

UNIP

UNIVERSIDADE PAULISTA

Processamento de Imagem

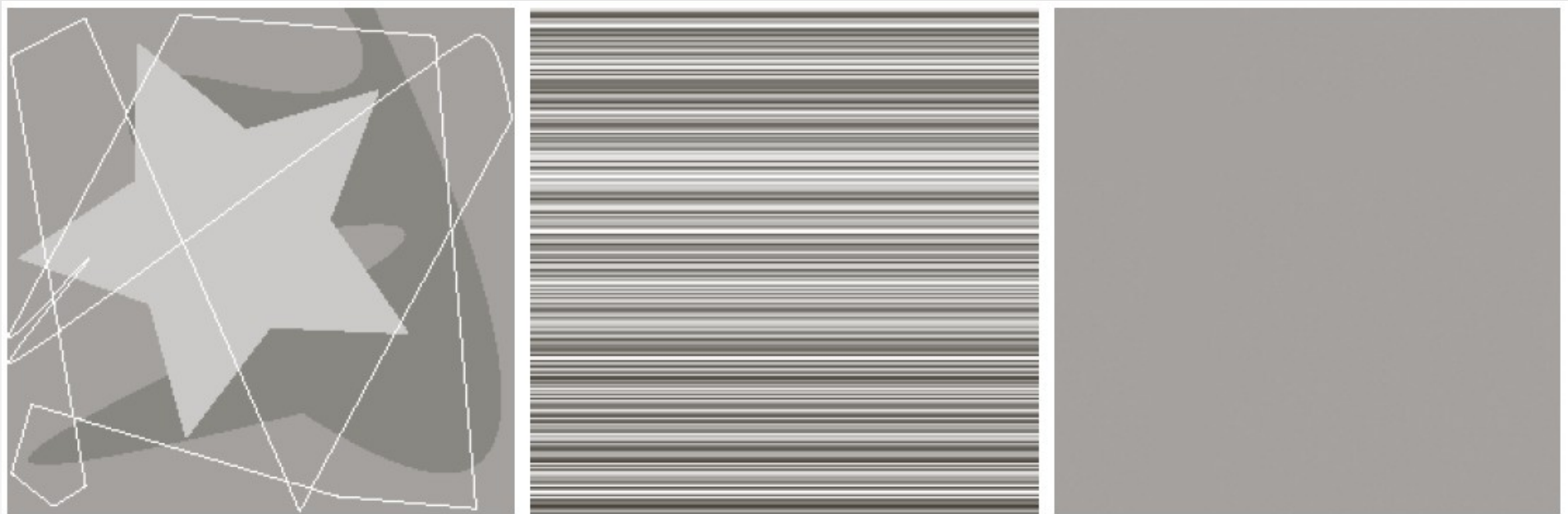


Compressão de Imagens

Professora Sheila Cáceres

Porque comprimir?

- Técnicas de compressão surgiram para reduzir o espaço requerido para armazenamento e o tempo necessário para transmissão de imagens.



a b c

FIGURE 8.1 Computer generated $256 \times 256 \times 8$ bit images with (a) coding redundancy, (b) spatial redundancy, and (c) irrelevant information. (Each was designed to demonstrate one principal redundancy but may exhibit others as well.)

Image Compression Standards, Formats, and Containers

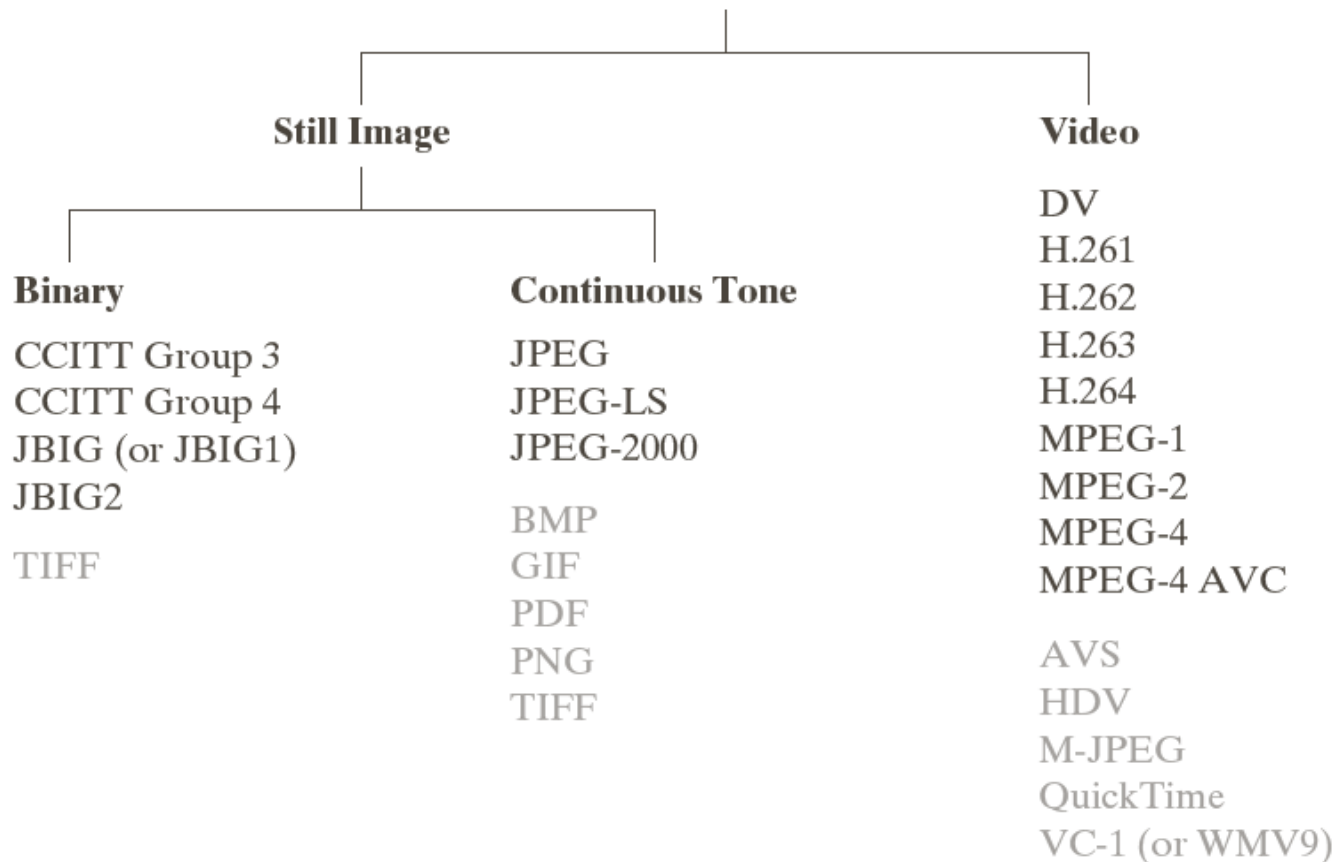


FIGURE 8.6 Some popular image compression standards, file formats, and containers. Internationally sanctioned entries are shown in black; all others are grayed.

Taxa de compressão e Redundância de dados

- A taxa de compressão pode ser expressa como>

$$C_R = \frac{n_1}{n_2}$$

- Onde n_1 é o tamanho da imagem original e n_2 é o tamanho da imagem comprimida.
- A redundância dos dados relativa R_D é uma entidade matematicamente quantificável:

$$R_D = 1 - \frac{1}{C_R}$$

- **Exemplo:** Uma taxa de compressão de 10 (ou 10:1) significa que o primeiro conjunto (img original) de dados possui 10 unidades de informação para cada unidade no segundo conjunto (comprimido). Ou seja 90% dos dados da imagem original são redundantes, dado que a img compactada tem a mesma quantidade de informação com apenas 10% dos dados utilizados na imagem original.

Redundância de Dados

- As redundâncias de dados podem ser classificadas em três categorias:
 - R. de codificação
 - R. interpixel
 - R. psicovisual

Taxa de compressão e Redundância de dados

- A taxa de compressão pode ser expressa como>

$$C_R = \frac{n_1}{n_2}$$

- Onde n_1 é o tamanho da imagem original e n_2 é o tamanho da imagem comprimida.
- A redundância dos dados relativa R_D é uma entidade matematicamente quantificável:

$$R_D = 1 - \frac{1}{C_R}$$

- **Exemplo:** Uma taxa de compressão de 10 (ou 10:1) significa que o primeiro conjunto (img original) de dados possui 10 unidades de informação para cada unidade no segundo conjunto (comprimido). Ou seja 90% dos dados da imagem original são redundantes, dado que a img compactada tem a mesma quantidade de informação com apenas 10% dos dados utilizados na imagem original.

Redundância de Codificação

- Se os códigos utilizados para representar os níveis de cinza tiverem um número de símbolos (bits) maior que o valor absolutamente necessário, a imagem apresentará redundância de codificação.

Redundância de Codificação

- Cada um dos níveis de cinza em uma imagem ocorre com probabilidade

$$p_i = \frac{n_i}{n}$$

Onde

* n_i é o número de pixels de intensidade i

* n é o número total de pixels na imagem

- O número médio de bits necessários para codificar cada pixel em uma imagem é dado por:

$$\bar{L} = \sum_{i=0}^{L-1} l(i) p_i$$

Onde

* $l(i)_i$ é o número de bits utilizados

para representar cada nível de cinza i

Redundância de Codificação, exemplo

Seja uma imagem com sete níveis de cinza distintos, conforme a distribuição de probabilidade mostrada na tabela a seguir.

Nível de cinza (i)	0	1	2	3	4	5	6
Probabilidade $P(i)$	$1/2$	$1/4$	$1/8$	$1/16$	$1/32$	$1/64$	$1/128$

Tabela: Distribuição dos níveis de cinza em uma imagem monocromática.

Como há sete níveis distintos, seriam necessários 3 bits para representar cada nível de cinza. Entretanto, caso fosse utilizada a representação da tabela a seguir

Nível de cinza (i)	0	1	2	3	4	5	6
Código	0	10	110	1110	11110	111110	1111110

Tabela: Codificação dos níveis de cinza.

Redundância de Codificação, exemplo

$$\begin{aligned}\bar{L} &= \sum_{i=0}^6 l(i) p_i = 1(1/2) + 2(1/4) + 3(1/8) + 4(1/16) + 5(1/32) + \\ &= 6(1/64) + 7(1/128) \approx 1.93 \text{ bits}\end{aligned}$$

A partir da equação 1, tem-se que a taxa de compressão resultante C_R é igual a $3/1.93$ ou 1.55 . Assim, o nível de redundância, determinado pela equação 2, é dado por

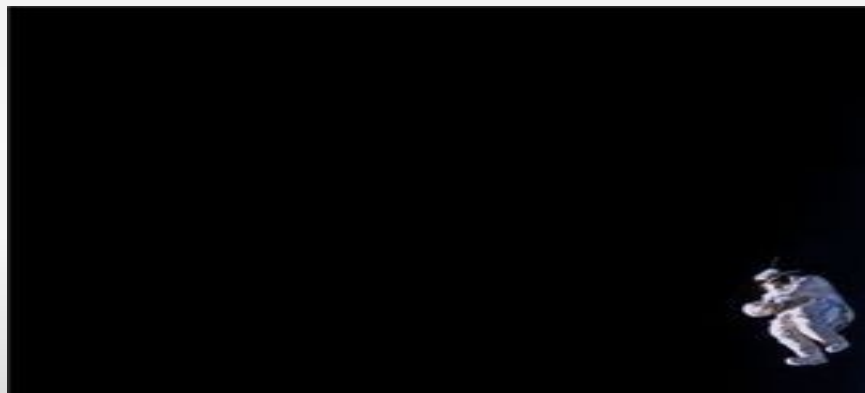
$$R_D = 1 - \frac{1}{1.55} \approx 0.35$$

Redundância Interpixel

- Explora a característica de que pixels vizinhos em uma imagem normalmente possuem alguma relação ou similaridade.
- Tipicamente a maioria dos pixels possui valores não muito diferentes de seus vizinhos.
- Assim, o valor de um pixel é razoavelmente previsível a partir dos valores dos seus vizinhos.
- A representação matricial não seria mais um modelo eficiente. Podemos utilizar modelos mais adequados evitando que as mesmas informações sejam associadas a cada pixel e seus vizinhos.

Redundância Interpixel

- Uma representação para aproveitar este tipo de redundância é a **Codificação por comprimento de corrida**.
- **Cod por comprimento de corrida** percorre cada linha da imagem e armazena apenas a **intensidade** ou cor e o **número de pixels** iguais para cada grupo de pixels idênticos
- Exemplo: na imagem a seguir, as primeiras linhas seriam representadas por (0, 300) onde a cor preta 0 acontece 300 vezes.



Redundância Psicovisual

- Podemos também explorar a imprecisão do sistema visual humano em perceber certos detalhes em uma imagem.
- Assim, uma pequena diferença de intensidade luminosa entre duas áreas pode não ser percebida pelo olho (tirar vantagem das informações psicolvisualmente redundantes).

Redundância Psicovisual, exemplo



(a) $L = 256$



(b) $L = 128$



(c) $L = 64$



(d) $L = 32$

Métodos de compressão de imagens

- Normalmente são classificadas em duas categorias
 - Sem perda
 - Com perda

Sem Perda

- A imagem resultante após o processo de descompactação é exatamente igual à imagem original.
- Aplicado em situações em que os dados são de difícil aquisição ou contêm informação que não deve ser alterada pelo processo de compressão (diagnóstico médico de imagens por exemplo).

Codificação de Huffman

- Explora apenas a redundância de codificação para descompactar imagens.
- Cada símbolo pode ser substituído por um código de comprimento variável onde os símbolos que ocorrem mais frequentemente recebem códigos menores, e códigos menos frequentes são de maior tamanho.
- Para a correta identificação, nenhum código pode ser prefixo de outro código de comprimento maior (códigos livres de prefixo)

Codificação de Huffman

A) Redução de fontes

- Primeiramente se ordenam decrescentemente os símbolos de acordo com as probabilidades com que ocorrem.
- Posteriormente se agrupa os dois símbolos de menor probabilidade em um novo símbolo.
- Esses dois primeiros passos são repetidos até que restem apenas dois símbolos.

Original source		Source reduction			
Symbol	Probability	1	2	3	4
a_2	0.4	0.4	0.4	0.4	0.6 0.4
a_6	0.3	0.3	0.3	0.3	
a_1	0.1	0.1	0.2	0.3	
a_4	0.1	0.1			
a_3	0.06	0.1			
a_5	0.04				

B) Posteriormente atribui-se um código a cada fonte reduzida iniciando na ordem inversa a que foram obtidas até a fonte original.

- Podemos atribuir os símbolos 0 a cima e 1 abaixo. Ou seja adiciona-se um bit a cada símbolo previamente agrupado. Essa operação é repetida para cada fonte reduzida até que a fonte original seja atingida
- O processo A, pode se assemelhar a representação das fontes em forma de árvore onde podemos atribuir 0 a cada filho da direita e 1 a cada filho da esquerda.

Original source		Source reduction								
Symbol	Probability	Code	1		2		3		4	
a_2	0.4	1	0.4	1	0.4	1	0.4	1	0.6	0
a_6	0.3	00	0.3	00	0.3	00	0.3	00	0.4	1
a_1	0.1	011	0.1	011	0.2	010	0.3	01		
a_4	0.1	0100	0.1	0100	0.1	011				
a_3	0.06	01010	0.1	0101						
a_5	0.04	01011								

- $L = 0.40 * 1 + 0.30 * 2 + 0.10 * 3 + 0.10 * 4 + 0.06 * 5 + 0.04 * 5 = 2.2 \text{ bits/símbolo}$

Algoritmo 1 Codificação de Huffman

- 1: Os símbolos da fonte são listados em ordem decrescente de probabilidade.
- 2: Os dois símbolos com menor probabilidade recebem o valor 0 e 1. Esses dois símbolos são combinados em um novo símbolo, cuja probabilidade é igual a soma das duas probabilidades originais. A probabilidade do novo símbolo é ordenada de acordo com seu valor.
- 3: O procedimento é repetido até que os dois últimos símbolos recebam o valor 0 e 1.
- 4: O código para cada símbolo original da fonte é formado pelo percurso inverso na sequência de valores 0 e 1.

Codificação de Huffman

- Explora apenas a redundância de codificação para descompactar imagens.
- Cada símbolo pode ser substituído por um código de comprimento variável onde os símbolos que ocorrem mais frequentemente recebem códigos menores, e códigos menos frequentes são de maior tamanho.
- Para a correta identificação, nenhum código pode ser prefixo de outro código de comprimento maior (códigos livres de prefixo)

Com Perda

- Nem toda a informação é recuperada após a descompactação.
- Aplicada tipicamente em videoconferência e televisão digital em que a perda de certas informações pode ser tolerada pelo receptor.

Value	Rating	Description
1	Excellent	An image of extremely high quality, as good as you could desire.
2	Fine	An image of high quality, providing enjoyable viewing. Interference is not objectionable.
3	Passable	An image of acceptable quality. Interference is not objectionable.
4	Marginal	An image of poor quality; you wish you could improve it. Interference is somewhat objectionable.
5	Inferior	A very poor image, but you could watch it. Objectionable interference is definitely present.
6	Unusable	An image so bad that you could not watch it.

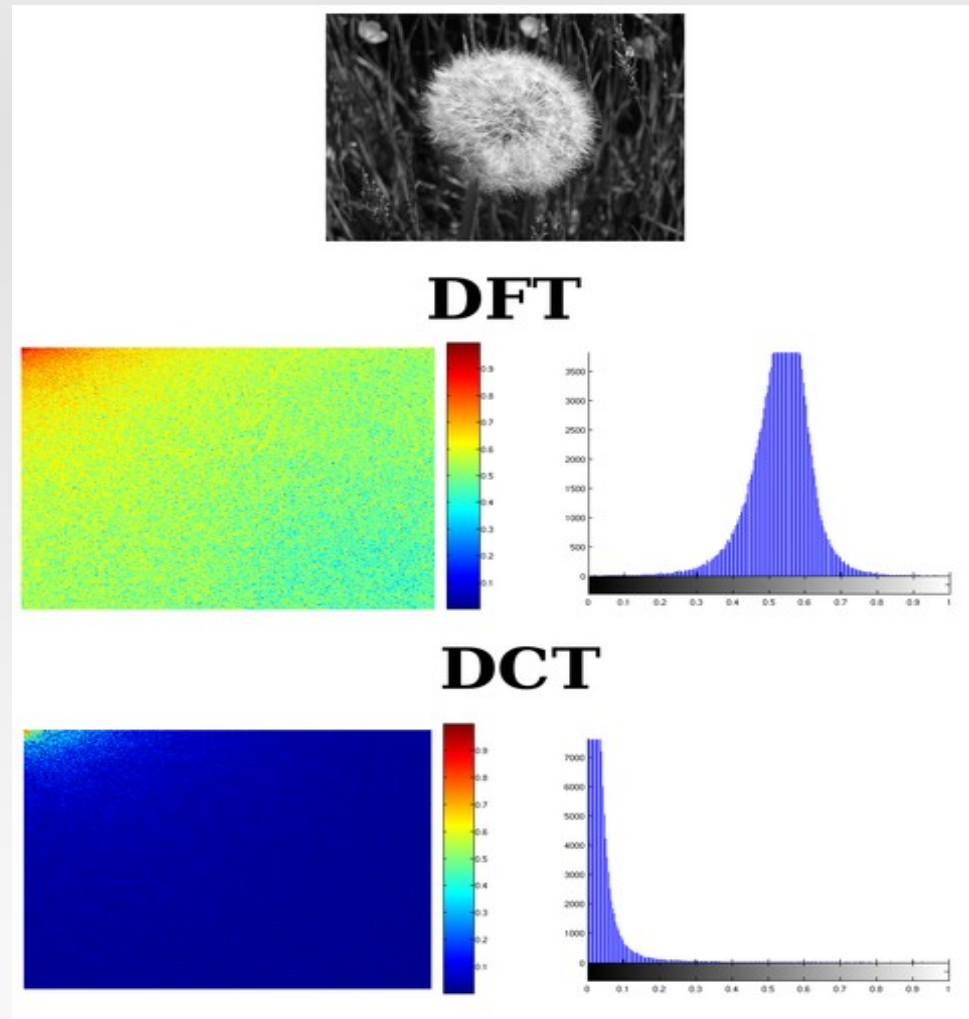
TABLE 8.2
Rating scale of the Television Allocations Study Organization. (Frendendall and Behrend.)

Transformada Discreta do Coseno

- Também chamada DCT do inglês Discrete Cosine Transform.
- Ela apresenta uma elevada compactação de energia em regiões próximas da origem, aproximando-se do desempenho de outra transformada conhecida como Karhunen-Loeve.
- Ou seja, grande parte da informação da imagem encontra-se próxima à origem.
- Com a compactação da energia após a execução da transformada, muitos coeficientes apresentarão valores **próximos a zero** e poderão ser **eliminados** sem que haja perda significativa de informações contidas na imagem.

Transformada Discreta do Coseno

Comparação entre a Transformada discreta de Fourier e a DCT: pode se observar o acúmulo dos coeficientes mais significativos no canto superior direito da imagem da DCT, proporcionando melhor capacidade de compressão



Transformada Discreta do Coseno

- A imagem é subdividida em blocos geralmente de 8 x 8 pixels (nas formulas consideraremos esse tamanho).
- A transformada discreta do cosseno é aplicada a cada um desses blocos conforme a equação:

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

- Onde

$$C(u) = C(v) = \begin{cases} \frac{1}{\sqrt{2}}, & \text{para } u, v = 0 \\ 0, & \text{para } u, v > 0 \end{cases}$$

- Em uma imagem típica muitos dos coeficientes $F(u, v)$ terão valor de zero ou próximo de zero no domínio DCT e poderão ser descartados.

Transformada Discreta do Coseno

- Para mapear de volta a informação do domínio de frequência para o domínio espacial usamos a transformada DCT **inversa**:

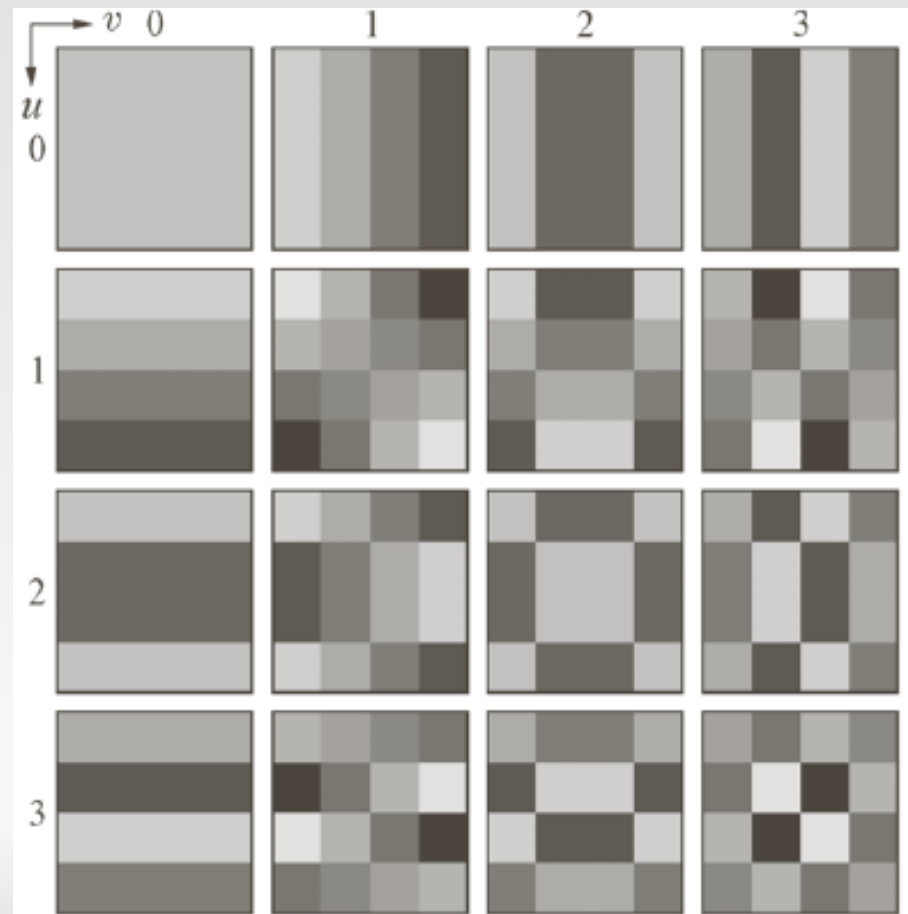
$$f(x, y) = \frac{1}{4} \sum_{u=0}^7 \sum_{v=0}^7 C(u)C(v)F(u, v) \cos \frac{\pi u(2x + 1)}{16} \cos \frac{\pi v(2y + 1)}{16}$$

Onde

$$C(u) = C(v) = \begin{cases} \frac{1}{\sqrt{2}}, & \text{para } u, v = 0 \\ 0, & \text{para } u, v > 0 \end{cases}$$

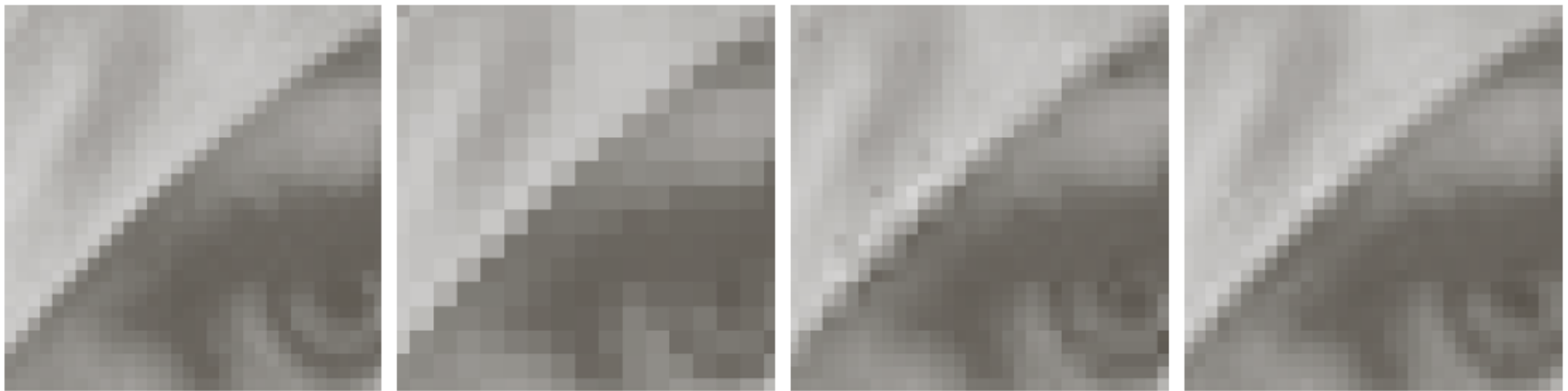
- Em uma imagem típica muitos dos coeficientes $F(u, v)$ terão valor de zero ou próximo de zero no domínio DCT e poderão ser descartados.

- Discrete-cosine basis function para 4. The origem de cada bloco esta no canto superior esquerdo



Transformada Discreta do Coseno

- É mais barato fazer varias DCTs em blocos pequenos do que uma DCT da imagem toda.
- Viu-se que uma boa aproximação é ter blocos de 8×8 .
- A padronização JPEG considera blocos de 8×8 .



a b c d

FIGURE 8.27 Approximations of Fig. 8.27(a) using 25% of the DCT coefficients and (b) 2×2 subimages, (c) 4×4 subimages, and (d) 8×8 subimages. The original image in (a) is a zoomed section of Fig. 8.9(a).

Transformada Discreta do Coseno



a	b
c	d

FIGURE 8.28
Approximations
of Fig. 8.9(a) using
12.5% of the
 8×8 DCT
coefficients:
(a)–(b) threshold
coding results;
(c)–(d) zonal
coding results. The
difference images
are scaled by 4.

Codificação baseada em Dicionário (LZW)

- Informações sobre a sequência de símbolos que estão nos dados originais são mantidas em uma tabela ou dicionário.
- Os dados codificados consistem em uma sequência de índices para este dicionário.
- Um problema com o algoritmo LZW é o tamanho do dicionário. À medida que o número de entradas do dicionário aumenta, a realização de buscas e inserções torna-se dispendiosa. Para minimizar esse problema, usa-se uma função de dispersão (hashing).

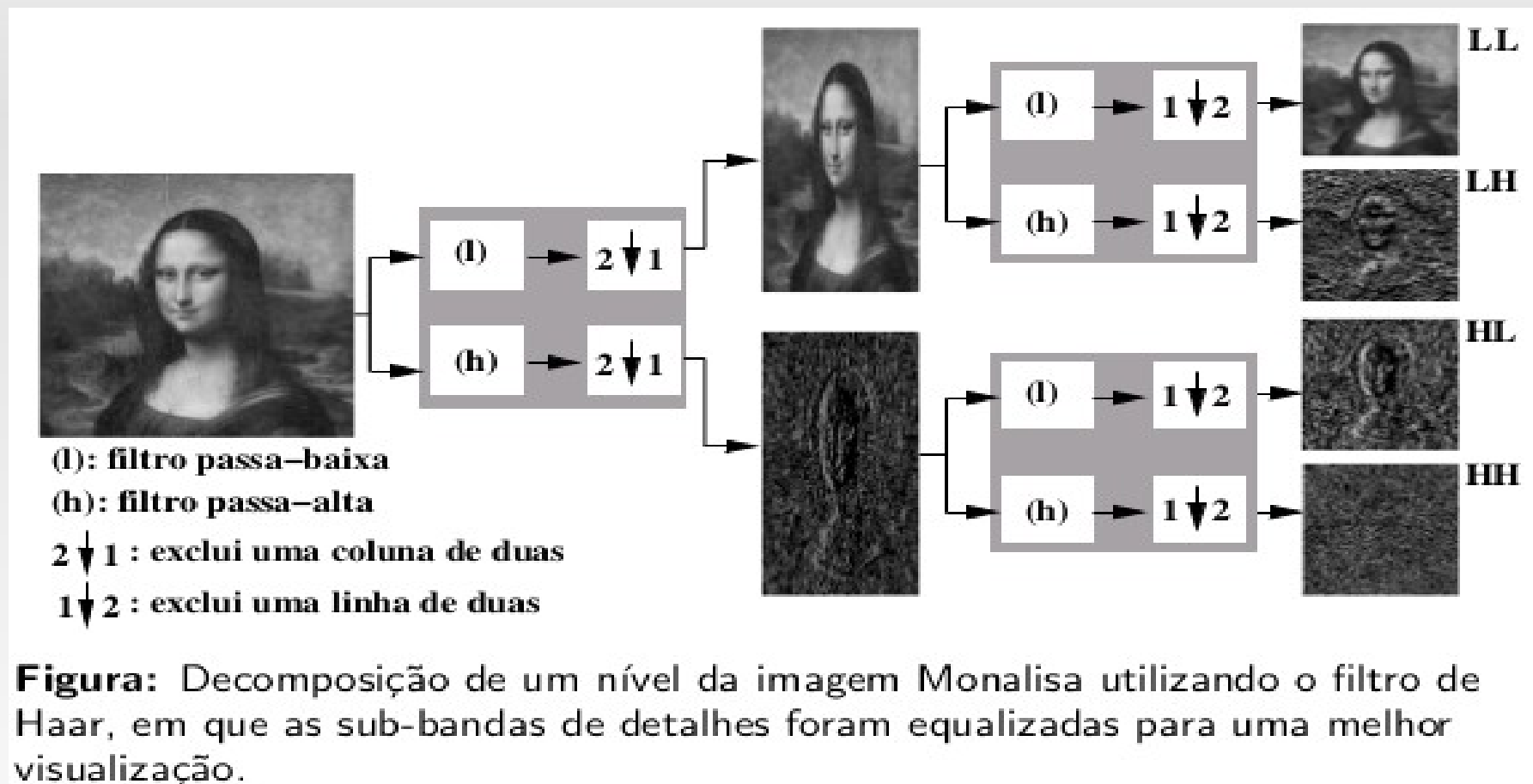
Codificação baseada em Dicionário (LZW)

- Informações sobre a sequência de símbolos que estão nos dados originais são mantidas em uma tabela ou dicionário.
- Os dados codificados consistem em uma sequência de índices para este dicionário.
- Um problema com o algoritmo LZW é o tamanho do dicionário. À medida que o número de entradas do dicionário aumenta, a realização de buscas e inserções torna-se dispendiosa. Para minimizar esse problema, usa-se uma função de dispersão (hashing).

Wavelets

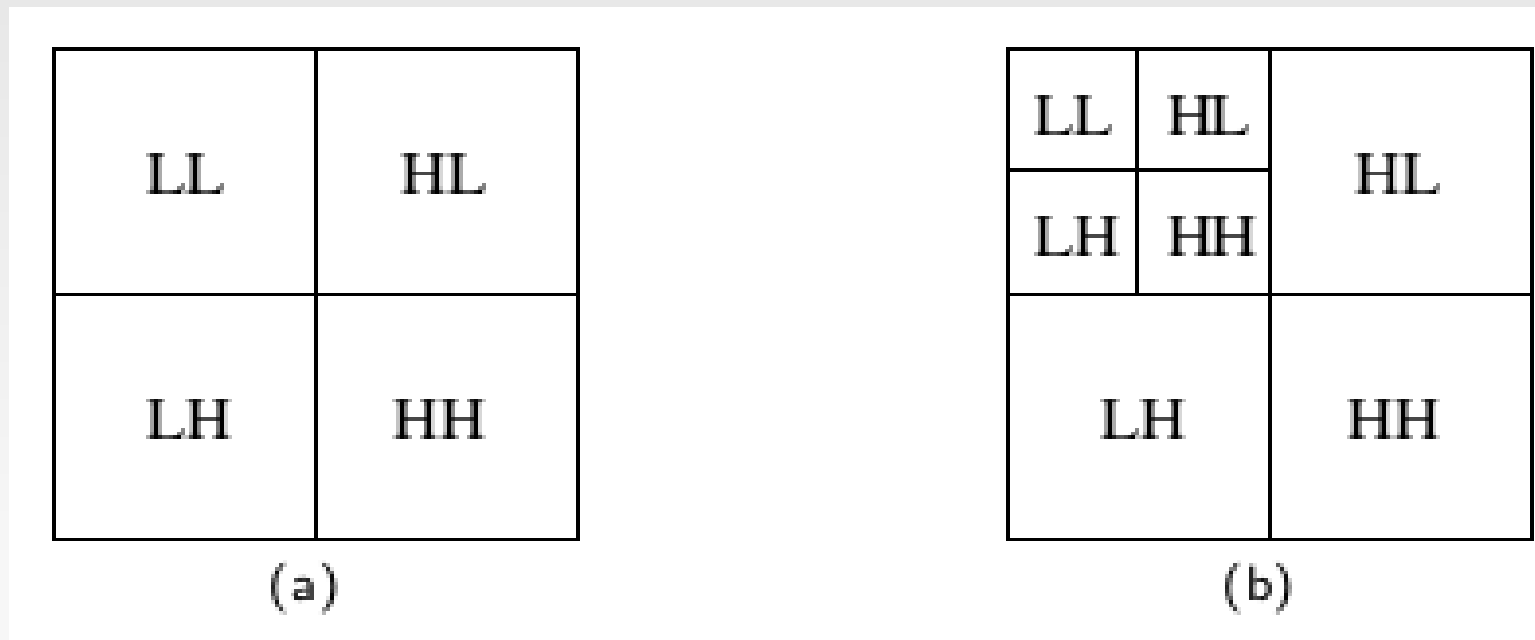
- Vimos que a transformada de Fourier permite descrever as diferentes frequências contidas na imagem sem considerar a localização espacial.
- Para adicionar informação espacial pode se usar a transformada de Wavelet.
- Assim, wavelets altera a representação de modo a obter informações sobre a frequência e informações espaciais.
- a transformada wavelet é semelhante a um banco de filtros que divide o sinal em bandas distintas formado por filtros passa-baixa e passa-alta sucessivos.

Wavelets



Wavelets

- Transformadas bidimensionais
- (a) decomposição em um nível
- (b) decomposição em dois níveis



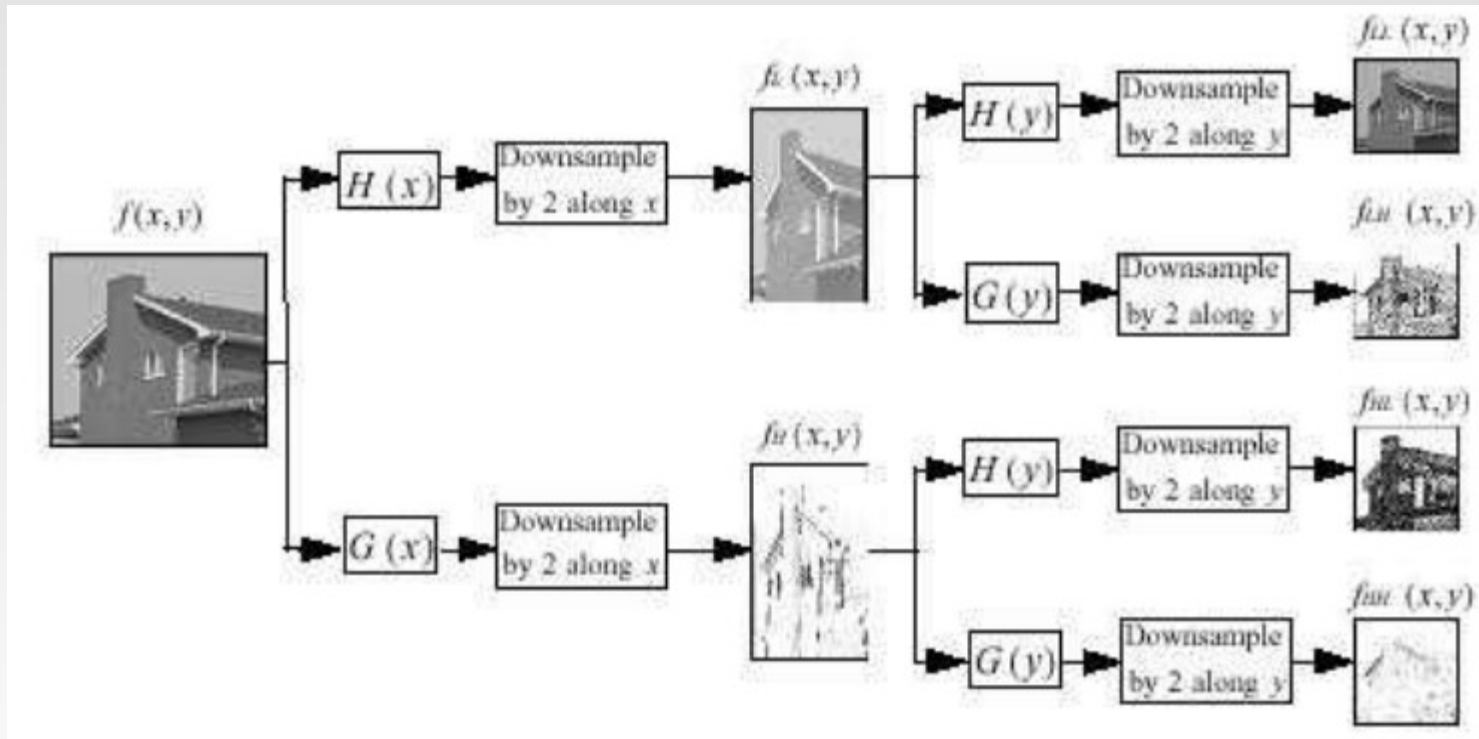
Compressão por Wavelets

- Suponha a imagem $f(x)$ expressa pela soma de funções base

$$f(x) = \sum_{i=1}^m c_i u_i(x)$$

- O conjunto de dados neste caso consiste de coeficientes c_i . Procura-se uma função que aproxima $f(x)$, mas com menos coeficientes:

$$\tilde{f}(x) = \sum_{i=1}^{\tilde{m}} \tilde{c}_i \tilde{u}_i(x) \cong f(x)$$



Compressão por Wavelets



Bibliografia

- PEDRINI, H. e SCHWARTZ, W. R., "Análise de Imagens Digitais", São Paulo, Thomson, 2008, 508p e slides.
- FALCÃO, A.
(<http://www.ic.unicamp.br/~afalcao/mo443/>)
- Wilhelm BURGER e Mark James BURGE. Digital Image Processing, An Algorithmic Introduction using Java.
- GONZALEZ e WOODS. Processamento de Imagens Digitais, Segunda edição.

Várias imagens foram extraídas do material mencionado acima com fines didáticos.