

## ESTUDO DE ALGORITMOS PARA A COMPRESSÃO DE IMAGENS

**Gilmara LINHARES; Janderson DINIZ; Suzete CORREIA**

Centro Federal de Educação Tecnológica da Paraíba, Av. 1º de maio, 720, CEP: 58015-430,  
(083) 3208-3000, fax (083)3208-3088, [www.cefetpb.edu.br](http://www.cefetpb.edu.br)  
[gilmara\\_linhares@yahoo.com.br](mailto:gilmara_linhares@yahoo.com.br), [janderson.diniz@gmail.com](mailto:janderson.diniz@gmail.com), [suzete.correia@gmail.com](mailto:suzete.correia@gmail.com)

### RESUMO

Em razão da implantação da TV Digital, no Brasil, o estudo de técnicas para a compressão de imagens tornou-se indispensável para a transmissão ou armazenamento de dados de forma eficiente. Grande parte da informação gerada pela digitalização de uma imagem é composta por redundâncias que podem ser eliminadas ou reduzidas sem acarretar uma imagem visualmente disforme. A compressão de imagens é empregada nos CODECs de TV Digital a fim de proporcionar o uso eficiente da largura de banda do canal através de uma multiprogramação em uma mesma frequência. Este artigo tem por objetivo descrever os principais algoritmos de compressão de imagens estáticas que são empregados nos codificadores de vídeo, tais como: Huffman (Entropia), RLE (*Run Length Encoding*) e DCT (*Discrete Cosine Transform*).

**Palavras-chave:** compressão de imagens, código de Huffman, RLE (*Run Length Encoding*) e DCT (*Discrete Cosine Transform*).

## 1. INTRODUÇÃO

A digitalização de uma imagem produz uma grande quantidade de bits, o que requer um canal com maior capacidade de transmissão e uma quantidade maior de memória para o seu armazenamento. Com o intuito de solucionar esse impasse técnicas de compressão de imagens foram e continuam a ser desenvolvidas.

Grande parte da informação contida numa imagem pode ser removida sem causar distorções, uma vez que o sistema visual humano não é sensível às altas frequências menos contributivas para a imagem (WALDMAN, 2000). As técnicas de compressão de imagens tentam fazer dessa limitação do olho humano a sua aliada para reduzir a quantidade de bits representativos da imagem.

A compressão de imagens encontra aplicações em diversos segmentos tais como computação multimídia, transmissão de televisão, imageamento médico e geográfico, entre outras.

Em razão da implantação da TV Digital, no Brasil, essas técnicas tornaram-se indispensáveis à otimização do canal de transmissão, sendo empregadas nos codificadores e decodificadores (CODECs) de vídeo dos diferentes padrões de TV Digital.

Neste artigo serão apresentados os principais algoritmos de compressão de imagens estáticas que são empregados nos codificadores de vídeo, tais como: Huffman (Entropia), RLE (*Run Length Encoding*) e DCT (*Discrete Cosine Transform*).

## 2. COMPRESSÃO DE IMAGENS

### 2.1 Definição

O processo pelo qual se digitaliza uma imagem gera uma enorme quantidade de dados. A imagem analógica representada por uma função intensidade de luz bidimensional  $f(x,y)$  é amostrada e quantizada para ser digitalizada (GONZALEZ, 2000).

A amostragem consiste em discretizar as coordenadas espaciais  $(x,y)$  em uma matriz de pontos elementares, em que cada ponto recebe o nome de *pixel*. Quanto maior a quantidade de *pixels* melhor a resolução da imagem e conseqüentemente maior o seu tamanho. A quantização consiste em representar cada *pixel* por um valor que indica a intensidade de brilho, denominado nível de cinza. A quantidade de níveis de cinza depende da quantidade de bits usada na representação de cada *pixel*. No entanto, os coeficientes quantizados nem sempre correspondem à intensidade de brilho do sinal contínuo da imagem original, dando origem aos erros de quantização (BATISTA, 2005a).

Reduzir a quantidade de dados de uma imagem digital é importante, para tornar eficiente o seu armazenamento, processamento e transmissão. Para tal finalidade, foram desenvolvidos algoritmos de compressão. Os algoritmos de compressão pretendem reduzir a taxa de bits, principalmente pela eliminação de informação visual redundante e irrelevante.

Na compressão de imagens, há dois tipos de redundância que são levadas em consideração: redundância espacial e temporal e redundância psicovisual (TUDOR, 1995).

Na redundância espacial e temporal os *pixels* de um quadro de imagem apresentam correlação entre si sendo possível prever o valor de um pixel pelos valores dos pixels vizinhos.

Na redundância psicovisual existem informações que podem ser eliminadas sem serem percebidas pela percepção visual humana. O olho humano não analisa cada pixel quantitativamente, mas procura perceber detalhes como bordas e textura da imagem, combinando-os em agrupamentos reconhecíveis (GONZALEZ, 2000).

De acordo com Gonzalez (2000) a compressão de imagens trata o problema de reduzir a quantidade de dados necessária para representar uma imagem digital. A base do processo de redução é remoção dos dados redundantes, ou seja, comprimir uma imagem implica em eliminar dados desnecessários. Quanto maior for a remoção de dados sem prejudicar a qualidade da imagem, melhor será o tipo de compressão.

## 2.2. Tipologia e Aplicações

Ao longo dos anos tem se tornado cada vez mais necessário comprimir imagens, com a finalidade de aumentar a capacidade de armazenamento e transmissão, e assim, melhorar o tráfego de informações no canal de transmissão, uma vez que uma maior quantidade de informações será enviada em um determinado instante.

Para isso deve-se eliminar ou reduzir informações redundantes contidas no sinal enviado, pois o armazenamento e transmissão de vídeo digital requerem grande capacidade de banda passante.

Por determinação da ANATEL (Agência Nacional de Telecomunicações) a transmissão de sinal de vídeo digital, no Brasil, ocupará a mesma largura de banda do sinal analógico (6 MHz) sendo possível transmitir simultaneamente 4 canais. Em virtude dessa necessidade, faz-se fundamental a compressão dos dados.

Há dois tipos de compressão de imagem: sem perdas e com perdas. A primeira consiste em reduzir o número de bits sem haver perdas de informação, enquanto que a segunda consiste em uma redução maior da taxa de bits até o limite de perdas admissíveis que não serão percebidas pelo observador.

A distinção do uso de um determinado tipo de compressão é em função de sua utilização. A compressão sem perdas é exigida para o processamento de imageamento médico, transmissão de facsímiles de imagens, imagens Landsat, em que tanto o uso quanto o custo da coleta dos dados torna qualquer perda indesejável. A compressão com perdas é aplicada para habilitar aplicações em tempo real envolvendo a manipulação e transmissão de informação audiovisual (GONZALEZ, 2000).

O padrão americano de televisão digital ATSC e o padrão japonês ISDB-T utilizam MPEG-2 como CODEC de vídeo enquanto que o padrão europeu DVB e o padrão brasileiro SBTVD-T adotam o H.264/AVC (MPEG-4 parte 10).

### 2.2.1. Compressão Sem Perdas

A técnica de compressão de imagens sem perdas tem como objetivo reduzir o número de bits representativos de um sinal de imagem digital sem a perda de informações importantes, ou seja, sem prejudicar a qualidade da imagem quando esta for descomprimida. Na compressão sem perdas, a imagem reconstruída será idêntica à imagem original.

O uso dessa técnica permite eliminar redundância com percentual de informações irrelevantes, e com isso, tenta-se obter uma imagem não necessariamente idêntica numericamente, mas, a forma visual idêntica à imagem original. Por essa razão é considerada compressão de alta qualidade.

Um sistema genérico de compressão é composto de um codificador e decodificador. O primeiro apresenta na entrada um sinal de imagem que é transformado em símbolos. Esses símbolos são transmitidos através de um canal até chegarem ao decodificador, onde é gerada uma imagem reconstruída.

De acordo com Karam (2000) o codificador de imagens sem perdas é composto por três estágios: transformação, mapeador e um codificador de símbolos, (ver Figura 1).

No primeiro estágio, os dados de imagem sofrem uma transformação que é reversível. Esta transformação consiste em converter o dado  $Y$  que entra em  $\mathbb{Y}$ , que poderá ser comprimido mais eficientemente.

No segundo estágio, o mapeador converte a imagem transformada  $\mathbb{Y}$  em símbolos. Segundo Karam (2000) a conversão em símbolos pode ser feita através de particionamento ou codificação por comprimento de corrida (RLC – *Run Length Coding*). Nesta etapa ocorre a redução de redundâncias interpixels da imagem da entrada.

O RLC, também conhecido como RLE (*Run Length Encoding*), trata-se de um método que preserva a informação e pode ser utilizado tanto em imagens binárias quanto em níveis de cinza. É obtido quando uma sequência de símbolos repetidos é gerada. Ao invés da fonte enviar cada símbolo isoladamente, é feito um mapeamento que envia o símbolo e o número de vezes que ele aparece na sequência. Por exemplo, a sequência aaabbaabbb pode ser representada como (3,a) (2,b) (2,a) (3,b).

Na terceira etapa, o codificador de símbolos gera uma sequência binária que resulta em uma palavra-código a fim de representar os símbolos de entrada.



Figura 1 - Esquema básico de um compressor sem perdas

### 2.2.2. Compressão Com Perdas

Diferentemente do que acontece com o método de compressão sem perdas, a compressão com perdas tem um aumento significativo na taxa de compressão, o que a torna mais vantajosa em relação ao método sem perdas. Essa vantagem é observada apenas quantitativamente e não qualitativamente.

Os compressores com perdas utilizam três operações básicas: transformada, quantização e codificação, (ver Figura 2).

Segundo Hunt (1978), a redundância existente em uma imagem se caracteriza pela correlação entre as amostras com um alto grau de predição estatística entre as amostras adjacentes.

A fim de eliminar a correlação entre as amostras é empregada uma operação matemática, denominada transformada, para concentrar energia em poucos coeficientes de valor absoluto elevado, os quais propiciam a recuperação das amostras do sinal sem haver perdas.

Os valores absolutos são obtidos após se aplicar a transformada a uma matriz representativa da imagem, cujos elementos correspondem aos níveis de intensidade luminosa, níveis de cinza, onde ao se ter  $n$  bits atribuídos a um pixel implica em  $2^n$  níveis de cinza.

Os valores possíveis para os pixels são  $0, 1, 2, \dots, (2^n - 1)$ . Nesse caso o valor 0 corresponde à cor preta,  $(2^n - 1)$  a cor branca e os valores intermediários correspondem a diferentes tonalidades de cinza, sendo que estas se intensificam gradativamente à medida que se aproxima de  $(2^n - 1)$ .

A transformada converte sinais cujo domínio é o tempo ou o espaço para o domínio da frequência, dependendo da transformada empregada. Em uma grande sequência a ser transformada há um alto custo computacional, então esta é particionada em blocos de tamanho fixo ao objetivar a facilitação do processamento. Os padrões JPEG e MPEG usam blocos de  $8 \times 8$  pixels e o H.264 (MPEG-4 parte 10) utiliza  $4 \times 4$ , embora se saiba que quanto maior o bloco menor será a relação entre compressão e distorção (BATISTA, 2005). Dentre as transformadas mais utilizadas por compressores de alto desempenho tais como JPEG, MPEG e H.264 pode-se citar a DCT (*Discrete Cosine Transform*).

A etapa posterior à transformada é a quantização. De acordo com Tudor (1995) a quantização é usada para reduzir o número de valores possíveis a serem transmitidos e ajustar a imagem dentro de um número finito de níveis de cinza que depende da quantidade de bits empregados na representação de cada *pixel*.

Depois de realizar a quantização é feita a codificação dos coeficientes quantizados. Em geral usam-se a codificação de Huffman ou codificação aritmética (BATISTA, 2005), em virtude de já ter havido perda de informação e, por conseguinte, mais perdas são inadmissíveis.

A quantização é um processo irreversível o que causa perda de informação, ou seja, o seu uso estabelece a compressão com perdas. Gonzalez (2000) afirma que esse estágio reduz a acurácia de saída do mapeador de acordo com algum critério de fidelidade preestabelecido. É nesta etapa do processo que as redundâncias psicovisuais da imagem são reduzidas.

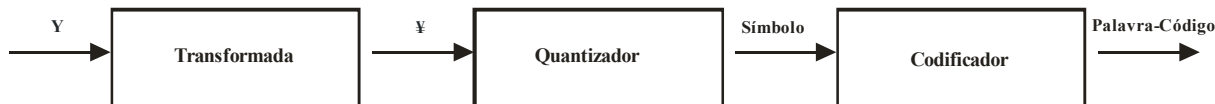


Figura 2 - Esquema básico de um compressor com perdas

### 3. ALGORITMOS DE COMPRESSÃO SEM PERDAS

#### 3.1. Codificação de Huffman

A codificação de Huffman é uma das mais utilizadas para redução de redundância. Segundo Gonzalez (2000) a codificação de Huffman cria um código ótimo para um conjunto de símbolos, uma vez que explora eficientemente as probabilidades de emissão de cada símbolo pela fonte. Contudo, para uma grande quantidade de símbolos-fonte o código de Huffman torna-se dispendioso em termos de complexidade computacional.

Este código foi desenvolvido por D. Huffman para construir códigos de prefixo. Esses códigos caracterizam-se pelo fato de uma palavra-código não ser prefixo de outra, o que garante a não ambigüidade do código.

A codificação de Huffman satisfaz o Primeiro Teorema de Shannon (KARAM, 2000) mostrado na equação abaixo:

$$H(s) \leq l_{med} \leq H(s) + 1 \quad [\text{Eq. 01}]$$

em que  $H(s)$  é a entropia da fonte e  $l_{med}$  o comprimento médio do código.

Com o objetivo de tornar o processo de codificação mais eficiente, Huffman procurou atender ao seguinte critério: representar símbolos com maior probabilidade de ocorrência com um número menor de bits diferentemente daqueles de menor probabilidade, os quais são representados por um número maior de bits.

O processo de codificação de Huffman é realizado ao se ordenar os símbolos por ordem decrescente de probabilidade para formar o diagrama de árvore, (ver Figura 3).

As etapas para a construção do diagrama de árvore são apresentadas abaixo:

1. Dada uma fonte com alfabeto  $S = \{s_0, s_1, s_2, \dots, s_k\}$  com suas respectivas probabilidades, ordena-se os símbolos da fonte em ordem decrescente de probabilidade de cima para baixo.
2. Ao ordenar as probabilidades dá-se início à redução da fonte a partir da soma das duas probabilidades menores.
3. A probabilidade resultante dessa soma será colocada em outra coluna, em que as probabilidades da coluna anterior serão repetidas e o novo valor será alocado de maneira a manter a ordem decrescente. O processo continua até que a soma das probabilidades seja igual a 1, número associado à raiz da árvore.
4. Após a etapa 3, a cada símbolo é atribuído um código que é obtido ao pegar os bits da raiz como sendo o início da seqüência e segue a ramificação até chegar ao símbolo  $s_k$ . A Tabela 1 mostra os códigos obtidos para o exemplo da Figura 3.

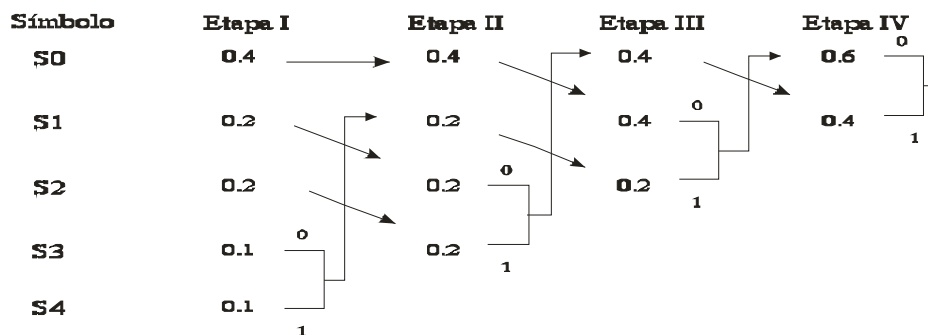


Figura 3 - Árvore de codificação de Huffman

**Tabela 1 - Palavras-código obtidas pela codificação**

Símbolo	Probabilidade	Palavra-código
$s_0$	0,4	00
$s_1$	0,2	10
$s_2$	0,2	11
$s_3$	0,2	010
$s_4$	0,1	011

Sabendo-se que a entropia é dada por:

$$H(s) = -\sum_{i=0}^k p(s_i) \log_2 p(s_i) \text{ (bits/símbolo)} \quad [\text{Eq. 02}]$$

e o comprimento médio por:

$$l_{med} = \sum_{i=0}^k p(s_i) l_i \quad [\text{Eq. 03}]$$

então, a eficiência do código é obtida por:

$$\eta = \frac{H(x)}{l_{med}} \quad [\text{Eq. 04}]$$

A partir dos dados contidos na Tabela 1 e ao usar as equações 02, 03 e 04 é possível calcular a entropia da fonte, o comprimento médio e a eficiência da codificação exemplificada na Figura 3.

Dessa forma, tem-se:

$$\begin{aligned} H(x) &= -(0,4 \log_2 0,4 + 0,2 \log_2 0,2 + 0,2 \log_2 0,2 + 0,2 \log_2 0,2 + 0,1 \log_2 0,1) \\ &= 2,26 \text{ bits/símbolo} \end{aligned}$$

$$\begin{aligned} l_{med} &= 0,4 \cdot 2 + 0,2 \cdot 2 + 0,2 \cdot 2 + 0,2 \cdot 3 + 0,1 \cdot 3 \\ &= 2,5 \text{ bits/símbolo} \end{aligned}$$

$$\eta = 0,904$$

Como já foi visto o processo de codificação consiste em mapear os símbolos-fonte em uma sequência de palavras-código. A decodificação pode ser realizada ao construir uma *lookup table* (KARAM, 2000).

A *lookup table* possui o número de entradas igual a  $2^{l_{\max}}$ , onde  $l_{\max}$  é o comprimento da maior palavra-código. Um símbolo  $s_k$  terá  $2^{(l_{\max} - l_i)}$  entradas relacionadas a ele, em que  $l_i$  é o seu comprimento. Para cada símbolo mapeado em palavras-código nomeia-se em todas as entradas da tabela um par  $(s_k, l_i)$ .

Ao decodificar um trem de bits é identificado  $l_{\max}$  e este é usado como índice, na *lookup table*, para obter  $s_k$  decodificado cujo tamanho corresponde a  $l_i$ . Em seguida, o índice relacionado ao primeiro  $l_i$  é descartado e a sua direita anexa-se o próximo  $l_i$ , e assim por diante, até decodificar todo o trem de bits.

A *lookup table* tem por vantagem tornar o processo de decodificação mais rápido, no entanto, o tamanho da memória e complexidade cresce exponencialmente com  $l_{\max}$ .

Na codificação de imagens os símbolos-fonte são os níveis de cinza e aos que têm maior probabilidade de ocorrência se atribui os códigos de menor tamanho.

A eficiência do código de Huffman está diretamente relacionada à maneira como a fonte é modelada, ou seja, ao grau de variação das suas probabilidades.

Em fontes cujo arranjo de probabilidades varia com o tempo, pode ocorrer falha de codificação. Ao procurar solucionar esse problema encontrou-se a solução na codificação adaptativa de Huffman. Através desta é

introduzido um contador de número de ocorrências de um determinado símbolo e atualiza-se a tabela de mapeamento dos símbolos-fonte em palavras-código, usada tanto no codificador quanto no decodificador. Assim, o código de Huffman é adaptado dinamicamente para diferentes estatísticas de fonte.

### 3.2. Codificação Aritmética

Ao contrário da codificação de Huffman que atribui para cada símbolo-fonte uma palavra-código de pelo menos 1 bit, a codificação aritmética permite representar uma seqüência de símbolos por apenas uma palavra-código. Por isso, esta consegue taxas de compressão mais alta quando comparada à codificação de Huffman (KARAM, 2000).

A codificação aritmética remonta ao trabalho de Elias (ELIAS *apud* Gonzalez, 2000) e foi desenvolvida por Pasco e Rissanen (PASCO *apud* KARAM, 2000).

Em codificação aritmética a modelagem estatística da fonte pode ser separada do processo de codificação e decodificação. Por conseguinte, torna-se preferencial para codificação adaptativa, em relação à Huffman, porque neste tipo de código é necessária a alteração das árvores de codificação e decodificação quando as estatísticas da fonte variam.

O processo de codificação aritmética é executado a partir do momento em que se tem uma seqüência de símbolos-fonte na entrada do decodificador. A palavra-código resultante dessa seqüência estará contida no intervalo de  $[0,1)$ . Vale ressaltar que o intervalo  $[0, 1)$  é o ponto de partida para executar a codificação. Nele encontram-se os símbolos da mensagem agrupados em subintervalos de acordo com suas respectivas probabilidades.

À medida que aparece um novo símbolo é criado um novo subintervalo  $[l_i, l_s)$ , onde  $l_i$  indica o limite inferior e  $l_s$  indica o limite superior, com tamanho menor que o anterior e dentro do mesmo os símbolos-fonte tem seu subintervalo reajustado de acordo com a probabilidade de origem. Este processo continua até o fim da seqüência.

O subintervalo final corresponde à palavra-código representante da seqüência de símbolos-fonte. A palavra-código corresponde a uma fração binária. Ex.: 0,1 representa  $1/2$ , em virtude da transformação do intervalo representado em forma decimal ( $1/2$ ) em numeração binária (0,1).

Realizar a codificação aritmeticamente exige a execução dos seguintes passos:

1. Ao considerar uma fonte com alfabeto S, todos os símbolos estão contidos no intervalo  $[l_i, l_s)$ , cujo valor inicial é  $[0,1)$ , de acordo com as suas probabilidades de ocorrência.

2. Calcula-se a faixa do subintervalo:

$$\text{faixa} = l_s - l_i \quad [\text{Eq. 05}]$$

3. Dado um símbolo  $s_k$ , o subintervalo é atualizado:

$$l_i = l_i + \text{faixa} \cdot l_{isk} \quad [\text{Eq. 06}]$$

$$l_s = l_s + \text{faixa} \cdot l_{ssk} \quad [\text{Eq. 07}]$$

em que  $l_{isk}$  é o limite inferior do símbolo  $s_k$  e  $l_{ssk}$  é o limite superior do símbolo  $s_k$ .

4. Até o término da seqüência de entrada deve-se repetir todos os procedimentos já expostos a partir da etapa 2.

Dessa maneira chega-se ao subintervalo final, cujos valores podem ser usados para representar a palavra-código.

De posse da palavra-código é possível fazer a decodificação ao considerar:

1.  $l_i = 0$  e  $l_s = 1$

2. Calcula-se a faixa do subintervalo utilizando a equação 05:

3. Em seguida, pela equação 08:

$$\frac{\text{palavra} - \text{código(decimal)} - l_i}{\text{faixa}} \quad [\text{Eq. 08}]$$

Obtém-se o valor da igualdade, identifica-se o subintervalo de probabilidade de ocorrência no qual ele está contido e atribui-se o símbolo correspondente.

4. Sabendo-se o subintervalo em que está contido o símbolo  $s_k$  é necessário atualizar o intervalo. Para isso, utilizam-se as equações 06 e 07.

5. Enquanto o último símbolo não for decodificado repete-se a operação a partir da etapa 2.

Quanto maior a seqüência a ser codificada maior será o comprimento da palavra-código devido à necessidade de grande precisão.

Com a finalidade de facilitar a compreensão da codificação aritmética a Figura 4 mostra as etapas de codificação e decodificação para um alfabeto  $A = \{a_1, a_2, \text{EOS}\}$ , em que EOS (*End-of-sequence*) representa o *código de final de seqüência*, com probabilidades  $p(A) = \{0,5, 0,4, 0,1\}$ , sabendo-se que a seqüência codificada é  $a_2, a_1, a_2, \text{EOS}$ .

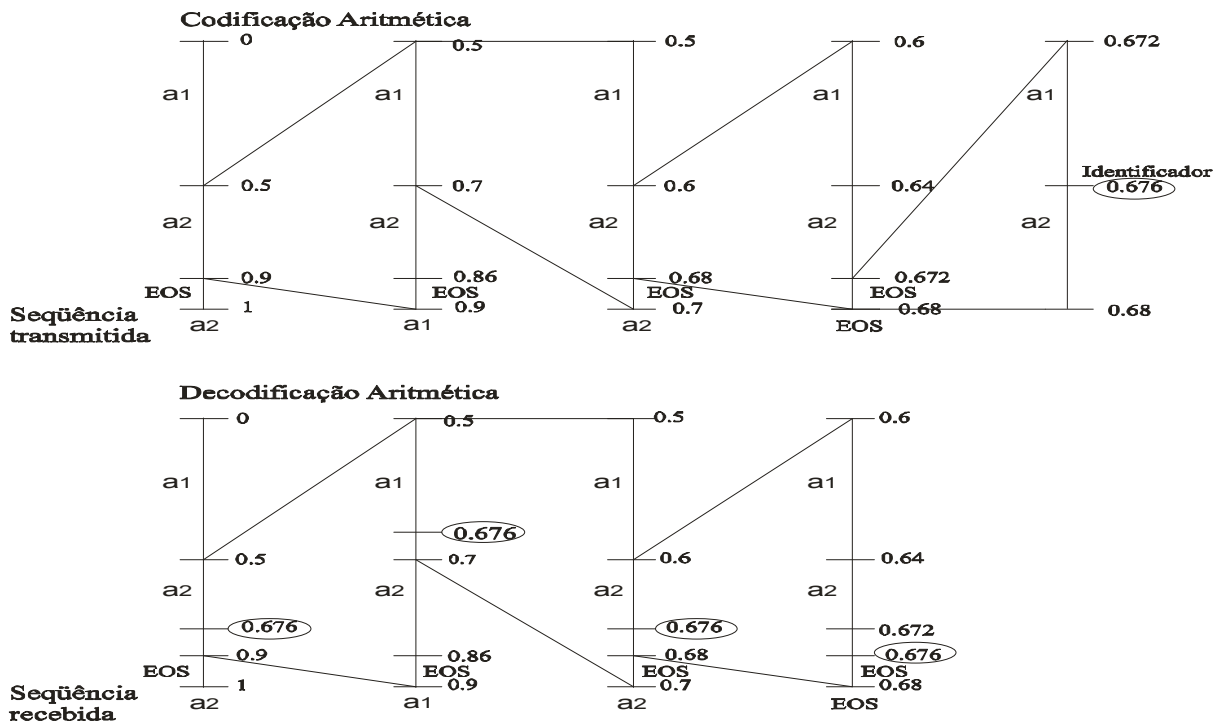


Figura 4 - Exemplos de codificação e decodificação aritmética (SOUZA, 2005)

## 4. ALGORITMOS DE COMPRESSÃO COM PERDAS

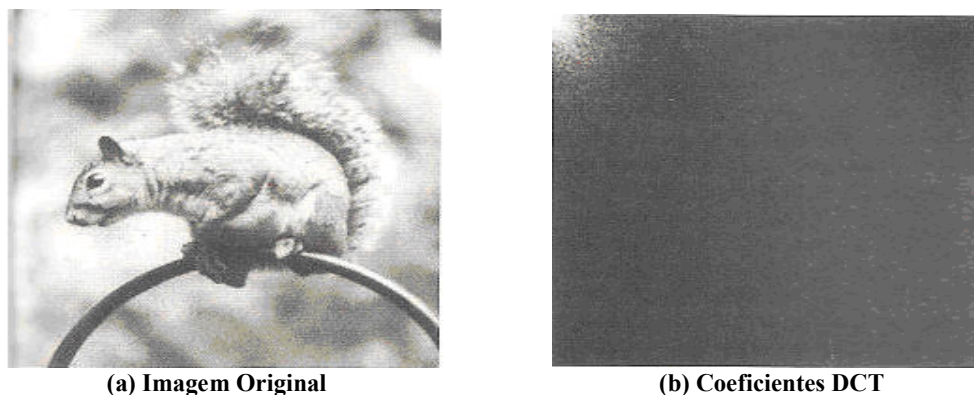
### 4.1. DCT (*Discrete Cosine Transform*)

A DCT é uma transformada ortogonal (OPPENHEIM, 1998) cuja propriedade de concentração de energia em poucos coeficientes constitui um atrativo para seu uso em compressão de imagens. A concentração de energia de uma matriz de pixels em poucos coeficientes proporciona o descorrelacionamento dos pixels vizinhos e torna possível eliminar ou reduzir a presença das altas freqüências de uma imagem durante o processo de quantização.

Quando se aplica a DCT a uma determinada imagem  $f(i,j)$  os valores mais significativos ficam concentrados no canto superior esquerdo dessa imagem.



O resultado da aplicação da transformada DCT a uma imagem pode ser visualizado na Figura 5. Os coeficientes mais significativos podem ser vistos na região mais clara da imagem dos coeficientes, enquanto que a parte escura corresponde aos coeficientes com pouca energia.



**Figura 5: Compressão de uma imagem através da DCT (GONZALEZ, 2000).**

Para uma matriz  $N \times N$  com coeficientes  $x_{ij}$ ,  $i, j = 0, 1, \dots, N - 1$ , ao aplicar a transformada DCT bidimensional obtém-se os coeficientes transformados  $X_{ij}$ ,  $i, j = 0, 1, \dots, N - 1$ . (BATISTA, 2005).

$$X_{ij} = \frac{1}{\sqrt{2N}} c_i c_j \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} x_{ij} \cos \left[ \frac{(2i+1)i\pi}{2N} \right] \cos \left[ \frac{(2j+1)j\pi}{2N} \right], \quad i, j = 0, 1, \dots, N - 1 \quad [\text{Eq. 10}]$$

em que:

$$c_i, c_j = \begin{cases} (1/2)^{1/2} & \text{para } i, j = 0 \\ 1 & \text{caso contrário} \end{cases}$$

Como já foi mencionado o olho humano não tem acuidade visual para altas frequências, portanto, eliminá-las não acarreta distorções para a imagem. Nesse caso a DCT age agrupando as altas e baixas frequências em coeficientes distintos, sendo que os coeficientes de baixas frequências concentram mais energia, diferentemente dos de alta frequência.

As magnitudes dos coeficientes de alta frequência, geralmente, são próximas ao valor 0 (zero) o que permite que estes possam ser eliminados durante a quantização.

Devido a sua alta capacidade de concentração de energia, o que implica um número menor de coeficientes em relação a outras transformadas, a DCT tem como principal aplicação a compressão de sinais (OPPENHEIM, 1998).

## 5. CONSIDERAÇÕES FINAIS

Mediante o cenário de convergência tecnológica, em que o tráfego de imagens e vídeo entre os equipamentos multimídia já é uma realidade surge o problema de como possibilitar maior fluxo de dados sem sobrecarregar o canal de transmissão. Além disso, a migração do sistema de TV analógico para o digital, no Brasil, exige o desenvolvimento de alternativas que otimizem o uso do canal de transmissão já existente para permitir multiprogramação e interatividade ao telespectador.

A fim de atender às necessidades de otimizar os processos de transmissão e armazenamento, a compressão é um método eficiente para alcançar tal objetivo. Para isso elimina as redundâncias de uma imagem que são irrelevantes para os observadores humanos.

Neste artigo viu-se os principais algoritmos empregados tanto na compressão de imagens estáticas quanto nos codificadores de TV Digital: Huffman, aritmética e DCT. Em geral, para aumentar a taxa de compressão combinam-se diferentes tipos de codificação em um mesmo codificador.

As codificações de Huffman e aritmética são muito usadas para compressão sem perdas. A primeira não exige grande complexidade computacional e seu uso é preferível em fontes que não tenham uma alta variação de ocorrência. No caso da codificação aritmética, uma palavra-código representa uma sequência inteira de símbolos de entrada e em relação a Huffman é preferível como codificação adaptativa.

A DCT embora não seja diretamente responsável pela redução das redundâncias é muito utilizada para compressão por perdas devido ao seu alto poder de concentração de energia em poucos coeficientes. Dessa forma possibilita a eliminação de um número maior de coeficientes, com valores próximos de zero, no processo de quantização.

Dessa maneira, a utilização dos diferentes tipos de codificação depende do tipo de aplicação desejada e podem ser utilizados em um mesmo codificador como forma de obter taxas de compressão mais elevadas ao suprir as deficiências das outras.

## REFERÊNCIAS

BATISTA, L. V. **Notas de Aula da Disciplina Introdução à Teoria da Informação**. 2005.

\_\_\_\_\_. **Notas de Aula da Disciplina Introdução ao Processamento Digital de Imagens**. 2005a.

FERRAZ, M. C. **Codificação de Imagens**. 1998. 232 f. Tese (Mestrado em Ciências) – Instituto de Matemática Pura e Aplicada, Rio de Janeiro, 1998.

GONZALEZ, Rafael C.; WOODS, Richard E. **Processamento de Imagens Digitais**. 1. ed. São Paulo: Edgard Blücher, 2000.

HUNT, B. R. Digital Image Processing. In: OPPENHEIM, A. V. **Applications of Digital Signal Processing**. Englewood Cliffs: Prentice Hall, 1978. p. 169-187.

KARAM, L. J. Lossless Coding. In: BOVICK, A. C. **Handbook of Image and Video Processing**. New York: Academic Press, 2000. p. 461-470.

OPPENHEIM, A. V.; SCHAFER, R. W.; BUCK, J. R. **Discrete-Time Signal Processing**. 2. ed. Upper Saddle River: Prentice Hall, 1998. p. 589-599.

SOUZA, M. S. **Codificação de Imagem utilizando Turbo Quantização Codificada por Treliças**. 2005. 98 f. Tese (Mestrado em Ciências em Engenharia Elétrica) – Coordenação dos Programas de Pós-Graduação de Engenharia, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2005.

TUDOR, P. N. **MPEG-2 Video Compression**. *IEE J Langham Thompson Prize: Eletronics & Communication Engineering Journal*, dez. 1995. Disponível em: <[http://www.bbc.co.uk/rd/pubs/papers/paper\\_14/paper\\_14.shtml](http://www.bbc.co.uk/rd/pubs/papers/paper_14/paper_14.shtml)>. Acesso em: 15 nov. 2006.

WALDMAN, H.; YACOUR, M. D. **Telecomunicações: Princípios e Tendências**. 4. ed. São Paulo: Érica, 2000. p. 83-98. Série Reuniversidade.