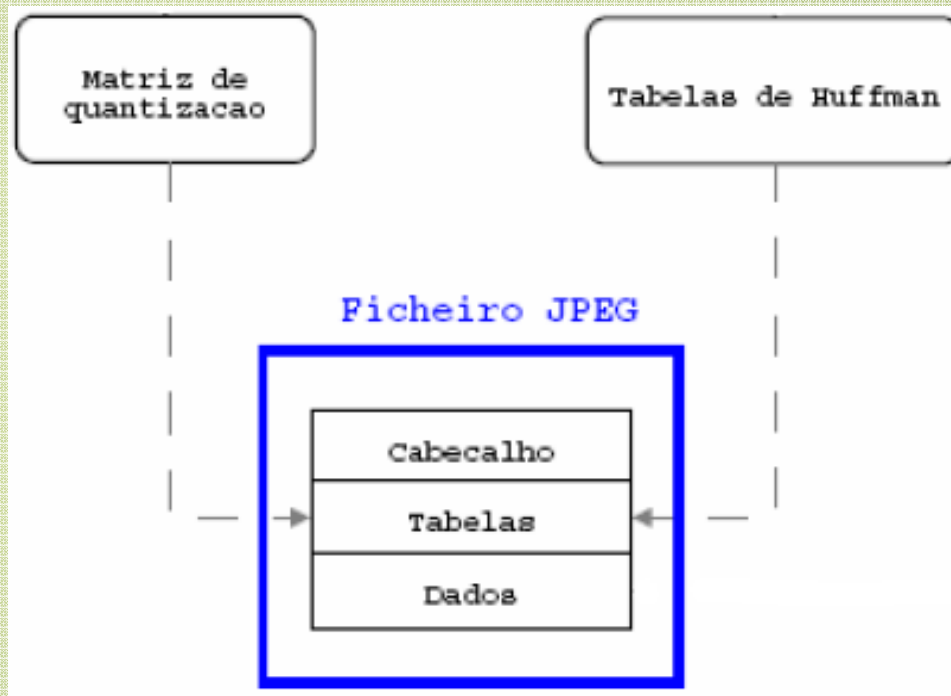
- Detalhe da codificação para uma componente (Y ou CbCr)





Codificação por entropia

- Arquivo JPEG Final





Descodificação

- Dado um arquivo JPEG (codificado), para descodificá-lo tem-se que executar os passos anteriores de forma inversa.
 - Descodifica os dados através das tabelas de Huffman
 - Descodifica os zeros agrupados pelo RLE
 - “Desquantiza” os blocos
 - Aplica a função inversa da transformada DCT2
 - Transforma de YCbCr para RGB



Bibliografia

- MiBench (2008), www.eecs.umich.edu/mibench/, Outubro/2008
- Recommendation T.81 (CCITT), 1992
- Wallace, G. K. "The JPEG Still Picture Compression Standard", 1991
- Agostini, L. V. "Projeto de Arquitetura Integrada de um Compressor de Imagens para JPEG", UFRGS, 2000