

Tópicos

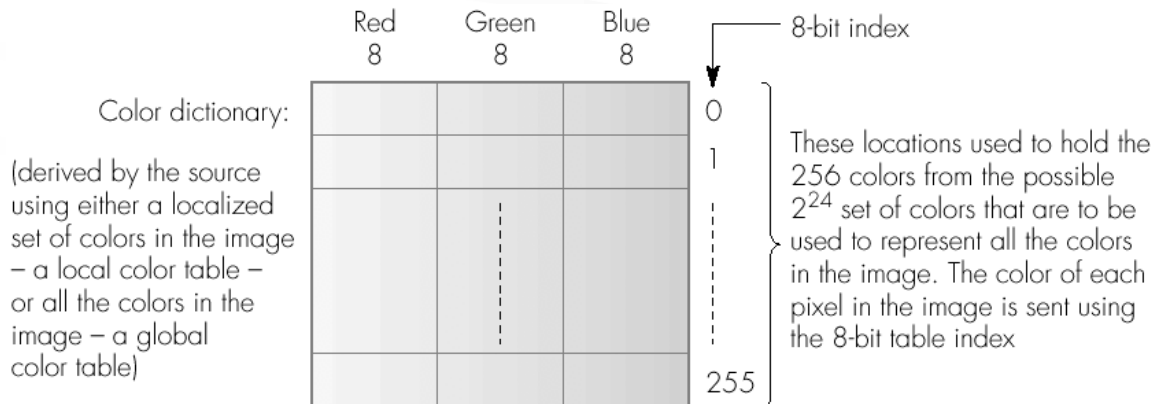
- √ Introdução
- √ Representação de Informação Multimídia
- √ Digitalização
- √ Mídias Texto, Imagem, Áudio e Vídeo
- • Compressão, Padrões de Compressão (Texto, Imagem, Áudio e Vídeo)
- Comunicação Multimídia
- Protocolos de Rede, Redes Locais
- Redes Sem Fio, Bluetooth
- Sincronismo de Mídias
- Qualidade de Serviço
- Tópicos Avançados: Criptografia, Watermarking;
- Realidade Virtual
- Serviços Multimídia: Vídeo Sob Demanda, Videoconferência
- Sistemas Multimídia Avançados: Ambientes Virtuais Colaborativos

Compressão de Imagens

- Existem, basicamente, dois tipos de imagens:
 - Geradas por Computador (Gráficos).
 - Armazenadas (e transmitidas) como um conjunto de instruções (formato de programa) que geram a imagem, ao invés de um formato de matriz de pixels
 - Quando uma imagem é transmitida no formato de programa, algum esquema de compressão sem perdas tem de ser utilizado.
 - Imagens Digitalizadas (Fotos escaneadas, etc.).
 - Armazenadas em formato matricial (pixels).
 - Dois métodos de compressão (padronizados) básicos são utilizados:
 - Combinação de codificação estatística e por repetição de série (run-length) - Compressão sem perdas de documentos digitalizados.
 - Combinação de codificações por transformadas, diferenças e por repetição de série (run-length) - Caso genérico.

GIF (*Graphics Interchange Format*)

- Permite codificação de imagens com 24 bits por pixel (8 para cada componente RGB), embora o formato seleccione as 256 (do conjunto de 2^{24}) cores que melhor representam aquelas presentes na imagem.
- A lista de 256 cores resulta numa tabela de cores, cada entrada contendo um valor de 24 bits de cor.

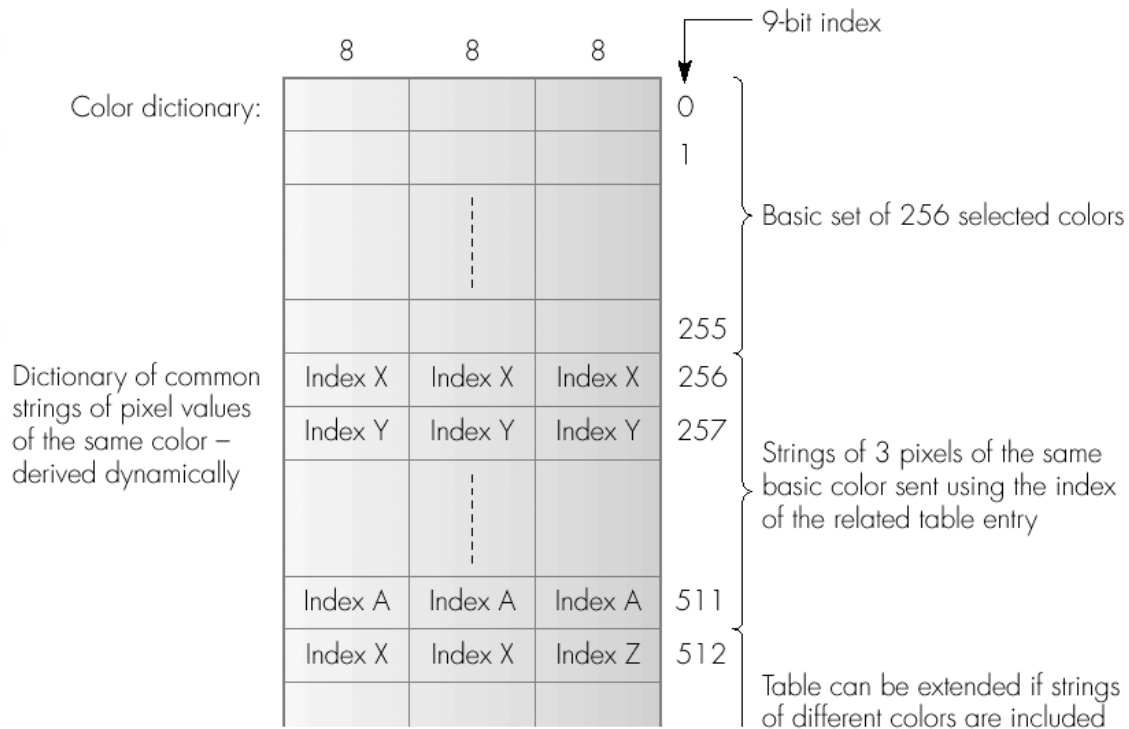


The color dictionary, screen size, and aspect ratio are sent with the set of indexes for the image.

GIF (*Graphics Interchange Format*)

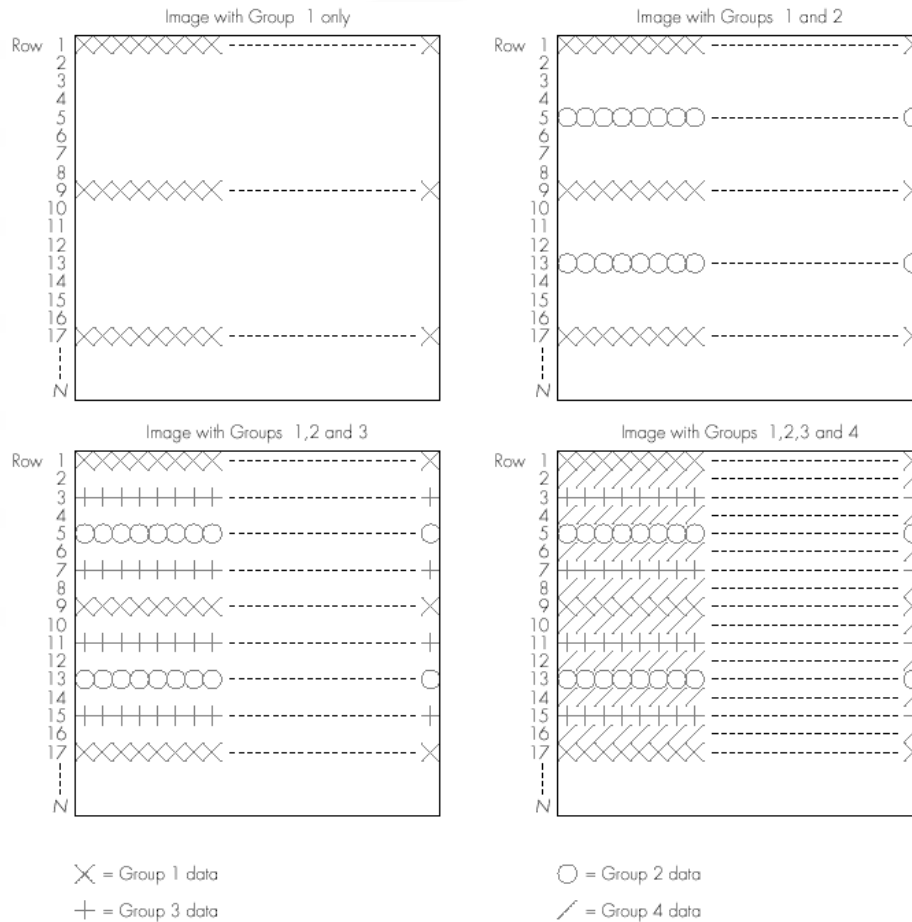
- Cada pixel é então codificado com um elemento da tabela (8 bits) ao invés de 24, com uma compressão 3:1.
- A Tabela de Cores pode ser
 - Global: é utilizada na imagem inteira.
 - Local: é utilizada apenas em parte da imagem.
- O conteúdo da tabela é enviado pela rede (ou armazenado), juntamente com informação sobre as dimensões da imagem e os dados referentes aos pixels da imagem propriamente ditos.
- A codificação LZW pode ser utilizada para obter maior compressão.
- A tabela de valores vai sendo dinamicamente extendida à medida que strings de cores vão aparecendo.
- Usualmente a tabela começa com 512 posições, 256 das quais contém a tabela original. Quando a tabela está cheia, a mesma pode crescer em incrementos de 1 bit (2^9 cores, 2^{10} cores, 2^{11} cores, etc.).

GIF (*Graphics Interchange Format*)



GIF (*Graphics Interchange Format*)

- Pode-se armazenar e transmitir a imagem em modo entrelaçado (*Interlaced*).
 - A imagem a ser transmitida é organizada de modo que a imagem possa ser reconstruída, com qualidade menor, mais rapidamente. Em seguida mais níveis de detalhe são acrescentados à imagem até que a imagem completa é apresentada.
 - Muito útil para canais de comunicação lentos, pois rapidamente já se recebe um esboço da imagem (um usuário pode cancelar a transferência se a imagem não for de interesse, por exemplo).
 - A imagem é dividida em quatro grupos, o primeiro com 1/8 da informação da imagem, o segundo com outros 1/8, o terceiro com 1/4 e o último com o 1/2 restante.



TIFF (*Tagged Image File Format*)

- Suporta resolução de pixels de até 48 bits (16 bits para cada componente RGB).
- Desenvolvido tanto para envio de imagens como de documentos digitalizados.
- A imagem pode ser armazenada em 5 modos distintos, sendo o modo 1 completamente sem compressão e o modo 5 um modo comprimido através de codificação LZW. Os modos 2, 3 e 4 são utilizados para codificação de documentos digitalizados, com algoritmos similares àqueles utilizados por máquinas de facsimile.
- A codificação LZW utilizada é a mesma do padrão GIF, iniciando com um dicionário de 256 cores, que pode crescer até 4096 entradas.

Documentos Digitalizados

- Como anteriormente discutido, uma página de facsimile resulta num montante de dados de 2Mbits, utilizando codificação de 1 bit por pixel. Tal massa de dados precisa ser transmitida sob linhas telefônicas padrão, através de modems and PSTN (*Public Switched Telephone Network*), o que acabaria levando a um custo muito elevado para o envio de um documento de várias páginas.
- A ITU-T criou alguns padrões para codificação de dados de facsimile:
 - T2 e T3 (Grupos 1 e 2) - padrões antigos, quase não usados hoje em dia.
 - T4 (Grupo 3) - padrão de transmissão digital usando modems sobre PSTN. Utiliza compressão de dados que excede 10:1 e permite a transferência de uma página em menos de um minuto.
 - T6 (Grupo 4) - padrão de transmissão completamente digital para uso sobre redes digitais, como RDSI (*ISDN*). Compressão excede 10:1. Transmite uma página em poucos segundos (RDSI provê 64kbps).

Documentos Digitalizados

- Parte do processo de padronização incluiu extensa análise da composição de documentos típicos, o que resultou em tabelas de seqüências de branco e preto nos documentos.
- Os códigos resultantes são armazenados em duas tabelas:
 - *Termination-codes table* (a): seqüências de repetição (*run-length*) de branco e preto de tamanhos de 0 a 63 bits em incrementos simples.
 - *Make-up codes table* (b): seqüências de repetição de b&p de tamanhos múltiplos de 64.

| (a) | | | | (a) cont. | | | |
|------------------|-----------|------------------|---------------|------------------|-----------|------------------|---------------|
| White run-length | Code word | Black run-length | Code word | White run-length | Code word | Black run-length | Code word |
| 0 | 00110101 | 0 | 0000110111 | 56 | 01011001 | 56 | 000000101000 |
| 1 | 000111 | 1 | 010 | 57 | 01011010 | 57 | 0000001011000 |
| 2 | 0111 | 2 | 11 | 58 | 01011011 | 58 | 0000001011001 |
| 3 | 1000 | 3 | 10 | 59 | 01001010 | 59 | 000000101011 |
| 4 | 1011 | 4 | 011 | 60 | 01001011 | 60 | 000000101100 |
| 5 | 1100 | 5 | 0011 | 61 | 00110010 | 61 | 0000001011010 |
| 6 | 1110 | 6 | 0010 | 62 | 00110011 | 62 | 000000100110 |
| 7 | 1111 | 7 | 00011 | 63 | 00110100 | 63 | 000000100111 |
| 8 | 10011 | 8 | 000101 | | | | |
| 9 | 10100 | 9 | 000100 | | | | |
| 10 | 00111 | 10 | 0000100 | | | | |
| 11 | 01000 | 11 | 0000101 | | | | |
| 12 | 001000 | 12 | 0000111 | | | | |
| 13 | 000011 | 13 | 00000100 | | | | |
| 14 | 110100 | 14 | 00000111 | | | | |
| 15 | 110101 | 15 | 000001000 | | | | |
| 16 | 101010 | 16 | 0000010111 | | | | |
| 17 | 101011 | 17 | 0000011000 | | | | |
| 18 | 0100111 | 18 | 0000001000 | | | | |
| 19 | 0001100 | 19 | 00001100111 | | | | |
| 20 | 0001000 | 20 | 00001101000 | | | | |
| 21 | 0010111 | 21 | 00001101100 | | | | |
| 22 | 0000011 | 22 | 00000110111 | | | | |
| 23 | 0000100 | 23 | 00000101000 | | | | |
| 24 | 0101000 | 24 | 00000010111 | | | | |
| 25 | 0101011 | 25 | 00000011000 | | | | |
| 26 | 0010011 | 26 | 000011001010 | | | | |
| 27 | 0100100 | 27 | 000011001011 | | | | |
| 28 | 0011000 | 28 | 000011001100 | | | | |
| 29 | 00000010 | 29 | 000011001101 | | | | |
| 30 | 00000011 | 30 | 000001101000 | | | | |
| 31 | 00011010 | 31 | 000001101001 | | | | |
| 32 | 0001011 | 32 | 000001101010 | | | | |
| 33 | 0010010 | 33 | 000001101011 | | | | |
| 34 | 00010011 | 34 | 000011010010 | | | | |
| 35 | 00010100 | 35 | 000011010011 | | | | |
| 36 | 00010101 | 36 | 000011010100 | | | | |
| 37 | 00010110 | 37 | 000011010101 | | | | |
| 38 | 00010111 | 38 | 000011010110 | | | | |
| 39 | 00101000 | 39 | 000011010111 | | | | |
| 40 | 00101001 | 40 | 000001101000 | | | | |
| 41 | 00101011 | 41 | 000001101001 | | | | |
| 42 | 00101011 | 42 | 000011011010 | | | | |
| 43 | 00101100 | 43 | 000011011011 | | | | |
| 44 | 00101101 | 44 | 000001010100 | | | | |
| 45 | 00000100 | 45 | 000001010101 | | | | |
| 46 | 00000101 | 46 | 000001010110 | | | | |
| 47 | 000001010 | 47 | 000001010111 | | | | |
| 48 | 00001011 | 48 | 000001100100 | | | | |
| 49 | 01010010 | 49 | 000001100101 | | | | |
| 50 | 01010011 | 50 | 0000001010010 | | | | |
| 51 | 01010100 | 51 | 0000001010011 | | | | |
| 52 | 01010101 | 52 | 0000000100100 | | | | |
| 53 | 00100100 | 53 | 000000110111 | | | | |
| 54 | 00100101 | 54 | 000000111000 | | | | |
| 55 | 01011000 | 55 | 000000100111 | | | | |

| (b) | | | |
|------------------|--------------|------------------|---------------|
| White run-length | Code word | Black run-length | Code word |
| 64 | 11011 | 64 | 00000001111 |
| 128 | 10010 | 128 | 000011001000 |
| 192 | 010111 | 192 | 000011001001 |
| 256 | 0110111 | 256 | 0000001011011 |
| 320 | 00110110 | 320 | 000000110011 |
| 384 | 00101111 | 384 | 000000100100 |
| 448 | 01100100 | 448 | 00000010101 |
| 512 | 01100101 | 512 | 000000101100 |
| 576 | 01101000 | 576 | 000000101101 |
| 640 | 01100111 | 640 | 0000001001010 |
| 704 | 011001100 | 704 | 0000001001011 |
| 768 | 011001101 | 768 | 0000001001100 |
| 832 | 011010010 | 832 | 0000001001101 |
| 896 | 011010011 | 896 | 0000001110010 |
| 960 | 011010100 | 960 | 0000001110011 |
| 1024 | 011010101 | 1024 | 0000001110100 |
| 1088 | 011010110 | 1088 | 0000001110101 |
| 1152 | 011010111 | 1152 | 0000001110110 |
| 1216 | 011011000 | 1216 | 0000001110111 |
| 1280 | 011011001 | 1280 | 0000001010010 |
| 1344 | 011011010 | 1344 | 0000001010011 |
| 1408 | 011011011 | 1408 | 0000001010100 |
| 1472 | 010011000 | 1472 | 0000001010101 |
| 1536 | 010011001 | 1536 | 0000001011010 |
| 1600 | 010011010 | 1600 | 0000001011011 |
| 1664 | 011000100 | 1664 | 0000001001000 |
| 1728 | 010011011 | 1728 | 0000001001001 |
| 1792 | 00000001000 | 1792 | 00000001000 |
| 1856 | 00000001100 | 1856 | 00000001100 |
| 1920 | 00000001101 | 1920 | 00000001101 |
| 1984 | 000000010010 | 1984 | 0000000100100 |
| 2048 | 000000010011 | 2048 | 0000000100101 |
| 2112 | 000000010100 | 2112 | 0000000101000 |
| 2176 | 000000010101 | 2176 | 0000000101001 |
| 2240 | 000000010110 | 2240 | 0000000101100 |
| 2304 | 000000010111 | 2304 | 0000000101101 |
| 2368 | 000000011100 | 2368 | 0000000011100 |
| 2432 | 000000011101 | 2432 | 0000000011101 |
| 2496 | 000000011110 | 2496 | 0000000011110 |
| 2560 | 000000011111 | 2560 | 0000000011111 |
| EOL | 00000000001 | EOL | 00000000001 |

Documentos Digitalizados

- Uma técnica chamada *overscanning* é utilizada, e significa que todas as linhas iniciam com pelo menos um pixel branco, logo basta que se codifique os números alternados para branco e preto.
- O esquema de codificação também é conhecido como codificação de Huffman modificada, já que utiliza dois conjuntos de palavras de codificação (*termination* e *make-up*).
- Como exemplo, um conjunto de 12 pixels brancos são codificados como 001000. Um conjunto de 12 bits pretos pode ser codificado diretamente como 0000111.
- Um conjunto de 140 bits pretos são codificados com o código estendido 000011001000 + 0000111 (128 + 12 pixels).
- Uma sequência de mais de 2560 pixels são codificados com mais de um *make-up code*, seguido de um *termination code*.

Documentos Digitalizados - T4

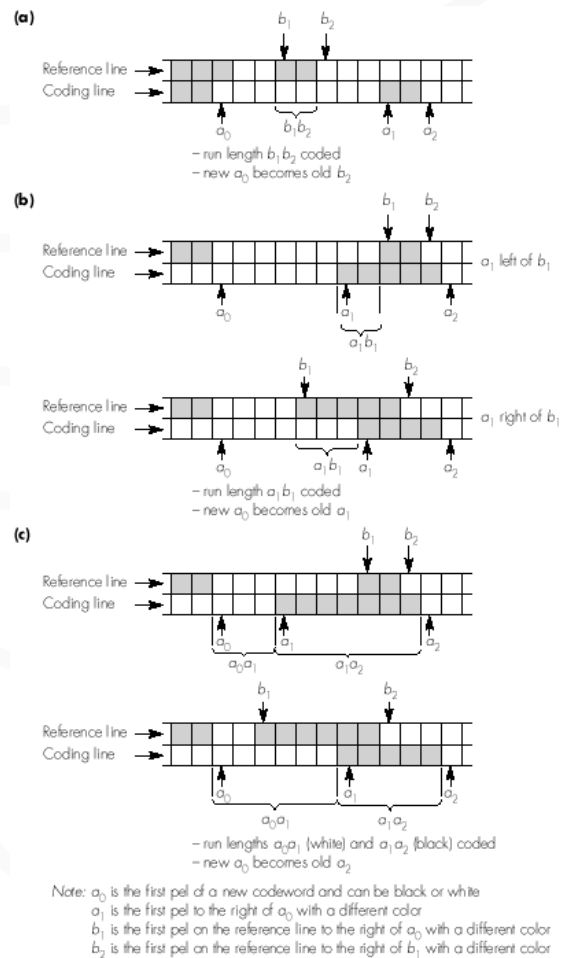
- O modelo Grupo 3 não prevê utilização de protocolo de correção de error. Para evitar inconsistência extensa entre transmissor e receptor, no final de cada linha um código bem definido de final de linha é enviado (EOL)
 - Se o decodificador detetar inconsistência (número de bits maior que uma linha deveria ter por exemplo), ele aguarda o próximo EOL para continuar a recepção.
 - Caso o receptor não consiga detetar o sinal de final de linha (EOL) por um determinado número de linhas a recepção é abortada e uma notificação enviada ao equipamento transmissor.
 - Um EOL é enviado antes de qualquer linha de uma página. Seis EOLs consecutivos são enviados como indicação de final de página.
- Cada linha é codificada independentemente, motivo pelo qual tal codificação também é conhecida como codificação uni-dimensional.
- Comprime bem documentos com grandes quantidades de branco/preto (cartas, desenhos, etc.). Não é bom para documentos com fotos, o que pode levar a compressão negativa.

Documentos Digitalizados - T6

- Criado para suprir uma solução no que se refere às fraquesas da codificação T4 (compressão negativa). É opcional em facsímiles do Grupo 3, mas obrigatórios nos do Grupo 4.
 - O marcador de final de linha (EOL) contém um bit adicional. Um bit um indica que a codificação da próxima linha usa o modelo T4, um bit zero indica que a codificação T6 é utilizada na próxima linha.
 - Conhecido como *modified READ (MMR) coding*, sendo também conhecida como codificação bidimensional, ou 2D, já que a mesma codifica preto e branco comparando linhas adjacentes. READ significa *Relative Element Address Designate*.
 - Utiliza o fato de que pouca mudança costuma existir entre linha adjacentes.
 - O código de repetição de série (*run-length*) é gerado comparando a linha a ser codificada (CL - *Coding Line*), comparada com a linha anterior (RL - *Reference Line*). A primeira linha é codificada em comparação com uma linha imaginária completamente branca. Linhas iniciam com um pixel branco imaginário.

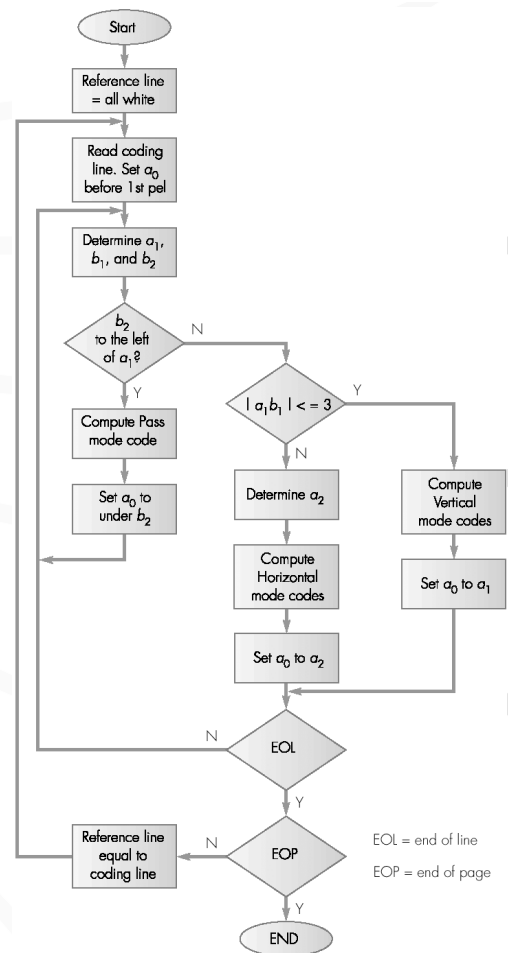
Documentos Digitalizados - T6

- Modos de codificação por repetição de séries utilizados dependem da posição da próxima repetição de série na linha de referência (b_1b_2) relativo ao início e final do próximo par de repetição de série na linha codificada (a_0a_1 e a_1a_2):
 - *Pass Mode*: Usado quando b_1b_2 está à esquerda de a_1a_2 (b_2 à esquerda de a_1).
 - *Vertical Mode*: Usado quando b_1b_2 se sobrepõe a a_1a_2 por no máximo ± 3 pels.
 - *Horizontal Mode*: Usado quando b_1b_2 se sobrepõe a a_1a_2 por mais de ± 3 pels.



Documentos Digitalizados - T6

- O primeiro a_0 é inicializado com um pel branco imaginário localizado antes do primeiro pixel da linha.
- Se em uma linha a_1 , a_2 , b_1 , b_2 não forem detectados, os mesmos são inicializados apontando para um pel imaginário localizado imediatamente depois do último pel da linha respectiva.
- O modo de codificação depende da posição relativa de b_2 em relação a a_1 . Se necessário, a magnitude da diferença entre b_2 e a_1 é então utilizada para selecionar entre modo horizontal e vertical.
- Os códigos são gerados alternando preto e branco até que o último pel da linha seja atingido (este é um pel imaginário localizado depois do último pel real e contendo cor oposta àquela do último pel)



Documentos Digitalizados - T6

- Uma tabela adicional de códigos bidimensionais é utilizada para definir o modo utilizado.
- O valor de extensão (*Extension*) permite que a codificação seja interrompida antes do final da página. Tal interrupção permite que parte de uma página seja transmitida com um mecanismo de codificação diferente, ou mesmo sem compressão.

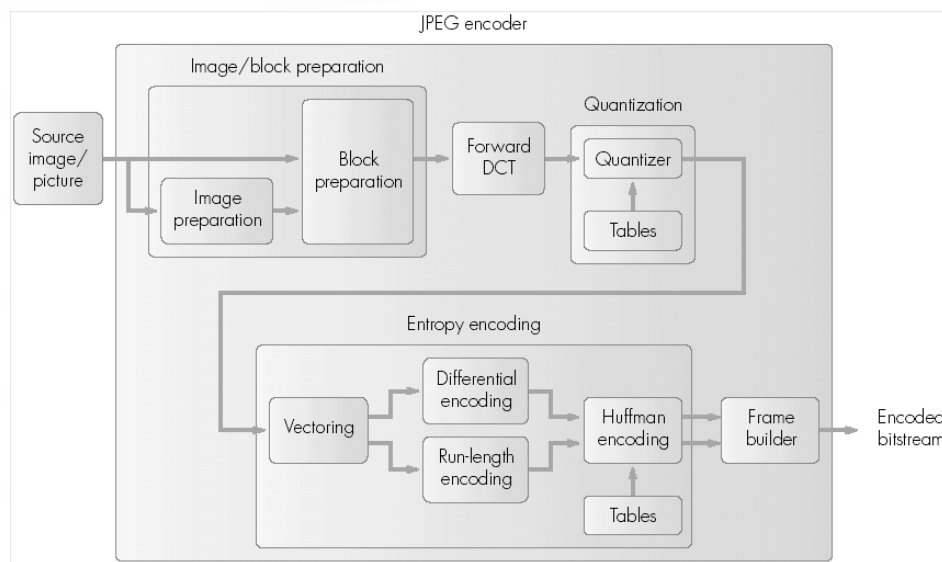
| Mode | Run-Length to be encoded | Abbreviation | Codeword |
|------------|--------------------------|--------------|---------------------------|
| Pass | $b_1 b_2$ | P | 0001 + $b_1 b_2$ |
| Horizontal | $a_0 a_1, a_1 a_2$ | H | 001 + $a_0 a_1 + a_1 a_2$ |
| Vertical | $a_1 b_1 = 0$ | V (0) | 1 |
| | $a_1 b_1 = -1$ | $V_R(1)$ | 011 |
| | $a_1 b_1 = -2$ | $V_R(2)$ | 000011 |
| | $a_1 b_1 = -3$ | $V_R(3)$ | 0000011 |
| | $a_1 b_1 = +1$ | $V_L(1)$ | 010 |
| | $a_1 b_1 = +2$ | $V_L(2)$ | 000010 |
| | $a_1 b_1 = +3$ | $V_L(3)$ | 0000010 |
| Extension | | | 0000001000 |

JPEG (IS 10918)

Joint Photographic Experts Group

- Um padrão para compressão e armazenamento de imagens desenvolvido por especialistas do ITU, ISO e IEC.
- Define vários modos de compressão, um para cada tipo de aplicação considerada.
 - Compressão sem perdas
 - Compressão com perdas
- Compressão Sequencial com Perdas (*lossy sequential mode* também conhecido como *baseline mode*) é o tipo de compressão mais adequada para multimídia, já que é o modo indicado para compressão de imagens digitalizadas, tanto coloridas como monocromáticas.

JPEG - *Baseline Mode*

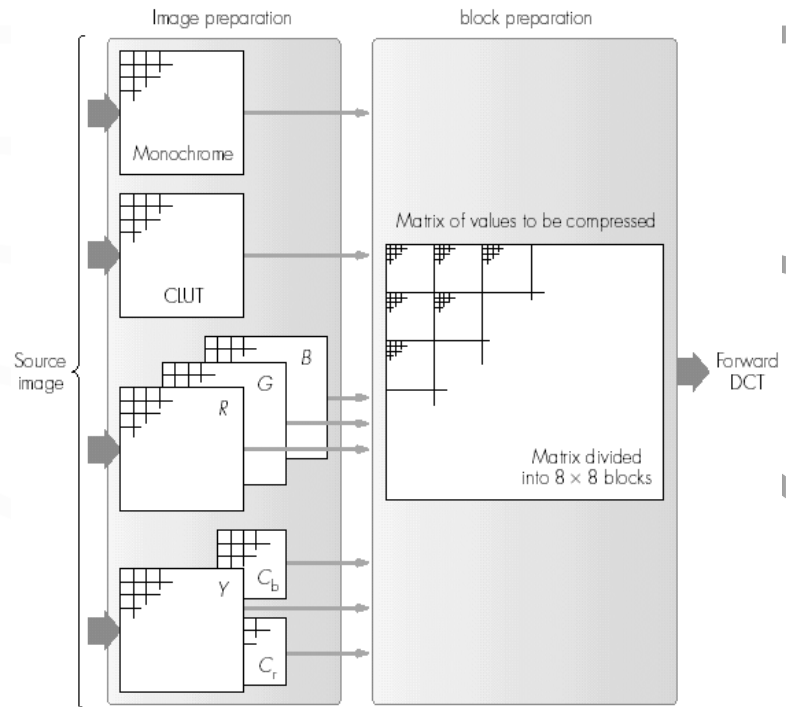


- Contém cinco etapas principais:
 - Preparação de Imagem e Bloco
 - Transformada Discreta de Cosenos
 - Quantização
 - Codificação por Entropia
 - Construção de Quadro

JPEG

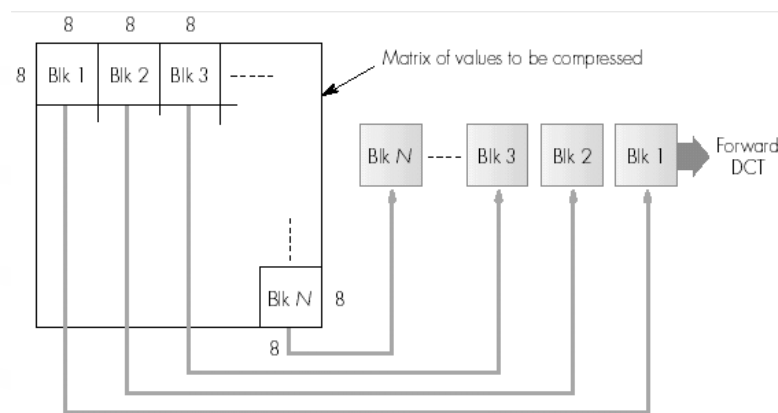
Preparação da Imagem e Bloco

- Seleciona-se o tipo de imagem a ser comprimida. Cada matriz segue, então, ao passo seguinte.
- Partição da imagem em blocos de 8×8 pels. Tal divisão permite uma execução mais eficiente do passo seguinte DCT.



JPEG

Preparação da Imagem e Bloco



- Os vários blocos são, então, submetidos um a um e em sequência à Transformada Discreta de Cosenos.

JPEG

Transformada Discreta de Cosenos

- Para o cálculo da DCT, os valores de pixel são padronizados com valores de -128 a +127, já que os valores anteriormente podem estar entre 0 e 255 (R, G, B, Y) ou -128 a +127 (C_R , C_B). Isso é realizado subtraindo-se 128 de componentes de luminância ou R, G, B.
- Considerando que $P[x,y]$ representa a matriz do bloco de 8×8 pixels e $F[i,j]$ representa a matriz transformada, os valores de DCT são calculados como segue:

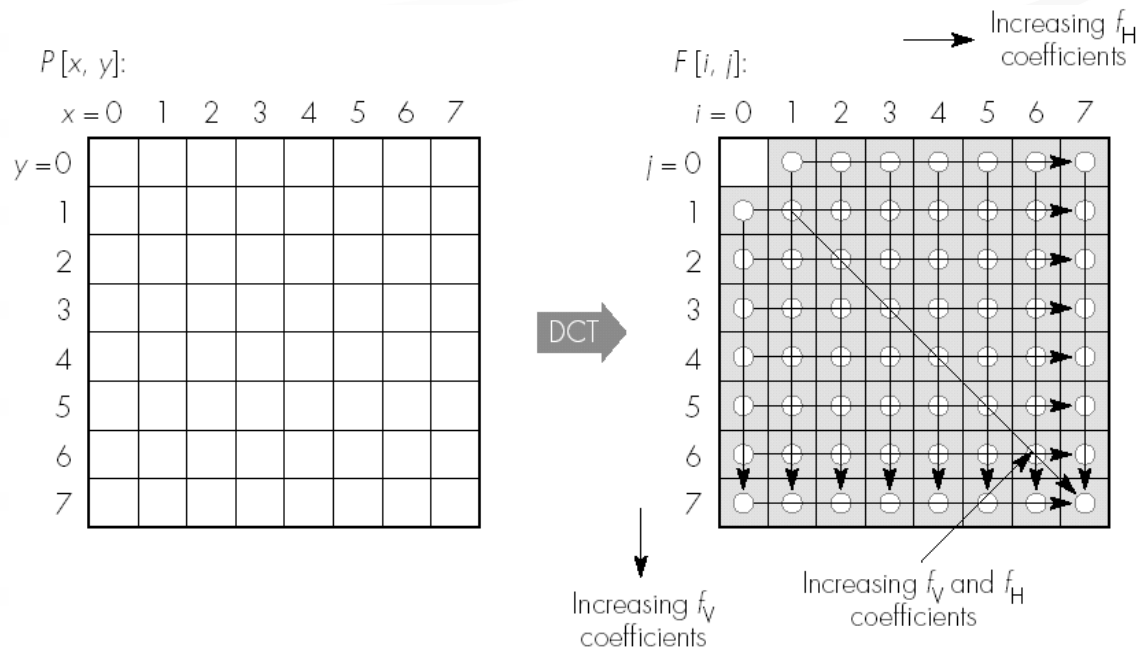
$$F[i, j] = \frac{1}{4} C(i)C(j) \sum_{x=0}^7 \sum_{y=0}^7 P[x, y] \cos \frac{(2x+1)i\pi}{16} \cos \frac{(2y+1)j\pi}{16}$$

onde $C(i)$ e $C(j) = 1/\sqrt{2}$ para $i,j=0$
= 1 para todos os outros valores de i e j .
 x, y, i e j todos variam de 0 a 7.

JPEG

Transformada Discreta de Cosenos

- Todos os 64 valores da matriz de entrada $P[x,y]$ contribuem para cada valor na matriz transformada $F[i,j]$.
- Para $i=j=0$, os termos de cosseno são ambos 1. Como $\cos(0)=1$, o valor em $F[0,0]$ da matriz transformada consiste de nada mais que uma função do somatório dos valores da matriz de entrada P . Mais precisamente, $F[0,0]$ é o valor médio dos 64 valores da matriz e é conhecido como coeficiente DC.
- Todos os outros valores na matriz são chamados coeficientes AC, já que os mesmos tem um componente de frequência horizontal ($x=1-7$, $y=0$), vertical ($y=1-7$, $x=0$) ou ambos ($x=1-7$, $y=1-7$) relacionados.
- Para $j=0$ (ou $i=0$) somente frequências horizontais (verticais) estão presentes com aumento de frequências para $i=1-7$ ($j=1-7$).
- Todas as outras posições ($j \neq 0$, $i \neq 0$) contém ambos componentes de frequência horizontais e verticais.



$P[x, y] = 8 \times 8$ matrix of pixel values

$F[i, j] = 8 \times 8$ matrix of transformed values/spatial frequency coefficients

In $F[i, j]$: = DC coefficient = AC coefficients

f_H = horizontal spatial frequency coefficient

f_V = vertical spatial frequency coefficient

JPEG

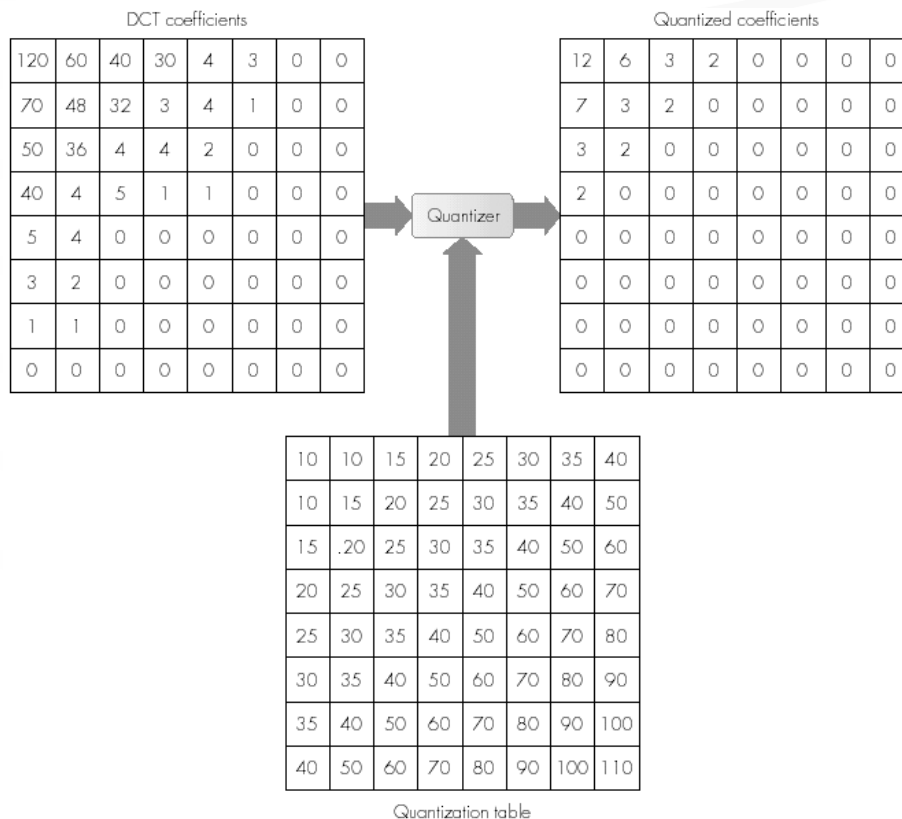
Transformada Discreta de Cosenos

- Considerando uma imagem de 640×480 pixels, a compressão da mesma iniciaria com a identificação do tipo de imagem, seguida do particionamento da imagem em blocos de 8×8 pixels. Seriam, portanto, necessários 80×60 blocos (um total de 4800 blocos). Se se considerar que a imagem enchesse uma tela de 16" (40cm), cada bloco teria apenas $5\text{mm} \times 5\text{mm}$.
 - As regiões que contém uma única cor gerarão matrizes transformadas cujos coeficientes DC tem valores idênticos (ou próximos) e poucos coeficientes AC. As regiões da imagem original que possuam transição de cores seriam as que levariam à criação de matrizes transformadas com coeficientes DC distintos e vários componentes AC.
 - São estas características que são utilizadas nos passos seguintes.
- DCT é uma operação que não introduz perdas. Alguma perda, porém, advém da aritmética de ponto flutuante da máquina em questão.

JPEG

Quantização

- O olho humano tem resposta considerável para o coeficiente DC e aqueles coeficientes com menor frequência espacial. Esta característica é utilizada pela quantização para descartar aqueles coeficientes que estejam acima de um determinado limite, selecionado de acordo com a capacidade visual humana.
- A quantização também busca diminuir a quantidade de dados a ser transmitidos através da divisão dos valores por aqueles numa tabela pré-definida (a tabela de quantização), o que diminui os coeficientes proporcionalmente à posição dos mesmos na matriz, sob pena de perda parcial de informação. A informação descartada não é, no caso ideal, perceptível ao olho humano.
- JPEG define duas tabelas padrão de quantização (uma para matrizes de luminância e outra para componentes de crominância, ele também permite o uso de tabelas customizadas).



JPEG

Quantização

- O cálculo dos valores durante a Quantização envolve arredondamento dos coeficientes resultantes.
- Os valores na tabela de quantização aumentam à medida que aumenta a frequência espacial (lembre que o olho humano é menos sensível à medida que a frequência espacial aumenta)
- O Coeficiente DC é o que é o maior armazenado.
- Muitos dos coeficientes de maior frequência espacial são “zerados” na matriz depois da Quantização.
- Os dois últimos ítems são utilizados na etapa seguinte da compressão.

JPEG

Codificação por Entropia - Vetorização

- Os algoritmos de codificação por entropia apresentados anteriormente operam sobre um vetor de símbolos. Deste modo é necessário converter o formato matricial de um bloco para um modo de vetor unidimensional.
- Considerando a matriz resultante quantizada, a simples utilização das linhas da matriz em sequência geraria um vetor com distribuição de zeros e não zeros em todo o vetor, o que não é muito bom para compressão. Para tomar melhor vantagem da matriz de quantizada, uma linearização em zig-zag é utilizada.
 - Tal metodologia de vetorização deixa o coeficiente DC (o maior) sendo o primeiro de todos, seguido imediatamente dos coeficientes AC de menor frequência espacial e no final os de maior frequência espacial.
 - A maioria dos valores não-zero se concentram no início do vetor.

JPEG - Codificação por Entropia

Codificação por Diferença

- O primeiro elemento do vetor é o coeficiente DC, que representa a cor/luminância/crominância média do bloco de 8×8 pixels, sendo o maior de todos. Devido à sua importância o coeficiente DC é mantido com a maior resolução possível durante a quantização.
- Por causa das pequenas dimensões de um bloco, os coeficientes DC de blocos vizinhos variam pouco de um para outro (imagine blocos vizinhos na textura de uma foto de um carro vermelho).
- A Codificação por Diferença é utilizada para codificação dos coeficientes DC, i.e. cada coeficiente é codificado como uma diferença para o coeficiente anterior (O primeiro DC é codificado em relação ao zero). Deste modo o número de bits, necessários para armazenar o resultado, é reduzido.
 - DCs: 12, 13, 11, 11, 10, ...
 - Diferenças: 12, 1, -2, 0, -1, ...

JPEG - Codificação por Entropia

Codificação por Diferença

| 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 |
| | | ⋮ |

- Os valores de diferenças são codificados no formato (*SSS, value*), onde *SSS* representa o número de bits necessários para armazenar o valor e *value* é a representação do valor naquele número de bits.
- valores positivos são codificados usando o formato sem sinal (*unsigned*) e valores negativos são codificados como complemento.
- Zero é codificado com um único bit zero no campo *SSS*.

JPEG - Codificação por Entropia

Codificação por Repetição de Série

- Os demais 63 valores do resultado da vetorização da matriz quantizada (a qual contém uma grande sequência de zeros) são codificados como uma string de pares de valores (*skip*, *value*), onde *skip* representa o número de zeros na sequência e *value* é o próximo coeficiente não zero.
 - (0,6)(0,7)(0,3)(0,3)(0,3)(0,2)(0,2)(0,2)(0,2)(0,0)
- A sequência (0,0) indica que todos os demais valores no vetor são zeros.
- O campo *value* acima é codificado no formato (*SSS*, *value*) descrito a pouco.
- Por causa do grande uso de codificação com palavras de tamanho variável, a etapa de codificação por entropia de JPEG também é conhecida como etapa *variable-length coding (VLC)*.

JPEG - Codificação por Entropia

Codificação de Huffman

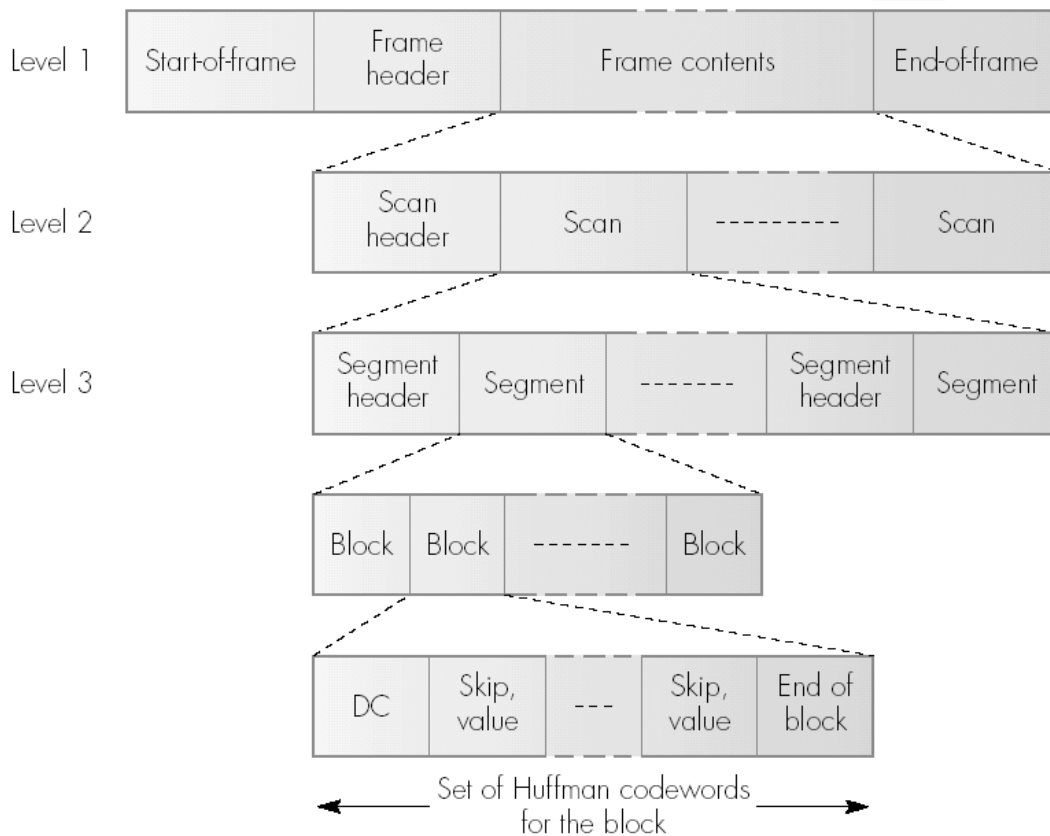
- Codificação de Huffman é utilizada numa etapa seguinte para codificar o resultado tanto da codificação diferencial quanto da codificação por repetição e série.
- Os bits do campo *SSS* são enviados como códigos de Huffman, o que permite que o receptor identifique os códigos sem ambiguidade, de acordo com o princípio do prefixo da codificação de Huffman.
- Ambos os componentes *skip* e *SSS* são codificados ou pelo código de Huffman ao lado ou por uma tabela customizada que é armazenada (enviada) no formato JPEG.
- Para permitir o decodificador diferenciar os códigos *skip* e *SSS* cada um é codificado separadamente e o par resultante é então substituído por um código de Huffman equivalente (ver tabela em anexo).

| 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 - Codificação por Entropia

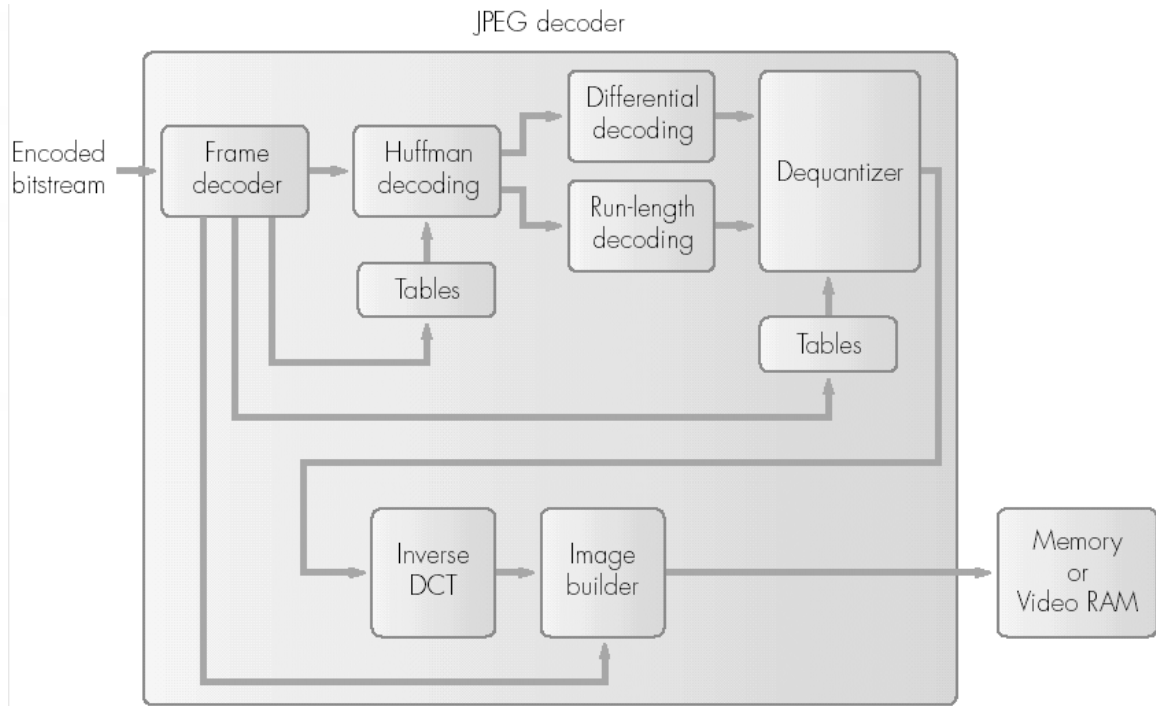
Construção de Quadro

- A etapa de construção de quadro consiste de encapsular os dados resultantes das etapas anteriores de um modo que seja reconhecível por um decodificador. Tal formato hierárquico é composto de várias camadas, cada uma com seus cabeçalhos próprios e demais campos:
 - Nível 1 (*Frame*), cujo cabeçalho define:
 - Largura e comprimento da imagem.
 - Número e tipo de componentes que formam a imagem (CLUT, RGB, $Y_C R_C B_C$).
 - Tipo de digitalização utilizada (4:2:2, 4:2:0, etc.).
 - Nível 2 (*Scan*), cujo cabeçalho define:
 - Identidade do componente (R, G, B, etc.).
 - Número de bits utilizado para codificar cada componente.
 - Tabelas de quantização utilizadas na codificação de cada componente.
 - Nível 3 (*Segment*):
 - O cabeçalho define as Tabelas de Códigos de Huffman utilizadas para codificar cada componente, caso as tabelas padrão não sejam utilizadas.
 - Um segmento é composto de um grupo de blocos.



JPEG

Decodificação



JPEG

Decodificação

- O Decodificador de Quadros identifica o conjunto de dados e as várias tabelas incluídas no quadro. O conjunto de dados passa então pelo decodificador de Huffman, que gera os coeficientes DC e AC de cada bloco. Estes passam então pelos decodificadores diferenciais ou de repetição de série (respectivamente para DCs e ACs). O resultado passa então pelo dequantizador usando a tabela apropriada e finalmente a Transformada Discreta de Cosenos Inversa (IDCT), que executa a seguinte operação:

$$P[x, y] = \frac{1}{4} \sum_{i=0}^7 \sum_{j=0}^7 C(i)C(j)F[i, j] \cos \frac{(2x+1)i\pi}{16} \cos \frac{(2y+1)j\pi}{16}$$

Onde $C(i)$ e $C(j) = \frac{1}{\sqrt{2}}$ para $i, j \neq 0$
 $= 1$ caso contrário.

- O construtor de imagens finalmente gera a imagem original a partir dos vários blocos gerados, utilizando a informação de controle recebida pelo decodificador de frames (tamanho da imagem, etc.)

JPEG

Considerações Finais

- Embora JPEG seja relativamente complicado por conta do número de etapas envolvidas, níveis de compressão de mais de 20:1 são alcançados com boa qualidade de imagem. Tal nível de compressão, no entanto, se aplica a imagens relativamente simples (poucas transições de cor). Níveis de compressão da ordem de 10:1 são mais comuns para imagens típicas.
- Como em GIFs, pode-se também codificar uma imagem JPEG em modo progressivo, i.e. envio de uma versão simplificada da imagem, seguida de níveis de detalhes, o que é útil para transmissões na Internet. Metodologias de transmissão progressiva incluem:
 - Modo progressivo: primeiro são transmitidos os coeficientes DC e componentes de baixa frequência dos blocos, seguidos dos coeficientes de maior frequência.
 - Modo hierárquico: A imagem é enviada inicialmente com uma resolução menor (ex. 320×240), seguida de maior resolução (ex. 640×480).

Exercícios

Exercício 1

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.

Exercício 2

Determine the encoded version of the following difference values which relate to the encoded DC coefficients from consecutive DCT blocks:

12, 1, -2, 0, -1

Answer:

| Value | SSS | Value |
|-------|-----|-------|
| 12 | 4 | 1100 |
| 1 | 1 | 1 |
| -2 | 2 | 01 |
| 0 | 0 | |
| -1 | 1 | 0 |

Exercício 3

Derive the binary form of the following run-length encoded AC coefficients:

(0,6) (0,7) (3,3) (0,-1) (0,0)

Answer:

| <i>AC coefficients</i> | <i>Skip</i> | <i>SSS/Value</i> | |
|------------------------|-------------|------------------|-----|
| 0,6 | 0 | 3 | 110 |
| 0,7 | 0 | 3 | 111 |
| 3,3 | 3 | 2 | 11 |
| 0,-1 | 0 | 1 | 0 |
| 0,0 | 0 | 0 | |

Exercício 4

Determine the Huffman-encoded version of the following difference values which relate to the encoded DCT coefficients from consecutive DCT blocks.

12, 1, -2, 0, -1

Use for example purposes, the default Huffman codewords defined earlier in Figure 3.19(b).

Answer:

| <i>Value</i> | <i>SSS</i> | <i>Huffman-encoded SSS</i> | <i>Encoded value</i> | <i>Encoded bitstream</i> |
|--------------|------------|----------------------------|----------------------|--------------------------|
| 12 | 4 | 101 | 1100 | 1011100 |
| 1 | 1 | 011 | 1 | 0111 |
| -2 | 2 | 100 | 01 | 10001 |
| 0 | 0 | 010 | | 010 |
| -1 | 1 | 011 | 0 | 0110 |

As we can deduce from the set of Huffman-encoded SSS fields, they illustrate that, providing the decoder uses the same set of codewords, it can readily determine the SSS field from the received (encoded) bitstream by searching the bitstream bit-by-bit – starting from the leftmost bit – until it reaches a valid codeword. The number of bits in the corresponding SSS value is then read from the table in Figure 3.19(b) and this is used to determine the number of following bits in the bitstream that contain the related *value*.

Exercício 5

Derive the composite binary symbols for the following set of run-length encoded AC coefficients:

(0,6) (0,7) (3,3) (0, -1) (0,0)

Assuming the *skip* and SSS fields are both encoded as a composite symbol, use the Huffman codewords shown in Table 3.2 to derive the Huffman-encoded bitstream for this set of symbols.

Answer:

The *skip* and *SSS* fields for this set of AC coefficients were derived earlier in Example 3.6. Hence:

| <i>AC coefficient</i> | <i>Composite symbol skip</i> | <i>SSS</i> | <i>Huffman codeword</i> | <i>Run-length value</i> |
|-----------------------|----------------------------------|------------|-------------------------|-----------------------------|
| 0,6 | 0 | 3 | 100 | 6 = 110 |
| 0,7 | 0 | 3 | 100 | 7 = 111 |
| 3,3 | 3 | 2 | 111110111 | 2 = 10 |
| 0, -1 | 0 | 1 | 00 | 1 = 0 |
| 0,0 | 0 | 0 | 1010 | |

The Huffman-encoded bitstream is then derived by adding the run-length encoded value to each of the Huffman codewords:

100110 100111 11111011110 000 1010