

## Capítulo 3

# Compressão de Dados Multimídia

Como visto no capítulo anterior, áudios, imagens e vídeos digitais são codificados geralmente em uma grande quantidade de dados, e podem gerar grandes taxas de bits na transmissão. Assim, a compressão de dados é essencial para que as informações ocupem espaço aceitável em disco e que possam ser transmitidas via rede em taxas razoáveis de transmissão. Não existiria multimídia hoje sem o drástico progresso que ocorreu nos últimos anos em algoritmos de compressão e suas implementações [Fluckiger, 95].

## 3.1 A Necessidade da Compressão

Compressão de dados é uma forma de codificar certo conjunto de informações de maneira que o código gerado seja menor que a fonte. O uso de técnicas de compressão é claramente essencial para aplicações multimídia. As razões são as seguintes [Furht, 94]:

- O grande requisito de armazenamento de dados multimídia.
- A velocidade relativamente lenta dos dispositivos de armazenamento que não podem apresentar dados multimídia (principalmente vídeo) em tempo-real.
- A largura de banda da rede que não permite a transmissão de vídeo em tempo-real.

### Requisitos de armazenamento

Caso nenhuma técnica de compressão for utilizada, 80 MB do disco de um PC seriam ocupados por 8 minutos de som estereofônico qualidade CD, ou 3,5 segundos de vídeo de qualidade TV. No caso dos CD-ROMs, eles podem ocupar 72 minutos de música de alta fidelidade, mas apenas 30 segundos de vídeo de qualidade TV.

Uma aplicação multimídia típica contém mais de 30 minutos de vídeo, 2000 imagens e 40 minutos de som estéreo. Sendo assim, a aplicação necessitaria, caso não fosse aplicada nenhuma compressão, de aproximadamente 50 GB para armazenar o vídeo, 15 GB para armazenar as imagens e 0,4 GB para armazenar o áudio. O que significa um total de 65,4 GB de armazenamento no disco. Assim é extremamente necessária a utilização de técnicas de compressão de dados multimídia para viabilizar o armazenamento destas informações.

### Velocidade de transferência dos dispositivos de armazenamento

Mesmo que tenhamos enorme capacidade de armazenamento, nós não seríamos capazes de apresentar um vídeo em tempo-real devido à taxa de bits insuficiente dos dispositivos de armazenamento. Por exemplo, um dispositivo de armazenamento deveria ter uma taxa de 30 MBps para apresentar um vídeo em tempo real com um quadro de 620x560 píxeis a 24 bits por pixel de 30 fps. A tecnologia de leitores de CD-ROM de hoje não fornece largura de banda suficiente para esta qualidade. Por exemplo, para um leitor de 52x a largura de banda nominal é de 7,2 MBps. No estado atual da tecnologia de armazenamento, a única solução é compactar o dado antes de armazenar e descompactar ele antes da apresentação.

### Largura de banda da rede

A taxa de bits gerada por áudio, vídeo e imagem depende dos parâmetros de digitalização. O quadro abaixo resume as taxas geradas por diversos tipos de aplicações.

Aplicações	Taxa de bits (Kbps)
CD-Audio	1.411
DAT	1.536
Telefone Digital	64
Radio digital, long play DAT	1.024
DVD	249.600
SDTV	486.600
HDTV	2.986.000

Algumas destas aplicações não exigem compressão, como a transmissão de áudio na qualidade telefone digital, que gera apenas 64kbps, aceitável em muitas redes. Mas maior parte das aplicações exigem compressão dos dados multimídia. Por exemplo, caso um som estereofônico de qualidade CD não compactado devesse ser transmitido, a rede deveria suportar uma taxa de 1,4 Mbps. Isto é possível em redes locais (onde a LAN normalmente suporta uma taxa de 100 Mbps, no caso da tecnologia Ethernet), mas em redes de média e longa distância, esta taxa torna-se muitas vezes inviável. No caso de um vídeo de qualidade PAL não compactado é necessário uma taxa de 160 Mbps, isto é incompatível com a maior parte das redes locais e WANs.

Técnicas de compressão modernas de imagem e vídeo reduzem tremendamente os requisitos de armazenamento e, portanto, os requisitos de largura de banda da rede e do dispositivo de armazenamento. Técnicas avançadas podem comprimir uma imagem típica em uma razão variando de 10:1 a 50:1 e para vídeo de até 2000:1.

## 3.2 Princípios de Compressão

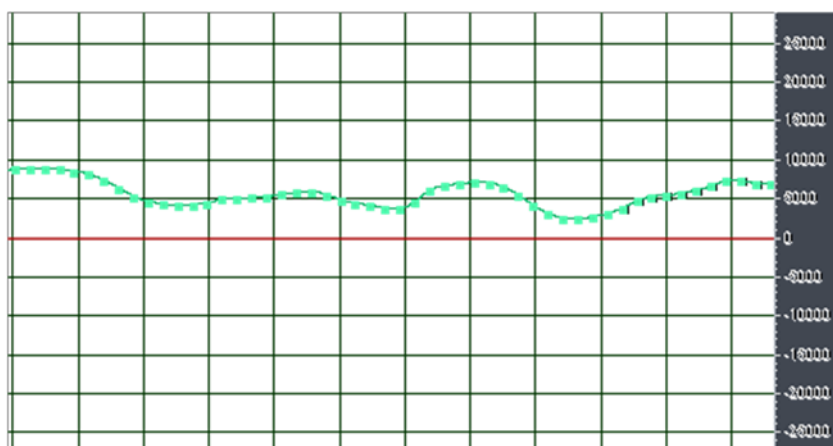
Técnicas de compressão de dados multimídia exploram basicamente dois fatores: a redundância de dados e as propriedades da percepção humana.

### 3.2.1 Redundância de Dados

Recapitulando o capítulo anterior, um áudio digital é uma série de valores amostrados; uma imagem digital é uma matriz de valores amostrados (pixels); e um vídeo digital é uma sequência de imagens apresentadas numa certa taxa. Geralmente amostras subsequentes de áudios e imagens (para vídeo) não são inteiramente diferentes. Valores vizinhos são geralmente de algum modo relacionados. Esta correlação é chamada **redundância**. A remoção desta redundância não altera o significado do dado, existe apenas uma eliminação da replicação de dados.

#### **Redundância em áudio digital**

Em muitos casos, amostragens de áudio adjacentes são similares. A amostra futura não é completamente diferente da passada, o próximo valor pode ser previsto baseado no valor atual (Figura 22). A técnica de compressão que se aproveita desta característica do áudio é chamada de **codificação preditiva**. Técnicas de compressão preditiva são baseadas no fato que nós podemos armazenar a amostra anterior e usar esta para ajudar a construir a próxima amostra.



**Figura 22.** Amostras vizinhas do áudio são semelhantes

No caso da voz digital há outro tipo de redundância: nós não falamos todo o tempo. Entre uma rajada e outra de informação há instantes de silêncio (Figura 23). Este período de silêncio pode ser suprimido sem a perda de informação, sabendo que este período é mantido. Esta técnica de compressão é chamada da **Remoção de silêncio**.

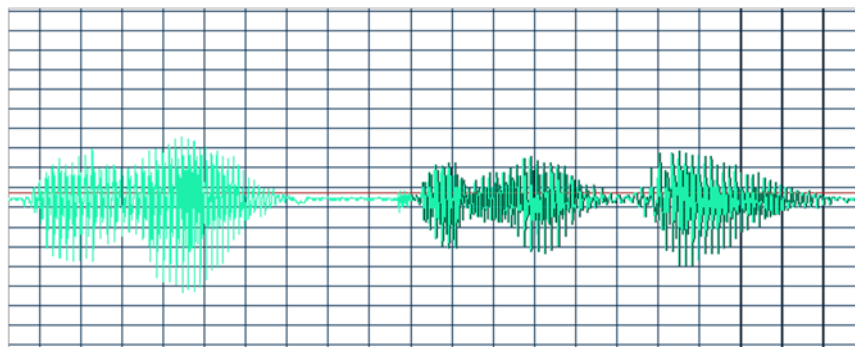


Figura 23. Silêncio entre as palavras “Ola como Vai”

### Redundância em imagem digital

Em imagens digitais, as amostras vizinhas em uma linha de escaneamento e as amostras vizinhas em linhas adjacentes são similares (Figura 24). Esta similaridade é chamada de **Redundância espacial**. Ela pode ser removida, por exemplo, utilizando técnicas de codificação preditiva ou outras.



Figura 24. Píxeis vizinhos de uma imagem

### Redundância em vídeo digital

Vídeo digital é uma sequência de imagens, portanto ele também tem redundância espacial. Além disso, imagens vizinhas em vídeos são geralmente similares (Figura 25). Esta redundância é chamada de **redundância temporal**. Ela pode também ser removida, por exemplo, utilizando técnicas de codificação preditiva.



Figura 25. Quadros Vizinhos de um vídeo

## 3.2.2 Propriedades Percepção Humana

Os usuários finais das aplicações multimídia são geralmente humanos. Humanos podem tolerar alguns erros de informação ou perda sem afetar a efetividade da comunicação. Isto implica que a versão comprimida não necessita representar exatamente a informação original. Isto é bem diferente dos dados alfanuméricos que não se tolera qualquer erro ou perda (por exemplo, se ocorrer uma perda de uma informação em um programa ele pode não funcionar corretamente).

Como os sentidos humanos não são perfeitos, pequenas perdas e erros em áudios e vídeos não são percebidos. Além disso, algumas informações são mais importantes para a percepção humana que outras (por exemplo, no caso de imagens, a intensidade luminosa é mais importante que a cor). Assim, na hora de compactar as informações, dados de imagens, vídeos e sons, que não é “percebível” pela maioria dos humanos pode ser descartada.

### 3.3 Classificação das Técnicas de Compressão

Existem várias técnicas de compressão, elas podem ser classificadas de diversas maneiras: baseadas no algoritmo de compressão e no resultado das técnicas de compressão. Neste documento, elas serão classificadas quanto ao resultado. Esta classificação é mais útil para o usuário final e para desenvolvedores de aplicações multimídia do que a classificação baseada em algoritmo.

#### 3.3.1 Classificação das Técnicas de Compressão

Se a informação, após sua compressão, pode ser exatamente reconstruída a técnica de compressão é dita sem perdas (Figura 26) ou também chamada de codificação por entropia (*Entropy encoding*). Este tipo de compressão trata de cadeias de bits sem levar em conta seu significado [Tanembau, 97]. É uma técnica genérica, sem perda e totalmente reversível, que pode ser aplicada a todos os dados. Esta técnica deve ser utilizada obrigatoriamente para compactar programas e documentos legais ou médicos. Como estas técnicas exploram apenas estatísticas de dados (redundância de dados) e a taxa de compressão é normalmente baixa. Um exemplo deste tipo de compressão é substituir caracteres de espaços ou zeros sucessivos por um flag especial e o número de ocorrências. Algumas técnicas de compressão sem perdas são: codificação aritmética, codificação Huffman e codificação Run-length.

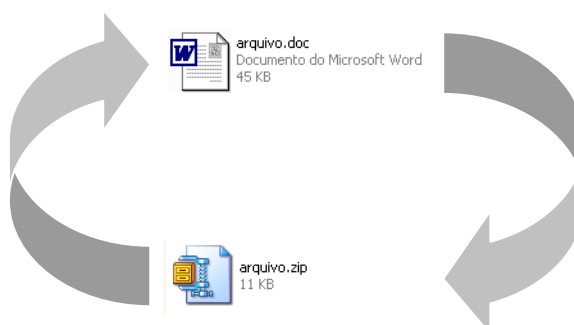


Figura 26. Técnica sem perdas

Técnicas de compressão com perdas são utilizadas para compressão de áudio, imagens e vídeos, onde erros e perdas são toleráveis. Também chamadas de Codificação na origem (*Source coding*), este tipo de técnica processa o dado original distinguindo o dado relevante e o irrelevante. Elas levam em consideração a semântica dos dados. Removendo os dados irrelevantes comprime o dado original. Sendo assim, o dado original será diferente do dado descompactado (Figura 27). Como exemplo de técnicas de compressão da origem, nós temos: DPCM (*Differential pulse code modulation*), DCT (*discrete cosine transform*) e DWT (*Discrete wavelet transform*).



Figura 27. Técnica com perdas

Codificação híbrida é a combinação de técnicas de compressão sem perdas e técnicas de codificação na origem. Normalmente, várias destas duas técnicas são agrupadas para formar uma nova técnica de codificação híbrida. Normalmente, estas técnicas inicialmente eliminam informações irrelevantes para a percepção humana (assim com perda de dados) e sobre os dados restantes é aplicada uma técnica para eliminação da redundância (sem perdas), conforme ilustrado na Figura 28. Como exemplo deste tipo de técnica de compressão, pode-se citar os padrões H.261, H.263, JPEG, MPEG vídeo e áudio.



Figura 28. Técnicas de compressão híbridas

### 3.4 Medição do Desempenho de Compressão

No desenvolvimento de uma aplicação multimídia, os autores devem escolher que técnica de compressão utilizar. Esta escolha geralmente baseia-se nas classificações apresentadas anteriormente, nos parâmetros de desempenho da técnica e nos requisitos da aplicação. Os parâmetros de desempenho mais usados são:

- **Taxa de compressão:** razão entre o tamanho do dado original e o tamanho do dado após a compressão. No caso de técnicas sem perda, quanto maior a taxa de compressão melhor é a técnica de compressão. Para técnicas de compressão com perda deve-se considerar também a qualidade da mídia restituída. Como ilustrado na **Erro! Fonte de referência não encontrada.**, para a imagem Lena de exemplo, a compressão PNG resultou em uma maior taxa de compressão que o GIF.

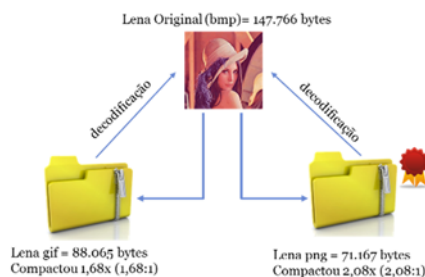


Figura 29. Taxa de Compressão de técnicas sem perda

- **Qualidade da mídia reconstituída:** medida em SNR (Razão Sinal/Ruído). Este parâmetro é aplicável apenas para técnicas com perda. Para a escolha de uma técnica de compressão com perdas, deve-se optar pelo compromisso entre uma alta taxa de compressão e a qualidade desejada para a aplicação em desenvolvimento. Como ilustrado na Figura 30, para selecionar o codec com perdas, é necessário avaliar a taxa de compressão e a qualidade da mídia reconstituída. No caso, a técnica B teve uma menor taxa de compressão, mas a qualidade da mídia reconstituída é melhor.

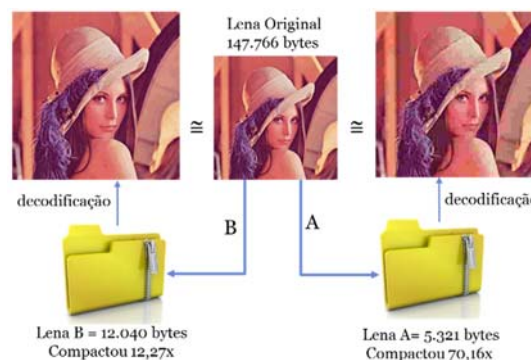


Figura 30. Técnicas de compressão com perdas

- **Complexidade da implementação e velocidade de compressão:** geralmente quanto mais complexa a técnica, menor é a velocidade de compressão. No caso de aplicações tempo-real como videoconferência, como ilustrado na Figura 31, estes parâmetros devem ser considerados. Isto, pois a compressão/descompressão deve ser realizada em tempo-real. No caso de aplicações do



Este algoritmo pode ser facilmente otimizado, por exemplo, pode-se substituir seqüências maiores que um. Para isso, é necessário que o tamanho da seqüência seja codificado ou pode-se usar um caractere especial de fim. Por exemplo:

- entrada: UFYUGDUFHUFHUFHUFHUFHBF
- saída: UFYUGD!5UFH\$BFD

Esta técnica é aplicada em formatos padrões, como PCX, BMP (RLE) e Photoshop. O BMP RLE permite compactar imagens no formato Windows Bitmap, sendo que a imagem deve ser baseada em paleta de 256 cores.

Este método só traz ganhos relevantes se houver grandes agrupamentos de símbolos iguais. As principais aplicações do método de Run-Length são em imagens binárias, imagens com grandes espaços envolvendo uma só cor e em imagens geradas por computador, onde os dados estão agrupados de forma mais geometricamente definida.

### 3.5.2 Codificação de Huffman

Neste método de compressão, são atribuídos menos bits a símbolos que aparecem mais frequentemente e mais bits para símbolos que aparecem menos. Assim, os tamanhos em bits dos caracteres codificados serão diferentes.

Codificação de Huffman é um exemplo de técnica de codificação estatística, que diz respeito ao uso de um código curto para representar símbolos comuns, e códigos longos para representar símbolos pouco frequentes. Esse é o princípio do código Morse, em que E é • e Q é --•-, e assim por diante. Esta técnica é usada, por exemplo, em uma das etapas da compressão JPEG.

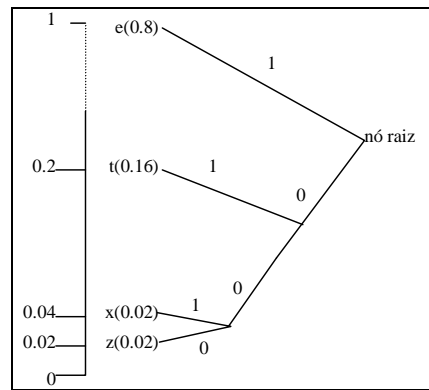
A	•-	M	--	Y	-•••	6	-••••
B	-•••	N	-•	Z	-•••	7	-•••••
C	-•••	O	---	Ä	•-•-	8	•••••
D	-••	P	••••	Ö	••••	9	••••••
E	•	Q	••••	Ü	•••••	.	•••••••
F	••••	R	•••	Ch	••••••	,	••••••••
G	-••	S	•••	0	•••••••	?	••••••••
H	••••	T	-•	1	••••••••	!	•••••••••
I	••	U	•••	2	•••••••••	:	••••••••••
J	•••••	V	••••	3	••••••••••	"	•••••••••••
K	-••	W	•••	4	••••••••••	'	•••••••••••
L	-••	X	-••	5	••••••••••	=	•••••••••••

Nós usaremos um exemplo para mostrar como a codificação de Huffman funciona. Suponha que nós temos um arquivo contendo 1000 caracteres, que são e, t, x e z. A probabilidade de ocorrência de e, t, x, e z são 0.8, 0.16, 0.02, e 0.02 respectivamente. Em um método de codificação normal, nós necessitamos 2 bits para representar cada um dos quatro caracteres. Assim, nós necessitamos de 2000 bits para representar o arquivo. Usando a codificação de Huffman, nós podemos usar quantidades de bits diferentes para representar estes caracteres. Nós usamos bit 1 para representar e, 01 para representar t, 001 para representar x e 000 para representar z. Neste caso, o número total de bits necessários para representar o arquivo é  $1000 \cdot (1 \cdot 0.8 + 2 \cdot 0.16 + 3 \cdot 0.02 + 3 \cdot 0.02) = 1240$ . Assim, embora tenhamos utilizado mais bits para representar x e z, desde que seus aparecimentos são mais raros, o número total de bits necessários para o arquivo é menor que o esquema de codificação uniforme. As regras para atribuir bits (códigos) aos símbolos formam o chamado um codebook. Codebooks são normalmente expressos em tabelas, para o exemplo:  $w(e)=1$ ,  $w(t)=01$ ,  $w(x)=001$ ,  $w(z)=000$ .

Agora vamos ver como os códigos Huffman são gerados. O procedimento é o seguinte (Figura 33):

- Coloque de todos os símbolos ao longo de uma linha de probabilidade acumulativa na seguinte ordem: probabilidade dos símbolos aumenta de baixo para cima. Se dois símbolos têm a mesma probabilidade, eles podem ser colocados em qualquer ordem.
- Se Junta os dois símbolos de menor probabilidade a um nó para formar dois ramos na árvore.
- A nova árvore formada é tratada como um símbolo único com a probabilidade igual à soma dos símbolos ramos.
- Repita os passos (b) e (c) até que todos os símbolos sejam inseridos na árvore. O último nó formado se chama o nó raiz.
- Partindo do nó raiz, atribua o bit 1 ao ramo de maior prioridade e bit 0 ao ramo de menor prioridade de cada nó.
- O código para cada símbolo é obtido montando códigos ao longo dos ramos do nó raiz para a posição do símbolo na linha de probabilidade. Por exemplo, lendo do nó raiz ao símbolo x, nós obtemos o código 001 para x.





**Figura 33.** Exemplo de árvore de codificação Huffman

A operação computacional mais custosa na determinação do código Huffman é a adição de floats, mais especificamente, a adição da probabilidade da ocorrência no processo de redução. Isto ocorre no lado do codificador. No lado do decodificador, ele tem que apenas realizar uma simples verificação na tabela. Portanto, o decodificador necessita da tabela Huffman usado no codificador. Esta tabela é parte do fluxo de dados ou já é conhecida pelo decodificador. Em áudio e vídeo, tabelas Huffman padrões são utilizadas com muita frequência, isto é, tabelas são conhecidas pelo codificador e decodificador. A vantagem é a obtenção de uma codificação mais rápida, pois as tabelas não precisam ser calculadas. A desvantagem é que tabelas Huffman padrões obtêm um fator de compressão um pouco menor porque as tabelas não são necessariamente ótimas para o dado a ser codificado. Portanto, métodos de compressão executados em tempo-real usam normalmente tabelas padrões, pois a codificação é mais rápida. Se alta qualidade é necessária e o tempo de codificação não é importante, as tabelas Huffman otimizadas podem ser utilizadas.

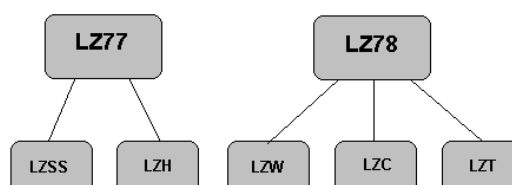
Normalmente nem todos os caracteres tem uma representação codificada na tabela Huffman: apenas aqueles caracteres com alta probabilidade de ocorrência. Todos os outros são codificados diretamente e marcados com um flag especial. Esta técnica é útil quando um número de caracteres diferentes é muito grande, mas apenas alguns deles têm uma alta probabilidade de ocorrência.

### 3.5.3 Codificação de Lempel-Ziv (LZ)

Codificação LZ é baseada na construção de um dicionário de frases (grupos de um ou mais caracteres) a partir do fluxo de entrada. Quando uma nova frase é encontrada, a máquina de compressão adiciona ao dicionário e um token que identifica a posição da frase no dicionário substitui a frase. Se a frase já foi registrada, ela é substituída pelo token de posição no dicionário. Esta técnica é boa para compressão de arquivos de texto, onde temos uma grande repetição de frases. Por exemplo, em português: "ela", "Contudo", ", ", "onde, ", aparecem frequentemente no texto.

LZ são na realidade um conjunto de codificação baseadas na construção de dicionários, que foram inicialmente criados pelos autores Jacob Ziv e Abraham Lempel no final dos anos 70. Muitas variantes com o objetivo de solucionar limitações das versões originais foram propostas.

Como ilustrado na Figura, existem diversas variações de codecs LZ, cada um aprimorando um determinado aspecto do processo de codificação.



**Figura 34.** Codecs LZ

O seguinte exemplo mostrará o poder da codificação LZ. Suponha que temos um arquivo de 10000 caracteres. Se nós representarmos o arquivo usando 8 bits por caractere, o arquivo requer 80000 bits para representá-lo. Usando o algoritmo LZ e assumindo que o arquivo tenha 2000 palavras ou frases das quais 500 são diferentes, então nós necessitamos 9 bits como token para identificar cada palavra ou frase



distinta. Assim, nós precisamos de  $9 \times 2000$  bits para codificar o arquivo. Com isto nós obtemos uma taxa de compressão de 4,4. Na prática, o dicionário armazenando todas as frases únicas deve ser armazenado também, baixando a taxa de compressão obtida.

### Exemplo de Compressão

O funcionamento básico de LZ será ilustrado através da compressão da cadeia de caracteres TOBEORNOTTOBEORTOBEORNOT# (Fonte: <http://en.wikipedia.org/wiki/Lempel%E2%80%93Ziv%E2%80%93Welch>). O caractere # indica o fim da sequência. Este exemplo limita-se a compressão de texto plano usando apenas caracteres maiúsculos (26 letras de A a Z). Para a codificação das 26 letras mais o código de parada # são necessários 5 bits ( $2^5 = 32$  valores possíveis).

- O primeiro passo consiste em iniciar o dicionário com todos os 27 símbolos

Símbolo	Binário	Índice	Símbolo	Binário	Índice
#	00000	0	O	01111	15
A	00001	1	P	10000	16
B	00010	2	Q	10001	17
C	00011	3	R	10010	18
D	00100	4	S	10011	19
E	00101	5	T	10100	20
F	00110	6	U	10101	21
G	00111	7	V	10110	22
H	01000	8	W	10111	23
I	01001	9	X	11000	24
J	01010	10	Y	11001	25
K	01011	11	Z	11010	26
L	01100	12			
M	01101	13			
N	01110	14			

- A codificação é apresentada na tabela abaixo.

Sequencia atual	Próx. caractere	Saída		Dicionário extendido	Comentários
		Código	Bits		
NULL	T				
T	O	20	10100	27: TO	27 = próximo índice disponível
O	B	15	01111	28: OB	
B	E	2	00010	29: BE	
E	O	5	00101	30: EO	
O	R	15	01111	31: OR	
R	N	18	10010	32: RN	32 requer 6 bits, assim as próximas saídas usam 6 bits
N	O	14	001110	33: NO	
O	T	15	001111	34: OT	
T	T	20	010100	35: TT	
TO	B	27	011011	36: TOB	
BE	O	29	011101	37: BEO	
OR	T	31	011111	38: ORT	
TOB	E	36	100100	39: TOBE	
EO	R	30	011110	40: EOR	
RN	O	32	100000	41: RNO	
OT	#	34	100010		# para o algoritmo; envia o caractere; envia sequencia atual
		0	000000		E o código de parada

Em relação a taxa de compressão obtida no exemplo acima, o tamanho não codificado tem um tamanho de 25 símbolos x 5 bits/símbolo = 125 bits. O tamanho do dado codificado terá 6 códigos x 5 bits/código) + (11 códigos x 6 bits/código) = 96 bits. Com isso, o arquivo terá uma redução de 29 bits de 125, reduzindo o tamanho em 22%.

### **Aplicação do LZ**

Os algoritmos LZ podem ser encontrados em diversos compactadores largamente utilizados atualmente:

- UNIX Compression: O algoritmo LZW é usado pelo utilitário “compress” do sistema operativo UNIX.
- GIF (Graphics Interchange Format): Muito similar ao “compress” do UNIX, também usa o algoritmo LZW.
- Protocolo V.42bis (compressão de dados em Modem): Usa uma variante do LZW (LZT).
- O Zip e o gzip usam uma variante do LZ77 combinada com Huffman estático.
- O ARJ usa a codificação de Huffman e o algoritmo LZSS.
- O WINRAR usa o LZ77 e Huffman.
- O WINZIP entre outros algoritmos usa o LZW.

### **Formato de imagem GIF (Graphical Interchange Format)**

Uma das aplicações do LZW é a variação implementada no formato GIF, utilizado no armazenamento de imagens. Atualmente, este é um dos formatos de armazenamento de imagens sem perdas que oferece as melhores taxas de compressão. Sendo sem perdas, o GIF preserva todos os dados visuais na figura: nenhuma informação é descartada ou alterada durante a compressão/descompressão. Com isto, as taxas de compressão não são muito grandes, em geral 4:1.

O formato GIF apenas admite o tratamento de imagens com uma profundidade de cor de até 8 bits/pixel, ou seja, imagens com um máximo de 256 cores. Se possuímos uma imagem true color, com 24 bits/pixel, ao convertermos esta imagem para o formato GIF, estamos perdendo grande parte da informação de cor. Neste caso, verifica-se que a qualidade da imagem obtida é bastante inferior à qualidade da imagem original, devido ao aparecimento de ruído em toda a imagem.

O algoritmo LZW é propriedade da Unisys. No início este algoritmo era de domínio público. No entanto, há relativamente pouco tempo, a Unisys resolveu passar a cobrar uma taxa pela sua utilização. Por este motivo, começou a pensar-se numa alternativa válida ao formato GIF, tendo surgido o formato PNG (Portable Network Graphics). PNG suporta até true color 48 bits por pixel ou 16 bits por pixel para escalas de cinza e não suporta animação. PNG usa os algoritmos de compressão Lempel-Ziv 77 (LZ77) e de Huffman.

## **3.6 Técnicas de Compressão de Vídeo e Imagem**

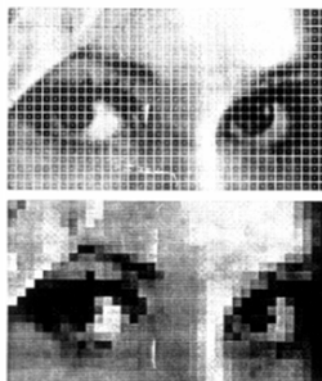
Imagens e vídeos digitais puros são codificadas em PCM e eles são representados por vetores bi-dimensionais (possivelmente tridimensionais para imagens coloridas) de pixels. Reduzindo o conjunto de dados necessário à reprodução de imagens e vídeos (compressão) reduz requisitos de armazenamento, aumenta a velocidade de acesso, e é a única forma de obter vídeo digital em computadores pessoais.

Técnicas de compressão de imagens e vídeos digitais baseiam-se na alta redundância espacial e temporal destas mídias. No caso das imagens, geralmente certas áreas das figuras são uniformemente coloridas ou altamente correlatas (podendo formar padrões). Isto é chamado de redundância espacial ou correlação espacial. Esta redundância pode ser removida tanto quanto possível para certa qualidade de apresentação. No caso do vídeo, além da redundância espacial nós temos a redundância temporal ou correlação temporal, isto, pois geralmente não existem grandes diferenças entre quadros de um vídeo. Para a compressão de vídeo, tanto a redundância temporal quanto a redundância espacial pode ser removida e uma alta taxa de compressão pode ser obtida.

A seguir são apresentadas algumas técnicas de codificação de vídeo.

### 3.6.1 Técnica de Redução da Resolução Geométrica

Uma técnica de compressão é simples, ela consiste em reduzir a resolução das imagens (e resolução de quadros de vídeo). Esta técnica é atrativa, pois ela é extremamente simples, porém a taxa de compressão não é elevada e existe uma redução de qualidade das imagens: a dimensão espacial do píxel será reduzida (se visualizada a imagem no tamanho original (ver imagem ao lado). Por exemplo, em imagens coloridas com 800x600 e 24 bits por píxel, ou seja 1,37 MB, poderiam ser reduzidas para 400x300 bits e 24 bits por píxel, perfazendo 351 KB. A Figura 35 apresenta a redução da resolução espacial de uma imagem.



**Figura 35.** Redução Geométrica de Imagem

### 3.6.2 Técnica de Truncagem

Outra técnica de compressão é simples, ela consiste em truncar dados arbitrariamente (baixando o número de bits por píxel). Isto é feito pela eliminação dos bits menos significativos de cada píxel. Esta técnica é atrativa, pois ela é extremamente simples. Por exemplo, imagens coloridas com 24 bits por píxel poderiam ser reduzidas para 8 bits baseada em paleta (Figura 36).



Imagem 24 bits por píxel      Imagem 256 cores

**Figura 36.** Redução do número de cores da imagem

### 3.6.3 Codificação de Sub-Amostragem Espacial e Temporal

Nesta técnica (também chamada de técnica interpolativa), na codificação um píxel entre alguns píxeis e um quadro entre alguns quadros são selecionados e transmitidos. No decodificador, os píxeis e quadros faltantes são interpolados, baseado nas píxeis e quadros recebidos, para gerar uma imagem ou uma sequência de vídeo de baixa resolução. Como alternativa, o decodificador pode simplesmente decodificar e apresentar as imagens espacialmente sub-amostradas.

Esta técnica é utilizada em aplicações que não necessitam de alta resolução, como na videofonia. Ela é muito simples mais pouco eficiente. Por exemplo, quando um píxel entre quatro ou um quadro entre quatro são transmitidos, uma taxa de compressão 8:1 é obtida. Para aumentar a taxa de compressão, dados de imagens sub-amostradas temporalmente e espacialmente podem ser adicionalmente compactados usando outras técnicas.

### 3.6.4 Codificação Preditiva

Similar à codificação preditiva de áudio, esta técnica considera a estatística do sinal da imagem e o sistema visual humano para a compressão de áudios e vídeos. Em geral, os valores amostrados de elementos vizinhos de imagens são correlacionados e quadros de um vídeo têm correlação temporal entre os elementos de imagens em quadros sucessivos.

Correlação ou dependência estatisticamente linear indica que uma previsão linear dos valores amostrados pode ser feita. Esta previsão é baseada nos valores amostrados dos elementos vizinhos da figura. Geralmente esta previsão resulta em um erro de predição que tem uma variância menor que os valores amostrados.

Algoritmos de predição unidimensional usam correlação de elementos de imagem adjacentes dentro de uma linha. Outros esquemas mais complexos também exploram correlações linha-a-linha e quadro-a-quadro e são chamados de predição bidimensional e tridimensional, respectivamente.

### DPCM

A forma mais simples de compressão preditiva opera ao nível de pixel com uma técnica chamada de PCM diferencial (DPCM). No DPCM, nós comparamos pixels adjacentes e apenas as diferenças são transmitidas. Como pixels adjacentes são geralmente similares, o valor da diferença tem uma alta probabilidade de ser menor que o valor do novo pixel e ele pode ser expresso com uma quantidade menor de bits. Na descompressão, a informação da diferença é usada para modificar o pixel anterior para obter o novo pixel.

Veja na **Erro! Fonte de referência não encontrada.** alguns preditores típicos para imagens:

$$\begin{aligned} \hat{s}_n &= 0.97s_{n-1} && \text{Preditor de 1ª ordem, 1D} \\ \hat{s}_{m,n} &= 0.48s_{m,n-1} + 0.48s_{m-1,n} && \text{Preditor de 2ª ordem, 2D} \\ \hat{s}_{m,n} &= 0.8s_{m,n-1} - 0.62s_{m-1,n-1} + 0.8s_{m-1,n} && \text{Preditor de 3ª ordem, 2D} \end{aligned}$$

$s_{m-1,n-1}$	$s_{m,n-1}$	
$s_{m-1,n}$	$s_{m,n}$	

**Figura 37.** Preditores típicos para Imagem

Veja na Figura 38 um exemplo de compressão de uma imagem original, usando para a primeira fila e primeira coluna o preditor de 1ª ordem, e para as outras linhas e colunas o preditor de 3ª ordem. A Saída DPCM é calculada subtraindo a saída prevista com os valores originais.

$\begin{bmatrix} 20 & 21 & 22 & 21 \\ 18 & 19 & 20 & 19 \\ 19 & 15 & 14 & 16 \\ 17 & 16 & 15 & 13 \end{bmatrix}$	$\begin{bmatrix} 20 & 19.4 & 20.37 & 21.34 \\ 19.4 & 18.8 & 19.78 & 19.16 \\ 17.46 & 19.24 & 16.22 & 14.00 \\ 18.43 & 13.82 & 14.70 & 16.2 \end{bmatrix}$	$\begin{bmatrix} X & 1.6 & 1.63 & -0.34 \\ -1.4 & 0.20 & 0.22 & -0.16 \\ 1.54 & -4.24 & -2.22 & 2.00 \\ -1.43 & 2.18 & 0.30 & -3.12 \end{bmatrix}$
Original	Saída prevista	Saída DPCM

**Figura 38.** Codificação DPCM de uma imagem

Em suma, a saída DPCM contém com apenas o erro de predição. Veja na Figura 39 uma imagem original e a imagem com os erros de previsão. Se os pixels tiverem valores muito próximos, pode-se usar um número menor de bits para armazenar o erro de predição do que aquele usado para codificar o valor absoluto.



**Figura 39.** Codificação DPCM

### ADPCM

Há muitas maneiras de se implementar a técnica DPCM adaptativa (ADPCM), a mais comum é variar o tamanho de passo representado pelos bits diferenças. Por exemplo, se um passo preto-para-branco for detectado, pode-se aumentar o passo de quantificação antes do passo preto-para-branco chegar.

Tanto DPCM e ADPCM não são muito utilizados para compressão de vídeo, eles causam uma sobrecarga proibitiva quando se tenta uma taxa de compressão superior a 2:1. De qualquer modo, estas técnicas são utilizadas em conjunto com outras técnicas mais poderosas em algoritmos de compressão mais complexos.

### 3.6.5 Preenchimento Condicional

Neste esquema, a imagem é segmentada em áreas estacionárias e com movimento e apenas são transmitidos dados de áreas em movimento. Um detector de movimento localiza diferenças interquadros significantes.

Uma forma particular de DPCM onde se envia o erro de predição se este for superior a um dado limite. Veja na Figura 40 um exemplo de cálculo do erro



Quadro Preditor – Quadro Atual – Diferença

**Figura 40.** Visualizando o erro DPCM

### 3.6.6 Estimativa e Compensação de Movimento

Este esquema explora redundância temporal em vídeos. Animação de imagens usualmente implica que pixels na imagem anterior estão em diferentes posições que na imagem atual. Como ilustrado na **Erro! Fonte de referência não encontrada.** Nesta técnica, cada imagem é dividida em blocos de tamanho fixo. Um casamento para cada bloco é procurado na imagem anterior. O deslocamento entre estes dois blocos é chamado vetor de movimento. Uma diferença de blocos é obtida calculando diferenças pixel a pixel. O vetor de movimento e a diferença de bloco são codificados e transmitidos. É normalmente muito menor e mais eficiente transmitir o vetor de movimento mais a diferença que transmitir a descrição do bloco atual.



**Figura 41.** Estimativa e Compensação de Movimento

### 3.6.7 Técnica Transform Coding

Aqui o termo *transform* é o processo que converte um bloco de dados em uma forma substituta que é mais conveniente para algum propósito particular. Ele é um processo reversível, o dado original pode ser restaurado. No caso de compressão de imagem e vídeos, um bloco de dados é um grupo de pixels (usualmente um vetor bidimensional de pixels para uma imagem). Uma transformação é feita para criar uma forma substituta que pode ser transmitida ou armazenada. No tempo de descompressão, uma transformação inversa é executada sob o dado para reproduzir o pixel original.

Nesta técnica, uma imagem é dividida em blocos ou sub-imagens. Uma transformada matemática unitária é aplicada a cada sub-imagem que transforma a sub-imagem do domínio do espaço para o domínio da frequência. Exemplos de transformadas aplicáveis são: transformada de Karhunen-Loeve (KLT), transformada discreta de co-seno (DCT), transformada Walsh-Hadamard (WHT) e transformada de

Fourier Discreta (DFT). Como resultado, muito da energia da imagem é concentrada em poucas amostragens na área de baixa frequência. Note que dados não são perdidos aplicando uma transformada na imagem. Se os dados no domínio espacial são altamente correlatos, o dado resultante no domínio da frequência estará em uma forma desejável para a redução de dados com técnicas de compactação sem perdas.

### 3.7 Padrões de Compressão Multimídia

Existem várias técnicas e produtos para compressão de imagens e vídeos. A definição e a utilização de padrões internacionais de compressão de dados multimídia promovem a compatibilidade entre diferentes produtos. Os cinco principais padrões de compressão audiovisuais mais utilizados são:

- ISO JBIG para compressão sem perda de imagens bi-níveis (1 bit/pixel) para transmissão fac-símile
- ISO/IEC JPEG para compressão de imagens;
- ITU-TS H.261 para videofonia e aplicações de teleconferências na taxa de bits múltiplos de 64 Kbps;
- ITU-TS H.263 para aplicações de videofonia na taxa abaixo de 64 Kbps;
- ISO MPEG para compressão de vídeo e áudio associado;

### 3.8 Padrões de Compressão de Imagens

#### 3.8.1 Padrão de Compressão de Imagens JPEG

O padrão JPEG (*Joint Photographic Expert Group*) foi desenvolvido, em 1992, pelo ISO/IEC em colaboração com a ITU-TS. Ele representa uma das melhores tecnologias de compressão de imagem. Dependendo da imagem, taxas de compressão podem alcançar 25 para 1 sem degradações notáveis. Em média, JPEG obtém uma taxa de compressão de 15:1.

O JPEG é usado hoje em dia por muitas aplicações envolvendo imagens. Ele pode ser implementado em software e hardware. Embora este padrão tenha sido projetado inicialmente para imagens, codificação e decodificação JPEG tempo-real tem sido implementada para vídeo. Esta aplicação é chamada de *Motion JPEG* (MJPEG).

Um dos objetivos do JPEG é cobrir uma grande faixa de qualidades de imagens e permitir especificar o comportamento do codificador a partir de parâmetros. Assim, a relação entre a taxa de compressão e a qualidade resultante pode ser selecionada pelo usuário ou pelo software aplicativo que usa JPEG. Outro objetivo foi permitir que a compressão fosse possível em uma grande diversidade de computadores com diferentes poderes de processamento. Esta meta levou à definição de quatro modos de operação:

- Codificação sequencial: modo com perdas baseada em DCT. Cada componente de imagem é codificado em uma única varredura da esquerda para direita e de cima para baixo. Este modo é chamado de baseline e deve ser suportado por toda implementação JPEG.
- Codificação progressiva: com perdas baseada em DCT expandido. Fornece avanços ao modo baseline. Uma expansão importante é a codificação progressiva (varreduras sucessivas), em que a imagem é codificada em varreduras múltiplas para produzir uma imagem de maneira rápida e rústica quando a largura de banda de transmissão é baixa.
- Codificação sem perda: o processo de compressão é reversível, assim a reprodução é exata. Este modo sem perda não permite a obtenção de altos fatores de compressão, mas muitas aplicações necessitam armazenamento sem perda de imagens, como fotografias de raio X. Existem algumas variações: o original, que foi normalizado em 1992, que se baseia no DPCM e o uso da condificação por entropia (de Huffman ou aritmética), ilustrado na Figura 42; o JPEG-LS, que deverá deixar obsoleto o formato JPEG “lossless” original, que utiliza a codificação de Golomb-Rice e RLE; e o JPEG 200, que utiliza a técnica de compressão wavelets.
- Codificação hierárquica: a imagem é codificada em resoluções múltiplas. Neste esquema nós temos uma codificação piramidal de uma imagem em resoluções espaciais múltiplas. Neste formato a imagem é comprimida em múltiplas resoluções (vários tamanhos), sendo que a de menor resolução (menor tamanho) é codificada e transmitida primeiro e na sequência são compactadas e transmitidas as de melhores resoluções (as de maiores tamanhos) (CONCOLATO,2000) em ordem crescente de tamanho, e por fim é enviada a imagem completa em

seu tamanho original. Os elementos de imagem das resoluções já recebidas são utilizados na próxima resolução, diminuindo desta forma o tamanho do arquivo.

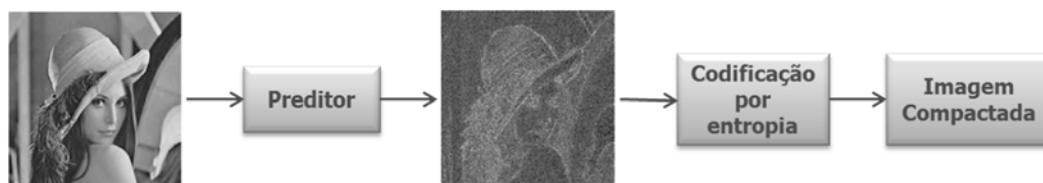


Figura 42. JPEG sem perdas

### Algoritmo de compressão JPEG (Codificação Sequencia e Progressiva)

O algoritmo é composto de uma série de etapas, apresentadas na Figura 43.



Figura 43. Etapas do Algoritmo JPEG (Sequencial, Progressiva)

Em uma primeira etapa do algoritmo JPEG, ocorre a transformação do espaço de cores. Nela, os componentes "RGB" da imagem são convertidos para componentes de luminância ("Y") e crominância ("Cr" e "Cb"). A luminância é uma escala de representação numérica do cinza, enquanto a crominância são duas escalas numéricas, que juntas representam as cores. A escala de luminância, por convenção (ITU-R 601), é quantizada com valores entre "16" e "235", sendo que, o valor "16" representa o negro absoluto e o valor "235" representa o branco absoluto. As duas escalas de crominância contêm valores que variam entre "16" e "240" (com 128 como valor central). Os valores não utilizados, dos "256" possíveis, servem como códigos de controle (MELO,1999).

As matrizes de conversão entre os componentes "RGB" e os componentes "YUV" para o sistema PAL (*Phase Alternate Line* sistema de TV a cores utilizado na Europa, Brasil e Argentina) são:

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} 0.2990 & 0.5870 & 0.1140 \\ -0.1687 & -0.3313 & 0.5000 \\ 0.5000 & -0.4187 & -0.0813 \end{bmatrix} * \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1,00000 & 0,00000 & 1,40200 \\ 1,00000 & -0,34414 & -0,71414 \\ 1,00000 & 1,77200 & 0,00000 \end{bmatrix} * \begin{bmatrix} Y \\ U \\ V \end{bmatrix}$$

A conversão do espaço de cores YCbCr permite uma maior compressão sem um efeito significativo na qualidade da imagem percebida.

A segunda etapa do jpeg é a subamostragem, onde é feita uma redução da resolução das matrizes YCbCr. As taxas de subamostragem que são normalmente aplicados no JPEG são 4:4:4 (sem subamostragem), 4:2:2 onde as matrizes de crominância são reduzidas na taxa de 2:1 horizontalmente (cada duas linhas é convertida em uma), e mais comumente 4:2:0 (redução do fator 2 nas direções horizontais e verticais). A matriz de luminância não é reduzida, pois o olho humano é mais sensível à luminância (tonalidade de cinza) do que à crominância (tonalidades das cores), o que permite maior taxa de perda de crominância



sem que esta perda seja percebida pelo espectador (Li, Ze-Nian; Zhong, Willian; 2000). No resto do processo de compressão, Y, Cb e Cr são processadas separadamente de maneira muito similar.

Na terceira etapa do algoritmo JPEG ocorre a transformada discreta de cosseno (DCT). O algoritmo JPEG decompõe a imagem de entrada em blocos fonte de 8x8 pixels e então transforma estes blocos no domínio da frequência usando a DCT. O DCT efetua uma separação dos componentes de baixa e de alta frequência presentes numa imagem, permitindo que se faça uma seleção destas últimas, de acordo com a qualidade pretendida para a imagem comprimida.

O algoritmo JPEG decompõe a imagem de entrada em blocos fonte de 8x8 pixels e então transforma estes blocos no domínio da frequência usando a transformada discreta de co-seno (DCT). O DCT efetua uma separação dos componentes de baixa e de alta frequência presentes numa imagem, permitindo que se faça uma seleção destas últimas, de acordo com a qualidade pretendida para a imagem comprimida.

O formato JPEG consegue alcançar boas taxas de compressão à custa da exploração das limitações da visão humana, a qual apresenta sensibilidades diferentes relativamente às componentes de frequência presentes numa dada imagem. Como a visão humana é menos sensível às componentes de alta frequência do que às de baixa frequência, aquelas podem ser desprezadas sem que resultem grandes alterações no conteúdo dessa mesma imagem. O JPEG é parametrizável neste sentido, quanto maior é a compressão escolhida, menor é o número de componentes de alta frequência desprezado.

Como ilustrado na Figura 44, o sinal discreto de 64 pontos (um para cada pixel) transformado é uma função de duas dimensões espaciais, x e y. Estes componentes são chamados coeficientes DCT ou frequências espaciais. Para um bloco de imagem 8x8 típico, maior parte dos coeficientes DCT tem valores zero ou muito próximo de zero e não necessitam ser codificados. No domínio da frequência, as mudanças abruptas que acontecem nos contornos de uma figura estão concentradas nas frequências mais altas. Portanto, uma imagem com poucos contornos deve concentrar seus coeficientes nas frequências baixas. Note que os coeficientes das frequências altas são menos importantes e perdas nesses coeficientes podem diminuir um pouco a nitidez da imagem, mas para muitas aplicações isto pode ser aceitável.



Figura 44. Coeficientes DCT

No próximo passo, todos os coeficientes DCT são quantificados usando valores de quantificação especificados em uma tabela de quantificação. Quantificação prioriza a baixa frequência, que são os coeficientes DCT mais próximos do canto superior esquerdo na matriz da Figura 45. Os coeficientes gerados são quantizados de forma diferenciada, usando uma maior precisão para as frequências mais baixas. Esta quantificação reduz a amplitude dos coeficientes que contribuem pouco ou nada para a qualidade da imagem, aumentando assim o número de coeficientes de valor zero. O fator de quantificação (Qc) controla a qualidade final da imagem, que é definido pela qualidade selecionada pelo usuário.

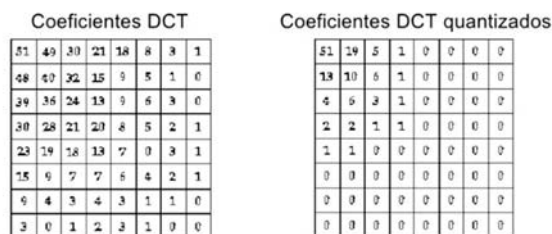
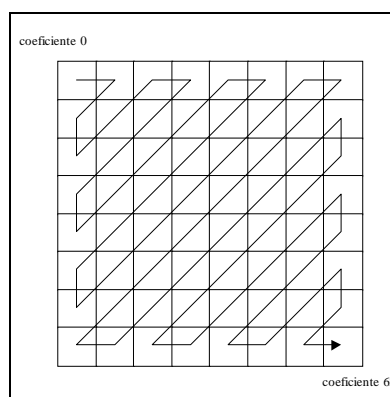


Figura 45. Quantização dos coeficientes DCT

Após a quantificação, os coeficientes DCT são ordenados em uma sequência zig-zag, mostrada na Figura 46, para obter uma sequência unidimensional de dados para ser usado na codificação por entropia. O propósito do escaneamento zig-zag é ordenar os coeficientes em uma ordem crescente de frequências

espectral: os coeficientes de alta frequência (no canto direito inferior) têm valores mais próximos a zero que os coeficientes de baixa frequência. Isto leva a uma maior eficiência da codificação por entropia.



**Figura 46.** Ordenamento zig-zag dos coeficientes DCT

Finalmente, o último passo do JPEG é a codificação por entropia. O padrão JPEG define dois métodos de codificação por entropia: codificação de Huffman e codificação aritmética. Esta etapa fornece uma compressão adicional. Codificação aritmética é normalmente 10% mais eficiente que a codificação de Huffman. Apenas a codificação Huffman é especificada no modo baseline. Para as demais, as duas técnicas podem ser usadas.

### ***Taxas de compressão obtidas***

Como já referimos, quanto maior for a taxa de compressão, maior será o número de componentes de alta frequência desprezados. Para obtermos taxas de compressão muito elevadas, temos que deitar fora um número bastante significativo de componentes de alta frequência. Isso leva o aparecimento do efeito de bloco, ou seja, perda de definição nos contornos das imagens. Geralmente obtêm-se bons resultados com a utilização de taxas de compressão da ordem dos 10% a 50%. Comparando as taxas de compressão com a qualidade de imagem obtida, podemos afirmar que:

- Taxas de compressão de 10:1 a 20:1 – Alta qualidade de imagem;
- Taxas de compressão de 30:1 a 50:1 – Média qualidade de imagem;
- Taxas de compressão de 60:1 a 100:1 – Fraca qualidade de imagem.

### ***JPEG é para imagens fotográficas***

Este formato apresenta ótimas taxas de compressão para imagens fotográficas naturais multi-tonais, diminuindo consideravelmente quando aplicado a imagens gráficas com contornos e áreas bem definidas de cor ou a imagens com texto, como é o caso dos logotipos. O JPEG introduz ruído nas zonas de imagem compostas por cores sólidas, o qual pode distorcer o aspecto geral da imagem. Comparado ao GIF, verifica-se que a imagem GIF comprime mais eficazmente que a JPEG e que apresenta uma melhor definição dos contornos do texto.

## **3.9 Técnicas de Compressão de Áudio Digital**

### **3.9.1 Codificação Preditiva**

Na codificação preditiva, em vez de se transmitir uma amostra, a diferença entre uma previsão do valor da amostra e do valor real é codificada e transmitida. Esta diferença é chamada de erro de previsão. Se esta diferença é quantificada e codificada, o esquema de codificação é chamado de PCM diferencial (DPCM), também chamado de modulação delta. Desde que o valor previsto por cada amostra é calculado apenas a partir do passado do sinal codificado, o valor previsto é disponível tanto para o codificador quanto para o decodificador. No momento da decodificação, o erro de previsão é adicionado ao valor previsto da amostra.

A eficiência desta técnica é obtida explorando o fato que valores de amostras vizinhas são correlacionados e o erro de previsão será menor que o valor original da amostra. Em muitos casos, poucos bits por amostra necessitam ser transmitidos. Por exemplo, compactando um fluxo de áudio de 16 bits, utilizando 4 bits

para representar o erro de predição, a qualidade pode ser mantida se o erro de predição for sempre menor que 16 passos de quantificação original.

A modulação delta apresenta alguns problemas. O variante mais prático é chamado de DPCM adaptativo (ADPCM). Para manipular sinais que mudam rapidamente tão bem quanto sinais que mudam lentamente, o tamanho passo de quantificação aumenta com o aumento da variação do sinal. Então se a forma de onda está trocando rapidamente, grandes passos de quantificação são utilizados e vice-versa. Este é o método usado em CD-I (compact disc-interactive).

Esta técnica reduz a taxa de bits de áudio de alta qualidade de 1,4 Mbps para 32 kbps.

### 3.9.2 Recomendações ITU-TS para codificação de voz

A ITU-TS (*International Telecommunication Union - Telecommunications Sector*) recomenda uma série de esquemas de compressão de voz. Essas recomendações são resumidas na tabela abaixo. Esta tabela apresenta a técnica de compressão utilizada pela recomendação, a largura de banda de voz e a taxa de amostragem consideradas na digitalização da voz, e a taxa de bits do fluxo de voz compactado gerada ao nível de codec. Note que esta taxa de bits vai aumentar até alcançar o nível de rede devido às diversas sobrecargas de protocolo.

Recomend.	Técnica de Compressão	Largura de banda da voz (kHz)	Taxa de amostragem (kHz)	Taxa de bit compactado (kbps)	Duração do payload (ms)	Tamanho do payload (bytes)
G.711	PCM não linear	3,4	8	64	20	160
G.721	ADPCM	3,4	8	32		
G.722	ADPCM sub-banda	7	16	48, 56, 64		
G.723.1	MP-MLQ	3,4	8	6.4	30	24
G.723.1	ACELP	3,4	8	5.3	30	20
G.726	ADPCM	3,4	8	16, 24, 32, 40	15	60
G.728	LD-CELP	3,4	8	16	20	40
G.729A	CS-CELP	3,4	8	8	20	20

O G.711 define a representação de voz não compactada, usando modulação PCM. O sinal de voz é amostrado a 8kHz, codificado em 8 bits e a quantificação é não linear usando as escalas semilogarítmicas A ou  $\mu$ . Este codec, e outros, executam a conversão do sinal da voz de analógico para digital amostrando o sinal analógico em intervalos regulares (125 $\mu$ s ou 8000amostras/s) e então o convertendo em uma representação numérica (a quantificação e codificação). Quando utilizado em aplicações conversacionais (voz sobre IP, etc.), amostras de som são agrupadas em blocos para envio na forma de pacotes IP. Este bloco de informação, chamado de payload, é enviado em intervalos constantes. O valor típico do payload no G.711 é de 160 amostras, constituindo um tempo de voz de 20ms. O tamanho do payload em bytes pode ser calculado pela seguinte fórmula:  $\text{Payload} = (\text{taxa} \times \text{duração}) / 8$ .

Menores os payloads de voz, maior é a banda do canal necessária, isto devido ao aumento da sobrecarga dos diversos protocolos de transmissão da voz (tamanho do cabeçalho dos protocolos é considerável em relação à parte de dados). Entretanto, quando se aumenta o tamanho do payload aumenta-se também o atraso na aplicação (para aguardar a montagem do payload).

A recomendação G.721 converte um fluxo de bits de 64 kbps em um fluxo de 32 kbps. G.721 é baseado na técnica ADPCM e sua taxa de amostragem é de 8 kHz.

O G.722 fornece uma melhor qualidade de som (considera frequências de 50 até 7kHz) que o esquema convencional G.711 PCM ou a técnica de compressão G.721. Esta técnica utiliza 14 bits por amostra, em vez de 8 bits como as outras. G.722 é baseado no método ADPCM Sub-banda, onde o sinal de voz é dividido em duas sub-bandas (bandas de alta e de baixa frequência). O propósito desta subdivisão é que a sub-banda de baixa frequência é mais importante, assim ela precisa ser codificada com maior precisão. Este algoritmo é próprio para aplicações de videoconferência uma vez que telefones comuns não respondem na faixa de 7kHz [Soares, 2002].

O G.726 utiliza o ADPCM (*Adaptative Differential PCM*) a 40, 32, 24 e 16 kbps. Nesta técnica, a codificação da diferença entre amostras consecutivas é feita de forma não linear, já que o valor do passo de quantificação é variável. O sinal de voz é amostrado a 8kHz, codificado em 8 bits (leis A ou  $\mu$ ) e são

transmitidas diferenças entre amostras com 5, 4, 3 ou 2 bit em quantificação adaptativa. Neste algoritmo, o valor típico de tamanho do pacote de voz (payload) é de 15ms (60 bytes).

O G.728 adota a técnica de codificação LD-CELP (*Low-Delay, Code-Excited Linear Prediction*), gerando uma taxa de bits de 16 kbps. Em cada janela de 0,625ms do sinal de voz, são analisadas 5 amostras de 8 bits (que internamente voltam para PCM linear) e é gerado 1 código de 10 bits [Soares, 2002]. A tabela (codebook) utilizada é formada por 1024 valores. Esta tabela contém os valores de códigos (vetores) que representam as possíveis amostras do sinal de voz. Neste algoritmo, o valor típico de tamanho do pacote de voz (payload) é de 20ms (40 bytes).

O G.729 é bastante popular em aplicações de voz sobre frame relay e em modems V.70 para voz e dados. Ele usa uma técnica de codificação LD-CELP, gerando uma taxa de bits de 8 kbps. O G.729A a codificação CS-ACELP (Algebraic-ACELP). A cada janela de 10ms do sinal de voz, são analisadas 80 amostras de 8 bits para geração de 10 códigos de 8 bits [Soares, 2002]. Neste algoritmo, o valor típico de tamanho do pacote de voz (payload) é de 20ms (20 bytes).

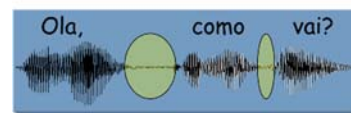
G.723.1 é o codec mínimo de referência para o padrão H.323 (seção 0). Ele opera a 6,4 kbps (*Multipulse-Maximum Likelihood Quantification*) e a 5,3 kbps (*Algebraic-Code-Excited Linear Prediction*). Em cada janela de 30 ms do sinal de voz, são analisadas 240 amostras de 16 bits do sinal de voz (tomadas a 8kHz) para identificação de padrões repetitivos (pitches) e são gerados 12 ou 10 códigos de 16 bits, conforme o algoritmo esteja configurado para uma taxa de 6,3 ou 5,3 kbps [Soares, 2002]. Neste algoritmo, o valor típico de tamanho do pacote de voz (payload) é de 30ms (20 ou 24 bytes).

### 3.9.3 Supressão do silêncio e remoção de sons repetitivos

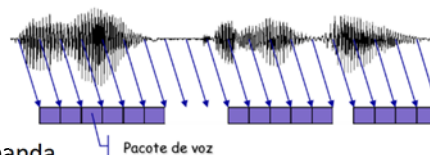
A compressão de voz pode ser obtida também através da remoção dos períodos de silêncio e de informações redundantes encontrados na fala humana. Estas informações existem na fala humana, mas não são necessárias para que uma comunicação efetiva exista através de uma rede. Os sons repetitivos, inerentes à voz, são causados pela vibração das cordas vocais. A transmissão destes sons idênticos não é necessária para efetivação da comunicação e a sua remoção resulta em um aumento de eficiência na utilização da banda de rede [Soares, 2002].

Apenas 22% do que se fala são componentes essenciais da comunicação e devem ser transmitidos para o entendimento do diálogo, outros 22% são padrões repetitivos e o restante, 56% representa as pausas entre falas [Bahner, 1996] [Soares, 2002].

Nos equipamentos de tratamento de voz sobre redes de pacotes, a supressão de silêncio é executada através da função VAD (*Voice Activity Detection* - detecção de atividade de voz). A VAD é um modo eficiente de liberar dinamicamente a largura de banda, proporcionando uma economia de até 50% da banda, permitindo que esta seja alocada para outras aplicações.



O DTX (Discontinuous Transmission) é a capacidade de um codec de parar de transmitir quadros quando o VAD tiver detectado um período de silêncio. Portanto, a combinação VAD + DTX é um modo eficiente de liberar dinamicamente a banda, proporcionando uma economia de até 50% da banda. Alguns codecs avançados não vão interromper a transmissão completamente. Em vez disso, vão para um modo de silêncio no qual usam muito menos largura de banda e enviam apenas os parâmetros mínimos para que o receptor possa restituir o ruído de fundo (intensidade, etc.)



Alguns pontos devem ser considerados na supressão do silêncio [Soares, 2002]:

- Quando a fala é muito frequente, contínua, os ganhos com a supressão do silêncio não são alcançados;
- Os algoritmos de compressão avançados já possuem integradas as funções de VAD;
- Como a detecção da presença de voz na transmissão não é imediata, ou seja, a fala está presente antes do início de execução da função de detector, pode ocorrer o corte das primeiras sílabas da locução. Este fenômeno é denominado de *clipping*;

- Quando o ruído de fundo é muito alto, torna-se difícil distinguir entre o que é ruído e o que realmente é fala. Corre-se o perigo de empacotamento de ruído.

### 3.9.4 MPEG Áudio

Até este ponto foram apresentadas algumas técnicas de codificação de áudio especialmente projetadas para a voz, com uma suposição de que a largura de banda do áudio está dentro de 3,4 kHz até 7 kHz. Esta seção introduz uma técnica de compressão de áudio genérico que pode ser usada para comprimir sons dentro da faixa dos sons audíveis (até 20 kHz).

MPEG-Audio é um padrão de compressão de áudio genérico. Diferente de muitos outros codificadores especialmente projetados para sinais de voz, o codificador MPEG-Audio realiza a compressão sem a suposição acerca da natureza da fonte do áudio. Em vez disso, ele explora as limitações de percepção do sistema auditivo humano.

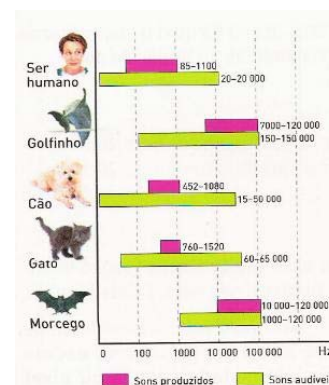
MPEG-Audio permite três frequências de amostragens: 32, 44.1 ou 48 kHz. A sequência de bits compactada pode suportar um ou dois canais de áudio. O fluxo comprimido pode ter uma das várias taxas de bits fixas e predefinidas variando de 32 a 320 Kbits/s. Dependendo da taxa de amostragem do áudio, o codificador MPEG-Audio pode ter uma razão de compressão variando de 2,7 a 24. Algumas taxas de bit e de amostragem foram definidas no padrão MPEG-2,5 (não oficial): taxas de bit de 8, 16, 24, e 144 kbit/s e amostragem de 8, 11.025, 12, 16, 22.05 e 24 kHz..

A utilização dos limites da audição humana baseia-se em três princípios básicos [<http://pt.wikipedia.org/wiki/MP3>]:

- Faixa de frequência audível dos seres humanos;
- Limiar de audição na faixa de frequência audível;
- Mascaramento em frequência e mascaramento temporal.

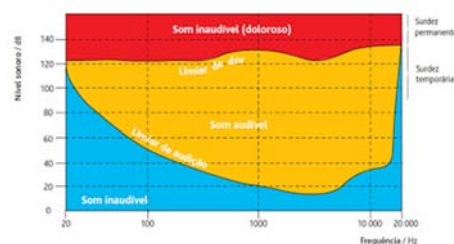
#### **Faixa de frequência audível humana**

O ouvido humano, devido às suas limitações físicas, é capaz de detectar sons em uma faixa de frequência que varia de 20Hz a 20KHz. É importante observar que esta faixa pode variar de indivíduo para indivíduo e também com a idade (com o envelhecimento perdemos a capacidade de ouvir frequências mais altas). Desta forma, não faz sentido armazenar dados referentes a sons fora desta faixa de frequência, pois ao serem reproduzidos, os mesmos não serão percebidos por um ser humano. Esta é a primeira limitação da audição humana do qual o sistema MP3 faz uso para alcançar altas taxas de compressão. De acordo com o Teorema de Nyquist, para garantir a reprodução de um sinal, temos de amostrá-lo pelo menos a duas vezes sua frequência máxima. Ou seja, neste caso, como a frequência máxima de interesse é 20KHz, basta amostrar a 40KHz. Utilizam-se 44100 Hz como taxa de amostragem, pois levam-se em consideração 10% de tolerância e busca-se um valor produto dos quatro primeiros números primos. (Obs.  $(2 \times 3 \times 5 \times 7)^2 = 44100$ ). Desta forma, esta taxa de amostragem funciona como um filtro passa-baixas, que remove todos os componentes de frequência fora da faixa de interesse, neste caso, acima de 20 KHz.



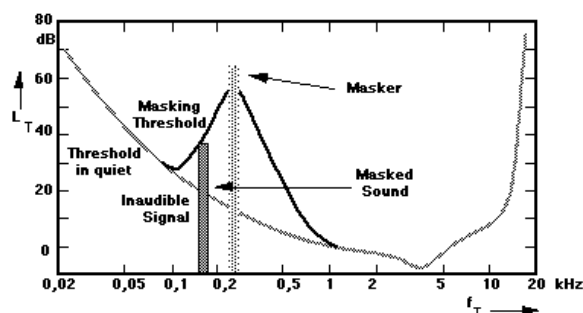
#### **Limiar de audição na faixa de frequência audível**

Outro fator utilizado pela codificação MP3 é a curva de percepção da audição humana dentro da faixa de frequências audíveis, ou **Limiar de Audição**. Apesar da faixa de audição humana variar entre 20Hz e 20KHz, a sensibilidade para sons dentro desta faixa não é uniforme. Ou seja, a percepção da intensidade de um som varia com a frequência em que este se encontra. Desta forma, o MP3 utiliza-se desta propriedade para obter compressão em arquivos de áudios. Esta abordagem é bastante intuitiva, sendo que o que se faz é descartar amostras que se encontrem abaixo deste limiar.



### Mascaramento em frequência e mascaramento temporal

Por fim, uma última propriedade da audição humana ainda é utilizada pelo método é o chamado mascaramento auditivo, ou “audibilidade diminuída de um som devido à presença de outro”, podendo este ser em frequência ou no tempo. O mascaramento em frequência ocorre quando um som que normalmente poderia ser ouvido é mascarado por outro, de maior intensidade, que se encontra em uma frequência próxima. Ou seja, o limiar de audição é modificado (aumentado)

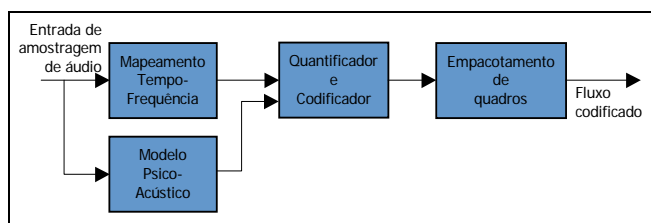


na região próxima à frequência do som que causa a ocorrência do mascaramento, sendo que isto se deve à limitação da percepção de frequências do ouvido humano. O mascaramento em frequência depende da frequência em que o sinal se encontra, podendo variar de 100Hz a 4KHz. Em função deste comportamento, o que o método de compressão do MP3 faz é identificar casos de mascaramento em frequência e descartar sinais que não serão audíveis devido a este fenômeno. Além do mascaramento em frequência, temos ainda o mascaramento no tempo, sendo que este ocorre quando um som forte é precedido por um mais fraco que se encontra em uma frequência próxima à do primeiro. Se o intervalo de tempo entre os dois for suficientemente pequeno, este som mais fraco não será percebido pela audição humana. Se um som é mascarado após um som mais forte, temos o chamado pós-mascaramento. No caso de um som ser mascarado antes do som mais forte, temos o que chamamos de pré-mascaramento. O pré-mascaramento existe só por um curto momento, cerca de 20ms, enquanto que o pós-mascaramento tem efeito por até 200ms. O método de compressão do MP3 utiliza-se, portanto deste fenômeno, identificando casos onde o mesmo ocorre e descartando sons que seriam mascarados, o que permite reduzir a informação de áudio consideravelmente sem mudança audível.

### Um codificador básico MPEG-Audio

A Figura 47 mostra um diagrama de blocos de um codificador MPEG-Audio básico:

- Bloco mapeamento tempo-frequência: divide a entrada em sub-bandas de frequências múltiplas.
- Bloco modelo psico-acústico: cria um conjunto de dados para controlar a operação do bloco quantificador e codificador.
- Bloco quantificador e codificador: cria um conjunto de símbolos de código. As sub-bandas menos importantes e áudios inaudíveis são removidos.
- Bloco Empacotamento de quadros: monta e formata os símbolos de código e adiciona outras informações (tal como correção de erros) se necessário, para formar um fluxo de áudio codificado.



**Figura 47.** Codificador MPEG-Áudio Básico

O padrão MPEG-1-Audio define um fluxo de bits que pode suportar um ou dois canais de áudio: um canal único, dois canais independentes, ou um sinal estéreo. Os dois canais de um sinal estéreo podem ser processados independentemente ou como um estéreo conjunto (joined stereo) para explorar a redundância estéreo (correlações entre canais estéreo).

O padrão MPEG-2-Audio estende as funcionalidades do seu predecessor pela codificação multicanal com até cinco canais (esquerdo, direito, centro, e dois canais surround), mais um canal adicional de baixa frequência, e/ou até sete canais multilíngues/comentários. Ele também estende a codificação estérea e mono do MPEG-1-Audio com taxas de amostragens adicionais.

## MP3

MPEG Audio especifica uma família de 3 esquemas de codificação de áudio, chamadas de Layer-1, Layer-2 e Layer-3. O padrão especifica o formato do fluxo de bits e o decodificador para cada esquema de codificação. Ele não especifica o codificador em si para possibilitar avanços futuros.

De Layer-1 a Layer-3, a complexidade e desempenho (qualidade de som e taxa de bits) aumentam. Os três codificadores são compatíveis no modo hierárquico. Isto é, o decodificador Layer-N é capaz de decodificar um fluxo de bits fluxo codificado com codificador Layer-N e abaixo de N.

Dentre estes Layers, o esquema de codificação mais conhecido é o MPEG 1 Layer 3, conhecido como MP3. Este formato substituir o WAVE (arquivos de som do windows 95). Sua vantagem é a compressão, uma música de 4 min ocupando 45 MB em wave será reduzida a 4,5 MB no formato mp3. Existem várias aplicações para o MP3: para ouvir MP3 são necessários players especiais, como RealPlayer ou WinAmp; para criar MP3 são necessários codificadores de WAVE em MP3 e direto do CD para MP3.

O princípio de funcionamento básico do mp3 é buscar num sinal de áudio normal, como um arquivo wave, todas os sinais redundantes e irrelevantes que não sensibilizam nosso ouvido. Isto é relativo. O algoritmo de compactação do mp3 corta frequências muito altas, acima dos 20kHz, que não são audíveis pelo ouvido humano. Só aí já são muitos bits economizados. Em qualquer música, se duas frequências muito próxima foram "tocadas" ao mesmo tempo nosso ouvido somente ouvirá a mais forte, ou seja, o mp3 simplesmente diminui o número de bits desse sinal mais fraco e mantém os bits do sinal mais forte, diminuindo assim o tamanho final do arquivo na proporção 12:1 (qualidade semelhante ao CD).

Com a redução dos requisitos de armazenamento das músicas pensou-se em criar aparelhos digitais de música que tocassem o formato MP3: walkmans, portáteis e para carros, periféricos para computador, etc.

## 3.10 Técnicas de Compressão de Vídeo Digital

### 3.10.1 ISO/IEC MPEG (Motion Picture Expert Group)

O grupo da ISO/IEC MPEG foi estabelecido em 1988 para desenvolver padrões para representação codificada de vídeos, áudios associados, e suas combinações quando usados para armazenamento e recuperação em *Digital Storage Media* (DSM). O conceito DSM inclui os dispositivos de armazenamento convencionais, como CD-ROMs, drivers de fita, discos rígidos e canais de telecomunicação (ISDN e LAN).

MPEG usa a compressão interquadros (redundância temporal), obtendo taxas de compressão de até 200:1 pelo armazenamento apenas das diferenças entre quadros sucessivos. Especificações MPEG também incluem um algoritmo para compressão de áudio a taxas de 5:1 a 10:1.

#### Grupos de Trabalho MPEG

MPEG teve diversos grupos de trabalho:

- MPEG-1 (1993) visa a codificação de vídeo com qualidade VHS: 360x280 pixels com 30 quadros por seg. na taxa de 1.5 Mbps (taxa dos drivers de CD-ROM da época). Quando se fala codificação MPEG é MPEG-1 que se está referenciando.
- MPEG-2 (1994) visa a codificação de vídeo com qualidade de televisão digital CCIR 601: 720x480 pixels com 30 quadros por seg. na taxa entre 2 a 10 Mbps.
- MPEG-3 visava a codificação de vídeo com qualidade HDTV na taxa de 40 Mbps. Estes trabalhos foram interrompidos em julho 1992.
- MPEG-4 (1998) objetiva a codificação audiovisual a taxas de bits muito baixas. A taxa de bits considerada aqui varia de 4,8 a 64 Kbps.
- MPEG-7 (2001) define uma interface de descrição de conteúdo Multimídia, um padrão de descrição de dados multimídia (informações audio-visuais) permitindo a busca e filtragem.

### 3.10.2 MPEG-1

MPEG 1 (ISO/IEC-11172) tem cinco partes:



- ISO/IEC-11172-1 MPEG-Sistemas: sincronização e multiplexação de fluxos de áudio e vídeo compactados;
- ISO/IEC-11172-2 MPEG-Vídeo: compressão de sinais de vídeo;
- ISO/IEC-11172-3 MPEG-Áudio: compressão de um sinal de áudio digital;
- ISO/IEC-11172-4 Teste de Conformidade: especifica procedimento para determinar as características dos fluxos codificados e para testar a conformidade com os requisitos identificados no Áudio, Vídeo e Sistemas;
- ISO/IEC-11172-5 Simulação de Software: oferece uma implementação de referência.

Anteriormente nesta apostila foi apresentada a compactação MPEG-1 Audio. Na sequência será apresentada a compactação MPEG-1 Video.

### Hierarquia de codificação do MPEG

Afim de ilustrar a sintaxe dos bitstreams, a estrutura de um fluxo de vídeo MPEG-1 é apresentada na Figura 48, sendo que a Figura 49 ilustra graficamente esta estrutura. Ela é composta das seguintes camadas:

- Sequência: composta de um cabeçalho de sequência, seguido de um ou mais grupos de imagens e termina com um fim de sequência.
- GOP (Grupo de imagens): fornece um ponto de acesso aleatório.
- Camada de imagem: contem todas as informações codificadas de uma imagem. A este nível o cabeçalho contém a referência temporal de uma imagem, o tipo de codificação, etc.
- Imagem: é a unidade elementar para a codificação do vídeo. Uma imagem MPEG é definida por um grupo de três matrizes retangulares que representam a luminância (Y) e a croma (Cr e Cb). Um elemento da matriz é 1 pixel. A representação YCrCb é equivalente ao RGB. É preferível YCrCb pois o olho é mais sensível a luminosidade que a croma, com isto é armazenado menos informação nas matrizes Cr e Cb que na matriz Y. Por exemplo, na codificação 4:2:2 (mais corrente), as matrizes Cr e Cb são de dimensão duas vezes menor que a matriz Y.
- Camada Pedacos: as imagens são divididas em pedaços (slices), cada pedaço consiste de um número de macroblocos de 16x16 pixels. Esta camada é importante para o controle de erro. Se existe um erro no fluxo de dados, o decodificador pode saltar um pedaço. Sendo que maior o número de pedaços, melhor é o tratamento de erro
- Camada Macrobloco: um bloco é uma matriz 8x8 de valores de pixels tratados como unidades e entrada para o DCT.

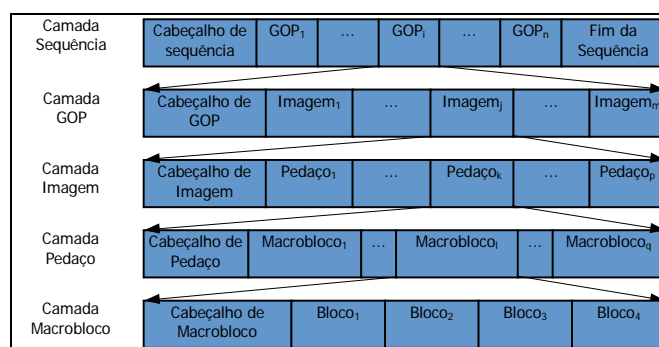
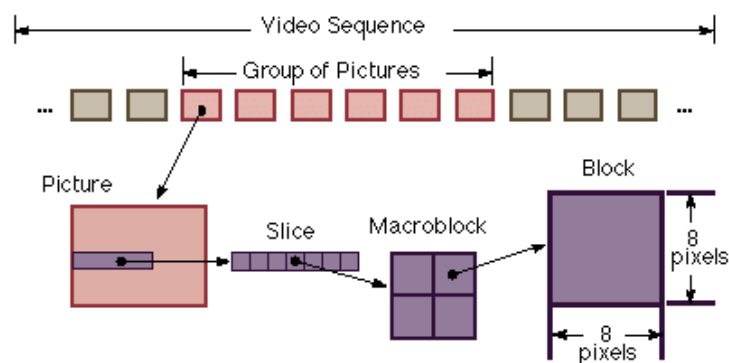


Figura 48. Estrutura do Fluxo de Vídeo MPEG-1



**Figura 49.** Hierarquia de um fluxo MPEG

Uma das características mais importantes do MPEG é que ele especifica apenas a sintaxe dos bitstreams codificados para que decodificadores possam decodificar. Este padrão não especifica como gerar o bitstream. Esta escolha permite inovações no projeto e implementação de codificadores.

### **Compactação MPEG-1 Video**

No vídeo, existem dois tipos de redundância: espacial e temporal. O MPEG-1 explora as duas. A redundância espacial pode ser explorada pela simples codificação em separado de cada quadro com o JPEG. Essa estratégia é utilizada às vezes, em especial quando há a necessidade de se acessar cada quadro, como na edição de produções de vídeo.

Uma compactação adicional pode ser obtida se nos beneficiamos do fato de que dois quadros consecutivos são, com frequência, quase idênticos. Esta é a redundância temporal. Por exemplo, para cenas em que a câmera e o fundo da cena permanecem estáticos e um ou dois atores se movimentam lentamente, quase todos os pixels serão idênticos de um quadro para outro. Neste caso, subtrair cada quadro do seu antecessor e executar o JPEG na diferença seria suficiente. Entretanto, para cenas onde a câmera faz uma tomada panorâmica ou uma mudança rápida de plano, essa técnica é insuficiente. É necessária alguma forma de compensação de movimento. Isso é exatamente o que o MPEG faz; essa é, também, a diferença mais importante entre o MPEG e o JPEG.

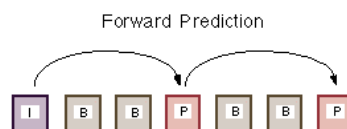
### **Grupo de Imagem consiste de quatro tipos de imagens**

A saída do MPEG-1 consiste de quatro tipos de quadros:

- Quadros I (Intracoded): imagens estáticas, independentes e codificadas com JPEG.
- Quadro P (Predictive): Diferença bloco a bloco com o último quadro.
- Quadro B (Birectional): Diferença com o último quadro e com o quadro seguinte.
- Quadro D (DC-coded): Médias de bloco usadas para o avanço rápido (fast forward).

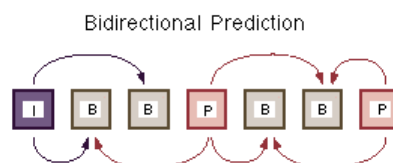
É necessário que quadros I apareçam periodicamente no fluxo de saída por três motivos: no caso de transmissão multicast, os receptores podem entrar no grupo em tempos distintos, requerendo um quadro I para começar a decodificação MPEG-1; se um quadro for recebido com erro, a decodificação não será mais possível; e sem quadros I, se houvesse um avanço rápido ou um retrocesso, o decodificador teria de calcular todos os quadros exibidos de modo a saber o valor total do quadro em que parou. Portanto, os quadros I são inseridos na saída uma ou duas vezes por segundo.

Os quadros P, ao contrário, codificam as diferenças entre os quadros. Ele tem é geralmente 50% do tamanho do quadro I. Eles se baseiam na ideia dos macroblocos, que cobre 16x16 pixels. Um macrobloco é codificado da seguinte forma: tentando-se localizá-lo, ou algo bem parecido com ele, no quadro anterior. Decodificar quadros P requer que o decodificador armazene o I ou P quadro anterior em um buffer e, depois, construa o novo quadro em um segundo buffer baseado em macroblocos completamente codificados e macroblocos contendo diferenças com o quadro anterior (Figura 50).



**Figura 50.** Decodificando quadros P

Quadros B são semelhantes aos quadros P, a diferença é que eles permitem que o macrobloco de referência esteja tanto no quadro I ou P anterior quanto no quadro seguinte (Figura 51). Com isto, o quadro B tem em média 15% do tamanho do quadro I. Essa liberdade acarreta uma melhoria na compensação do movimento, e também é útil quando, no vídeo, objetos passam na frente ou por trás de outros objetos. Para realizar a codificação de quadros B, o codificador precisa manter três quadros decodificados na memória ao mesmo tempo: o quadro anterior, o atual e o próximo.



**Figura 51.** Decodificando quadros B

Os quadros só são usados para possibilitar a apresentação de uma imagem de baixa resolução quando um avanço rápido, ou um retrocesso, estiver sendo realizado.

Uma sequência de quadros codificados teria a seguinte forma: IBBPBBPBBPBBIBBPBBPBBPB.....

A codificação MPEG-2 é fundamentalmente semelhante à codificação MPEG-1, com quadros I, P e B. Os quadros D, entretanto, não são aceitos. Além disso, a transformação discreta de co-seno é de 10x10 em vez de 8x8, para proporcionar mais 50 por cento de coeficientes e, com isso, melhor qualidade.

### MPEG-1 Sistemas

Enquanto MPEG-1 Vídeo e MPEG-1 Áudio especificam representação para áudios e vídeos, o MPEG-1 Sistemas define uma estrutura multiplexada para combinar fluxos elementares, incluindo áudio, vídeo, e outros fluxos de dados. Estes fluxos, chamados de fluxos MPEG, podem multiplexar até 32 fluxos de áudio MPEG, 16 fluxos de vídeo MPEG e 2 fluxos de dados de diferentes tipos. MPEG Sistemas também especifica o modo de representar as informações temporais necessárias para reprodução de sequências sincronizadas em tempo real.

A especificação da codificação MPEG Sistemas fornece campos de dados para suportar as seguintes funções: sincronização de fluxos elementares; gerenciamento de buffer nos decodificadores; acesso aleatório; identificação do tempo absoluto do programa codificado.

### 3.10.3 MPEG-4

O padrão MPEG4 começou a ser concebido em julho de 1993, tendo sido aprovado como padrão internacional em 2000. MPEG-4 absorve muita das características do MPEG-1 e MPEG-2 e outros padrões relacionados. Vários vídeos transmitidos pela Internet fazem uso deste padrão, assim como telefones celulares que utilizam imagens. Também é utilizado em diversos padrões de transmissão de TV digital, especialmente os de alta definição (HDTV) em sua versão AVC, como visto mais adiante.

MPEG-4, assim como MPEG-1 e MPEG-2, definem um conjunto de diversos tópicos denominados "parts". Cada parte aborda um aspecto diferente do padrão. Assim por exemplo, no MPEG-4 a parte 1 descreve a sincronização de áudio e vídeo; a parte 2 é uma tecnologia de compressão de vídeo; a parte 3 o processo de compressão do áudio; a parte 10 do padrão foi incluída quando uma versão mais otimizada da parte 2 (compressão de vídeo) foi desenvolvida. A tabela abaixo apresenta as partes do MPEG-4.

Parte	Data	Título	Descrição
Part 1	2010	Systems	Descreve sincronização e multiplexação de áudio e vídeo. Especifica o MPEG-TS (Transport Stream)
<a href="#">Part 2</a>	2004	Visual	Compressão de vídeo.
<a href="#">Part 3</a>	2009	Audio	Formatos de compressão de áudio, incluindo AAC, ALS, SLS.
Part 4	2004	Conformance testing	Procedimento para testes de desempenho
Part 5	2001	Reference software	Software de referência.
<a href="#">Part 6</a>	2000	Delivery Multimedia Integration Framework (DMIF)	Interface entre a aplicação e o transporte
Part 7	2004	Optimized reference software for coding of audio-visual objects	Exemplos de como melhorar implementação.
Part 8	2004	Carriage of ISO/IEC 14496 contents over IP networks	Métodos para transportar conteúdo MPEG-4 em redes IP.
Part 9	2009	Reference hardware description	Provê projetos de hardware
<a href="#">Part 10</a>	2012	Advanced Video Coding (AVC)	Um formato de compressão para vídeo (ITU-T H.264).
<a href="#">Part 11</a>	2005	Scene description and application engine	BIFS, XMT, MPEG-J. Define posicionamento de objetos, representação de objetos sintéticos 2D e 3D, ...
<a href="#">Part 12</a>	2012	ISO base media file format	Um formato de arquivo para armazenar conteúdo de mídia baseado em tempo
Part 13	2004	Intellectual Property Management and Protection (IPMP) Extensions	Gerenciamento de propriedade intelectual e proteção
<a href="#">Part 14</a>	2003	MP4 file format	Formato de arquivo MPEG-4 versão 2
Part 15	2010	Advanced Video Coding (AVC) file format	Formato de arquivo MPEG-4 Parte 10
Part 16	2011	Animation Framework eXtension (AFX)	Especifica o modelo MPEG-4 AFX para representação de conteúdos gráficos 3D
<a href="#">Part 17</a>	2006	Streaming text format	Formato de legenda
Part 18	2004	Font compression and streaming	
Part 19	2004	Synthesized texture stream	
<a href="#">Part 20</a>	2008	Lightweight Application Scene Representation (LASer) and Simple Aggregation Format (SAF)	Baseado no SVG Tiny. Para portais interativos, TV móvel, desenhos 2D, mapas gráficos interativos, etc.
Part 21	2006	MPEG-J Graphics Framework eXtensions (GFX)	Descreve um ambiente programático leve para aplicações multimídia interativas.
Part 22	2009	<a href="#">Open Font</a> Format	
Part 23	2008	Symbolic Music Representation (SMR)	
Part 24	2008	Audio and systems interaction	
Part 25	2009	3D Graphics Compression Model	
Part 26	2010	Audio Conformance	
Part 27	2009	3D Graphics conformance	
Part 28	2012	Composite font representation	
Part 29	2014	Web video coding	
Part 30	2014	Timed text and other visual overlays in ISO base media file format	
Part 31	2014	Video Coding for Browsers (VCB)	Codec para browsers

O MPEG-4 Parte 2 é uma tecnologia de compressão de vídeo desenvolvida pela MPEG. Ele é um padrão de compressão DCT (Discrete Cosine Transformation), similar aos padrões anteriores, como MPEG-1 e MPEG-2. Para permitir seu uso em várias aplicações, variando de câmeras de segurança de baixa qualidade, baixa resolução a HDTVs e DVDs, muitos padrões de vídeo agrupam características em perfis (profiles) e níveis. MPEG-4 parte 2 tem aproximadamente 21 perfis. O perfil Simple Profile (SP) é usado em situações onde a baixa taxa de bits e baixa resolução são mandatórios devido a largura de banda da rede, tamanho do dispositivo, etc, como no caso de telefones celulares, sistemas de segurança, etc. O perfil Advanced Simple Profile (ASP) é muito similar ao H.263, incluindo suporte para a quantificação do

estilo MPEG, suporte a vídeo entrelaçado, suporte a imagens do tipo B, compensação de movimento QPel (Quarter Pixel) e Global (GMC).

MPEG-4 Parte 10, também conhecidos como H.264 ou AVC (Advanced Video Coding) é um padrão de codec de vídeo digital que tem a característica de alta taxa de compressão. O padrão define 7 perfis, voltadas a classes de aplicações específicas. Por exemplo Baseline Profile (BP) é voltado para aplicações de custo mais baixo com limitado recursos computacionais, usado em aplicações de videoconferência e móveis. O Extended Profile (XP) é voltado para streaming de vídeo, com alta taxa de compressão e robustez para perda de dados. O High Profile (HiP) é o principal perfil para aplicações de armazenamento em disco e broadcast, particularmente para aplicações de ADTV e adotado pelos discos HD-DVD e Blu-ray.

### 3.10.4 CCITT H.261

Reconhecendo a necessidade de fornecer serviços de vídeo onipresentes na Rede Digital de Serviços Integrados (ISDN), o CCITT Study Group XV estabeleceu em 1984 um Grupo Especialista em Codificação para Telefonia Visual com o objetivo de recomendar uma codificação padrão de vídeo para transmissão a taxas, surgiu então a recomendação H.261.

H.261 é um dos padrões da família H.320 para videofonia e tele-conferência na taxa de 64 Kbps a 2 Mbps. Ele fornece poucos quadros por segundos com uma resolução cerca de oito vezes mais baixa que a qualidade TV PAL/SECAM.

O padrão H.261, também chamado de px64, obtém grandes taxas de compressão para a transmissão de vídeo colorido tempo-real. O algoritmo combina codificação intraquadro e interquadro (redundância espacial e temporal) para fornecer um rápido processamento para compressão/descompressão tempo-real de vídeo, otimizado para aplicações tal como telecomunicações baseadas em vídeo. Como estas aplicações usualmente não são a movimentos intensos, o algoritmo usa uma limitada estratégia de busca e estimação de movimento para obter taxas de compressão mais altas. H.261 pode obter taxas de compressão de 100:1 a mais de 2000:1.

A recomendação H.261 define um padrão de codificação de vídeo para transmissão na taxa de  $p \times 64$  Kbps ( $p=1,2,3,4,\dots,30$ ) que cobre as capacidades do canal ISDN. Sendo que as aplicações alvo desta recomendação são a videofonia e a teleconferência, onde o algoritmo de compressão de vídeo opera em tempo-real com atraso mínimo. Para  $p = 1$  ou  $2$ , devido à limitação de taxa de bits, apenas movimentos lentos, comunicação visual face-a-face (videofonia) são apropriados. Para  $p > 5$ , com uma maior taxa de bits disponível, imagens mais complexas podem ser transmitidas com melhor qualidade (videoconferência). Note que a máxima taxa de bits disponível é 1,92 Mbps ( $p=30$ ), que é suficiente para obter imagens de qualidade VHS [Raghavan, 98].

H.261 opera com dois formatos de imagem: CIF (*Common Intermediate Format*) e QCIF (quarter-CIF). CIF, de 320x288, permite usar um formato único dentro e entre regiões usando padrões de TV de 625 e 525 linhas. QCIF, de tamanho 160x144, é mais útil em taxas de bit menores ( $p < 6$ ).

O algoritmo de codificação é um híbrido de predição inter-quadro, transform coding (DCT), similar ao JPEG, e compensação de movimento. A taxa de dados do algoritmo de codificação foi projetada para ser capaz de suportar 40 kbps e 2Mbps. A predição interquadro remove a redundância temporal. O transform coding remove a redundância espacial. Para remover redundâncias adicionais no bitstream a ser transmitido, uma codificação por entropia (normalmente codificação de Huffman) é utilizada para reduzir ainda mais o vídeo.

### 3.10.5 H.263

H.263 é um padrão de vídeo a baixa taxa de bits para aplicações de teleconferência que opera a taxas abaixo de 64 Kbps. A codificação de vídeo é uma extensão do H.261 e descreve um método de codificação DPCM/DCT.

Uma ideia interessante do H.263 é o quadro PB. Ele consiste de duas imagens codificadas em uma unidade. O nome PB é derivado da terminologia MPEG dos quadros P e B. Assim, um quadro PB consiste de um quadro P que é produzido a partir do último quadro P decodificado e um quadro B que é produzido a partir do último quadro P decodificado e do quadro P sendo decodificado.

H.263 suporta cinco resoluções. Além do QCIF e CIF que é suportado pelo H.261, existem o SQCIF, 4CIF e 16CIF. SQCIF é aproximadamente a metade da resolução do QCIF. 4CIF e 16CIF são aproximadamente 4 e 16 vezes a resolução do CIF. O suporte do 4CIF e 16CIF significa que o codec poderia então competir com outras codificações de mais altas taxas de bits como os padrões MPEG.

Testes atuais mostram que o H.263 tem um desempenho 1 a 2,5 melhor que o H.261. Isto significa que, dada uma qualidade de imagem, a taxa de bits H.261 é aproximadamente 2,4 vezes a gerada pelo H.263.