



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

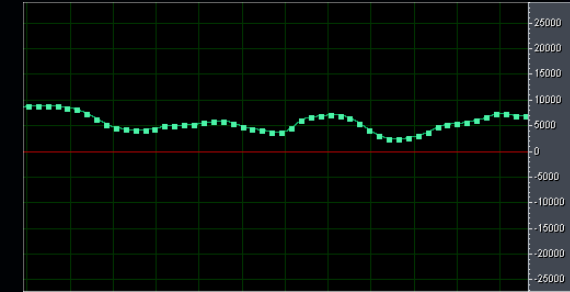


- **Fatores explorados pelas técnicas de compressão**

- Redundância de dados
- Propriedades da percepção humana

- **Redundância de Dados**

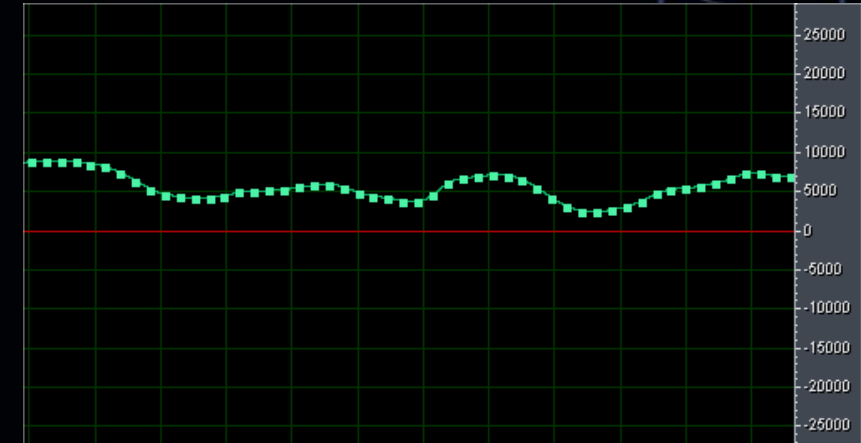
- Representação de dados multimídia
 - áudio digital é uma série de valores amostrados
 - imagem é uma matriz de valores amostrados (píxeis)
 - vídeo é uma sequência de imagens apresentadas numa certa taxa
- Amostras vizinhas não são inteiramente diferentes
 - valores vizinhos são de algum modo relacionados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

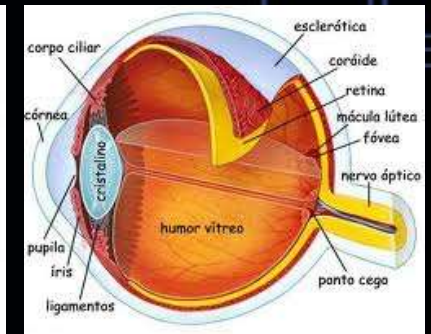
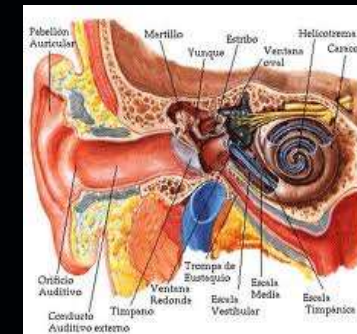
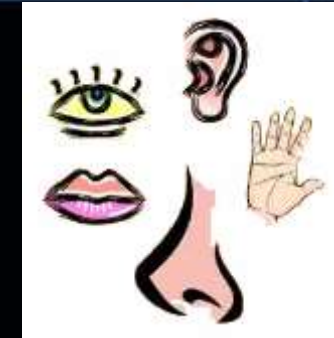


- **Humanos não são perfeitos**

- 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**

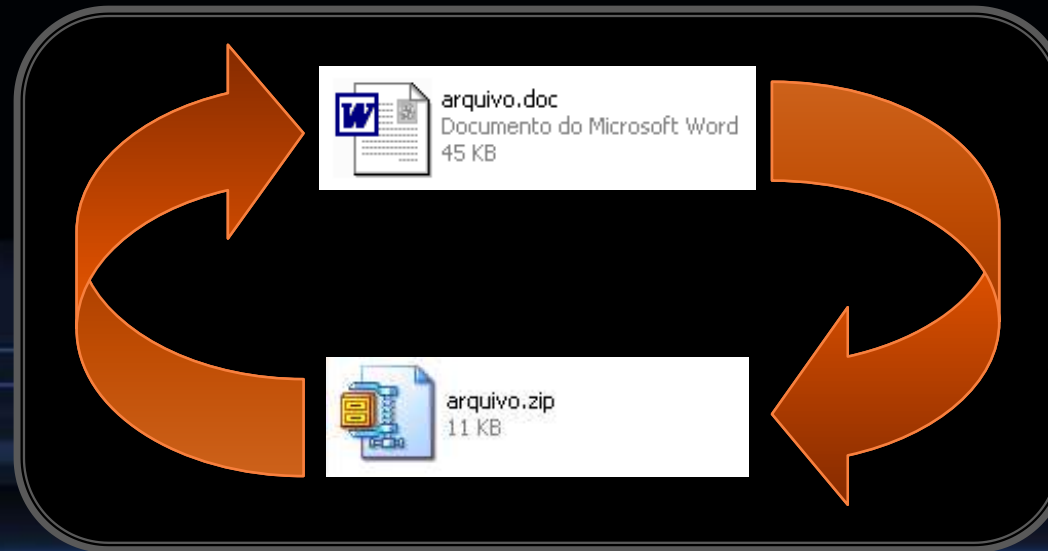
Cap 3. Compressão de Dados Multimídia



- **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



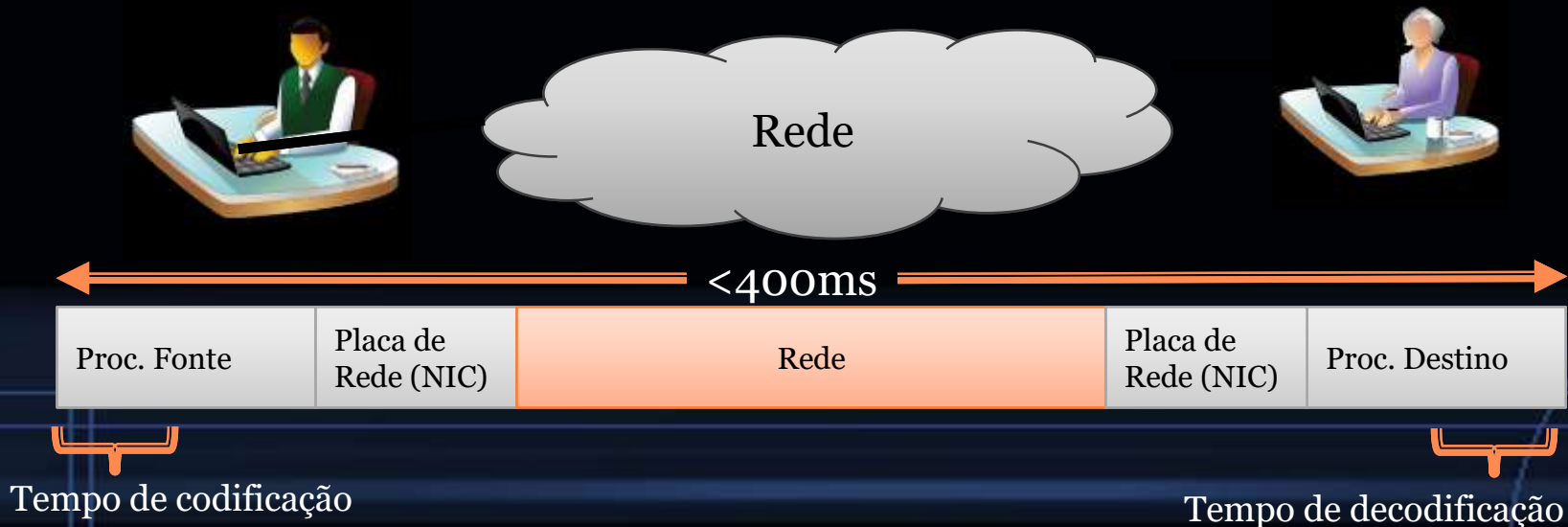
- **Qualidade da média reconstituída**
 - Relação Sinal-Ruído de Pico (**PSNR** – Peak Signal-to-Noise Ratio), definida (em dB)

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

- 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 ($\text{Orig} = \text{Dec}$)
 - $\text{PSNR}(\text{Orig}, \text{Dec}) = \infty$

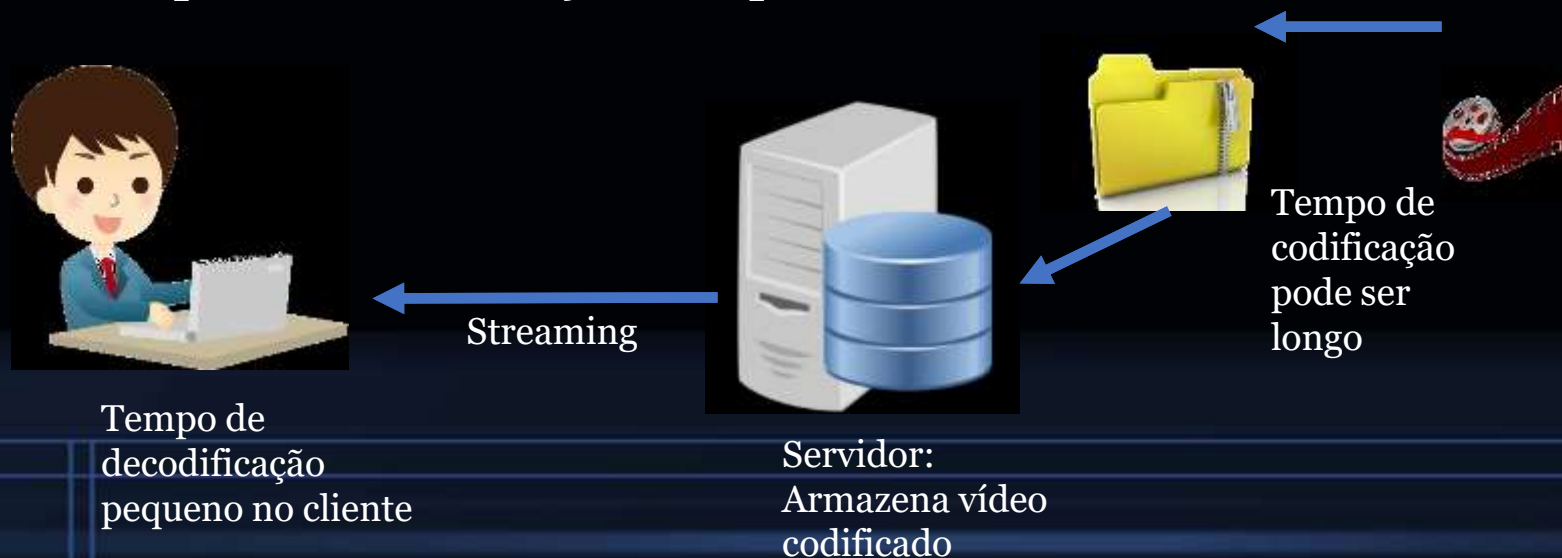
Parâmetros de Desempenho

- **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



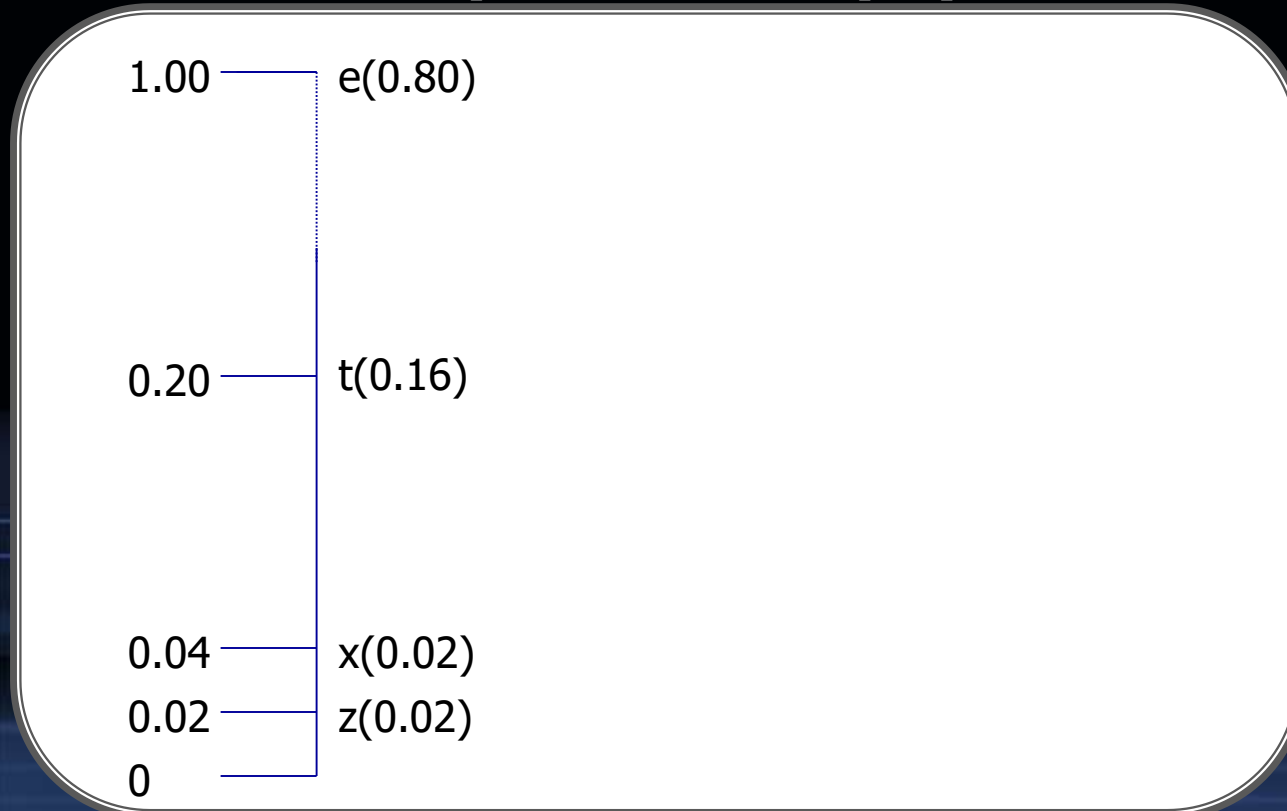
- **Codificação de Huffman (Codificação Estatística)**
 - Exemplo de funcionamento:
 - Suponha um arquivo de 1000 símbolos: e, t, x, z.
 - Frequência de ocorrência: $e = 0.8$, $t = 0.16$, $x = 0.02$ e $z = 0.02$
 - Original: eeteeeezeeteeteeteeetexeeeeeeeteteee.....
 - Necessitamos de 2 bits para representar cada um dos 4 símbolos
 - $e = 00$, $t = 01$, $x = 10$ e $z = 11$
 - Original: 0000000100000000110001000001000001000000000001001000000000000000000000010001000000.....
 - Tamanho: $2 * 1000 = 2000$ bits
 - Usando codificação de Huffman podemos usar quantidades diferentes para representar estes símbolos (de acordo com a frequência de ocorrência)
 - $e = 0$, $t = 10$, $x = 110$ e $z = 111$
 - Codificado: 000100000111010001000t0000010011000000000010010000.....
 - Tamanho: $1000 * (1 * 0.8 + 2 * 0.16 + 3 * 0.02 + 3 * 0.02) = 1240$
 - apesar de x e z terem sido representados com um maior número de bits

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

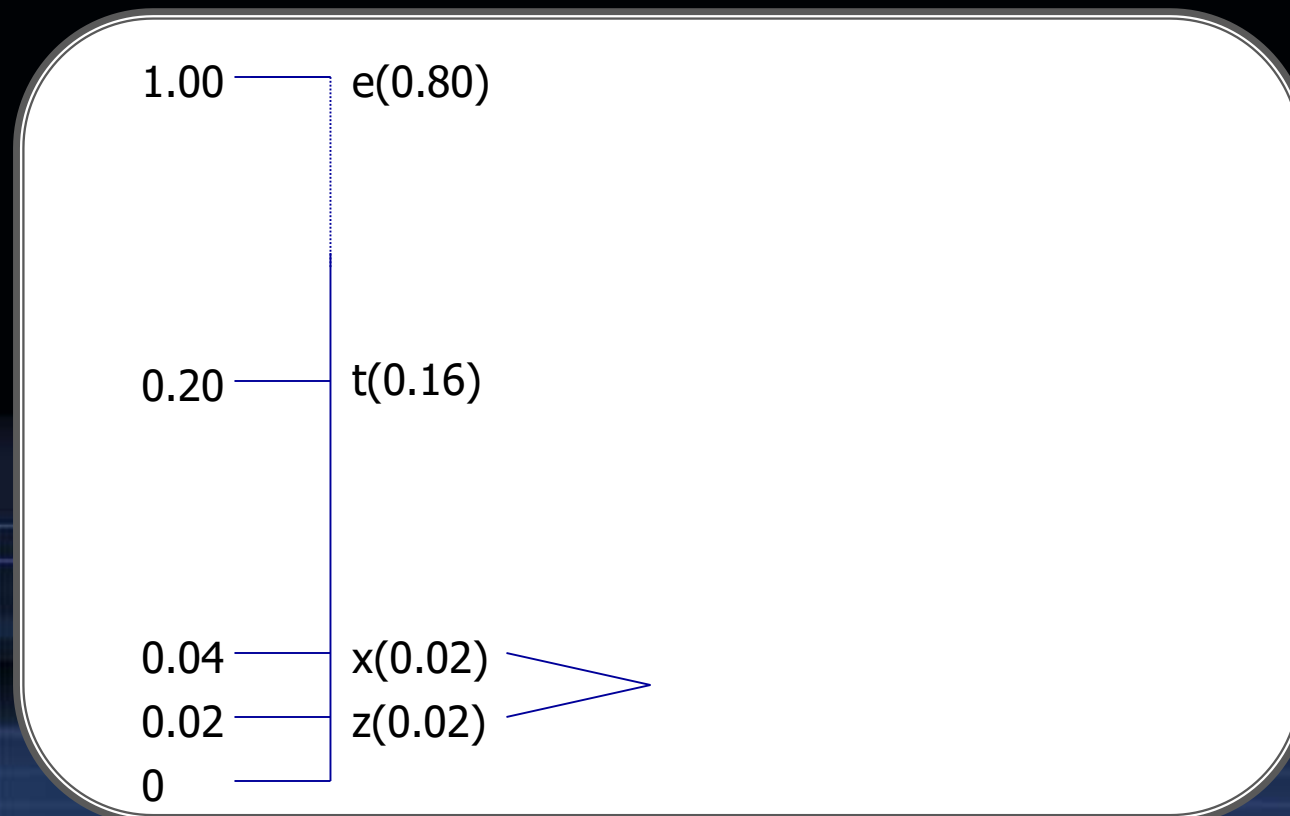


Técnicas de compressão sem perdas

- **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

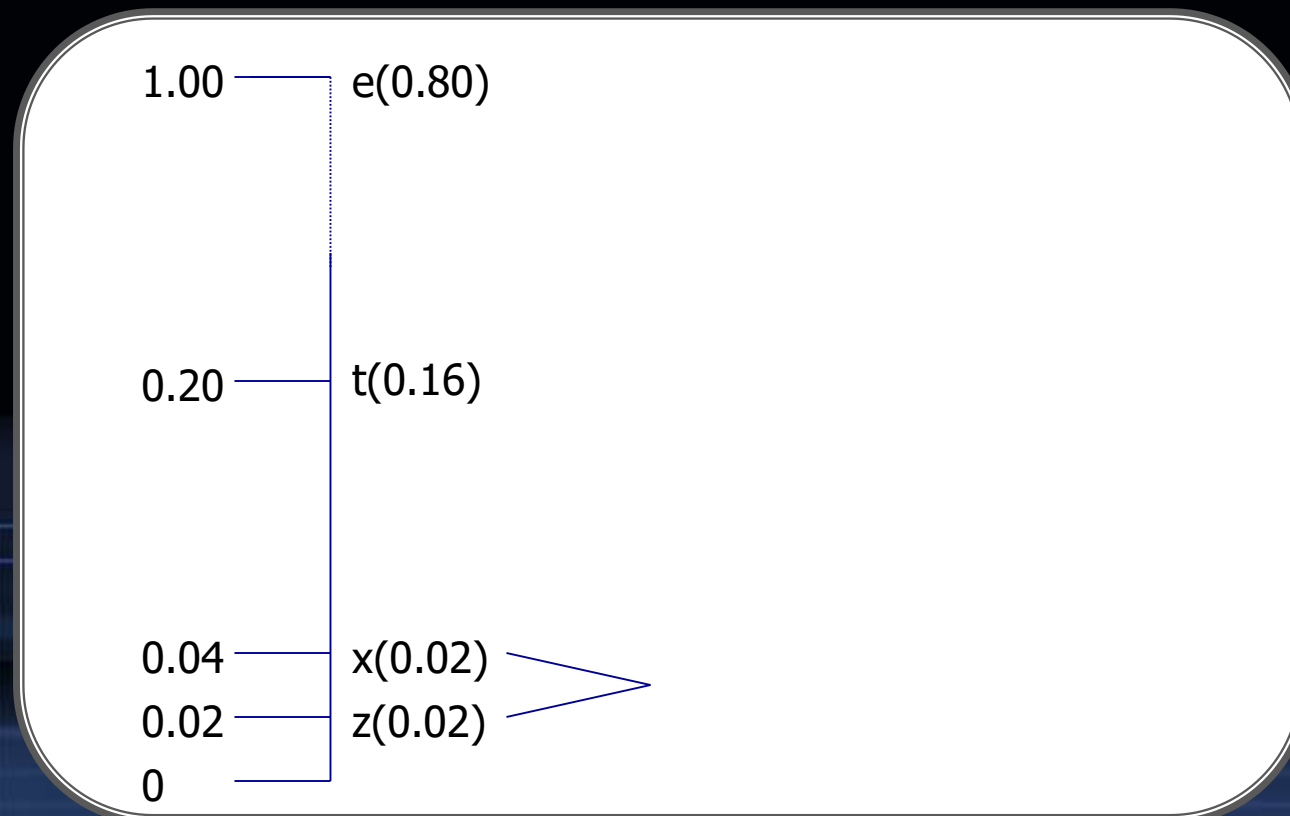


Técnicas de compressão sem perdas

- **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

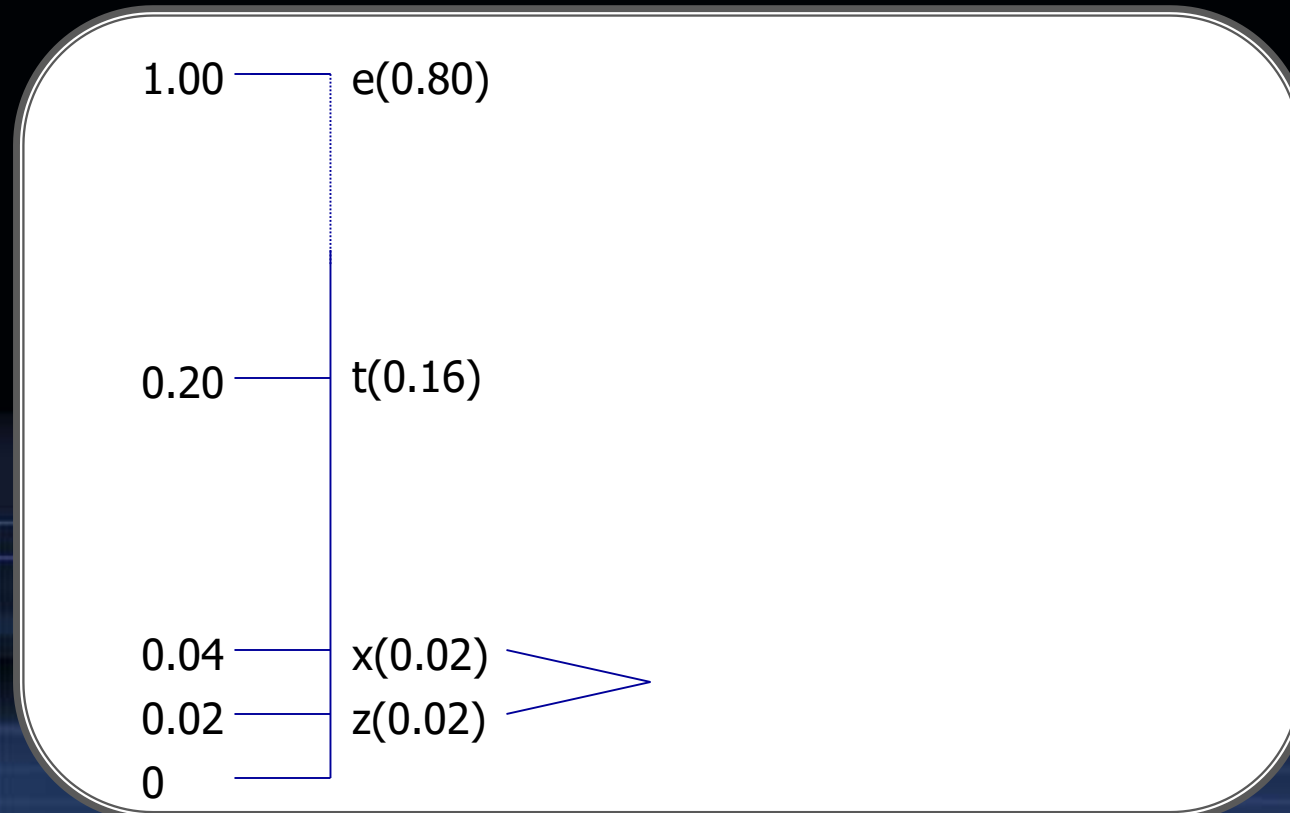


Técnicas de compressão sem perdas

- **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

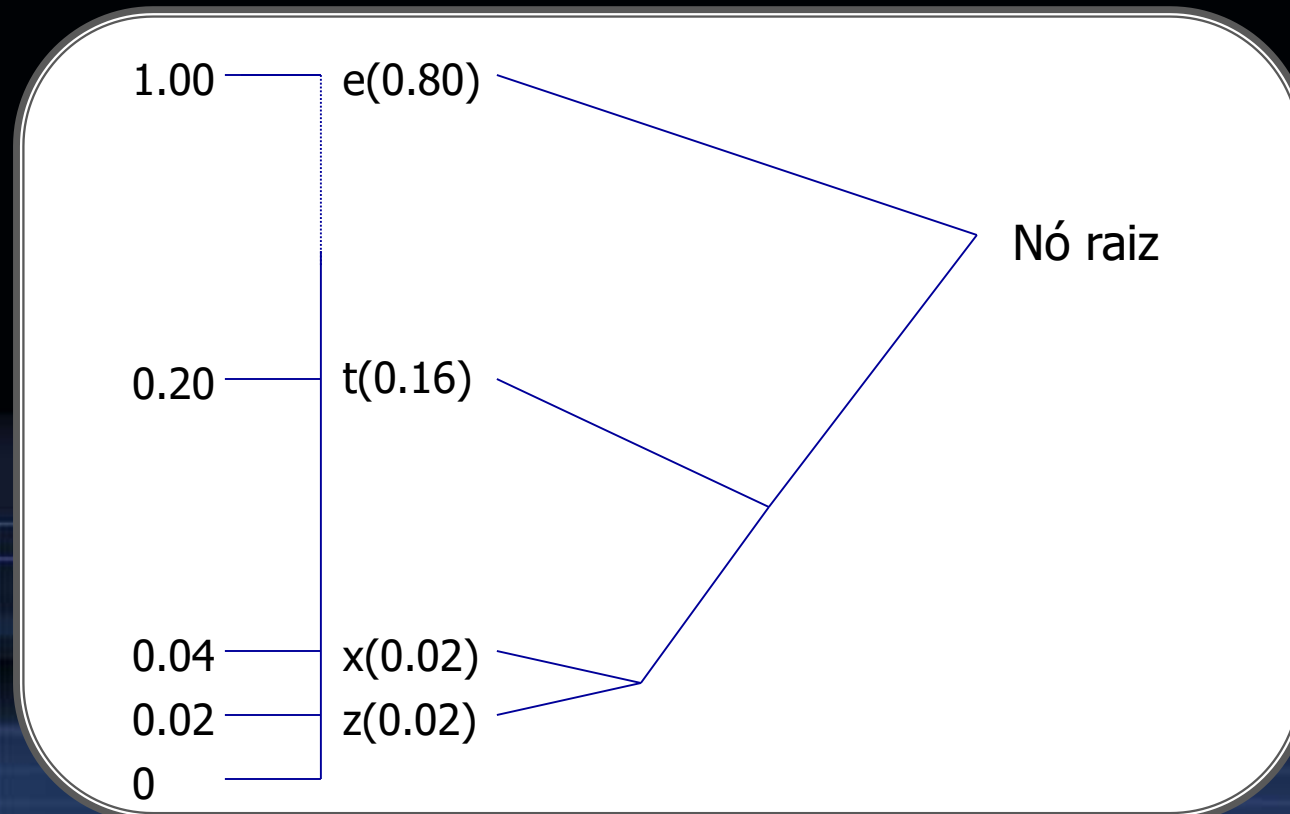


Técnicas de compressão sem perdas

- **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

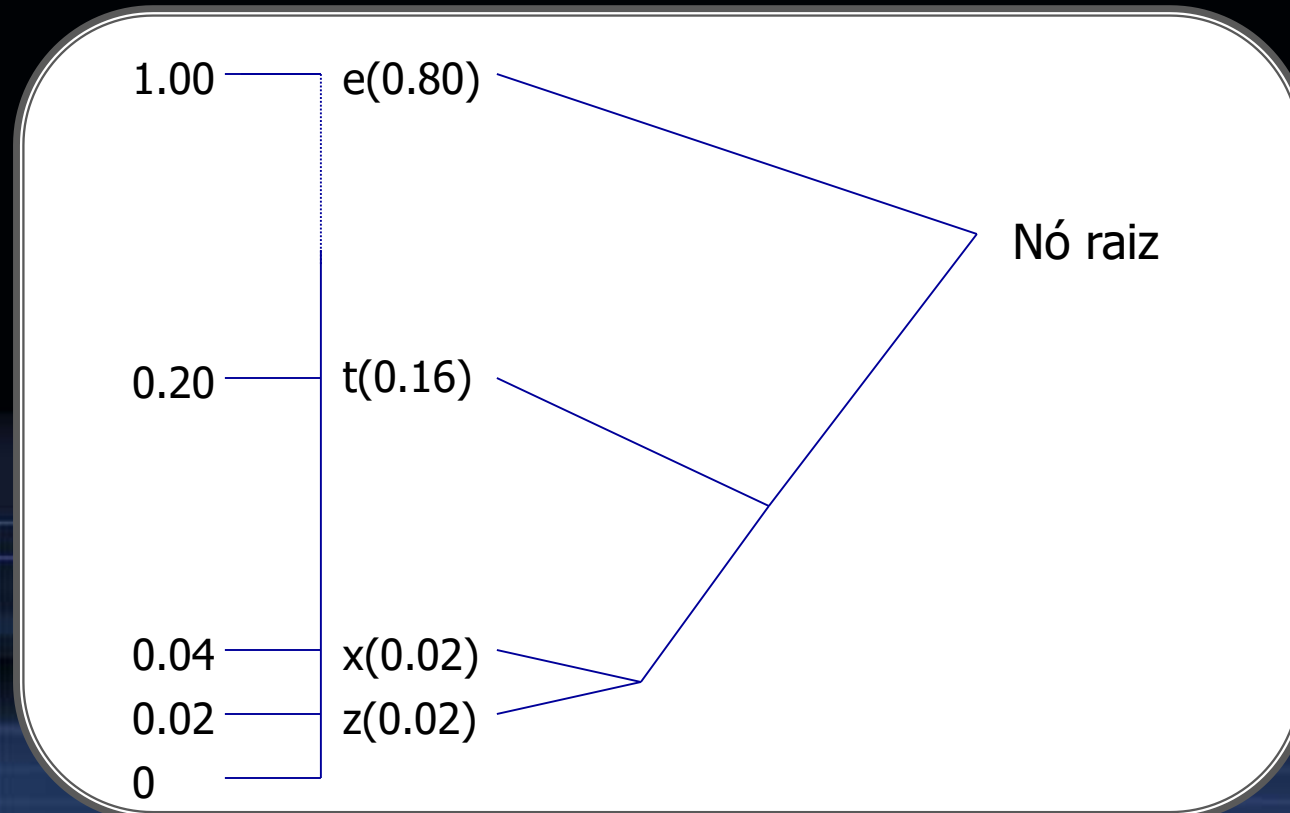


Técnicas de compressão sem perdas

- **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

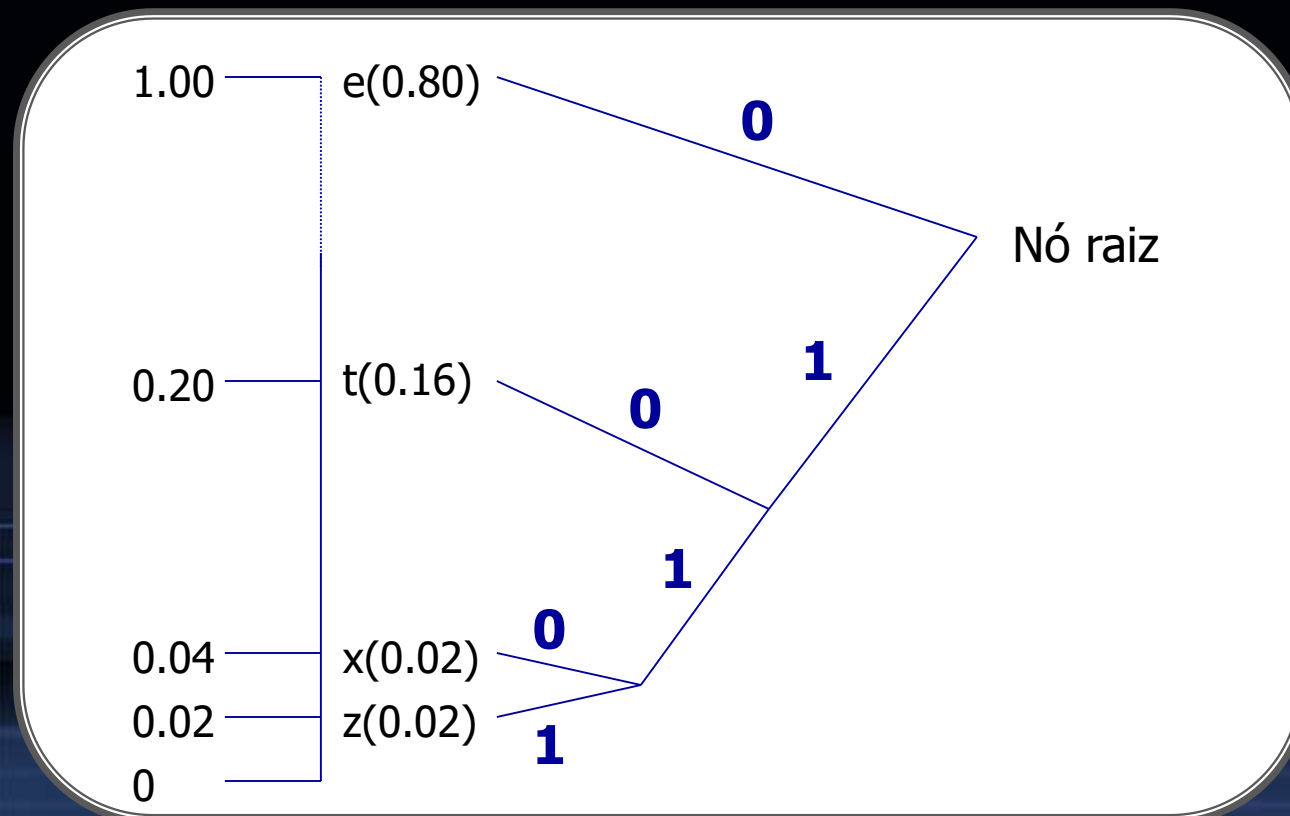


Técnicas de compressão sem perdas

- **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

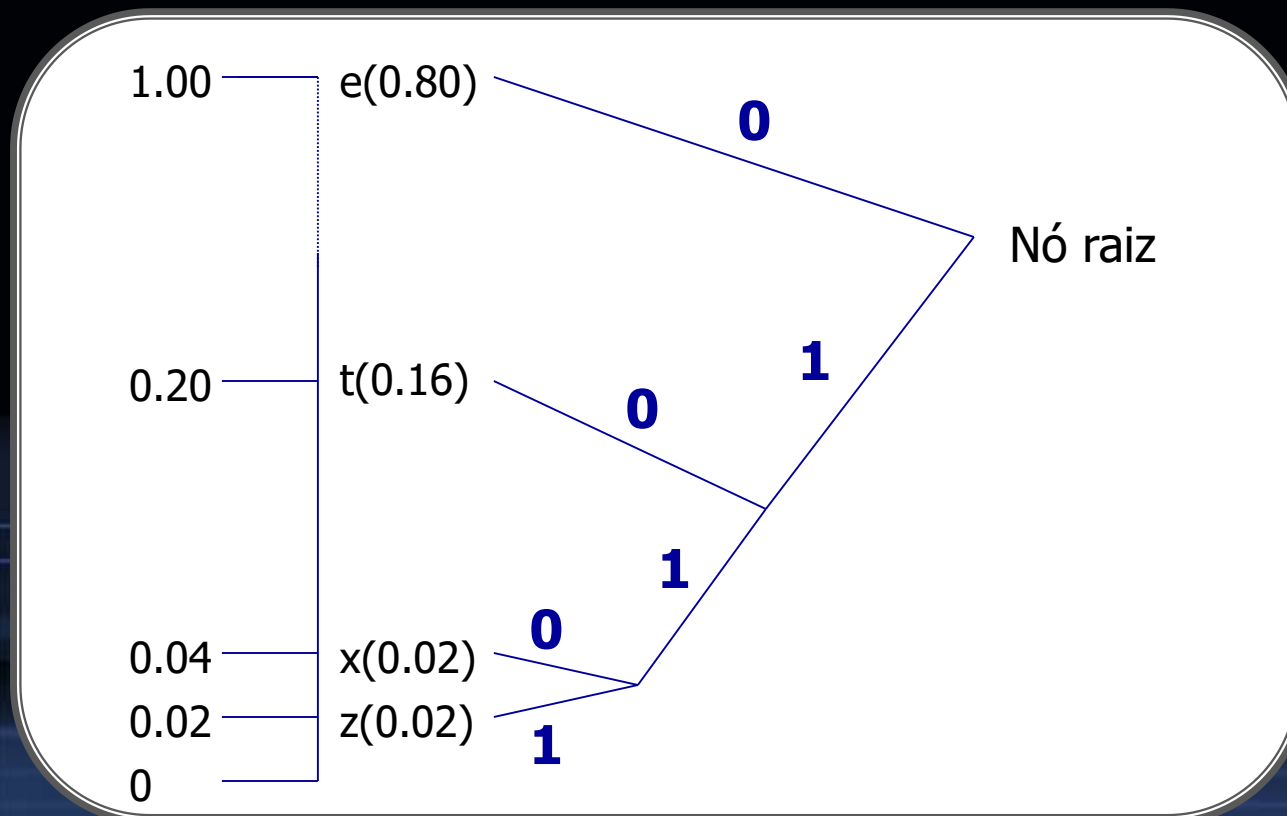


Técnicas de compressão sem perdas

■ 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

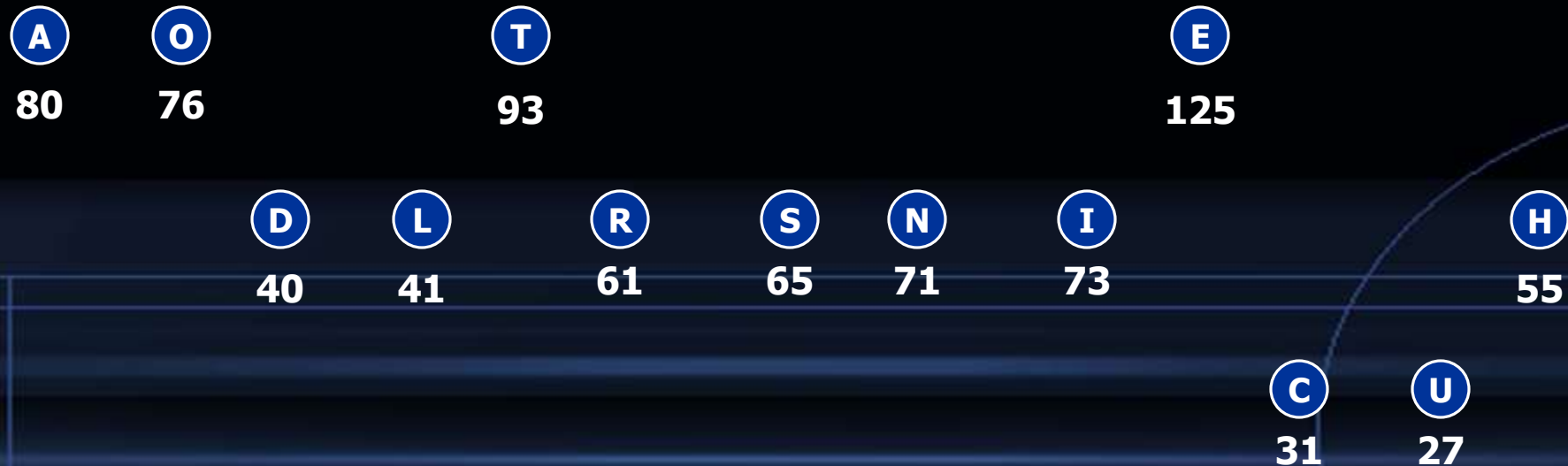


- **Outro exemplo de codificação de Huffman**
 - Frequências dos caracteres
 - Gere a tabela de Huffman para o arquivo.

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

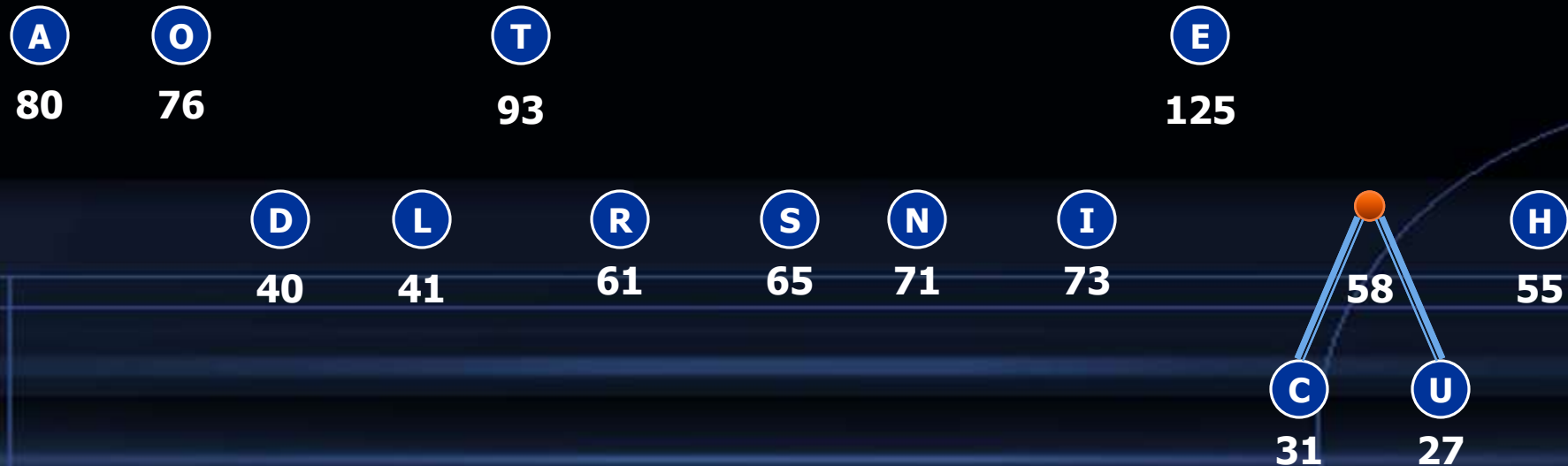
Técnicas de compressão sem perdas

- **Outro exemplo de codificação de Huffman**



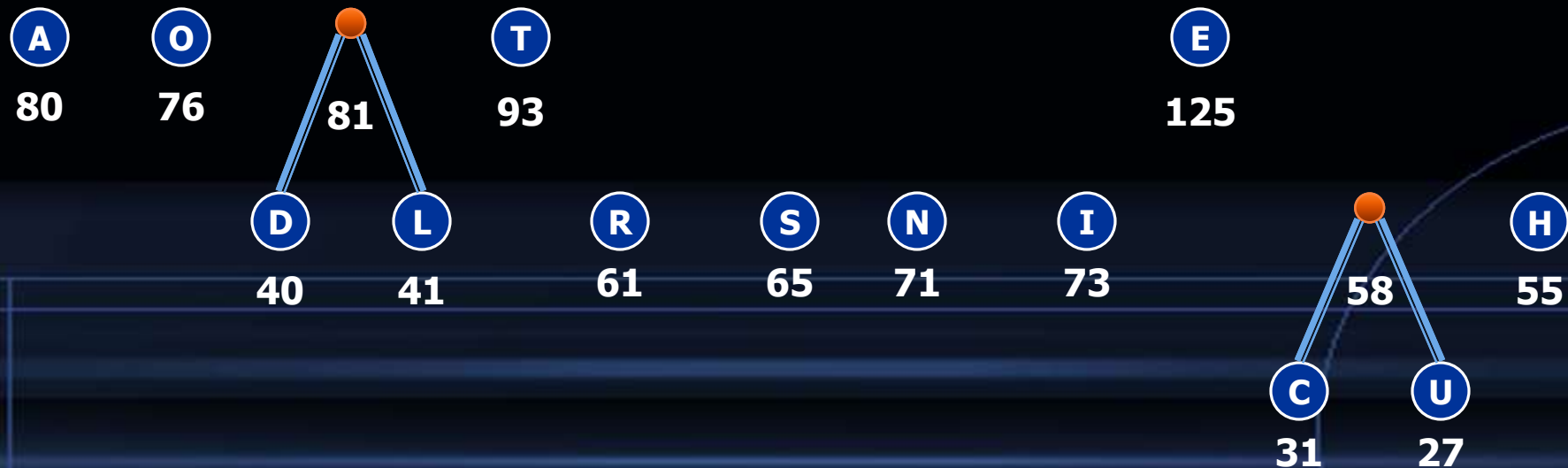
Técnicas de compressão sem perdas

- **Outro exemplo de codificação de Huffman**



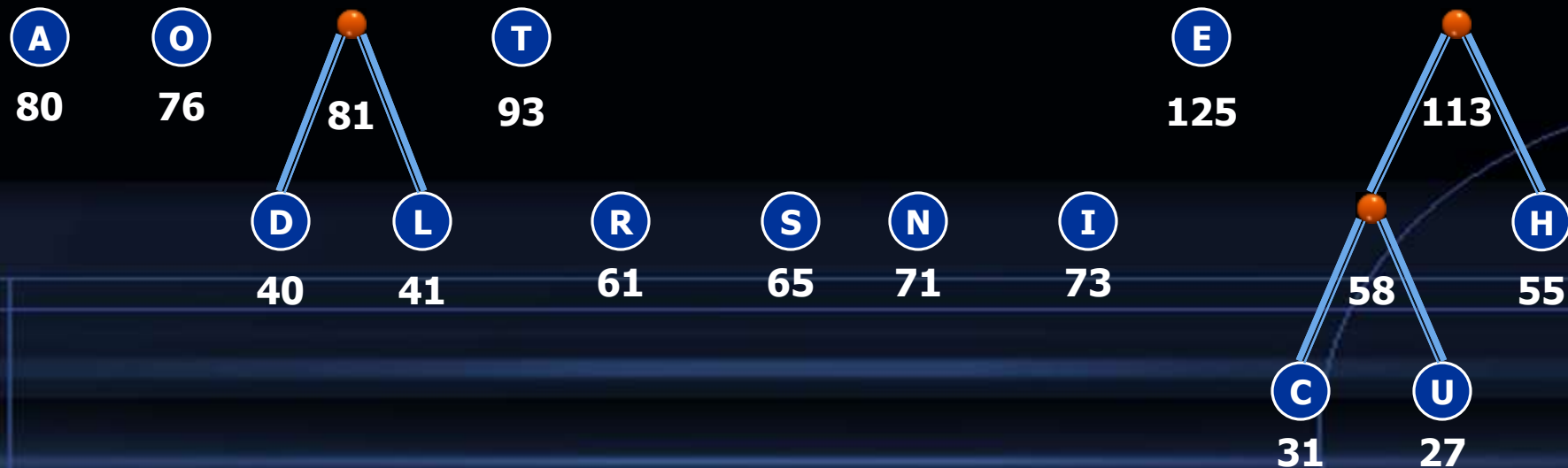
Técnicas de compressão sem perdas

- **Outro exemplo de codificação de Huffman**



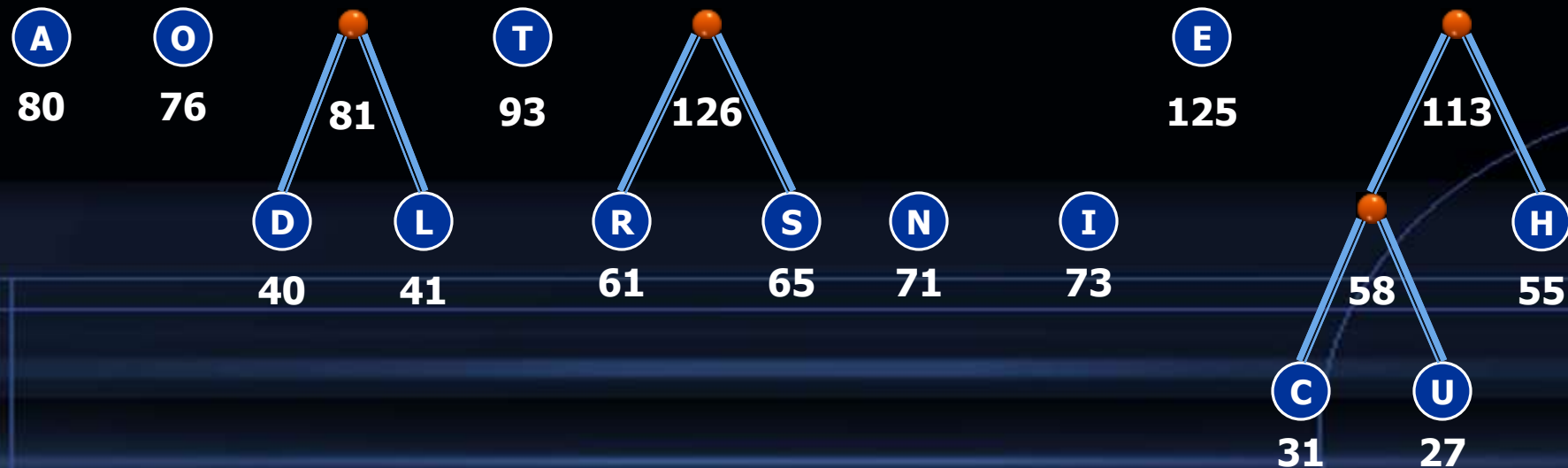
Técnicas de compressão sem perdas

- **Outro exemplo de codificação de Huffman**



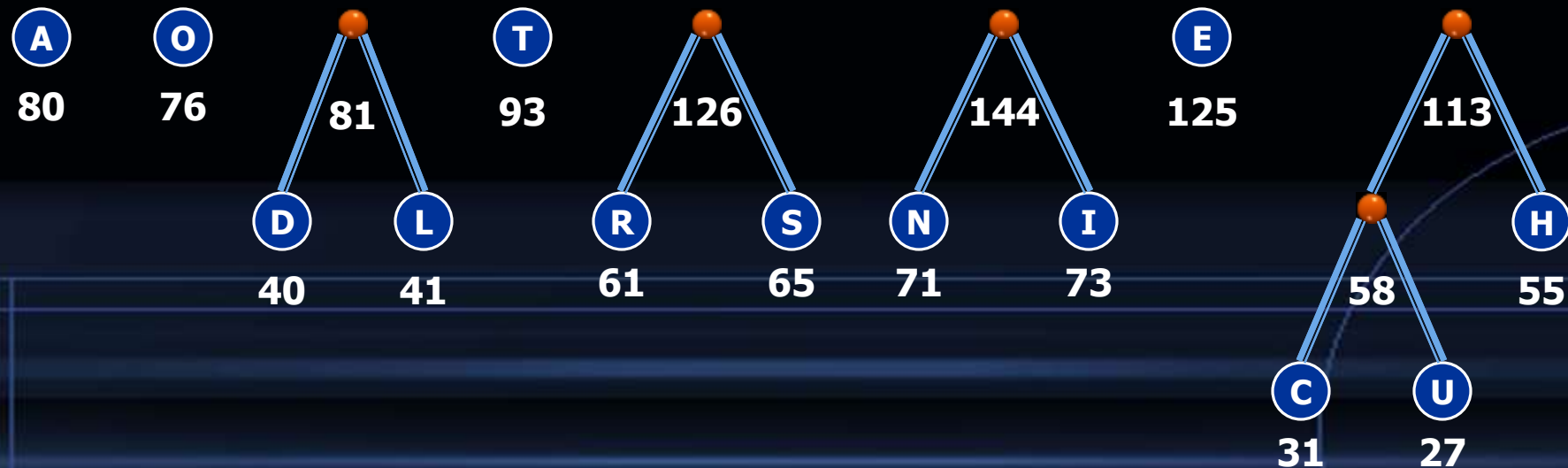
Técnicas de compressão sem perdas

- **Outro exemplo de codificação de Huffman**



Técnicas de compressão sem perdas

- **Outro exemplo de codificação de Huffman**



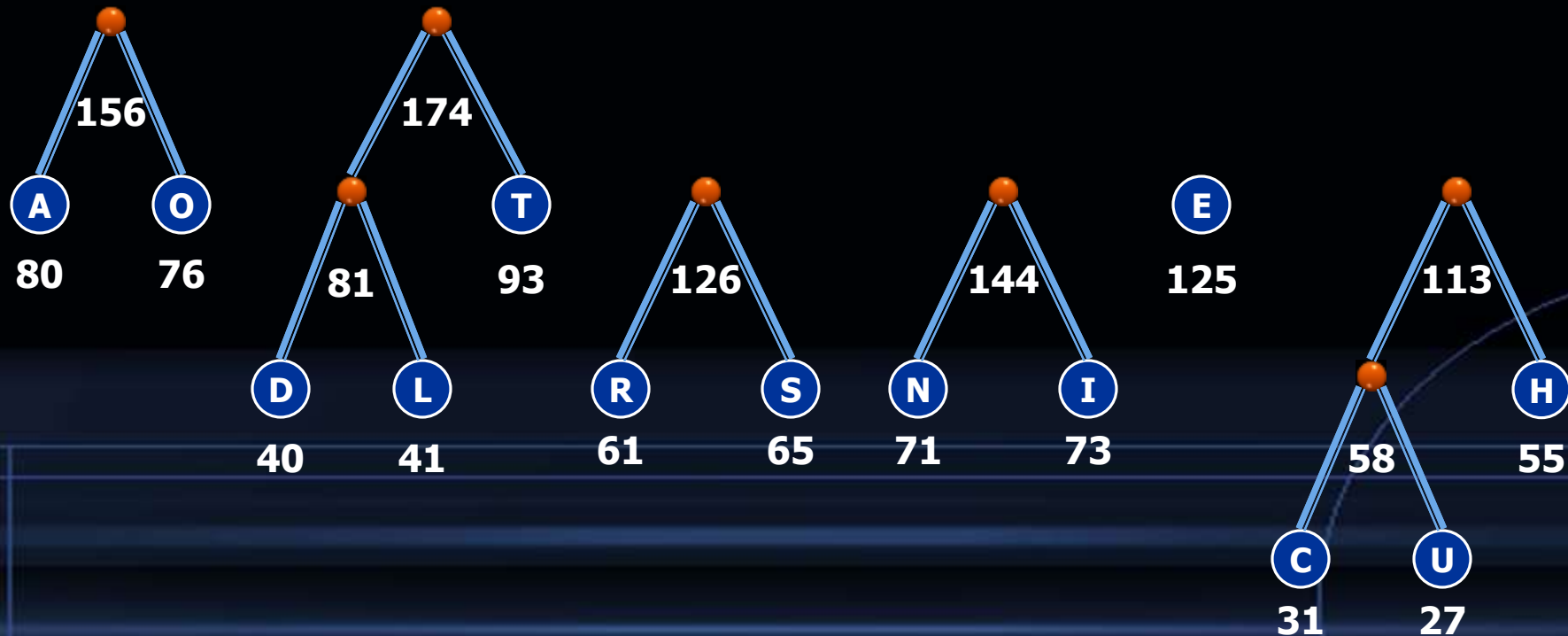
Técnicas de compressão sem perdas

- **Outro exemplo de codificação de Huffman**



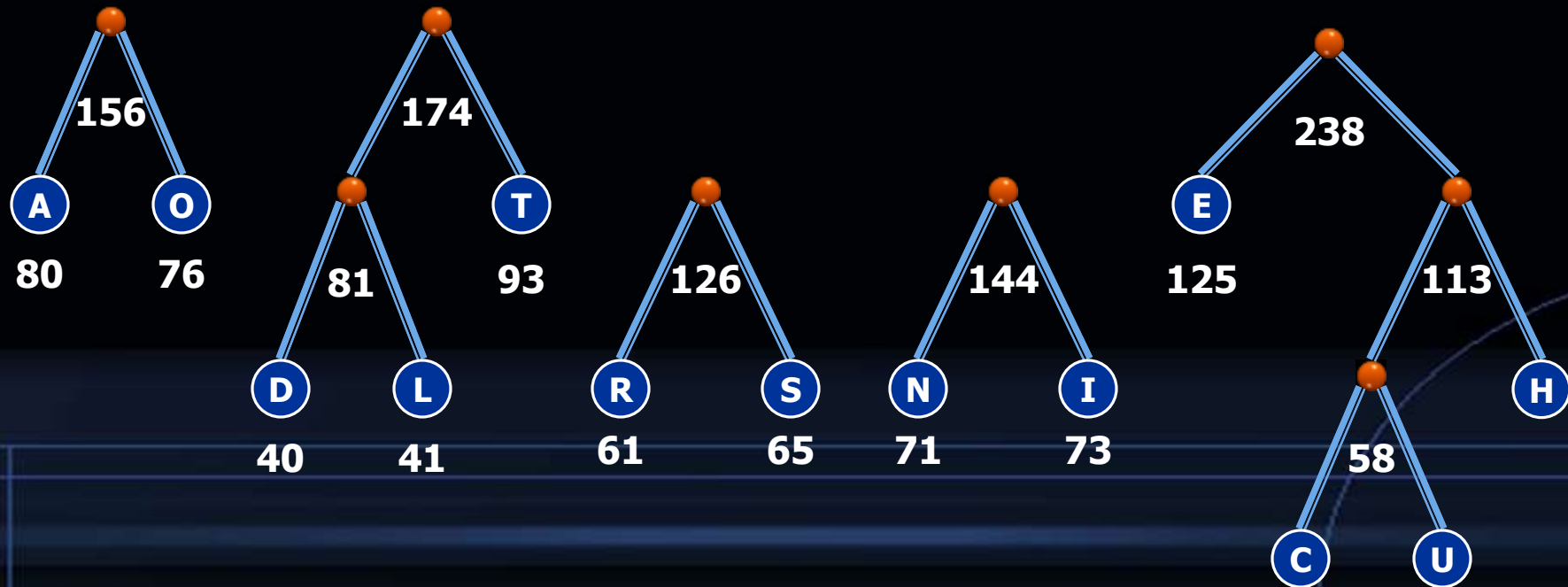
Técnicas de compressão sem perdas

- **Outro exemplo de codificação de Huffman**



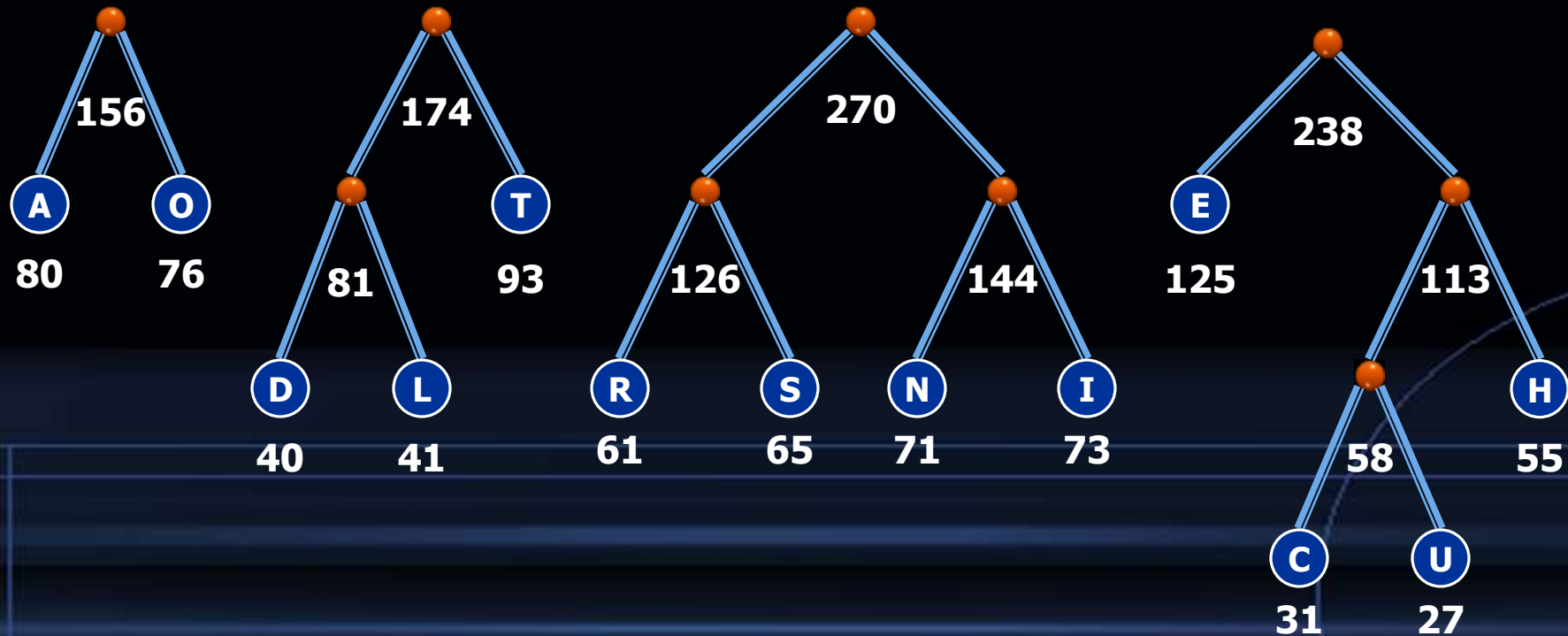
Técnicas de compressão sem perdas

- **Outro exemplo de codificação de Huffman**



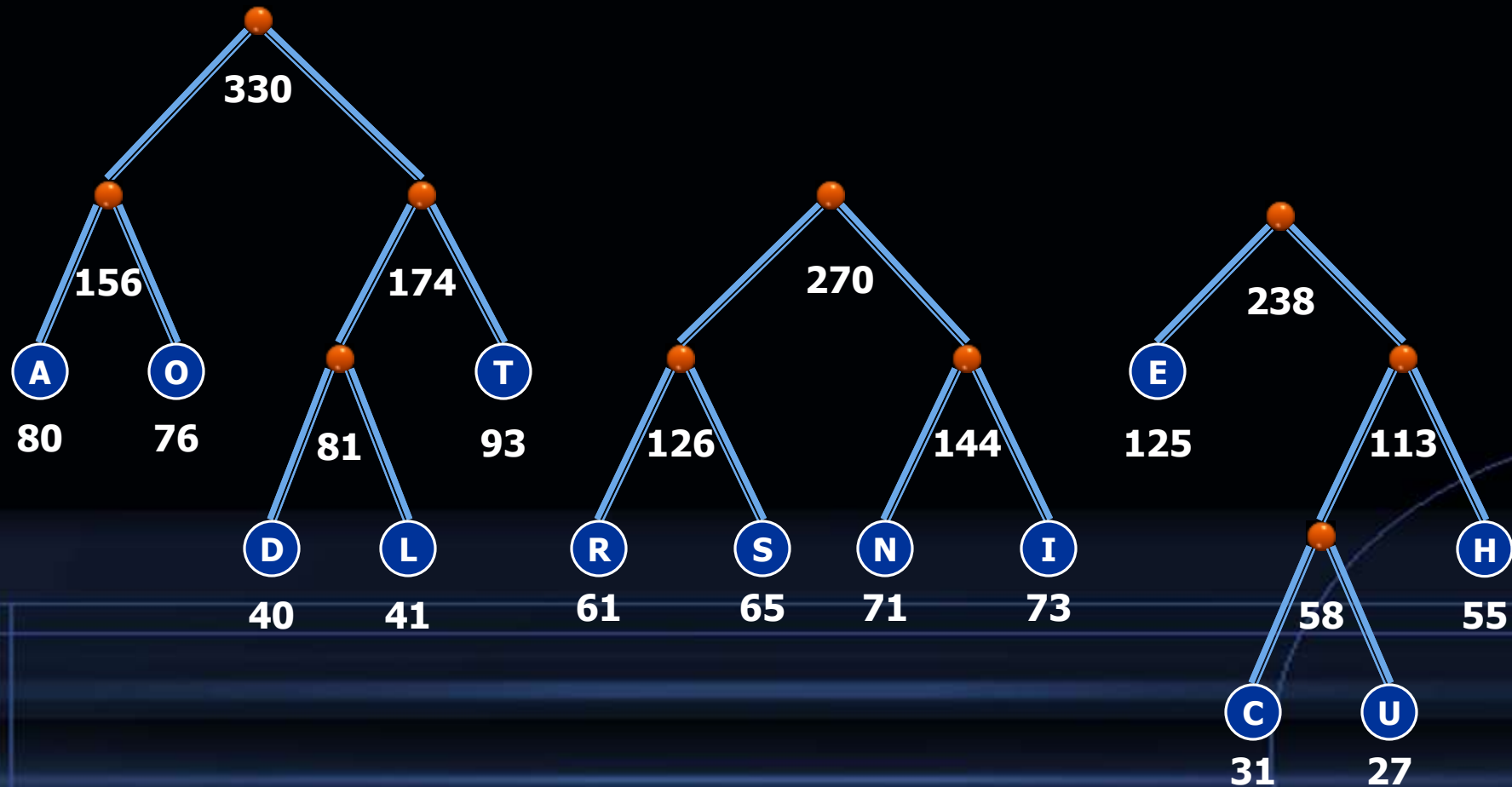
Técnicas de compressão sem perdas

- **Outro exemplo de codificação de Huffman**



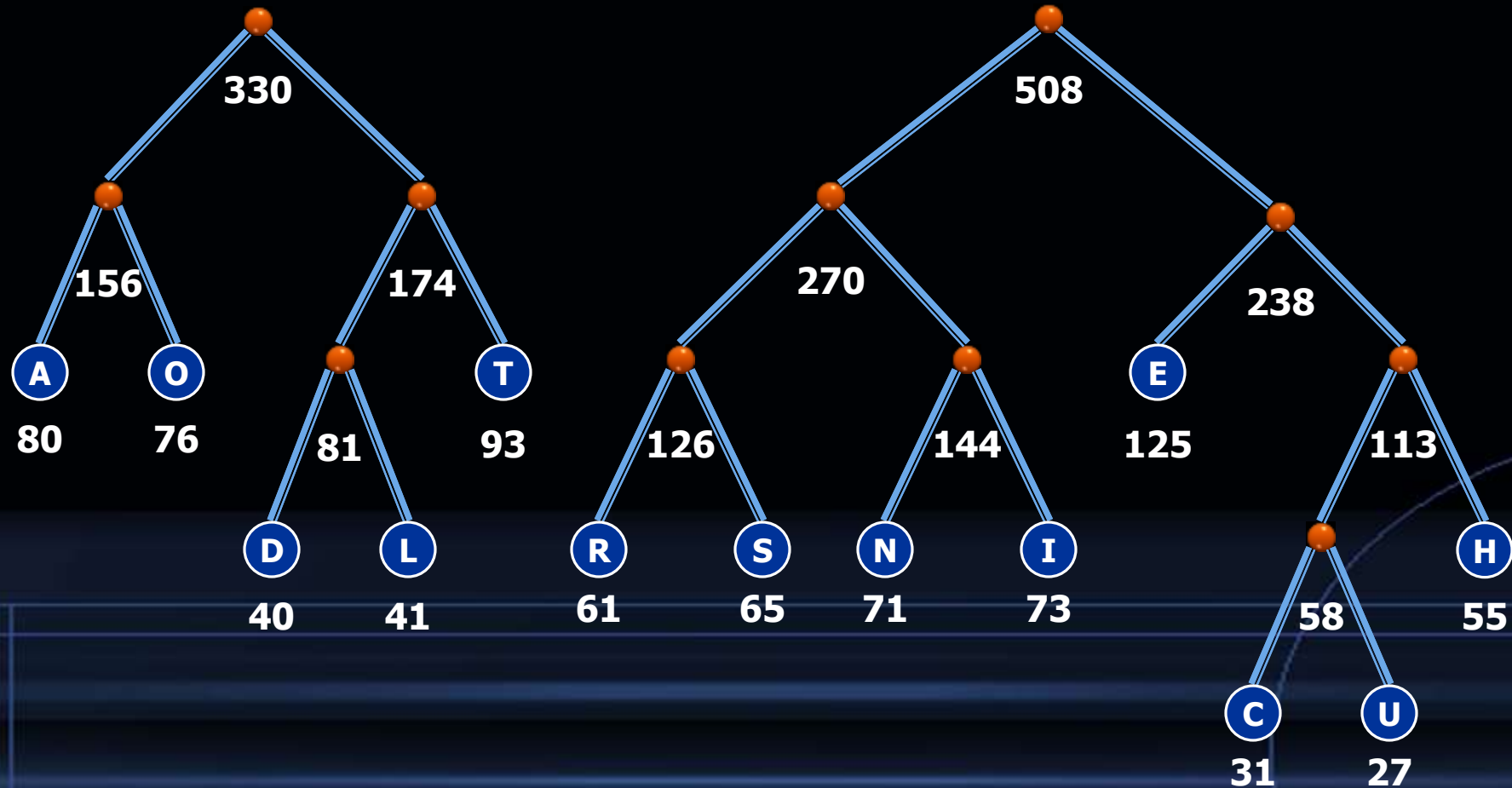
Técnicas de compressão sem perdas

- **Outro exemplo de codificação de Huffman**



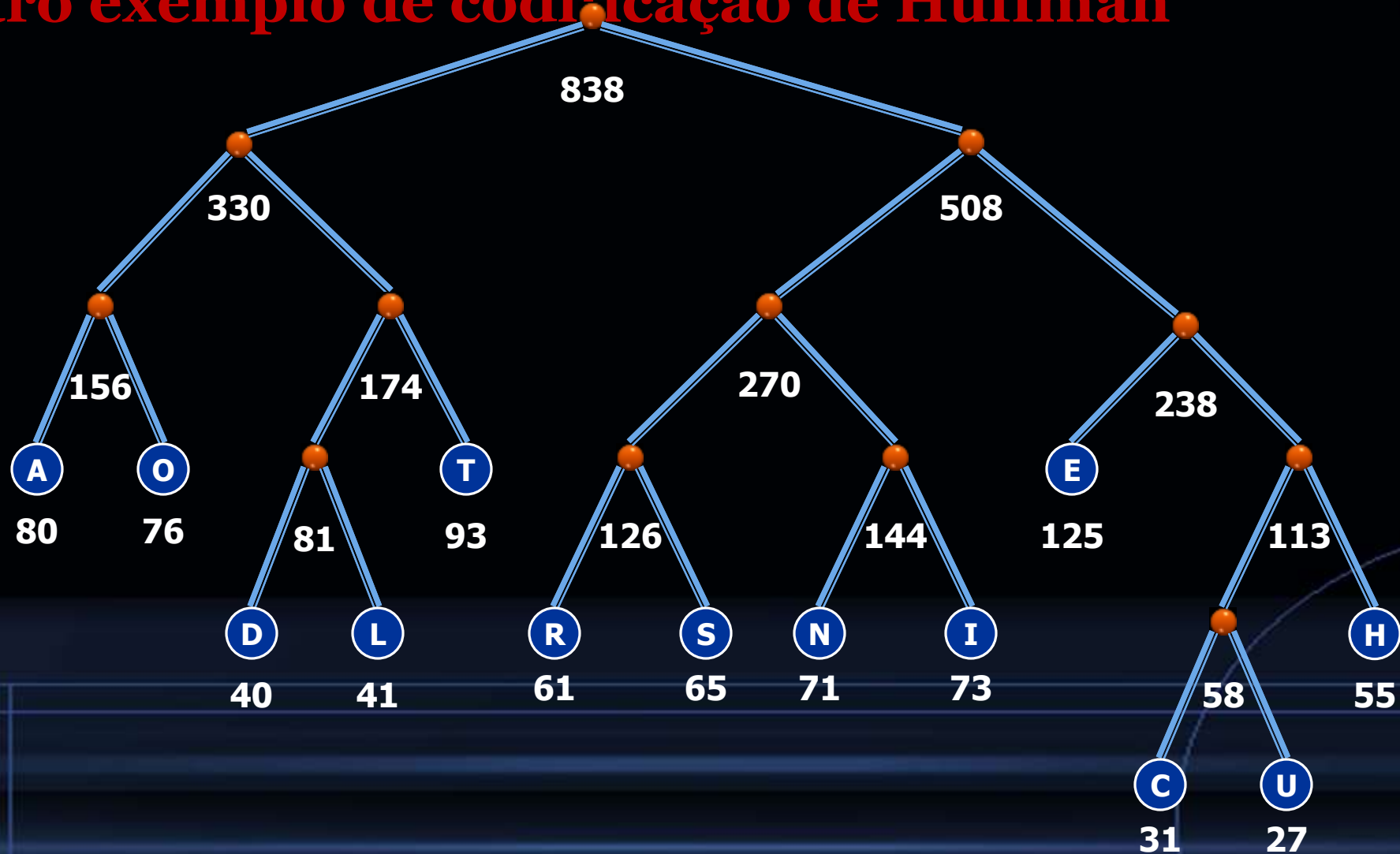
Técnicas de compressão sem perdas

- **Outro exemplo de codificação de Huffman**



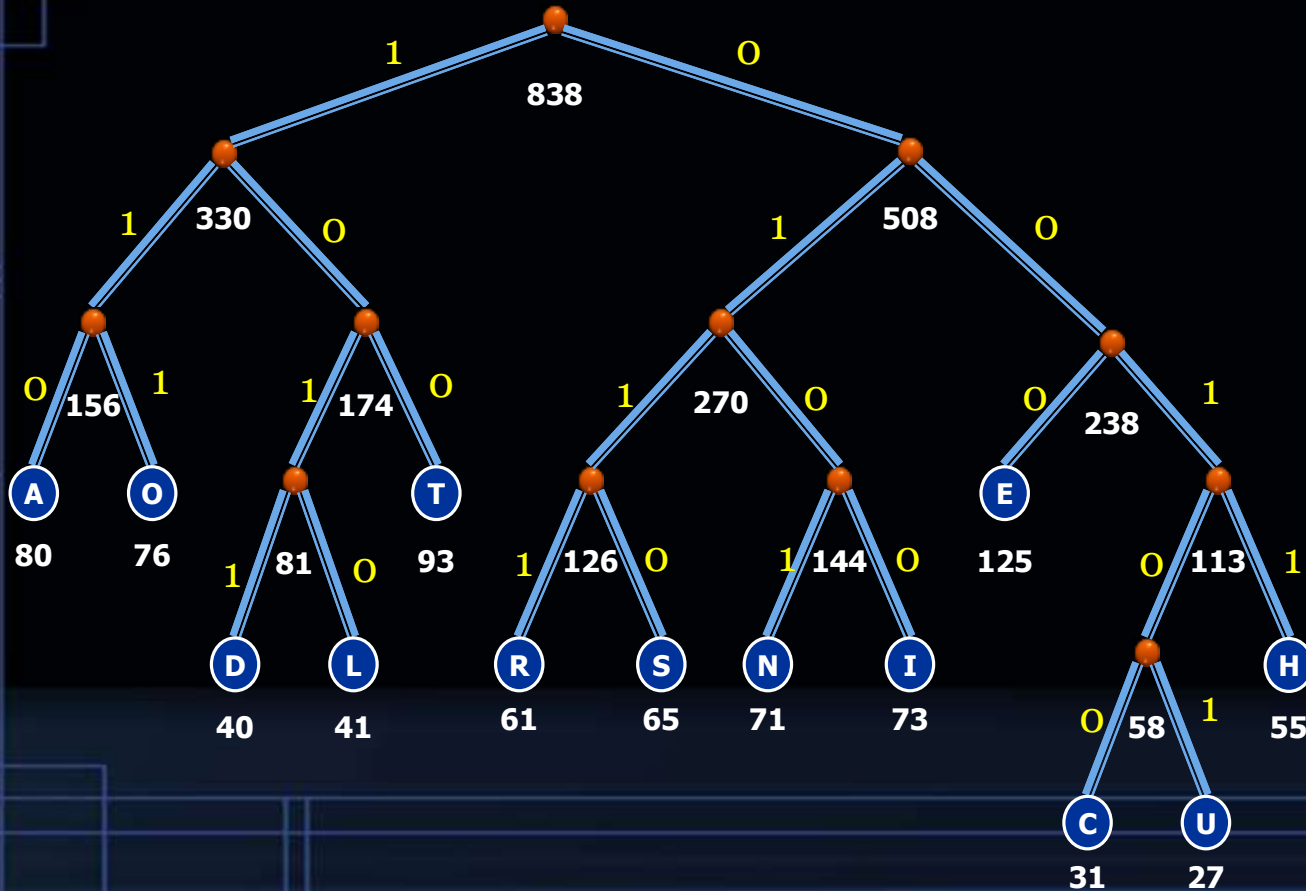
Técnicas de compressão sem perdas

- **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



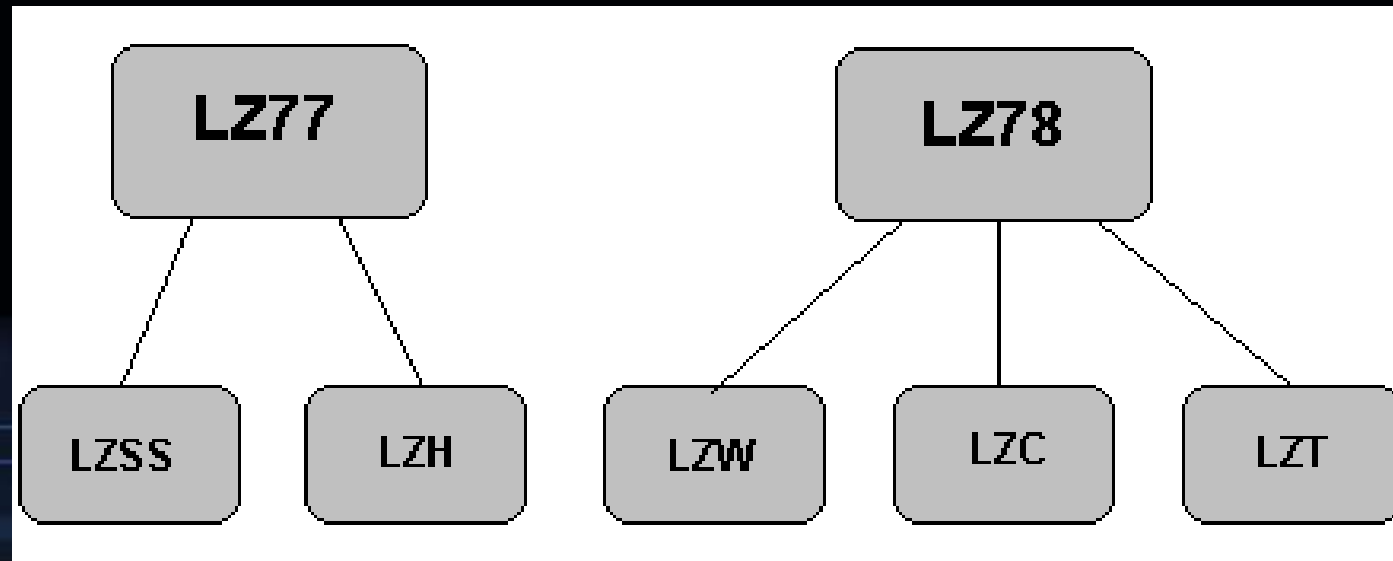
- **Codificação Predictiva**

- ▣ 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 LZW é usado pelo utilitário “compress” do sistema operativo UNIX.
 - GIF (Graphics Interchange Format)
 - Muito similar ao “compress” do UNIX, também usa o algoritmo LZW.
 - Protocolo V.42bis (compressão de dados em Modem)
 - Usa uma variante do LZW (LZT).
 - 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

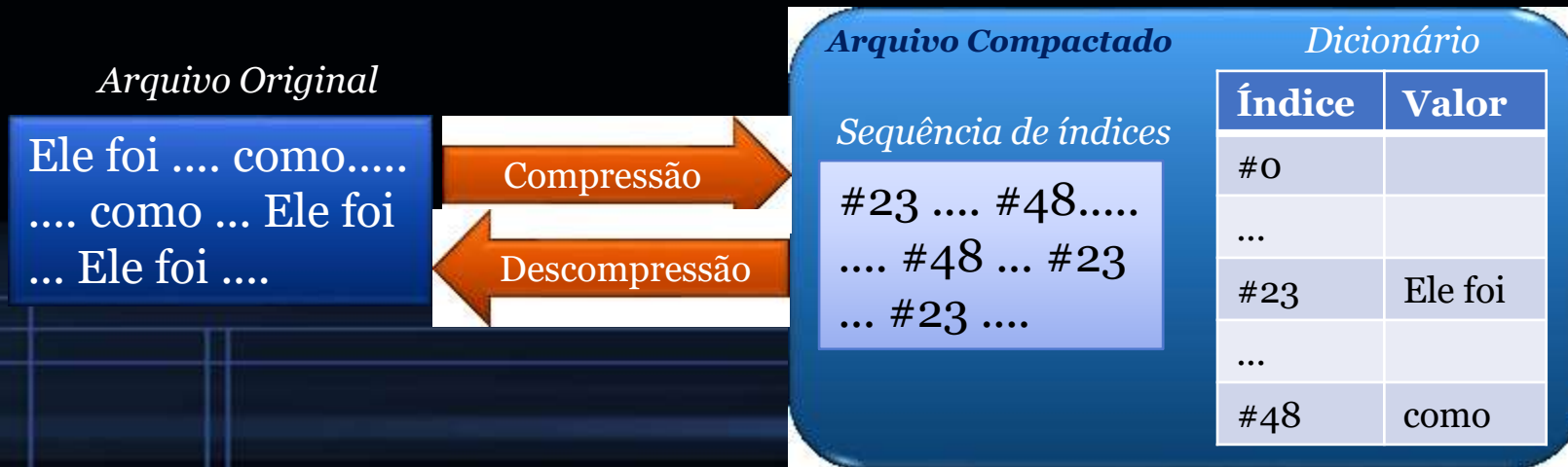


- **Codificações derivadas do Lempel-Ziv (LZ)**
 - Explora a redundância de dados -> a repetição de padrões de símbolos no arquivo
 - Baseada na construção de um dicionário de símbolos (grupos de um ou mais símbolos) a partir do fluxo de entrada
 - Ilustração em um arquivo de texto
 - Quando uma nova “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 9×2000 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

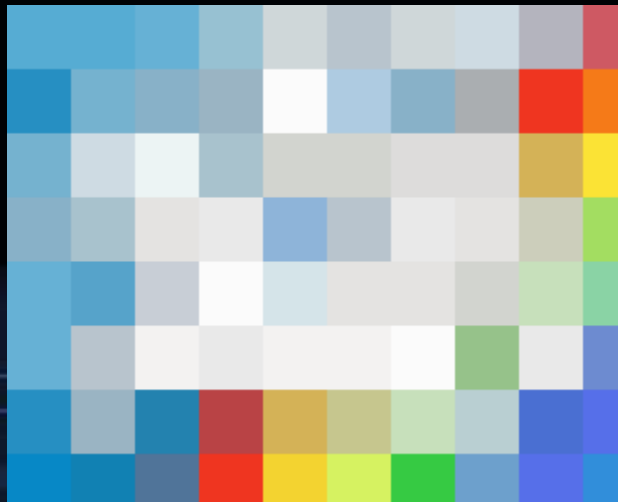


- **LZW e o formato de imagem GIF**
 - GIF utiliza a técnica LZW
 - GIF é um dos formatos de armazenamento de imagens 256 cores sem perdas
 - imagens com um máximo de 256 cores
 - ao converter imagem true color, com 24 bits/pixel, para o formato GIF, estamos perdendo grande parte da informação de cor
 - 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 EÀ).+;)ûQqâgu#..5l

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"ÄE

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.îVÜ+çÄæsk±@;Ü

0470h: 3B A5 B5 C8 62 3B EE C0 38 4E 3B 33 8E C3 8E 8D ;¥µÈb;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'i3oæéi>Ö<S'æ

04D0h: 76 E2 B9 CC 5B 6F 36 E3 56 9E 76 39 73 D7 9D 78 vâ'i[06âVžv9s*.x

Template Results - GIF.bt

Name	Value	Start	Size
ushort Height	960	8h	2h
struct LOGICALSCREENDESCRIPTOR_PACKEDFIELDS PackedFields		Ah	1h
UBYTE BackgroundColorIndex	0	Bh	1h
UBYTE 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
UBYTE 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
UBYTE 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 esgotadas 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



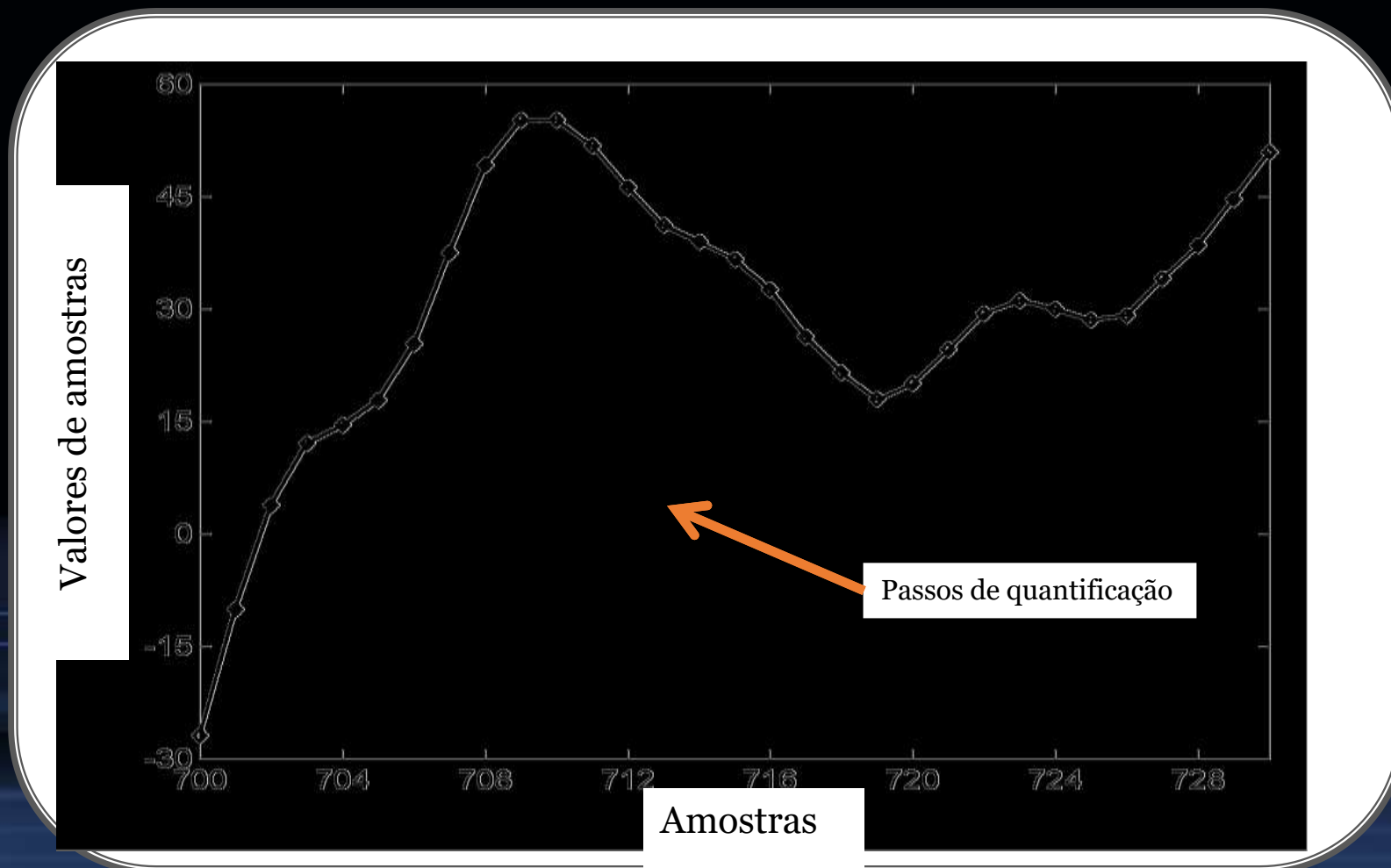
- **Conteúdo:**

- 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**

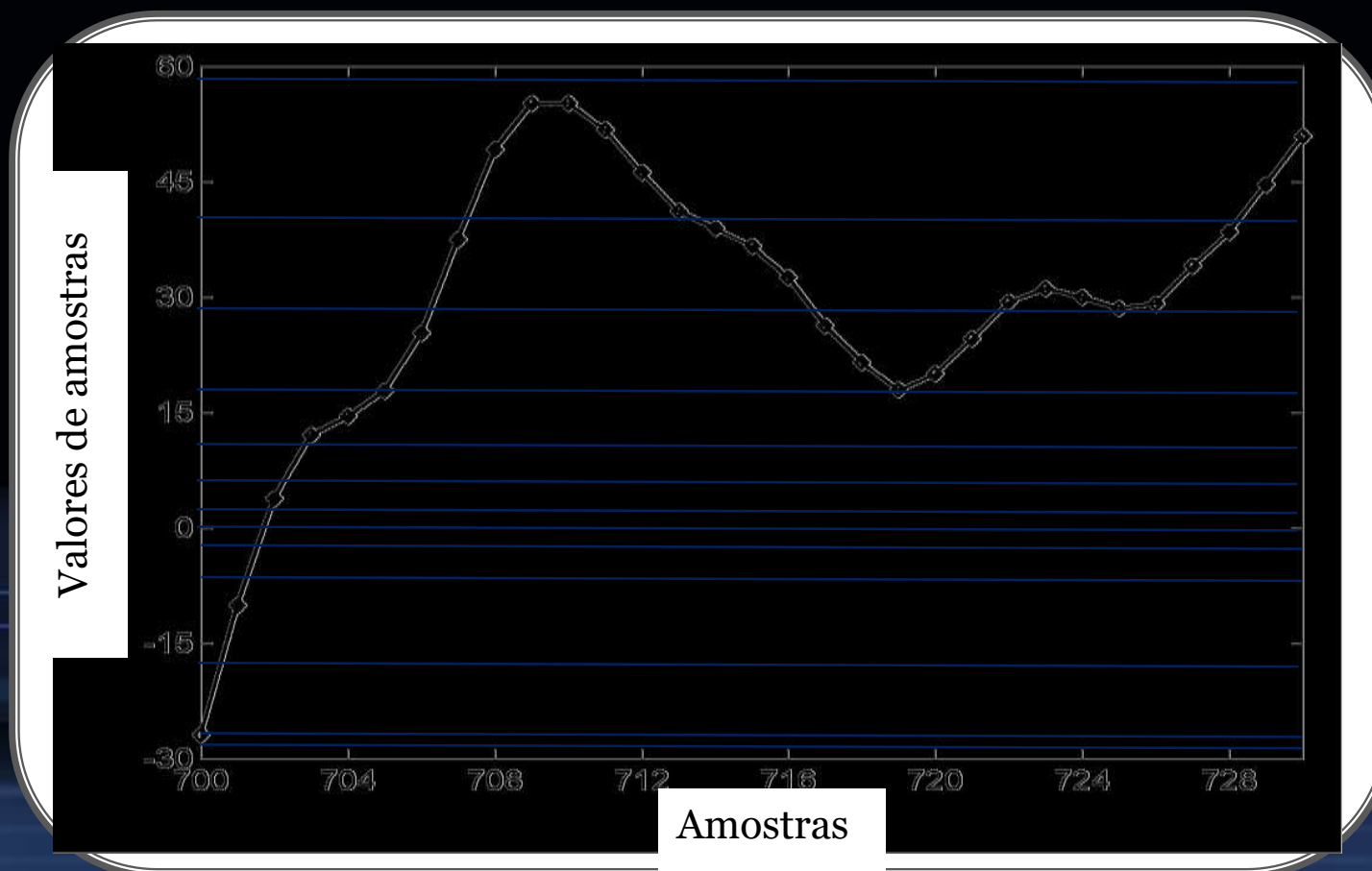
- Amostras são quantificadas com mesmo passo



Técnicas de Compressão de Áudio Digital

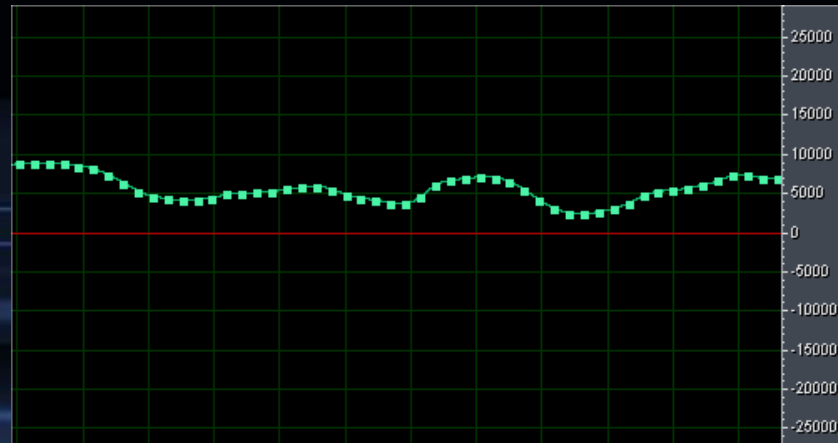
- **Codificação PCM não linear**

- 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



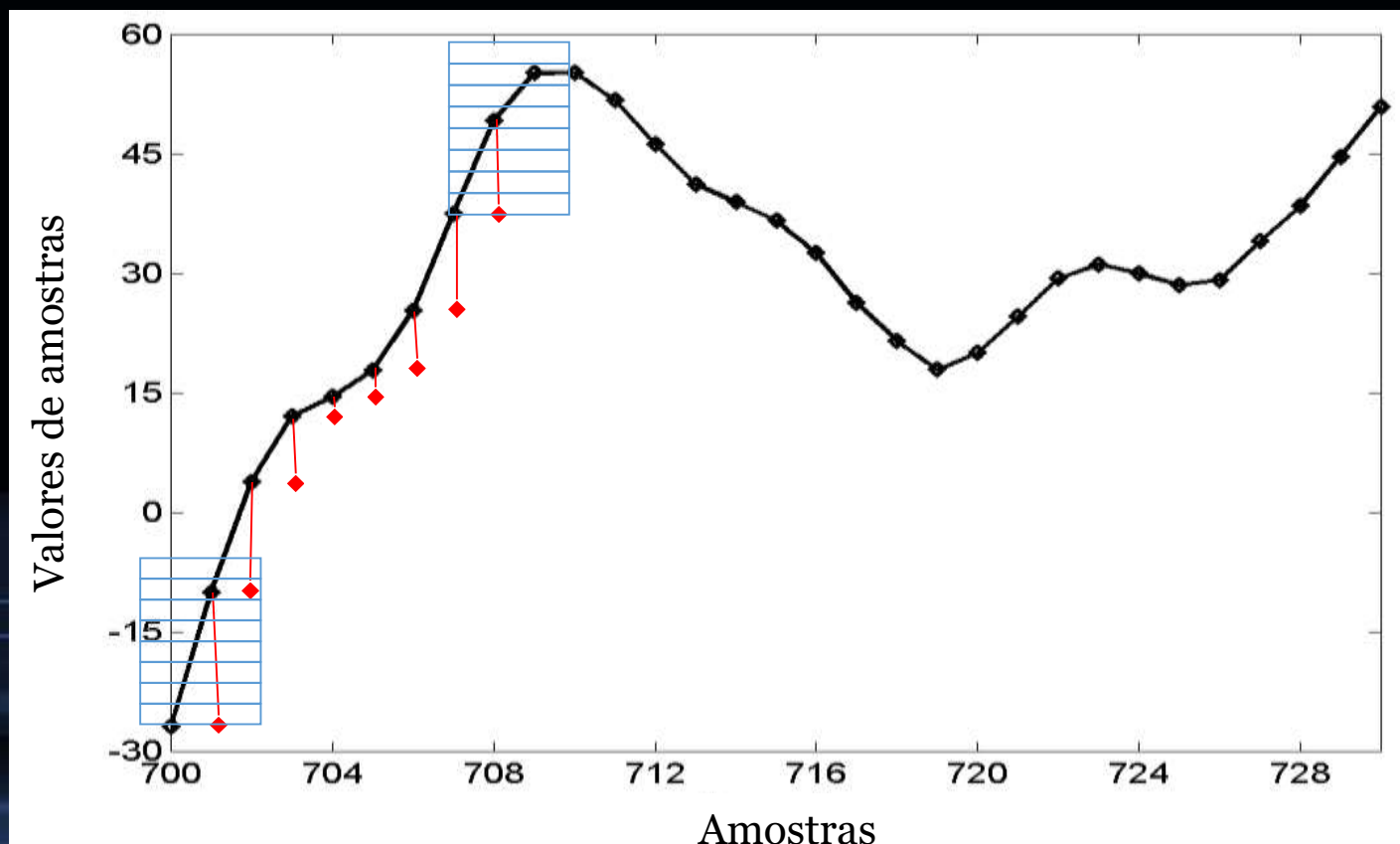
Técnicas de compressão de áudio

- **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

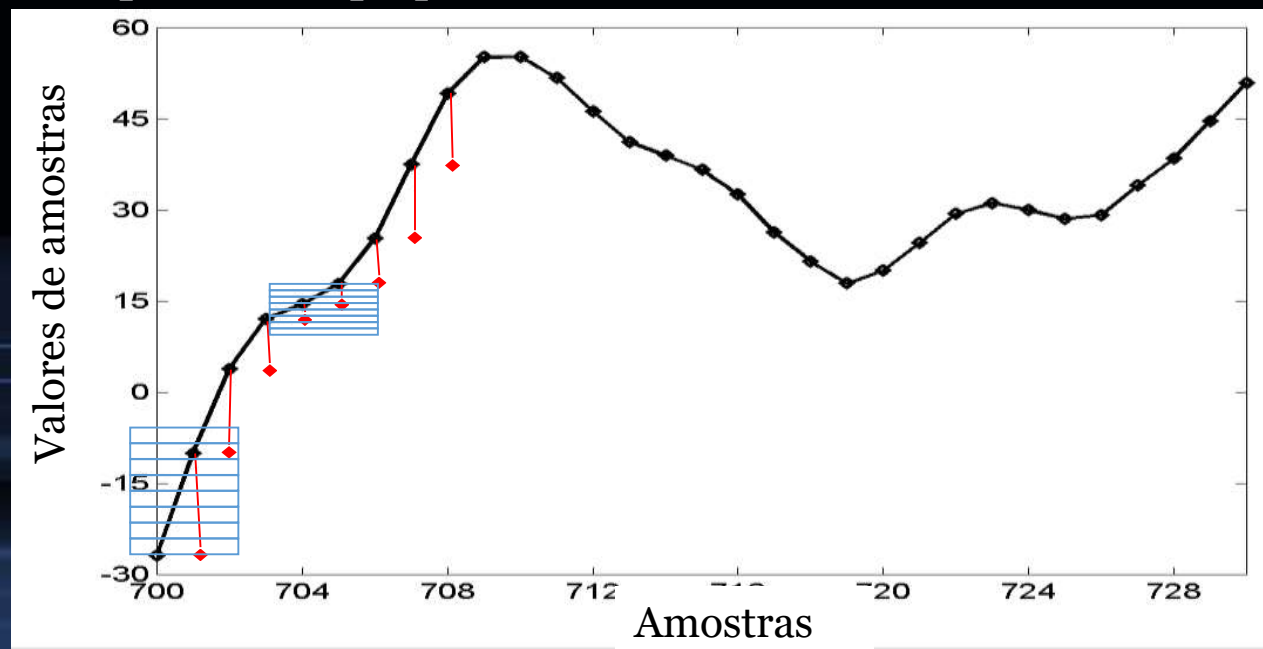
$$x[n] = \sum_{k=1}^P a_k x[n-k] + e[n]$$

- $x[n-k]$: 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

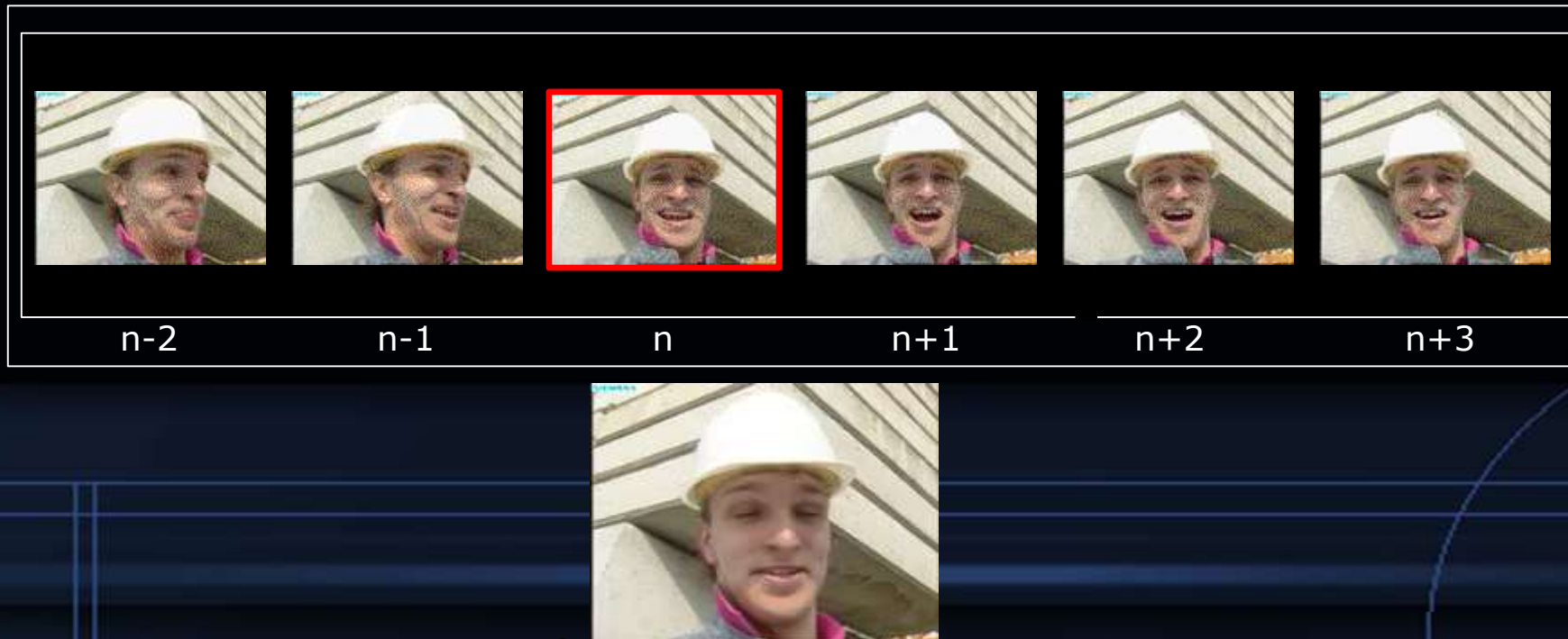
- **Imagens digitais puras são codificadas em PCM**
 - Representados por matrizes de píxeis



- 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)

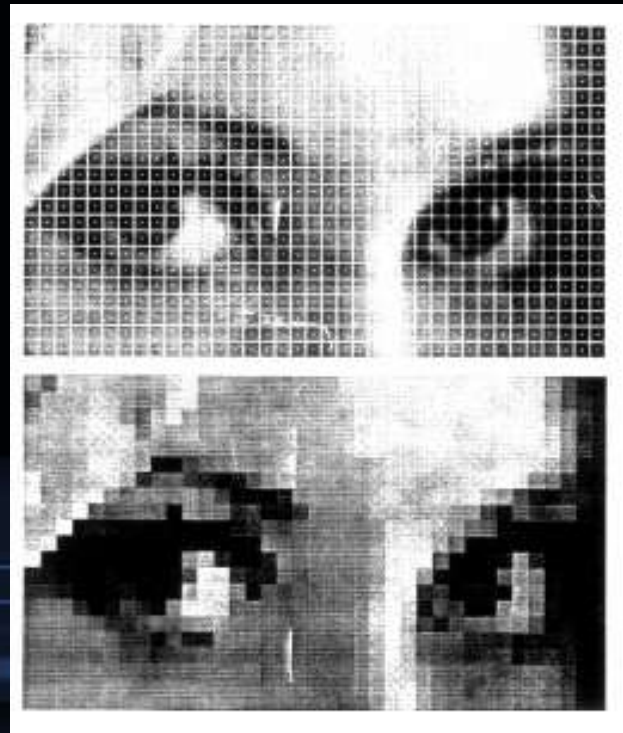
Técnicas de Compressão de Vídeo e Imagem

- **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

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

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

\wedge
 $s_{m,n} = 0.8s_{m,n-1} - 0.62s_{m-1,n-1} + 0.8s_{m-1,n}$ 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

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

Original

Saída prevista

Saída DPCM

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**

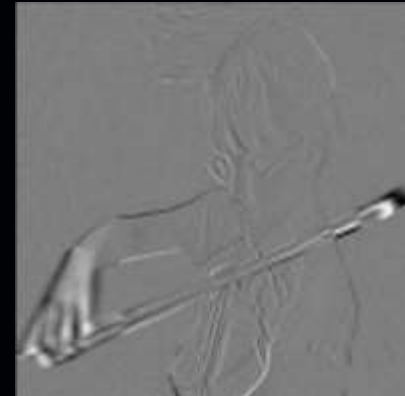
Quadro Preditor



Quadro Atual

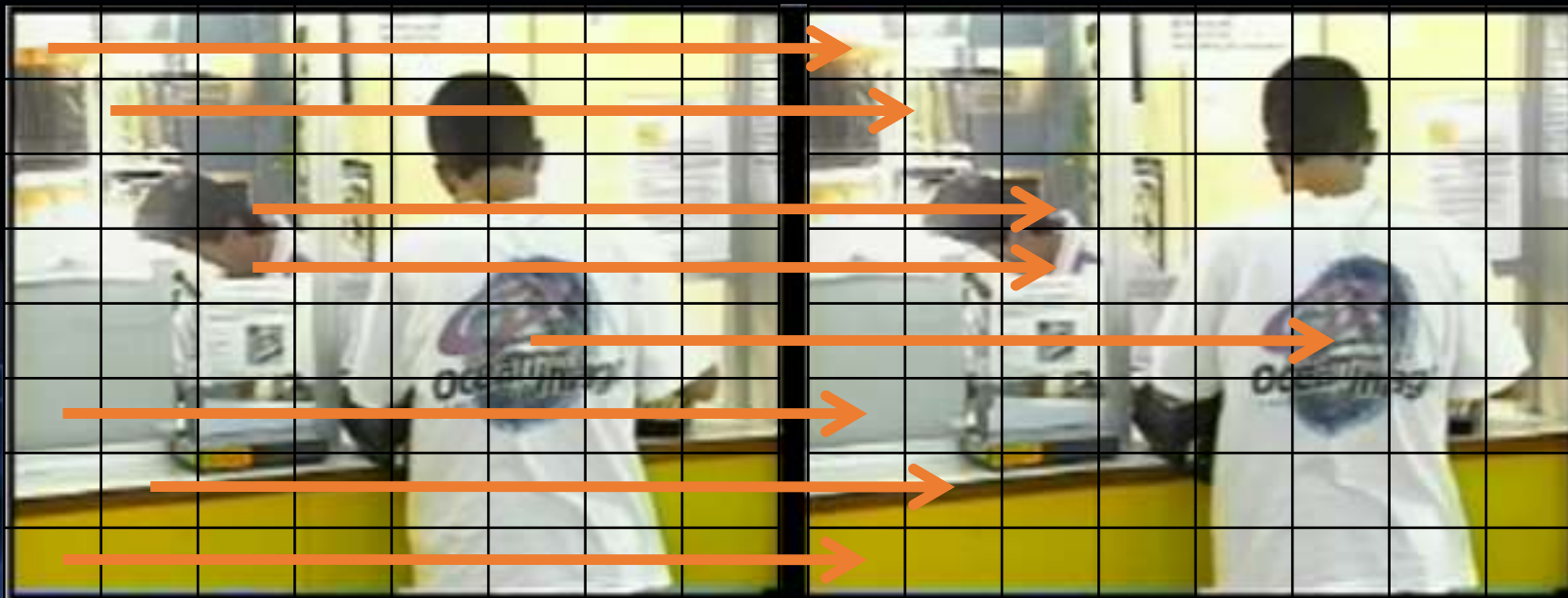


Diferença



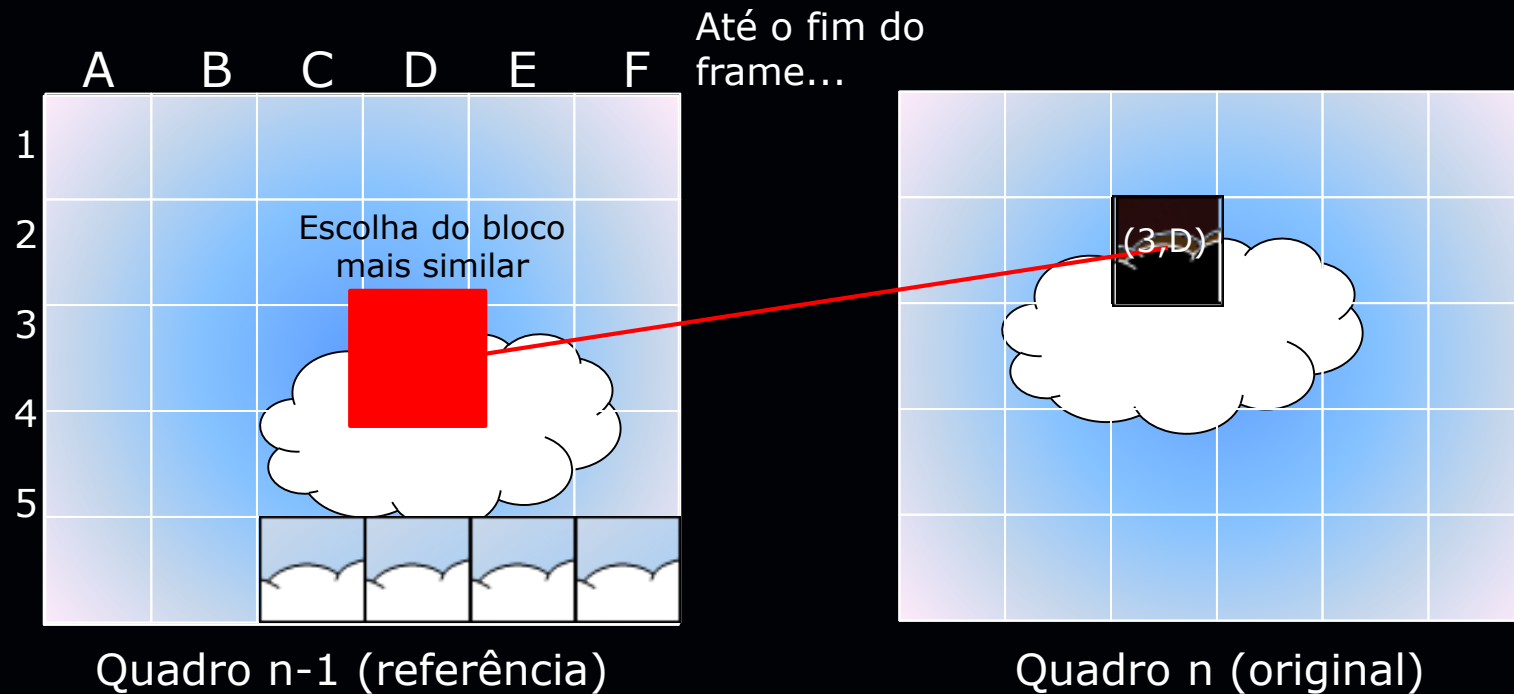
Técnicas de Compressão de Vídeo

- **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