

Capítulo 8

Compressão de Imagem

Compressão de Imagem: Definição

Formas de diminuir a área de armazenamento dos dados, reduzindo a quantidade de bits para representar os dados (imagem, texto, ou arquivo qualquer).

Em compressão de imagem define-se como a forma (algoritmos e métodos) de armazenar informações visuais mais compactamente.

Capítulo 8

- 8.1. Redundâncias na Imagem
- 8.2. Métodos de Compressão de Imagem
- 8.3. Elementos da teoria de informação
- 8.4. Entropia da Imagem
- 8.5. Métodos de Codificação sem perda
- 8.6. Transformada Discreta do Co-seno (DCT)
- 8.7. Compressão Fractal
- 8.8. Compressão por *Wavelets*
- 8.9. Padrões de Compressão de Imagem

8.1. Redundâncias na Imagem

8.1.1. Compressão de imagens e modelos de cores

8.1.2. Medição do Desempenho

8.1.3. Critérios de fidelidade objetivos

8.1.4. Critérios de fidelidade subjetivos

8.1.5. Modelos de compressão de imagens

8.1. Redundâncias na Imagem

Tipos de redundância em imagens:

De codificação de tons ou cor - quando os níveis de cinza ou as cores de uma imagem são codificados com mais símbolos de codificação do que o necessário.

Inter-pixel - resultantes das relações geométricas ou estruturais entre os objetos na imagem.

Espectral - ocorre em imagens com mais de uma faixa espectral, quando os valores espectrais, para a mesma posição na matriz de *pixels* de cada banda, são correlacionados.

Psicovisuais - relacionadas ao fato do sistema visual humano não responder com a mesma sensibilidade a todas as informações visuais.

8.1.1. Compressão de imagens e modelos de cores

- YIQ (para transmissão de televisão);
- RGB (para monitores de computador colorido); CMY (para impressoras coloridas);
- HSI (*Hue, Saturation, Intensity* ou matiz, saturação, intensidade);
- HSV (*Hue* ou matiz, Saturação e Valor);
- YCBCR - importante para a compressão de imagens (ele é usado no formato de imagens JPEG).

8.1.2. Medição do Desempenho

Medida de desempenho -> taxa de compressão (razão entre o tamanho do dado ou imagem original e o tamanho do dado após a compressão).

Técnicas sem perda: quanto maior a taxa de compressão melhor é a técnica de compressão.

Técnicas com perda: deve-se considerar também a qualidade do sinal ou dado reconstruído.

Critérios de fidelidade: se a remoção de dados causou perda de informação visual. Podem ser: quantitativos ou subjetivos.

8.1.3 - Critérios de fidelidade objetivos

Funções de avaliação:

- Erro Total.
- Raiz Quadrada do Quadrado da Média dos Erros (Root Mean Square Error - e_{rms}).
- Relação Sinal Ruído (*Signal To Noise Ratio* - SNR_{ms} SNR).
- Relação Sinal Ruído de Pico (*Peak Signal to Noise Ratio* - PSNR).

Sendo $F(x, y)$ a imagem original e $G(x, y)$ a imagem reconstruída, tem-se:

Erro Total ou absoluto:

$$e_t = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} |G(x, y) - F(x, y)| \quad (8.2)$$

Raiz Quadrada do Quadrado da Média dos Erros:

$$e_{rms} = \sqrt{\left[\frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} [G(x, y) - F(x, y)]^2 \right]} \quad (8.3)$$

Razão ou Relação Sinal Ruído:

$$SNR_{rms} = \sqrt{\frac{\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} G(x, y)^2}{\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} e(x, y)^2}} = \sqrt{\frac{\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} G(x, y)^2}{\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} [G(x, y) - F(x, y)]^2}} \quad (8.5)$$

Relação Sinal Ruído de Pico:

$$PSNR = 20 \log_{10} \left(\frac{2^n - 1}{e_{rms}} \right) \quad (8.6)$$

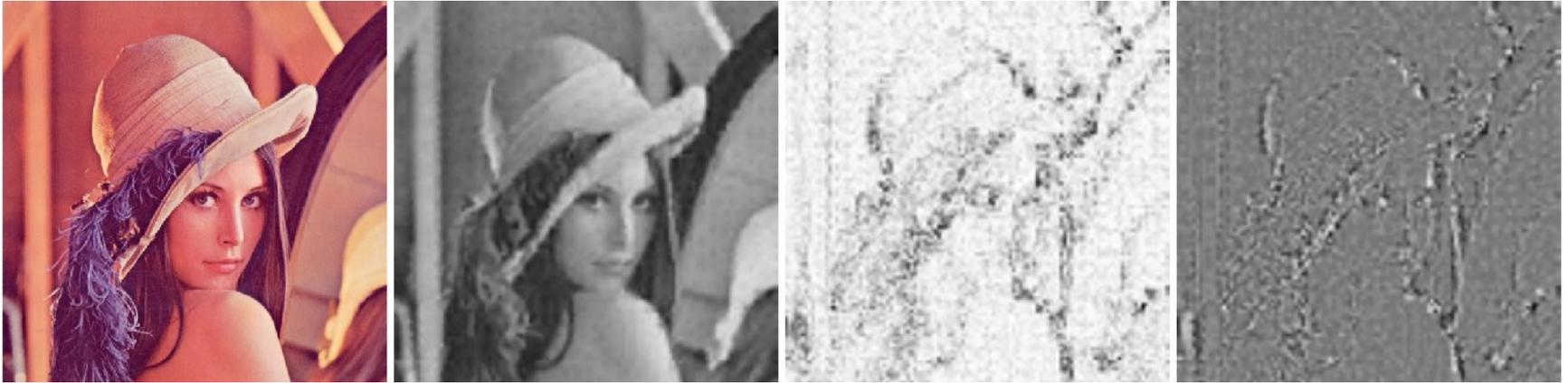


Figura 8.1. (a) imagem Lena original; (b) imagem comprimida e reconstruída usando compressão fractal; (c) imagem de diferença absoluta ampliada e (d) imagem de diferença relativa ampliada

Erro rms= 9,7622

SNR rms 10,4823

PSNR (dB)28,3398

8.1.4 - Critérios de fidelidade subjetivos

Tabela 8.1 – Notas para um critério de avaliação subjetivo usando HDTV.

Valor	Avaliação	Descrição
1	Excelente	Extrema alta qualidade.
2	Muito Bem	Alta qualidade, visualização agradável, a interferência não é desagradável.
3	Passável	Qualidade aceitável. A interferência não é desagradável.
4	Marginal	Qualidade pobre. Você desejaria melhorá-la. A interferência é de alguma forma desagradável.
5	Inferior	Muito pobre, mas ainda assim você poderia vê-la. Desagradável interferência presente.
6	Não Usável	Muito ruim, que não você não conseguiria ver.

8.1.5. Modelos de compressão de imagens

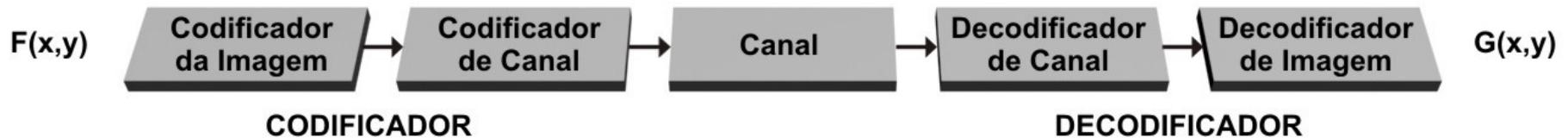


Figura 8.2. Modelo de sistema de compressão genérico de Imagem.



Figura 8.3. Etapas do codificador da imagem original ou fonte



Figura 8.4. Etapas do decodificador do arquivo para imagem

8.2. Métodos de Compressão de Imagem

1. Compressão sem perda ou codificação de redundância
2. Compressão com perda

8.2.1. Compressão sem Perda

- Explora a redundância entre *pixels* na codificação.
- Nenhum dado é perdido durante o processo de compressão.
- Preserva todas as informações que permitirão a reconstrução exata da imagem.
- Exemplos: RLE (*Run Length Encoding*), LZ (*Lempel Ziv*), LZW (*Lempel Ziv Welch*), algoritmo de Huffman (usadas nos formatos: PCX, PNG, GIF, TIFF).

8.2.2. Compressão com Perda

- Há perda de dados durante a compressão da imagem.
- É mais eficiente em relação à área final de armazenamento devido à sua razão de compressão ser maior que a sem perda.
- Em aplicações de sinal de satélite ou dados de imagens médicas, entre outras, muitas vezes não é admissível compressão com perda.
- Diferentes formas de compressão com perda causam visualmente diferentes degradações na imagem.

8.2.3. Porque pode haver perda de dados

- Métodos e algoritmos de compressão eficientes devem levar em conta as características da visão humana.
- Obtem-se um arquivo comprimido de menor dimensão, mantendo, no entanto, a qualidade aceitável em relação ao original, conforme o objetivo que se pretende.
- Os erros e falhas, causados pela compressão com perda de dados, que sejam perceptíveis para os sistemas visual e auditivo humano são conhecidos por artefatos de compressão (*compression artifacts*).

8.2.4. Compressão Simétrica versus Assimétrica

Classificação quanto ao tempo de compressão e descompressão.

Compressão simétrica: Transformadas de *Wavelets* (WT) e Transformadas de Cossenos (DCT, do inglês, *Discrete Cosine Transform*), onde o tempo de compressão é igual ao de descompressão.

Compressão assimétrica: fractal, onde o tempo de compressão é bem maior que o tempo de descompressão.

8.2.5. Compressão por Transformada

- Uma transformação linear inversível é usada para mapear a imagem para um conjunto de coeficientes, que são então quantizados e codificados.
- O estágio de quantização pode eliminar os coeficientes que carregam menos informação.

8.3 Elementos da teoria de informação

8.3.1 - Unidade de informação

8.3.2. Canal de informação

8.3.3. Elementos do canal de informação

8.3.4. Elementos da transmissão

8.3.1 - Unidade de informação

A informação é modelada como um processo probabilístico sendo tratada como um evento aleatório, E .

Sua ocorrência é definida com $p(E)$ que também representa a sua probabilidade.

Em compressão de imagem, E é o tom ou cor que a imagem possui e $p(E)$ o número de *pixels* deste mesmo tom ou cor dividido pelo número total de *pixels* da imagem.

Unidade de informação:

$$I(E) = \log \frac{1}{p(E)} = -\log p(E) \quad (8.10)$$

Em sistemas de computação a unidade de informação é o *bit*, ou seja, a base 2 é a utilizada.

Se o evento for a transmissão de um bit, se a probabilidade de seu valor se 1 ou 0 for a mesma então $p(E)=1/2$, e $I(E) = \log_2 (2) = 1$.

8.3.2. Canal de informação

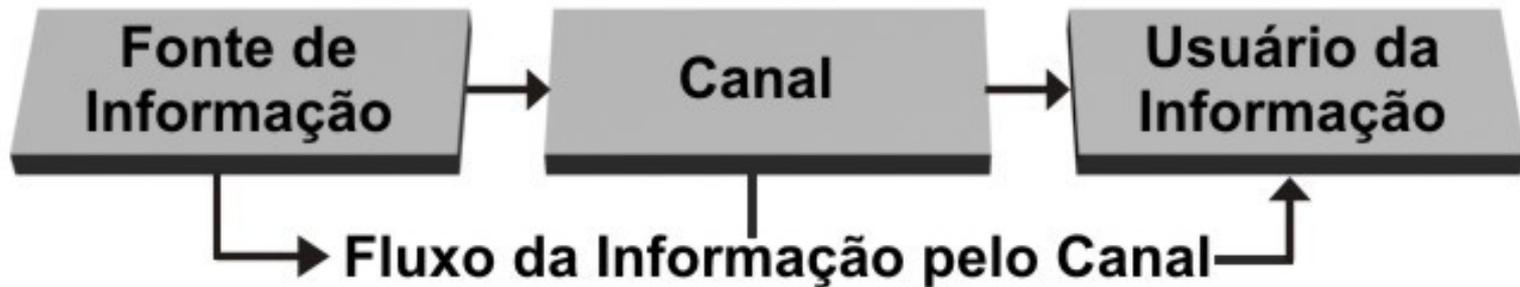


Figura 8.5 – Posição do canal ou Transmissão no processo de codificação.

8.3.3. - Elementos do canal de informação

Responsáveis pela transmissão e recepção da informação.

Os elementos relacionados à transmissão, ou fonte, são: o conjunto de símbolos de entrada possíveis (alfabeto produzível pela fonte de informação) e todas as probabilidades dos elementos pertencentes ao conjunto serem transmitidos pela fonte:

8.3.4 - Elementos da transmissão

Incerteza ou **entropia da fonte** (quantidade média de informação perdida):

$$H(Pa) = - \sum_{i=1}^J p(a_j) \log p(a_j) \quad (8.12)$$

Probabilidade do recebimento de b_j dado que a_j foi transmitido:

$$p(b_k) = \sum_{i=1}^j p(b_k | a_i) p(a_i) \quad (8.13)$$

Matriz do canal ou matriz de transição de canal direito:

$$Q = \begin{bmatrix} p(b_1 | a_1) & p(b_1 | a_2) & \cdots & p(b_1 | a_J) \\ p(b_2 | a_1) & p(b_2 | a_2) & \cdots & p(b_2 | a_J) \\ \vdots & \vdots & \vdots & \vdots \\ p(b_K | a_1) & p(b_K | a_2) & \cdots & p(b_K | a_J) \end{bmatrix} = [q_{kj}] \quad (8.14)$$

Equívoco dos conjuntos P_a com relação a P_b .

$$H(P_a | P_b) = - \sum_{j=1}^J \sum_{k=1}^K p(a_j, b_k) \log p(a_j | b_k) \quad (8.15)$$

Auto-informação mútua:

$$I(P_a, P_b) = \sum_{j=1}^J \sum_{k=1}^K p(a_j, b_k) \log \frac{p(a_j, b_k)}{p(a_j)p(b_k)} \quad (8.16)$$

Capacidade do canal:

$$C = \max_z [I(P_a, P_b)] \quad (8.17)$$

8.4 Entropia da Imagem

Entropia:

$$H(z) = - \sum_{j=1}^J P(a_j) \cdot \log P(a_j) \quad (8.18)$$

Tabela 8.2: Imagem $4 \times 8 = 32$ *pixels* em *grayscale* para efeito de cálculo.

4	4	4	4	64	64	128	128
4	4	4	4	64	64	128	128
4	4	16	16	128	128	128	128
4	4	16	16	128	128	128	128

Tabela 8.3: Probabilidades para cada nível de cinza.

Cor:	Total:	Probabilidade:
4	12	$12 / 32 = 3 / 8$
16	4	$4 / 32 = 1 / 8$
64	4	$4 / 32 = 1 / 8$
128	12	$12 / 32 = 3 / 8$

$$H(z) = -P(4) * \log_2(P(4)) - P(16) * \log_2(P(16)) - P(64) * \log_2(P(64)) - P(128) * \log_2(P(128))$$

$$H(z) = -[3/8 * \log_2(3/8) + 1/8 * \log_2(1/8) + 1/8 * \log_2(1/8) + 3/8 * \log_2(3/8)] = 1.81 \text{ bits/pixel} \quad (8.19)$$

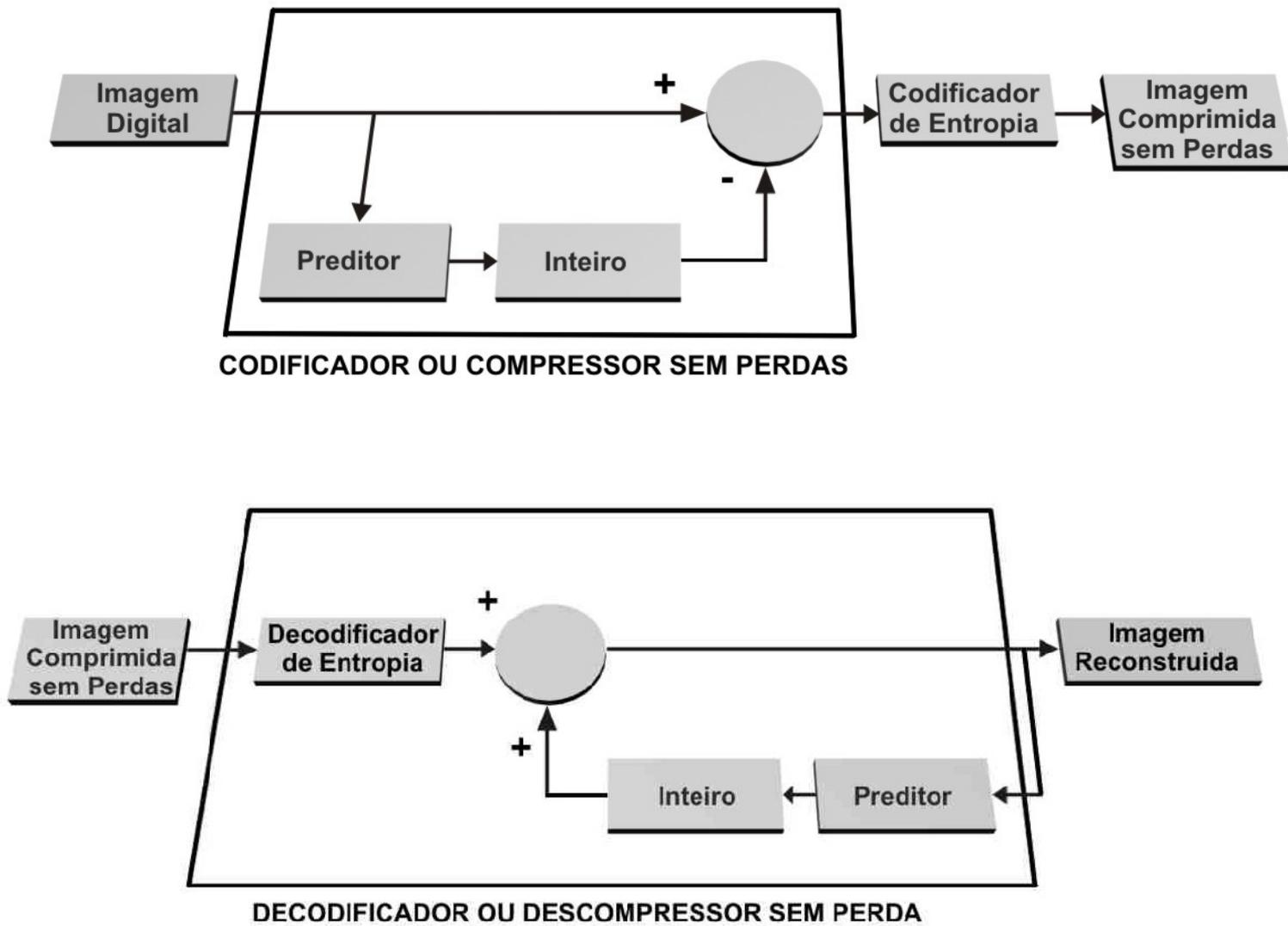


Figura 8.6 - Esquema de codificação e decodificação.

8.4.1. Teoremas fundamentais da codificação

1. Codificação sem ruído.
2. Codificação com ruído.
3. Codificação da fonte.

Também denominados como primeiro e segundo teoremas de Shannon (Shannon, 1948) respectivamente.

8.4.2. Teorema da codificação sem ruído

A entropia do decodificador será a mesma da fonte ou da imagem:

$$H(z') = - \sum_{i=1}^j p(\alpha_i) \log p(\alpha_i) \quad (8.20)$$

Comprimento médio da palavra:

$$\log \frac{1}{p(\alpha_i)} \leq l(\alpha_i) < \log \frac{1}{p(\alpha_i)} + 1 \quad (8.21)$$

Tem-se:

$$\sum_{i=1}^{j^n} p(\alpha_i) \log \frac{1}{p(\alpha_i)} \leq \sum_{i=1}^{j^n} p(\alpha_i) l(\alpha_i) < \sum_{i=1}^{j^n} p(\alpha_i) \log \frac{1}{p(\alpha_i)} \quad (8.22)$$

Primeiro teorema de Shannon para uma fonte de memória zero:

$$H(z) \leq \frac{L'_{avg}}{n} < H(z) + \frac{1}{n} \quad (8.24)$$

A eficiência η de um método de codificação:

$$\eta = n \frac{H(z')}{L'_{avg}} \quad (8.25)$$

8.4.3. Teorema da codificação ruidosa

Segundo teorema de Shannon: considera que o canal de comunicação tem erro e para solucionar este problema trabalha com a repetição da palavra com o objetivo de tornar o erro o menor possível.

Para uma razão de mensagem codificada R menor que a capacidade do canal C com matriz Q , a probabilidade de erro pode ser feita arbitrariamente pequena, a medida que a razão de mensagem codificada for menor que a capacidade do canal.

8.4.4 Teorema da codificação da fonte

Trata o problema quando o processo de codificação é que introduz erro e não o canal de informação.

A função de razão de distorção é:

$$R(D) = \min_{Q_\epsilon, Q_D} \{I(P_a, P_b) = \} \quad (8.28)$$

Pelo **teorema de codificação da fonte** para qualquer ϵ , existe um tamanho de bloco r e $R < R(D) + \epsilon$, tais que a distorção média seja:

$$d(Q) \leq D + \epsilon \quad (8.29)$$

8.5 Métodos de Codificação sem perda

8.5.1. Codificação de Huffman

8.5.2. Codificação por LZW

8.5.3. Codificação por LZ77

**8.5.4. Codificação por Código de Tons Corridos
- RLE**

8.5.1. Codificação de Huffman

A redundância de codificação é eliminada com base numa codificação que produz um código de tamanho variável, atribuindo os códigos de tamanhos menores aos níveis de cinza mais prováveis de ocorrer.

Duas etapas:

2. Cria-se uma série de reduções dos símbolos através da junção dos dois de menores probabilidades a cada iteração.
3. Codificam-se todos os símbolos que foram reduzidos começando com o de maior probabilidade que será associado ao menor código e voltando para os originais.

Exemplo: imagem de tamanho 10x10 e 6 tons de cinza ($a_1 a_2 a_3 a_4 a_5 a_6$), tendo as seguintes probabilidade de ocorrência: $5/8$ de a_1 ; $3/32$ de a_2 e a_3 , $1/32$ de a_6 e a_4 , e $1/8$ de a_5 .

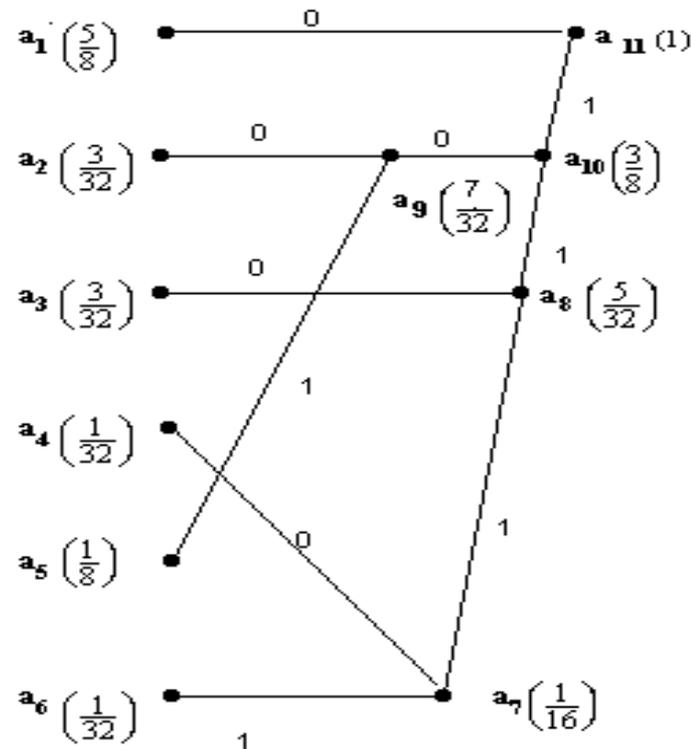


Figura 8.7 Primeira etapa da codificação de Huffman.

Tabela 8.3 - Segunda etapa da codificação de Huffman para as probabilidades das palavras mostradas na figura 8.7.

Informação	Probabilidade	Código
a_1	$5/8=20/32$	0
a_{10}	$3/8=12/32$	1
a_9	$7/32$	10
a_8	$5/32$	11
a_5	$1/8=4/32$	101
a_2	$3/32$	100
a_3	$3/32$	110
a_7	$2/32$	111
a_4	$1/32$	1110
a_6	$1/32$	1111

Para transmitir essa informação obtém uma taxa média de:

$$(5/8)*1 + (3/32)*3 + (3/32)*3 + (4/32)*3 + (1/32)*4 + (1/32)*4 = 1,813 \text{ bits/informação}$$

8.5.2.Codificação por LZW

Faz uso de um dicionário de palavras contendo os símbolos que serão codificados.

Exemplo:

Tabela 8.4 - Imagem a ser codificada.

39	39	126	126
39	39	126	126
39	39	126	126
39	39	126	126

Tabela 8.5 - Dicionário para entrada de símbolos.

Posição do Dicionário (Endereço):	Entrada:
0	0
1	1
...	...
255	255
256	—
...	—
511	—

Tabela 8.6 Preenchendo as entradas do dicionário e codificando.

Seqüência reconhecida:	<i>Pixel</i> processado:	Saída codificada:	Endereço do Dicionário:	Entrada do Dicionário
	39			
39	39	39	256	39-39
39	126	39	257	39-126
126	126	126	258	126-126
126	39	126	259	126-39
39	39			
39-39	126	256	260	39-39-126
126	126			
126-126	39	258	261	126-126-39
39	39			
39-39	126			
39-39-126	126	260	262	39-39-126-126
126	39			
126-39	39	259	263	126-39-39
39	126			
39-126	126	257	264	39-126-126
126		126		

8.5.3. Codificação por LZ77

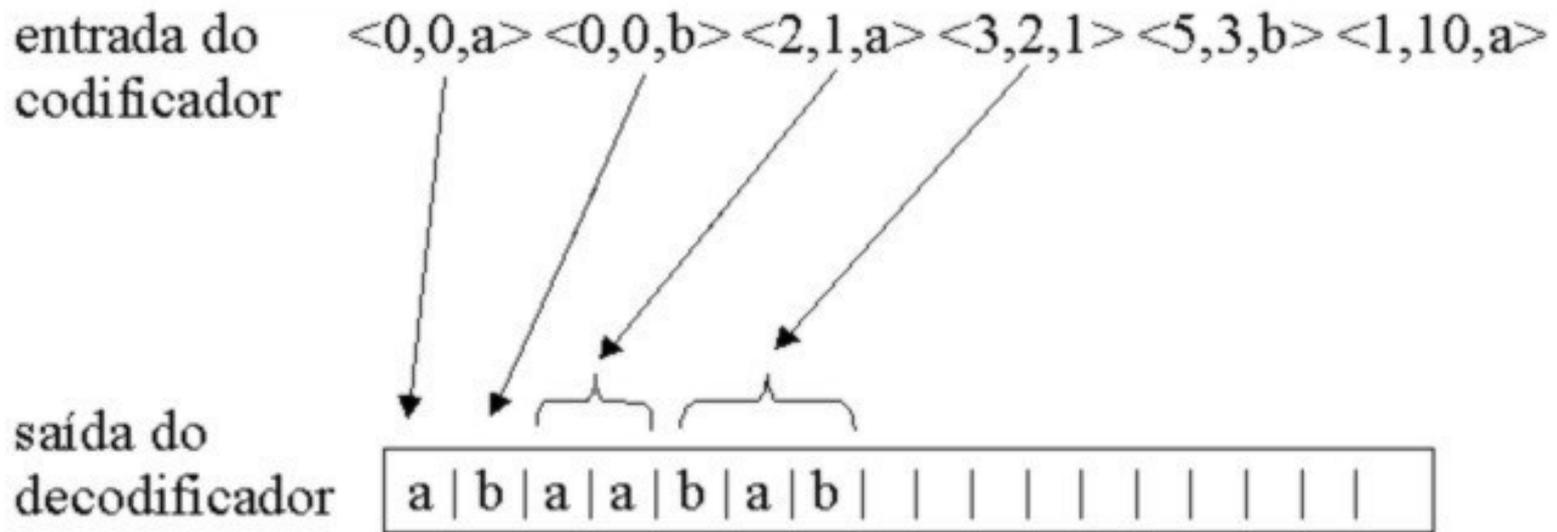


Figura 8.9. Algoritmo LZ77

8.5.4. Codificação por Código de Tons Corridos - RLE

- É a representação de cada linha de uma imagem por seqüências que descrevam trechos de *pixels* contínuos no mesmo tom.
- Na compressão por *pixels* consecutivos iguais (RLE), o número de *pixels* iguais e a cor destes são os valores a serem codificados e armazenados.
- É usada em algoritmos diferentes nos formatos de imagem PCX e BMP nas opções desses que usam tabelas de cores.

8.6. Transformada Discreta do Co-seno (DCT)

Transforma discreta de co-senos em 2-D :

$$T[i, j] = c(i, j) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} I[y, x] \cos \frac{(2y+1)i\pi}{2N} \cos \frac{(2x+1)j\pi}{2N} \quad (8.30)$$

Coeficientes $c(i, j)$:

$$c(i, j) = \sqrt{\frac{2}{N}} \quad , \text{ para } i \text{ e } j \neq 0 \text{ e}$$

$$c(i, j) = \sqrt{\frac{1}{N}} \quad , \text{ para } i \text{ e } j = 0 \quad (8.31)$$

Transformada Inversa IDCT 2-D:

$$I[y, x] = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} c(i, j) T[i, j] \cos \frac{(2y+1)i\pi}{2N} \cos \frac{(2x+1)j\pi}{2N} \quad (8.32)$$

Essa compressão é usada no formato JPEG padrão com valor de N igual a 8.

8.7. Compressão Fractal

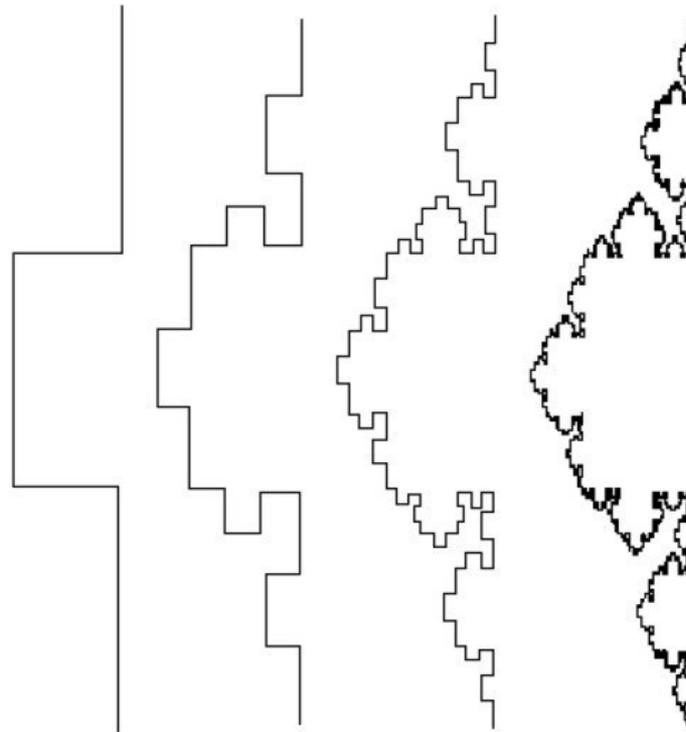


Figura 8.10. Curva de Koch

8.7.1 Comprimindo Imagens com a Geometria Fractal

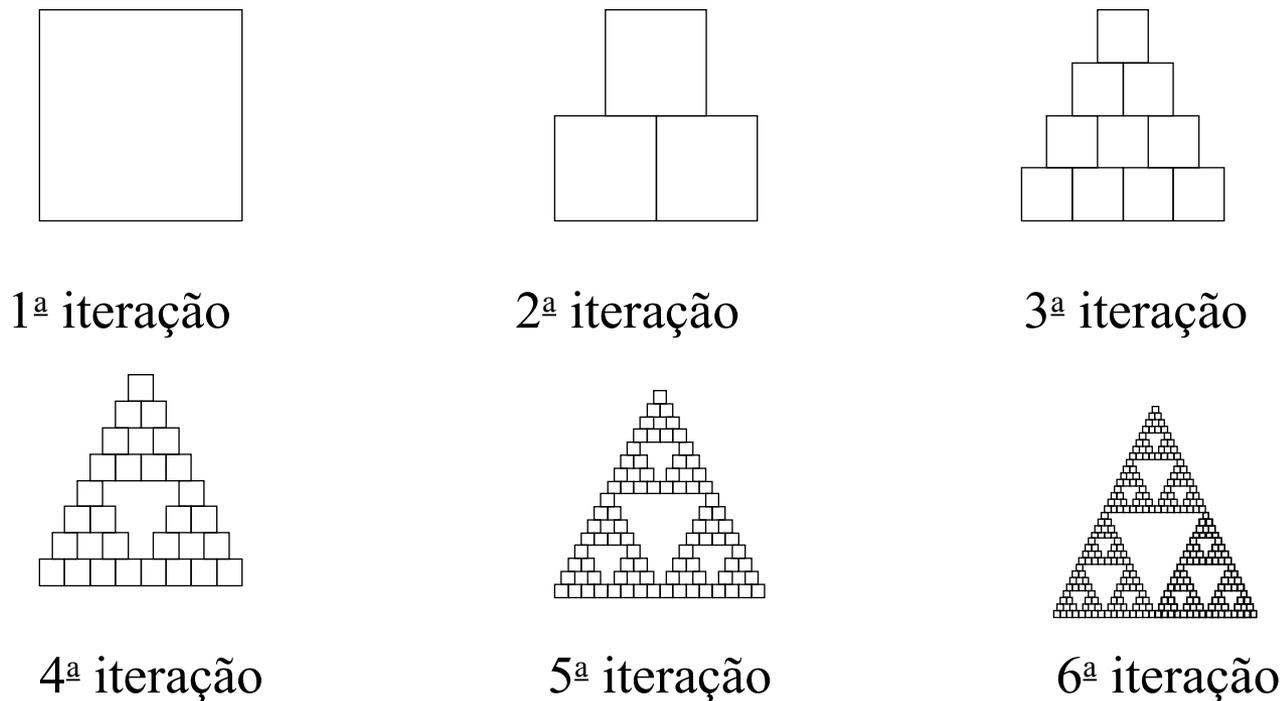


Figura 8.11 - Triângulo de Sierpinski

DF do Triângulo de Sierpinski:

$$DF = \log 3 / \log 2 = 1,5849625007211561814537389439478\dots \quad (8.39)$$

8.7.2- Teorema do Mapeamento de Contração e Teorema da Colagem

Partindo de qualquer conjunto inicial (imagem inicial) será encontrado o mesmo atrator para um determinado Sistema de Funções Iterativas (SFI).

Pelo Teorema da Colagem o código IFS de uma imagem pode ser obtido cobrindo-se essa imagem com cópias reduzidas dela mesma, e determinando as transformações que levam a imagem original em suas cópias.

8.7.3 – Determinando o SFI de imagens automaticamente

Sistemas de Funções de Iteração Particionados SFIP que buscam a auto-similaridade entre partes maiores e partes menores da imagem.

- As imagens são vistas como uma colagem de partes auto similares que podem ser mapeadas entre si.
- Como partes menores tomam-se blocos quadrados de $n \times n$ pixels, chamados de blocos imagem ou molde, e como partes maiores blocos com o dobro das dimensões do menor ($2n \times 2n$), chamados de blocos domínio.

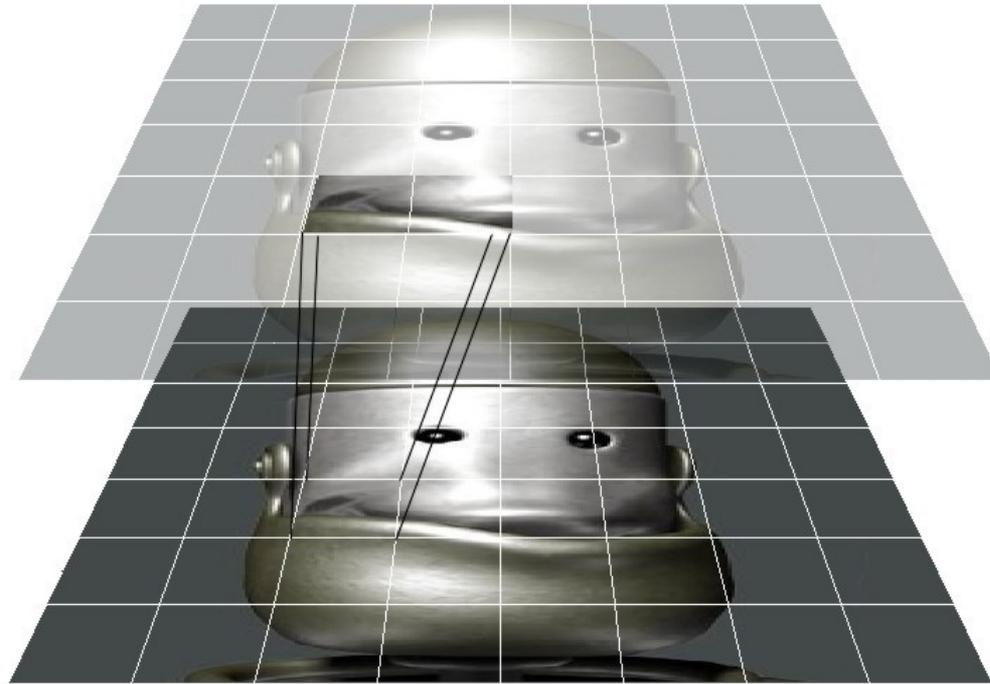


Figura 8.12 - Formação de um par de blocos domínio-molde ótimo.

A imagem deve ser entendida como um objeto tridimensional, o valor da tonalidade de um pixel (z) é tratado como sendo a terceira dimensão espacial. Os blocos tornam-se cubos.

8.7.4 Considerações sobre a simetria do bloco domínio

Para cada bloco domínio tem-se oito simetrias que serão testadas em relação ao bloco molde corrente.

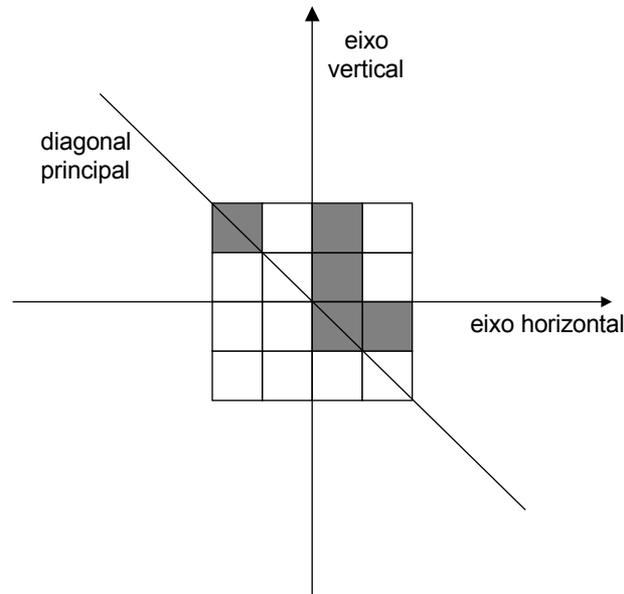
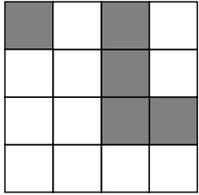
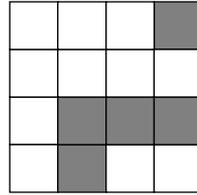


Figura 8.13 - Eixos de Reflexão considerados nas operações de simetria.



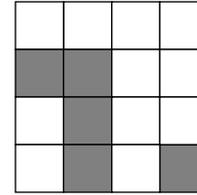
Identidade

$(0,0,0)$



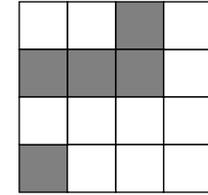
Rotacionada de 90°

$(1,1,0)$



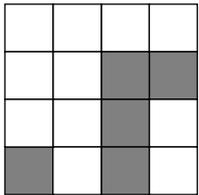
Rotacionada de 180°

$(0,1,1)$



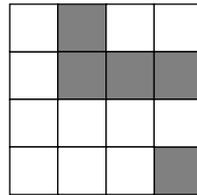
Rotacionada de 270°

$(1,0,1)$



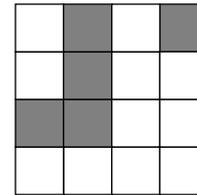
Ref. horizontal
imagem identidade

$(0,0,1)$



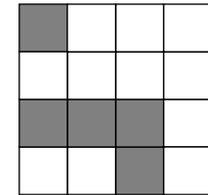
Ref. horizontal
da imagem rotacionada
de 90°

$(1,1,1)$



Ref. Vertical da
imagem identidade

$(0,1,0)$



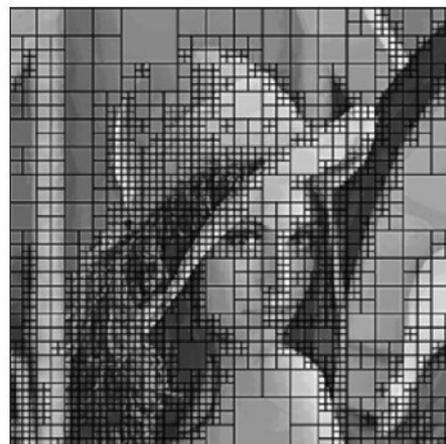
Ref. na diagonal da
principal $(i,j = j,i)$

$(1,0,0)$

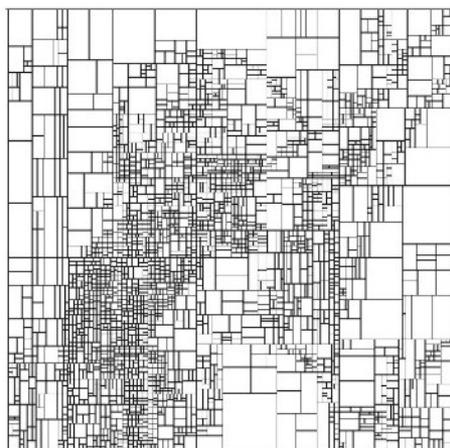
Figura 8.14 - Bloco molde e suas 8 possíveis simetrias.



A- ORIGINAL



B- QUADTREE



C- HV



D- TRIÂNGULO

Figura 8.15 - Diferentes formas de *particionamento* de uma imagem.

8.7.5 Etapas da Compressão Fractal Automática

Primeira etapa: Definição dos blocos molde

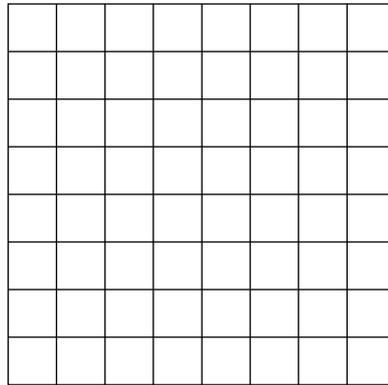


Figura 8.16 - Malha de blocos molde de uma imagem.

Segunda etapa: Definição dos blocos domínio

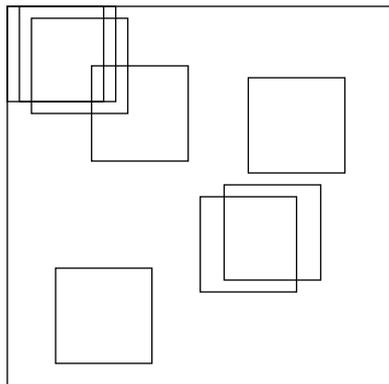


Figura 8.17 - Alguns blocos domínio da figura anterior.

Terceira etapa: Redução dos blocos domínio

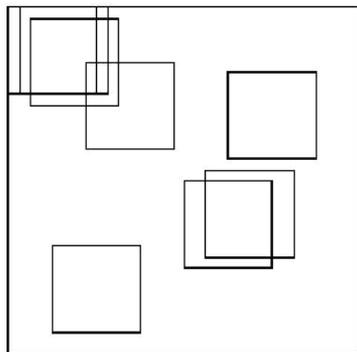


Figura 8.18 - Imagem reduzida por um fator igual a $\frac{1}{2}$, de onde sairão os blocos domínio reduzidos.

Quarta etapa: Busca do melhor par bloco domínio reduzido- molde

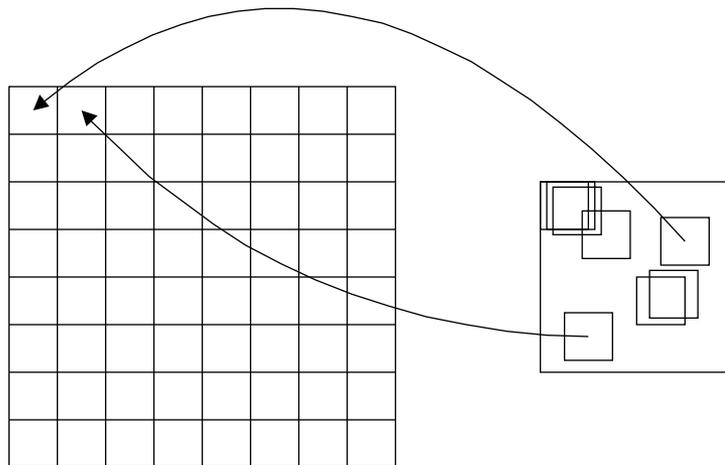


Figura 8.19 - Formação dos pares ótimos entre os blocos domínio e molde.

Quinta etapa:

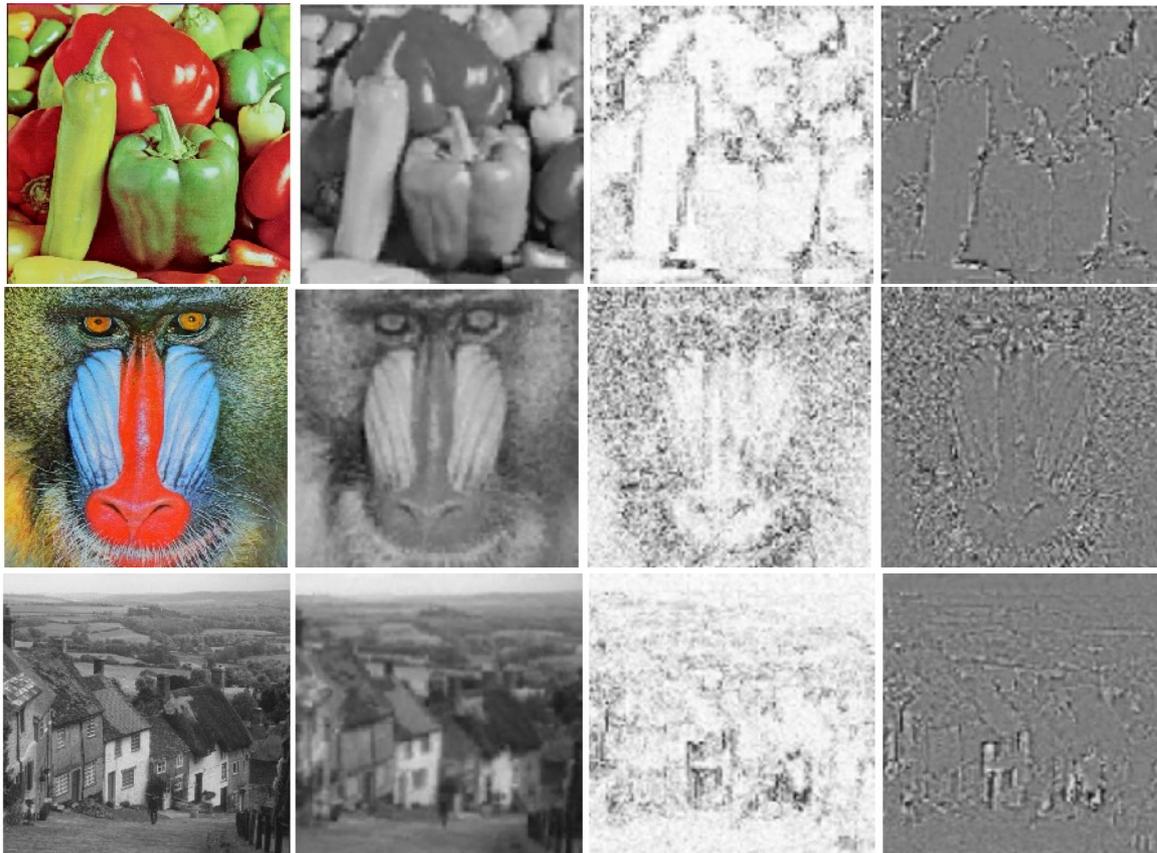


Figura 8.20 - Resultados obtidos da compressão Fractal usando a técnica de Sub Busca Local com $\Delta = 8$ e blocos molde de 4×4 *pixels*.



Figura 8.20 - Resultados obtidos da compressão Fractal usando a técnica de Sub Busca Local com $\Delta = 8$ e blocos molde de 4×4 *pixels*.

Tabela 8.7 - Resultados obtidos com o método de Sub Busca Local com $\Delta = 8$ e blocos molde de 4x4 *pixels* para as imagens da figura 8.1 e 8.20.

Nome da imagem	e_{rms}	SNR $_{rms}$	PSNR (dB)
<i>Lena</i>	9, 9057	10, 3072	28, 431
<i>Peppers</i>	10, 6933	8, 5312	27, 5485
<i>Mandrill</i>	15, 4048	7, 6832	24, 3777
<i>Goldhill</i>	9, 2310	10, 0295	28, 8259
<i>Woman1</i>	13, 5039	8, 5180	25, 546
<i>Milk</i>	12, 0226	6, 7880	26, 5309

8.8. Compressão por *Wavelets*

8.8.1. Perspectiva Histórica

8.8.2. Análise de *Wavelet*

8.8.3. Transformada de *Wavelet* Contínua

8.8.4. Transformada de *Wavelet* Discreta

8.8.5. Semelhanças entre Transformada de Fourier e *Wavelet*

8.8.6. Diferenças entre Transformada de Fourier e Transformada de *Wavelet*

8.8.7. *Wavelets* Unidimensional

8.8.8. *Wavelet* Bidimensional

8.8.9. Aproximações e Detalhes

8.8.10. Banco de Filtros

8.8.1. Perspectiva Histórica

Wavelets são conjuntos de funções matemáticas que são usadas para representar dados ou outras funções.

1. A demonstração de Fourier (1807)
2. A primeira menção a *wavelet* - Alfred Haar (1909)
3. Guido Weiss e Ronald R. Coifletfman
4. Grossman e Morlet
5. S. Mallat
6. Yves Meyer
7. Ingrid Daubechies

Qualquer função, com período 2π , pode ser reescrita como a soma dos termos da Série de Fourier:

$$a_0 + \sum_{n=1}^{\infty} (a_n \cos nt + b_n \operatorname{sen} nt) \quad (8.46)$$

Os coeficientes são calculados por:

$$a_0 = \frac{2}{T} \int_0^{2\pi} f(t) dt \quad (8.47)$$

$$a_n = \frac{2}{T} \int_0^{2\pi} f(t) \cos(nt) dt \quad (8.48)$$

$$b_n = \frac{2}{T} \int_0^{2\pi} f(t) \operatorname{sen}(nt) dt \quad (8.49)$$

8.8.2. Análise de *Wavelet*

A Análise de *Wavelet* é uma ferramenta matemática para decomposição em nível hierárquico em um conjunto de aproximações e detalhes.

O nível hierárquico corresponde à escala diática (formado por potência de 2).

Permite a descrição de uma função em termos globais, mais termos que variam de detalhes globais até detalhes finos.

A função em questão pode ser uma imagem, uma curva ou uma superfície.

8.8.3. Transformada de *Wavelet* Contínua

Transformada de *Wavelet* Contínua é a soma ao longo do tempo de um sinal multiplicado por uma escala, e deslocado por uma função *wavelet* (Ψ), também chamada *wavelet* mãe:

$$C(\textit{escala}, \textit{posição}) = \int_{-\infty}^{\infty} f(t) \Psi(\textit{escala}, \textit{posição}, t) dt \quad (8.50)$$

A Transformada de *Wavelets* contínua em $F(a,b)$ é:

$$F(a,b) = \int f(t) \Psi_{a,b}(t) dt \quad (8.51)$$

A função $\Psi_{a,b}(t)$ é denominada *wavelet*:

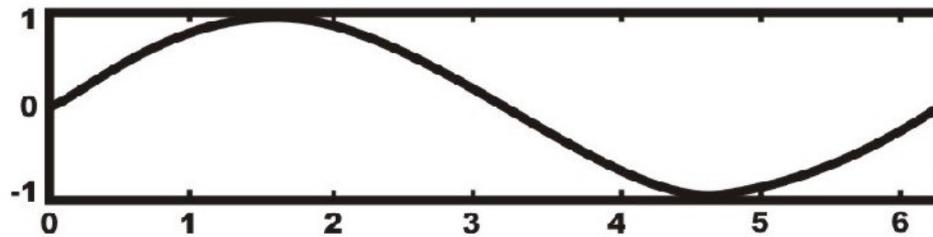
$$\Psi_{a,b}(t) = \frac{1}{\sqrt{a}} \Psi\left(\frac{t-b}{a}\right), \quad a \neq 0, \quad b \in \mathbb{R} \quad (8.52)$$

As funções *wavelets* são derivadas segundo os critérios:

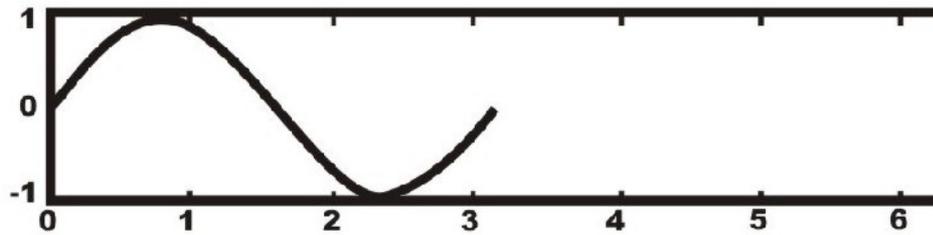
$$C_{\Psi} = 2\pi \int |u|^{-1} |\hat{\Psi}(u)|^2 du < \infty \quad (8.53)$$

8.8.3.1. Parâmetro de Escala

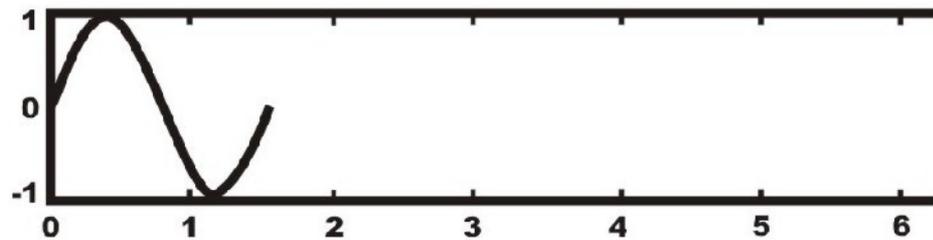
O fator escala a representa uma contração ou dilatação no sinal. Para $a > 1$ a função sofre uma dilatação, para $a < 1$ obtem-se uma contração do sinal.



$$f(t) = \sin ; a = 1$$

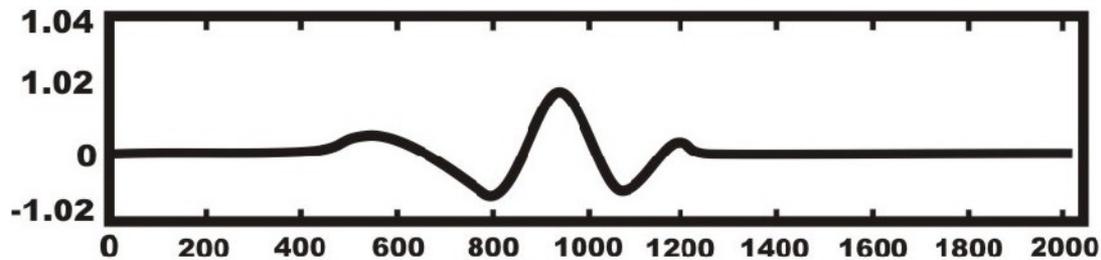


$$f(t) = \sin(2t) ; a = \frac{1}{2}$$

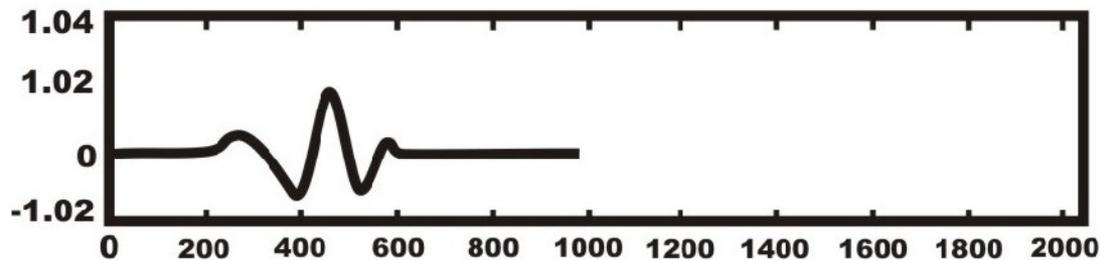


$$f(t) = \sin(4t) ; a = \frac{1}{4}$$

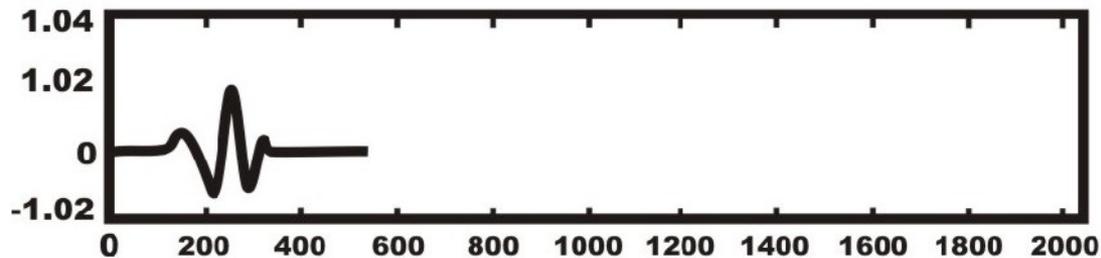
Figura 8.21 - Fator de escala de uma função



$$f(t) = \Psi(t) ; a = 1$$



$$f(t) = \Psi(2t) ; a = \frac{1}{2}$$



$$f(t) = \Psi(4t) ; a = \frac{1}{4}$$

Figura 8.22 - Fator de escala de uma função *wavelet*

8.8.3.2. Parâmetro de Posição ou Deslocamento



Figura 8.23. - Fator de deslocamento, à direita função $\Psi(t)$ *wavelet*, à esquerda função $\Psi(t-b)$ *wavelet* deslocada

8.8.3.3. Cálculo da Transformada de *Wavelet* Contínua

(1) Deve-se escolher a *wavelet* e fazer a comparação em uma parte inicial do sinal original.

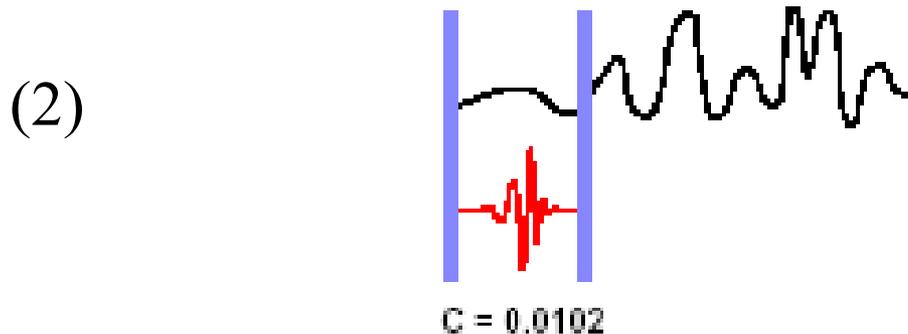


Figura 8.24 – Comparação do sinal original com a *wavelet*

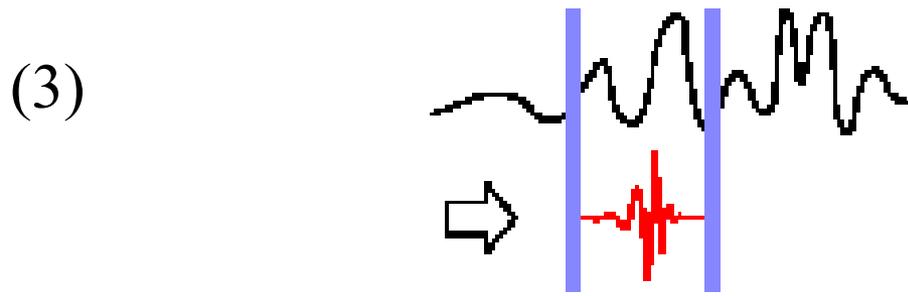


Figura 8.25 – Desloca-se a *wavelet* para a direita para calcular novo C

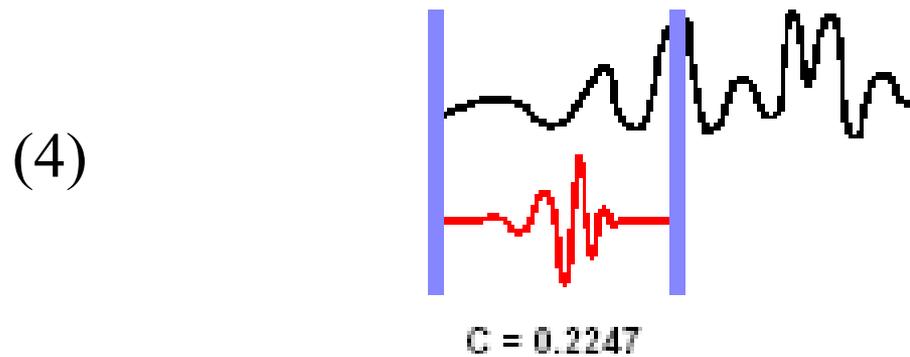


Figura 8.26 – Dilata-se a *wavelet* e repetem-se os passos (1) e (3)

(5)

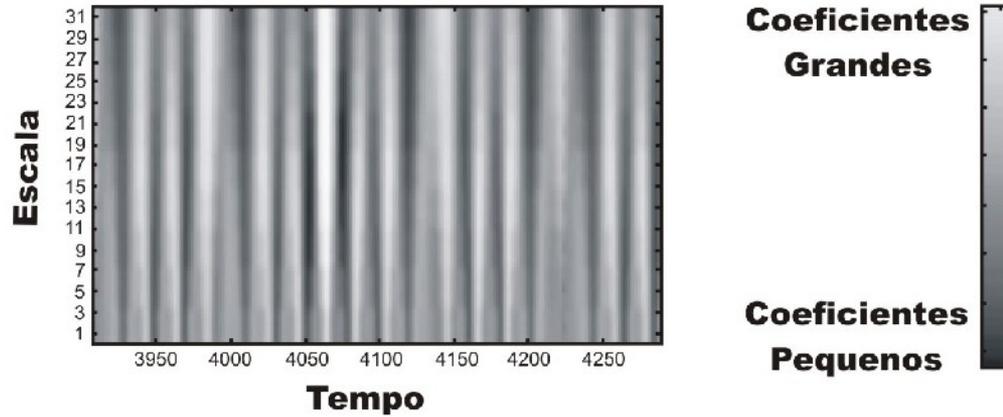


Figura 8.27 – Repetem-se os passos de (1) até (4) para todas as escalas.

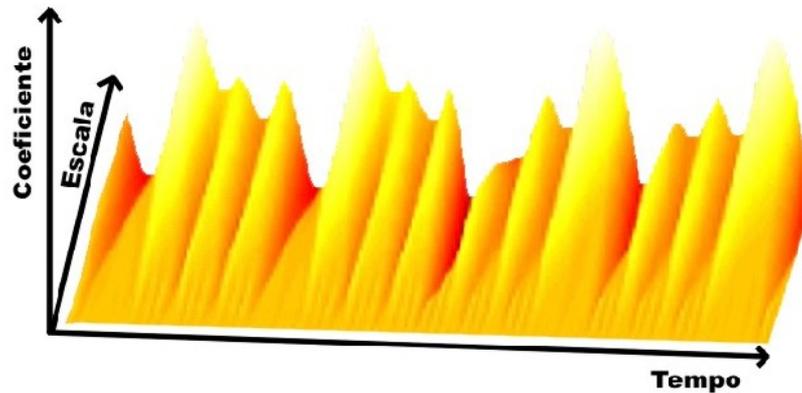


Figura 8.28 – Representação 3D da Transformada de *Wavelet*.

8.8.4. Transformada de *Wavelet* Discreta

$$\Psi_{a,b}(t) = \frac{1}{\sqrt{a}} \Psi\left(\frac{t-b}{a}\right), \quad a = 2^j, b = k 2^j, \quad (j,k) \in \mathbb{Z}^2 \quad (8.54)$$

8.8.5. Semelhanças entre Transformada de Fourier e *Wavelet*

- São ambas operações lineares que geram uma estrutura de dados que contém segmentos de vários comprimentos.
- As transformadas matriciais inversas da FFT e da DWT são as transpostas das originais.
- As funções base são localizadas no domínio da frequência tornando-as ferramentas matemáticas poderosas na análise espectral de potência.

8.8.6. Diferenças entre Transformada de Fourier e Transformada de *Wavelet*

- As funções individuais *wavelet* são *localizadas no espaço*. Já as funções de Fourier, seno e co-seno não são.
- Esta característica junto com a localização em frequência das *wavelets*, fazem muitos operadores e funções usarem *wavelets esparsas* quando transformados para o domínio de *wavelet*.

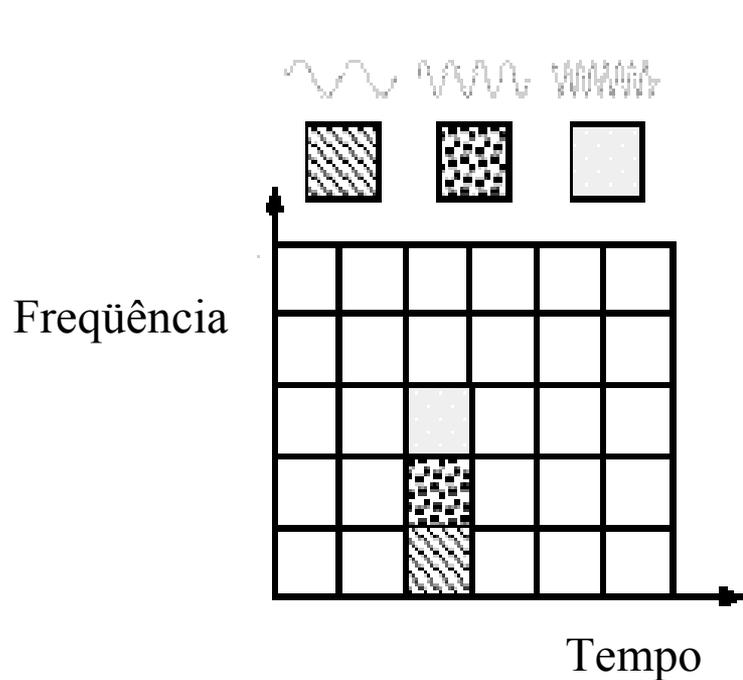


Figura 8.29 - Funções base de Fourier descritas no plano Tempo x Frequência.

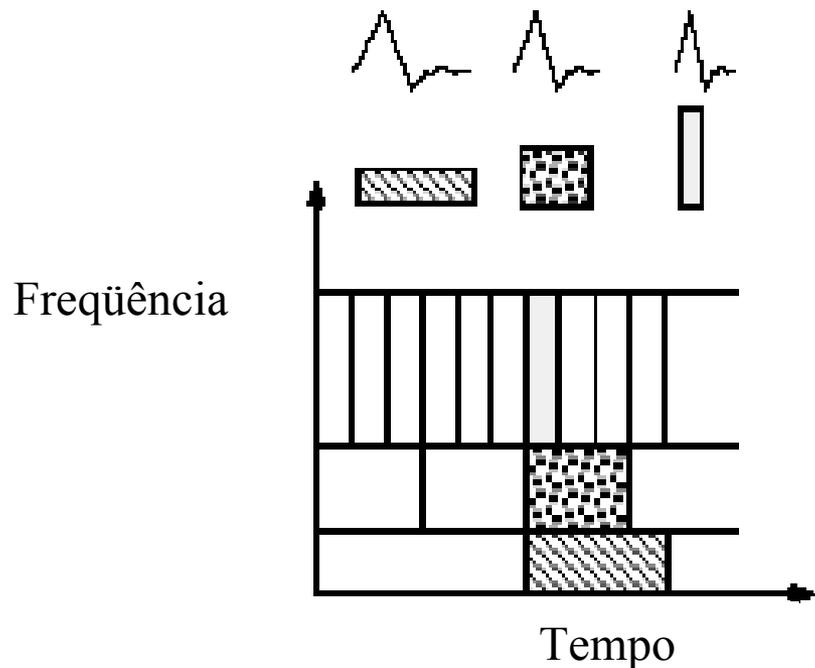


Figura 8.30 - Função base *wavelet* de Daubechies descritas no plano Tempo x Frequência.

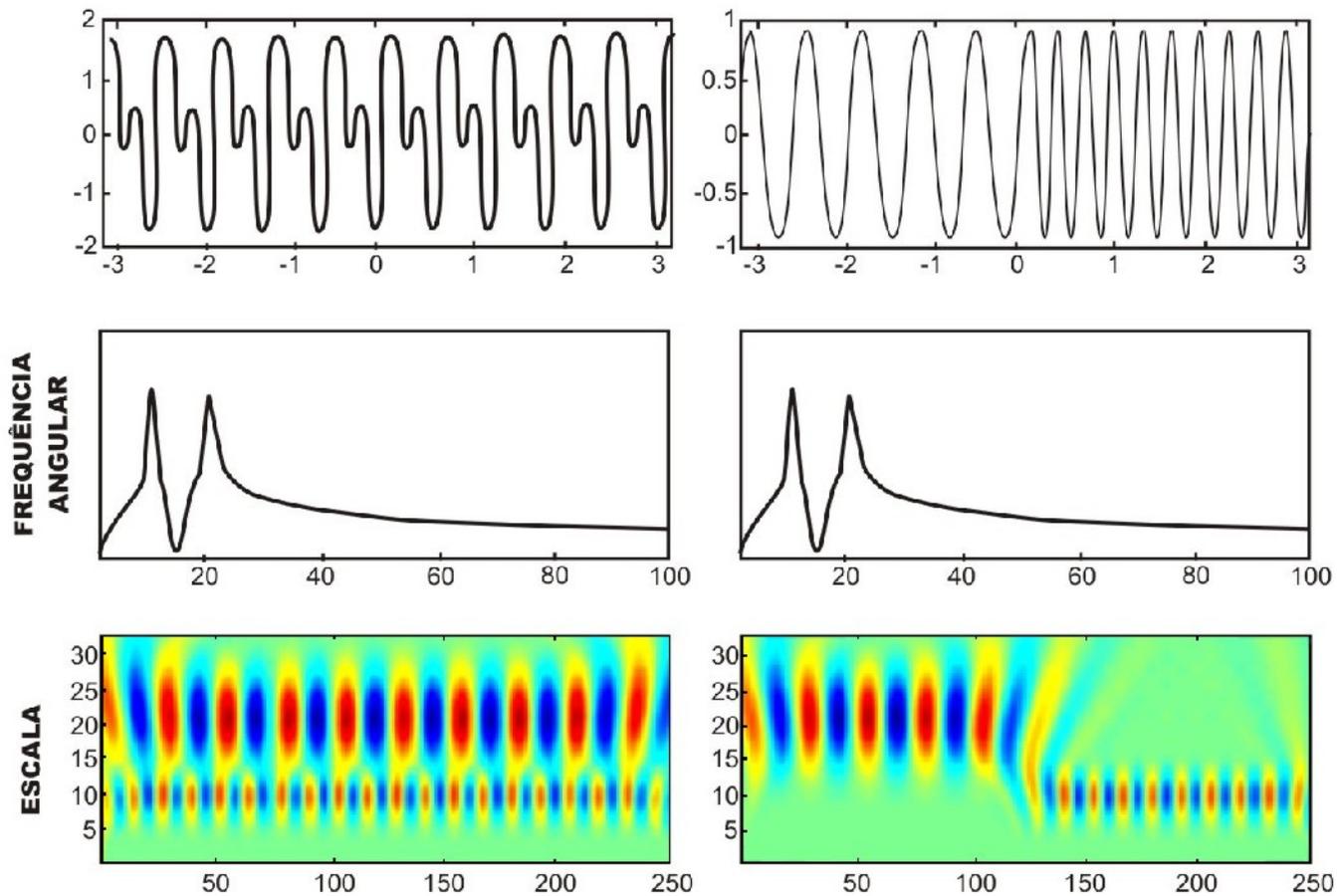


Figura 8.31 – Comparação entre Transformada de Fourier e Transformada de *Wavelet*.

8.8.7. *Wavelets*

Unidimensional

8.8.7.1. Transformada *Wavelet* de Haar Unidimensional

8.8.7.2. Funções bases de *Wavelet* de Haar Unidimensional

8.8.7.3. Ortogonalidade

8.8.7.4. Normalidade

8.8.7.5. Compressão

8.8.7.1. Transformada *Wavelet* de Haar Unidimensional

Exemplo:

- 1- Suponha uma seqüência de uma dimensão com uma resolução de quatro *pixels*, tendo valores: [9 7 3 5].
- 2- Calcule primeiro a média dos valores em pares, obtendo os novos valores em resolução da imagem:[8 4].
- 3- Armazene alguns *coeficientes de detalhes*, que capturam a informação perdida. No exemplo: 1.
- 4- Repetindo este processo recursivamente até a decomposição tem-se a tabela 8.8.

Tabela 8.8 - Decomposição em coeficientes de aproximação e detalhes.

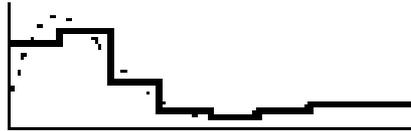
Resolução	Média / Valores	Coefficientes de Detalhes
4	[9 7 3 5]	
2	[8 4]	[1 -1]
1	[6]	[2]

Para a base de Haar unidimensional, a transformada de *wavelet* da imagem original de quatro *pixels* é dada por: [6 2 1 -1]

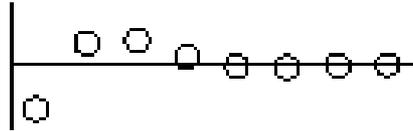
Figura 8.32 - Seqüência de aproximação e coeficientes de detalhes.



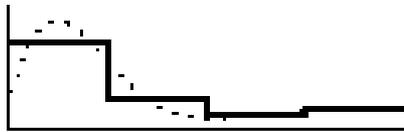
Aproximação V^4



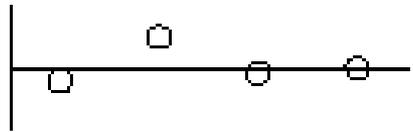
Aproximação V^3



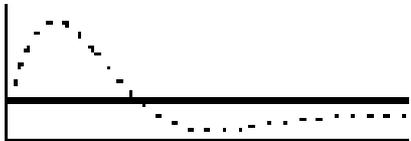
Coeficientes de detalhes W^3



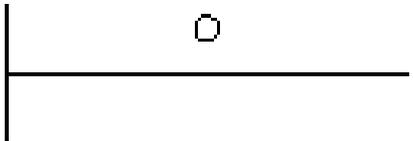
Aproximação V^2



Coeficientes de detalhes W^2



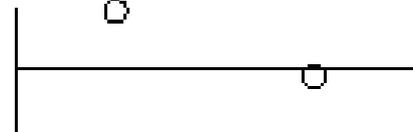
Aproximação V^1



Coeficientes de detalhes W^1



Aproximação V^0



Coeficientes de detalhes W^0

8.8.7.2. Funções bases de *Wavelet* de Haar Unidimensional

Wavelets são coleções de funções $\psi_i^j(x)$ linearmente independente que geram o espaço W^j .

Estas funções de base têm as seguintes propriedades:

1. As *wavelets* ψ_i^j bases de W^j , juntamente com as funções base ϕ_i^j de V^j e ϕ_i^{j+1} de V^{j+1} formam a base para W^j .
2. Toda a função base ψ_i^j de W^j são ortogonais a todas as bases ϕ_i^k de V^k sob um certo produto interno escolhido.

As *wavelets* que correspondem à base quadrada são conhecidas como *wavelets* de Haar e são dadas por:

$$\psi_i^j(x) := \psi(2^j x - i), \quad i = 0, \dots, 2^j - 1 \quad (8.57)$$

Onde:

$$\psi(x) := \begin{cases} 1 & \text{se } 0 \leq x < \frac{1}{2} \\ -1 & \text{se } \frac{1}{2} \leq x < 1 \\ 0 & \text{se } x < 0 \text{ ou } x \geq 1 \end{cases}$$

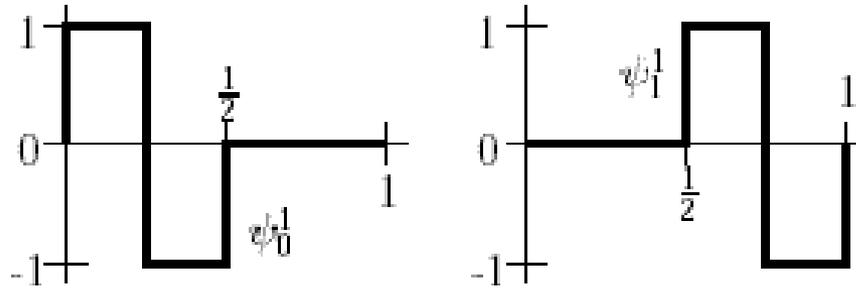


Figura 8.34 - As *wavelets* de Haar para W^1 .

Expressando a imagem original unidimensional (ex. do item 8.8.7.1) como uma combinação linear das funções de base quadrada em V^2 ; pode-se escrever:

$$I(x) = c_0^2 \phi_0^2(x) + c_1^2 \phi_1^2(x) + c_2^2 \phi_2^2(x) + c_3^2 \phi_3^2(x)$$



Figura 8.35 – Descrevendo a imagem por V^2

É possível reescrever a expressão para $I(x)$ em termos das funções base em V^1 e W^1 e, usando coeficientes de média:

$$I(x) = c_0^1 \phi_0^1(x) + c_1^1 \phi_1^1(x) \quad (8.58)$$

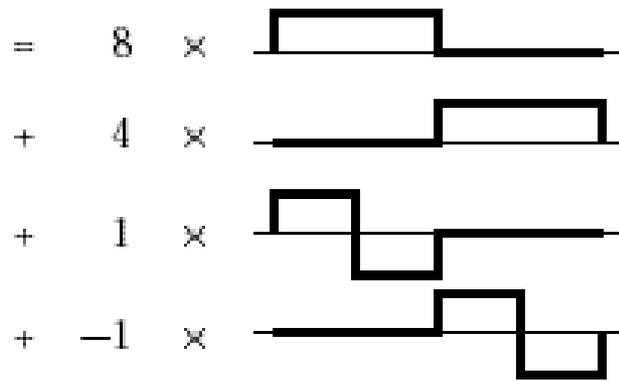


Figura 8.36 – Descrevendo a imagem por média e detalhes

Pode-se reescrever $I(x)$ como uma soma de funções de base em V^0 , W^0 e W^1 :

$$I(x) = c_0^0 \phi_0^0(x) + d_0^0 \psi_0^0(x) + d_0^1 \psi_0^1(x) + d_1^1 \psi_1^1(x) \quad (8.59)$$

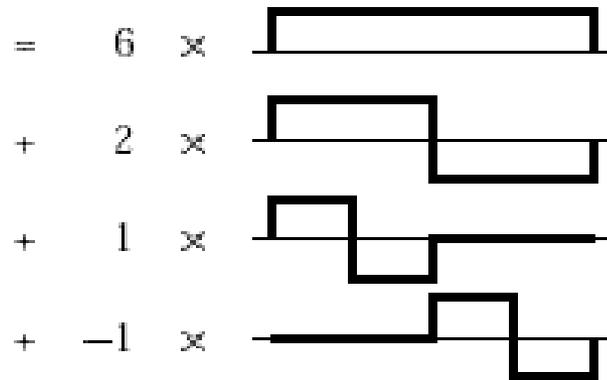


Figura 8.37 – Descrevendo a imagem como média total e detalhes

8.8.7.3. Ortogonalidade

- Uma base ortogonal é aquela que todas as funções base, isto é $\phi_0^0, \psi_0^0, \psi_0^1, \psi_1^1, \dots$ são ortogonais entre si.
- A base de Haar possui propriedade de ortogonalidade.

8.8.7.4. Normalidade

Uma função base $u(x)$ é normalizada se $\langle u | u \rangle = 1$

Pode-se normalizar a base de Haar substituindo as equações por:

$$\phi_i^j(x) := 2^{j/2} \phi(2^j x - i), \quad i = 0, \dots, 2^j - 1 \quad (8.60)$$

$$\psi_i^j(x) := 2^{j/2} \psi(2^j x - i), \quad i = 0, \dots, 2^j - 1 \quad (8.61)$$

No exemplo, os coeficientes não normalizados [6 2 1 -1] se tornam: $\left[6 \quad 2 \quad \frac{1}{\sqrt{2}} \quad \frac{-1}{\sqrt{2}} \right]$

8.8.7.5. Compressão

O objetivo da compressão é expressar um conjunto inicial de dados usando outro conjunto menor, com ou sem perda de informação.

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

$$f(x) = \sum_{i=1}^m c_i u_i(x) \quad (8.62)$$

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

$$\tilde{f}(x) = \sum_{i=1}^{\tilde{m}} \tilde{c}_i \tilde{u}_i(x) \cong f(x) \quad (8.63)$$

8.8.8. *Wavelet* Bidimensional

Para entender a compressão de imagem, descrevem-se as funções escalar e de *wavelets* que formam as bases de *wavelet* bidimensionais.

8.8.8.1. Transformada de *Wavelet* de Haar bidimensional

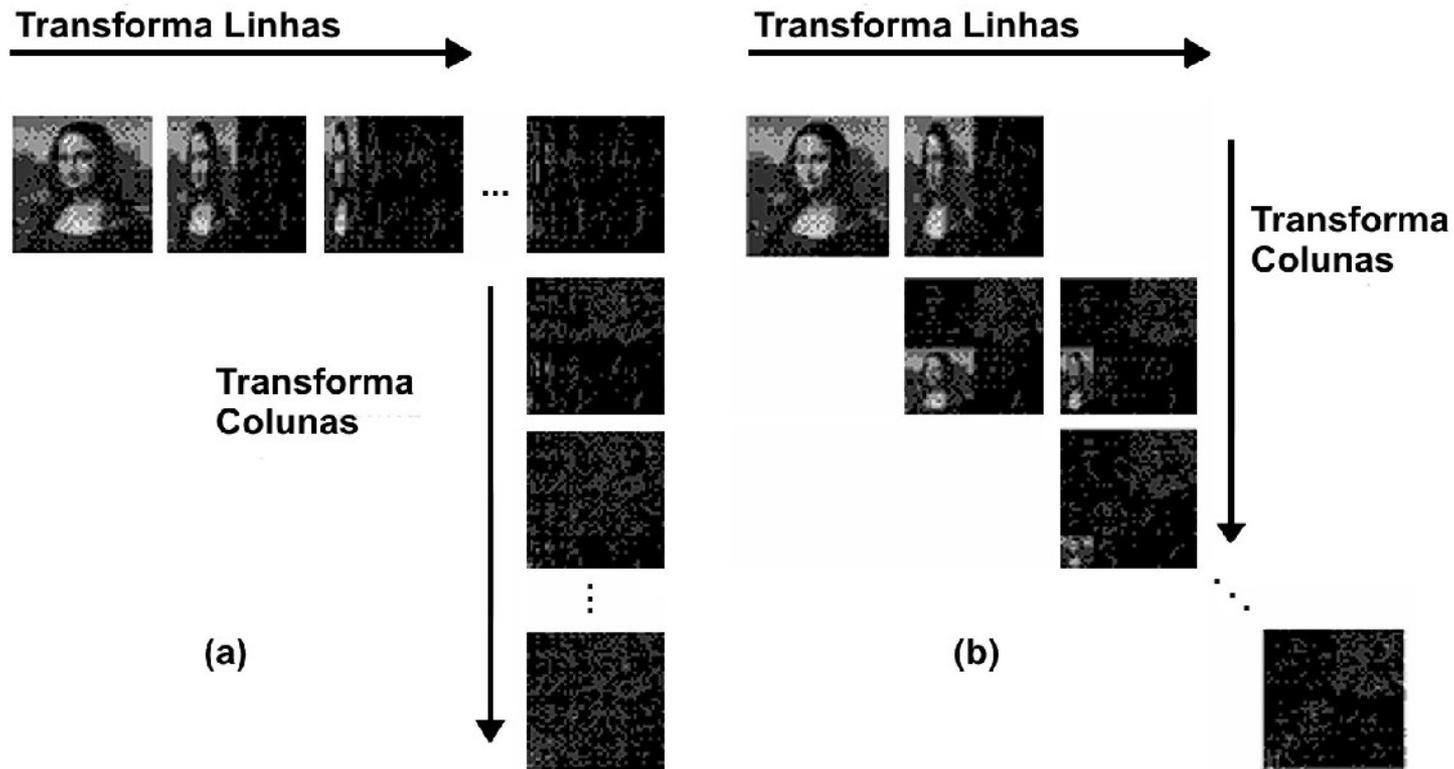


Figura 8.38 – (a) Decomposição padrão, (b) Decomposição não padrão.

8.8.9. Aproximações e Detalhes

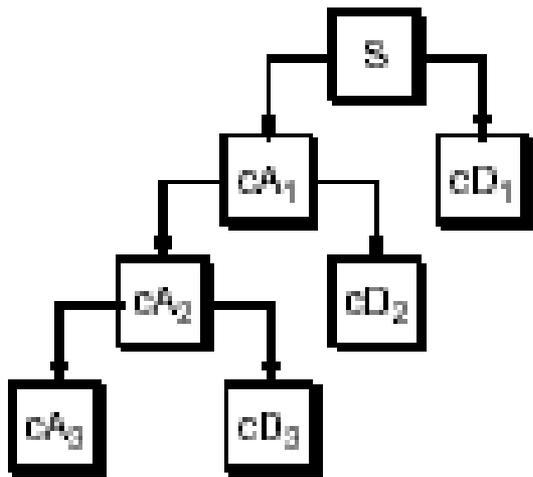


Figura 8.39 – Árvore de Decomposição *Wavelet*

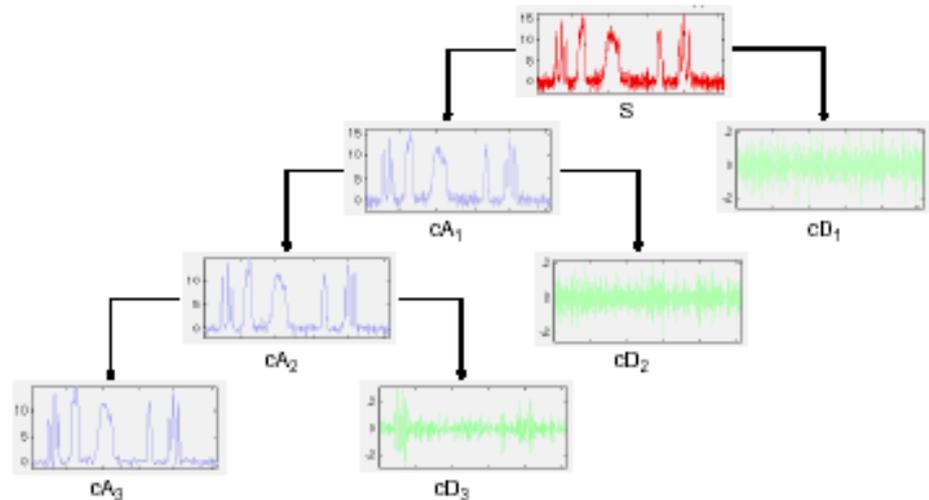


Figura 8.40 – Árvore de Decomposição *Wavelet* de um sinal

8.8.10. Banco de Filtros

A forma de implementação eficiente do algoritmo de compressão por *wavelet* é por intermédio de um Banco de Filtros em Quadratura Conjugado.

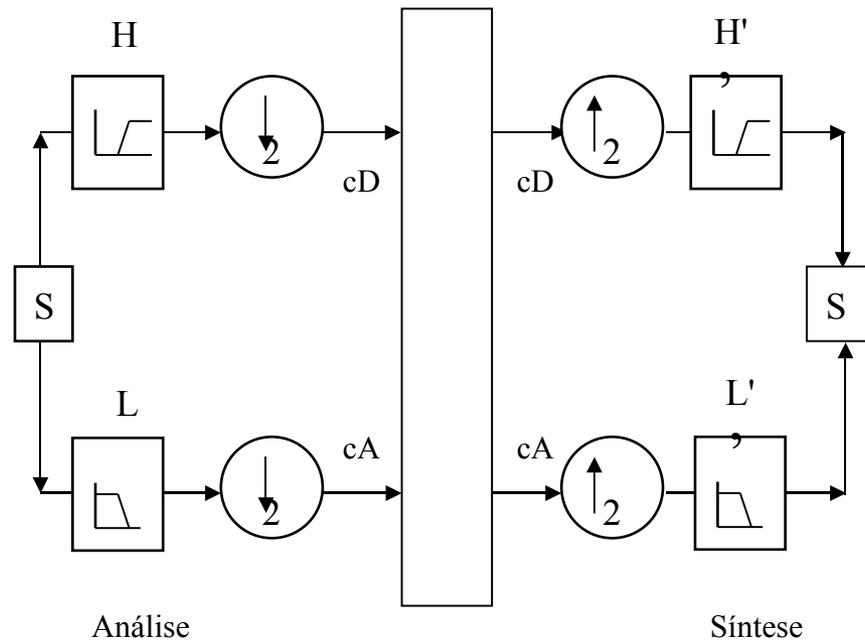


Figura 8.41 - Banco de Filtros

8.9. Padrões de Compressão de Imagem

8.9.1. GIF

8.9.2. PNG

8.9.3. JPEG

8.9.4. JPEG2000

8.9.5. MJPEG

8.9.6. BMP

8.9.7. Formato PCX

8.9. Padrões de Compressão de Imagem

Tabela 8.9 – Comparação entre alguns formatos de arquivos de imagens

Formato	Sistema de Cor	Compressão
GIF	RGB com tabela de até 256 cores	LZW
TIFF	RGB*, CMYK, YCbCr, Lab, Luv	RLE, LZW, JPEG, JBIG, Huffman ou nenhuma e outros
JPEG	RGB, YCbCr, CMYK	DCT , Huffman
PCX	RGB*	RLE
BMP	A BGR*	RLE ou nenhuma
TGA	RGB*	RLE ou nenhuma
PNG, MNG	RGBA (alfa em 256 tons)	LZ77 + Huffman = deflate
JPEG2000	RGB, YCbCr,	DWT (<i>wavelets</i>) ou nenhuma

8.9.1. GIF

- GIF (Graphics Interchange Format) - usa o algoritmo LZW.
- Pode armazenar mais de uma imagem no arquivo (animações).
- Armazena apenas imagens em cinza ou RGB com tabelas de 256 cores ou menos.
- **Assinatura ‘GIF’**: A assinatura é ‘GIF’ + ano/versão.

8.9.1.1. Formato Geral de Arquivos 'GIF'

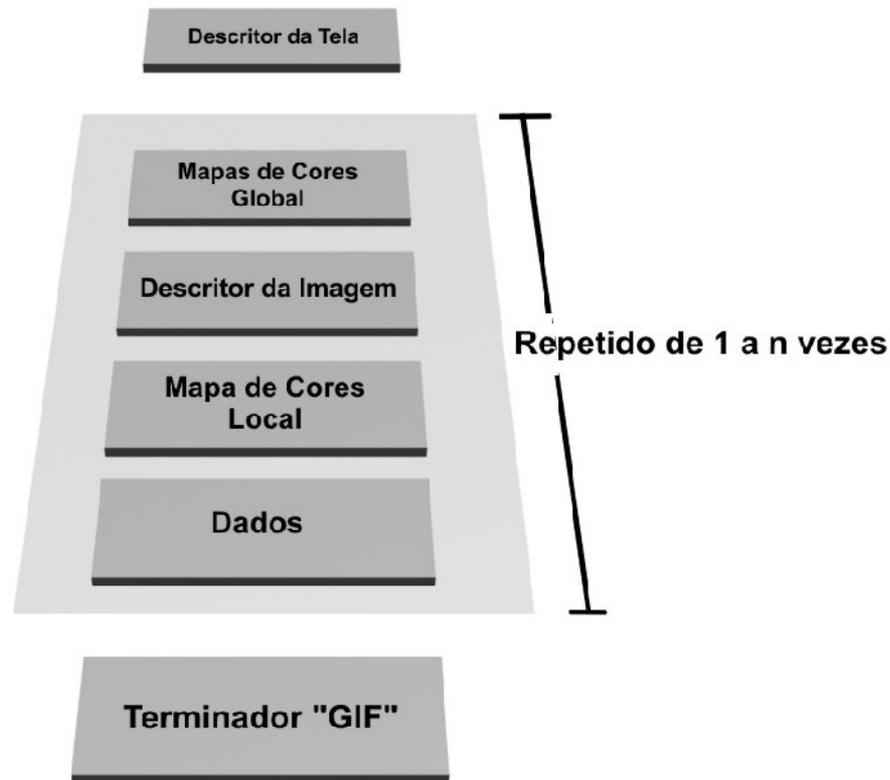


Figura 8.42 - elementos principais do formato

8.9.1.2. Descritores de Tela



Figura 8.43- Campos descritores da tela virtual.

8.9.1.3. Mapa de Cores Global

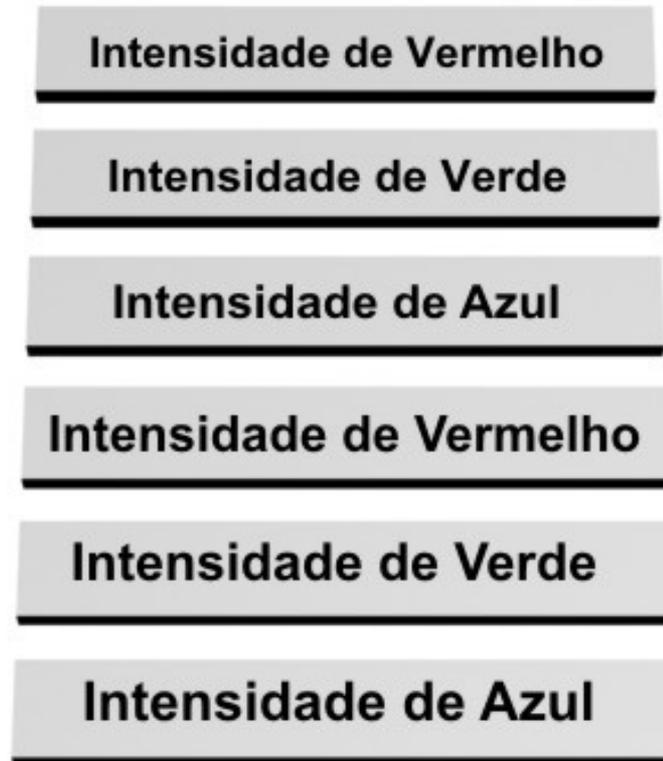


Figura 8.44 - Formação das cores na tabela de cores global.

8.9.1.4. Descritores da Imagem

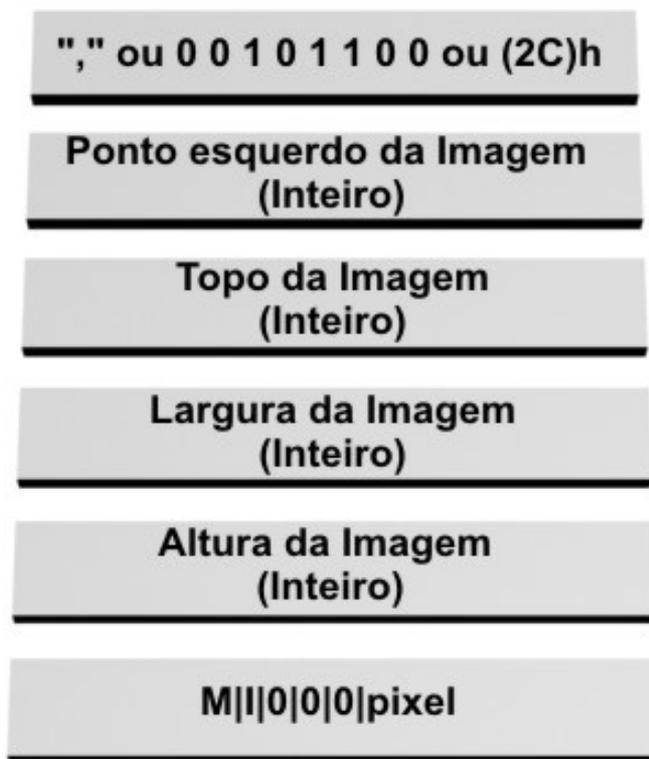


Figura 8.45 - Descritores de imagem.

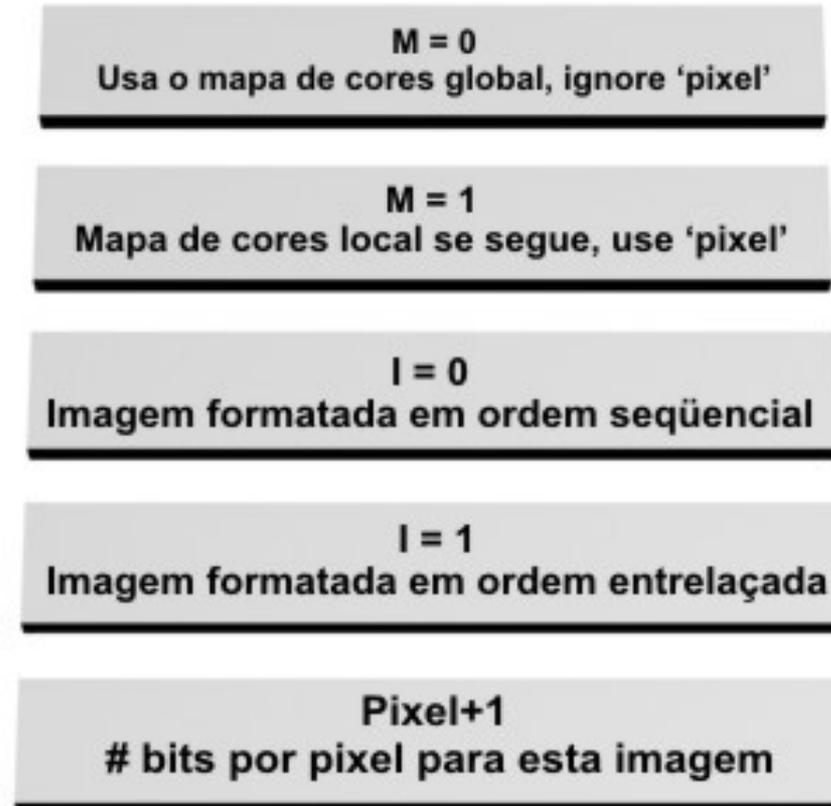


Figura 8.46 - Significado dos bits do ultimo byte do descritor da imagem - versão 87^a.

8.9.1.5. Mapa de Cores Local

- É opcional.
- Se o bit ‘M’ do byte 10 do descritor da imagem estiver “ligado”, logo um mapa de cores se segue ao descritor da imagem.

8.9.1.6. Dados da imagem

Linha	Passo 1	Passo 2	Passo 3	Passo 4	Resultado
0	**1a**				**1a*
1				**4a**	**4a**
2			** 3a**		**3a**
3				**4b**	**4b**
4		**2a**			**2a**
5				**4c**	**4c**
6			**3b**		**3b**
7				**4d**	**4d**
8	**1b**				**1b**
9				**4e**	**4e**
10			**3c**		**3c**
11				**4f**	**4f**
12		**2b**			**2b**

8.9.1.7. Terminador GIF

O software de decodificação para quando o caracter 0x3B hex ou ‘;’ for encontrado após a imagem ser processada.

Espera então por uma ação indicando que o usuário esta pronto para continuar.

Depois, o software de decodificação sai do modo gráfico e finaliza quaisquer processos anteriores.

8.9.2. PNG

- PNG (Portable Network Graphics) - usa uma variação do algoritmo Lempel-Ziv 77 e também compressão Huffman , depois da compressão LZ, numa forma denominado LZH ou de Deflate/Inflate.
- PNG com animação é conhecido como MNG (Multiple Image Network Graphics).
- PNG surgiu em 1996 como substituto para o formato GIF.
- Permite comprimir as imagens sem perda de qualidade com muitas cores.

8.9.3. JPEG

- JPEG (Joint Photographic Experts Group) - compressão com perdas.
- Permite o uso de até 2^{24} · 16 milhões de cores.
- O tamanho dos seus arquivos de imagens costuma ser bem pequeno.
- Usa a compressão DCT (e depois Huffman).
- Permite gravar imagens sem usar tabelas de cores usando toda a informação de Tons de Cinza ou RGB.
- A compressão é realizada em três passos: computação; quantização; atribuição do código de tamanho variável.



Figura 8.47 – Imagem Mac em JPG com 75% de compressão e 10%.



JPEG



GIF

Figura 8.48 – Comparando GIF com JPEG.

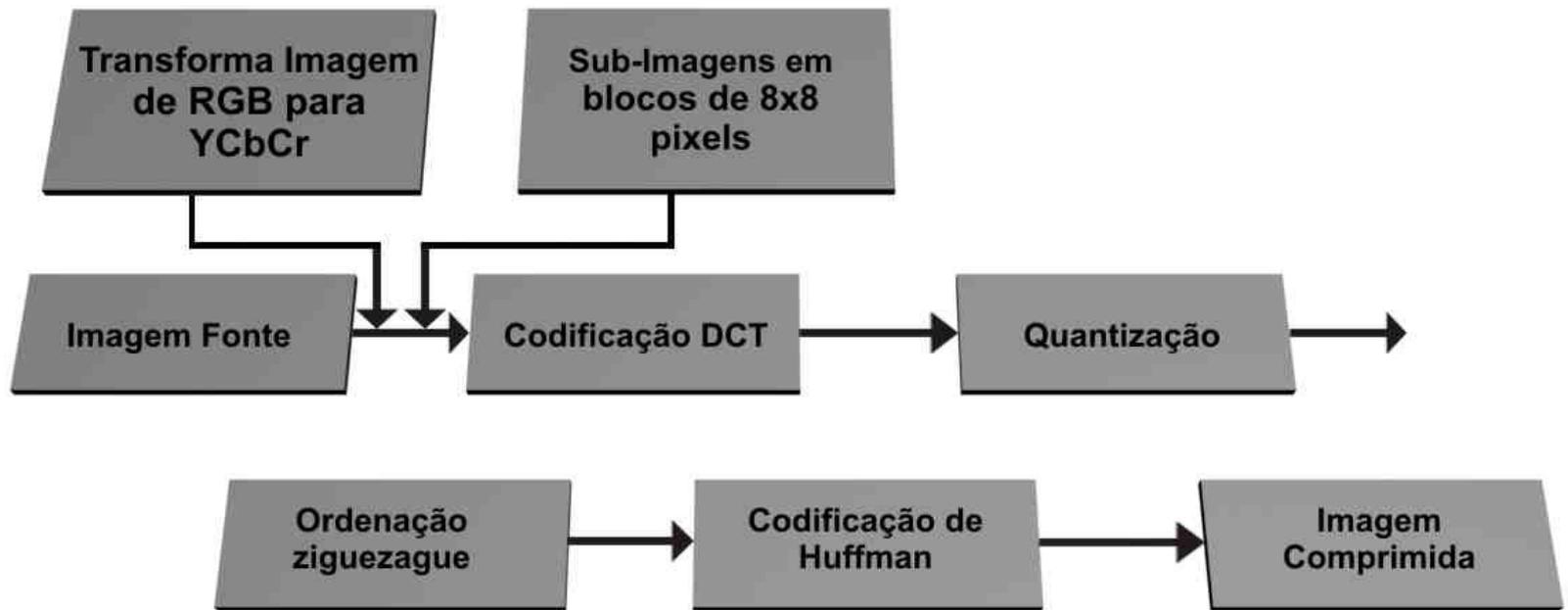


Fig. 8.49. Operações da compressão JPEG.

O processo de compactação JPEG é composto das seguintes fases:

- A imagem é dividida em blocos de 8x8 *pixels* e em cada um destes blocos é calculada a DCT (discrete cosine transform);
- Os coeficientes gerados pela DCT são quantizados e alguns coeficientes até eliminados.
- Na última etapa a codificação de Huffman é aplicada aos coeficientes quantizados.

O formato JPEG utiliza uma matriz de quantização $Q[i,j]$.

Uma das possibilidades é dada pela expressão:

$$Q[i, j] = 1 + (1 + i + j) * \text{fator_de_quantização} \quad (8.64)$$

Obs: O fator_de_quantização é considerado entre 2 e 25 e os índices iniciam no zero.

Tabela 8.10. Matriz de quantização para fator 2.

3	5	7	9	11	13	15	17
5	7	9	11	13	15	17	19
7	9	11	13	15	17	19	4
9	11	13	15	17	19	4	23
11	13	15	17	19	4	23	25
13	15	17	19	4	23	25	27
15	17	19	4	23	25	27	29
17	19	4	23	25	27	29	31

Tabela 8.11. Matriz de quantização para fator 5.

6	11	16	4	26	31	36	41
11	16	4	26	31	36	41	46
16	4	26	31	36	41	46	51
4	26	31	36	41	46	51	56
26	31	36	41	46	51	56	61
31	36	41	46	51	56	61	66
36	41	46	51	56	61	66	71
41	46	51	56	61	66	71	76

Tabela 8.12. Coeficientes gerados pela transformada do co-seno, antes da quantização com fator 5.

92.1	45	-32	-7.3	1.1	-1	0	1.9
68.5	-87	49.8	28.1	11.3	-1.3	2.7	1.3
-49	69.1	-61	-13	10.1	1.7	1.9	0
23.5	-52	33.4	45.4	-17	2	0.1	23
-41	19.4	-5.9	7.8	19.2	-5.1	-1.3	0
17.3	-5.2	11.7	2.9	-13	11.3	2.5	-1.7
-41	1.7	3.2	-1.1	5.3	2.5	-0.7	0
0 -	1.9	2.1	0	5.2	0	1.2 -	1

Tabela 8.13. Coeficientes reconstruídos depois de quantização com fator 5.

90	44	-32	0	0	0	0	0
66	-80	42	26	0	0	0	0
-48	63	-52	0	0	0	0	0
4	-52	31	36	0	0	0	0
-26	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
36	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

8.9.4. JPEG2000

- JPEG 2000 é um formato de codificação de imagem que usa técnicas de compressão *wavelets*.
- Foi criado em 1999 utilizando métodos de lógica nebulosa para criar os dados de origem.
- Pode compactar até 90% do arquivo original sem perder a qualidade de imagem,

8.9.5. MJPEG

- O MJPEG (“motion-JPEG”) - comprime na forma JPEG cada quadro de vídeo antes da transmissão.
- Não usa compressão inter-quadro.
- Escolhe automaticamente a taxa de quadro vídeo, baseado no codec usado (NTSC, PAL, SECAM, etc).

8.9.5.1. MPEG-1

- MPEG (Motion Pictures Experts Group) arquiva a imagem em três estágios: redução da banda passante, compressão com perda, e o estágio final uma compressão com menos perda.
- O algoritmo de compressão é baseado em duas técnicas básicas: “block based motion compensation” e domínio da transferência (Transformada Discreta do Cosseno).
- O MPEG-1 é o padrão original do MPEG e é capaz de codificar áudio e vídeo a uma taxa de 1,15 MB/s.
- O MPEG define três níveis ou camadas de compressão para áudio.

8.9.5.2. MPEG-2

- É um padrão de compactação de maior qualidade.
- Pode ser utilizado em transmissões a taxas de 4 a 9 MB/s.
- Uma versão modificada do MPEG-2 é usada pelo padrão HDTV e também nos DVDs.

8.9.5.3. MPEG-4 e derivados

- MPEG 4 e seus derivados (DiVX, XViD, etc.) são os mais usados atualmente.
- Podem oferecer qualidade semelhante à MPEG-2 sem ocupar tanto espaço.

8.9.6. BMP

- BMP foi desenvolvido pela Microsoft.
- É o formato nativo de mapa de bits do Windows (a partir da versão 3.00).
- Sua estrutura possui simplicidade.

8.9.6.1. Plataformas de utilização do formato BMP

- Projetado para sistemas operacionais que rodem sobre a plataforma INTEL (windows, linux, etc).
- Em outros tipos de arquiteturas como, por exemplo, Macintosh, deve ser usado outro formato mais adequado (PCX, GIF, TIFF etc).

8.9.6.2. Forma de armazenamento de arquivos BMP: DIB

- São armazenados no formato DIB (Device-Independent Bitmap).
- Permite exibir a imagem em qualquer dispositivo; ou seja o bitmap especifica a cor do *pixel* numa forma independente do método usado pelo dispositivo para representá-la.
- O BMP usa Formato Posicional, onde o significado do byte depende de sua posição no arquivo.

8.9.6.3. Versões de BMP quanto à quantidade de cor

- Podem ser classificados conforme a quantidade de bits para representar 1 *pixel* (bit/*Pixel*).

1 bit/*pixel* ($2^1=2$ cores),

4 bits/*pixel* ($2^4=16$ cores),

8 bits/*pixel* ($2^8=256$ cores),

24 bits/*pixel* (*true color* com até $2^{24}=16$ milhões de cores),

32 bits (*true color* com até $2^{32}=4$ bilhões de cores).

8.9.6.4.RLE

- Arquivos formato BMP podem, nas versões de 4 e 8 bits/*pixel*, utilizar a compressão RLE (Run Length Encoded)
- A técnica de compressão RLE é usada neste formato somente até 256 cores

8.9.6.5 - 8 bits/pixels

- Este formato o RLE tem usa 2 modos, denominados *Encoded mode* e *Absolute mode*.
- **Encoded mode:** compressão RLE em 2 bytes. O 1º byte especifica o número de *pixels* consecutivos que serão desenhados usando o índice de cor contido no 2º byte.
- **Absolute mode:** o 1º byte do par é ZERO e o 2º byte é um valor entre 0x03 e 0xff (isto é, 3 ou 255). Neste caso o valor do 2º byte representa o número de bytes seguintes que serão descritos na forma não comprimida.

Tabela 8.15 – Exemplo de Bitmap BMP e seu significado.

Dados Comprimidos	Dados Expandidos	Tipo
03 04	04 04 04	Encode
05 06	06 06 06 06 06	Encode
00 03 45 56 67	45 56 67	Indicativo Absolute Mode
00 02 05 01	delta de 5 <i>pixels</i> p/esquerda e 1 p/baixo	Indicativo Delta
02 78	78 78	Encode
00 00	Fim de linha	Indicativo Fim Linha
09 1E	1E 1E 1E 1E 1E 1E 1E 1E 1E	Encode
00 01	Fim da Imagem	Indicativo Fim Arquivo

8.11.6.6 4 bits/pixels

- Usa os mesmos 2 modos anteriores, mas agora cada 4 bits representam um dado. Um diferencial é o conceito de cor primária e secundária.
- **Encoded mode:** O 1º byte do par contém o número de *pixels* que serão desenhados usando os índices de cores do 2º byte. O 2º byte contém 2 índices de cores
- **Absolute mode:** O 1º byte contém ZERO e o 2º byte contém o número de índices de cores que se seguem e os bytes seguintes indicam os índices de cores primárias e secundárias, sendo um índice de cores para cada *pixel*.

Tabela 8.16 – Dados expandidos.

Dados Comprimidos	Dados Expandidos
03 04	0 4 0
05 06	0 6 0 6 0
00 03 45 65 67	4 5 6 5 6 7
00 02 05 01	Delta: mova 5 p/esquerda e 1 p/baixo
02 78	7 8 7 8
00 00	Fim de Linha
09 1E	1 E 1 E 1 E 1 E 1
00 01	Fim dos dados da imagem

Delta: é entendido como um deslocamento do próximo *pixel* a ser representado na tela, da posição de sucessor do *pixel* anterior usual para a posição ocupada pelos próximos bytes do arquivo que seguem.

8.9.6.7 . Estrutura geral do formato BMP

- a) Cabeçalho de arquivo: Contém a assinatura BM e informações sobre o tamanho e lay-out do arquivo BMP.
- b) Cabeçalho de mapa de bits: Contém as informações da imagem contida no arquivo.
- c) Paleta ou mapa de cores (opcional): Somente estará presente em arquivos de imagens que usam 16 ou 256 cores.
- d) Área de dados da imagem contida no arquivo: Dados que permitem a exibição da imagem propriamente dita, o dados dos *pixels* a serem exibidos.

8.9.6.8 Estrutura detalhada do formato BMP

A) Cabeçalho de arquivo – informações do arquivo - Tamanho: 14 bytes

Campo	Bytes	Descrição
BfType	2	Assinatura do arquivo: os caracteres 4D)h.
BfSize	4	Tamanho do arquivo em Bytes
BfReser1	2	Campo reservado 1; deve ser ZERO
BfReser2	2	Campo reservado 2; deve ser ZERO
BfOffsetBits	4	Especifica o deslocamento, em de dados da imagem: - Se a imagem usa tamanho= $14+40+(4 \times \text{NumeroDeCores})$ - Se a imagem for true tamanho= $14+40=54$

B) Cabeçalho de mapa de bits – informações da imagem -
Tamanho: 40 bytes.

Campo	Bytes	Descrição
BiSize	4	Tamanho deste cabeçalho (40 bytes). Sempre (28) _h .
BiWidth	4	Largura da imagem em <i>pixels</i>
BiHeight	4	Altura da imagem em <i>pixels</i>
BiPlanes	2	Número de planos de imagem. Sempre 1
BiBitCount	2	Quantidade de Bits por <i>pixel</i> (1,4,8,24,32)
BiCompress	4	Compressão usada. Pode ser: 0 = BI_RGB _ sem compressão 1 = BI_RLE8 – compressão RLE 8 bits 2 = BI_RLE4 – compressão RLE 4 bits
BiSizeImag	4	Tamanho da imagem (dados) em byte - Se arquivo sem compressão, este campo pode ser ZERO. - Se imagem em true color, será Tamanho do arquivo (Bfsize) menos deslocamento (BfOffSetBits)
BiXPPMeter	4	Resolução horizontal em <i>pixels</i> por metro
BiYPPMeter	4	Resolução vertical em <i>pixels</i> por metro
BiClrUsed	4	Número de cores usadas na imagem.
BiClrImpor	4	Número de cores importantes (realmente usadas) na imagem.

C) Paleta cores - definição tabela de cores - Tamanho: 4 bytes x Número de Cores.

Campo	Bytes	Descrição
Blue	1	Intensidade de Azul. De 0 a 255
Green	1	Intensidade de Verde. De 0 a 255
Red	1	Intensidade de Vermelho. De 0 a 255
Reservado	1	Campo reservado deve ser ZERO sempre

D) Área de dados da imagem - cor que cada *pixel* deve ser ligado ou esses dados comprimidos - Tamanho: campo `BiSizeImg`, do cabeçalho de informações da imagem.

Varia conforme existência ou não de compressão.

Para imagens sem compressão, os dados são armazenados em uma ordem sequencial, que corresponde a posições na tela de vídeo.

8.9.7. Formato PCX

- Tornou-se popular com a distribuição do PC Paintbrush.
- É utilizado pela maioria dos scanners, programas de editoração eletrônica e fax.
- Utiliza a compressão RLE (Run Length Encoding).
- Os primeiro 128 bytes de um arquivo PCX constituem o cabeçalho ou HEADER.

INFORMAÇÃO	BYTES	DESCRIÇÃO
Identificação	1	Contém o valor A0 em hexadecimal
Versão	1	A versão do Paintbrush ou compatível
Compressão	1	Define o tipo de compressão
Bits-per- <i>pixel</i>	1	Indica quantos bits consecutivos no arquivo representarão um <i>pixel</i> na tela.
Xmin	2	Representa o limite inferior horizontal da imagem
Ymin	2	Representa o limite inferior vertical da imagem
Xmax	2	Representa o limite superior horizontal da imagem
Ymax	2	Representa o limite inferior vertical da imagem
Hres	2	Resolução horizontal da imagem.
Vres	2	Resolução vertical da imagem.
Palette	48	Contém a palette de cores do arquivo se esse possuir até 16 cores.
Reservado	1	Não é utilizado, mas contém o modo do vídeo (BIOS).
Bytes-per-line	2	Indica quantos bytes existem em cada scan-line.
Tipo de palette	2	Indica imagens em níveis de cinza ou coloridas.