



Universidade Federal da Paraíba - UFPB

Sistemas Multimídia

Imagem
Parte 02

Professor: Daniel Faustino

JPEG

Fundamentos de Sistemas Multimídia

JPEG

- ⇒ **Joint Photographic Experts Group**
- ⇒ **Padrão ISO – IS 10918**
- ⇒ **Define vários modos de compressão para uso em diversas aplicações**
 - *Modo seqüencial com perdas – modo básico*
 - Compressão de imagens monocromáticas e coloridas
- ⇒ **5 estágios principais:**
 - *Preparação da imagem/bloco, DCT, quantização, codificação por entropia, construção dos quadros*

JPEG

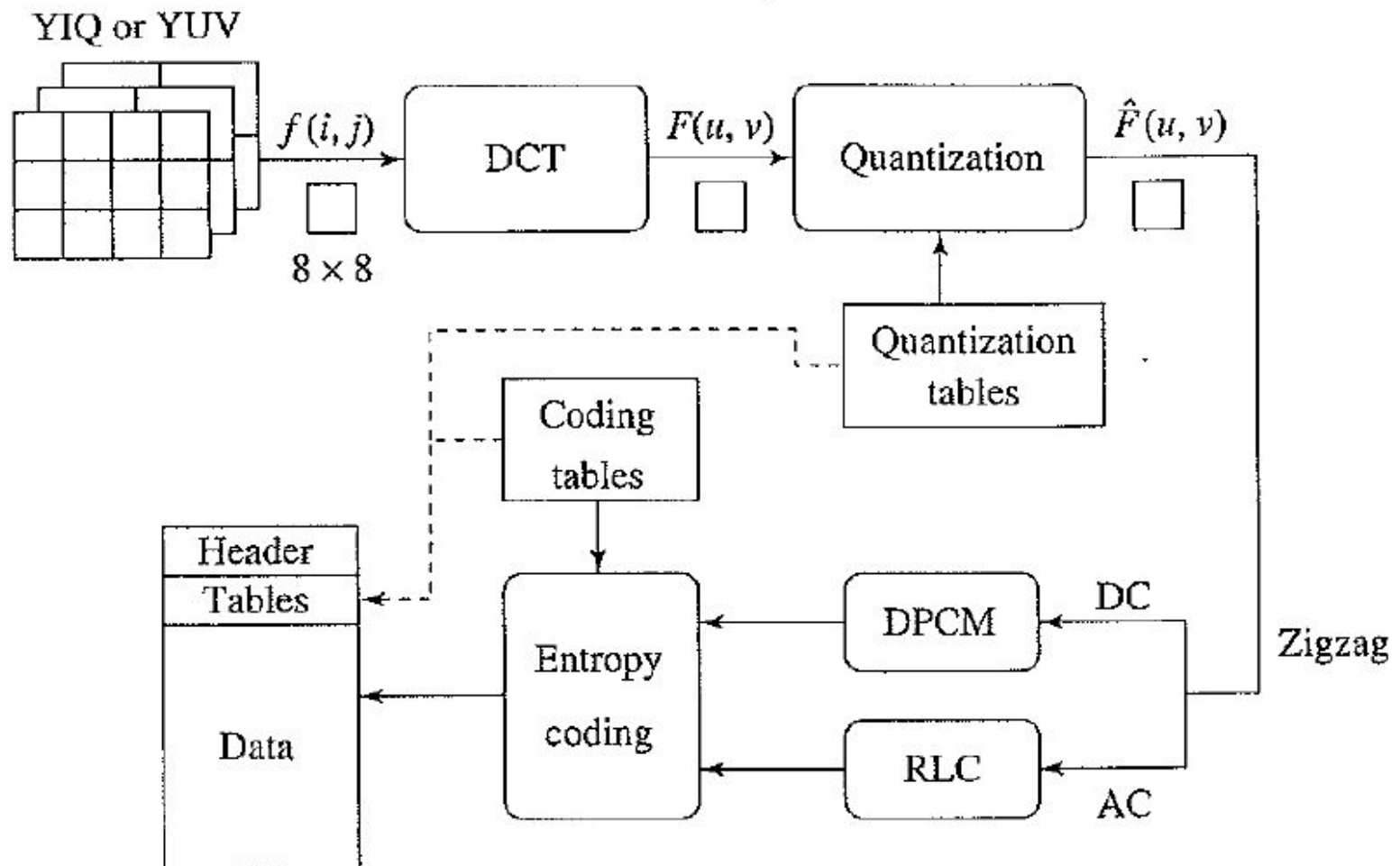


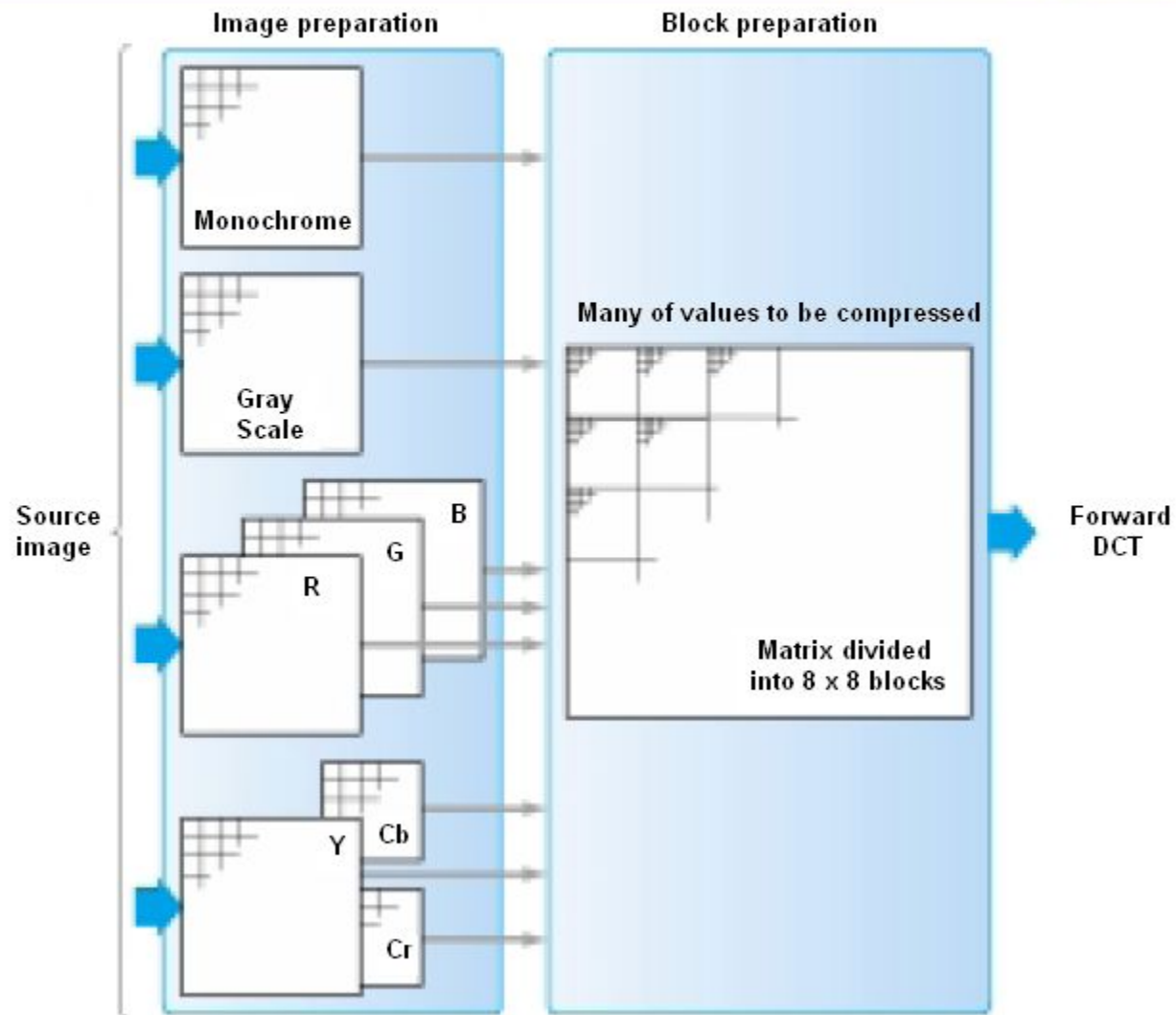
FIGURE 9.1: Block diagram for JPEG encoder.

JPEG

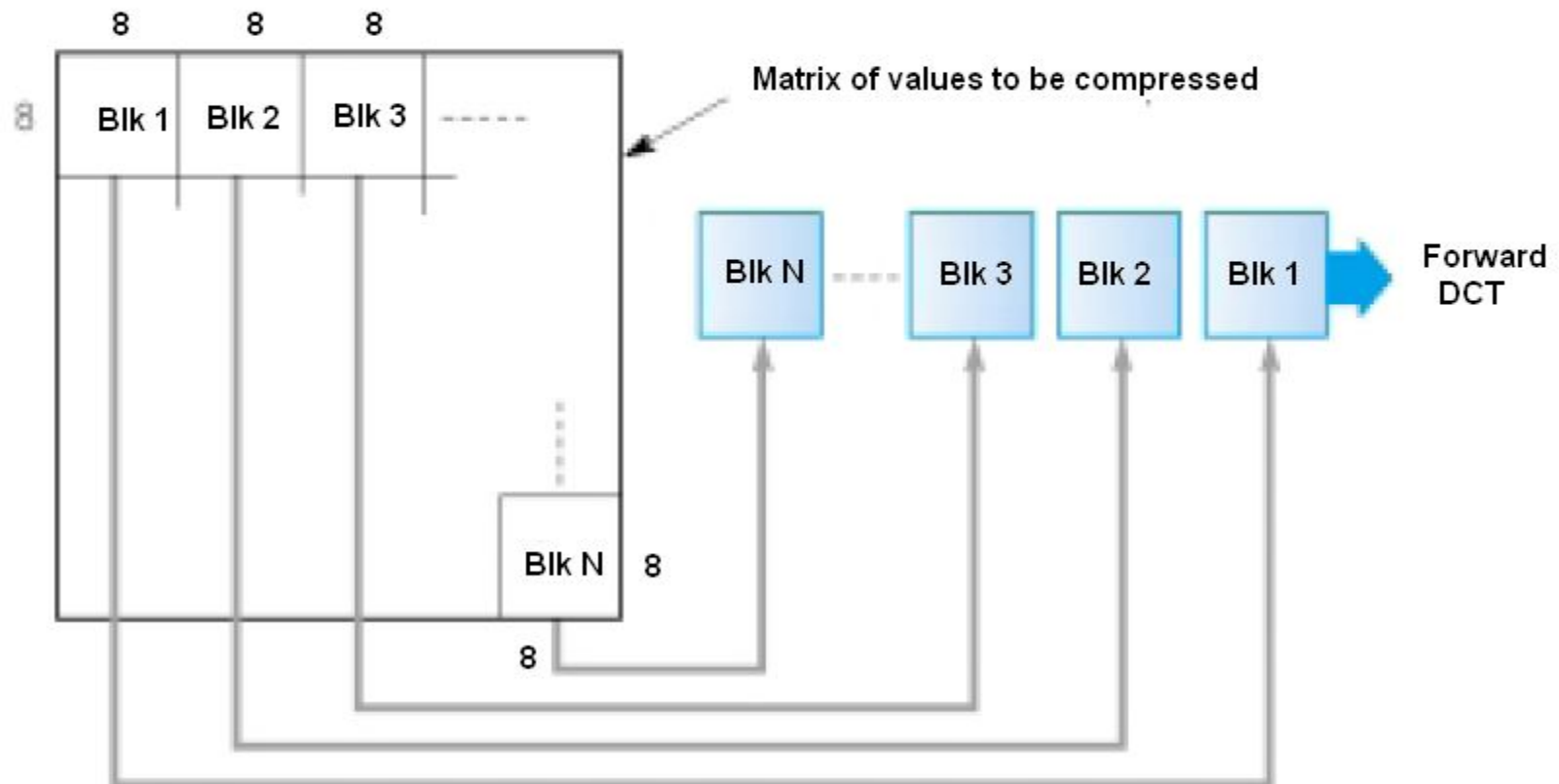
⇒ Preparação da imagem/bloco

- *Imagens monocromáticas, tons de cinza ou usando tabela de cores*
 - 1 matriz
- *Imagem em RGB ou YCrCb*
 - 3 matrizes, uma para cada componente
 - Matrizes de Cr e Cb podem ser menores (Ex.: 4:2:0)
- *Cada elemento de uma matriz pode ter 8 ou 12 bits*
- *Divisão das matrizes em blocos de 8 x 8 submatrizes*
 - Cálculo da transformada para cada posição usa todos os valores da matriz
 - Mais rápido em matrizes menores (8 x 8)
- *Ex.: imagem 640 x 480 => 80x60 (4800) blocos de 8x8*

JPEG



JPEG



JPEG

- A aplicação da Transformada Cosseno Discreta (DCT) na compressão JPEG é baseada em três observações
 - **Observação 01**
 - *Informações importantes de cor em imagem geralmente variam de forma relativamente lenta*
 - *A frequência espacial calculada pela DCT indica quantas vezes os valores dos pixel variam em blocos de imagem de 8x8. Nos dando uma informação de variação ou frequência de variação. Quanto mais variações ocorrem maior a frequência.*
 - **Observação 02**
 - *Humanos são muito menos sensíveis a perda de informações de componentes com alta frequência espacial do que componentes com baixa frequência espacial. Ou seja, percebemos mais informações de baixa frequência espacial.*
 - **Observação 03**
 - *A capacidade de distinguir variações nas informação de cor é maior em escala de cinza do que para informação colorida. Ou seja, variações de informação colorida em pequenos blocos espaciais são mais imperceptíveis.*

JPEG

- Transformada Cosseno Discreta 2D (2D DCT)

$$F(u, v) = \frac{2 C(u) C(v)}{\sqrt{M+N}} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} \cos \frac{(2i+1)u\pi}{2M} \cos \frac{(2j+1)v\pi}{2N} f(i, j)$$

Onde $u, i = 0, 1, \dots, M-1$ e $v, j = 0, 1, \dots, N-1$ e as constantes $C(u)$ e $C(v)$ são determinadas por:

$$C(\xi) = \begin{cases} \frac{\sqrt{2}}{2} & \text{if } \xi = 0, \\ 1 & \text{otherwise.} \end{cases}$$

JPEG

- **Transformada Cosseno Discreta 2D (2D DCT)**
 - No caso da compressão JPEG utilizamos blocos de 8 x 8 para aplicar a transformada. Desta forma nosso $M = N = 8$. Assim a transformada pode ser escrita da seguinte forma

$$F(u, v) = \frac{C(u) C(v)}{4} \sum_{i=0}^7 \sum_{j=0}^7 \cos \frac{(2i+1)u\pi}{16} \cos \frac{(2j+1)v\pi}{16} f(i, j)$$

JPEG

- Transformada Cosseno Discreta 2D Inversa (2D IDCT)

$$\tilde{f}(i, j) = \sum_{u=0}^7 \sum_{v=0}^7 \frac{C(u) C(v)}{4} \cos \frac{(2i+1)u\pi}{16} \cos \frac{(2j+1)v\pi}{16} F(u, v)$$

JPEG

- **Aplicação da transformada Cosseno Discreta na imagem**
 - *Cada imagem é decomposta em blocos de tamanho 8x8*
 - *A transformada é aplicada para cada bloco $f(i,j)$.*
 - *Como saída temos os coeficientes $F(u,v)$ da transformada para aquele bloco.*
 - *Blocos maiores poderiam contribuir de forma mais significativa para tratar componentes de mais baixa frequência (componentes mais importantes)*
 - *Os blocos possuem tamanho 8 x 8 para facilitar o cálculo da transformada, reduzindo o custo computacional*
 - *Por este motivo imagens JPEG ficam um pouco segmentada em blocos*

JPEG

⇒ Cálculo da Transformada Discreta de Co-senos

- *DCT*
- *Discrete Cosine Transform*
- *Todos os 64 valores na matriz de entrada $f(x,y)$ contribuem para cada entrada na matriz transformada $(F(u,v))$*
- *$F[0,0]$ = coeficiente DC – média de todos os 64 valores*
- *Outros valores são chamados coeficientes AC*
 - Para $j=0$ (linha) \Rightarrow coeficientes de frequência horizontal
 - Para $i=0$ (coluna) \Rightarrow coeficientes de frequência vertical

JPEG

⇒ Quantização

- *Olho humano é mais sensível ao coeficiente DC e coeficientes de frequências espaciais mais baixas*
- *Frequências muito altas não são percebidas*
 - Coeficientes descartados (setados como zero) na matriz transformada
- *Threshold para cada coeficiente (tabela de quantização)*
 - Divisão do coeficiente pelo threshold
 - Quociente da divisão é arredondado para valor inteiro (erro de quantização => perda de informação)
- *Tabela de quantização*
 - Padrão tem 2 tabelas defaults (luminância, cromaticidade)
 - Pode usar tabela específica

JPEG

- A quantização é dada pelo quociente da matriz de coeficientes da DCT para cada bloco dividido pela matriz de quantização

$$\hat{F}(u, v) = \text{round} \left(\frac{F(u, v)}{Q(u, v)} \right)$$

- A tabela de quantização possuem valores padrao para informação de luminância e cromaância (tabela abaixo)
 - *Estes valores foram determinados a partir de estudos psicofísicos que tiveram por objetivo maximizar a taxa de compressão e ao mesmo tempo minimizar a perda de percepção da imagem*

TABLE 9.1: The luminance quantization table.

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

TABLE 9.2: The chrominance quantization table.

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

JPEG

Assuming a quantization threshold value of 16, derive the resulting quantization error for each of the following DCT coefficients:

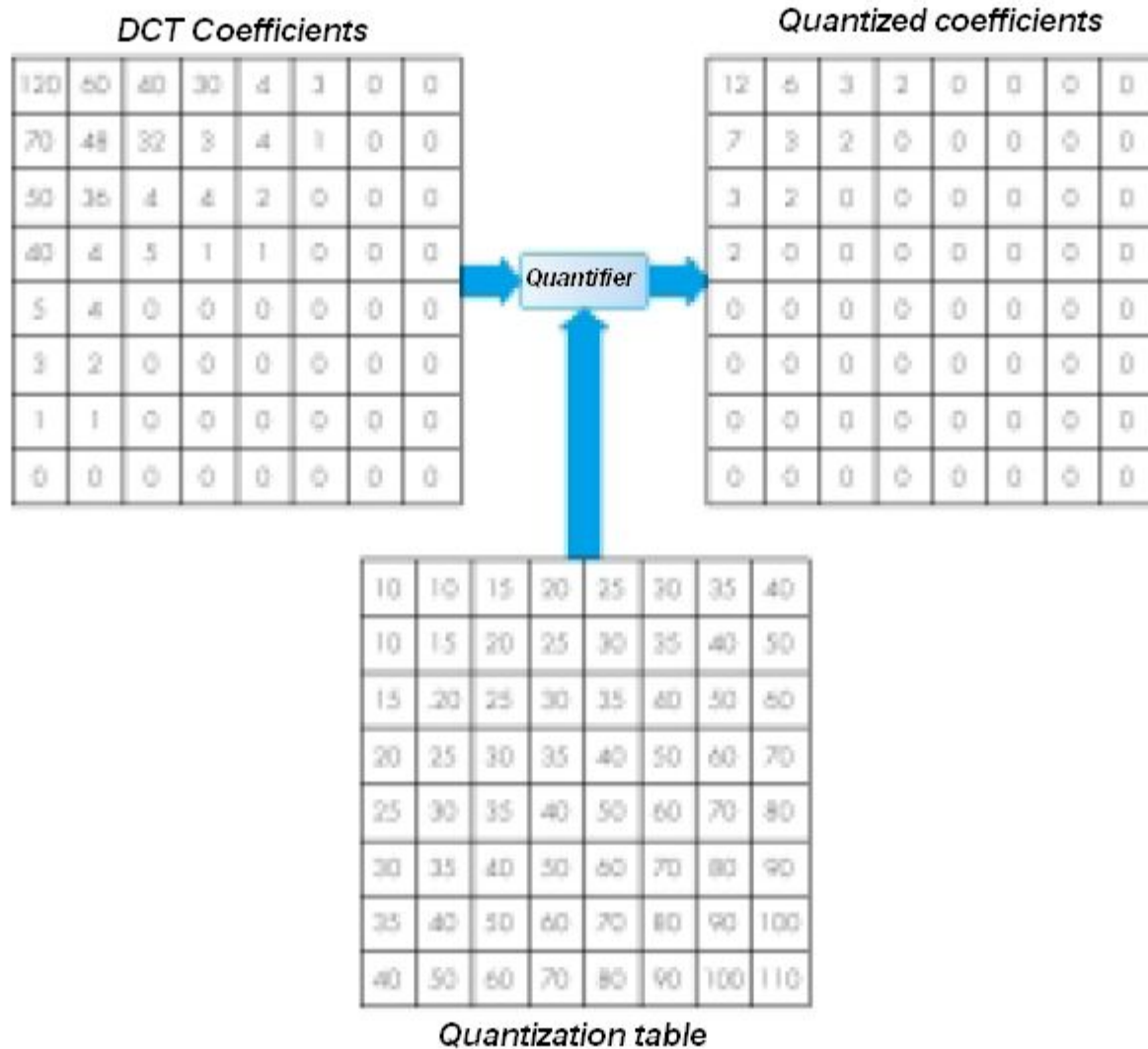
127, 72, 64, 56, -56, -64, -72, -128

Answer:

Coefficient	Quantized value	Rounded value	Dequantized value	Error
127	$127/16 = 7.9375$	8	$8 \times 16 = 128$	-1
72	4.5	5	80	+8
64	4	4	64	0
56	3.5	4	64	+8
-56	-3.5	-4	-64	-8
-64	-4	-4	-64	0
-72	-4.5	-5	-80	-8
-128	-8	-8	-128	0

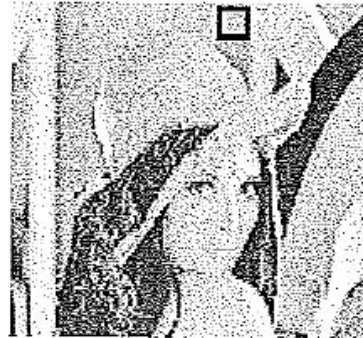
As we can deduce from these figures, the maximum quantization error is plus or minus 50% of the threshold value used.

JPEG



JPEG

- Exemplo de cálculo dos coeficientes DCT e quantização em um bloco 8x8 da imagem Lena



An 8×8 block from the Y image of 'Lena'

200	202	189	188	189	175	175	175
200	203	198	188	189	182	178	175
203	200	200	195	200	187	185	175
200	200	200	200	197	187	187	187
200	205	200	200	195	188	187	175
200	200	200	200	200	190	187	175
205	200	199	200	191	187	187	175
210	200	200	200	188	185	187	186

$f(i, j)$

515	65	-12	4	1	2	-8	5
-16	3	2	0	0	-11	-2	3
-12	6	11	-1	3	0	1	-2
-8	3	-4	2	-2	-3	-5	-2
0	-2	7	-5	4	0	-1	-4
0	-3	-1	0	4	1	-1	0
3	-2	-3	3	3	-1	-1	-3
-2	5	-2	4	-2	2	-3	0

$F(u, v)$

JPEG

- Dequantização, cálculo da transformada inversa e estimação do erro devido a quantização

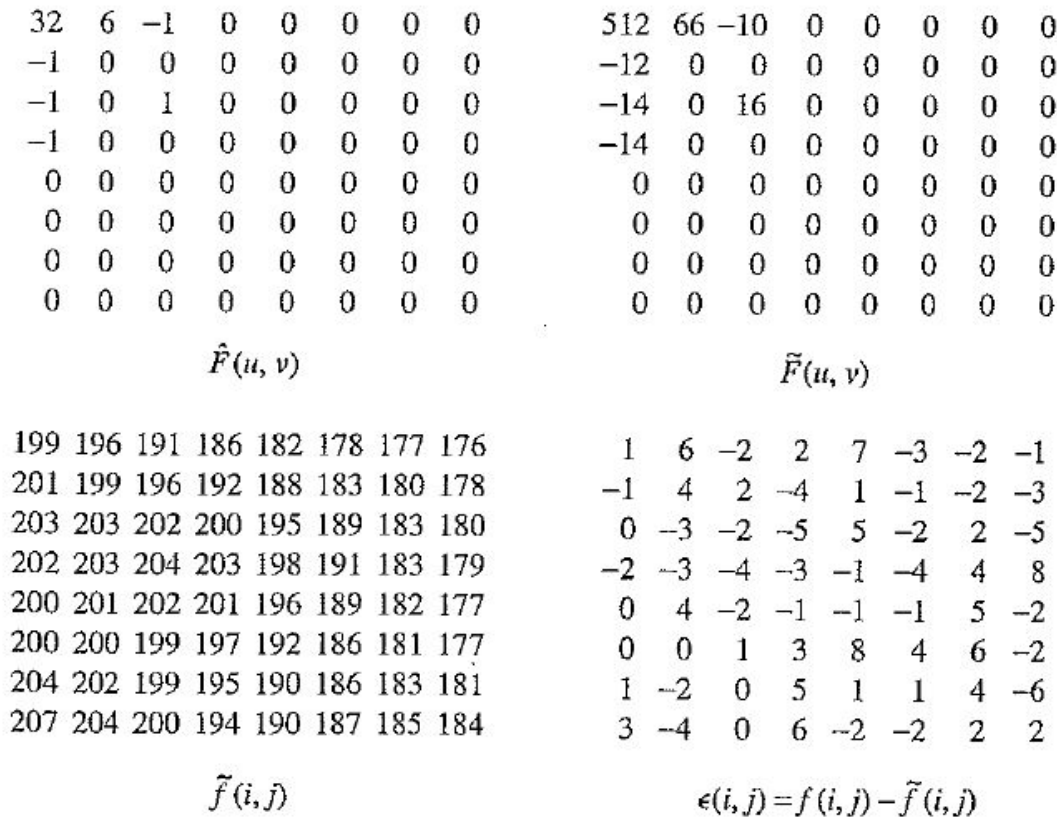


FIGURE 9.2: JPEG compression for a smooth image block.

JPEG

⇒ Conclusões sobre a quantização:

- *O cálculo dos coeficientes quantizados envolve arredondamento para o valor inteiro mais próximo*
- *Os valores de threshold usados aumentam à medida que as frequências espaciais aumentam*
- *O coeficiente DC é o maior na matriz transformada*
- *Muitos dos coeficientes de frequências mais altas são nulos*

⇒ Últimos 2 pontos são explorados na codificação por entropia (próximo estágio)

JPEG

⇒ Codificação por entropia (4 passos)

- *Vetorização (vectoring)*
- *Codificação diferencial (preditiva)*
- *Codificação por carreira (run-length)*
- *Codificação de Huffman*

⇒ Estágio também conhecido como

- *Variable-length coding (VLC) stage*

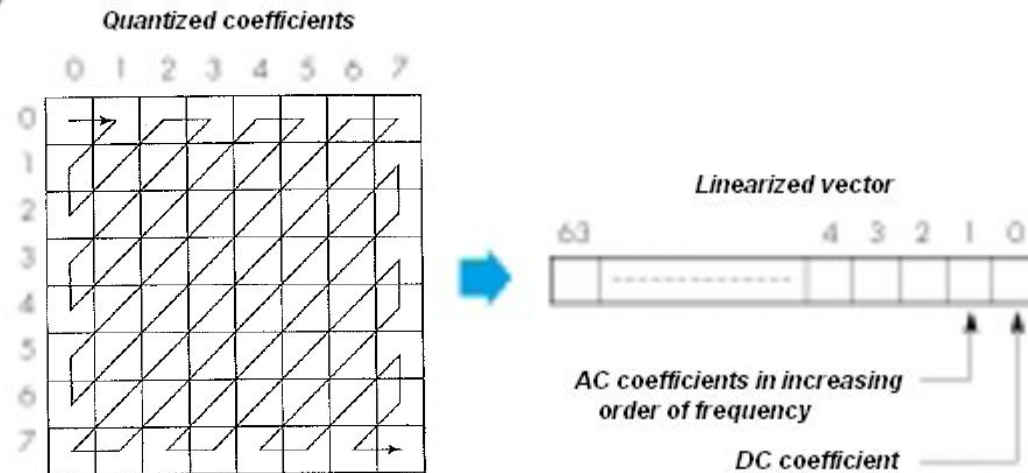
⇒ Utiliza codewords de tamanho variável

JPEG

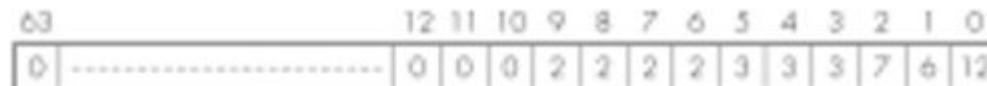
⇒ Vetorização

- *Transformação da matriz de 2 dimensões em um vetor*
- *Coeficientes mais baixos são maiores*
- *Varredura da matriz em zig-zag*

(a)



(b)



JPEG

Fundamentos de Sistemas Multimídia

- ⇒ **Codificação diferencial**
 - *Coeficiente DC*
- ⇒ **Codificação por carreira**
 - *Coeficientes AC*

JPEG

⇒ Codificação diferencial

- *Coeficiente DC*
- *Como blocos são pequenos, não tem muita variação entre coeficientes DC de blocos sucessivos*
 - Valor é codificado como diferença do valor do bloco corrente para o anterior
 - O primeiro em relação a 0 (zero)
- *Ex.: 12, 13, 11, 11, 10, ...*
- *Diferenças: 12, 1, -2, 0, -1, ...*
- *Codificação usa o formato (SSS, valor), onde SSS indica o número de bits necessários para codificar o valor*

JPEG

⇒ Codificação Diferencial

- *Exemplo: 12, 1, -2, 0, 1*
- *Valores negativos são codificados com o complemento do binário (sem sinal)*

⇒ Valor	SSS	Valor Codificado
• 12	4	1100
• 1	1	1
• -2	2	01
• 0	0	
• -1	1	0

JPEG

⇒ Regras usadas na codificação diferencial

Difference value	Number of bits needed (SSS)	Encoded value
0	0	
-1, 1	1	1 = 1, -1 = 0
-3, -2, 2, 3	2	2 = 10, -2 = 01 3 = 11, -3 = 00
-7..-4, 4.. 7	3	4 = 100, -4 = 011 5 = 101, -5 = 010 6 = 110, -6 = 001 7 = 111, -7 = 000
-15...-8, 8...15	4	8 = 1000, -8 = 0111 ⋮

JPEG

⇒ **Codificação por carreira**

- *63 Coeficientes AC*
- *O vetor contém longas strings de 0 (zero)*
- *Coeficientes AC são codificados usando uma string de pares de valores*
- *Codificação usa o formato (skip, value)*
 - Skip = número de zeros
 - Value = próximo coeficiente não-nulo

[illegible]

- $(0,6)(0,7)(0,3)(0,3)(0,3)(0,2)(0,2)(0,2)(0,2)(0,0)$
- $(0,0)$ indica o final da string (todos os coeficientes restantes são zero - 0)

JPEG

⇒ Codificação do valor (*value*)

- *usa o formato (SSS, valor), onde SSS indica o número de bits necessários para codificar o valor*

⇒ Exemplo: (0,6)(0,7)(3,3)(0,-1),(0,0)

⇒ Coef. AC	Skip	SSS	Valor
• 0,6	0	3	110
• 0,7	0	3	111
• 3,3	3	2	11
• 0,-1	0	1	0
• 0,0	0	0	

JPEG

⇒ Codificação de Huffman

- *Usada para codificar a saída das codificações diferencial e por carreira*

⇒ Para coeficientes DC, o campo SSS é codificado com Huffman

- *Decodificador determina sem dúvidas o primeiro SSS do restante dos bits codificados*

⇒ Tabela de Huffman default ou específica

Number of bits needed (SSS)	Huffman codeword
0	010
1	011
2	100
3	00
4	101
5	110
6	1110
7	11110
⋮	
11	111111110

JPEG

⇒ Exemplo: 12, 1, -2, 0, 1

⇒ Valor	SSS	SSS(Huff.)	Valor	Bitstream
• 12	4	101	1100	1011100
• 1	1	011	1	0111
• -2	2	100	01	10001
• 0	0	010		010
• -1	1	011	0	0110

JPEG

- ⇒ Para coeficientes AC, os campos Skip e SSS são tratados em conjunto (como um símbolo composto)
 - *Skip varia de 0 a 15*
 - *SSS varia de 0 a 10*
- ⇒ Tabela de Huffman default ou específica (deve ser enviada junto com o bitstream)

JPEG

⇒ Exemplo: (0,6)(0,7)(3,3)(0,-1),(0,0)

Coef. AC	Skip/SSS	cód. Huff.	Valor
• 0,6	0/3	100	110
• 0,7	0/3	100	111
• 3,3	3/2	111110111	11
• 0,-1	0/1	00	0
• 0,0	0/0	1010 (end of block)	

⇒ Bitstream:

- 100110 100111 1111101111 000 1010

⇒ No exemplo:

- 30 bits para codificar coeficientes AC, assumindo 7 bits para coeficiente DC = total de 37 bits
- Suponha que cada pixel da matriz 8x8 original tivesse 8 bits
- Taxa de compressão resultante:
 - $512/37 \Rightarrow 13,8:1$

JPEG

⇒ Construção dos quadros

- *Definição da estrutura do bitstream completo de uma imagem*
 - Quadro (*frame*) JPEG
- *Estrutura hierárquica*

JPEG

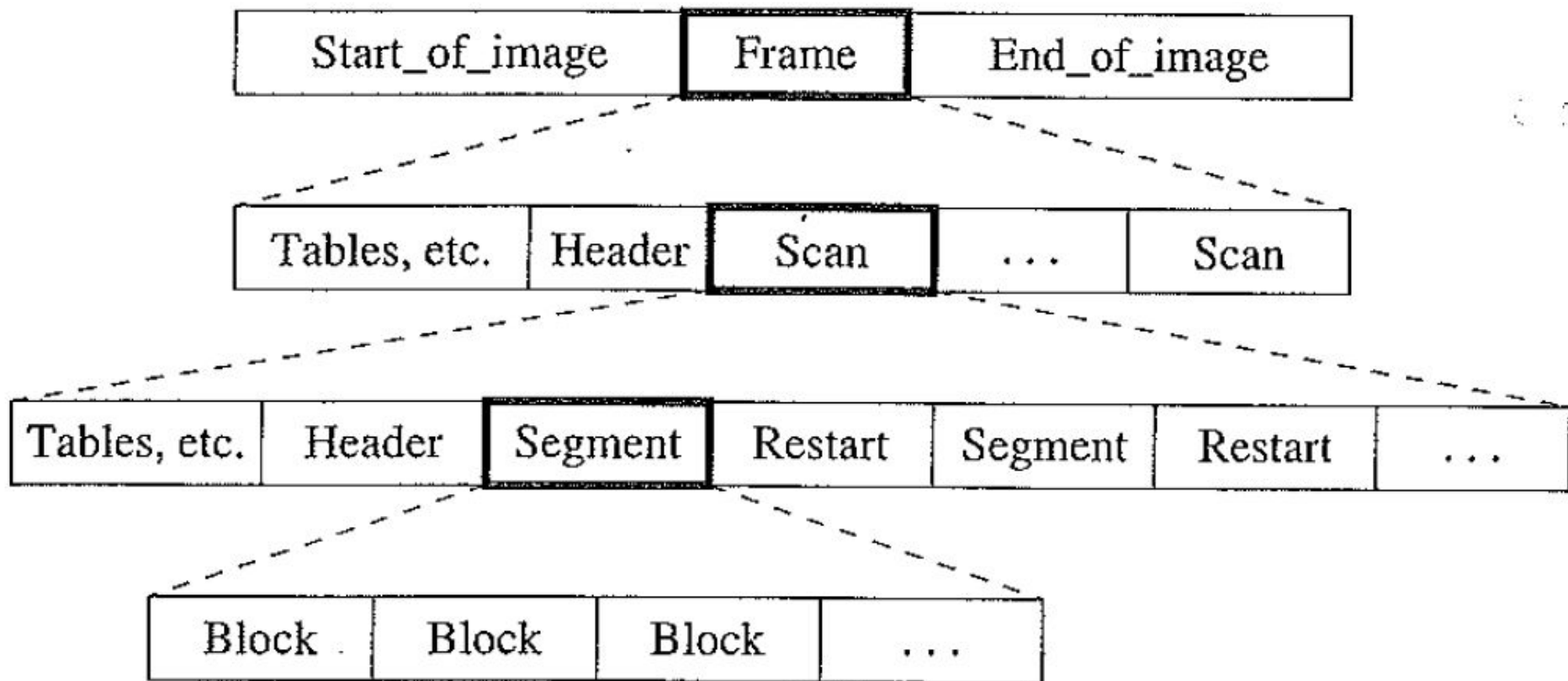


FIGURE 9.6: JPEG bitstream.

JPEG

⇒ Primeiro nível (level 1)

- *Delimitador de início*
- *Cabeçalho*
 - Tamanho total da imagem em pixels
 - Número e tipos dos componentes usados na representação
 - *Tons de cinza, RGB, YCrCb*
 - Formato de digitalização
 - *4:2:2, 4:2:0, ...*
- *Conteúdo*
- *Delimitador de fim*

JPEG

⇒ Segundo nível (level 2) – conteúdo do quadro

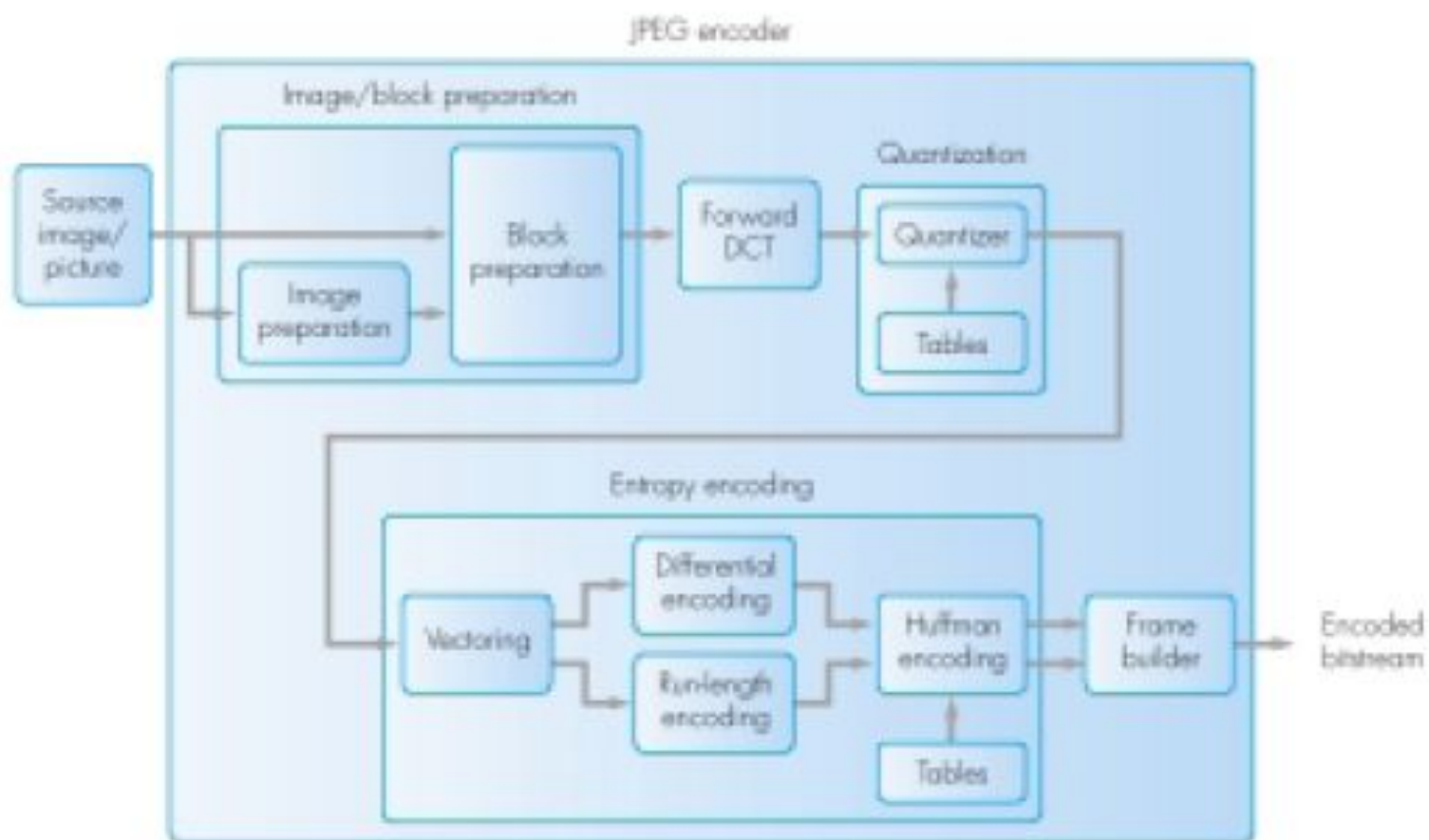
- *componentes chamados scan*
- *cabeçalho (scan header) que contém*
 - Identidade dos componentes (R, G, B, ...)
 - Número de bits usados para digitalizar cada componente
 - Tabelas de quantização utilizadas para codificar cada componente

JPEG

- ⇒ Terceiro nível (level 3)
- ⇒ Cada componente consiste de:
 - *1 ou mais segmentos e cabeçalhos*
- ⇒ Segmento
 - *contém um grupo de blocos 8x8*
 - DC, skip/value, skip/value, ..., end-of-block
- ⇒ Cabeçalho (*segment header*)
 - *Tabelas de Huffman utilizadas na codificação de cada segmento*
- ⇒ Cada segmento pode ser decodificado de forma independente dos outros
 - *impede a propagação de erros em outros segmentos*

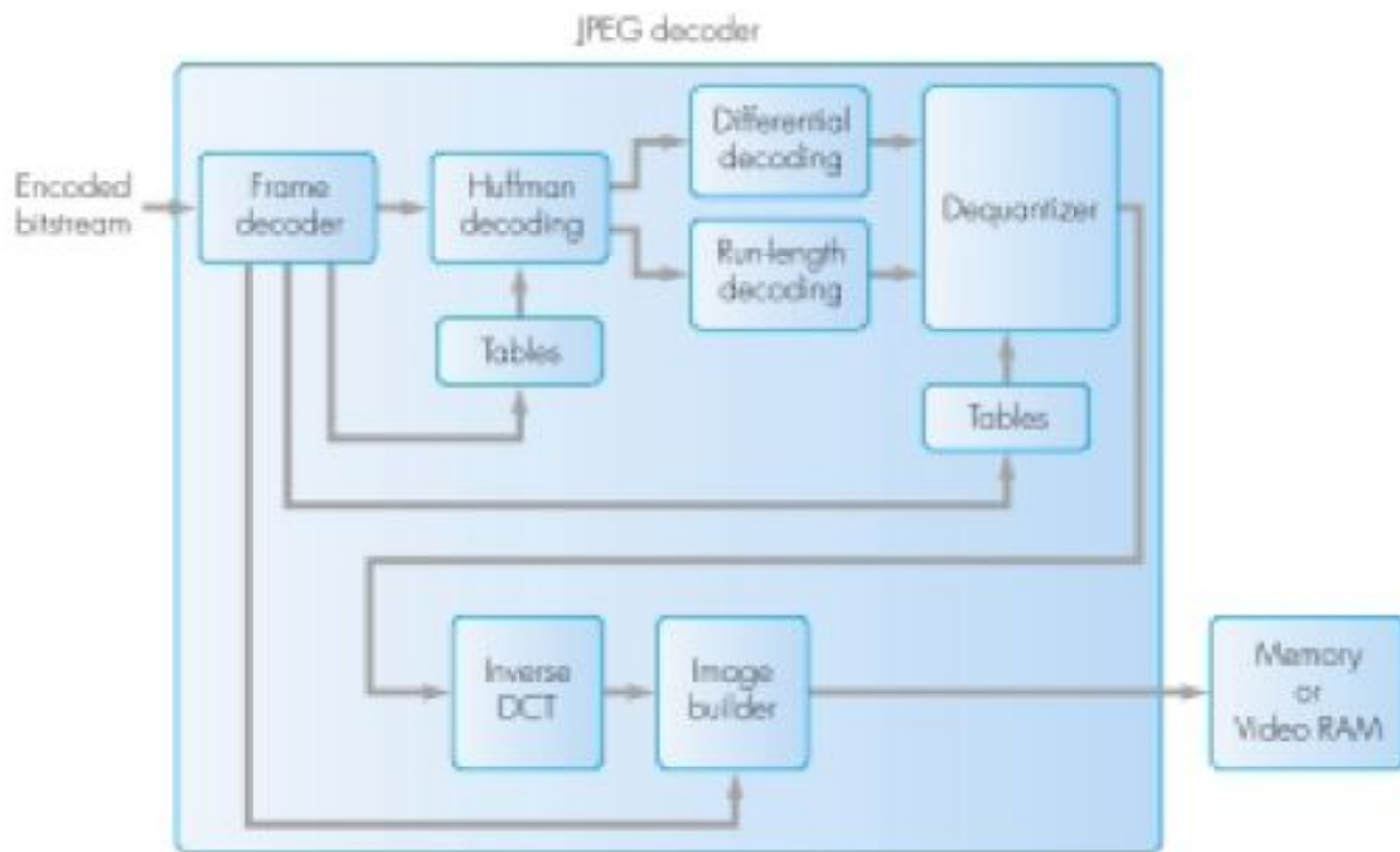
Codificador JPEG

Fundamentos de Sistemas Multimídia



Decodificador JPEG

Fundamentos de Sistemas Multimedia



Decodificador JPEG

Fundamentos de Sistemas Multimídia

- ⇒ 1. Decodificador de quadros
 - *Identifica tabelas utilizadas*
- ⇒ 2. Decodificação de Huffman
 - *Usando tabela identificada*
- ⇒ 3. Decodificação Diferencial (coef. DC) e por Carreira (coef. AC)
- ⇒ 4. “Desquantização”
 - *Usando tabela identificada*
- ⇒ 5. Transformada de Co-senos inversa
- ⇒ 6. Construção da imagem final

JPEG

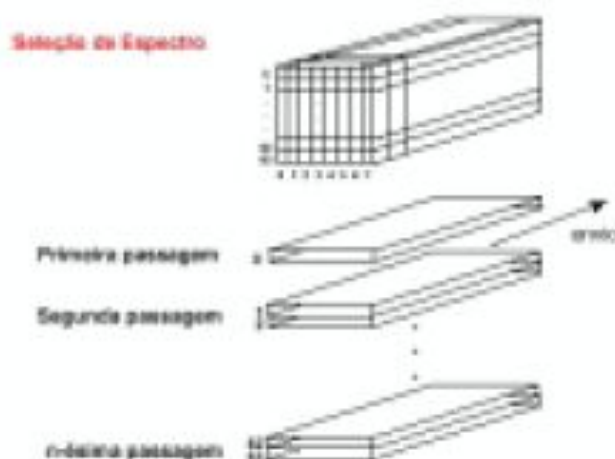
⇒ **Descompressão pode ser feita de forma progressiva**

- *Modo progressivo*
- *Modo hierárquico*

JPEG

⇒ Modo progressivo

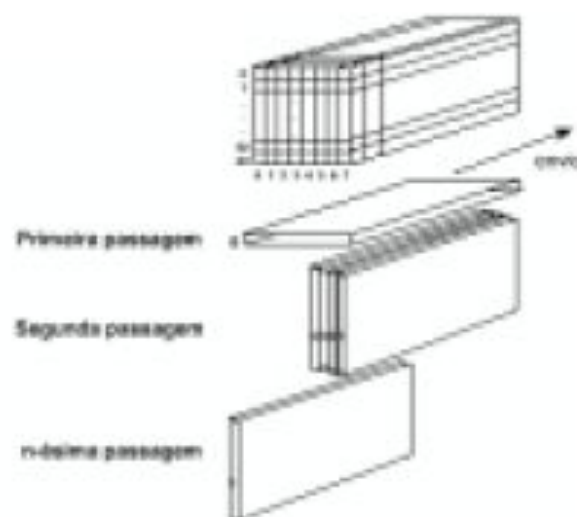
- *Seleção de espectro:*
 - Primeiro o coeficiente DC e os coeficientes de baixa frequência são enviados para cada bloco; depois os coeficientes de alta frequência



JPEG

⇒ Modo progressivo

- *Aproximação sucessiva:*
 - Primeiro o coeficiente DC é enviado para cada bloco;
 - depois os bits dos outros coeficientes do mais significativo para o menos significativo
 - Técnica mais interessante se imagem tem muita variação (coeficientes de alta frequência altos)



JPEG

⇒ Vantagens do modo progressivo:

- *envia os coeficientes mais importantes juntos*
- *Quando recebe os coeficientes mais importantes, já começa a mostrar a imagem*
 - Efeito: imagem aparece borrada e vai ficando mais nítida conforme vai chegando o resto

JPEG

⇒ Modo hierárquico:

- *Filtragem de amostras com redução de resolução pelo fator 2 em múltiplas dimensões*
- *Codificação da imagem reduzida*
- *Gradativamente envia restante das amostras*

⇒ Efeito no receptor:

- *Melhora gradativa da resolução geométrica*

⇒ Interessante quando dispositivo receptor não tem resolução suficiente para apresentar a imagem

- *Basta mandar a imagem reduzida*



Dúvidas?

Contato: daniel@dcx.ufpb.br

***Este material é uma cópia adaptada dos slides da Professora Débora Christina da Universidade Federal Fluminense**

****A versão original pode ser acessada através do site
<http://www.midiacom.uff.br/~debora/fsmm/>**



Universidade Federal da Paraíba - UFPB

Sistemas Multimídia

Imagem
Parte 02

Professor: Daniel Faustino