



**UFSC**

**Ciências da  
Computação**

# **CAP 3. COMPRESSÃO DE DADOS MULTIMÍDIA**

**INE5431 Sistemas Multimídia**  
**Prof. Roberto Willrich (INE/UFSC)**  
**roberto.willrich@ufsc.br**

**Aula 1: Necessidades e  
Princípios da Compressão**

# Cap 3. Compressão de Dados Multimídia



## ■ Conteúdo:

- Necessidade de compressão
- Entropia: Teorema da codificação da fonte
- Princípios da compressão
- Classificação das técnicas de compressão
- Medição do desempenho de compressão
- Técnicas de compressão sem perdas
  - RLE, Huffman, LZW (GIF), (A)DPCM
- Técnicas de compressão de áudio, vídeo e imagens
- Padrões de compressão multimídia
  - JPEG, MPEG, MPEG-4, H.261, H.263

# Necessidade da compressão



- **Técnicas de compressão são essenciais para as aplicações multimídia, devido**
  - ao grande requisito de espaço para armazenamento de dados multimídia
  - ao fato que a largura de banda da rede e de dispositivos de armazenamento que não permite a transmissão de dados multimídia de alta qualidade em tempo-real

# Necessidade da compressão

- **Requisitos de espaço para armazenamento**

<i>Aplicações</i>	<i>Requisitos de Armazenamento (MBytes)</i>
<i>Livro de 500 páginas</i>	1
<i>100 imagens monocr.</i>	7
<i>100 imagens coloridas</i>	100
<i>1h de áudio qual. telefone</i>	28,8
<i>1h de Áudio-CD</i>	635
<i>1h Vídeo qual. VHS</i>	24300
<i>1h TV</i>	97000
<i>1h HDTV</i>	389000

- É necessária a utilização de técnicas de compressão de dados multimídia para viabilizar o armazenamento

# Necessidade da compressão

## ■ Requisitos de largura de banda

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



- Transmissão de som de qualidade CD não compactado
  - é possível em redes locais
    - 10, 100, 1000 Mbps
  - redes de media e longa distância depende da taxa de upload
- Transmissão de vídeo de qualidade televisão
  - incompatível com qualquer rede local e transmissão em WAN



# Necessidade da compressão

- **Pequena largura de banda dos dispositivos de armazenamento**
  - Dispositivo de armazenamento deveria ter uma taxa de 30 MBytes/s para apresentar um vídeo em tempo real de 620x560 pixels a 24 bits por pixel a 30 fps
    - 1x no CD = 150 kbps (velocidade para cd áudio)
    - 1x no DVD = 1,385 MBps
    - Tecnologia de CD-ROM de hoje fornece uma taxa de transferência de 7,62 MBps (x52) a 10,8 MBps (x70)
    - Drivers de DVD convencionais são de 16x (22,16 MBps)
  - Não é possível apresentar vídeo não compactado em tempo-real devido a taxa de bits insuficiente de alguns dispositivos de armazenamento
    - Única solução é compactar o dado



# Necessidade da compressão



- **Conclusão**

- É necessário compactação afim de armazenar, apresentar e transmitir informações multimídia
  - técnicas de compressão modernas reduzem os requisitos de armazenamento e portanto os requisitos de largura de banda da rede e do dispositivo de armazenamento

# Teorema de codificação da fonte



- **Teorema de Shannon**
  - Estabelece os limites da compressão de dados
- **Informação (amostra de áudio, pixel de imagem, etc.) deve ser codificadas para fins de transmissão e armazenamento**
  - Representada por um número de símbolos
  - Eficiência do codificador: uso de uma menor quantidade de símbolos médios possíveis





# Teorema de codificação da fonte



- **Dado um alfabeto com  $s$  símbolos, quantos bits ( $n$ ) são necessários para codificá-los?**

$$R: n = \lceil \log_2 s \rceil \Leftrightarrow 2^n = s$$

- Ex.: Se precisamos representar 200 símbolos, é necessário  $\log_2(200) = 7,64 \Rightarrow 8$  bits.
- **Verdadeiro se...**
  - Não for conhecida a distribuição de probabilidades...
  - Se a probabilidade da ocorrência de cada símbolo for idêntica (distribuição uniforme)

# Teorema de codificação da fonte

- **Shannon (1948) definiu uma medida chamada de entropia, definida como:**
  - Seja um alfabeto  $X = \{x_1, x_2, \dots, x_n\}$ , cujos símbolos apresentam probabilidades de ocorrência  $P = \{p_1, p_2, \dots, p_n\}$ , a entropia  $H(X)$  é definida como:

$$H(X) = - \sum_{i=1}^n p_i \times \log_2 \left( \frac{1}{p_i} \right)$$

Entropia é a média da quantidade de dados mínima para representar a informação

Base 2 fornece o resultado em bits, ou shannons...

A entropia do lance de uma moeda é de 1 bit ( $p_{\text{cara}} = p_{\text{coroa}} = 0.5$ )  
 $H(x) = -1 \cdot (0.5 \cdot \log_2(1/0.5) + 0.5 \cdot \log_2(1/0.5)) = 1$

# Princípios da Compressão de Dados

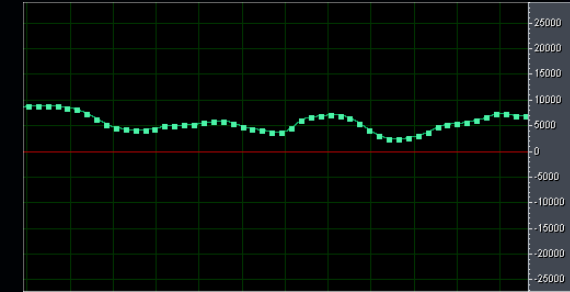


- **Fatores explorados pelas técnicas de compressão**

- Redundância de dados
- Propriedades da percepção humana

- **Redundância de Dados**

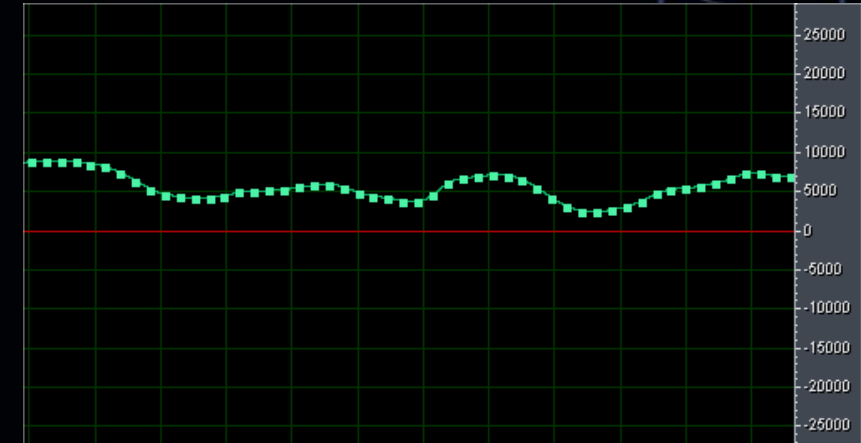
- Representação de dados multimídia
  - áudio digital é uma série de valores amostrados
  - imagem é uma matriz de valores amostrados (píxeis)
  - vídeo é uma sequência de imagens apresentadas numa certa taxa
- Amostras vizinhas não são inteiramente diferentes
  - valores vizinhos são de algum modo relacionados (redundância)
- Remoção da redundância não altera o significado do dado



# Princípios da Compressão de Dados



- **Redundância em áudio digital**
  - ▣ Amostragens adjacentes são similares:
    - próximo valor pode ser previsto baseado no valor atual
      - técnicas de compressão: Codificação preditiva
      - Exemplo ilustrativo:
        - Original (amostras de 8bits)
          - 23, 24, 26, 25, 27 ( $8 \times 5 = 40$  bits)
        - Compactado com função de predição  $a_i = a_{i-1} + \text{erro}$ 
          - 23, +1, +2, -1, +2
          - Tamanho:  $8 + 4 \times 4 = 24$  bits



# Princípios de Compressão: Redundância

- **Redundância em imagem digital**
  - Amostras vizinhas são similares
    - chamada de redundância espacial
      - removida utilizando técnicas de codificação preditiva ou outras

22	23	24
21	21	22



22	+1	+1
-1	0	+1





# Princípios de Compressão: Redundância

- **Redundância em vídeo digital**
  - Vídeo é uma sequência de imagens
    - imagens tem redundância espacial
  - Imagens vizinhas são normalmente similares
    - redundância temporal
      - removida utilizando técnicas de codificação preditiva ou outras



# Princípios de Compressão: Percepção Humana

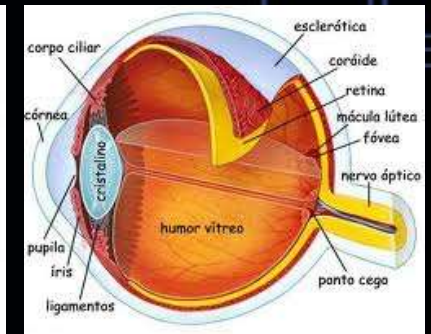
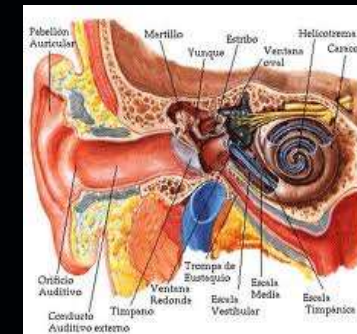
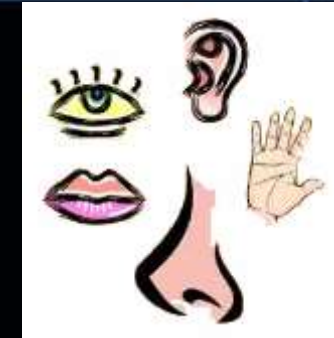


- **Humanos não são perfeitos**

- Não percebemos todas as informações sonoras e visuais
- Podem tolerar alguma perda de informação sem afetar a efetividade da comunicação
  - versão compactada não necessita representar exatamente a informação original

- **Algumas informações são mais importantes para a percepção humana que outras**

- Técnicas de compressão podem remover informações desnecessárias
  - áudios mascarados, intensidade luminosas/cor



# Pontos Importantes

## Teorema da codificação da fonte

- Entender a Entropia

## Princípios da compressão

- Redundância de dados
- Limitações da percepção humana



**UFSC**

**Ciências da  
Computação**

# **CAP 3. COMPRESSÃO DE DADOS MULTIMÍDIA**

**INE5431 Sistemas Multimídia**  
**Prof. Roberto Willrich (INE/UFSC)**  
**roberto.willrich@ufsc.br**

**Aula 2: Classificação e Medidas de  
desempenho da compressão**



# Cap 3. Compressão de Dados Multimídia



- **Conteúdo:**
  - Necessidade de compressão
  - Entropia: Teorema da codificação da fonte
  - Princípios da compressão
  - **Classificação das técnicas de compressão**
  - **Medição do desempenho de compressão**
  - Técnicas de compressão sem perdas
    - RLE, Huffman, LZW (GIF)
  - Técnicas de compressão de áudio, vídeo e imagens
    - Técnicas de compressão de voz
    - Técnicas de compressão de som
  - Padrões de compressão multimídia
    - JPEG, MPEG, MPEG-4, H.261, H.263



# Classificação das Técnicas de Compressão

- **Sem perda (Codificação por Entropia):**
  - Dado original pode ser exatamente reconstruído (reversível)
    - Técnica genérica: trata cadeias de bytes sem levar em conta seu significado
    - técnicas de compressão sem perda exploram apenas estatísticas de dados (redundância de dados)
      - baixas taxa de compressão



# Classificação das Técnicas de Compressão

- **Com perda (codificação na origem)**
  - Utilizado em dados multimídia onde erros e perdas são toleráveis
    - Utiliza propriedades da percepção humana
      - altas taxa de compressão
    - Leva em consideração a semântica dos dados
      - removendo dados irrelevantes compactando o dado original



# Classificação das Técnicas de Compressão

- **Codificações Híbridas**
  - Combinam técnicas com perda e sem perdas
    - várias técnicas são agrupadas para formar uma nova técnica de codagem
  - Taxa de compressão mais altas



Descarta dados irrelevantes para a percepção (com perdas)

Elimina Redundância (sem perdas)



# Parâmetros de Desempenho

- **Taxa de compressão**
  - Tamanho do dado original/tamanho do dado após a compressão
    - para sem perdas: quanto maior esta taxa melhor é a técnica

Lena Original (bmp)= 147.766 bytes





# Parâmetros de Desempenho

- **Qualidade da mídia reconstituída (técnicas com perdas)**
  - medida em SNR (Razão Sinal/Ruído)
  - maior SNR melhor é a qualidade





# Parâmetros de Desempenho

- **Qualidade da média reconstituída**
  - Há diversas formas para medir o erro gerado pelo codificador
    - Uma delas é a Média dos Erros Quadráticos (**MSE** – Mean Squared Error)
    - Considerando que tanto Ori quanto Dec tenham tamanho n, cada



$$\text{MSE}(\text{Orig}, \text{Dec}) = \frac{1}{n} \sum_{i=1}^n (\text{ori}_i - \text{dec}_i)^2$$

- No exemplo:  $\text{MSE} = \frac{1}{4} ((11-12)^2 + (12-12)^2 + (12-12)^2 + (14-15)^2) = 0,5$

# Parâmetros de Desempenho



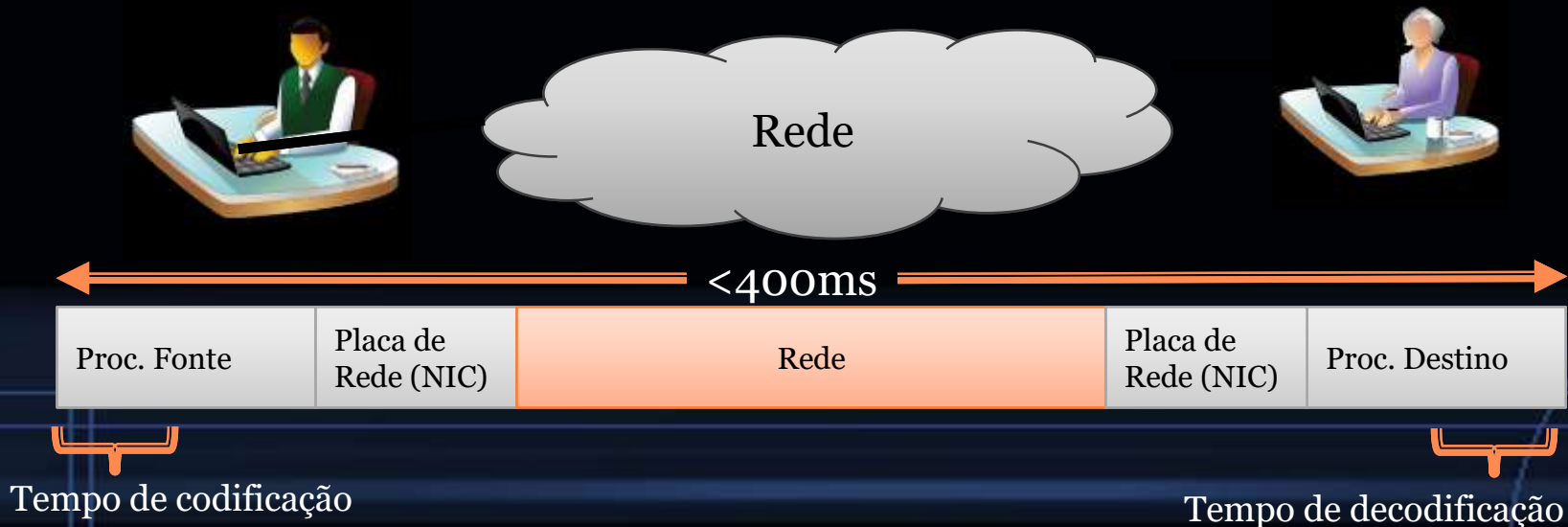
- **Qualidade da média reconstituída**
  - Relação Sinal-Ruído de Pico (**PSNR** – Peak Signal-to-Noise Ratio), definida (em dB)

$$\text{PSNR}(\text{Orig}, \text{Dec}) = 10 \times \log_{10} \left( \frac{(2^b - 1)^2}{\text{MSE}(\text{Orig} - \text{Dec})} \right)$$

- b = número de bits por símbolo
- Assumindo 8 bits no exemplo anterior:
  - $\text{PSNR}(\text{Orig}, \text{Dec}) = 10 \times \log_{10} \left( \frac{(2^8 - 1)^2}{0,5} \right) = 27,08 \text{ db}$
- Se não há perdas (Orig = Dec)
  - $\text{PSNR}(\text{Orig}, \text{Dec}) = \infty$

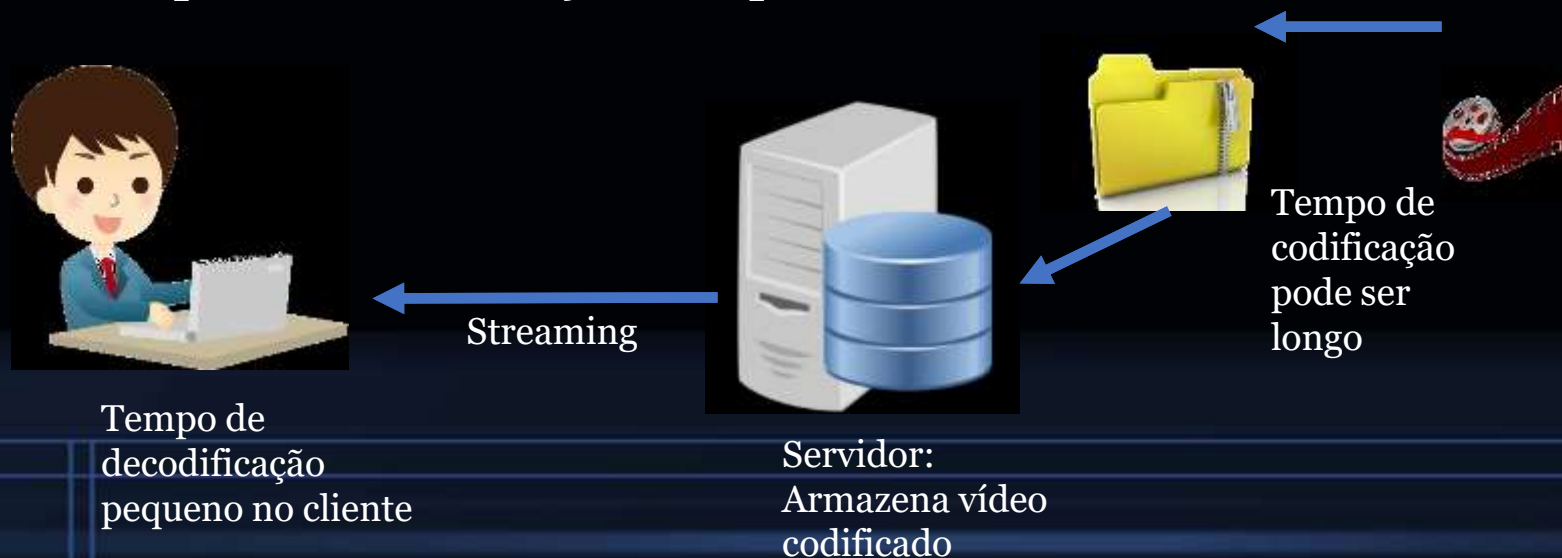
# Parâmetros de Desempenho

- **Complexidade de implementação e velocidade de compressão**
  - Importante para aplicações tempo-real (como videoconferência)
    - compressão e descompressão devem ser realizadas em tempo-real



# Parâmetros de Desempenho

- **Complexidade de implementação e velocidade de compressão**
  - Para aplicações de streaming ou não tempo-real
    - Tempo de codificação não é muito importante
    - Tempo de decodificação é importante



# Pontos Importantes

## Tipos de técnicas de compressão

- Entender os três tipos de compressão

## Parâmetros de desempenho das técnicas de compressão

- Taxa de Compressão
- Relação SNR
- Complexidade do algoritmo vs atraso de codificação





**UFSC**

**Ciências da  
Computação**

# **CAP 3. COMPRESSÃO DE DADOS MULTIMÍDIA**

**INE5431 Sistemas Multimídia**  
**Prof. Roberto Willrich (INE/UFSC)**  
**roberto.willrich@ufsc.br**

**Aula 3: Técnicas de Codificação  
sem perdas: RLE e Codificação  
de Huffman**

# Cap 3. Compressão de Dados Multimídia



- **Conteúdo:**
  - Necessidade de compressão
  - Entropia: Teorema da codificação da fonte
  - Princípios da compressão
  - Classificação das técnicas de compressão
  - Medição do desempenho de compressão
  - **Técnicas de compressão sem perdas**
    - RLE, Huffman, LZW (GIF)
  - Técnicas de compressão de áudio, vídeo e imagens
    - Técnicas de compressão de voz
    - Técnicas de compressão de som
  - Padrões de compressão multimídia
    - JPEG, MPEG, MPEG-4, H.261, H.263

# Técnicas de compressão sem perdas



- **Codificação RLE (Codificação por entropia)**
  - Técnica simples de compressão de dados: dados podem ser compactados através da supressão de sequências de mesmos símbolos
  - Aplicação: formatos padrões como PCX, BMP (RLE) e Photoshop
    - BMP RLE suporta 256 cores
  - Um exemplo simples
    - Original: 12 12 12 12 09 09 09 21 21 23 24 25 25 25 25 25 25 25
    - Compactado: 04 12 03 09 02 21 01 23 01 24 08 25
      - Cada repetição é codificada como:  
(número de repetição, símbolo repetindo)

# Técnicas de compressão sem perdas



- **Codificação RLE (Codificação por entropia)**







- Tem diversas variações

- Sequências idênticas são substituídas por um símbolo especial, número de ocorrências e o símbolo repetido
      - Original: UHHHHHHIMMG1223      Compactado: U!6HIMMG1223
      - Se o símbolo especial ocorrer no dado de entrada, ele deve ser substituído por dois símbolos
        - entrada: U!HIIIIID      saída: U!!H!5ID
      - Técnica não é utilizada para sequências menores que 4
        - exemplo: U!6HI!2MG1223 (não a compactação)
    - Algoritmo pode ser facilmente otimizado
      - pode-se substituir sequências maiores que um
      - requer que o tamanho da sequência seja codificado ou pode-se usar um caractere especial de fim
        - entrada: UFYUGDUFHUFHUFHUFHUFHBFD
        - saída: UFYUGD!5UFH\$BFD

# Técnicas de compressão sem perdas

- **Codificação RLE (Codificação por entropia)**
  - Fator de compressão depende do dado de entrada
    - Demonstração usando BMP RLE



Nome	Tamanho	Tipo
 Ninféias24bits.bmp	1.407 KB	Imagem de bitmap
 Ninféias8bits.bmp	470 KB	Imagem de bitmap
 Ninféias8bitsRLE.bmp	389 KB	Imagem de bitmap
 Jazz Man24bits.bmp	1.650 KB	Imagem de bitmap
 Jazz Man8bits.bmp	552 KB	Imagem de bitmap
 Jazz Man8bitsRLE.bmp	210 KB	Imagem de bitmap



# Técnicas de compressão sem perdas

## ■ Codificação Run-Length

- Só traz ganhos relevantes se houver grandes agrupamentos de símbolos iguais
- As principais aplicações são imagens bitmap
  - em imagens com grandes espaços envolvendo uma só cor
  - em imagens geradas por computador
    - onde os dados estão agrupados de forma mais geometricamente definida



Nome	Tamanho	Tipo
Ninféias24bits.bmp	1.407 KB	Imagem de bitmap
Ninféias8bits.bmp	470 KB	Imagem de bitmap
Ninféias8bitsRLE.bmp	389 KB	Imagem de bitmap
Jazz Man24bits.bmp	1.650 KB	Imagem de bitmap
Jazz Man8bits.bmp	552 KB	Imagem de bitmap
Jazz Man8bitsRLE.bmp	210 KB	Imagem de bitmap

# Técnicas de compressão sem perdas

- **Codificação de Huffman (Codificação Estatística)**
  - Método que atribui menos bits a símbolos que aparecem mais frequentemente e mais bits para símbolos que aparecem menos
  - Ideia usada no código de Morse

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	. . . .	=	- . . .

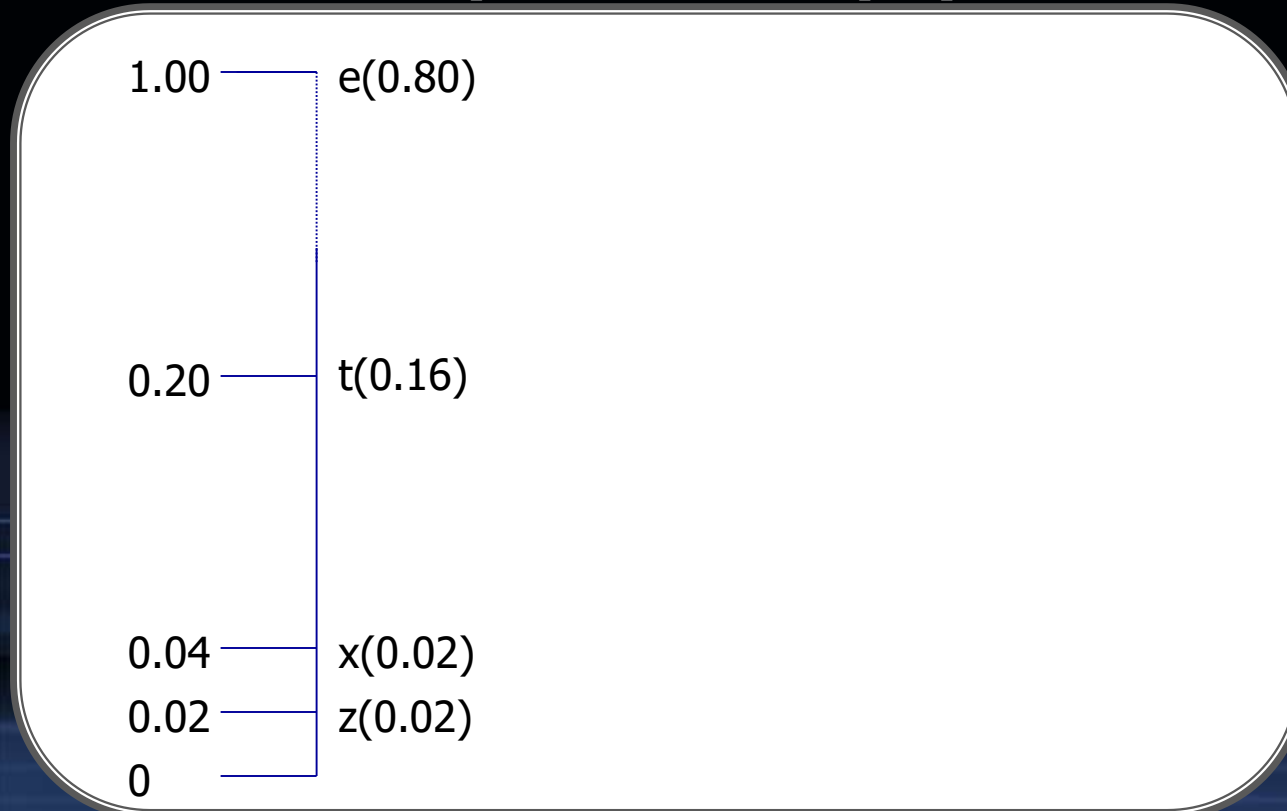


# Técnicas de compressão sem perdas

## ■ Codificação de Huffman (Original)

### ▫ Geração dos códigos Huffman

- a) colocação dos símbolos ao longo de uma linha de probabilidade acumulada (probabilidade aumenta de baixo para cima)
  - símbolos de mesma frequência: colocar em qualquer ordem

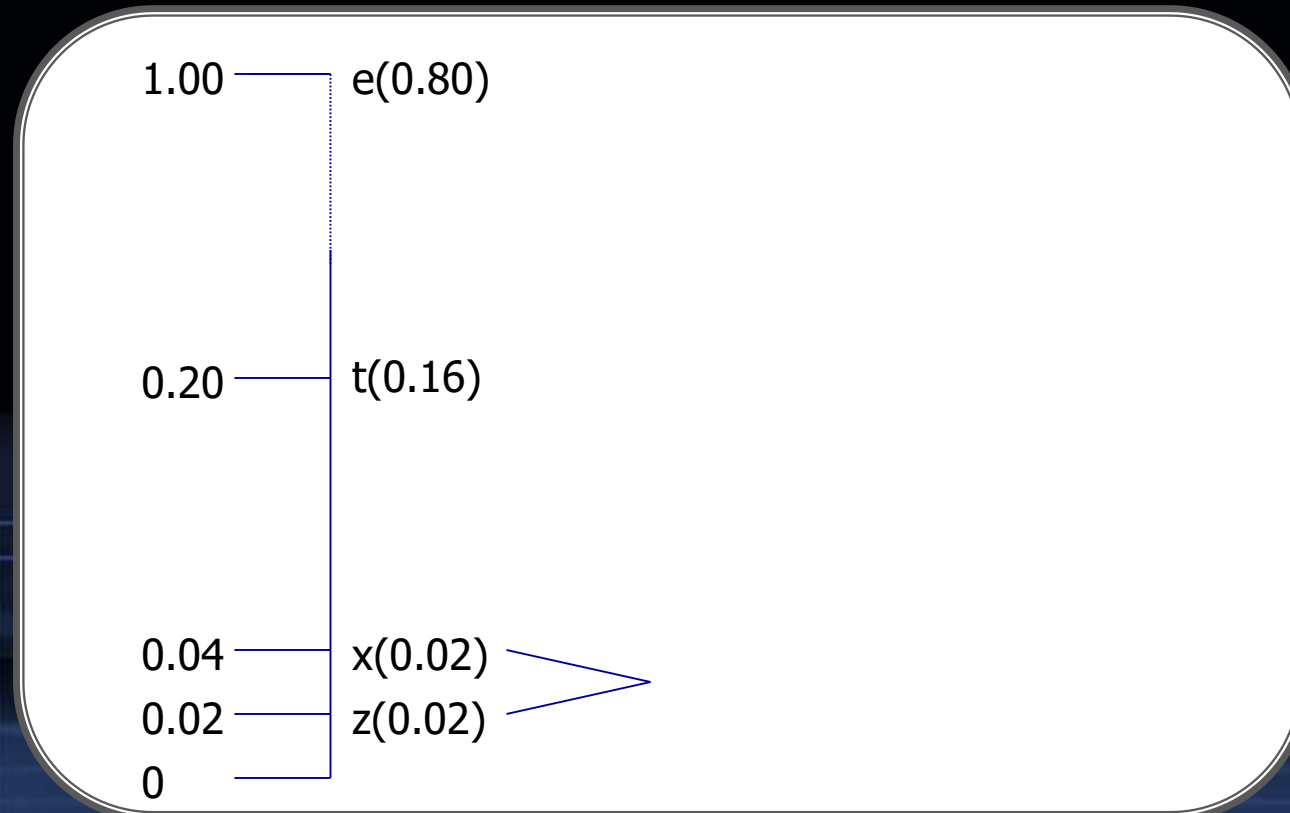


# Técnicas de compressão sem perdas

- **Codificação de Huffman**

- ▣ Geração dos códigos Huffman

- b) Junta-se os dois símbolos de menor probabilidade a um nó para formar dois ramos na árvore



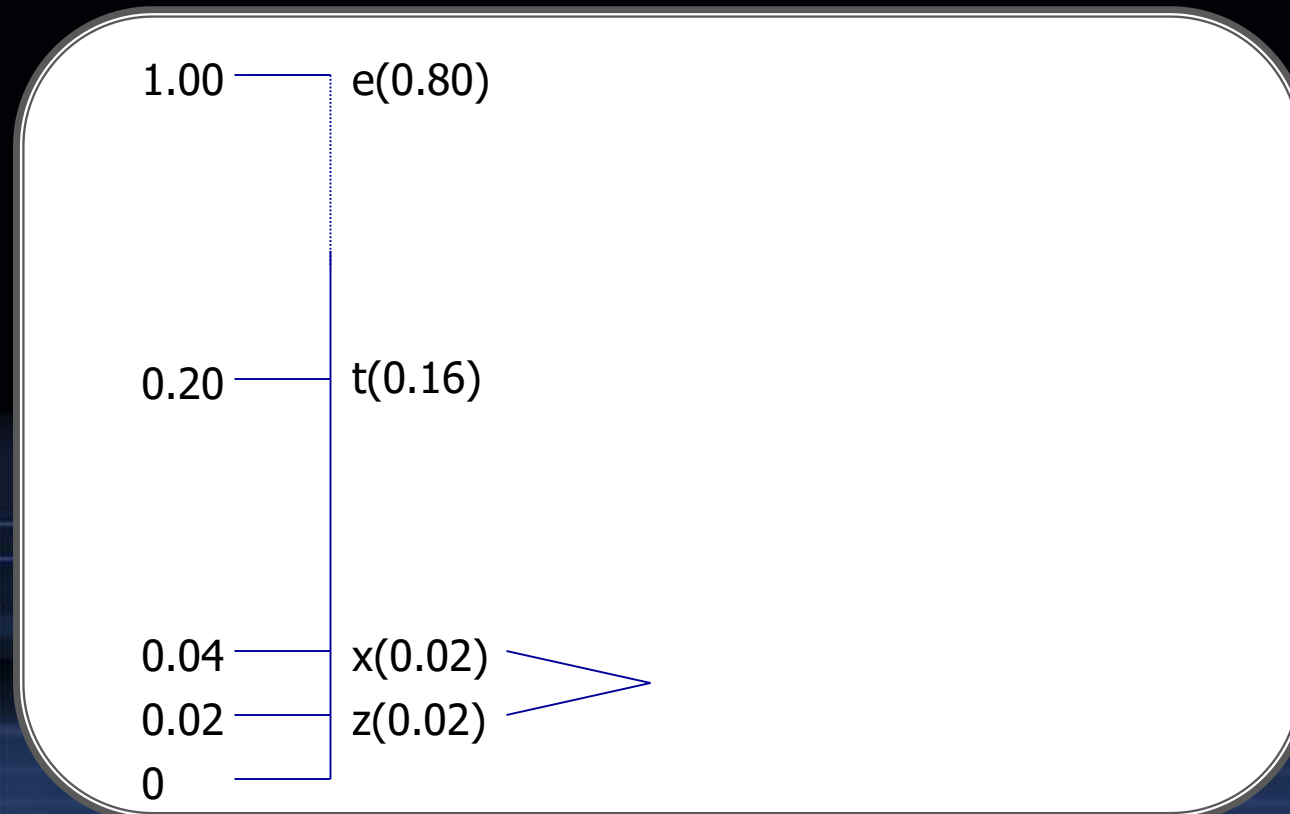


# Técnicas de compressão sem perdas

- **Codificação de Huffman**

- Geração dos códigos Huffman

- b) Junta-se os dois símbolos de menor probabilidade a um nó para formar dois ramos na árvore

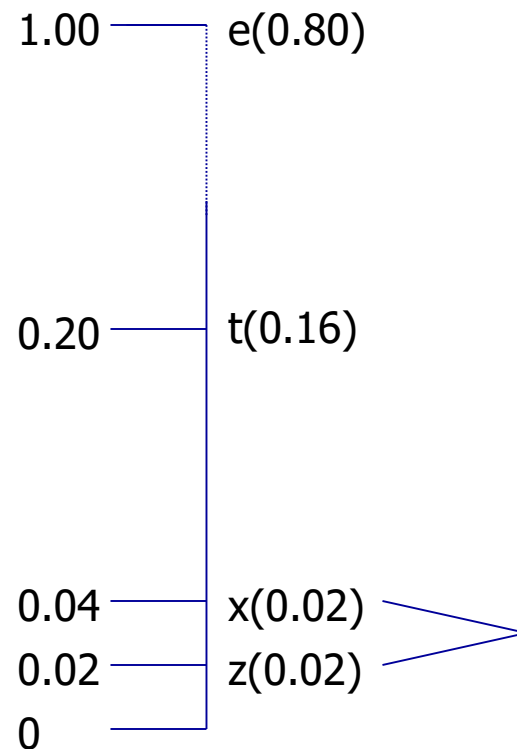


# Técnicas de compressão sem perdas

- **Codificação de Huffman**

- Geração dos códigos Huffman

- c) Nova árvore formada é tratada como um símbolo único com a probabilidade igual a soma dos símbolos ramos

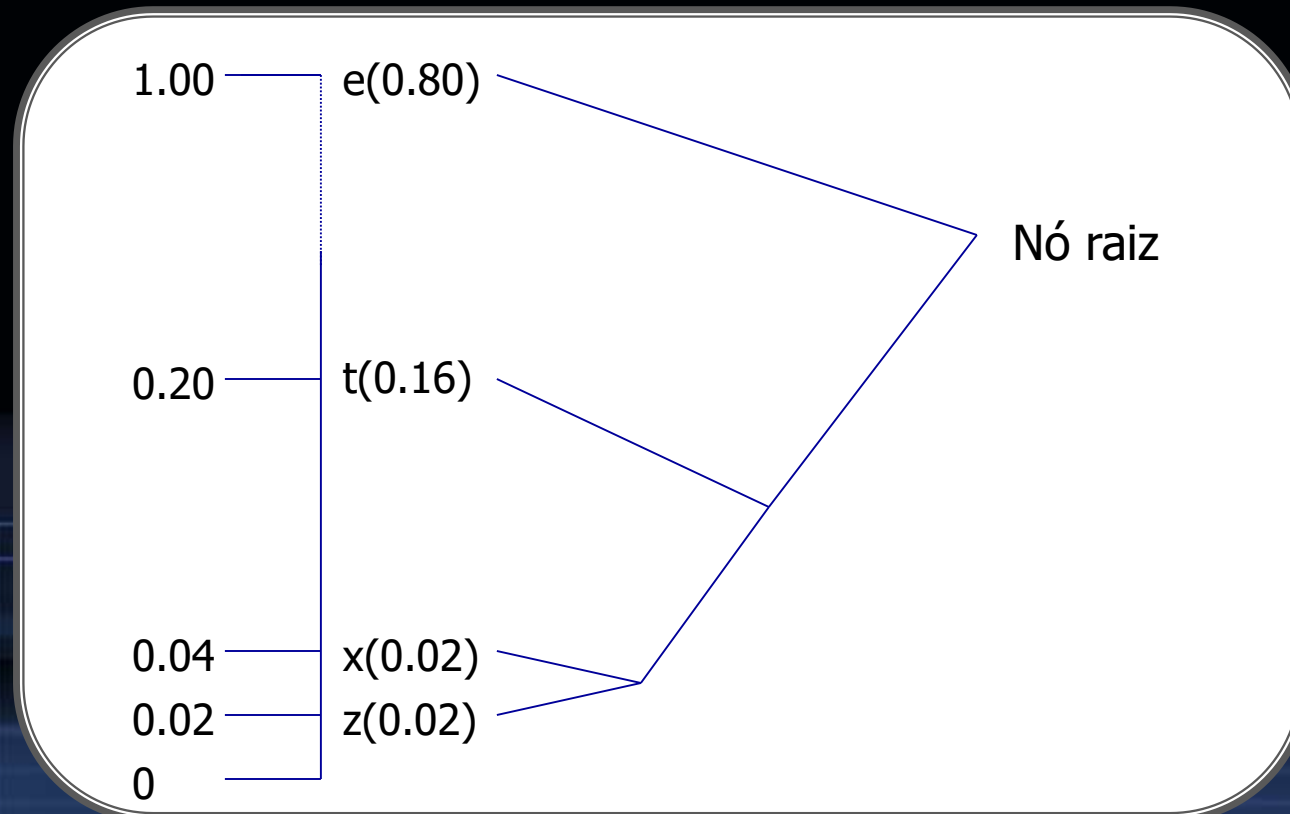


# Técnicas de compressão sem perdas

## ■ Codificação de Huffman

### ▣ Geração dos códigos Huffman

- d) Repita b) e c) até que todos os símbolos sejam inseridos na árvore
  - último nó é chamado de raiz

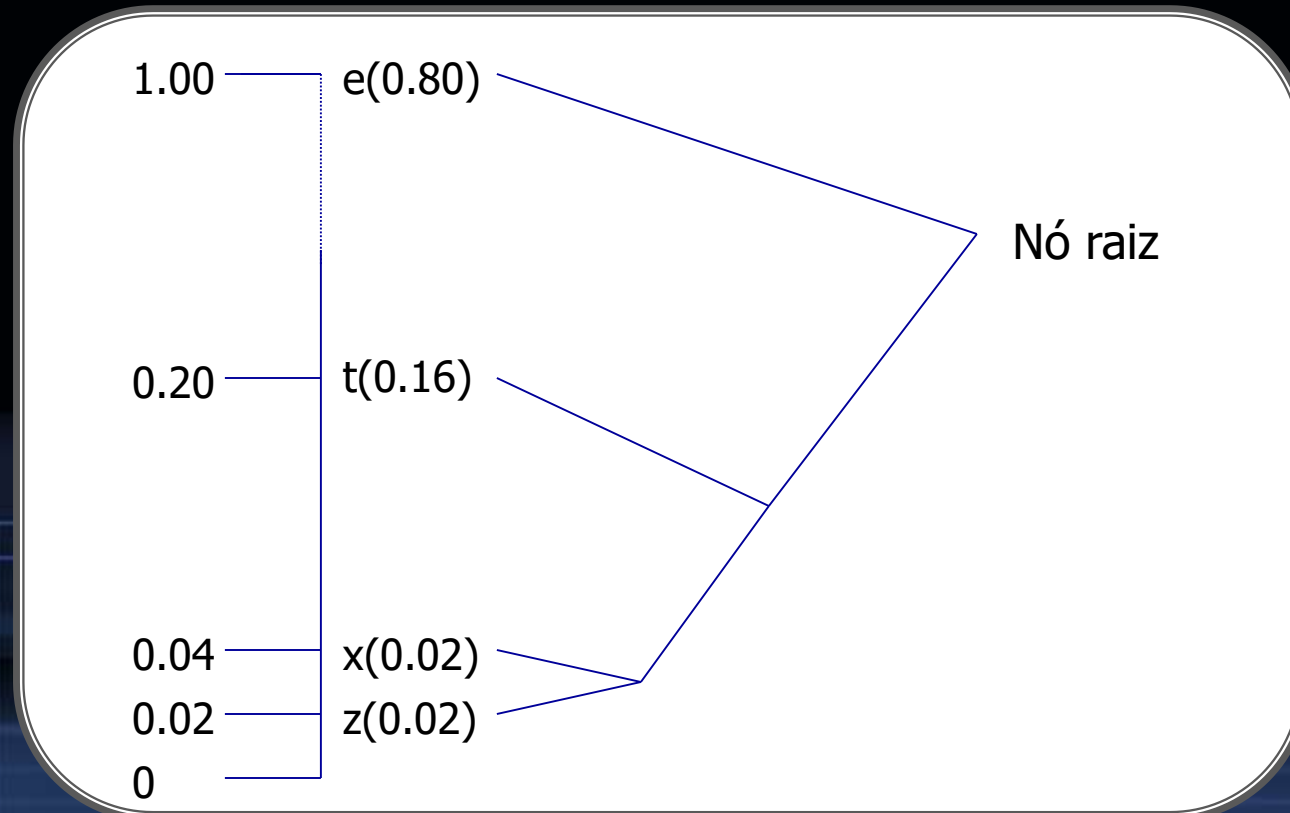


# Técnicas de compressão sem perdas

- **Codificação de Huffman**

- Geração dos códigos Huffman

- d) Repita b) e c) até que todos os símbolos sejam inseridos na árvore
  - último nó é chamado de raiz

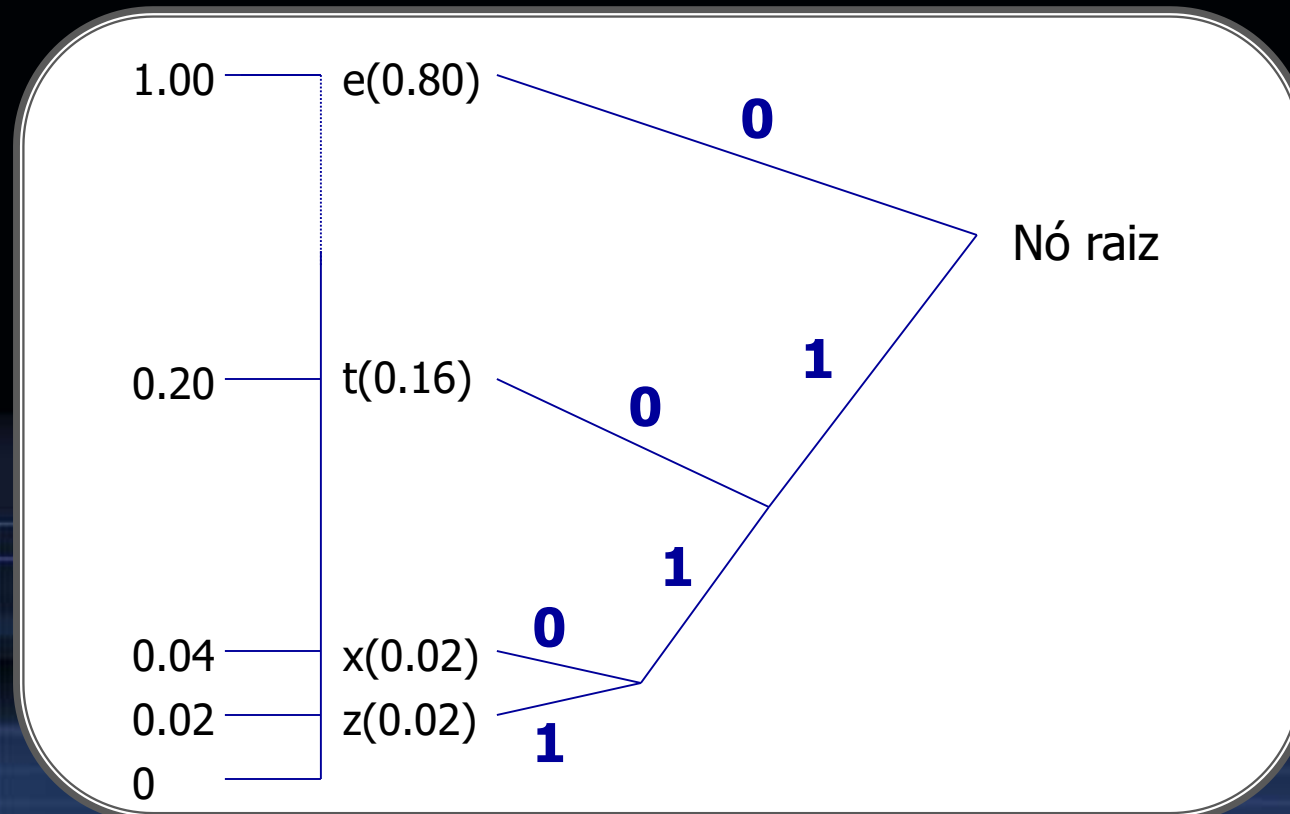


# Técnicas de compressão sem perdas

## ■ Codificação de Huffman

### ▣ Geração dos códigos Huffman

- e) Partindo do nó raiz, atribua bit 0 ao ramo de maior prioridade e bit 1 ao ramo de menor prioridade de cada nó



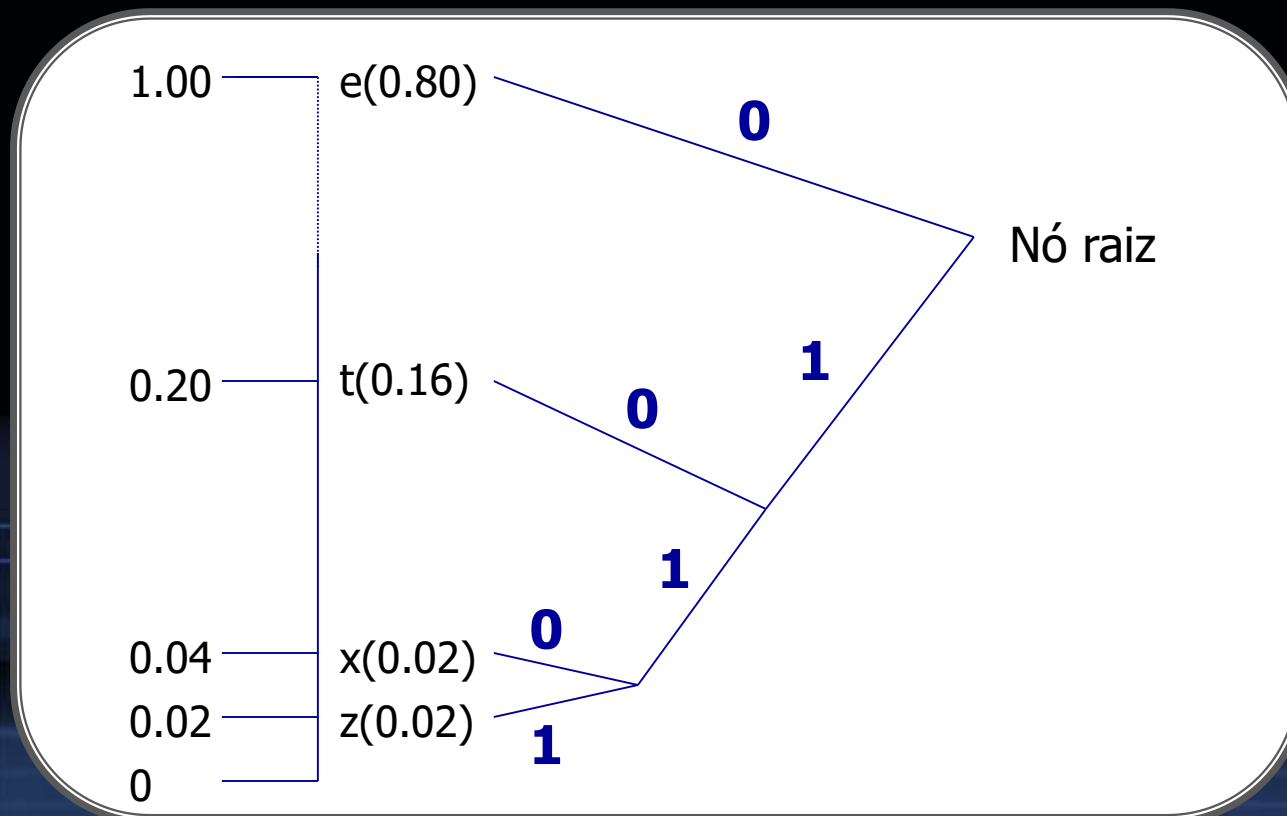


# Técnicas de compressão sem perdas

## ■ Codificação de Huffman

### ▣ Geração dos códigos Huffman

- f) Código para cada símbolo é obtido montando códigos ao longo do caminho entre nó raiz ao símbolo



Codebook

Símbolo	Código
e	0
t	10
x	110
z	111

# Técnicas de compressão sem perdas

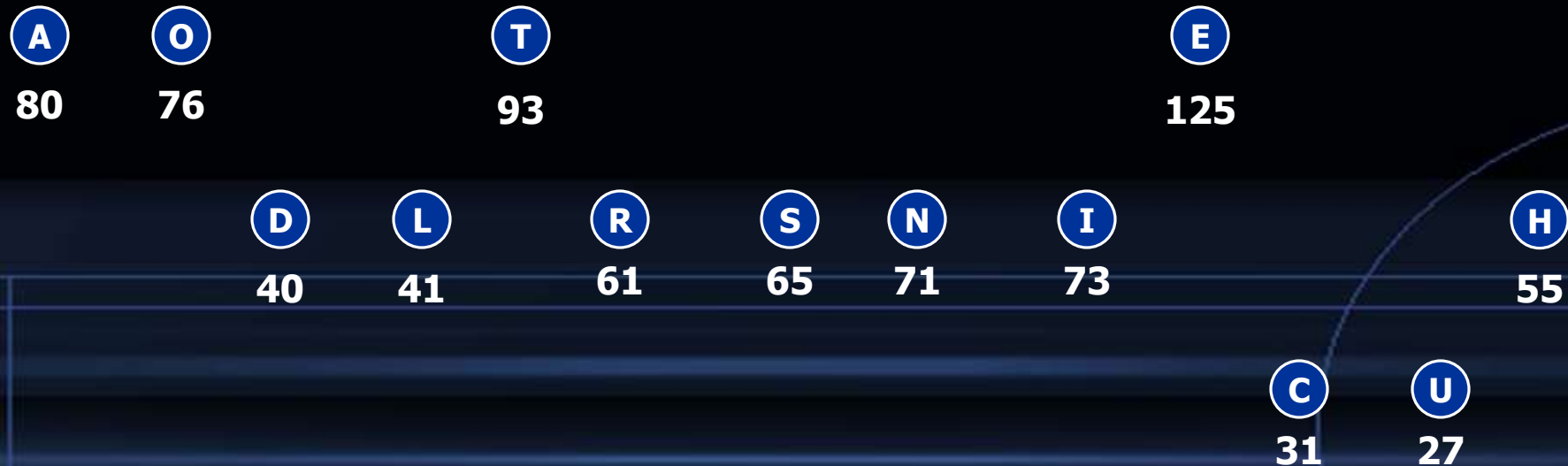


- **Outro exemplo de codificação de Huffman**
  - Frequências dos caracteres
  - Gere a tabela de Huffman para o arquivo.

Char	Freq	Fixo
E	125	0000
T	93	0001
A	80	0010
O	76	0011
I	73	0100
N	71	0101
S	65	0110
R	61	0111
H	55	1000
L	41	1001
D	40	1010
C	31	1011
U	27	1100
Total	838	4.00

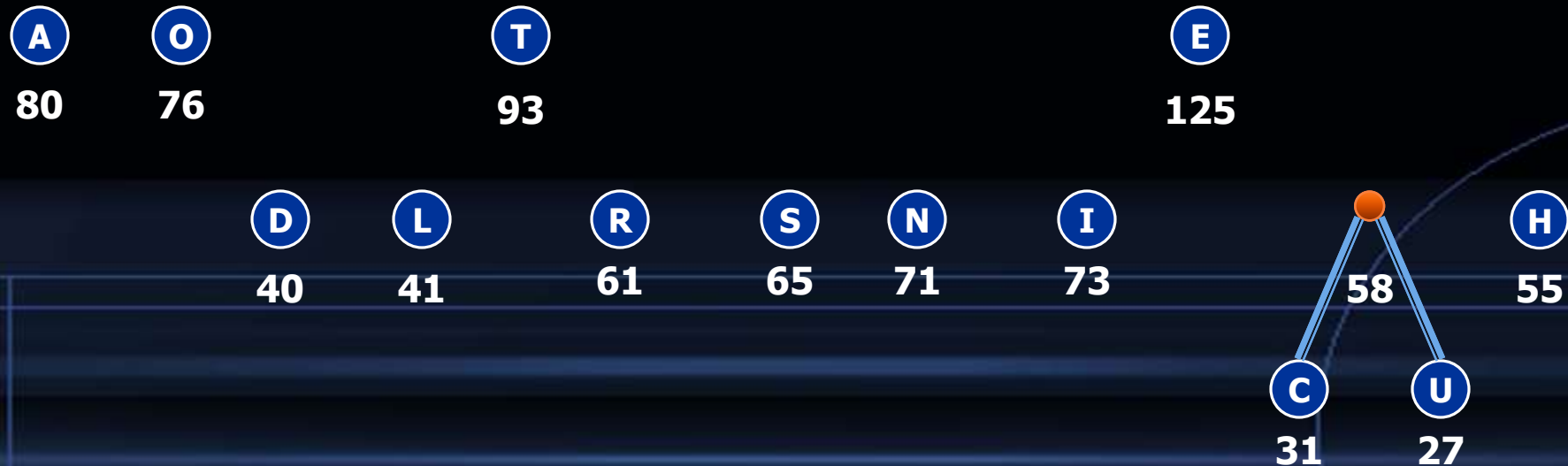
# Técnicas de compressão sem perdas

- **Outro exemplo de codificação de Huffman**



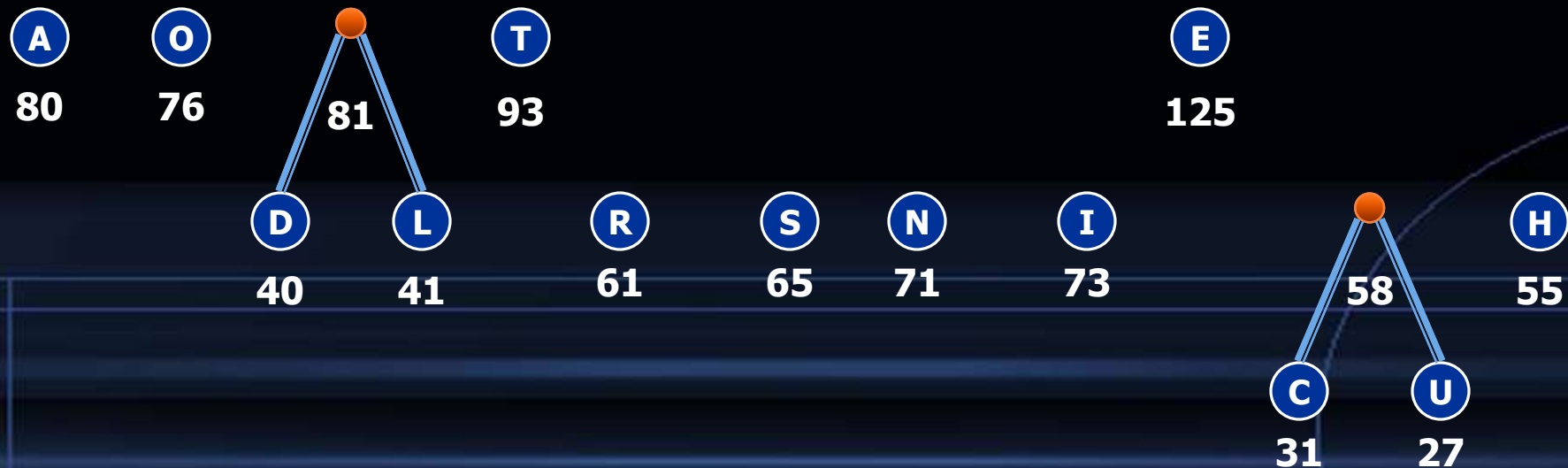
# Técnicas de compressão sem perdas

- **Outro exemplo de codificação de Huffman**



# Técnicas de compressão sem perdas

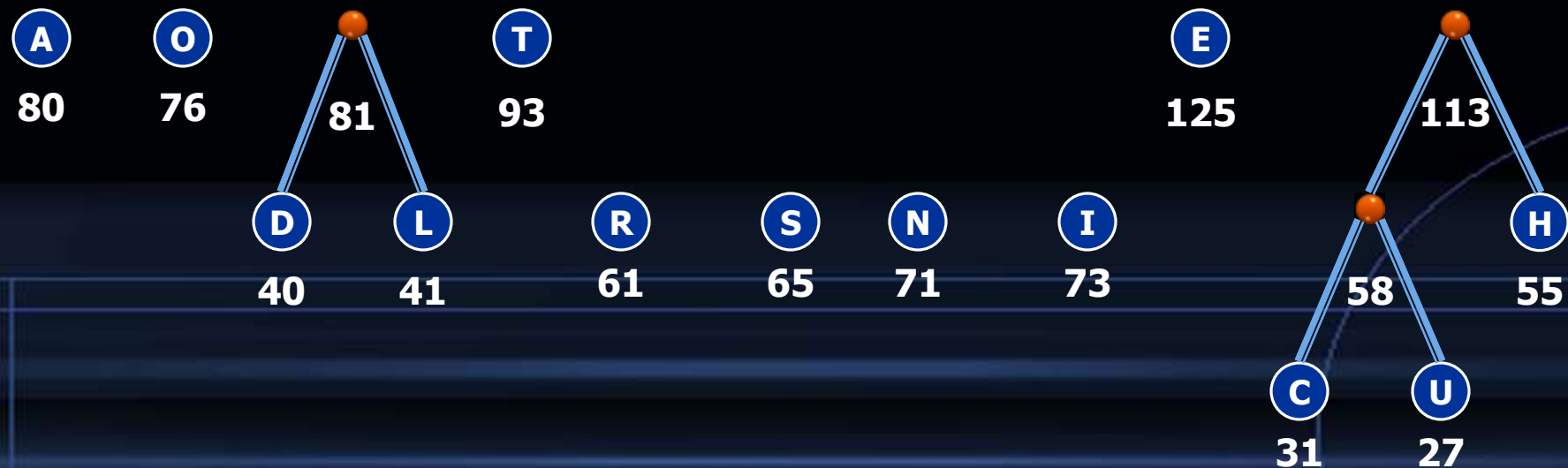
- **Outro exemplo de codificação de Huffman**





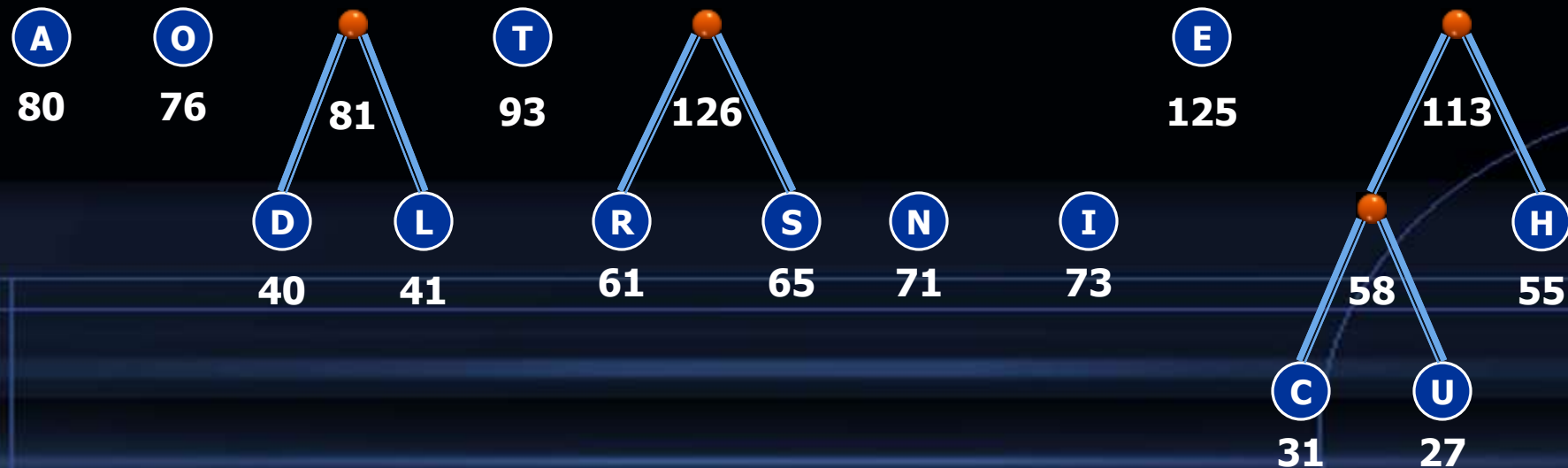
# Técnicas de compressão sem perdas

- **Outro exemplo de codificação de Huffman**



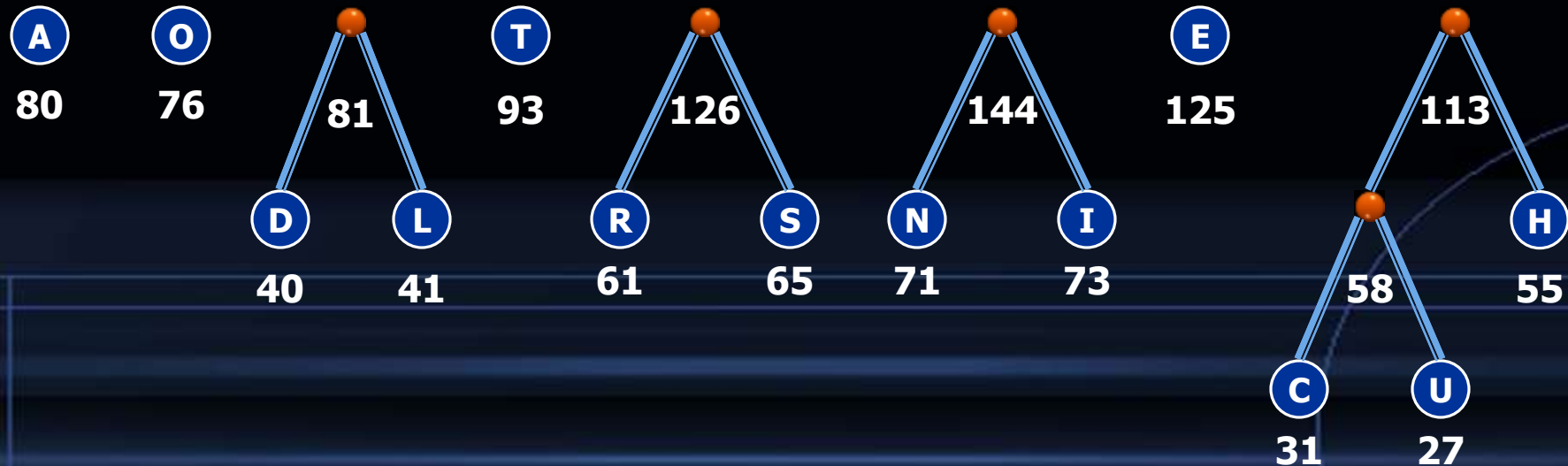
# Técnicas de compressão sem perdas

- **Outro exemplo de codificação de Huffman**



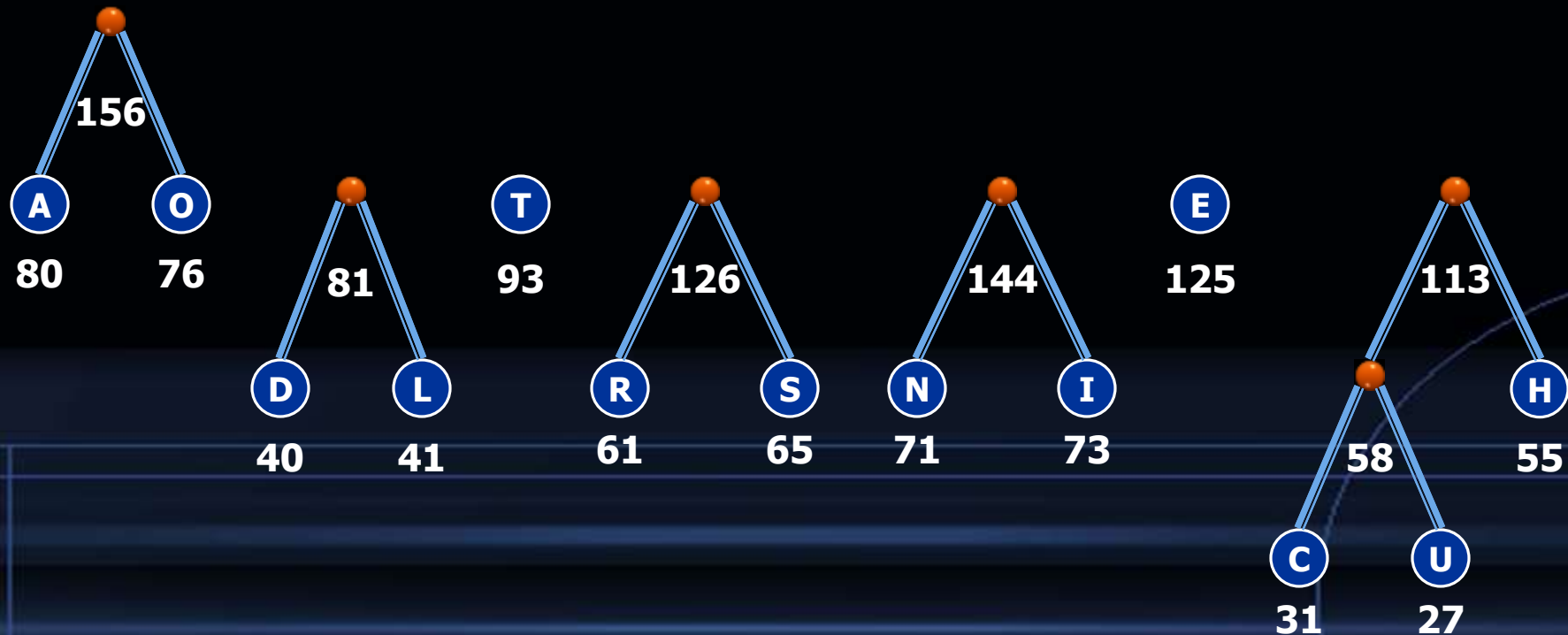
# Técnicas de compressão sem perdas

- **Outro exemplo de codificação de Huffman**



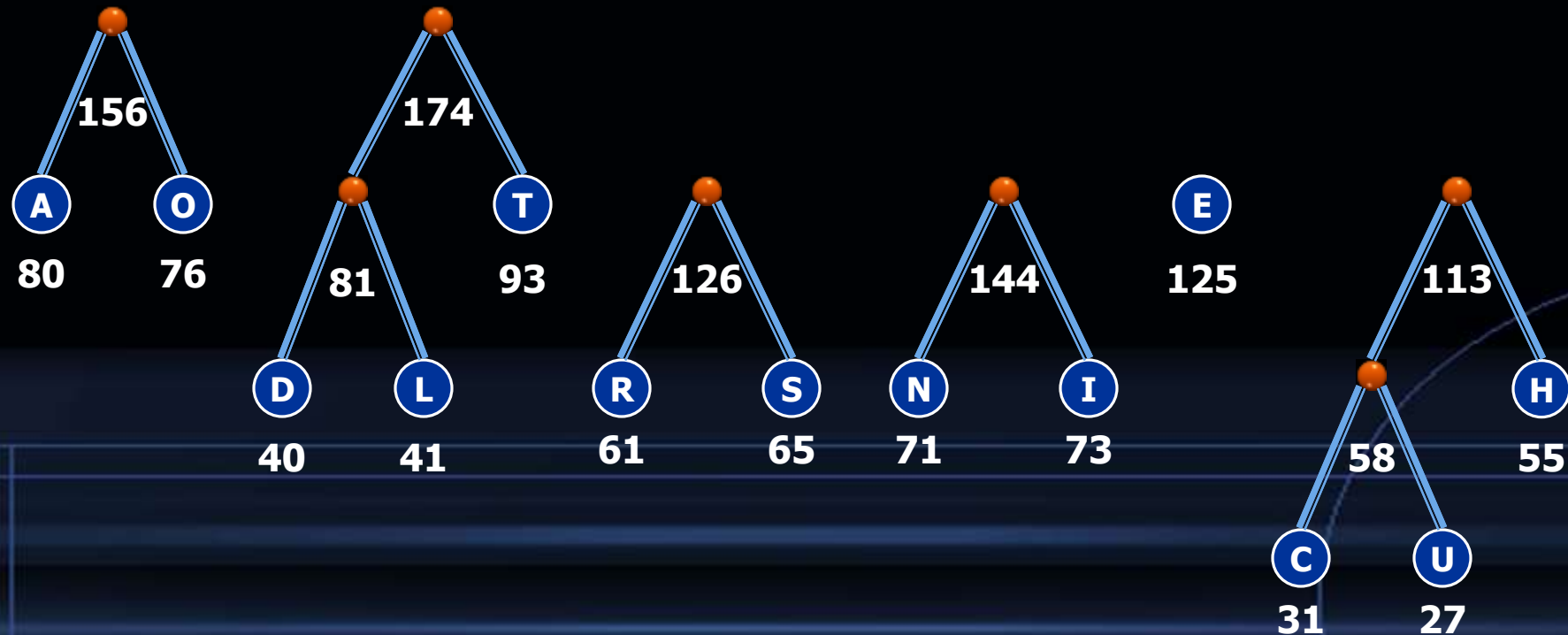
# Técnicas de compressão sem perdas

- **Outro exemplo de codificação de Huffman**



# Técnicas de compressão sem perdas

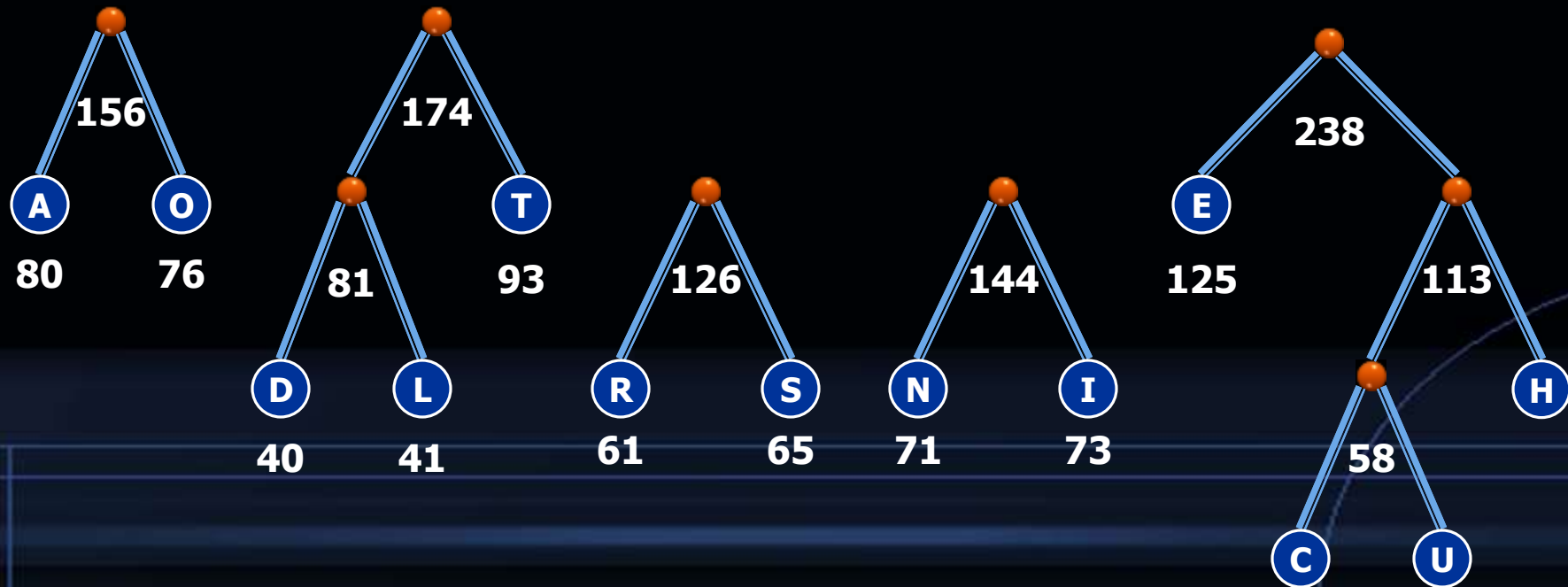
- **Outro exemplo de codificação de Huffman**





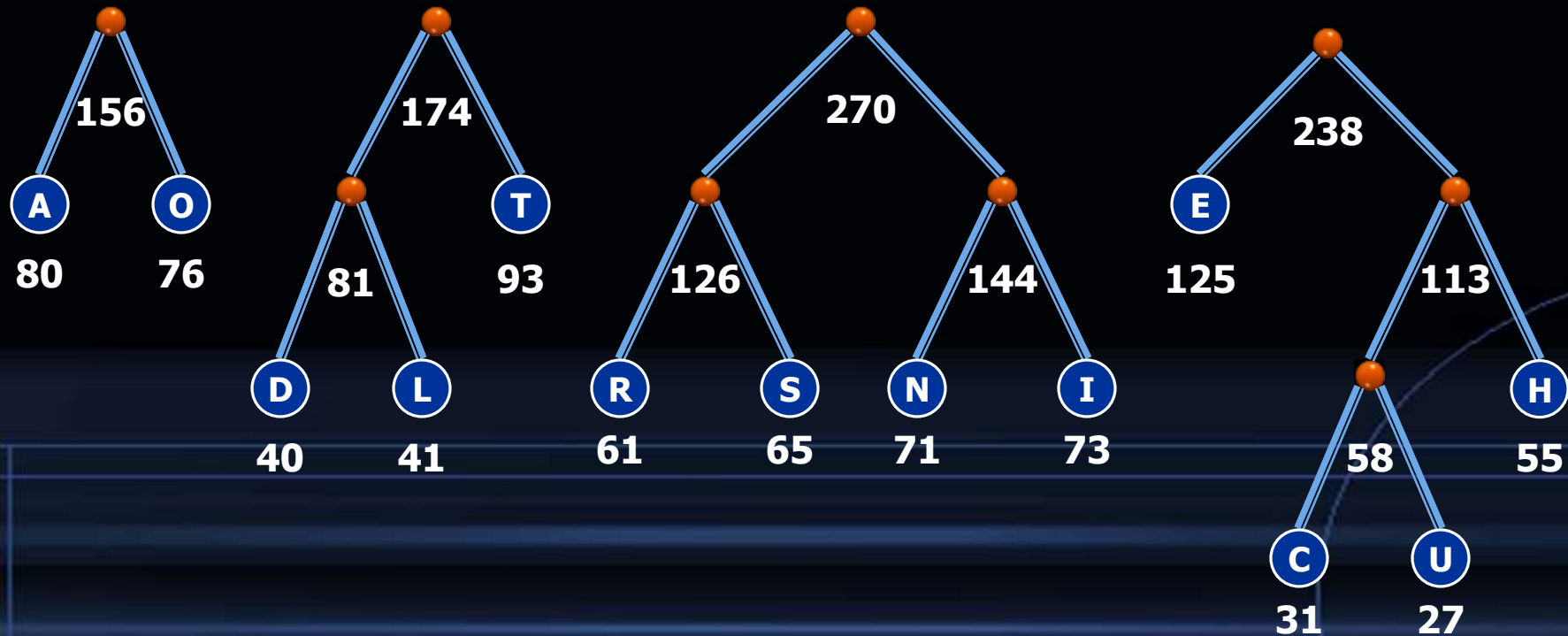
# Técnicas de compressão sem perdas

- **Outro exemplo de codificação de Huffman**



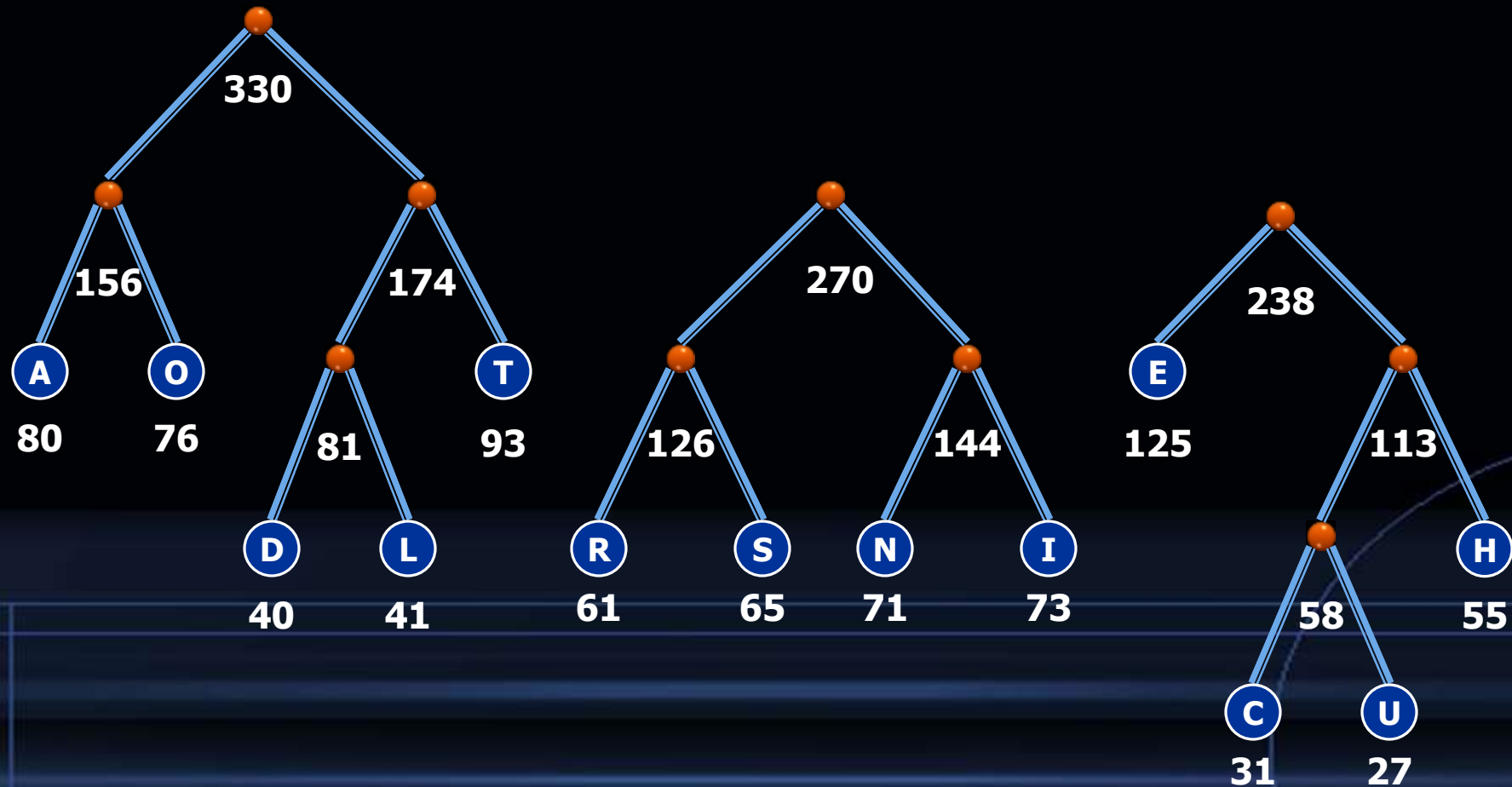
# Técnicas de compressão sem perdas

- **Outro exemplo de codificação de Huffman**



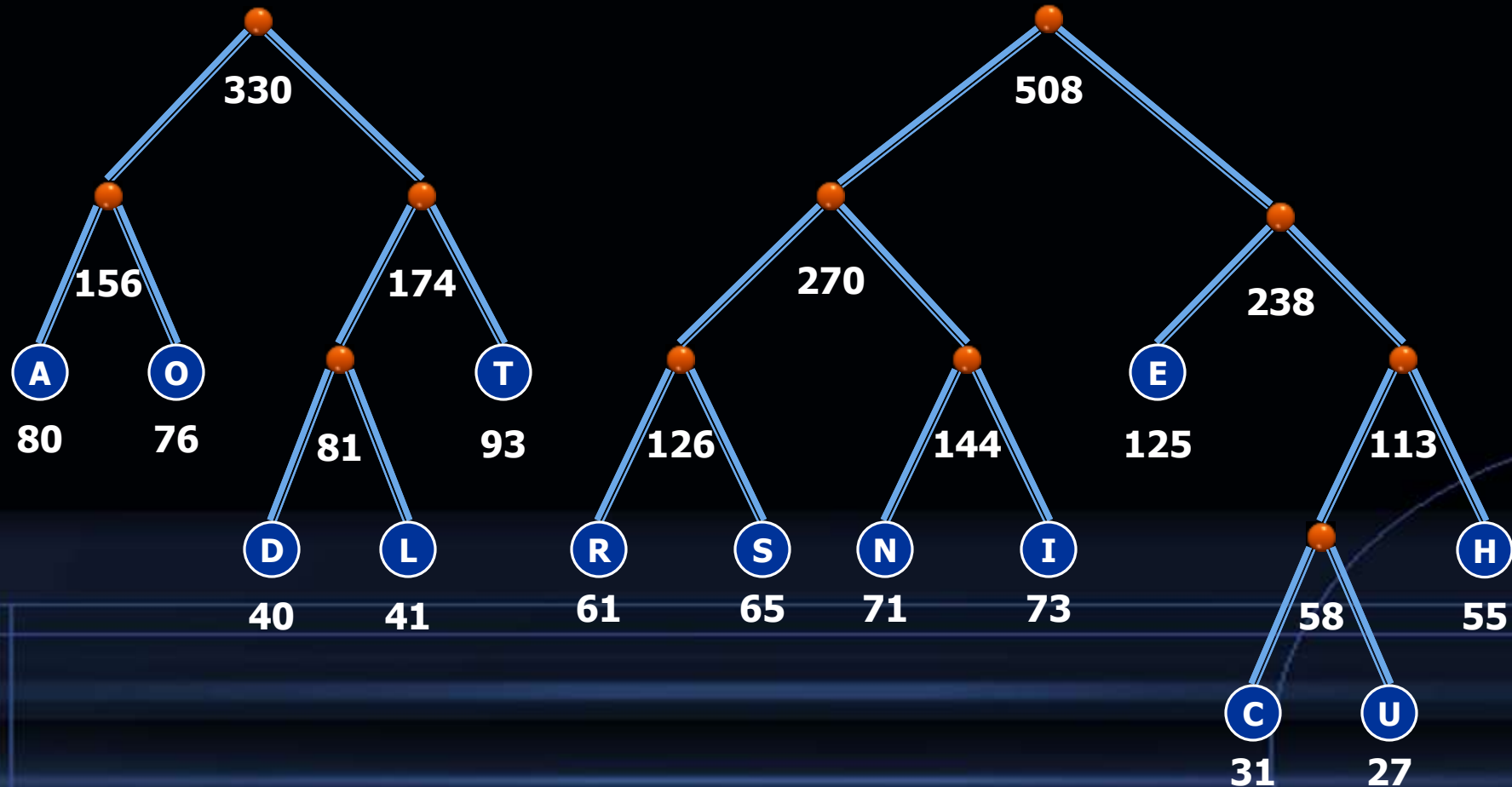
# Técnicas de compressão sem perdas

- **Outro exemplo de codificação de Huffman**



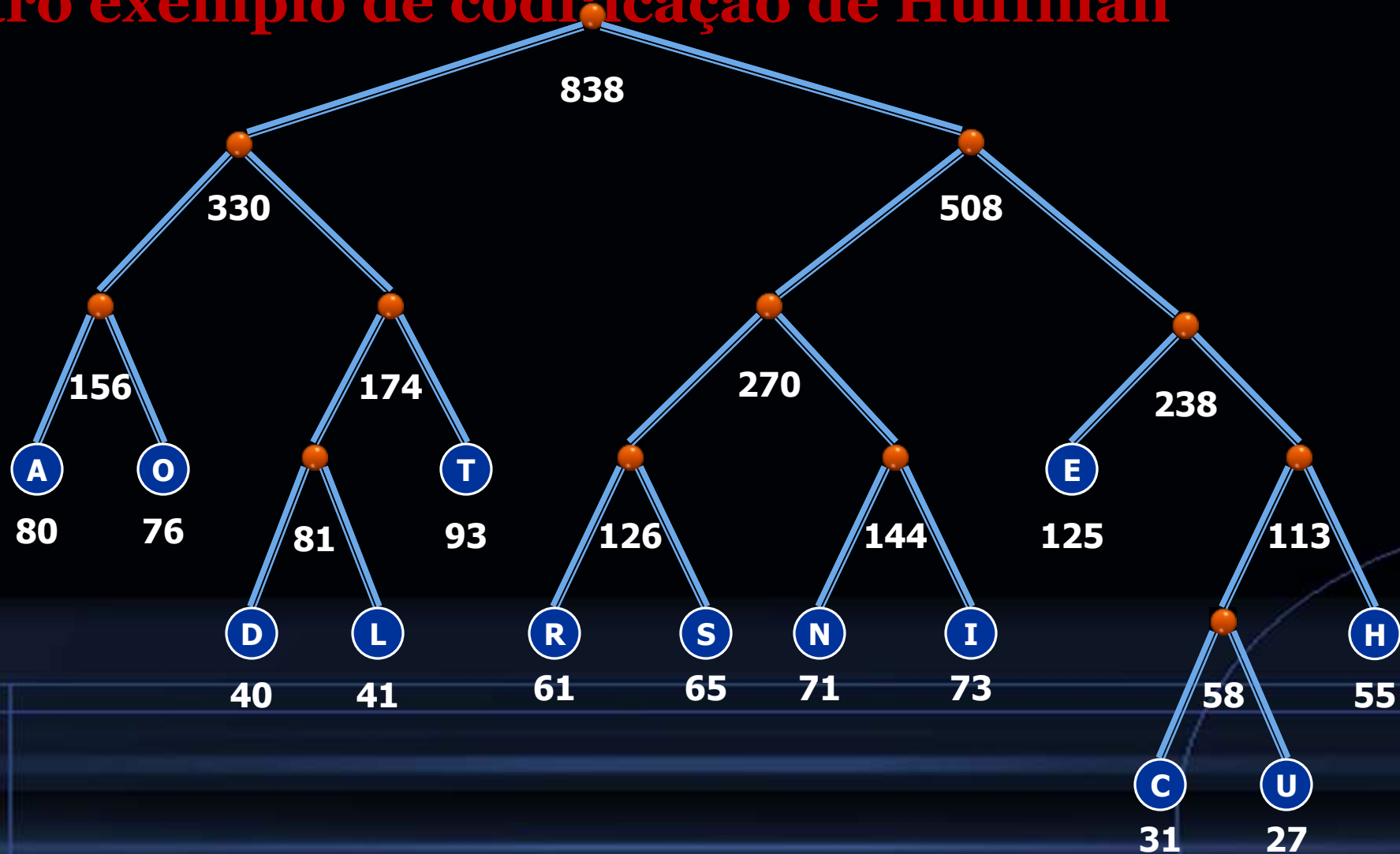
# Técnicas de compressão sem perdas

- **Outro exemplo de codificação de Huffman**



# Técnicas de compressão sem perdas

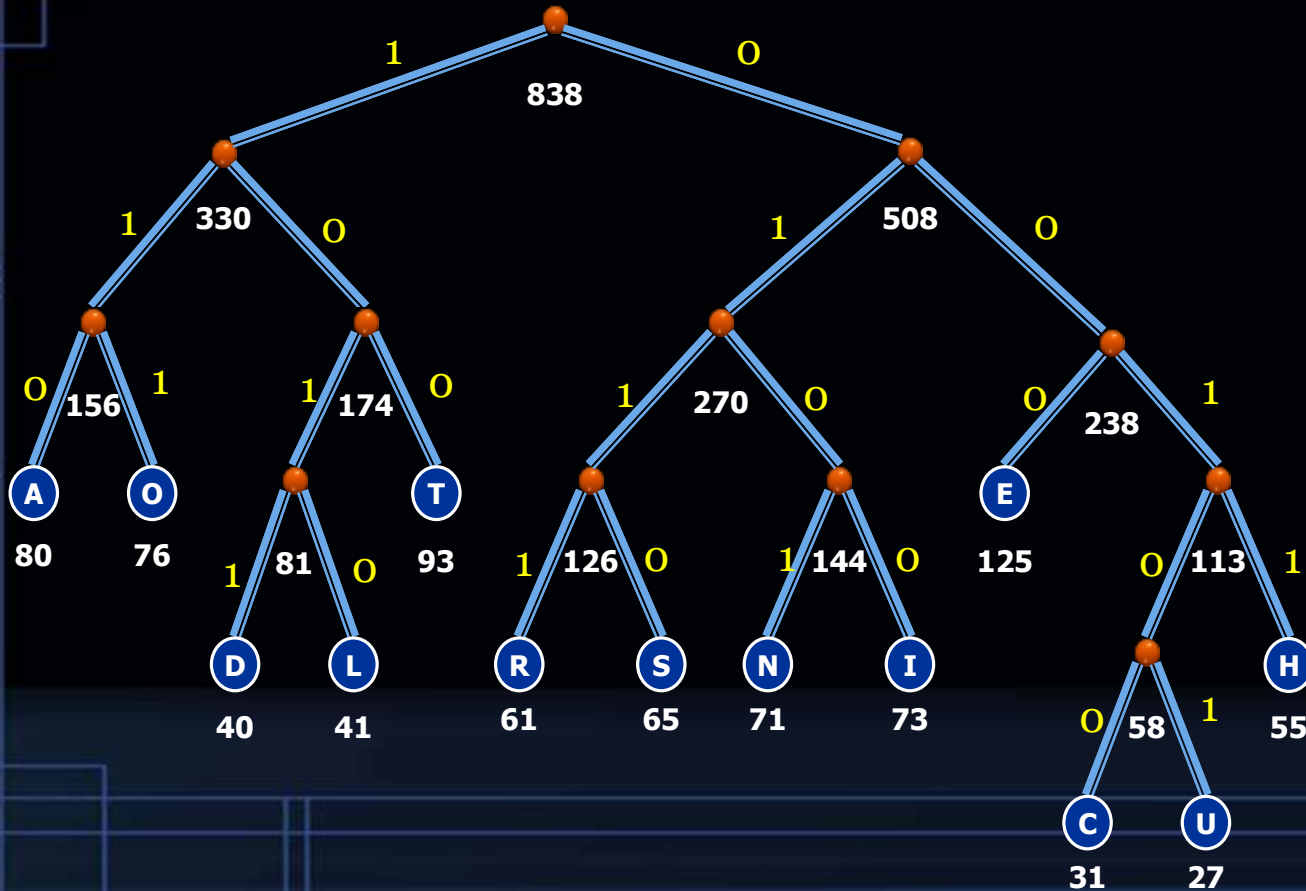
- **Outro exemplo de codificação de Huffman**





# Técnicas de compressão sem perdas

## Exemplo de codif. de Huffman



Char	Freq	Huff
E	125	000
T	93	100
A	80	110
O	76	111
I	73	0100
N	71	0101
S	65	0110
R	61	0111
H	55	0011
L	41	1010
D	40	1011
C	31	00100
U	27	00101
Total	838	3.62

# Técnicas de compressão sem perdas

- **Exemplo de codif. de Huffman**

Char	Freq	Fixo	Huff
E	125	0000	000
T	93	0001	100
A	80	0010	110
O	76	0011	111
I	73	0100	0100
N	71	0101	0101
S	65	0110	0110
R	61	0111	0111
H	55	1000	0011
L	41	1001	1010
D	40	1010	1011
C	31	1011	00100
U	27	1100	00101
Média		4.00	3.62
Total	383	3352	3036

# Técnicas de compressão sem perdas



- **Codificação de Huffman**
  - Operação computacional mais custosa na codificação
  - No decodificador
    - realiza uma simples verificação na tabela de Huffman
      - tabela de Huffman é parte do fluxo de dados ou é conhecida pelo decodificador
  - Tabelas de Huffman padrões são muito usadas
    - usada para vídeo em tempo-real
    - tabelas são conhecidas pelo codificador e decodificador
      - codificação e decodificação são mais rápidas
    - desvantagem: tabelas padrões obtém fator de compressão um pouco menores
      - elas não são necessariamente ótimas

# Técnicas de compressão sem perdas



## ■ Huffman - otimalidade

- Huffman é ótimo para codificação símbolo-a-símbolo com uma distribuição de probabilidade conhecida, porém como trabalha com números binário inteiros há algumas redundâncias.
- Ainda assim, é garantido que:
  - $H(X) \leq Huffman(X) \leq H(X) + 1$

Entropia

Média de bits por símbolo após  
a codificação por Huffman

# Técnicas de compressão sem perdas



- **Outros métodos**

- É possível melhorar ainda mais a codificação de Huffman
  - Huffman adaptativo:
    - Constrói a árvore dinamicamente
    - Cálculo das probabilidades são dinâmicas com base nas frequências recentes na sequência de símbolos, e altera a estrutura da árvore para atualizar probabilidades estimadas.
- Estado-da-arte: Codificação aritmética!



# Pontos Importantes

## RLE e Codif. de Huffman

- Entender o princípio geral, vantagens e desvantagens



**UFSC**

**Ciências da  
Computação**

# **CAP 3. COMPRESSÃO DE DADOS MULTIMÍDIA**

**INE5431 Sistemas Multimídia**  
**Prof. Roberto Willrich (INE/UFSC)**  
**roberto.willrich@ufsc.br**

**Aula 4: Técnicas de Codificação  
sem perdas: (A)DPCM e LZ\***

# Cap 3. Compressão de Dados Multimídia



## ■ Conteúdo:

- Necessidade de compressão
- Entropia: Teorema da codificação da fonte
- Princípios da compressão
- Classificação das técnicas de compressão
- Medição do desempenho de compressão
- **Técnicas de compressão sem perdas**
  - RLE, Huffman, (A)DPCM, LZ\*
- Técnicas de compressão de áudio, vídeo e imagens
- Padrões de compressão multimídia
  - JPEG, MPEG, MPEG-4, H.261, H.263

# Técnicas de compressão sem perdas



## ■ **Codificação Predictiva**

### ▣ DPCM (PCM diferencial)

- Técnica mais simples de codificação preditiva
- Compara símbolos adjacentes e apenas erros de predição são quantizados e codificados
  - Exemplo ilustrativo:
    - Original (amostras de 8bits)
      - 23, 24, 26, 25, 27 ( $8 \times 5 = 40$  bits)
    - Compactado com função de predição  $a_i = a_{i-1} + \text{erro}$ 
      - 23, +1, +2, -1, +2
  - Erro de predição tem uma alta probabilidade de ser menor que o valor sendo codificado
    - Erro pode ser expresso com uma quantidade menor de bits
    - No exemplo, usando 4 bits para codificar o erro, o tamanho será  $8 + 4 \times 4 = 24$  bits
  - Na descompressão
    - Função de previsão e erro são usados para restaurar o dado original

# Técnicas de compressão sem perdas



- **Codificação Predictiva**

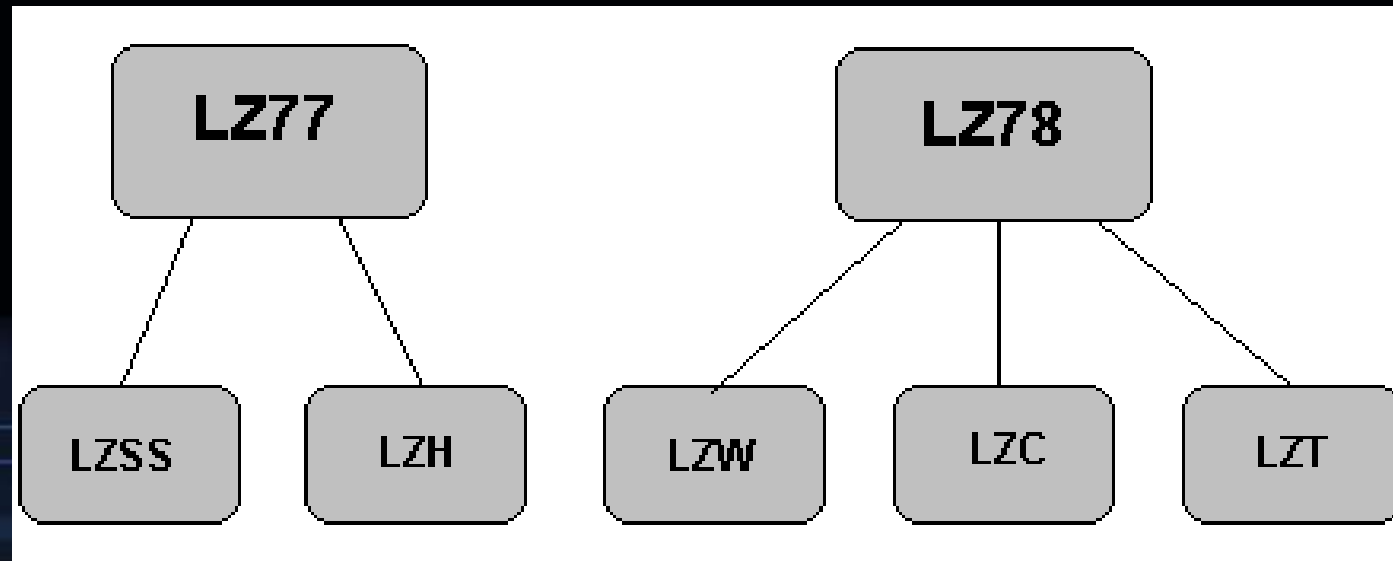
- ▣ ADPCM (DPCM Adaptativo)

- Existem várias maneiras de implementar ADPCM, a mais comum é variar o tamanho de passo de quantização representado pelos erros
      - Quando o erro é grande, o passo de quantização é maior (gerando perdas de qualidade)
      - Exemplo: se um passo preto-para-branco for detectado, pode-se aumentar o passo de quantificação antes deste passo chegar



# Técnicas de compressão sem perdas

- **Lempel-Ziv (LZ)**
  - Algoritmos de codificação baseada em dicionário
  - Proposta no final dos anos 70, Jacob Ziv e Abraham Lempel
    - Muitas variantes com objetivo de solucionar limitações das versões originais



# Técnicas de compressão sem perdas



- **Codificação derivadas do Lempel-Ziv (LZ): Aplicações**
  - UNIX Compression
    - O algoritmo LZC 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).
  - Zip e o gzip usam uma variante do LZ77 combinada com Huffman estático.
  - ARJ usa a codificação de Huffman e o algoritmo LZSS.
  - WINRAR usa o LZ77 e Huffman.
  - WINZIP entre outros algoritmos usa o LZW.

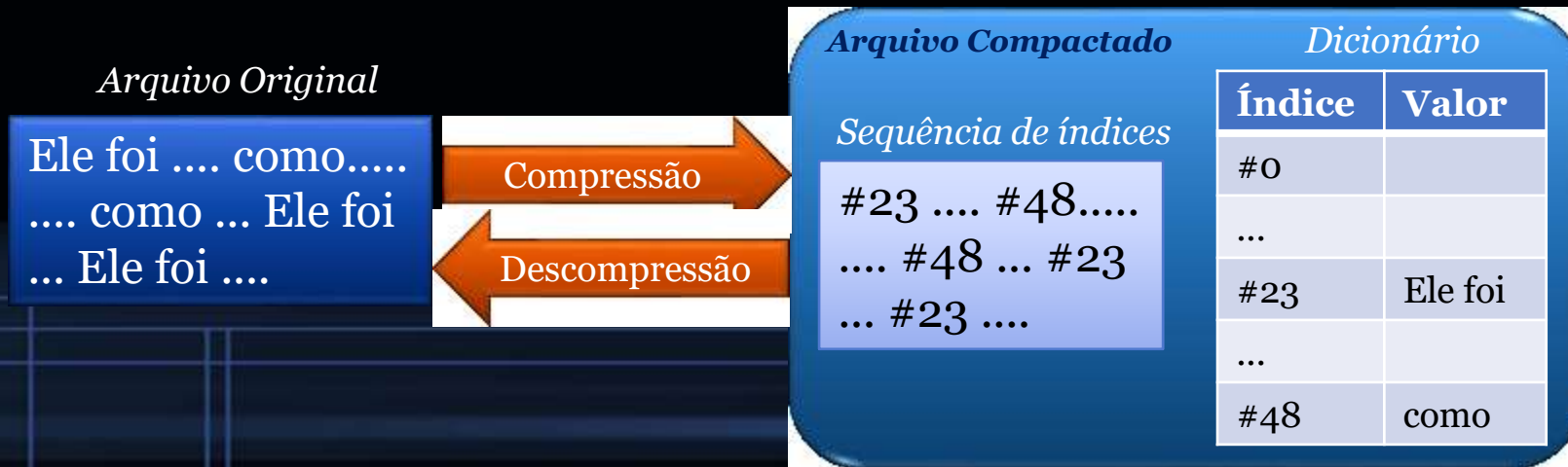
# Técnicas de compressão sem perdas



- **Codificações derivadas do Lempel-Ziv (LZ)**
  - Explora a redundância de dados -> a repetição de padrões de símbolos no arquivo
    - Baseada na construção de um dicionário de símbolos (grupos de um ou mais símbolos) a partir do fluxo de entrada
  - Ilustração em um arquivo de texto
    - Quando uma nova “frase” é encontrada
      - a máquina de compressão adicionada a “frase” no dicionário
      - um token que identifica a posição da “frase” no dicionário substitui a frase no documento
    - Se a “frase” já foi registrada
      - ela é substituída pelo token de sua posição no dicionário

# Técnicas de compressão sem perdas

- **Codificações derivadas do Lempel-Ziv (LZ)**
  - Explora a redundância de dados -> a repetição de padrões de símbolos no arquivo
    - Baseada na construção de um dicionário de símbolos (grupos de um ou mais símbolos) a partir do fluxo de entrada
  - Exemplo ilustrativo



# Técnicas de compressão sem perdas

- **Codificações derivadas do Lempel-Ziv (LZ)**
  - Exemplo do poder da codificação
    - Arquivo original de 10000 caracteres (8 bits/caractere)
      - arquivo requer 80000 bits para representá-lo
    - Assumindo que arquivo tem 2000 palavras ou frases das quais 500 são diferentes
      - necessitamos 9 bits como token para identificar cada palavra ou frase
      - precisamos de  $9 \times 2000$  bits para codificar o arquivo
    - obtemos uma taxa de compressão de 4,4
      - Dicionário armazenando todas as frases únicas deve ser armazenado também

Arquivo Original

Ele foi .... como.....  
.... como ... Ele foi  
... Ele foi ....

Compressão

Descompressão

Arquivo Compactado

Sequência de índices

#23 .... #48.....  
.... #48 ... #23  
... #23 ....

Dicionário

Índice	Valor
#0	
...	
#23	Ele foi
...	
#48	como



# Técnicas de compressão sem perdas



- **LZW e o formato de imagem GIF**
  - GIF utiliza a técnica LZW
  - GIF é um dos formatos de armazenamento de imagens 256 cores sem perdas
    - imagens com um máximo de 256 cores
    - ao converter imagem true color, com 24 bits/pixel, para o formato GIF, estamos perdendo grande parte da informação de cor
  - Taxas de compressão não são grandes
    - em geral 4:1
  - Extensão GIF89a permite
    - definir uma cor transparente
    - entrelaçamento
    - animação

# Técnicas de compressão sem perdas

- **LZW e o formato de imagem GIF**

- Extensão GIF89a permite
  - definir uma cor transparente
  - entrelaçamento
  - animação



Entrelaçado



Não Entrelaçado

# Técnicas

## ■ Cabeçalho

010 Editor - C:\Users\willrich\Dropbox\Cursos\INE\MM\midias\imagens\Dsc02273.gif

File Edit Search View Format Scripts Templates Tools Window Help

Startup Dsc02273.gif X

Edit As: Hex Run Script Run Template: GIF.bt

0123456789ABCDEF

0310h: 00 00 00 05 C0 03 00 08 FE 00 BD 79 B3 46 CD 5A ...Ä...p.~y'FIZ

0320h: 33 6F D2 04 2A 5C B8 D0 5A C3 81 03 1D 36 33 48 3oÖ.\*\,ðZÄ...63H

0330h: B1 22 35 6F 07 2F 0A D4 B8 F0 9B 40 73 E3 B6 8D ±"5o./,Ö,ð>@sÄq.

0340h: 1B E7 8D A4 39 81 E3 40 96 5C 19 D2 DB 49 96 0C .ç.49.â@-\,ÖÜI-

0350h: 63 CA 64 E8 11 A3 B7 9A 07 69 52 3B B8 CC E6 C1 cEdè.è.~.iR; ,iæÄ

0360h: 9F 0B C7 89 03 E9 0E 9D 3B 73 EE 46 82 4C 39 72 Ý.Ç.é...;siF,L9r

0370h: 64 C9 71 1E 6B 0A FC 79 B1 59 55 9F DE 76 26 A4 dÉq.k.üy±YUÿEv&K

0380h: 7A D1 63 53 92 60 05 E2 14 9B 95 E7 B2 B3 67 8D zÑcS'`.ä.>\*ç'g.

0390h: 2D 33 A6 D6 59 B3 67 70 E1 7E 83 EB ED D9 DC B9 -3;ÖY'gpá~feiÜÜ

03A0h: 76 E3 DE CD 9B 77 2F 5E BF 74 E5 F6 1D 0C F8 DB vâPî>w/^çtâö...øÜ

03B0h: DE C0 7D 0B F7 7D FB 8C 71 E3 67 75 23 03 35 6C ÈÀ).+;ûQqâgu# 51

03C0h: F7 66 D4 A8 5F 33 6B 1E 99 B4 33 E7 71 9E DD AD ÷fÔ`\_3k.'`3çqžÝ-

03D0h: 73 E7 EE DD 3B 79 EE E4 9D 4E 2D 6F 9E BC D6 AF sçîY;yiâ.N-ož4Ö

03E0h: 5D D7 93 57 AF 76 ED 7B B7 6D EB DE CD 1B 5F 6E j\*W`vi( .mëPî. n

03F0h: DC C0 EB 05 DF 77 0F DF BE E3 FD F6 F9 EB BC ÜÄè.âw.â%âýöüè .i

0400h: 39 73 E6 FF FE 2D 97 DE EF 1F BE 4A 8D 94 FF AB 9sæÿp--Pi.%J."ý«

0410h: 1E BD BB F7 E8 FE C0 FE 4B A7 EE AF B9 F9 E7 FB .~>÷èpÂpKSî`-ûçû

0420h: 98 2B 4F 7F BC BD FB 7B FB 8C D7 73 9F DC B9 F2 ~+O.~sû(ûè\*sÿÜ`ò

0430h: FA EE DF E7 CF 0F 9F F8 F1 FE FD ED D7 5E 71 FE úiBçî.Yøñpýi\*^qp

0440h: 19 57 60 81 F0 E1 93 A0 7F 01 0E 18 1F 82 C4 C9 .W'.ðá" .....ÄÈ

0450h: 17 21 70 0A 0A 37 5F 6E 16 E2 C6 DB 86 F3 D4 D3 .!p..7\_n.âÆÜ+óÖÖ

0460h: 21 6D 1D CE 56 DB 87 24 C2 E6 9A 6B B1 A9 A6 DA !m.ÍVÜ+çÄæsk±@;Ü

0470h: 3B A5 B5 C8 62 3B EE C0 38 4E 3B 33 8E C3 8E 8D ;¥uÈb;iâ@N;3žÄž.

0480h: 38 B2 C3 CE 37 3A 1A E6 23 37 3F FA 48 D9 90 77 8`Äî7;.æ#7?ûHÜ.w

0490h: 01 09 64 37 73 19 69 18 90 45 7E C3 A4 5D 4E DE ..d7s.i..E~ÄK]NÈ

04A0h: 15 A5 90 54 32 69 D8 8E 3D 0A 99 65 8F 59 12 A9 .¥.T2i0ž=.~e.Y.©

04B0h: 25 95 57 5A A9 E4 98 41 46 A9 17 5F 67 C6 25 98 %\*WZ@â`AF@. çÆ%~

04C0h: 33 6A B6 B9 CC 33 6F C6 E9 CC 9B D4 3C 53 27 9C 3jç'i3oEéî>Ö<S'æ

04D0h: 76 E2 B9 CC 5B 6F 36 E3 56 9E 76 39 73 D7 9D 78 vâ'i[o6ÄVžv9s\*.x

Template Results - GIF.bt

Name	Value	Start	Size
ushort Height	960	8h	2h
struct LOGICALSCREENDESCRIPTOR_PACKEDFIELDS PackedFields		Ah	1h
ushort BackgroundColorIndex	0	Bh	1h
ushort PixelAspectRatio	0	Ch	1h
struct GLOBALCOLORTABLE GlobalColorTable		Dh	300h
struct RGB rgb[256]		Dh	300h
struct DATA Data		30Dh	CDFEAh
struct IMAGEDESCRIPTOR ImageDescriptor		30Dh	Ah
ushort ImageSeparator	44	30Dh	1h
ushort ImageLeftPosition	0	30Eh	2h
ushort ImageTopPosition	0	310h	2h
ushort ImageWidth	1280	312h	2h
ushort ImageHeight	960	314h	2h
struct IMAGEDESCRIPTOR_PACKEDFIELDS PackedFields		316h	1h
struct IMAGEDATA ImageData		317h	CDFE0h
ushort LZWMinimumCodeSize	8	317h	1h
struct DATASUBBLOCKS DataSubBlocks		318h	CDFDfh

Selected: 768

Selected: 6 b



# Técnicas de compressão sem perdas



- **LZW e o formato de imagem GIF**
  - Algoritmo LZW do GIF era propriedade da Unisys
    - Era do domínio público e a Unisys resolveu passar a cobrar uma taxa pela sua utilização
    - Patentes estão espiradas desde 2006 (pode ser usado livremente)
  - Este motivo provocou a definição de uma alternativa válida ao formato GIF
    - formato PNG (Portable Network Graphics)
      - Suporta múltiplos níveis de transparência
      - Correção gama para ajuste da exibição da imagem às características do monitor
      - Entrelaçamento mais avançado que o GIF
      - suporta 48-bit truecolor ou 16-bit escalas de cinza
      - não suporta animação
      - usa os algoritmos LZ77 e de Huffman (DEFLATE)
    - Formatos MNG (Multiple-Image Network Graphics) e APNG
      - Extensões do PNG que suportam animações

# Pontos Importantes

## (A)DPCM e LZ\*

- Entender o princípio geral, vantagens e desvantagens

## GIF e PNG

- Saber comparar esses formatos





**UFSC**

**Ciências da  
Computação**

# **CAP 3. COMPRESSÃO DE DADOS MULTIMÍDIA**

**INE5431 Sistemas Multimídia**  
**Prof. Roberto Willrich (INE/UFSC)**  
**roberto.willrich@ufsc.br**

**Aula 5: Técnicas de compressão  
de áudio, vídeo e imagens**

# Compressão de Dados Multimídia



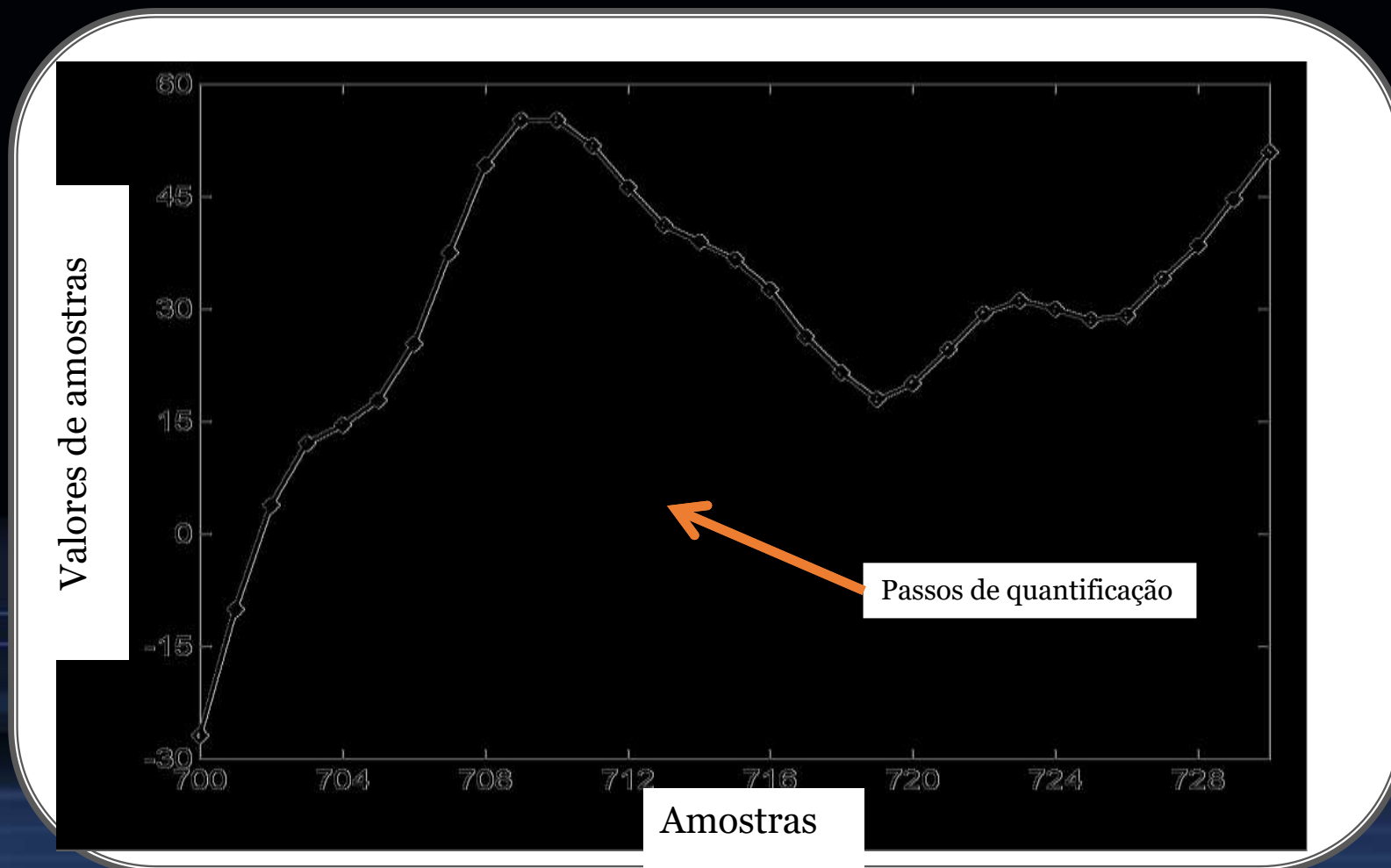
- **Conteúdo:**

- Necessidade de compressão
- Princípios da compressão
- Classificação das técnicas de compressão
- Medição do desempenho de compressão
- Técnicas de compressão sem perdas
  - RLE, Huffman, LZW (GIF), Codificação Preditiva
- **Técnicas de compressão de áudio, vídeo e imagens**
- Padrões de compressão multimídia
  - JPEG, MPEG, MPEG-4, H.261, H.263

# Técnicas de Compressão de Áudio Digital

- **Codificação PCM**

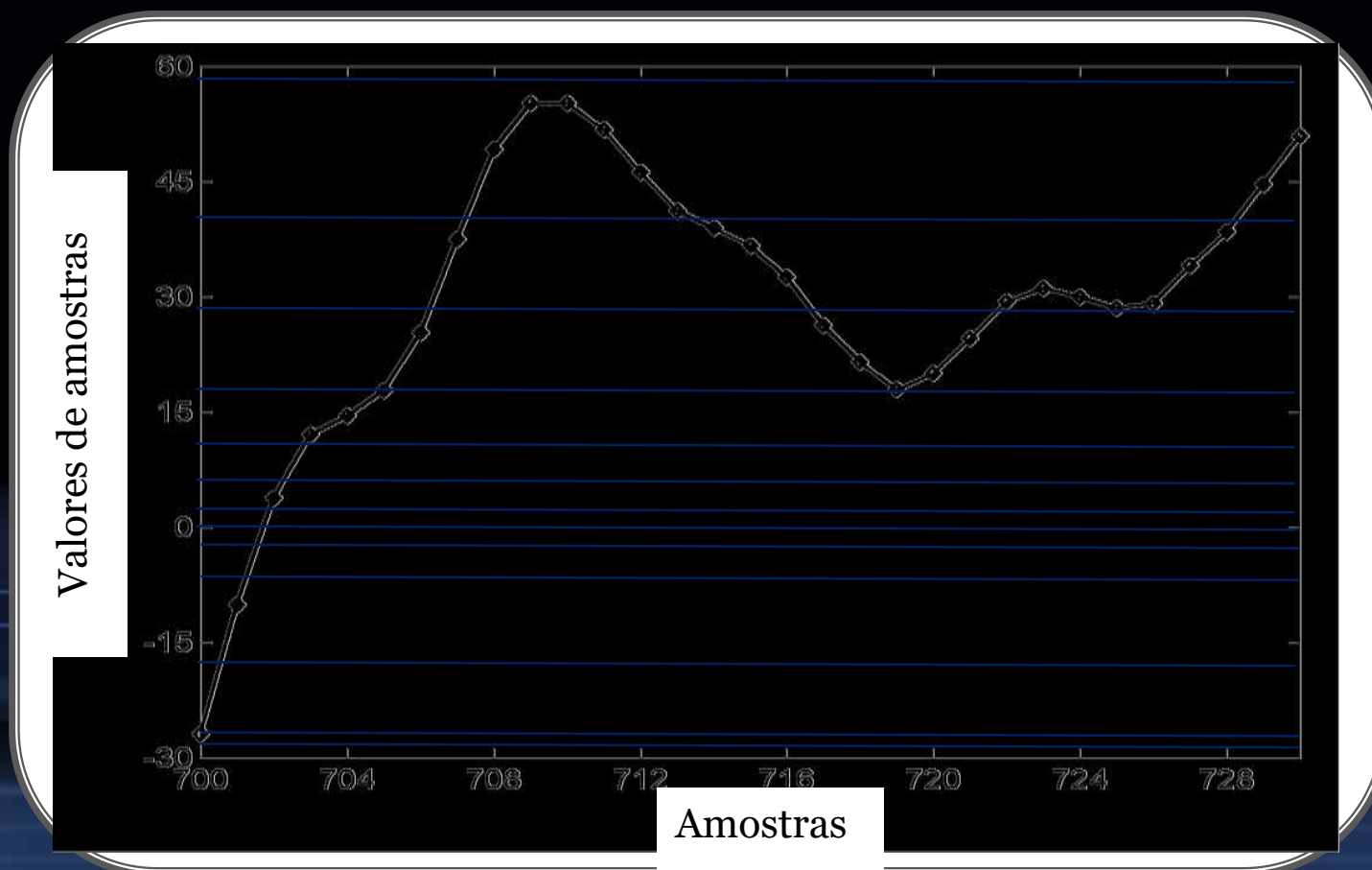
- Amostras são quantificadas com mesmo passo



# Técnicas de Compressão de Áudio Digital

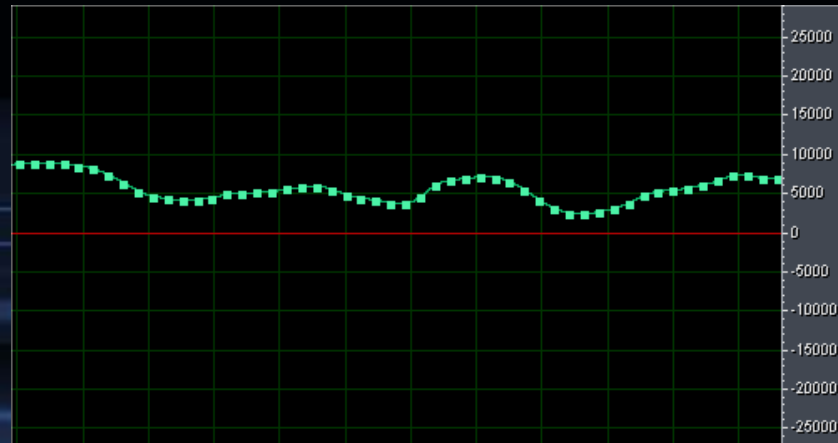
- **Codificação PCM não linear**

- Passo de quantificação aumenta com o aumento da amplitude do sinal
- Pode ser visto como compressão, pois melhora qualidade com a mesma taxa do PCM



# Técnicas de compressão de áudio

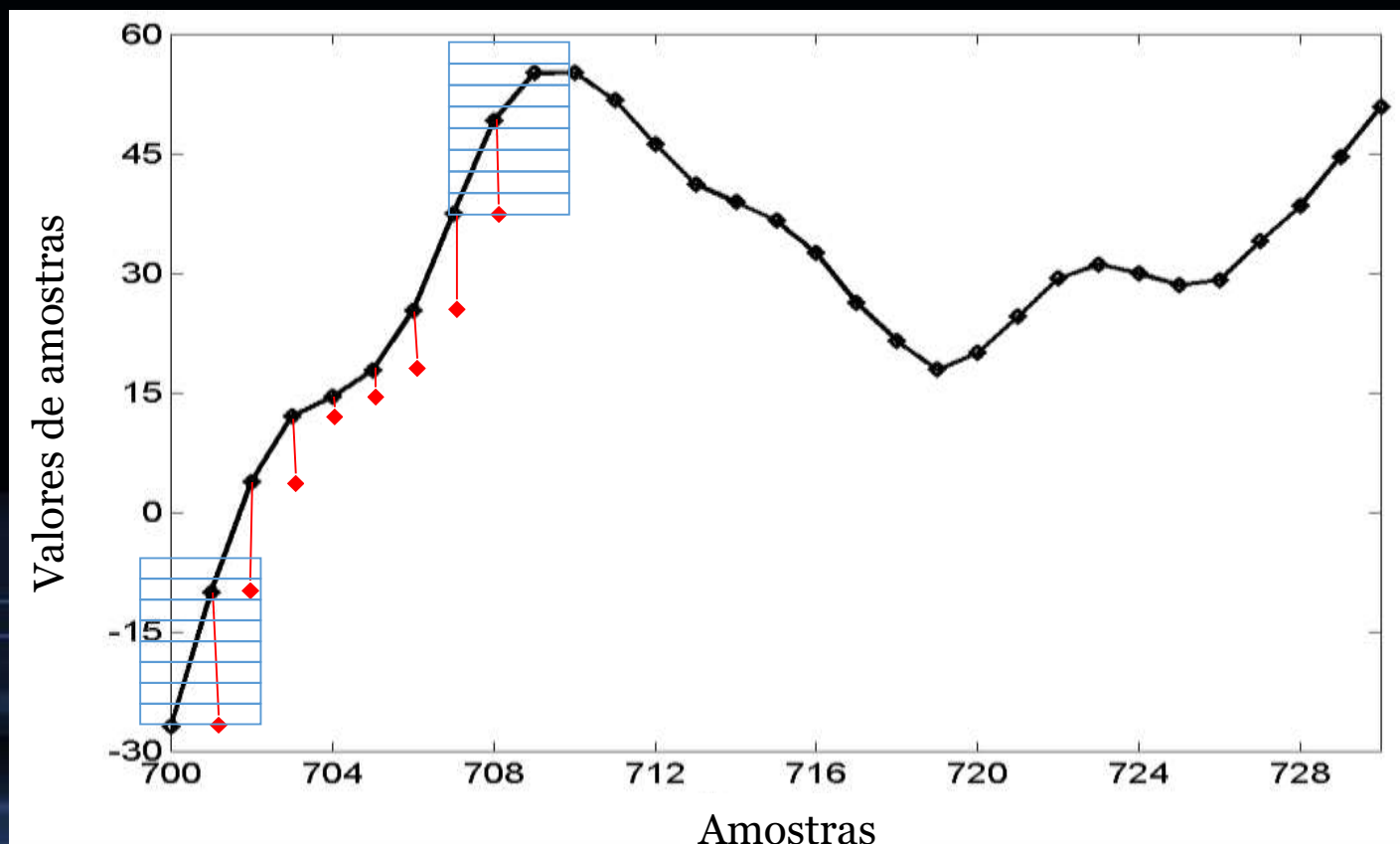
- **DPCM (Codificação Preditiva)**
  - Amostras adjacentes são similares:
    - próximo valor pode ser previsto baseado no valor atual
      - Exemplo ilustrativo:
        - Original (amostras de 8bits)
          - 23, 24, 26, 25, 27 ( $8 \times 5 = 40$  bits)
        - Compactado com função de predição  $a_i = a_{i-1} + \text{erro}$ 
          - 23, +1, +2, -1, +2 ( $8 + 4 \times 4 = 24$  bits)





# Técnicas de Compressão de Áudio Digital

- **Áudio DPCM: Quantização e codificação do erro de predição**
  - Exemplo de DPCM para áudio com função de predição  $a_i = a_{i-1} + \text{erro}$



# Técnicas de Compressão de Áudio Digital



- **No LPC (Linear Predictive Coding)**
  - Uma amostra de áudio é prevista com base nas amostras anteriores

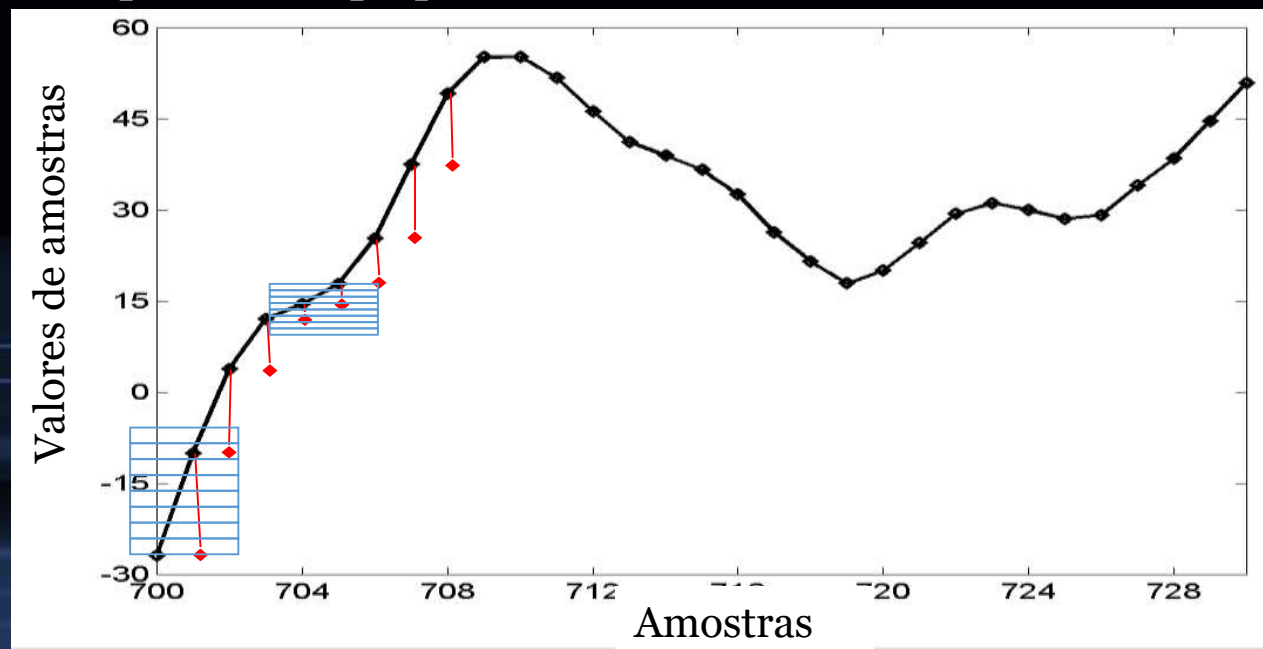
$$x[n] = \sum_{k=1}^P a_k x[n-k] + e[n]$$

- $x[n-k]$ : amostras anteriores
- $p$ : ordem do modelo
- $a_k$ : coeficiente de previsão
- $e[n]$ : erro de previsão

# Técnicas de Compressão de Áudio Digital

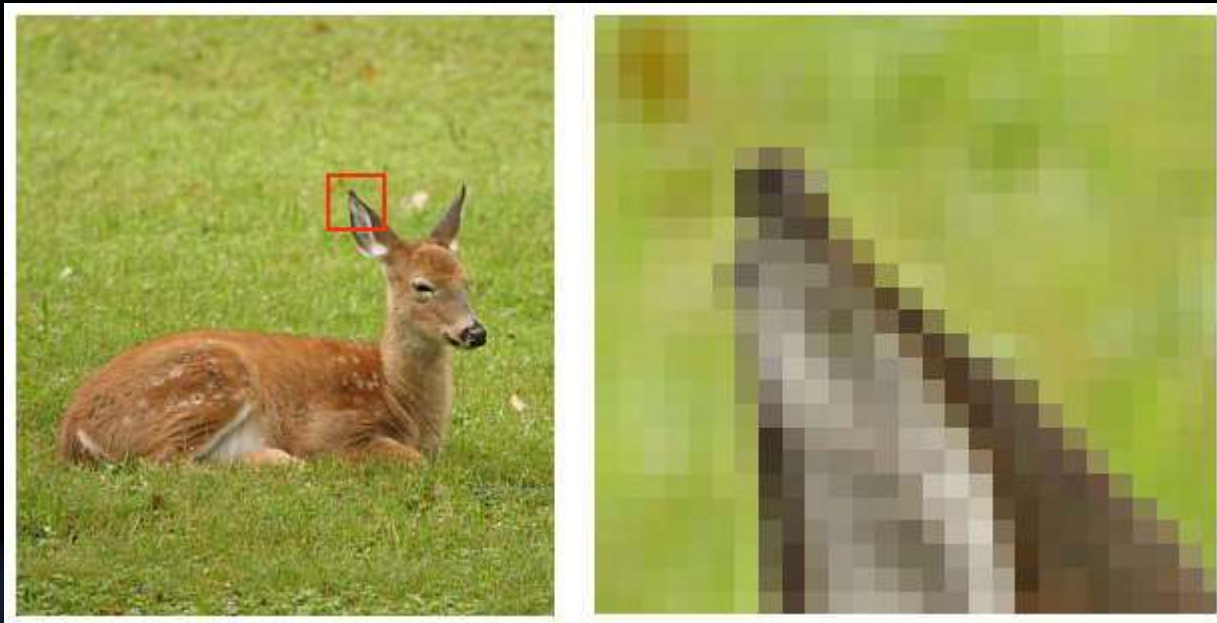


- **Codificação ADPCM (DPCM adaptativo)**
  - Existem várias maneiras de implementar ADPCM, a mais comum é variar o tamanho de passo representado pelos erros
    - tamanho passo de quantificação aumenta com o aumento da variação do sinal
      - Se o sinal passa bruscamente de uma tensão elevada a uma tensão baixa, o valor do passo será grande; ao contrário, se o sinal de entrada apresenta variações de tensão baixas, o tamanho do passo será pequeno



# Técnicas de Compressão de Vídeo e Imagem

- **Imagens digitais puras são codificadas em PCM**
  - Representados por matrizes de píxeis

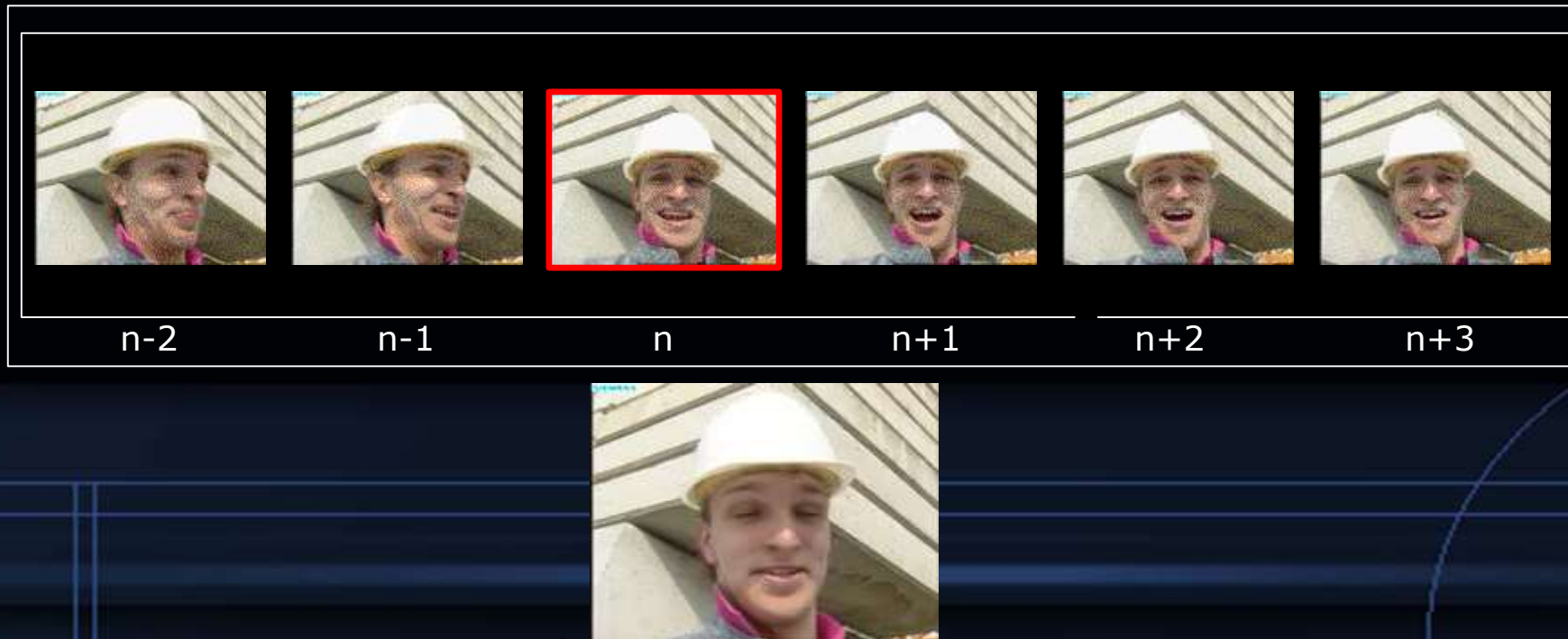


- Também é possível compactar usando DPCM e ADPCM



# Técnicas de Compressão de Vídeo e Imagem

- **Um vídeo é uma sequência de imagens amostradas rapidamente**
  - A velocidade da amostragem engana o cérebro, criando a ilusão de movimento



Foreman 30fps  
(30 imagens exibidas a cada segundo)



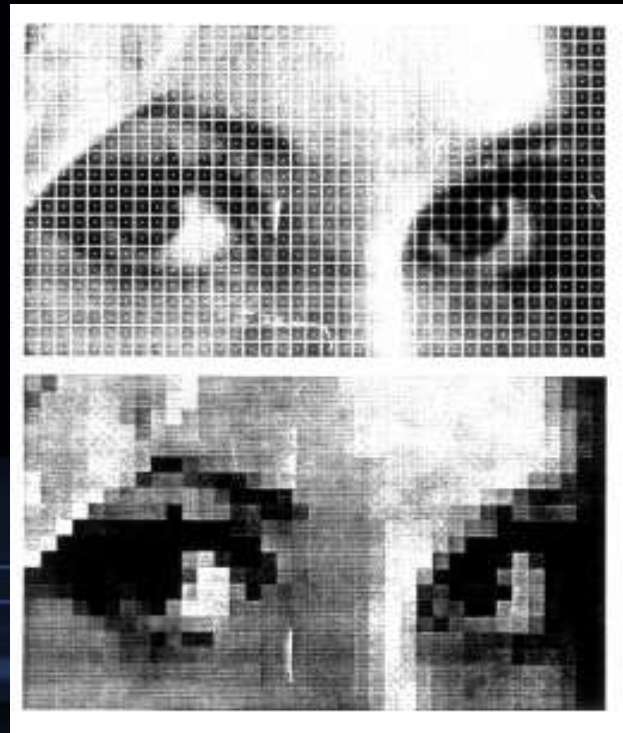
# Técnicas de Compressão de Vídeo e Imagem

- **Técnicas de compressão de vídeo e imagens**
  - Baseiam-se na alta redundância das imagens e vídeos
  - Certas áreas de figuras são uniformemente coloridas ou altamente correlatas (podendo formar padrões)
    - redundância espacial ou correlação espacial
    - removida tanto quanto possível para uma certa qualidade de apresentação
  - Não existem grandes diferenças entre quadros de um vídeo
    - redundância temporal ou correlação temporal
    - alta taxa de compressão



# Técnicas de Compressão de Vídeo e Imagem

- **Técnica de Redução da Resolução Geométrica**
  - Redução da resolução das imagens
    - Redução de linhas e colunas do bitmap



# Técnicas de Compressão de Vídeo e Imagem

## ■ Técnica de Truncagem

- Consiste em truncar dados arbitrariamente baixando o número de bits por pixel (imagem) ou taxa de quadros (vídeo)
  - feito pela eliminação dos bits menos significativos de cada pixel (imagem) e imagens por segundo (vídeo)
- Técnica é atrativa pois ela é simples



- Exemplo: imagens coloridas com 24 bits por pixel poderiam ser reduzidas para 8 bits



# Técnicas de Compressão de Vídeo e Imagem

- **Codificação Preditiva**

- Imagem original e imagem com apenas o erro de predição
  - 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



# Técnicas de Compressão de Vídeo e Imagem



## ■ Codificação Preditiva

### ▣ Preditores típicos

^

$s_n = 0.97s_{n-1}$  Preditor de 1ª ordem, 1D

^

$s_{m,n} = 0.48s_{m,n-1} + 0.48s_{m-1,n}$  Preditor de 2ª ordem, 2D

^

$s_{m,n} = 0.8s_{m,n-1} - 0.62s_{m-1,n-1} + 0.8s_{m-1,n}$  Preditor de 3ª ordem, 2D

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



# Técnicas de Compressão de Vídeo e Imagem



## ■ Codificação Preditiva

- Usar para a primeira fila e primeira coluna o preditor de 1ª ordem

$$\hat{s}_n = 0.97s_{n-1} \quad \text{Preditor de 1ª ordem, 1D}$$

- Para as outras filas e colunas o de 3ª ordem.

$$\hat{s}_{m,n} = 0.8s_{m,n-1} - 0.62s_{m-1,n-1} + 0.8s_{m-1,n} \quad \text{Preditor de 3ª ordem, 2D}$$

- Saída DPCM calculada subtraindo a saída predita com os valores originais

20	21	22	21
18	19	20	19
19	15	14	16
17	16	15	13

Original

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

Saída prevista

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

Saída DPCM

# Técnicas de Compressão de Vídeo

- **Preenchimento Condicional**

- Explora redundância temporal em vídeos
  - animação de imagens implica que píxeis na imagem anterior estão em diferentes posições que na imagem atual



# Técnicas de Compressão de Vídeo

## ■ Preenchimento Condicional

- Imagem é segmentada em áreas estacionárias e com movimento
  - são transmitidos apenas os dados de áreas com movimento
  - 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



# Técnicas de Compressão de Vídeo

- **Preenchimento Condicional**

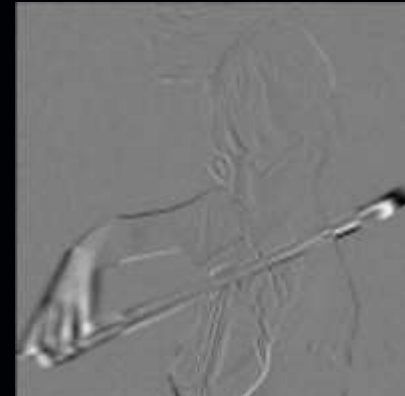
Quadro Preditor



Quadro Atual



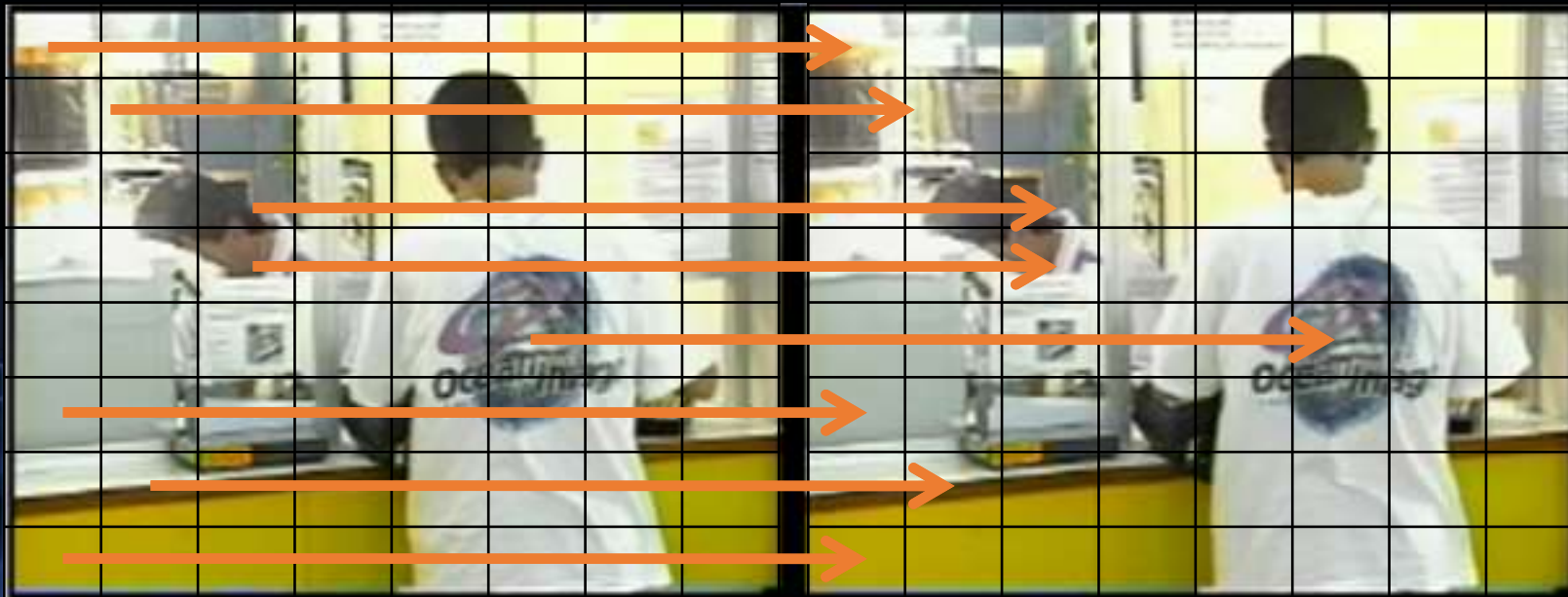
Diferença





# Técnicas de Compressão de Vídeo

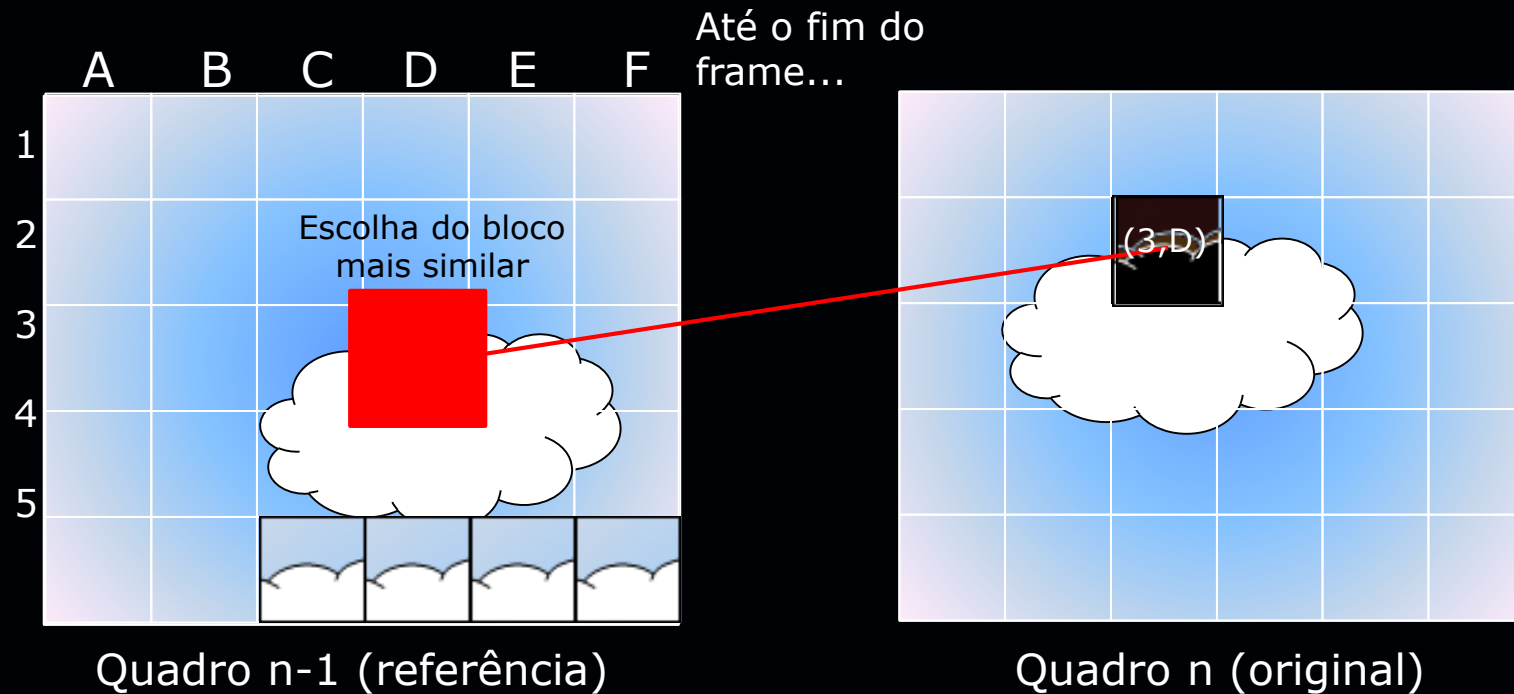
- **Estimativa e Compensação de Movimento**
  - Imagem é dividida em blocos de tamanho fixos
    - um casamento para cada bloco é procurado na imagem anterior
      - deslocamento entre estes dois blocos é chamado vetor de movimento
    - uma diferença de blocos é obtida calculando diferenças pixel a pixel
  - Vetor de movimento e a diferença de bloco é codificado e transmitido





# Técnicas de Compressão de Vídeo

- **Exemplo simples: Compara a similaridade entre blocos**



- Mantém a diferença entre os blocos (resíduo);
- Cria o vetor de movimento, referenciando o bloco do quadro anterior;

# Pontos Importantes

Técnicas gerais de compressão  
de áudio, imagens e vídeos

- Entender o princípio geral



**UFSC**

**Ciências da  
Computação**

# **CAP 3. COMPRESSÃO DE DADOS MULTIMÍDIA**

**INE5431 Sistemas Multimídia**  
**Prof. Roberto Willrich (INE/UFSC)**  
**roberto.willrich@ufsc.br**

**Aula 6: Padrões de compressão  
multimídia - JPEG**

# Compressão de Dados Multimídia



- **Conteúdo:**

- Necessidade de compressão
- Princípios da compressão
- Classificação das técnicas de compressão
- Medição do desempenho de compressão
- Técnicas de compressão sem perdas
  - RLE, Huffman, LZW (GIF), Codificação Preditiva
- Técnicas de compressão de áudio, vídeo e imagens
- **Padrões de compressão multimídia**
  - **JPEG**, MPEG, MPEG-4, H.261, H.263

# Padrões de Compressão Multimídia



- **Várias técnicas e produtos para compressão são disponíveis**
  - Utilização de padrões promove a compatibilidade entre diferentes equipamentos/aplicações (interoperabilidade)
- **Exemplos de padrões**
  - TIFF: padrão independente de fabricante para imagens
  - PNG: padrão de imagens alternativo ao GIF
  - ISO JPEG para compressão de imagens;
  - ISO JBIG para compressão sem perda de imagens bi-níveis (1 bit/píxel) para transmissão fac-símile
  - 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;



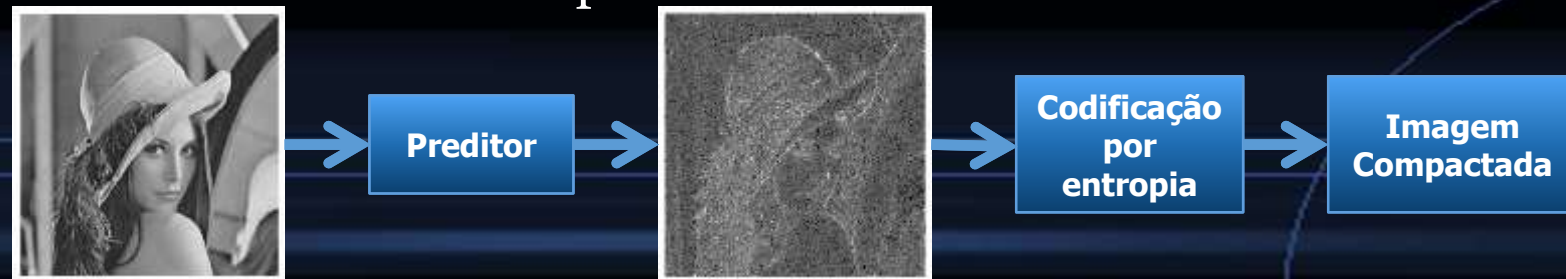
# Padrão de Compressão de Imagens JPEG



- **JPEG colaboração entre a ISO/IEC e a ITU-TS (1992)**
  - Uma das melhores tecnologia de compressão de imagem
  - Implementado em software e hardware
  - Codificação/decodificação JPEG tempo-real tem sido implementada para vídeo (Motion JPEG - MJPEG)
- **Possui versões diferentes:**
  - Versões para compressão sem perdas
  - Versões para compressão com perdas

# Padrão de Compressão de Imagens JPEG

- **Codificação JPEG sem perda**
  - Reprodução é exata
  - Necessária em aplicações que não toleram perdas (médicas e legais)
  - Existem variações
    - JPEG sem perdas original, que se baseia no DPCM e o uso de codificação por entropia (de Huffman ou aritmética)
    - JPEG-LS utiliza a técnica de codificação de Golomb-Rice e RLE
    - JPEG 2000 utiliza técnica de compressão wavelets



# Padrão de Compressão de Imagens JPEG



- **JPEG com perdas**

- Se baseia nas limitações da percepção humana e na codificação por entropia
- Compressão parametrizável
  - JPEG cobre grande faixa de qualidades de imagens e permite especificar o comportamento do codificador a partir de parâmetros
  - Quatro modos de operação:
    - Codificação sequencial (baseline)
    - Codificação progressiva
    - Codificação sem perda
    - Codificação hierárquica

# Padrão de Compressão de Imagens JPEG



- **Codificação Sequencial (baseline)**

- Suportado por toda implementação JPEG
- Modo com perdas baseada em DCT
- Componentes de imagem são codificados em uma única varredura da esquerda para direita e de cima para baixo

- **Codificação progressiva**

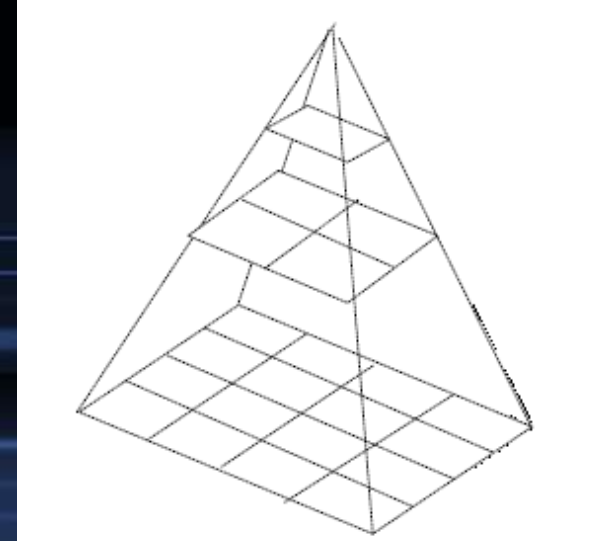
- Com perdas baseada em DCT expandido
  - Fornece avanços ao modo baseline
- Varreduras sucessivas
  - imagem é compactada em um processo de múltiplas linhas de varredura
- Geralmente utilizada em arquivos que são transmitidos pela Internet
  - pois possibilita a visualização da imagem inteira, em menor resolução, enquanto o restante da imagem esta sendo enviada

# Padrão de Compressão de Imagens JPEG



## ■ Codificação hierárquica

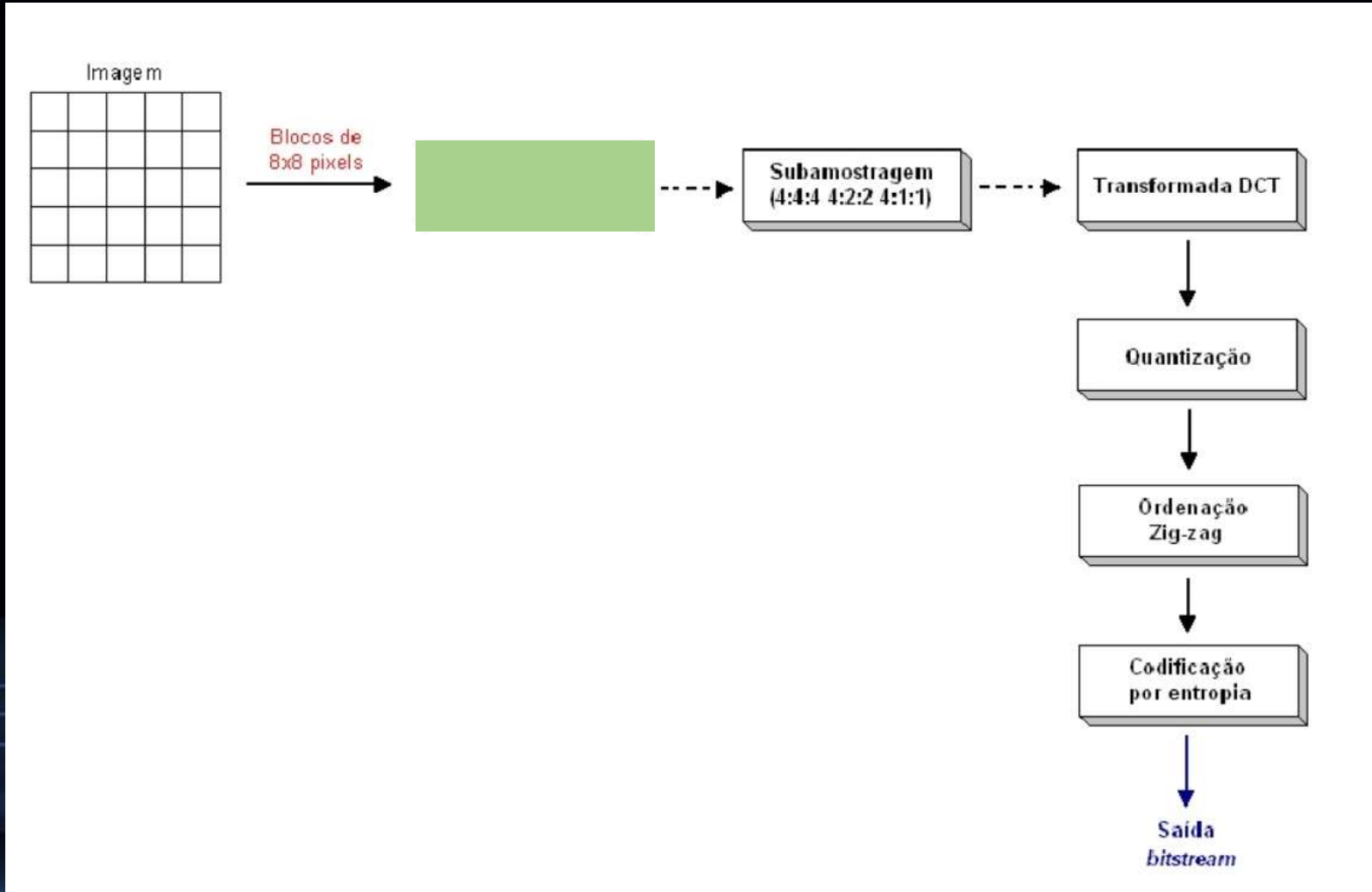
- Oferece uma codificação progressiva que aumenta de resolução espacial entre estágios progressivos
- Versões podem ser acessadas sem a necessidade de primeiro descompactar a imagem na resolução completa
- Os elementos de imagem das resoluções já recebidas são utilizados na próxima resolução, diminuindo desta forma o tamanho do arquivo
- Taxa de compressão é mais baixa que ter uma resolução única





# Algoritmo de compressão JPEG

## ■ Operações a compressão JPEG (Sequencial)



# Algoritmo de compressão JPEG



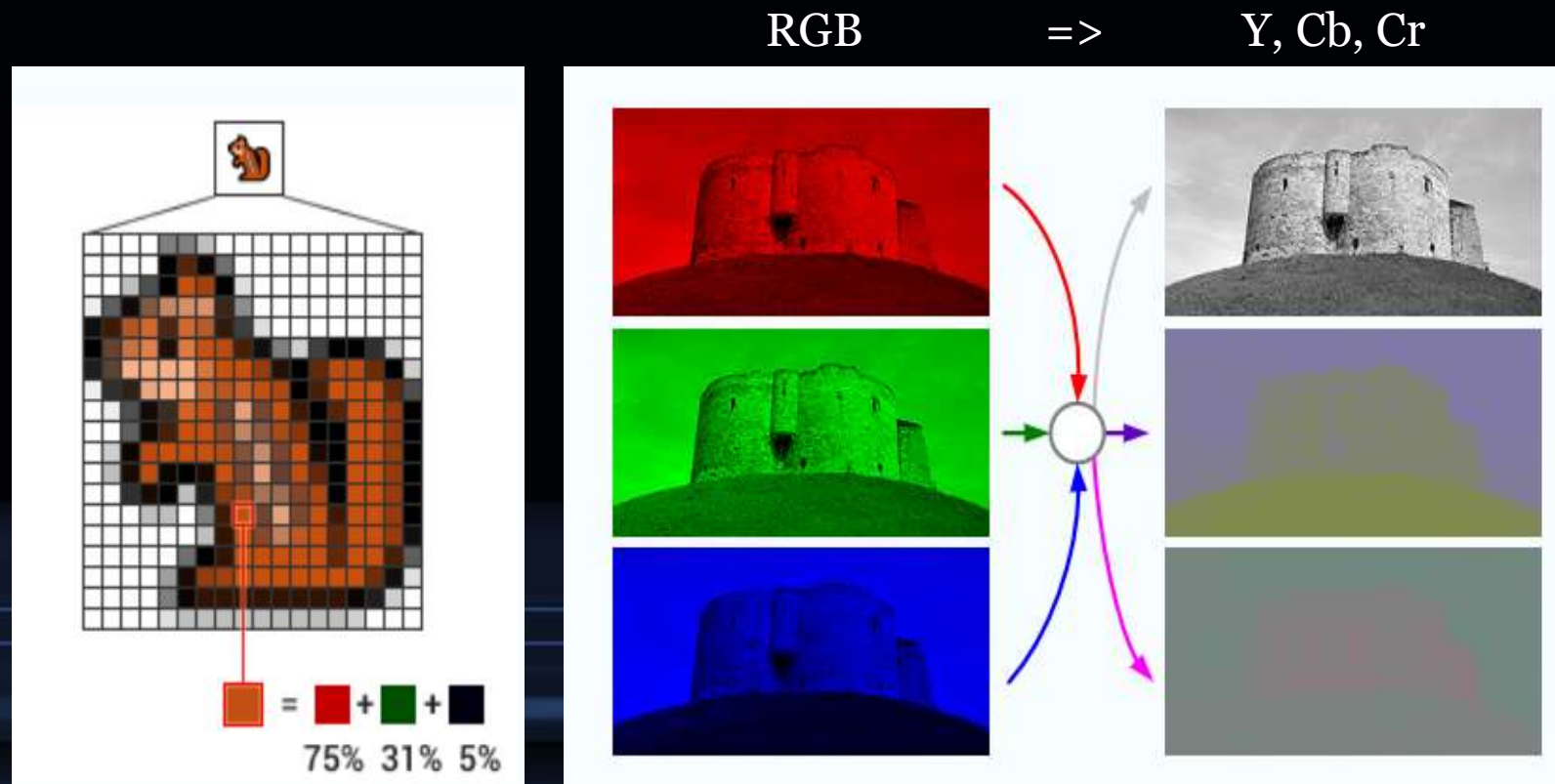
- **Transformação do espaço de cores para YCrCb**
  - Componentes “RGB” da imagem são convertidos para componentes de luminância (“Y”) e cromaticidade (“Cr” e “Cb”)
    - Y: Luminância é uma escala de representação numérica do cinza,
    - CrCb: Cromaticidade são duas escalas numéricas, que juntas representam as cores.

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

- YCbCr permite uma maior compressão sem um efeito significativo na qualidade da imagem percebida.

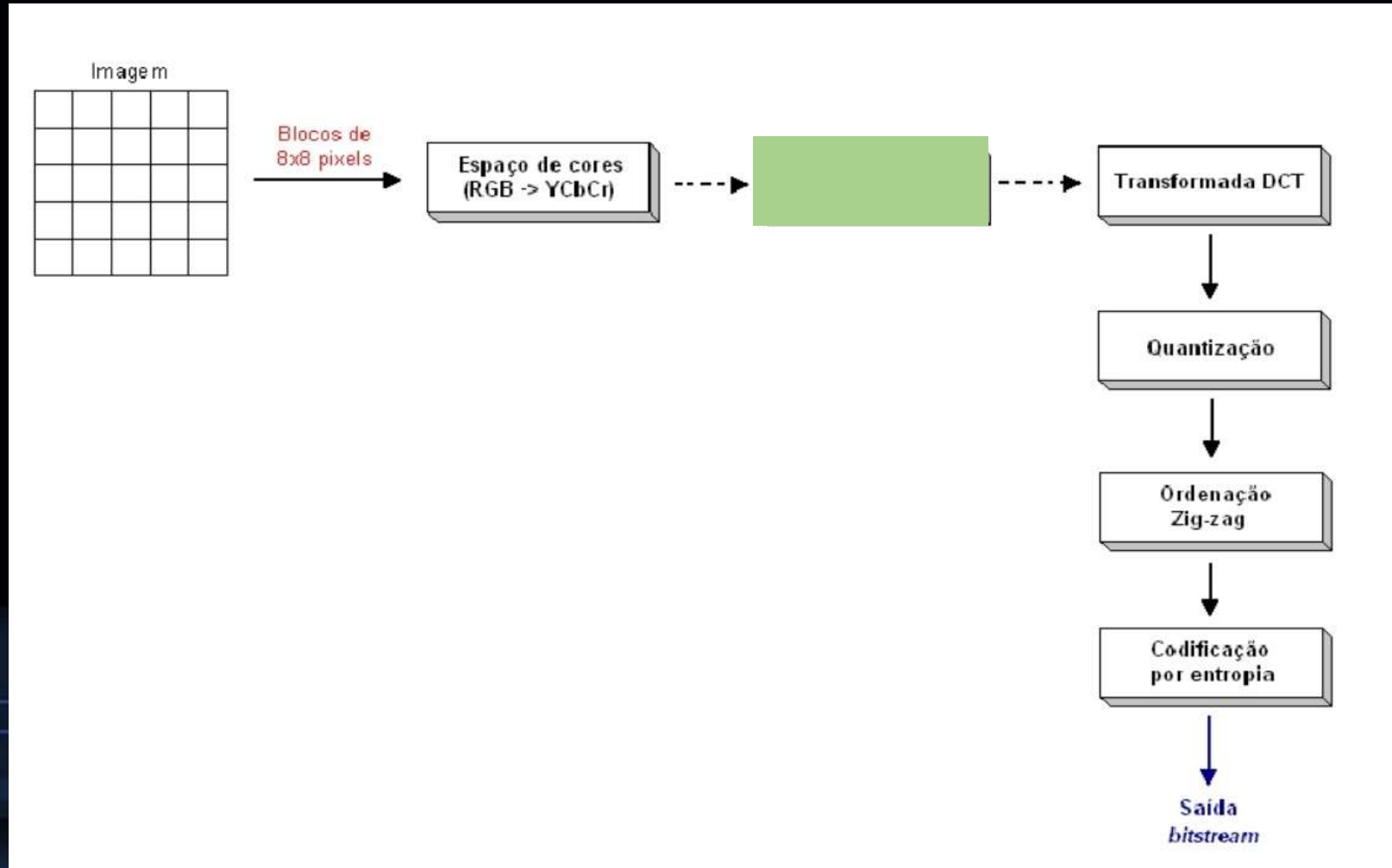
# Algoritmo de compressão JPEG

- **Transformação do espaço de cores para YCrCb**
  - Veja o tutorial em <https://cgjennings.ca/articles/jpeg-compression/>



# Algoritmo de compressão JPEG

## ■ Operações a compressão JPEG (Sequencial)



# Algoritmo de compressão JPEG



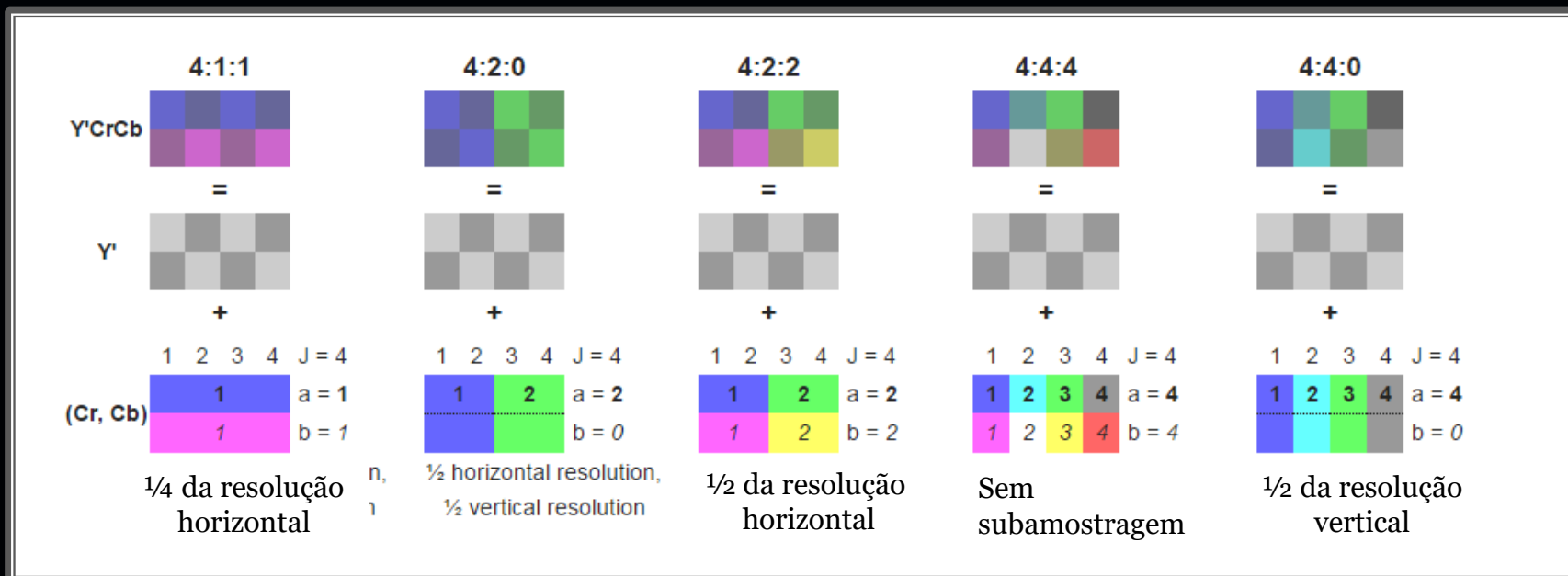
## ■ Subamostragem

- Onde é feita uma redução da resolução das matrizes YCbCr.
- Taxas de subamostragem que são normalmente aplicados no JPEG
  - 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),
  - 4:2:0 mais comumente adotada, onde a uma redução do fator 2 nas direções horizontais e verticais.
- A matriz de luminância geralmente 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
- No resto do processo de compressão, Y, Cb e Cr são processadas separadamente de maneira muito similar.



# Padrão ISO/IEC MPEG-1 Vídeo

## ■ Subamostragens YCbCr



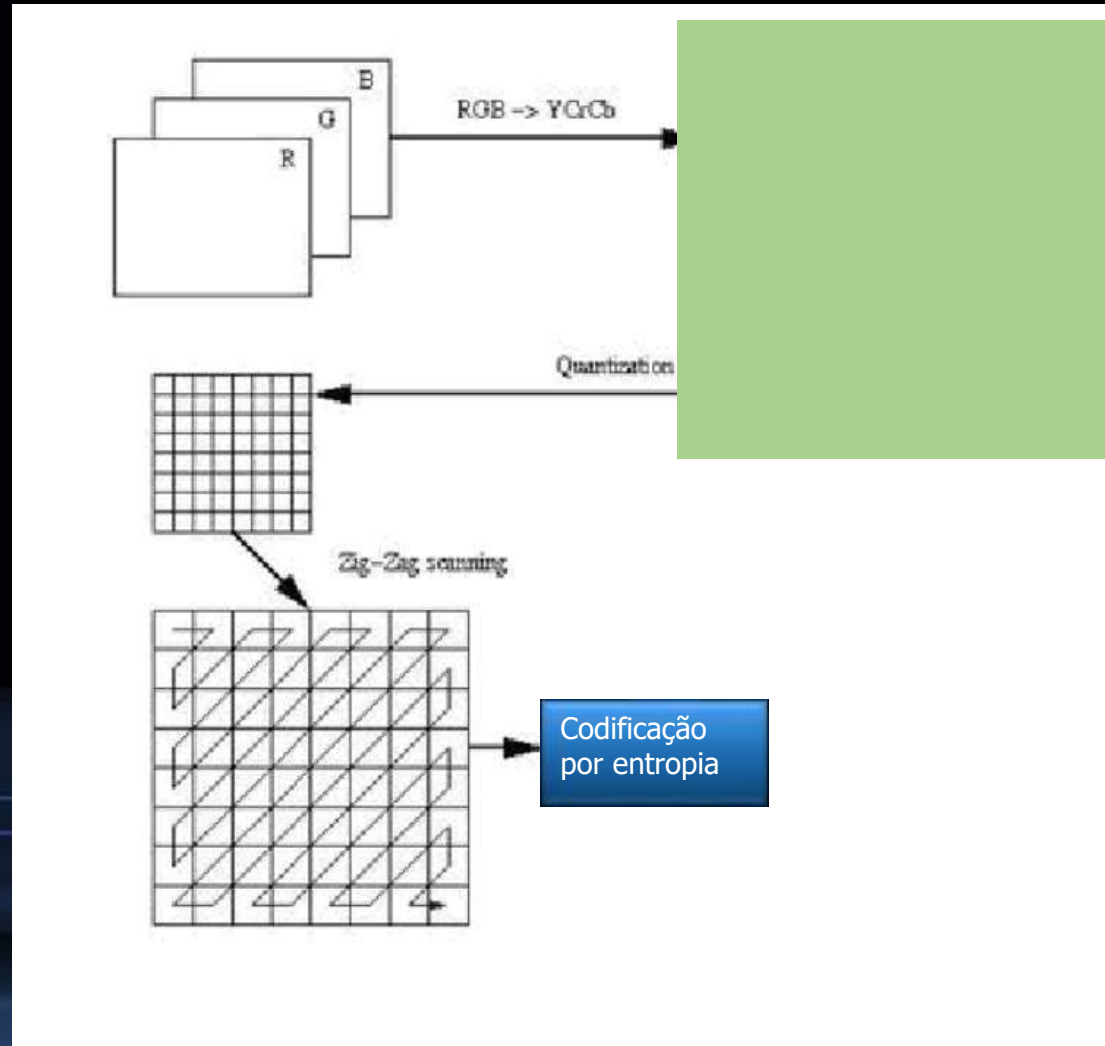
Original sem sub-amostragem de crominância, 200% de zoom



Original após sub-amostragem de crominância, 200% de zoom

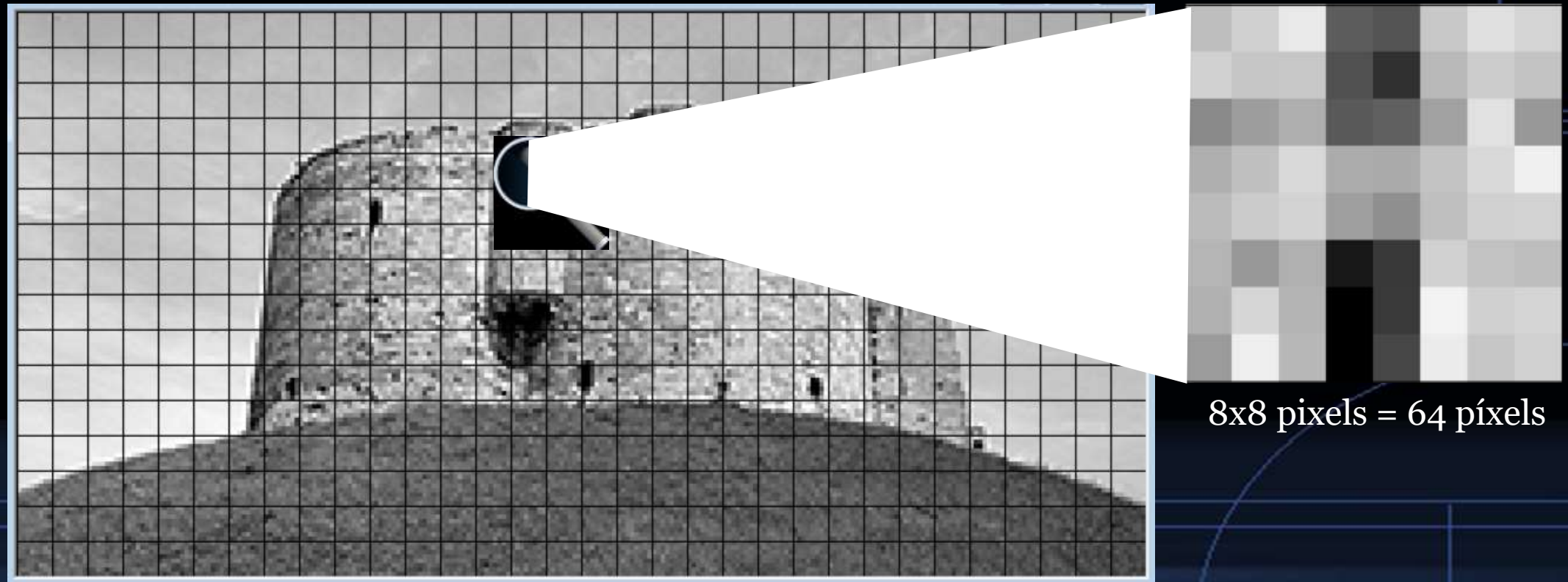
# Algoritmo de compressão JPEG

- **Operações a compressão JPEG**



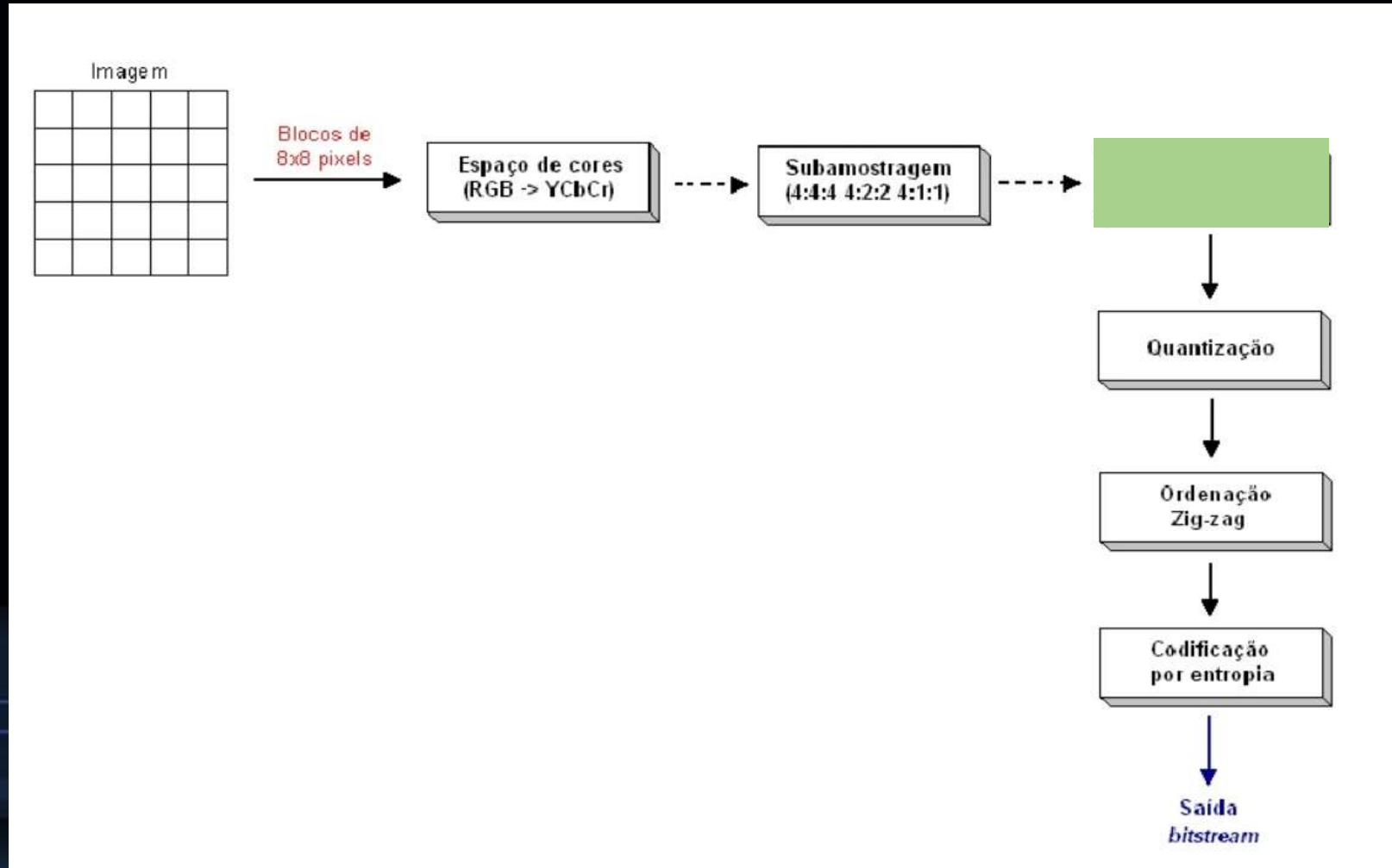
# Algoritmo de compressão JPEG

- **Decomposição das matrizes Y,Cb,Cr**
  - Decompostas em blocos de 8x8 píxels



# Algoritmo de compressão JPEG

## ■ Operações a compressão JPEG (Sequencial)



# Algoritmo de compressão JPEG



- **Transformação discreta de co-seno (DCT) dos blocos**
  - Blocos 8x8 são transformado para o domínio da frequência espacial usando a transformada DCT
  - Similar a transformada de Fourier (representação de sinal com somatório de senos de diferentes frequências, fases e amplitudes)

No espaço bidimensional de uma imagem de 8x8 pixels, a transformada discreta de co-senos (*FDCT: Forward Discrete Cosine Transform*) é dada por:

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

$$C(w) = \frac{1}{\sqrt{2}} \text{ para } w = 0$$

$$C(w) = 1 \text{ para } w = 1, 2, \dots, 7$$

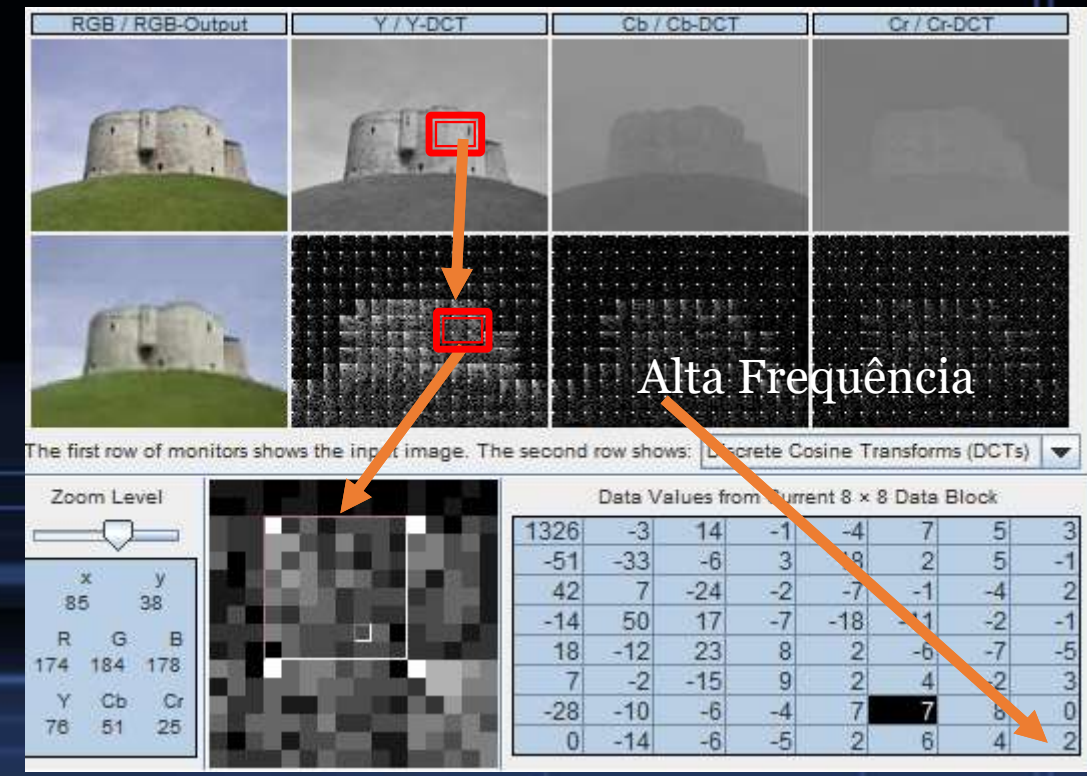
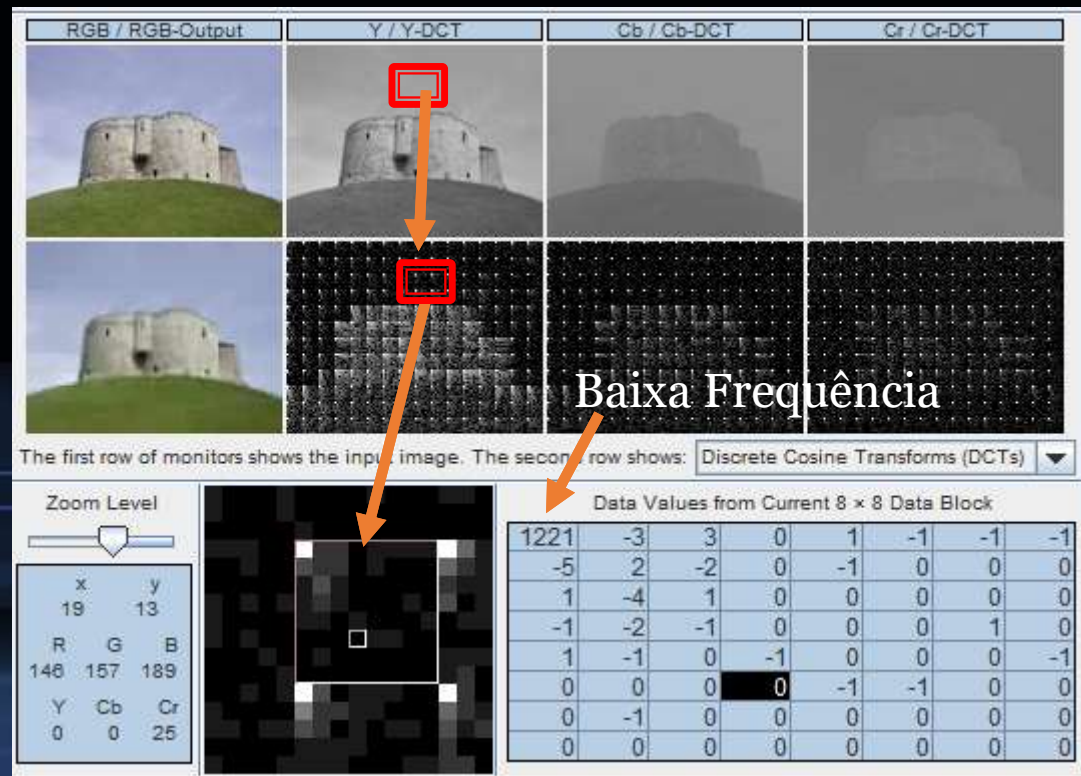
E a transformada inversa (*IDCT: Inverse Discrete Cosine Transform*) por:

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



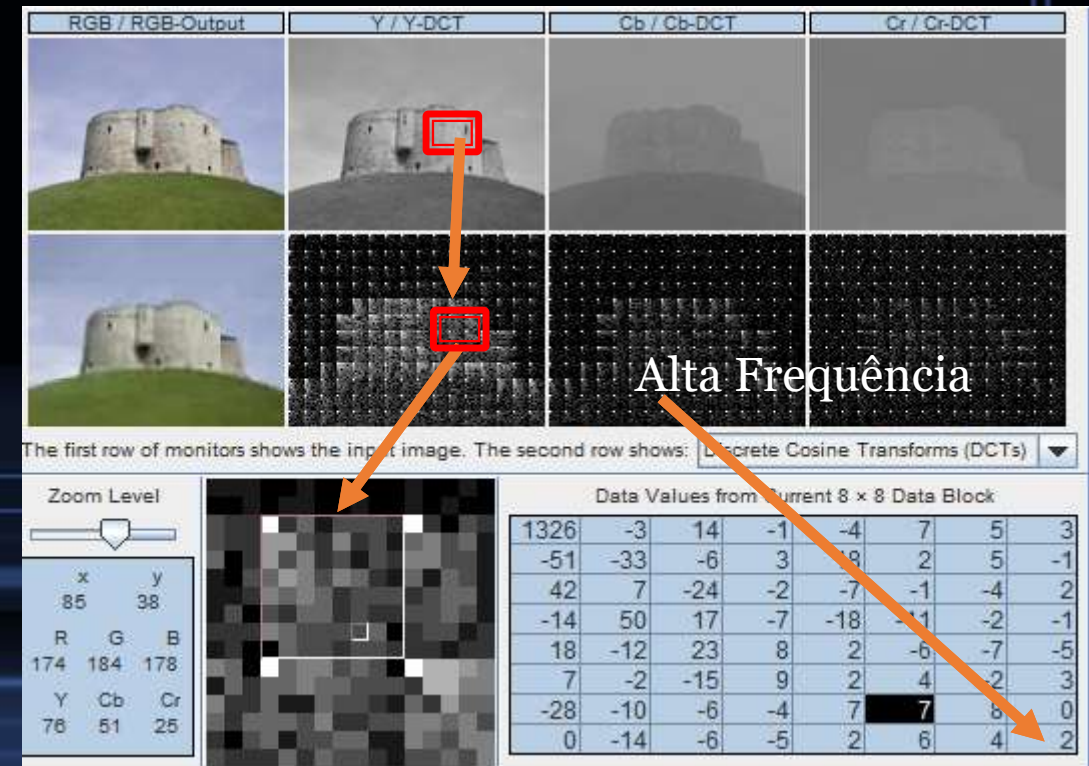
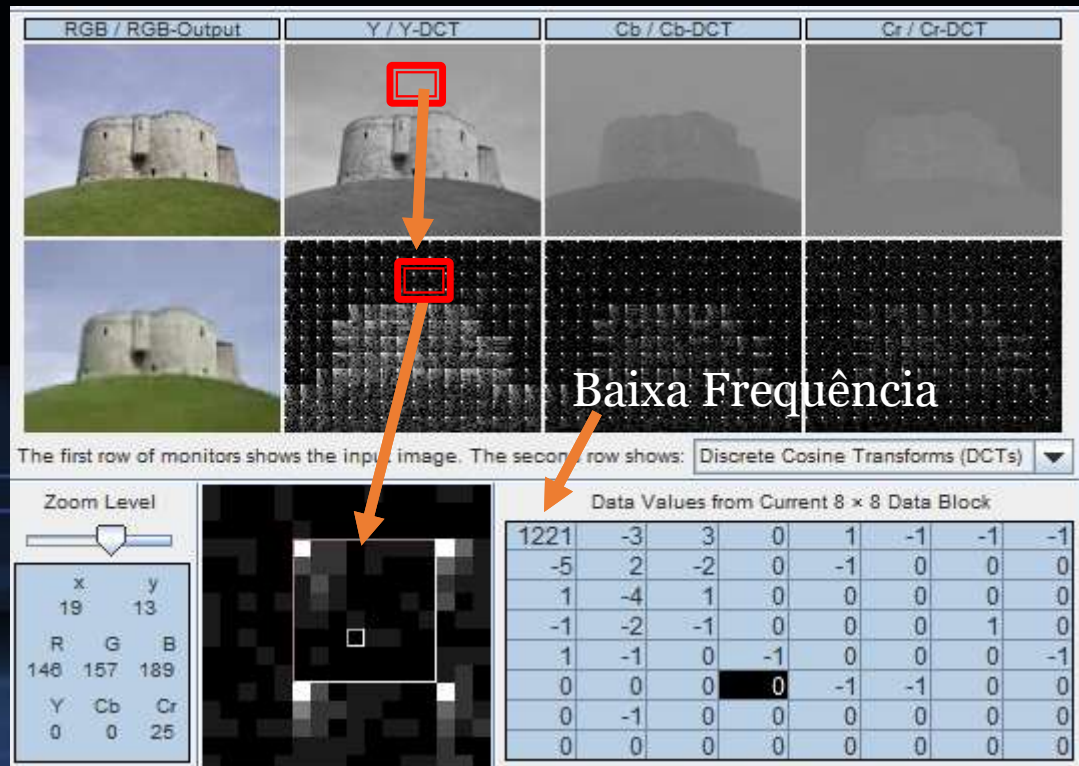
# Algoritmo de compressão JPEG

- **Transformação discreta de co-seno (DCT) dos blocos**
  - Sinal discreto de 64 pontos (um para cada bloco) transformado é uma função de duas dimensões espaciais, x e y
    - estas componentes são chamadas de frequências espaciais ou coeficientes DCT



# Algoritmo de compressão JPEG

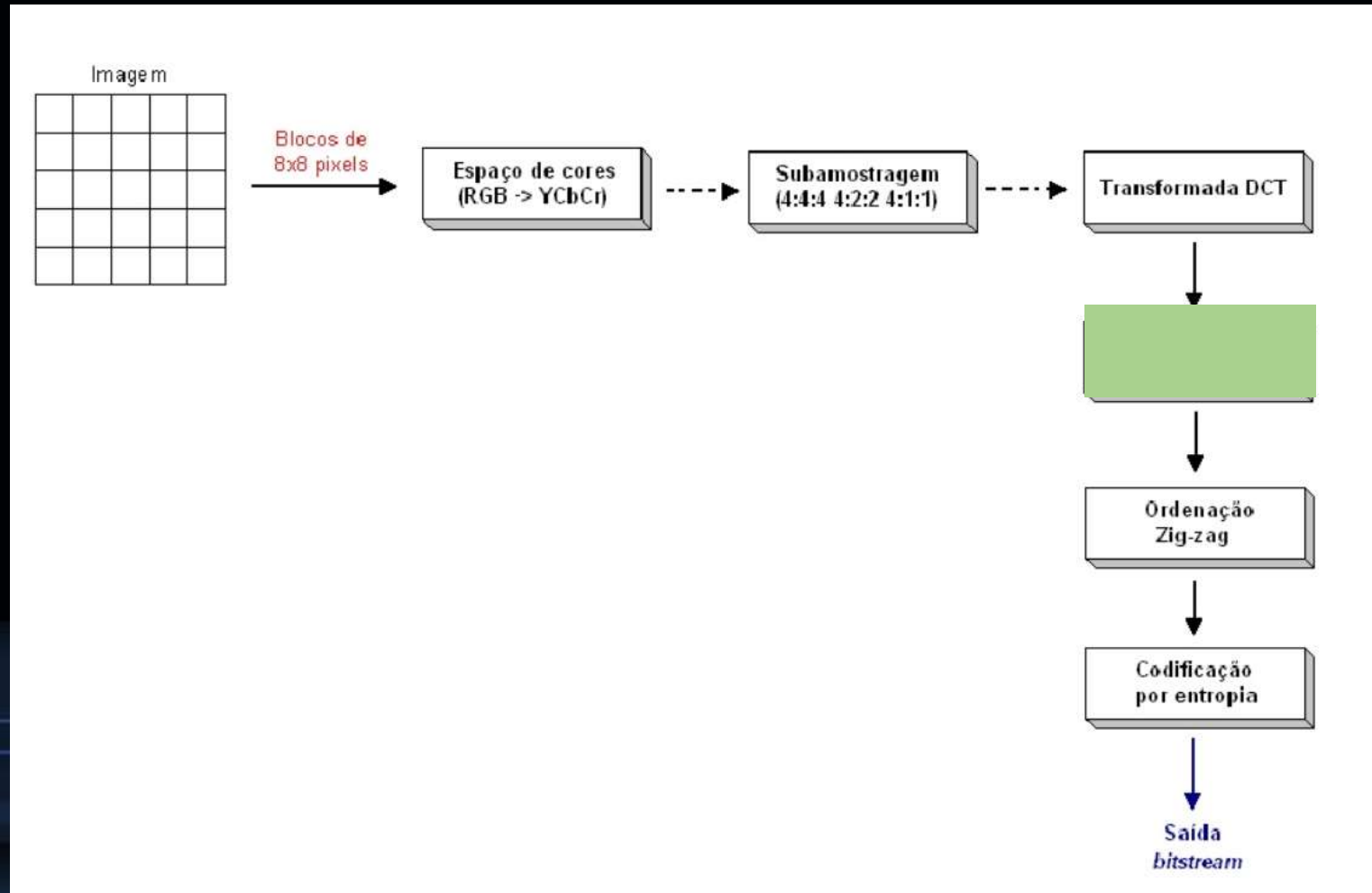
- **Transformação discreta de co-seno (DCT) dos blocos**
  - Mudanças abruptas que acontecem nos contornos de uma figura estão concentradas nas frequências mais altas.
    - uma imagem com poucos contornos deve concentrar seus coeficientes nas frequências baixas.
  - Coeficientes das frequências altas são menos importantes e perdas nesses coeficientes podem diminuir um pouco a nitidez da imagem, mas para muitas aplicações isto pode ser aceitável





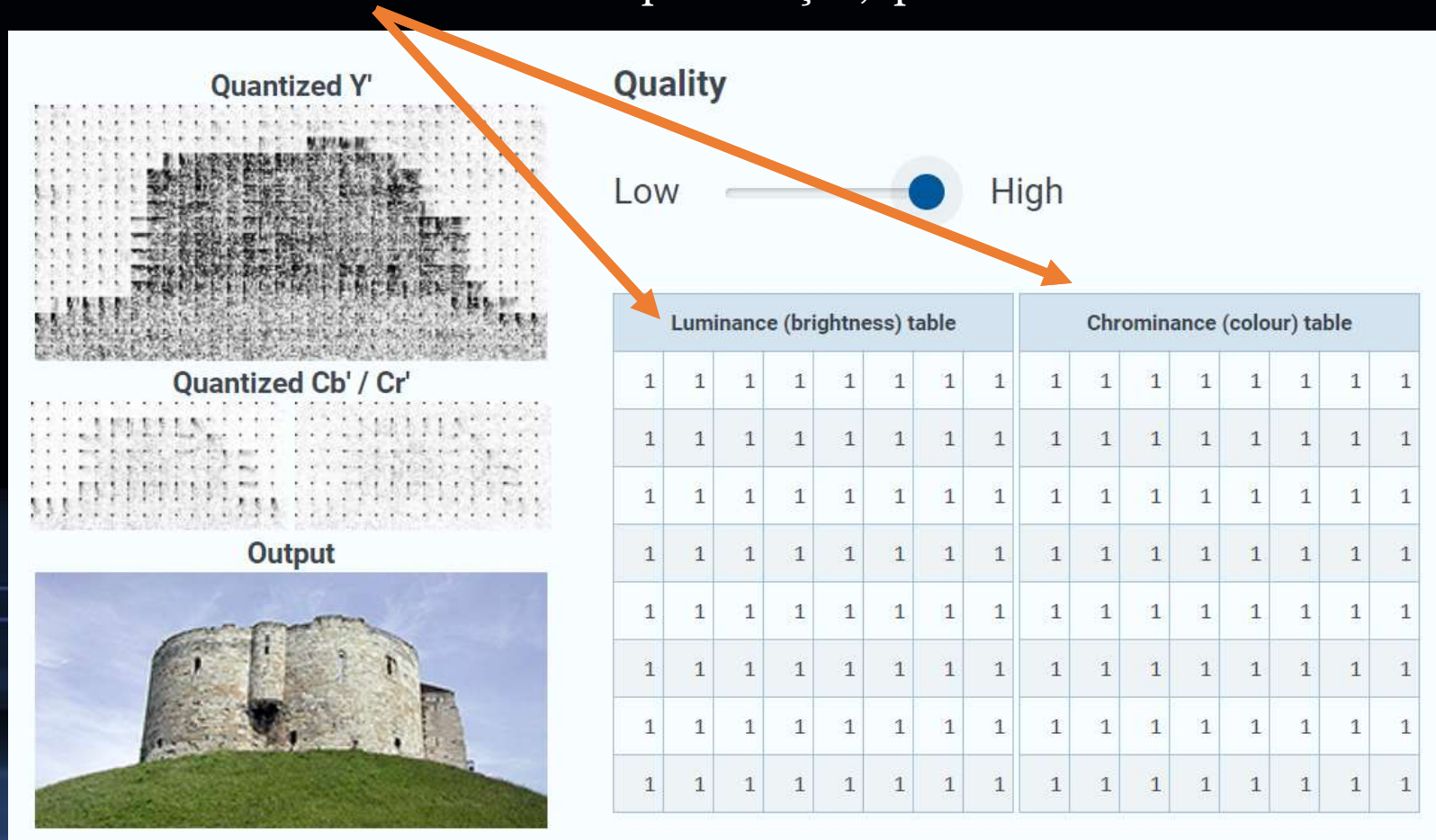
# Algoritmo de compressão JPEG

## ■ Operações a compressão JPEG



# Algoritmo de compressão JPEG

- **Quantificação (escolha da qualidade)**
  - Alta frequência é desprezada de acordo com a taxa de compressão desejada:
    - Utilizando duas tabelas de quantização, que variam de acordo com a qualidade desejada



# Algoritmo de compressão JPEG



## ■ Quantificação (escolha da qualidade)

- Tabelas de quantização são divisores: Em cada bloco 8x8 terá seus coeficientes DCT divididos pelo número correspondente em sua tabela de quantização.
- O resultado de cada divisão é arredondado para o número inteiro mais próximo e as partes fracionárias são jogadas fora

Coeficientes DCT

55	50	45	20	12	2	2	1
55	48	...	...	...	...	...	...
...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...

Tabela de quantização

1	2	3	3	3	6	6	12
1	3	...	...	...	...	...	...
...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...

Coeficientes DCT quantizado

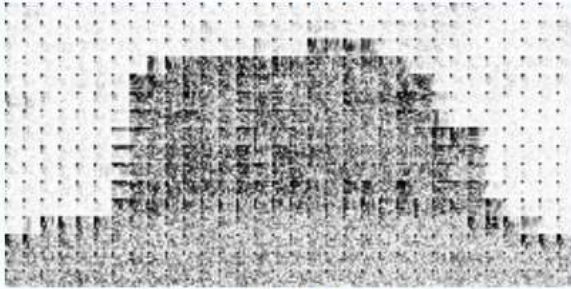
55	25	15	20	6	0	0	0
55	16	...	...	...	...	...	...
...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...




# Algoritmo de compressão JPEG

- **Quantificação (escolha da qualidade)**
  - Alta frequência é desprezada de acordo com a taxa de compressão desejada:
    - maior é taxa de compressão, maior é o número de componentes de alta frequência desprezados


Quantized Y'



Quantized Cb' / Cr'



Output



Quality

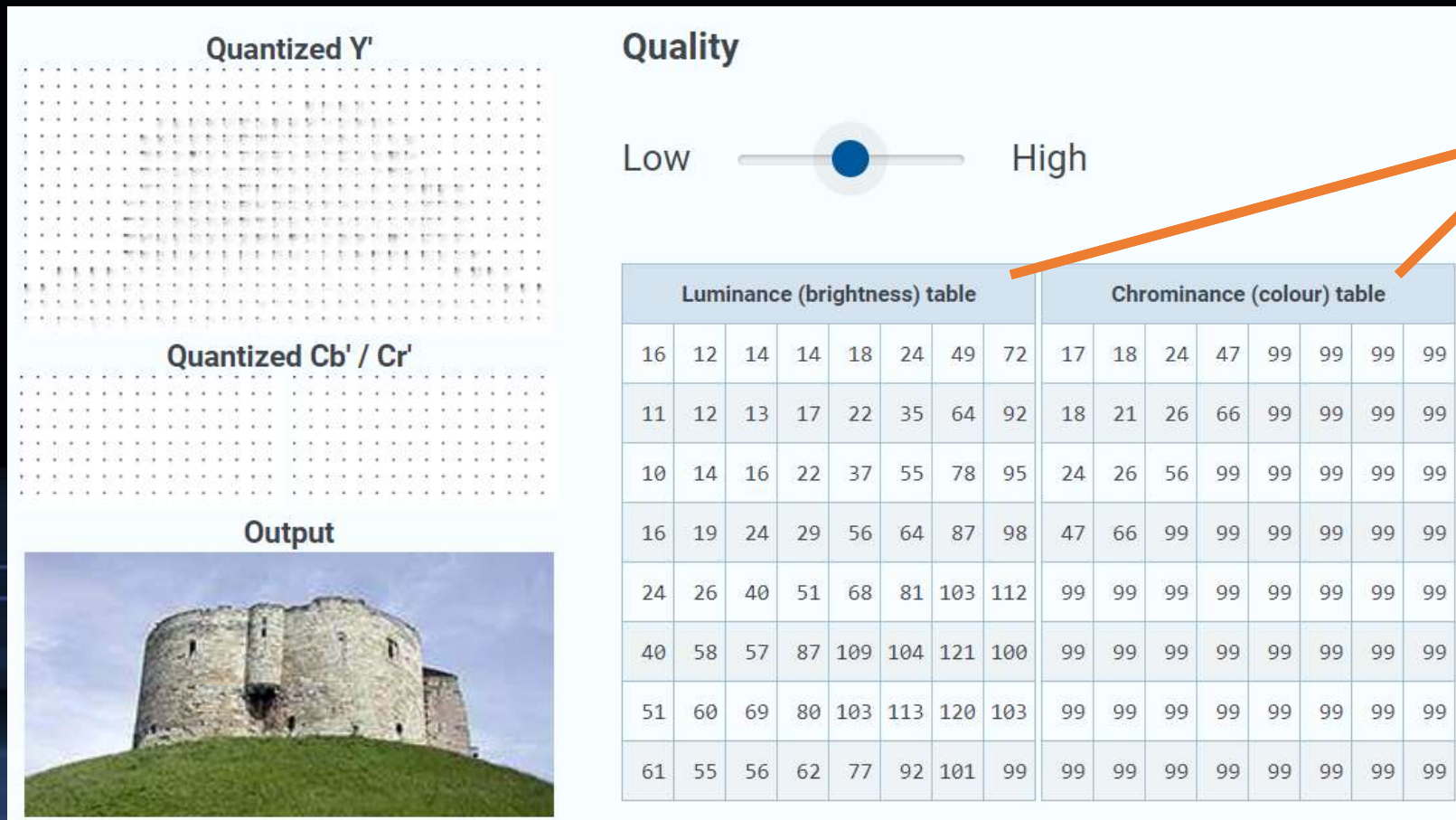
Low ☐ High

Luminance (brightness) table								Chrominance (colour) table							
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	

Alta qualidade preserva os coeficientes de alta e baixa frequência

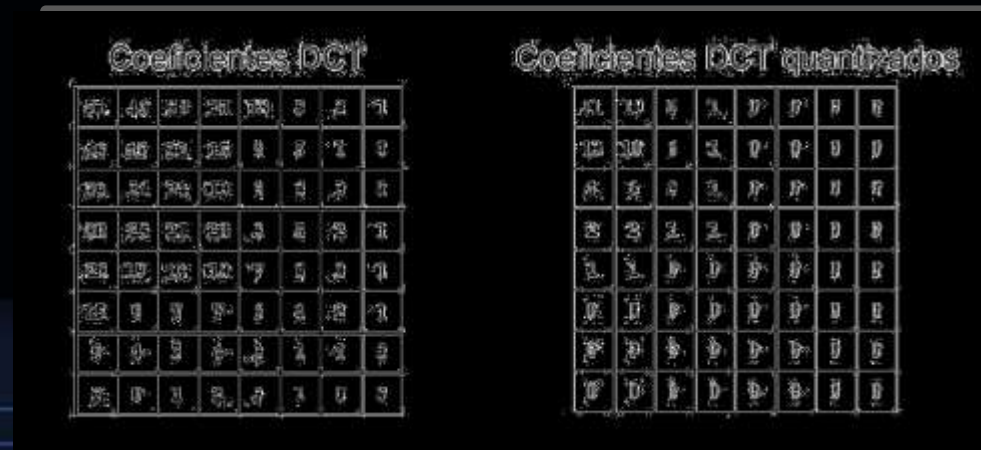
# Algoritmo de compressão JPEG

- **Quantificação (escolha da qualidade)**
  - Alta frequência é desprezada de acordo com a taxa de compressão desejada:
    - maior é taxa de compressão, maior é o número de componentes de alta frequência desprezados



# Algoritmo de compressão JPEG

- **Quantificação (escolha da qualidade)**
  - Quantificação prioriza a baixa frequência
    - os coeficientes gerados são quantizados de forma diferenciada, usando uma maior precisão para as frequências mais baixas.





# Algoritmo de compressão JPEG

- **Quantificação (escolha da qualidade)**
  - Alta frequência é desprezada de acordo com a taxa de compressão desejada:
    - maior é taxa de compressão, maior é o número de componentes de alta frequência desprezados

Quantized Y'

Quantized Cb' / Cr'

Output

Quality

Low  High

Luminance (brightness) table								Chrominance (colour) table							
112	84	98	98	126	168	343	504	119	126	168	329	693	693	693	693
77	84	91	119	154	245	448	644	126	147	182	462	693	693	693	693
70	98	112	154	259	385	546	665	168	182	392	693	693	693	693	693
112	133	168	203	392	448	609	686	329	462	693	693	693	693	693	693
168	182	280	357	476	567	721	784	693	693	693	693	693	693	693	693
280	406	399	609	763	728	847	700	693	693	693	693	693	693	693	693
357	420	483	560	721	791	840	721	693	693	693	693	693	693	693	693
427	385	392	434	539	644	707	693	693	693	693	693	693	693	693	693

Baixa qualidade descarta muitos coeficientes DCTs, gerando o efeito bloco

# Algoritmo de compressão JPEG

- **Quantificação (escolha da qualidade)**
  - Alta frequência é desprezada de acordo com a taxa de compressão desejada:
    - maior é taxa de compressão, maior é o número de componentes de alta frequência desprezados

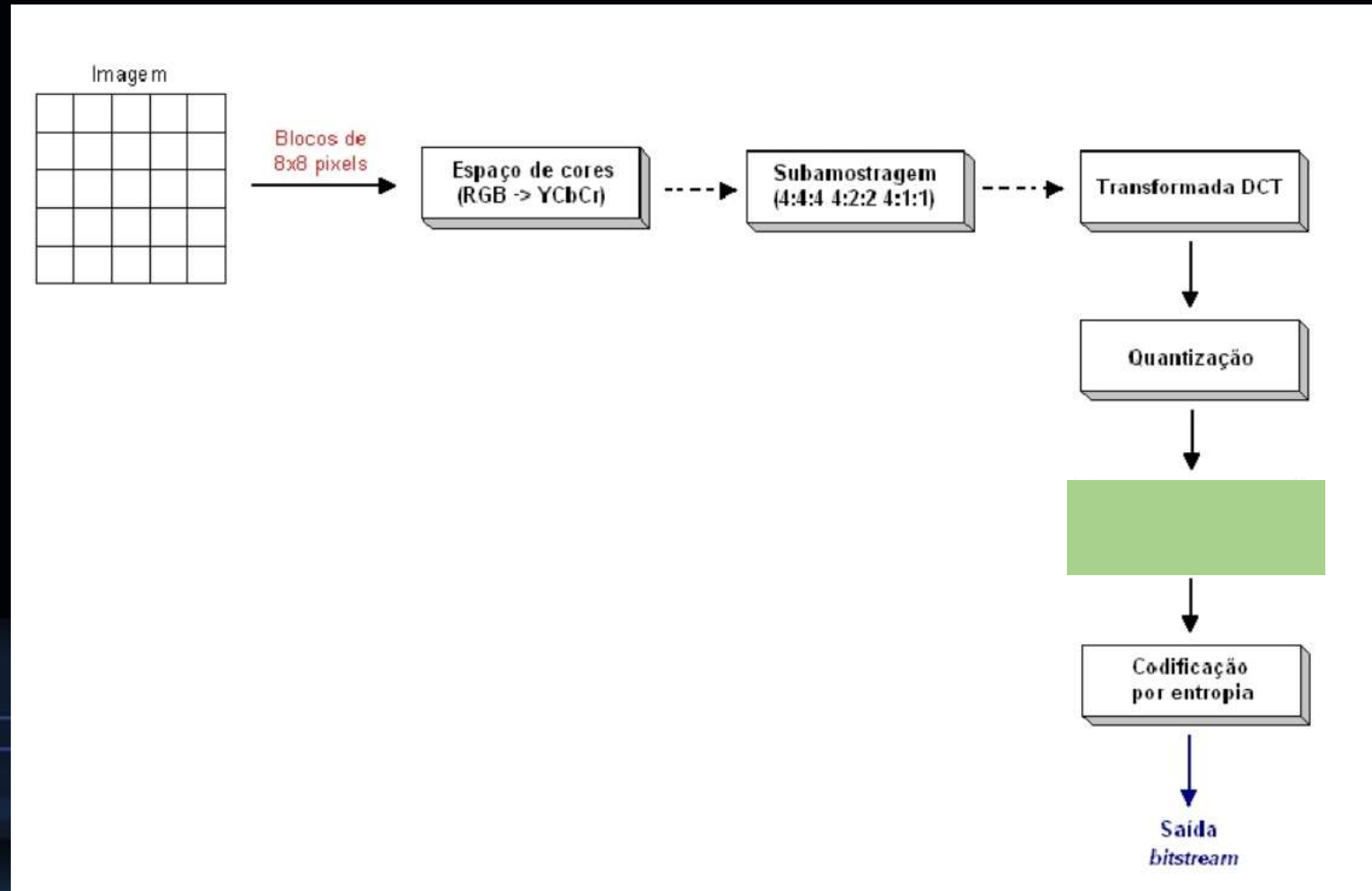


Baixa qualidade  
descarta muitos  
coeficientes DCTs,  
gerando o efeito bloco



# Algoritmo de compressão JPEG

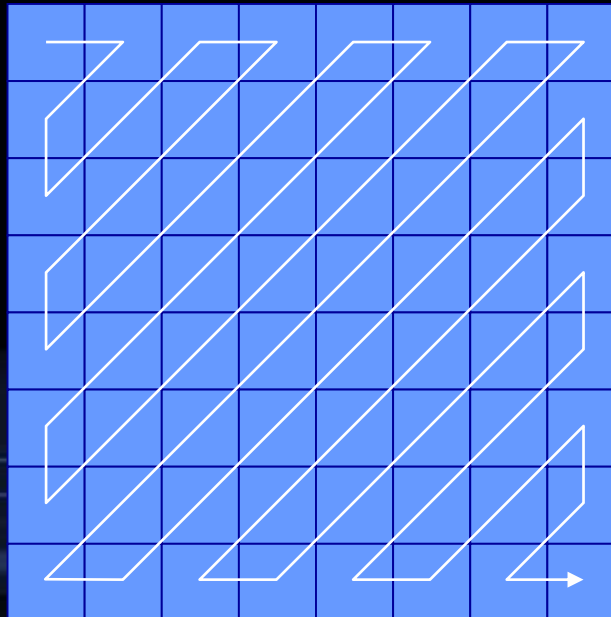
## ■ Operações a compressão JPEG



# Algoritmo de compressão JPEG

- **Ordenação dos coeficientes DCT**
  - Coeficientes DCT são ordenados em uma sequência zig-zag
    - para obter uma sequência unidimensional de dados para ser usado na codificação por entropia

coeficiente 0

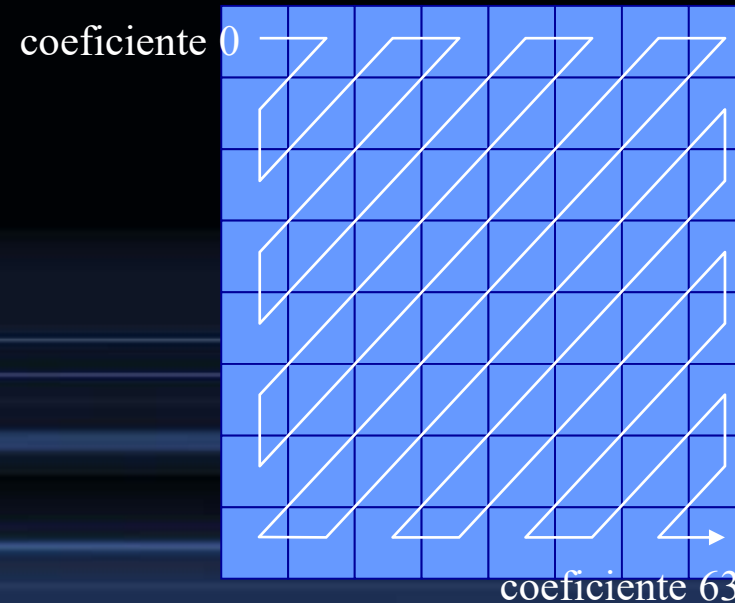


coeficiente 63

# Algoritmo de compressão JPEG

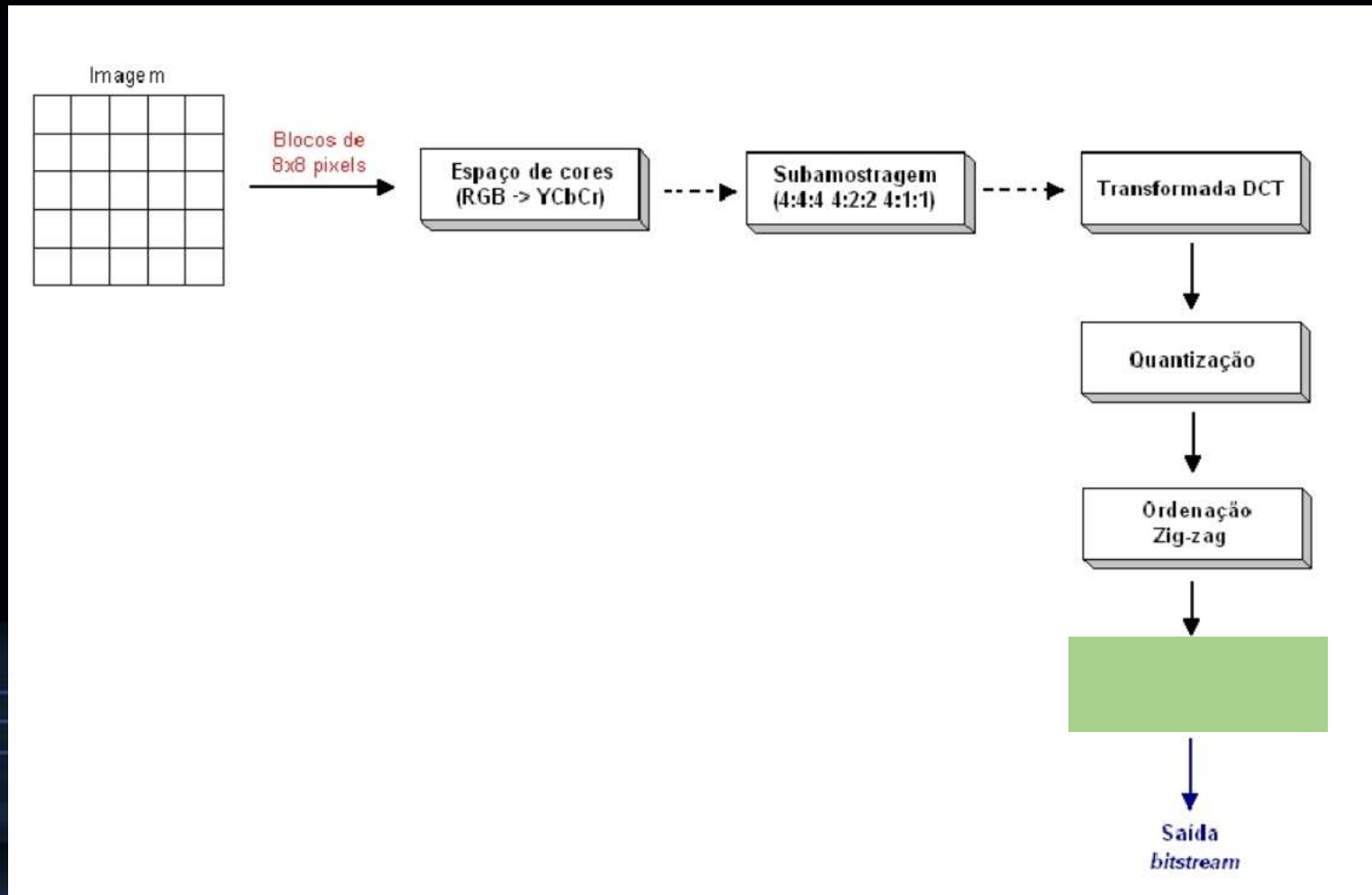
- **Ordenação dos coeficientes DCT**
  - Propósito do escaneamento zig-zag é ordenar os coeficientes em ordem decrescente de frequências espectral
    - coeficientes de alta frequências (no canto direito inferior) tem valores mais próximos a zero
      - isto leva a uma maior eficiência da codificação por entropia

1055	86	40	22	15	10	7	5
53	37	25	17	11	8	6	4
21	21	19	13	9	7	5	4
12	12	11	9	7	5	4	3
7	7	7	7	5	4	3	3
5	5	5	4	4	3	3	3
3	3	3	3	3	3	3	3
3	3	3	3	3	3	3	2



# Algoritmo de compressão JPEG

## ■ Operações a compressão JPEG



# Algoritmo de compressão JPEG



- **Codificação por entropia**
  - Esta etapa fornece uma compressão adicional
  - JPEG define dois métodos de codificação por entropia
    - Codificação de Huffman
      - única especificado no modo baseline
    - Codificação aritmética
      - normalmente 10% mais eficiente que a codificação de Huffman



# Padrão de Compressão de Imagens JPEG



- **Taxas de compressão obtidas**

- Quanto maior for a taxa de compressão maior será o número de componentes de alta frequência desprezados
  - para obter taxas de compressão muito elevadas é descartado um número significativo de componentes de alta frequência
    - levando ao aparecimento do efeito de bloco (perda de definição nos contornos das imagens).
- Valores médios
  - 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

# Algoritmo de compressão JPEG

- **Taxas de compressão obtidas e qualidades**

- Alta qualidade

- Taxa de 2.6:1



- Boa qualidade

- Taxa de 15:1



# Algoritmo de compressão JPEG

- **Taxas de compressão obtidas e qualidades**

- Qualidade média

- Taxa de 23:1



- Baixa qualidade

- Taxa de 46:1



# Algoritmo de compressão JPEG

- **Taxas de compressão obtidas e qualidades**

- Mais baixa qualidade
  - Taxa de 144:1



- **Demonstração:**

- <https://cgjennings.ca/articles/jpeg-compression/>



# Padrão de Compressão de Imagens JPEG



- **JPEG é para imagens fotográficas**
  - JPEG apresenta ótimas taxas de compressão para imagens fotográficas naturais multitonais
  - Qualidade diminui consideravelmente quando aplicado a
    - imagens gráficas com contornos e áreas bem definidas de cor, ou
    - imagens com texto, como é o caso dos logotipos



# Padrão de Compressão de Imagens JPEG

- **JPEG é para imagens fotográficas**
  - JPEG apresenta ótimas taxas de compressão para imagens fotográficas naturais multitonais

JPEG (50kB)




PNG (177KB)



# Padrão de Compressão de Imagens JPEG

- **JPEG é para imagens fotográficas**
  - Qualidade diminui consideravelmente quando aplicado a
    - imagens gráficas com contornos e áreas bem definidas de cor, ou
    - imagens com texto, como é o caso dos logotipos



PNG vs JPEG

The image shows the text 'PNG vs JPEG' in a large, bold, black font. The 'PNG' part is sharp and clear, while the 'JPEG' part is significantly blurred and pixelated, illustrating the loss of detail when compressing text-based images with JPEG.

# Padrão de Compressão de Imagens JPEG



- **Para imagens gráficas e com texto**
  - JPEG introduz ruído nas zonas de imagem compostas por cores sólidas
    - pode distorcer o aspecto geral da imagem
  - Imagem PNG compactam mais eficazmente que JPEG e apresenta uma melhor definição dos contornos do texto



# Pontos Importantes

## Algoritmo JPEG

- Saber descrever cada etapa do algoritmo de compressão JPEG



**UFSC**

**Ciências da  
Computação**

# **CAP 3. COMPRESSÃO DE DADOS MULTIMÍDIA**

**INE5431 Sistemas Multimídia**  
**Prof. Roberto Willrich (INE/UFSC)**  
**roberto.willrich@ufsc.br**

**Aula 7: Padrões de compressão  
multimídia – Codec de Voz**



# Compressão de Dados Multimídia



- **Conteúdo:**

- Necessidade de compressão
- Princípios da compressão
- Classificação das técnicas de compressão
- Medição do desempenho de compressão
- Técnicas de compressão sem perdas
  - RLE, Huffman, LZW (GIF), Codificação Preditiva
- Técnicas de compressão de áudio, vídeo e imagens
- **Padrões de compressão multimídia**
  - JPEG, **Codecs de Voz**, MPEG, MPEG-4, H.261, H.263

# Recomendações ITU-TS para Voz



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.723.1m	MP-MLQ	3,4	8	6.4	30	24
G.723.1a	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

# Codecs e Quadro de Voz



- **Quadro de voz**
  - A maioria dos codificadores de voz se baseia em quadros
    - Quadros de voz ou pacotes de voz
    - Cada quadro de voz tem uma duração de 1 a 30 ms
  - Codecs compactam quadros de voz
    - Contendo um número fixo de amostras
    - Número de amostras depende do codec utilizado

# CODECS E QUADRO DE VOZ



- **Montagem do quadro**
  - Fluxo de dados de áudio precisa ser acumulado
    - até que ele atinja o tamanho do bloco antes de ser processado pelo codificador
  - Acumulação de amostra leva tempo
    - soma-se ao atraso fim-a-fim.

# Codecs e Quadro de Voz





# Codecs e Quadro de Voz



# Codecs e QUADRO de Voz



- **Tamanho do quadro de voz (payload)**
  - Tamanho do payload (em bytes)
    - Taxa do Codec (em bits/sec) x tempo do quadro de voz (s)
    - Exemplo 1:
      - Codec G.711 => 64 kbps
      - Tamanho do pacotes = 20ms
      - Tamanho do payload =  $(64000 \times 0,02)/8 = 160$  bytes
      - Teríamos  $(1000/20) = 50$  pacotes de 160 B de dados a cada segundo
    - Exemplo 2:
      - Codec G.711 => 64 kbps
      - Tamanho do pacotes = 30ms
      - Tamanho do payload =  $(64000 \times 0,03)/8 = 240$  bytes
      - Teríamos 33 pacotes de 240 B de dados a cada segundo

# Codecs e QUADRO de Voz



- **Problema da sobrecarga de protocolos**
  - Para ser transmitido na rede, o quadro de voz deve ser encapsulado em diversos protocolos
    - Até chegar à camada de enlace, aos pacotes de voz vão ser adicionados 40 bytes: RTP (12 bytes) + UDP (8 bytes) + IP (20 bytes) = 40 bytes.
    - Se o quadro é pequeno, a sobrecarga de protocolos é maior:
      - Quadro de voz de 160B: pacote IP será de  $40+160=200$ B, sobrecarga de  $40/200 = 20\%$
      - Quadro de voz de 240B: pacote IP será de 280B, sobrecarga de  $40/280 = 14\%$



# Codecs, Quadros e Pacotes de Voz



- **Relação entre tamanho de quadro de voz e atraso**

- Para redução do atraso, o codec escolhido deveria ter um quadro de quadro pequeno
- Exemplo 1:
  - Codec G.711 => 64 kbps
  - Tamanho do pacotes = 20ms
  - Tempo de empacotamento será 20ms
- Exemplo 2:
  - Codec G.711 => 64 kbps
  - Tamanho do pacotes = 30ms
  - Tempo de empacotamento será de 30ms

# Codecs, Quadros e Pacotes de Voz

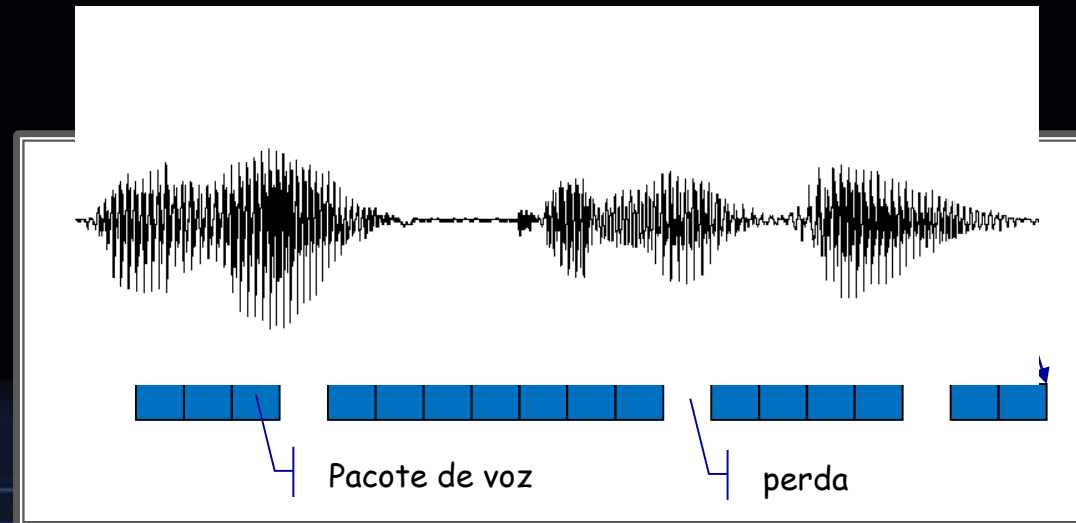


- **Relação entre tamanho de quadro de voz e taxa de transmissão**
  - Quadro pequeno (i.e. menor atraso) gera uma maior taxa de bits devido a sobrecarga dos protocolos
  - Exemplo 1: Codec G.711 => 64 kbps
    - Tamanho do pacote de voz = 20ms
    - Tamanho do Pacote IP  $40 + 160 = 200$  B
    - Teríamos 50 pacotes IP de 200 B de dados a cada segundo
      - Taxa de bits é de  $50 * 200 * 8 = 80$  kbps
  - Exemplo 2: Codec G.711 => 64 kbps
    - Tamanho do pacote de voz = 30ms
    - Tamanho do pacote IP =  $40 + 240 = 280$  B
    - Teríamos 33 pacotes de 280 B de dados a cada segundo
      - Taxa de bits é de  $33 * 280 * 8 = 73,9$  kbps



# Codecs, Quadros e Pacotes de Voz

- **Relação entre tamanho de quadro de voz e qualidade na ocorrência de perda de pacotes**
  - Se o pacote de voz é maior, a perda gera um tempo maior de ausência de som na saída
  - Maior o pacote, pior a tolerância à perda de pacotes.



# Recomendações ITU-TS para Voz



## ■ G.711

- Usa PCM compandido (escala não linear)
  - Serve para aumentar a resolução de sinais de baixa amplitude
    - Mais importante para os humanos
  - Operando de forma análoga ao ouvido humano
- Dois tipos de escala
  - A-law (Europa e Brasil)
  - $\mu$ -law (EUA e Japão)
- Usado na maioria dos backbones telefônicos digitais
- Fluxo de bits de 64 kbps
  - 8 bits por amostra, 8000 amostras/s (uma amostra a cada 125 $\mu$ s)
- Supressão de silêncio é opcional
  - Reduz a taxa de bits gerada

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

# Recomendações ITU-TS para Voz



## ■ **Tamanho do Payload**

- Quanto menor o payload de voz maior é a sobrecarga dos diversos protocolos de transmissão da voz
- Quanto maior o payload maior é o atraso na aplicação
  - para aguardar a montagem do payload

Recomendação	Taxa de bit compactado (kbps)	Duração do payload (ms)	Tamanho do payload (bytes)
G.711	64	20	160
G.723.1m	6.4	30	24
G.723.1a	5.3	30	20
G.726	16, 24, 32, 40	15	60
G.728	16	20	40
G.729A	8	20	20

# Recomendações ITU-TS para Voz



## ■ G.721

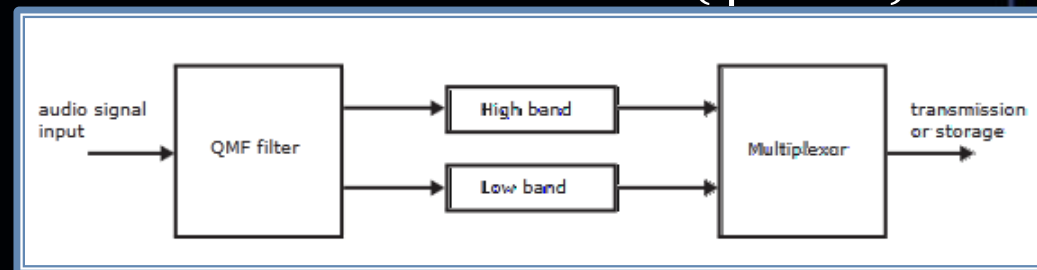
- Converte um fluxo de 64 kbps em um fluxo de 32 kbps aplicando uma compressão ADPCM
- A previsão e o tamanho do passo altera com o histórico do sinal
- Está obsoleto e substituído pelo G.726

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		

# Recomendações ITU-TS para Voz

## ■ G.722

- Fornece uma melhor qualidade que o G.711 e G.721: Utiliza 14 bits por amostra
- ADPCM Sub-banda: sinal de voz é dividido em duas sub-bandas: alta (4-8kHz) e baixa (0-4kHz), no 64kbps:
  - 2 bits/amostra para banda alta (16 kbps)
  - 6 bits/amostra para banda baixa (48 kbps)
- Próprio para aplicações de videoconferência uma vez que telefones comuns não respondem na faixa de 7kHz



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		



# Recomendações ITU-TS para Voz



## ■ G.723.1

- 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
- Valor típico de tamanho do pacote de voz (payload) é de 30ms (20 ou 24 bytes)

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.723.1m	MP-MLQ	3,4	8	6.4	30	24
G.723.1a	ACELP	3,4	8	5.3	30	20

# Recomendações ITU-TS para Voz



## ■ G.726

- O G.726 utiliza o ADPCM a 40, 32, 24 e 16 kbps
- Sinal de voz é amostrado a 8kHz, codificado em 8 bits (leis A ou m) e são transmitidas diferenças entre amostras com 5, 4, 3 ou 2 bit em quantificação adaptativa
- Valor típico de tamanho do pacote de voz (payload) é de 15ms (60 bytes)

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 ñ linear	3,4	8	64	20	160
G.723.1m	MP-MLQ	3,4	8	6.4	30	24
G.723.1a	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

# Recomendações ITU-TS para Voz



## ■ G.728

- Técnica de codificação LD-CELP (Low-Delay, Code-Excited Linear Prediction), gerando uma taxa de bits de 16 kbps
- Tabela (codebook) utilizada é formada por 1024 ( $2^{10}$ ) valores
  - contém os valores de códigos (vetores) que representam as possíveis amostras do sinal de voz
- Em cada janela de 0,625ms do sinal de voz são analisadas 5 amostras de 8 bits e é gerado 1 código de 10 bits

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 ã linear	3,4	8	64	20	160
G.723.1m	MP-MLQ	3,4	8	6.4	30	24
G.723.1a	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

# Recomendações ITU-TS para Voz



## ■ G.729

- Bastante popular em aplicações de voz sobre frame relay e em modems V.70 para voz e dados
- G.729 Técnica de codificação LD-CELP gerando uma taxa de bits de 8 kpbs e G.729A a codificação CS-ACELP (Algebraic-ACELP)

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 ñ linear	3,4	8	64	20	160
G.723.1m	MP-MLQ	3,4	8	6.4	30	24
G.723.1a	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

# Recomendações ITU-TS para Voz



## ■ G.729

- Em cada janela de 10ms do sinal de voz são analisadas 80 amostras de 8 bits para geração de 10 códigos de 8 bits
- Valor típico de tamanho do pacote de voz é de 20ms (20 bytes)

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 ñ linear	3,4	8	64	20	160
G.723.1m	MP-MLQ	3,4	8	6.4	30	24
G.723.1a	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



# Supressão de Silêncio e remoção de sons repetitivos



- **Compressão da voz via remoção dos períodos de silêncio e de informações redundantes encontradas na fala humana**
  - Existem informações na fala humana que não são necessárias para que uma comunicação efetiva exista através de uma rede
- **Sons repetitivos, inerentes à voz, são causados pela vibração das cordas vocais**
  - 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
- **Composição da fala**
  - 22% do que se fala são componentes essenciais da comunicação
    - devem ser transmitidos para o entendimento do diálogo
  - 22% são padrões repetitivos
  - 56% representa as pausas entre falas

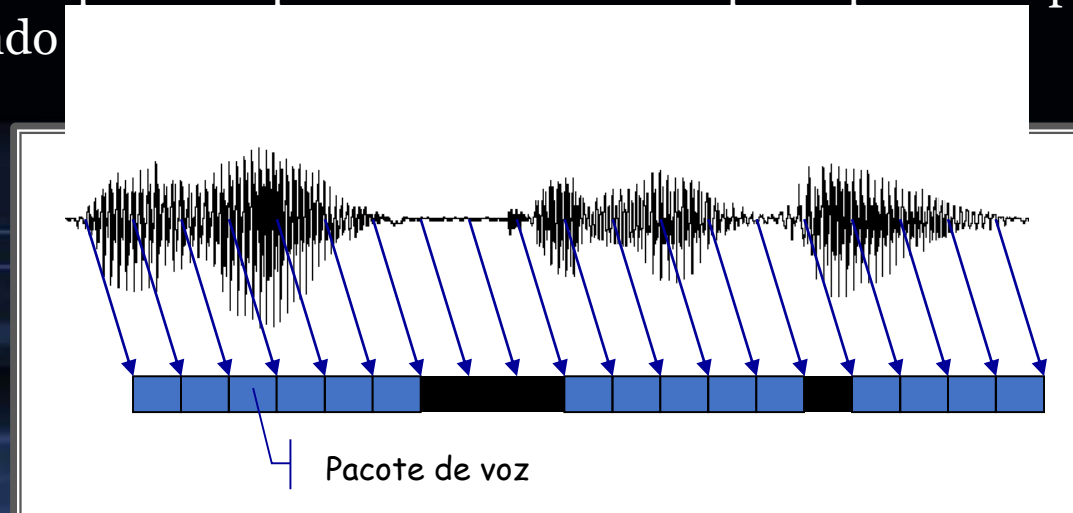
# Supressão do silêncio: Componentes

- **VAD (Detector de Presença de Voz)**
  - Responsável por determinar quando o usuário está conversando e quando ele está em silêncio
  - É útil para economizar energia no caso de dispositivos que funcionam a bateria
  - Deve ser bastante sensível
    - Caso contrário, o início das palavras podem ser perdidas e um silêncio inútil pode ser incluído no final das sentenças
    - Mas ao mesmo tempo, não pode ser tão sensível a ruídos de fundo.



# Supressão do silêncio: Componentes

- **DTX (Discontinuous Transmission)**
  - Capacidade de um codec de parar de transmitir quadros quando o VAD tiver detectado um período de silêncio
  - VAD + DTX: 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



# Supressão de Silêncio e remoção de sons repetitivos



- **Alguns pontos devem ser considerados na supressão do silêncio**
  - Quando a fala é muito frequente, contínua, os ganhos com a supressão do silêncio não são alcançados;
  - Como a detecção da presença de voz na transmissão não é imediata
    - Pode ocorrer o corte das primeiras sílabas da locução
    - 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.

# Pontos Importantes

## Codecs de Voz

- Entender o que são pacotes de voz e a relação do tamanho do pacote de voz com taxa de bits, atraso e impacto na qualidade quando da perda de pacotes

## Supressão de Silêncio

- Entender as vantagens e limitações





**UFSC**

**Ciências da  
Computação**

# **CAP 3. COMPRESSÃO DE DADOS MULTIMÍDIA**

**INE5431 Sistemas Multimídia**  
**Prof. Roberto Willrich (INE/UFSC)**  
**roberto.willrich@ufsc.br**

**Aula 8: Padrões de compressão  
multimídia – MPEG Áudio**

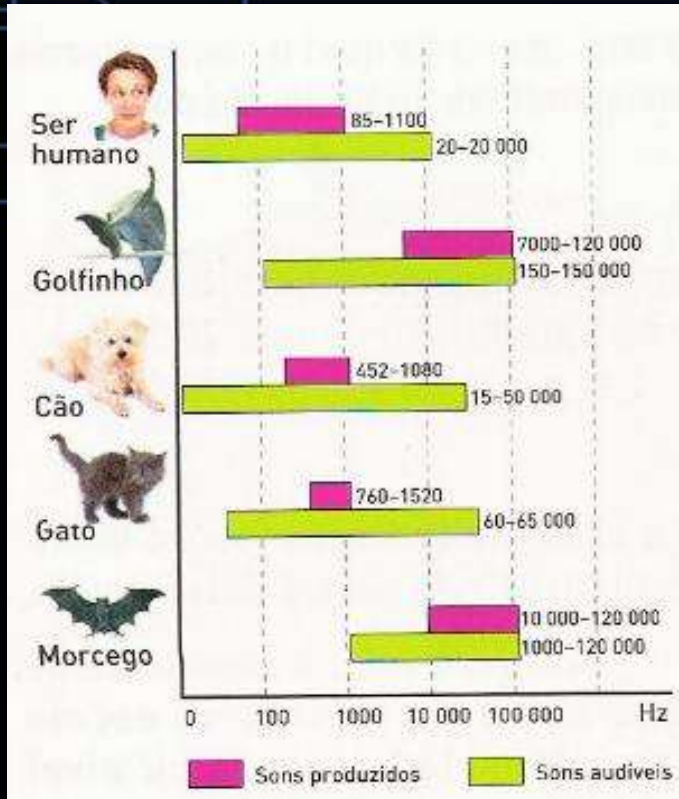
# Compressão de Dados Multimídia



- **Conteúdo:**

- Necessidade de compressão
- Princípios da compressão
- Classificação das técnicas de compressão
- Medição do desempenho de compressão
- Técnicas de compressão sem perdas
  - RLE, Huffman, LZW (GIF), Codificação Preditiva
- Técnicas de compressão de áudio, vídeo e imagens
- **Padrões de compressão multimídia**
  - JPEG, Codecs de Voz, **MPEG Áudio**, MPEG Vídeo, MPEG-4, H.261, H.263

# MPEG-Audio



- **Padrão de compressão de áudio genérico (até 20 kHz)**

- E não apenas para voz (de 3,4 a 7 kHz)
- Explora a percepção humana e não as características da fonte do áudio

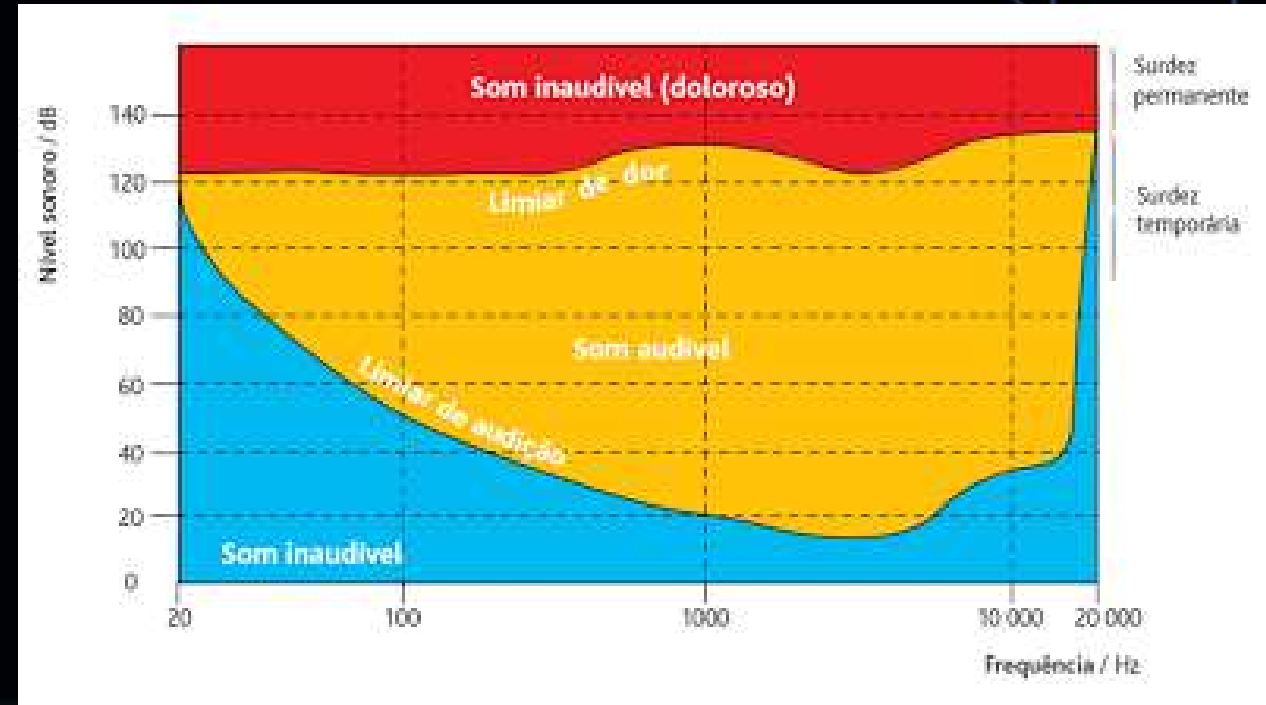
- **Princípio de Compressão**

- Faixa de frequência audível humana
  - Filtra sons acima de 20 KHz

# MPEG-Audio

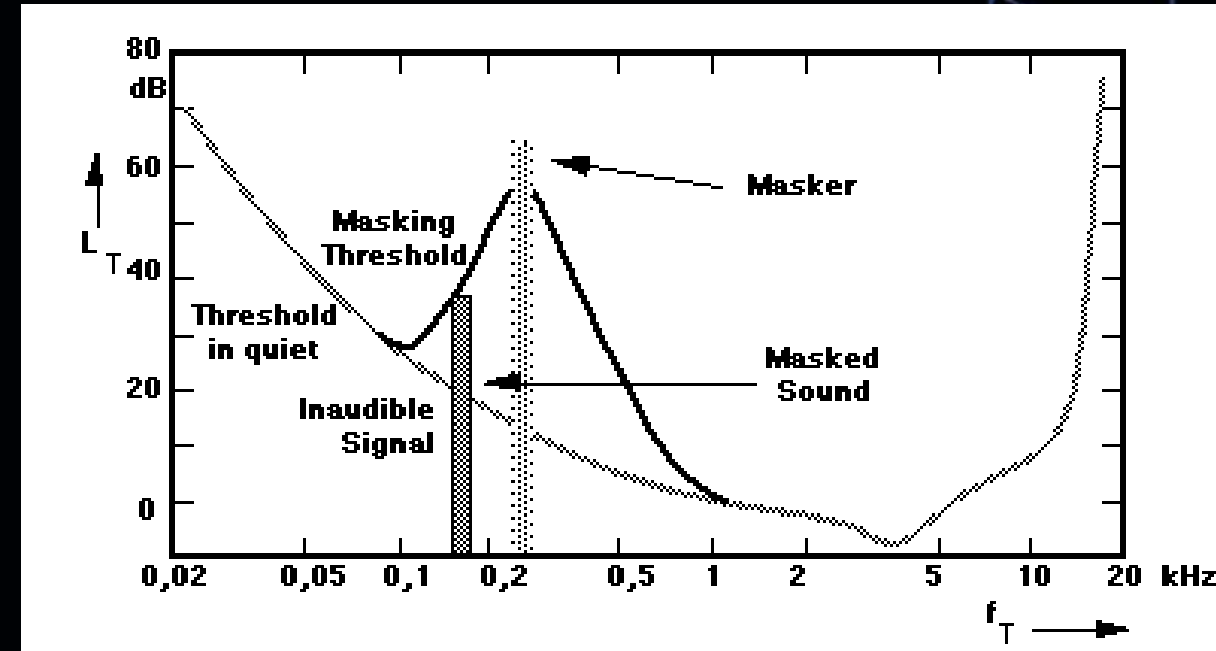
## ■ Princípio de Compressão

- Limiar de audição na faixa de frequência audível
  - Explora a curva de percepção da audição humana dentro da faixa de frequências audíveis (limiar de audição)
    - Sensibilidade para sons dentro desta faixa não é uniforme (depende da frequência)
    - O que se faz é descartar amostras que se encontrem abaixo deste limiar.



# MPEG-Audio

- **Princípio de Compressão**
  - Mascaragem: um som pode tornar outro impossível de ser ouvido, ou pode tornar o outro sem peso
    - tipos de mascaragem: total ou parcial
    - sons mascarados podem ser descartados (não são audíveis)
    - característica explorada pelo padrão MPEG-Áudio
      - explora as limitações perceptivas do sistema auditivo humano





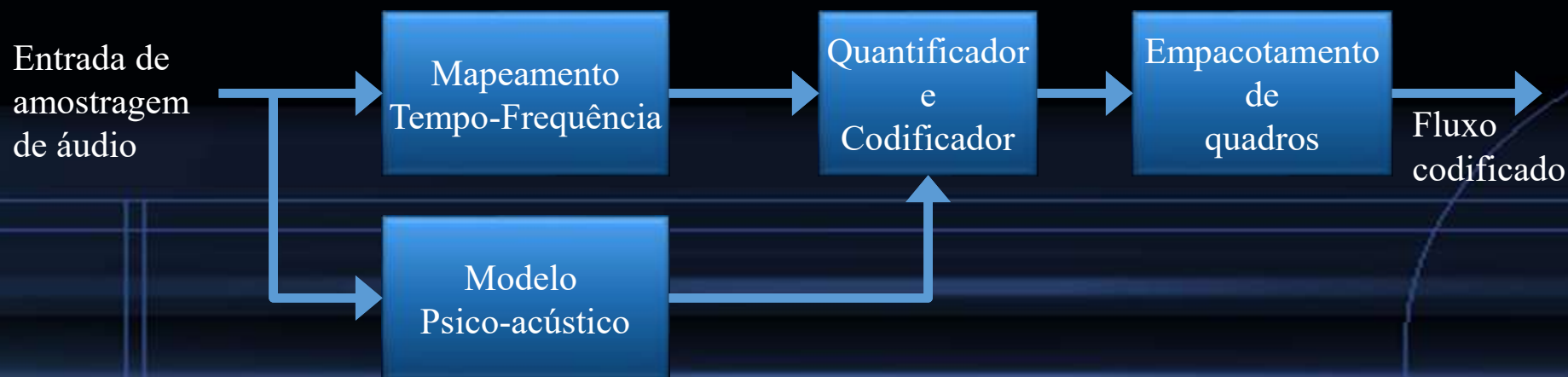
# MPEG-Audio



- **Principais características do MPEG-1 Audio:**
  - Sequência de bits compactada pode suportar um ou dois canais
    - um canal único
    - dois canais independentes
    - um sinal estéreo
  - Três taxas de amostragens
    - 32, 44.1 ou 48 kHz
    - MPEG-2.5 (não oficial) 8, 11.025, 12, 16, 22.05 e 24 kHz.
  - Fluxo compactado pode ter uma das várias taxas de bits fixas e predefinidas variando de 32 a 320 kbps
  - Padrão MPEG-2.5 (não oficial)
    - Taxa de bits de 8, 16, 24, e 144 kbps
  - Razão de compressão: 2,7 a 24 (depende da taxa de amostragem)
    - 6:1 ouvintes experientes não detectam diferenças

# Um codificador básico MPEG-Áudio

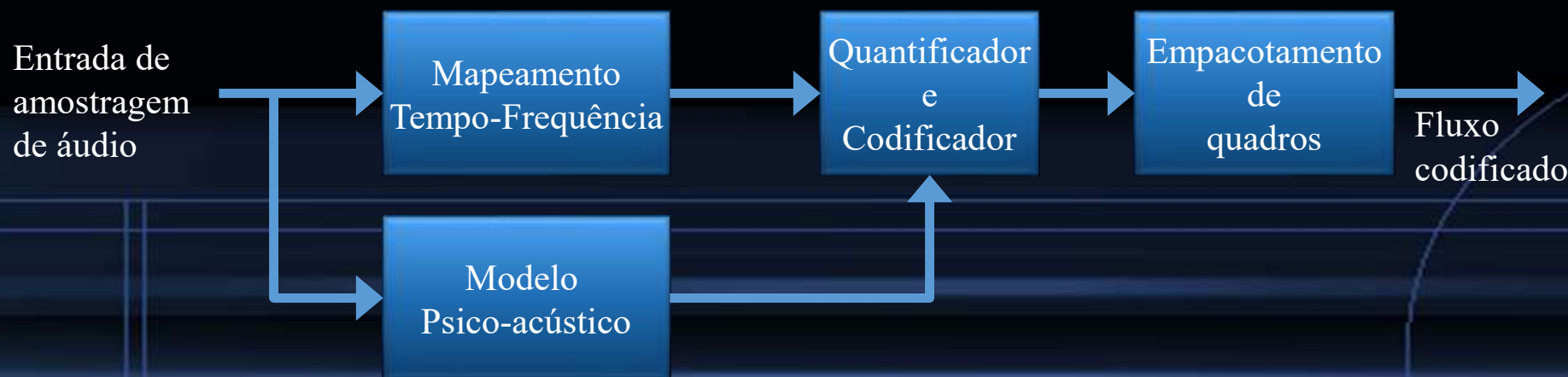
- **Bloco mapeamento tempo-frequência**
  - Divide a entrada em sub-bandas de frequências múltiplas
- **Bloco modelo psico-acústico**
  - Cria um conjunto de dados para controlar a operação do bloco quantificador e codificador
    - Considera limiar de audição, sons mascarados, etc.



# Um codificador básico MPEG-Áudio



- **Bloco quantificador e codificador**
  - Cria um conjunto de símbolos de código
    - sub-bandas menos importantes e áudios inaudíveis são removidos
- **Bloco Empacotamento de quadros**
  - Monta e formata os símbolos de código e adiciona outras informações



# MPEG-1 Audio



- **MPEG Audio especifica uma família de 3 esquemas de codificação de áudio**
  - Chamadas de Layer-1, Layer-2 e Layer-3
    - 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
    - decodificador Layer-N é capaz de decodificar um fluxo de bits fluxo codificado com codificador Layer-N e abaixo de N
  - MP3 é MPEG-1 Layer-3
- **Padrão especifica o formato do fluxo de bits e o decodificador para cada esquema de codificação**
  - não especifica o codificador para avanços futuros

# MPEG-2 Áudio



- **Estende as funcionalidades do MPEG-1 Áudio**
  - cinco canais (esquerdo, direito, centro, e dois canais surround)
  - mais um canal de baixa frequência
  - ou sete canais multilíngues/comentários
  - tem taxas de amostragens adicionais



# Pontos Importantes

## MPEG Áudio

- Entender os princípios gerais da compressão



**UFSC**

**Ciências da  
Computação**

# **CAP 3. COMPRESSÃO DE DADOS MULTIMÍDIA**

**INE5431 Sistemas Multimídia**  
**Prof. Roberto Willrich (INE/UFSC)**  
**roberto.willrich@ufsc.br**

**Aula 9: Padrões de compressão  
multimídia – MPEG Vídeo**

# Compressão de Dados Multimídia



- **Conteúdo:**

- Necessidade de compressão
- Princípios da compressão
- Classificação das técnicas de compressão
- Medição do desempenho de compressão
- Técnicas de compressão sem perdas
  - RLE, Huffman, LZW (GIF), Codificação Preditiva
- Técnicas de compressão de áudio, vídeo e imagens
- **Padrões de compressão multimídia**
  - JPEG, Codecs de Voz, MPEG Áudio, **MPEG Vídeo**, H.261, H.263

# Padrão ISO/IEC MPEG



- **Motion Picture Expert Group**

- Grupo de padrões de representação codificada de vídeos, áudios e suas combinações
  - armazenados e recuperados em Digital Storage Media (DSM)
    - dispositivos de armazenamento convencionais, CD-ROMs, drivers de fita, HDs, drivers ótico escrevíveis e canais de telecomunicação (redes de longa distância, locais, etc.)

- **Características**

- MPEG usa a compressão intra e inter-quadros de vídeos
  - obtém altas taxas de compressão devido a alta redundância dos vídeos
- Especificações MPEG também incluem um algoritmo para compressão de áudio
  - Compressão do áudio associado e a sincronização áudio-visual não podem ser independente da compressão do vídeo

# Padrão ISO/IEC MPEG



- **Vários itens de trabalho**

- MPEG-1 (1993)

- Vídeo pode codificar imagens de até 4096x4096 pixels e 60 fps.
    - A maioria das aplicações usam o formato SIF, com 240x352 pixels e 30fps, e subamostragem de croma 4:2:0.

- MPEG-2 (1994)

- Pode codificar imagens de até 16.383 x 16.383 pixels
    - Padrão organizado em perfis e níveis. Exemplos:
      - Nível baixo (240 x 352 pixels x 30 fps – idêntico ao SIF MPEG 1),
      - Nível principal, visando a codificação com qualidade de TV (720 x 480 e 30 fps), e
      - Níveis alto, visando a TV de alta resolução - HDTV, e a produção de filmes (em geral 1280 x 720 e 30fps; 1920 x 1080 e 30 fps ou 1440 x 1152 e 30 fps).
    - O padrão permite subamostragem de croma 4:2:0, 4:2:2 e 4:4:4.



# Padrão ISO/IEC MPEG



- **Vários itens de trabalho**

- MPEG-3

- Para vídeo com qualidade HDTV na taxa de 40 Mbps
    - interrompido em julho 1992

- MPEG-4 (1998)

- Objetivo inicial: codificação para audiovisual com muito baixa taxa de transmissão (variando de 4,8 a 64 Kbps)
    - Hoje: oferece soluções para vários tipos de aplicações com qualidades diferentes

- MPEG-7 (2001)

- Interface de Descrição de Conteúdo Multimídia: um padrão de descrição de dados multimídia (informações audiovisuais)
    - Permitindo a busca e filtragem

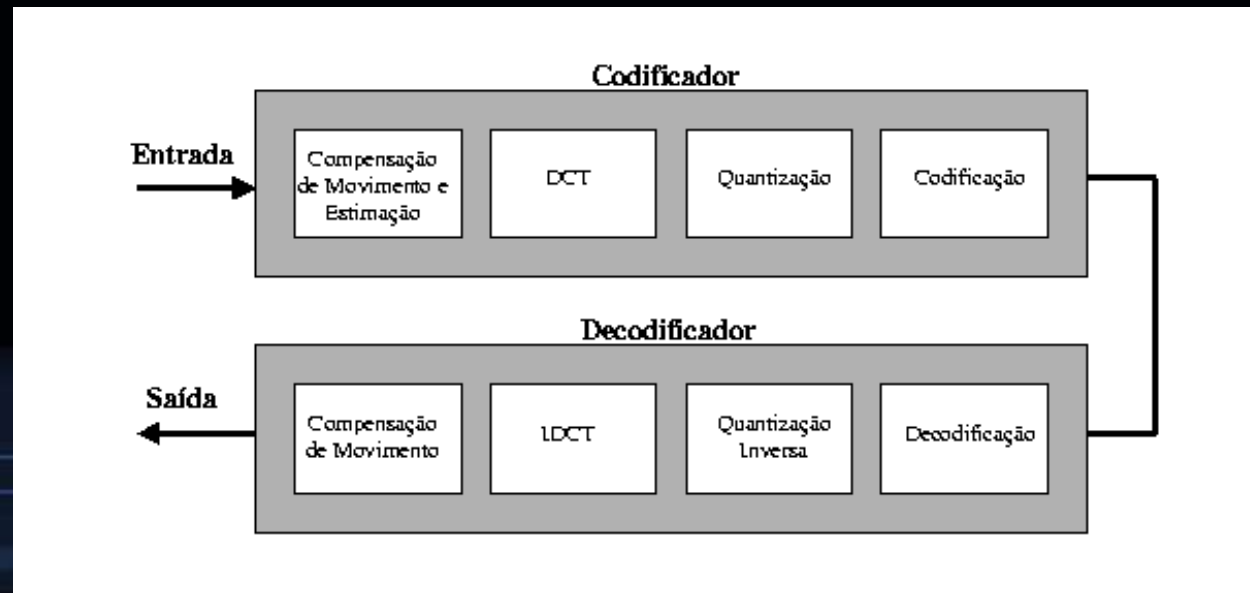
# Padrão ISO/IEC MPEG 1



- **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.
- **MPEG separa áudio e vídeo**
  - Compressão em três camadas: a camada de sistema, de áudio e de vídeo

# Padrão ISO/IEC MPEG-1 Vídeo

- **MPEG especifica a sintaxe dos fluxos codificados para que decodificadores possam decodificar**
  - como gerar o bitstream não é padronizado
    - permite inovações no projeto e implementação de codificadores



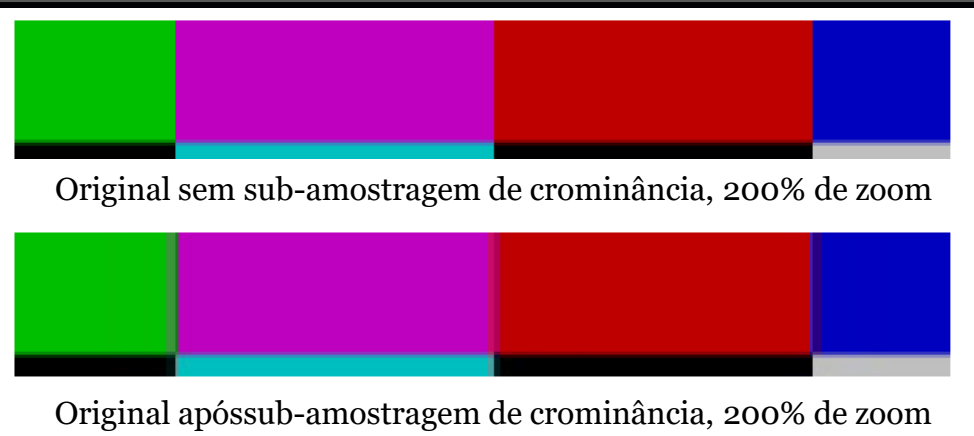
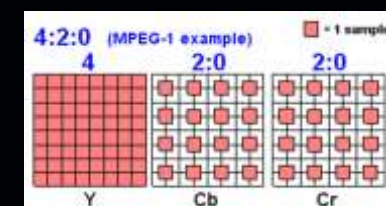
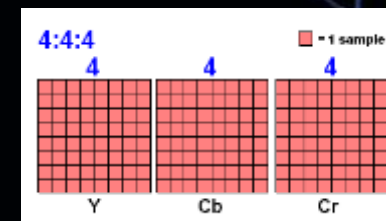
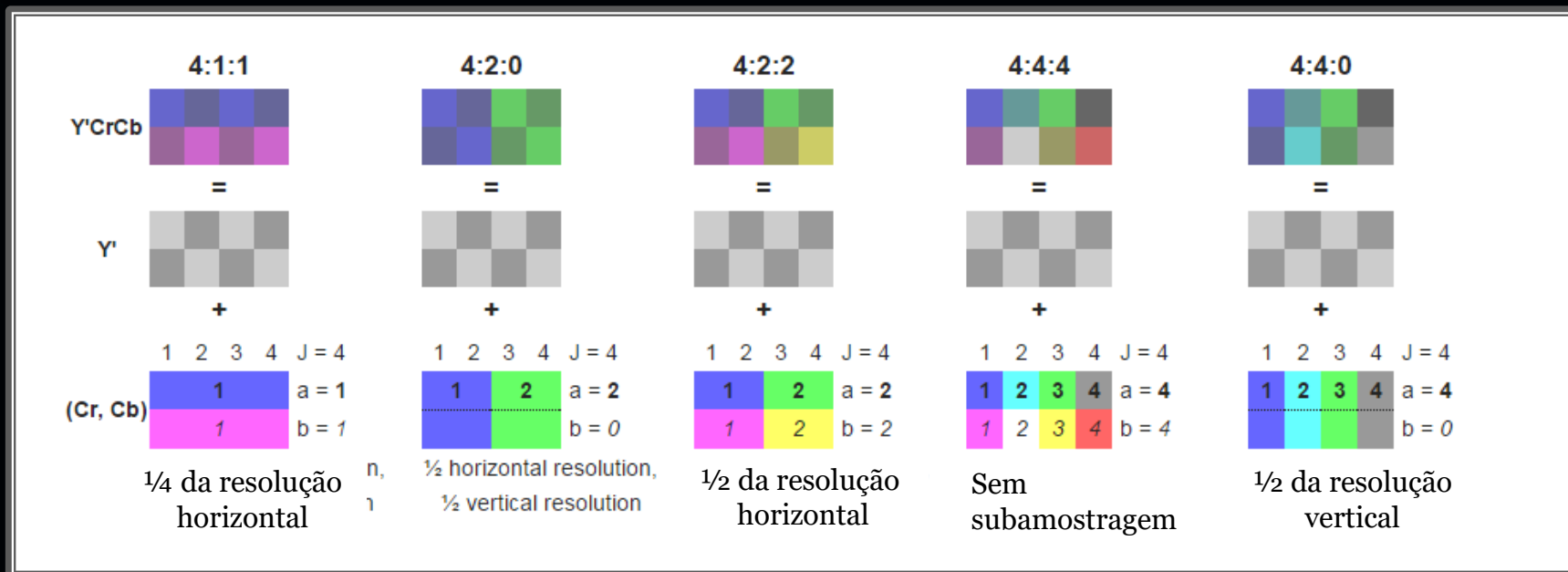
# Padrão ISO/IEC MPEG-1 Vídeo

- **Imagem MPEG (quadro do vídeo)**
  - É a unidade elementar para a codificação do vídeo
  - Imagens são representadas no espaço de cores YCbCr
    - Grupo de três matrizes retangulares que representam a luminância (Y) e a crominância (Cr e Cb)
    - É preferível YCrCb pois o olho é mais sensível a luminosidade que a crominância
      - Podem ser feita subamostragens nas matrizes Cb e Cr



# Padrão ISO/IEC MPEG-1 Vídeo

## ■ Subamostragens YCbCr





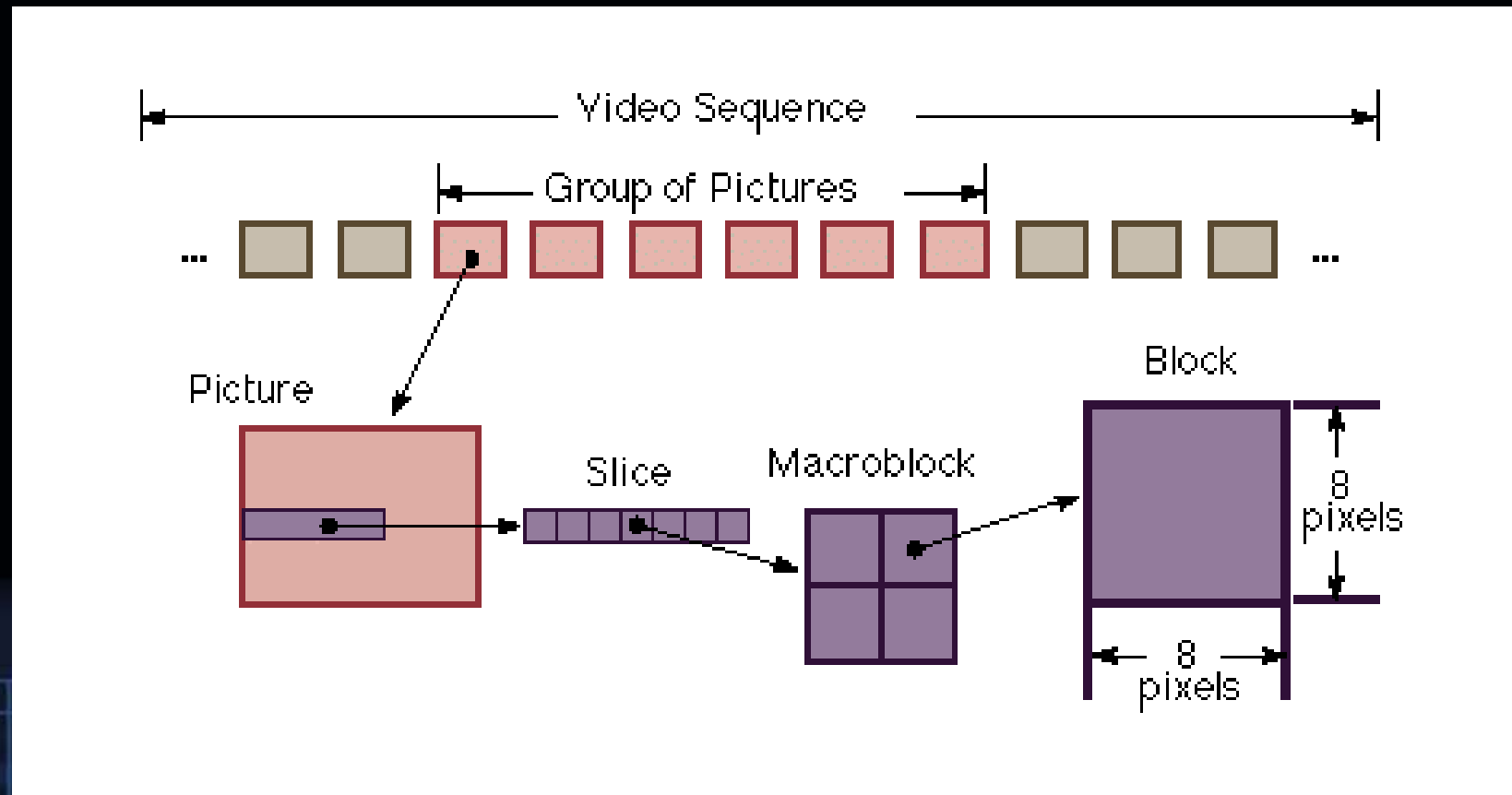
# Padrão ISO/IEC MPEG-1 Vídeo



- **Compactação MPEG-1 Video**
  - Em vídeo existem dois tipos de redundância: espacial e temporal
    - MPEG-1 explora estas duas redundâncias
- **Redundância espacial**
  - Pode ser explorada pela simples codificação em separado de cada quadro
  - Blocos de 8x8 pixels são compactados similar a uma compressão JPEG

# Padrão ISO/IEC MPEG-1 Vídeo

- Hierarquia do fluxo de dados MPEG-1**



# Padrão ISO/IEC MPEG-1 Vídeo

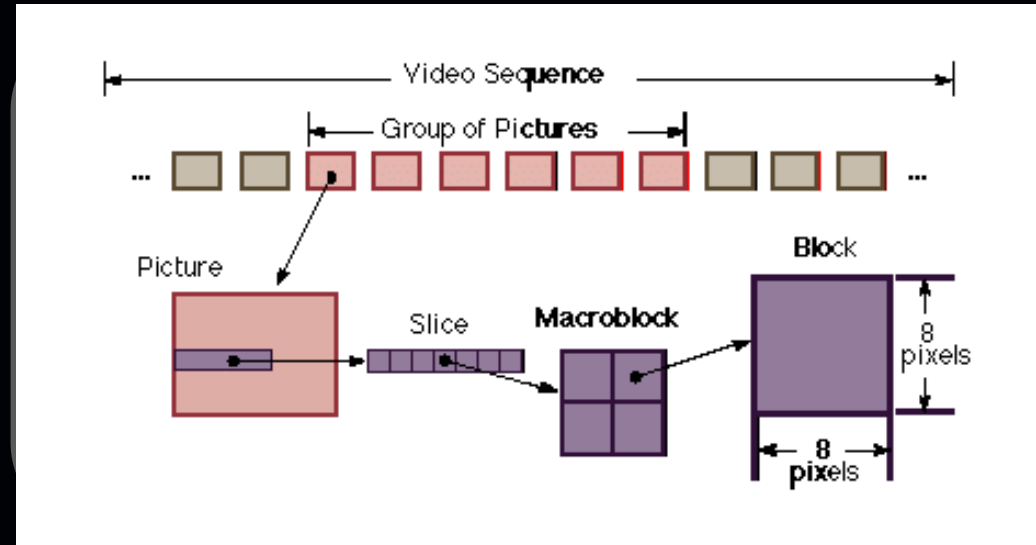
## ■ Fluxo de vídeo MPEG-1:



- GOP (Grupo de imagens): fornece um ponto de acesso aleatório
- Camada de imagem contem todas as informações codificadas de uma imagem
  - cabeçalho contem a referência temporal de uma imagem, o tipo de codificação, etc..

# Padrão ISO/IEC MPEG-1 Vídeo

## ■ Fluxo de vídeo

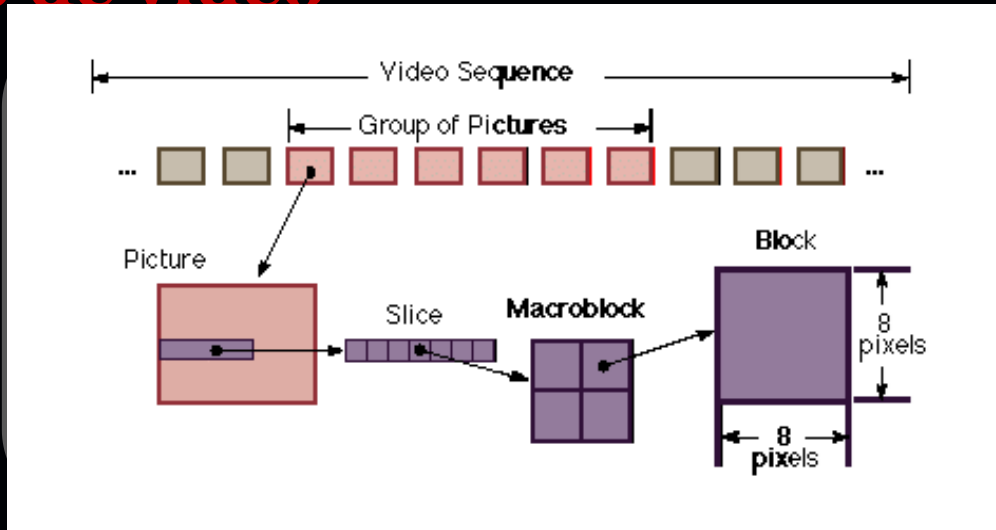


## ■ Imagens (Quadro do Vídeo)

- Podem ser codificadas com diferentes I-Frame, P-Frame, B-Frame
  - Compactados eliminando redundância espacial (I-Frame) ou espacial/temporal (P e B-Frames)
  - Compactados descartando informações pouco relevantes para a percepção humana

# Padrão ISO/IEC MPEG-1 Vídeo

## ■ Fluxo de vídeo

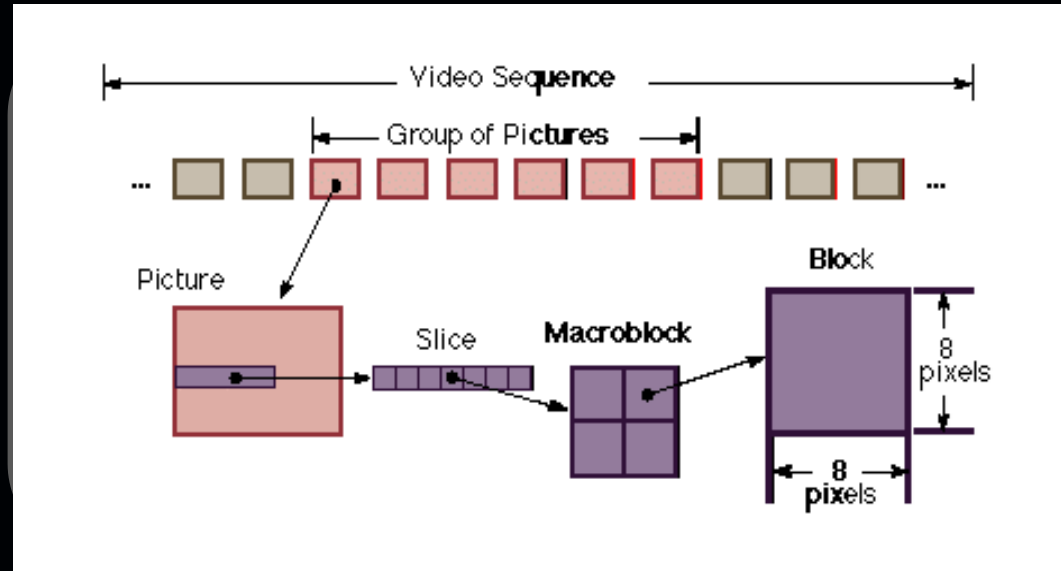


- Imagens são divididas em pedaços (slices):
  - Número de pedaços podem ser variáveis
  - Cada pedaço consiste de um número variável de macroblocos (16x16 pixels)
  - Cada pedaço pode ser codificados de maneira diferente
  - Importante para o controle de erro
    - Se existe um erro no fluxo de dados, o decodificador pode saltar um pedaço
    - Maior o número de pedaços, melhor é o tratamento de erro



# Padrão ISO/IEC MPEG-1 Vídeo

## ■ Fluxo de vídeo



### □ Macrobloco

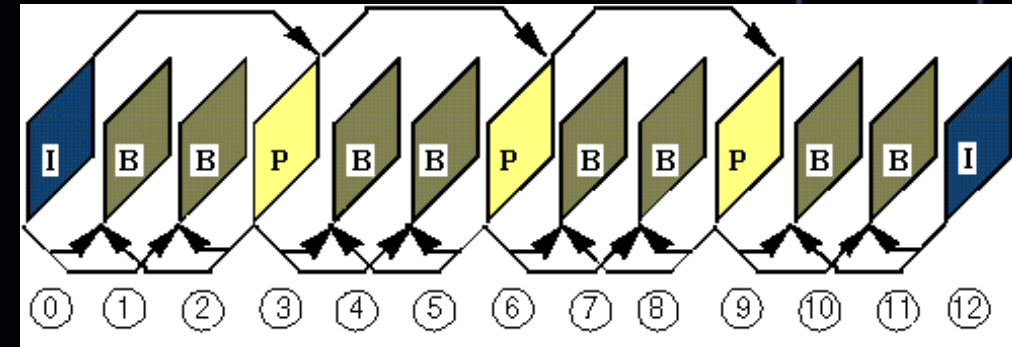
- Usado na estimativa e compensação do movimento

### □ Bloco de Imagem

- um bloco é uma matriz 8x8 pixels tratados como unidades e entrada para o DCT

# Padrão ISO/IEC MPEG-1 Vídeo

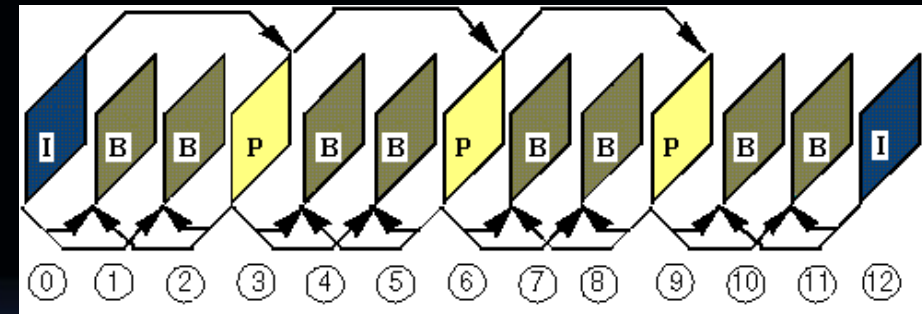
- **Grupo de Imagem consiste de quatro tipos de quadros:**
  - Quadros I (Intracoded)
    - imagens estáticas, independentes e codificadas com o JPEG.
  - Quadro P (Predictive)
    - diferença bloco a bloco com o quadro I ou P anterior
  - Quadro B (Birectional)
    - diferença com o último quadro e com o quadro seguinte
  - Quadro D (DC-coded)
    - Médias de bloco usadas para o avanço rápido (fast forward).



# Padrão ISO/IEC MPEG-1 Vídeo

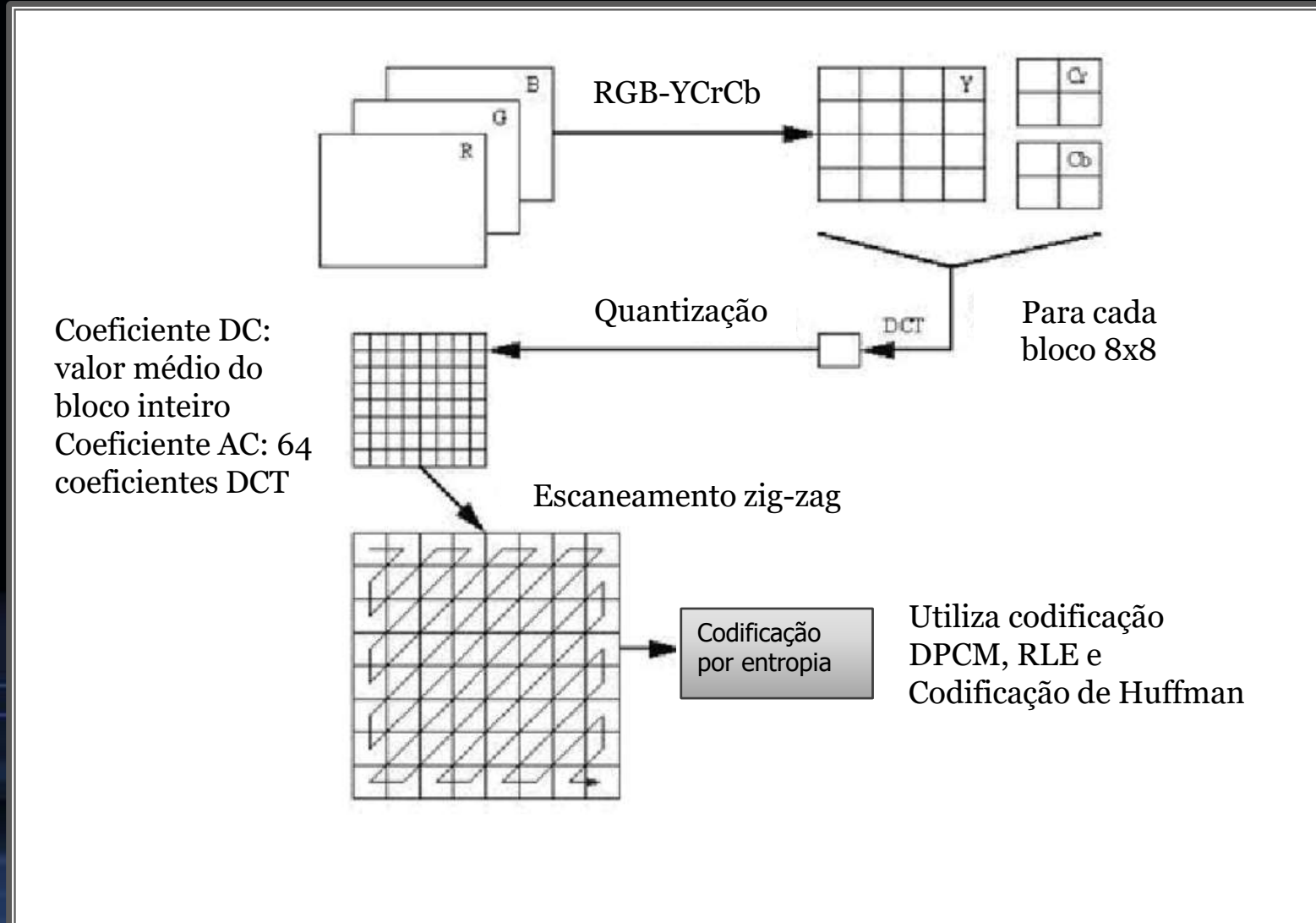
## ■ Quadro I

- Imagens estáticas, independentes e codificadas com o JPEG
- É necessário que quadros I apareçam periodicamente no fluxo de saída
  - no caso de transmissão multicast
    - 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
    - decodificação não será mais possível
- Quadros I são inseridos na saída uma ou duas vezes por segundo



# Padrão ISO/IEC MPEG-1 Vídeo

- **Compressão dos blocos 8x8 píxeis**



# Padrão ISO/IEC MPEG-1 Vídeo

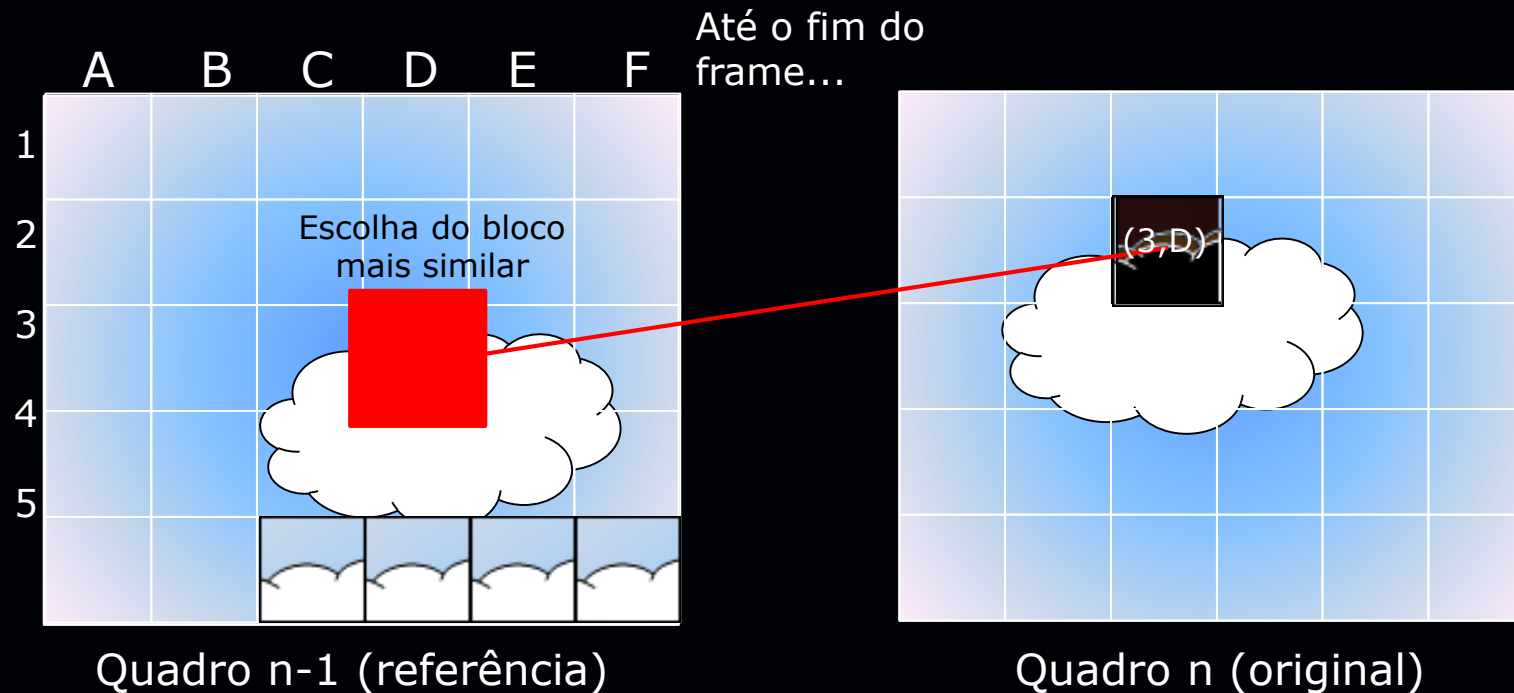
- **Quadros P e B exploram a redundância temporal**
  - Compactação adicional pode ser obtida explorando o fato de que dois quadros consecutivos são, com frequência, quase idêntico
  - MPEG faz compensação de movimento
    - Calcula o vetor de movimento dos macroblocos e a diferença macrobloco a macrobloco





# Padrão ISO/IEC MPEG-1 Vídeo

- **Exemplo simples: Compara a similaridade entre blocos**

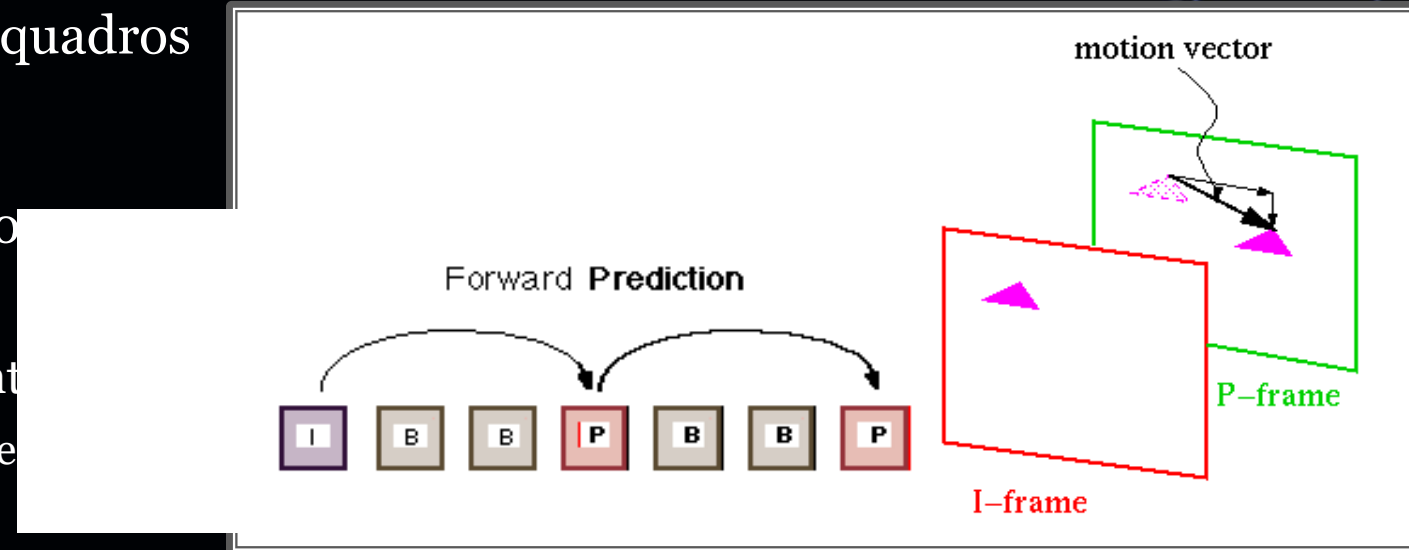


- ▣ Mantém a diferença entre os blocos (resíduo);
- ▣ Cria o vetor de movimento, referenciando o bloco do quadro anterior;

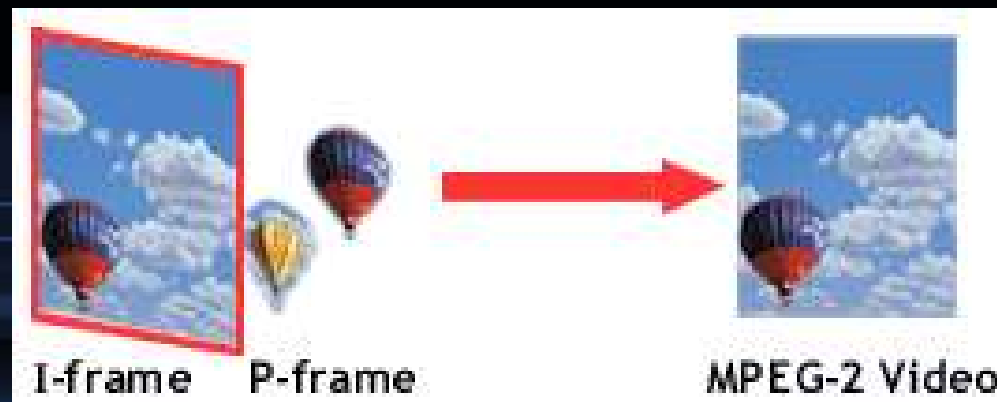
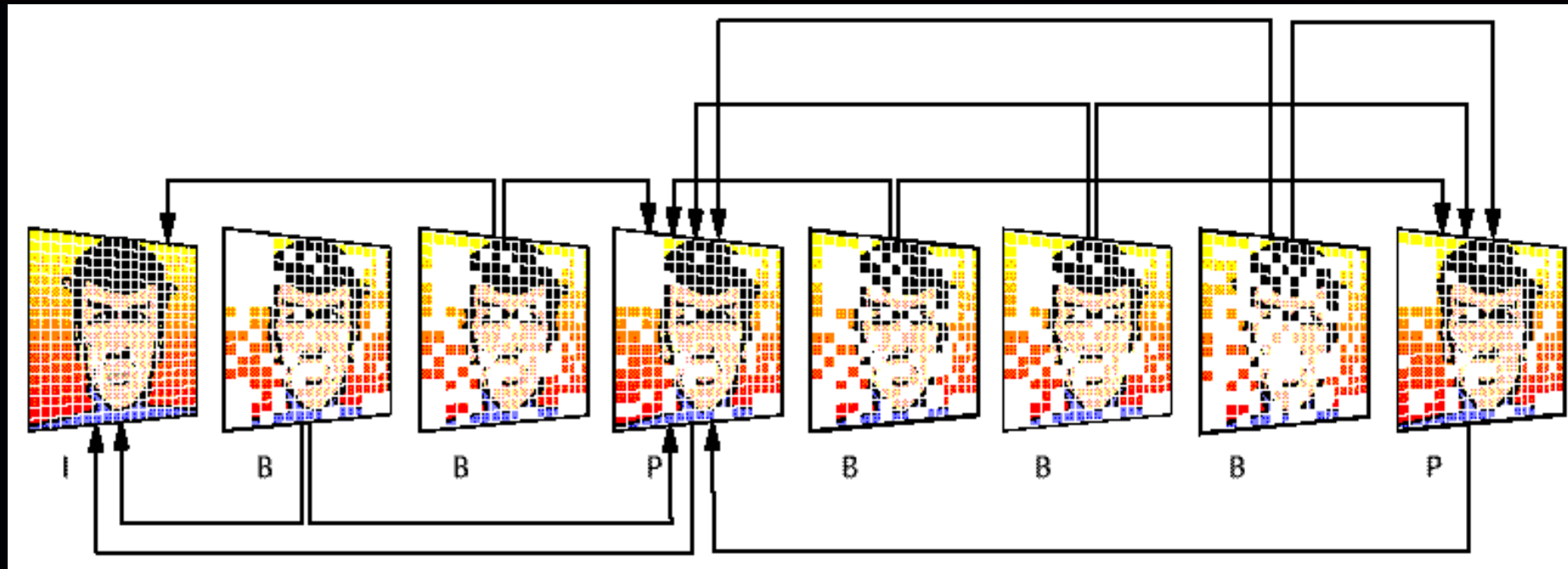
# Compactação MPEG-1 Video

## ■ Quadro P

- Codificam as diferenças entre os quadros
  - 50% do tamanho de um quadro I
- Se baseiam na idéia dos macrobloco de 16x16 pixels
  - macrobloco é codificado da seguinte forma:
    - tentando-se localizá-lo, ou algo parecido com ele, no quadro anterior
- Decodificar quadros P requer que o decodificador armazene o quadro I ou P anterior em um buffer
  - a partir do qual o novo quadro é construído baseado em macroblocos completamente codificados e macroblocos contendo diferenças com o quadro anterior

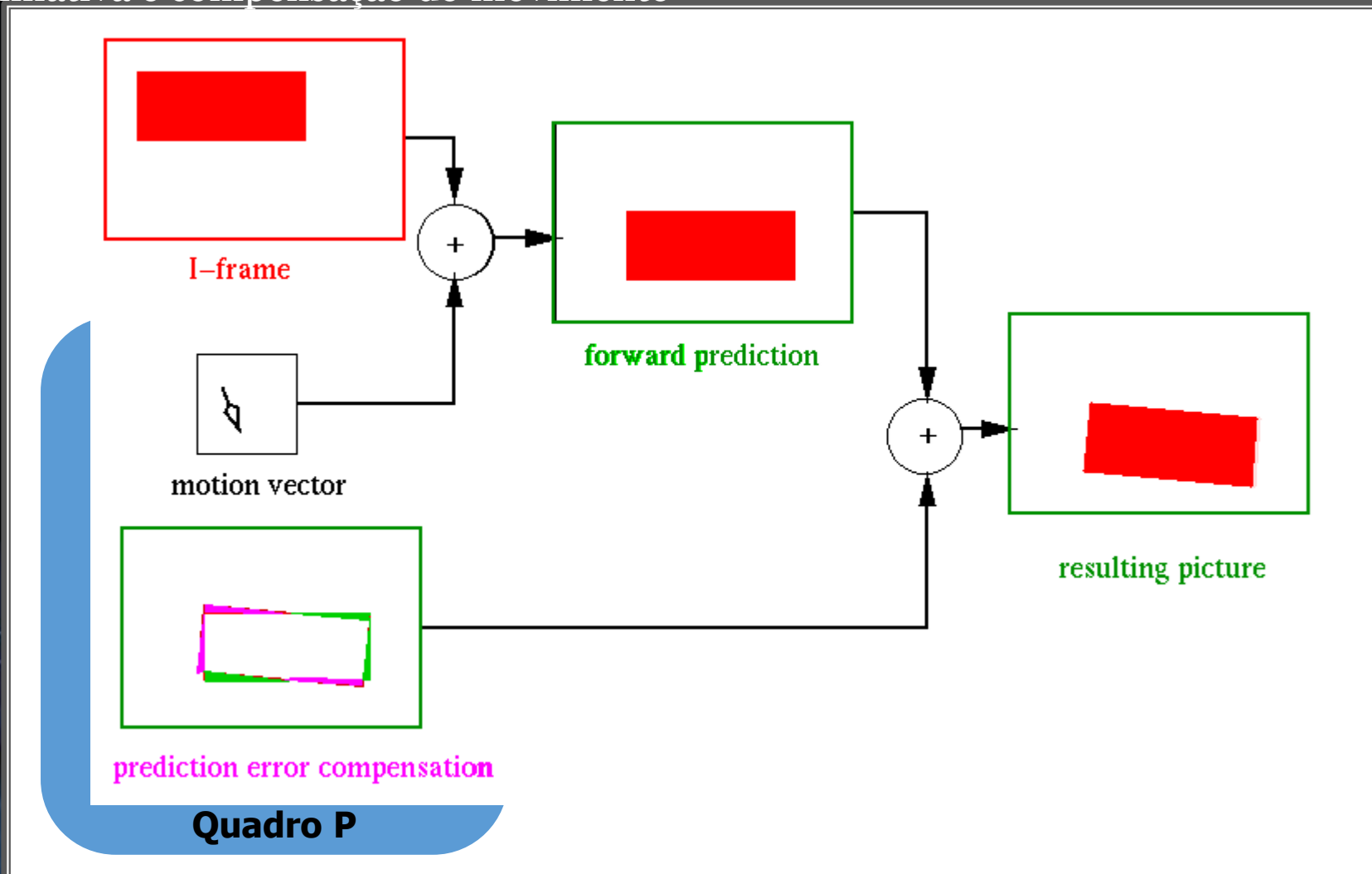


# Compactação MPEG-1 Video



# Compactação MPEG-1 Video

- **Quadro P**
  - Estimativa e compensação do movimento



# Compactação MPEG-1 Video

- **Quadro P**
  - Estimativa e compensação do movimento



I-frame



B-frame



B-frame



B-frame



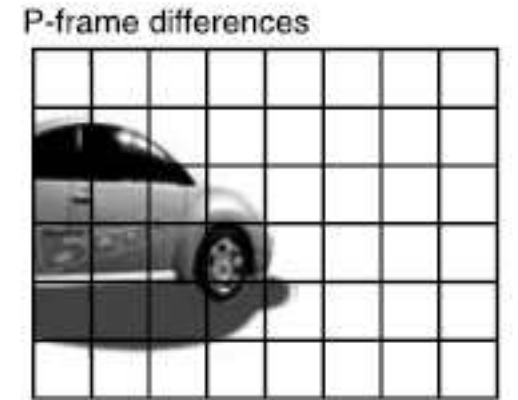
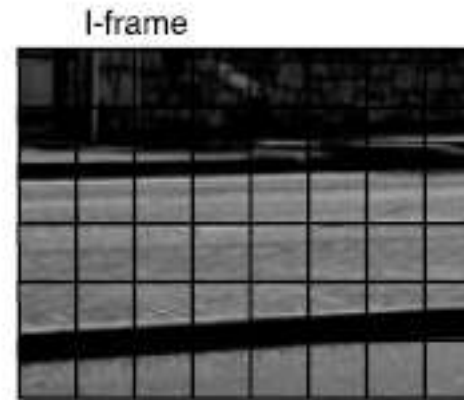
B-frame



B-frame



P-frame

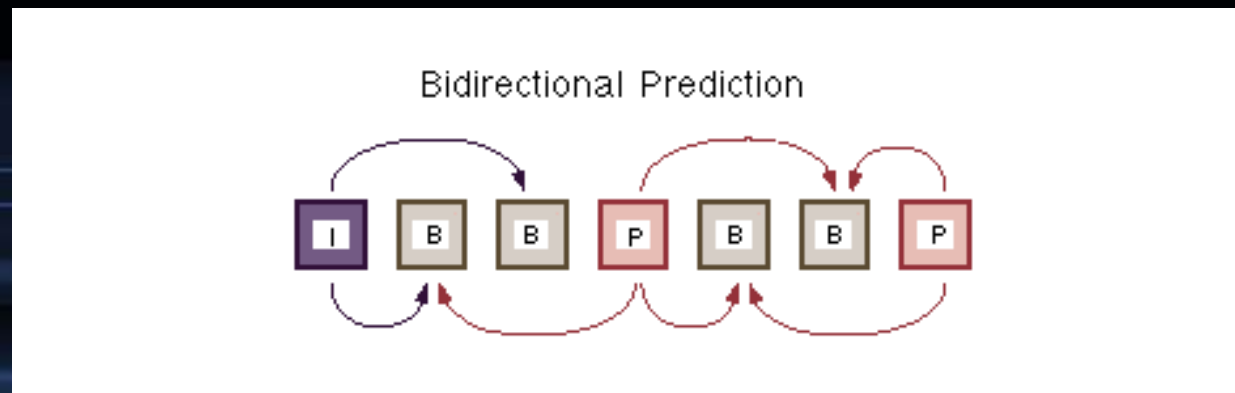




# Compactação MPEG-1 Video

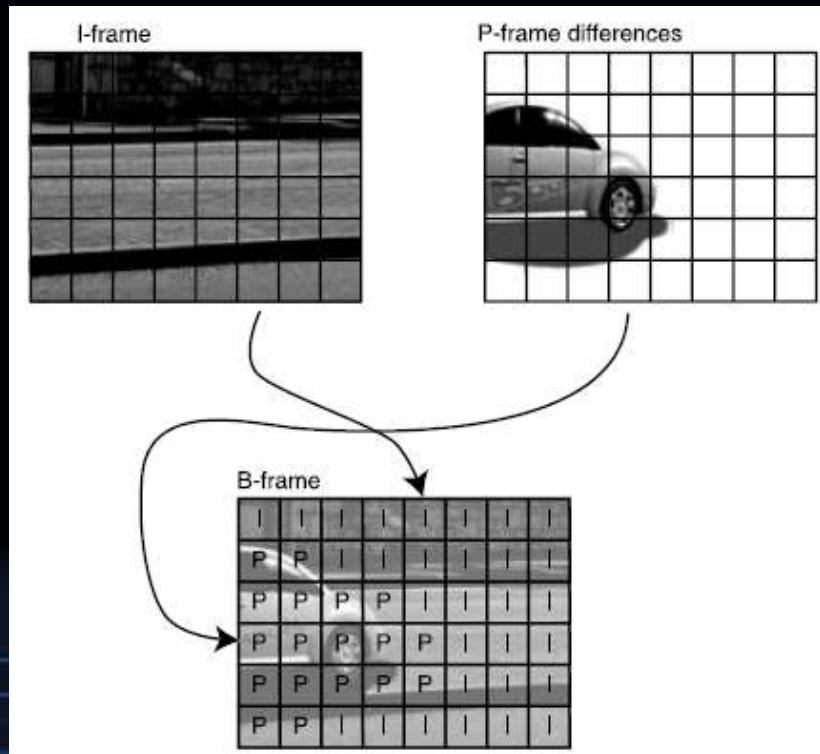
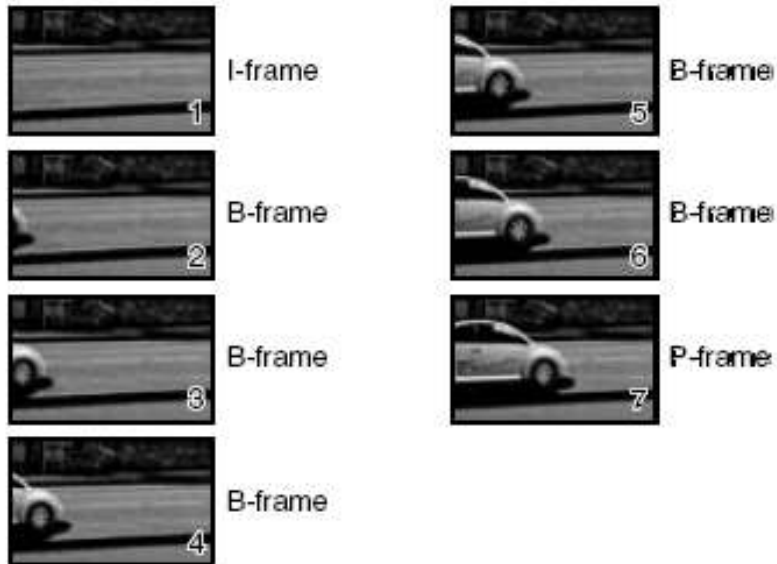
## ■ Quadro B

- Codificam as diferenças com o último quadro I ou P e com o quadro seguinte
  - 15% do tamanho de um quadro I
  - permitem que o macrobloco de referência esteja tanto no quadro anterior quanto no quadro seguinte
  - acarreta uma melhoria na compensação do movimento
- Para decodificar quadros B
  - decodificador precisa manter três quadros decodificados na memória ao mesmo tempo: o quadro anterior, o atual e o próximo



# Compactação MPEG-1 Video

- **Quadro B**
  - Estimativa e compensação do movimento



# Compactação MPEG-1 Video



- **Quadro D**

- Só são usado para possibilitar a apresentação de uma imagem de baixa resolução quando um avanço rápido ou um retrocesso

- **Um fluxo MPEG-1**

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

- **Codificação MPEG-2**

- É fundamentalmente semelhante à codificação MPEG-1
  - com quadros I, P e B
  - quadros D não são aceitos
- Transformação discreta de co-seno é de 10x10 em vez de 8x8
  - para proporcionar mais 50 por cento de coeficientes
    - melhor qualidade

# MPEG-1 Sistemas



- **Define uma estrutura para:**
  - Combinar fluxos elementares, incluindo áudio, vídeo e outros fluxos de dados
    - chamado de Fluxo MPEG
    - até 32 fluxos de áudio MPEG e 16 fluxos de vídeo MPEG podem ser multiplexados juntamente com 2 fluxos de dados de diferentes tipos
- **Especifica o modo de representar as informações temporais necessárias para reprodução de sequências sincronizadas em tempo real**
  - sincronização de fluxos elementares
  - gerenciamento de buffer nos decodificadores
  - acesso aleatório
  - identificação do tempo absoluto do programa codificado

# Pontos Importantes

## MPEG Vídeo (1 e 2)

- Conhecimentos dos princípios da compressão: redundância espacial (similar a JPEG) e temporal (estimativa e compensação do movimento)





**UFSC**

**Ciências da  
Computação**

# **CAP 3. COMPRESSÃO DE DADOS MULTIMÍDIA**

**INE5431 Sistemas Multimídia**  
**Prof. Roberto Willrich (INE/UFSC)**  
**roberto.willrich@ufsc.br**

**Aula 10: Padrões de compressão  
multimídia – MPEG-4 e H.26\***

# Compressão de Dados Multimídia



- **Conteúdo:**

- Necessidade de compressão
- Princípios da compressão
- Classificação das técnicas de compressão
- Medição do desempenho de compressão
- Técnicas de compressão sem perdas
  - RLE, Huffman, LZW (GIF), Codificação Preditiva
- Técnicas de compressão de áudio, vídeo e imagens
- **Padrões de compressão multimídia**
  - JPEG, Codecs de Voz, MPEG Áudio, MPEG Vídeo, **MPEG-4, H.26\***

# MPEG-4



- **Padrão MPEG-4**

- 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, adicionando novas características

- **Uso**

- 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)

# MPEG-4



## ■ Partes (padrões)

Parte	Data	Título	Descrição
<b>Part 1</b>	2010	Systems	Descreve sincronização e multiplexação de áudio e vídeo. Especifica o MPEG-TS (Transport Stream)
<b>Part 2</b>	2004	Visual	Compressão de vídeo.
<b>Part 3</b>	2009	Audio	Formatos de compressão de áudio: AAC, ALS, SLS.
<b>Part 4</b>	2004	Conformance testing	Procedimento para testes de conformidade
<b>Part 5</b>	2001	Reference software	Software de referência.
<b>Part 6</b>	2000	Delivery Multimedia Integration Framework (DMIF)	Interface entre a aplicação e o transporte
<b>Part 7</b>	2004	Optimized reference software for coding of audio-visual objects	Exemplos de como melhorar implementação.
<b>Part 8</b>	2004	Carriage of ISO/IEC 14496 contents over IP networks	Métodos para transportar conteúdo MPEG-4 em redes IP.
<b>Part 9</b>	2009	Reference hardware description	Provê projetos de hardware
<b>Part 10</b>	2012	Advanced Video Coding (AVC)	Um formato de compressão para vídeo (ITU-T H.264).

# MPEG-4

## ■ Partes (padrões)



Parte	Data	Título	Descrição
<b>Part 11</b>	2005	Scene description and application engine	BIFS, XMT, MPEG-J. Define posicionamento de objetos, representação de objetos sintéticos 2D e 3D, ...
<b>Part 12</b>	2012	ISO base media file format	Um formato de arquivo para armazenar conteúdo de mídia baseado em tempo
<b>Part 13</b>	2004	Intellectual Property Management and Protection (IPMP) Extensions	Gerenciamento de propriedade intelectual e proteção
<b>Part 14</b>	2003	MP4 file format	Formato de arquivo MPEG-4 versão 2
<b>Part 15</b>	2010	Advanced Video Coding (AVC) file format	Formato de arquivo MPEG-4 Parte 10
<b>Part 16</b>	2011	Animation Framework eXtension (AFX)	Especifica o modelo MPEG-4 AFX para representação de conteúdos gráficos 3D
<b>Part 17</b>	2006	Streaming text format	Formato de legenda
<b>Part 18</b>	2004	Font compression and streaming	
<b>Part 19</b>	2004	Synthesized texture stream	
<b>Part 20</b>	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.



# MPEG-4

## ■ Partes (padrões)

Parte	Data	Título	Descrição
<b>Part 21</b>	2006	MPEG-J Graphics Framework eXtensions (GFX)	Descreve um ambiente programático leve para aplicações multimídia interativas.
<b>Part 22</b>	2009	Open Font Format	
<b>Part 23</b>	2008	Symbolic Music Representation (SMR)	
<b>Part 24</b>	2008	Audio and systems interaction	
<b>Part 25</b>	2009	3D Graphics Compression Model	
<b>Part 26</b>	2010	Audio Conformance	
<b>Part 27</b>	2009	3D Graphics conformance	
<b>Part 28</b>	2012	Composite font representation	
<b>Part 29</b>	2014	Web video coding	
<b>Part 30</b>	2014	Timed text and other visual overlays in ISO base media file format	
<b>Part 31</b>	Em desenvol.	Video Coding for Browsers (VCB)	Codec para browsers
<b>Part 33</b>	Em desenvol.	Internet video coding	

# MPEG-4



## ■ MPEG-4 Parte 2

- Um padrão de compressão de vídeo DCT similar aos padrões MPEG-1 e MPEG-2
- 21 Perfis (Profiles)
  - Agrupam características em perfis (profiles) e níveis.
    - Para permitir seu uso em várias aplicações, variando de câmeras de segurança de baixa qualidade, baixa resolução a HDTVs e DVDs,
  - Perfil Simple Profile (SP): 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
    - telefones celulares, sistemas de segurança, etc.
  - Perfil Advanced Simple Profile (ASP): 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



## ■ MPEG-4 Parte 2

### ▫ Perfil Simple Studio Profile (SStP)

- Tem 6 níveis indo de SDTV até a resolução 4K
- Permite até profundidade de píxel de 12-bits e subamostragem de croma 4:4:4

Nível	Max bit depth and chroma subsampling	Max resolution and frame rate	Max data rate (Mbit/s)
1	10-bit 4:2:2	SDTV (e.g. 704x480)	180
2	10-bit 4:2:2	1920×1080 30p/30i	600
3	12-bit 4:4:4	1920×1080 30p/30i	900
4	12-bit 4:4:4	2K×2K 30p	1.350
5	12-bit 4:4:4	4K×2K 30p	1.800
6	12-bit 4:4:4	4K×2K 60p	3.600

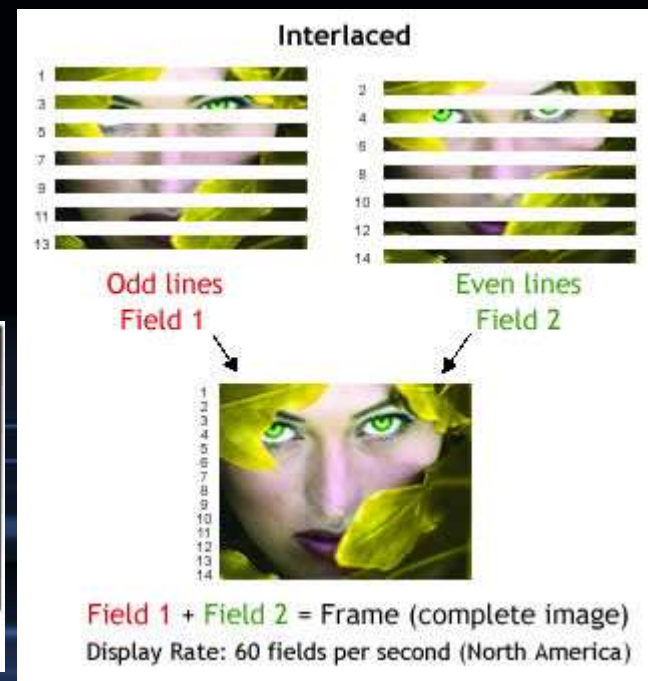
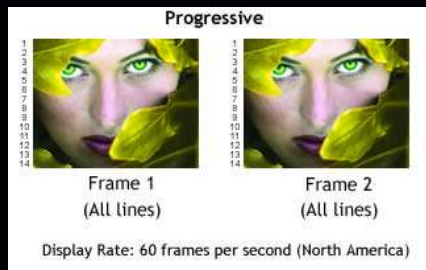
# MPEG-4

- **Varredura Progressiva (p)**

- “Varre” a tela inteira em uma única passada, transmitindo e exibindo todas as linhas da tela a cada atualização

- **Varredura entrelaçada (i)**

- Monta em cada passagem metade das linhas da tela, as linhas pares ou ímpares



# MPEG-4



## ■ MPEG-4 Parte 10

- Também conhecidos como H.264 ou AVC (Advanced Video Coding)
- Um padrão de codec de vídeo digital que tem a característica de alta taxa de compressão
- O padrão define 7 perfis, voltada 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.
  - Extended Profile (XP) é voltado para streaming de vídeo, com alta taxa de compressão e robustez para perda de dados.
  - High Profile (HiP) é o principal perfil para aplicações de armazenamento em disco e broadcast, particularmente para aplicações de HDTV e adotado pelos discos HD-DVD e Blu-ray.



# MPEG-4 BIFS

- **BIFS – Binary Format for Scenes**
  - MPEG-4 é um sistema baseado em objetos.
  - BIFS permite a organização no tempo e espaço de vários tipos de mídia:
    - Uma descrição de cena compõe estes objetos
    - Descreve interatividade com objetos
    - Anima objetos

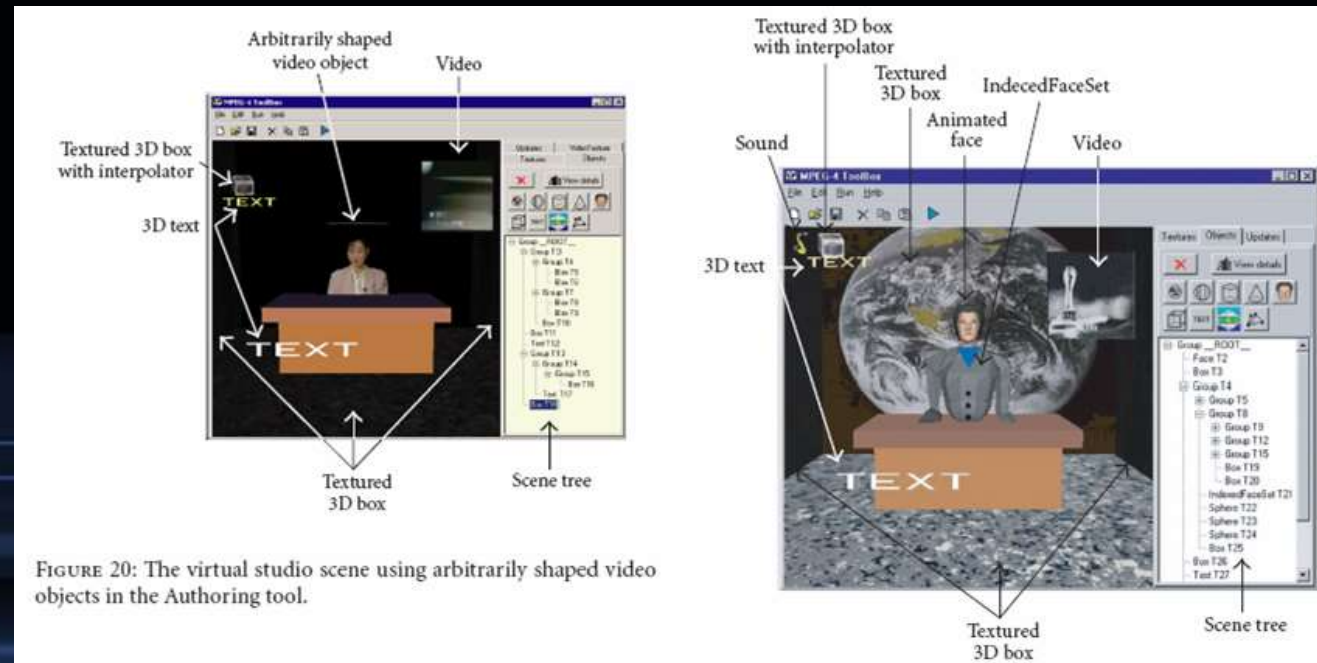


FIGURE 20: The virtual studio scene using arbitrarily shaped video objects in the Authoring tool.

# ITU-T H.261



## ■ Origem

- Necessidade de fornecer serviços de vídeo onipresentes na Rede Digital de Serviços Integrados (ISDN)
- ISDN: Tecnologia de transmissão digital de voz, vídeo e dados e outros serviços sobre a rede pública de telefonia comutada
  - Cada circuito garantindo 64 kbps sem variação de atrasos.
- Um dos padrões da família H.320 para videofonia e teleconferência na taxa de 64 Kbps a 2 Mbps

## ■ Padrão ITU-T

- Ratificado em novembro de 1988
- Influenciou o H.263, MPEG 1 a 4, etc.

# ITU-T H.261

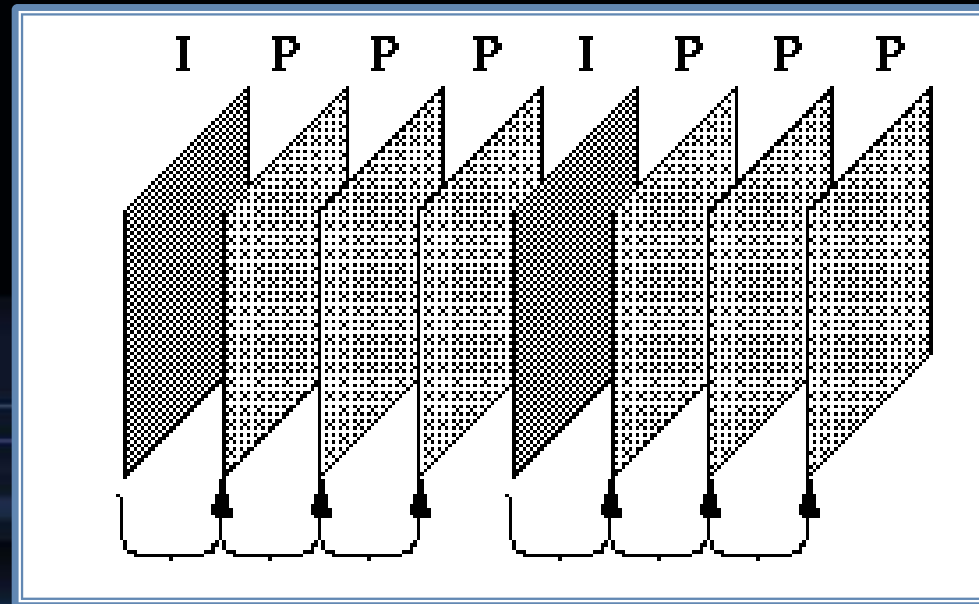


- **H.261 foi projetado para usar toda a capacidade do canal ISDN**
  - $p \times 64$  Kbps ( $p=1$  a  $30$ )
  - $p = 1$  ou  $2$  é apropriado para comunicação visual face-a-face e baixo movimento (videofonia)
  - $p > 5$  melhor qualidade (videoconferência)
  - Máxima taxa de bits disponível é 1,92 Mbps ( $p=30$ )
    - suficiente para obter imagens de qualidade VHS

- **Principais características**
  - Para aplicações de videofonia e teleconferência
  - Algoritmo de compressão de vídeo opera em tempo-real com atraso mínimo
  - Fornece uma resolução cerca de oito vezes mais baixa que a qualidade TV PAL/SECAM
  - É para aplicações usualmente sem movimentos intensos
    - algoritmo usa uma limitada estratégia de busca e estimação de movimento para obter taxas de compressão mais altas

# ITU-T H.261

- **Algoritmo de compressão**
  - Dois tipos de quadros: intraquadros (quadros I) e interquadros (quadros P)
    - I fornece um ponto de acesso e usa basicamente JPEG
    - P usa estimativa e compensação do movimento do quadro anterior



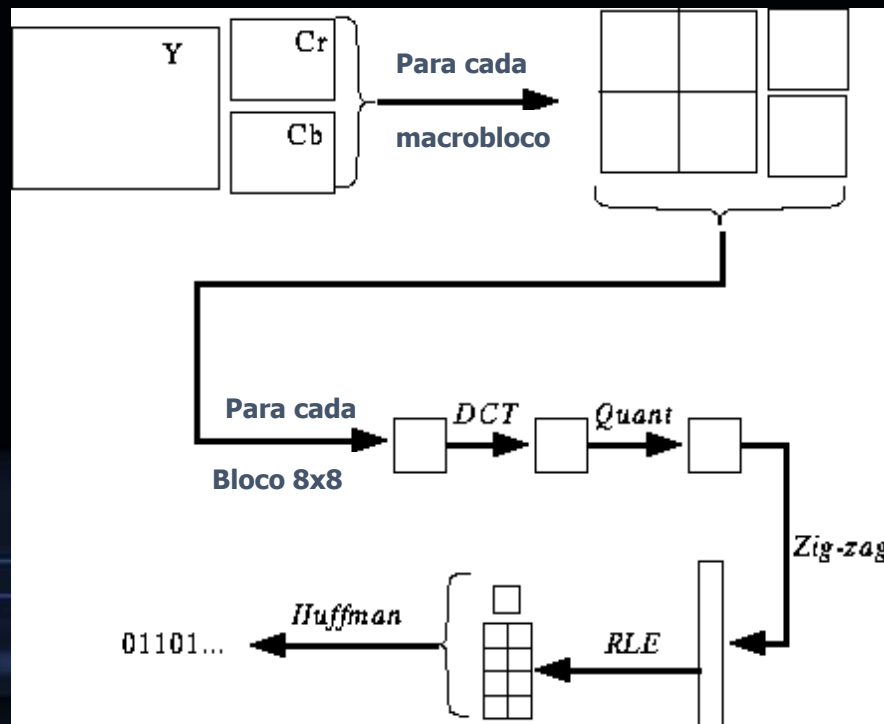


# ITU-T H.261

- **Algoritmo de compressão**

- Quadros I

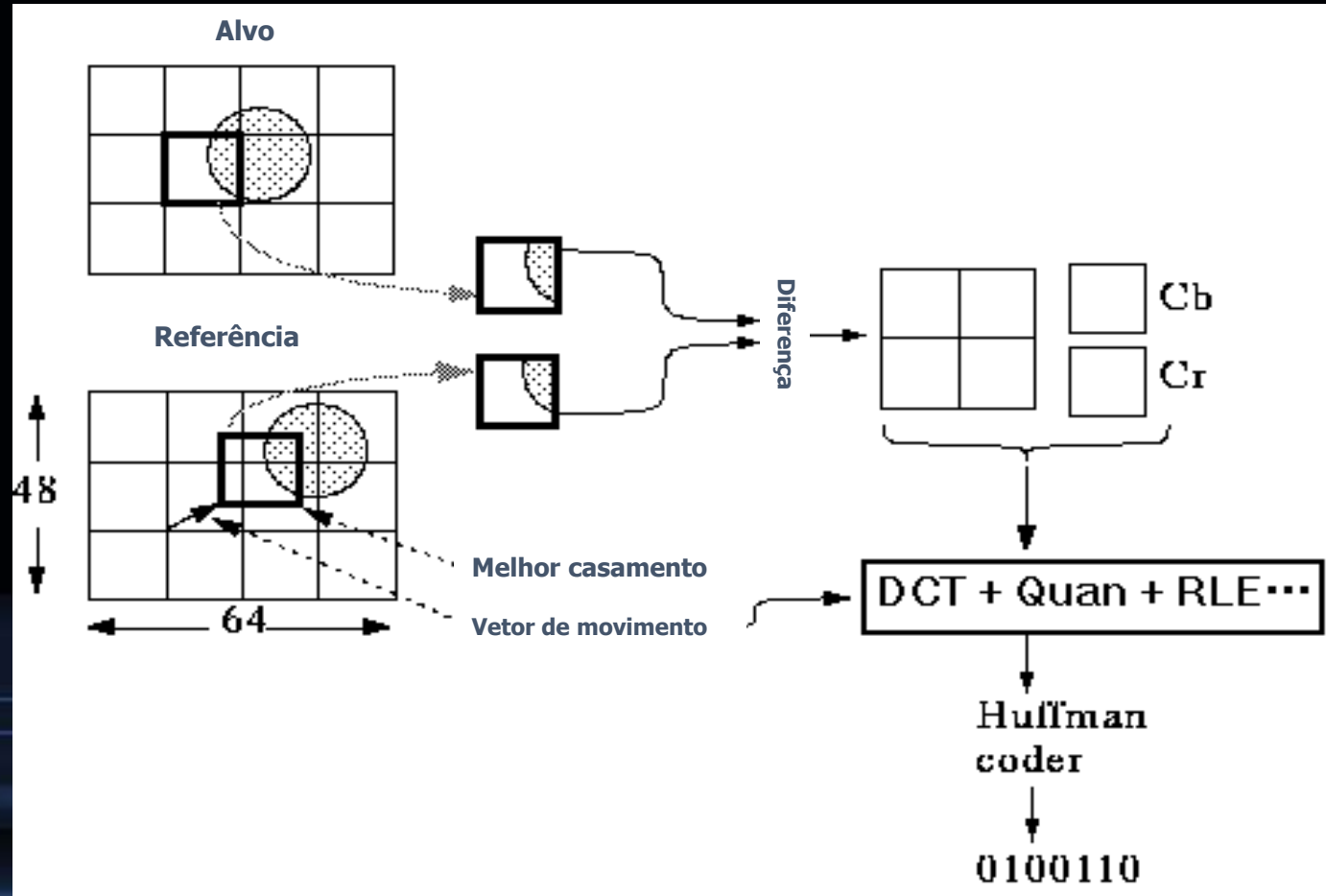
- Usa o conceito de macrobloco: área de 16x16 píxeis no Y e 8x8 no Cb e Cr (4:2:0)



# ITU-T H.261

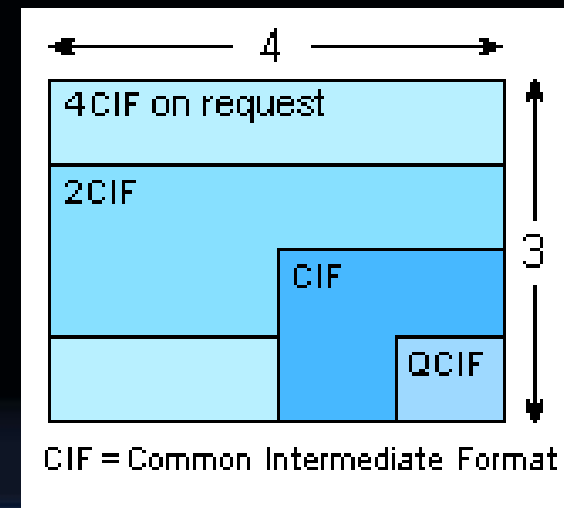
## ▣ Quadros P

- Usa o conceito de macrobloco: área de 16x16 pixels no Y e 8x8 no Cb e Cr (4:2:0)



# ITU-T H.261

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



# ITU-T H.26\*



- **H.262**
  - MPEG-2 pela ISO/IEC
- **H.263 (1995)**
  - Padrão de vídeo a baixa taxa de bits para aplicações de teleconferência que opera a taxas abaixo de 64 Kbps
- **H.263 v1 H.263 v2 (H.263+, 1997),**
- **H.263 v3 (H.263++, 2000), H.26L (2002)**
- **H.264/AVC (2003)**
- **H.265/HEVC (MPEG-H Part 2)**
  - 25% a 50% mais compressão que AVC na mesma qualidade