

Técnicas de Compactação

Profa. Débora Christina Muchaluat Saade

debora@midia.com.uff.br

Algoritmo LZ77

- ✓ **Variação do LZW usado em vários compactadores comerciais**
 - *PKZIP/PKUNZIP (para o DOS)*
 - *ARJ*
 - *GZIP/GUNZIP*
 - *Compress/Uncompress (para o UNIX)*
- ✓ **Princípio de funcionamento:**
 - *Em uma sequência de caracteres, o algoritmo procura na janela corrente a maior sub-sequência que casa com o início do lookahead buffer e dá como saída um ponteiro para o início da sub-sequência na janela e o primeiro caracter no lookahead buffer que não casa com a sequência. Se não houver sub-sequência, a saída é um ponteiro nulo e o caracter na posição atual da sequência original.*

Algoritmo LZ77

✓ Termos utilizados no algoritmo:

- ***Sequência de entrada***
 - Sequência de caracteres a ser comprimida
- ***Character:***
 - Elemento básico da sequência de entrada
- ***Posição atual:***
 - Posição do character na sequência de entrada que está sendo codificado no momento (início do lookahead buffer)
- ***Lookahead buffer:***
 - Sequência de caracteres a partir da posição atual até o fim da sequência de entrada

Algoritmo LZ77

✓ Termos utilizados no algoritmo:

- ***Janela:***

- Janela de tamanho W contém W caracteres a partir da posição atual em direção ao início da sequência de entrada, i.e. os últimos W caracteres processados

- ***Ponteiro:***

- Um ponteiro aponta para a sequência que casa na janela e também indica seu comprimento (L)

Algoritmo LZ77

✓ **Algoritmo:**

1. *Faça posição atual igual ao início da sequência de entrada*
2. *Encontre a maior sub-sequência (longest match) na janela que casa como conteúdo do lookahead buffer*
3. *Dê como saída o par (P,C), onde:*
 - P é o ponteiro para a sub-sequência na janela
 - C é o primeiro caracter no lookahead buffer que não casa com a sub-sequência
4. *Se o lookahead buffer não estiver vazio, mova a posição atual (e a janela) L+1 caracteres para frente e volte para o passo 2 (L é o comprimento da sub-sequência).*

Algoritmo LZ77

✓ Exemplo: sequência **AABCBBABC**

Pos	1	2	3	4	5	6	7	8	9
Char	A	A	B	C	B	B	A	B	C

✓ Processo de Codificação

W (janela)

A
AAB
AABC
AABCBB
AABCBBABC

Step	Pos	Match	Char	Saída
1	1	--	A	(0,0) A
2	2	A	B	(1,1) B
3	4	--	C	(0,0) C
4	5	B	B	(2,1) B
5	7	AB	C	(5,2) C

Algoritmo LZ77

✓ **Decodificação:**

- *A janela é mantida da mesma forma que na codificação.*
- *Em cada passo, o algoritmo lê o par (P,C) da entrada e dá como saída a sequência da janela a partir de P seguida do caracter C*

✓ **Exemplo: $(0,0)A$ $(1,1)B$ $(0,0)C$ $(2,1)B$ $(5,2)C$**

1. *$(0,0)A \Rightarrow W = \text{nula}, \text{saída} = A$*
2. *$(1,1)B \Rightarrow W = A, \text{saída} = AAB$*
3. *$(0,0)C \Rightarrow W = AAB, \text{saída} = AABC$*
4. *$(2,1)B \Rightarrow W = AABC, \text{saída} = AABCBB$*
5. *$(5,2)C \Rightarrow W = AABCBB, \text{saída} = AABCBBABC$*

Algoritmo LZ77

✓ Considerações principais:

- *Boa taxa de compressão para vários tipos de dados*
- *Codificação pode ser lenta, já que várias comparações são feitas entre o lookahead buffer e a janela*
- *Decodificação muito simples e rápida*
- *Requisitos de memória são poucos tanto para codificação quanto para decodificação*
 - A única estrutura mantida em memória é a janela, que normalmente tem tamanho entre 4 e 64 KB.

Técnicas de Compressão

Profa. Débora Christina Muchaluat Saade

debora@midia.com.uff.br

Técnicas de Compressão

✓ Compressão:

- *Quando, na redução dos dados, há perda de informação*
- *Compressão com perdas*
- *Algumas técnicas são usadas em sinais específicos*
- *Compressão perceptualmente sem perdas*
 - humanos não percebem
 - Ex.: MP3 para áudio

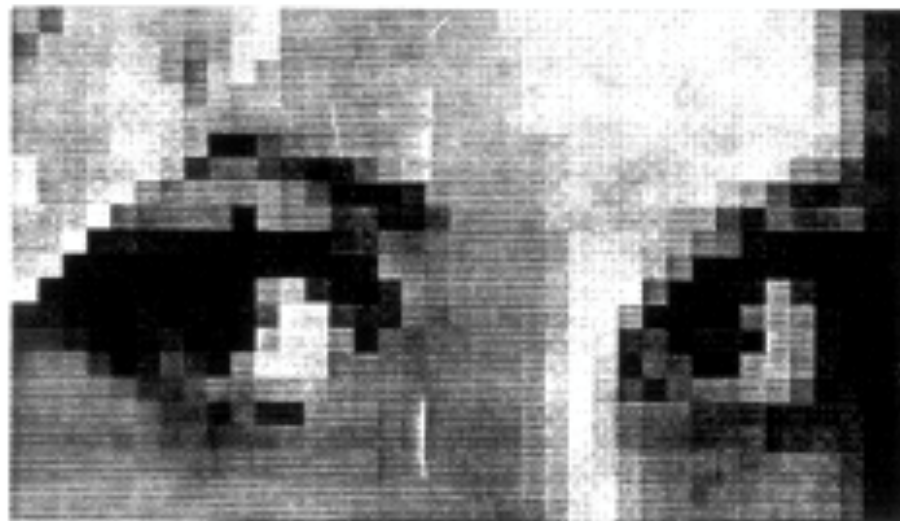
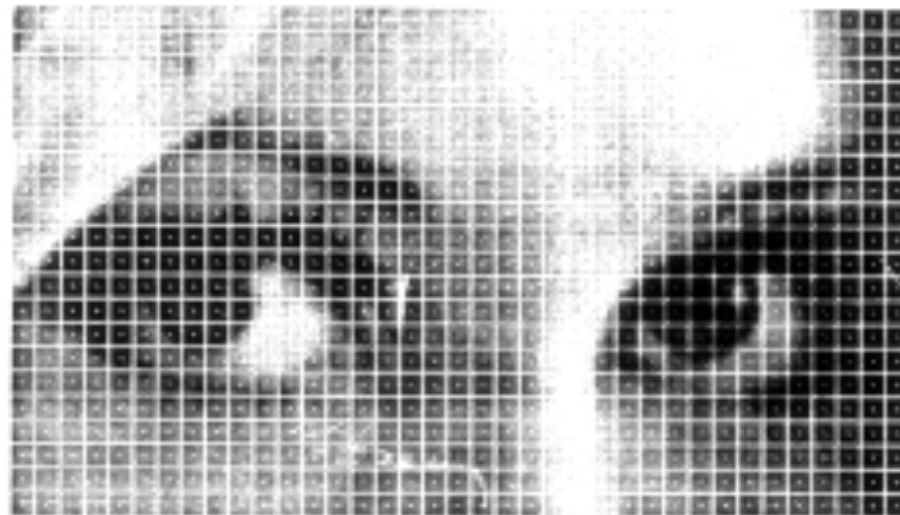
✓ Técnicas

- *Redução do domínio*
- *Redução do espaço de quantização*
- *Codificação preditiva*
- *Codificação por sub-bandas*
- *Codificação por transformadas*
- *Quantização vetorial*

Redução do Domínio

- ✓ **Idéia básica**
 - *Simplemente descarta algumas amostras*
- ✓ **Ex.: padrões para vídeo digital**
 - *Formatos 4:2:2, 4:2:0*
 - *Informações de luminância são mais importantes que crominância, então o número de amostras de crominância pode ser menor*
- ✓ **Ex.: compressão em imagens**
 - *diminui a resolução geométrica*
 - *aumenta o tamanho do pixel*

Redução do domínio em imagens



Redução do Espaço de Quantização

- ✓ **Idéia básica**

- *Diminuir a quantidade de bits por amostra*

- ✓ **Ex.: compressão de imagens**

- *Imagem original com 24 bits por pixel (16 milhões de cores)*
- *Imagem comprimida com 8 bits por pixel (256 cores) fazendo uma correspondência entre os códigos*

Codificação Preditiva

- ✓ **Codificação diferencial ou codificação relativa**
- ✓ **Idéia básica**
 - *Amplitude de uma amostra é grande, mas a diferença de amplitude entre amostras sucessivas é relativamente pequena*
 - *Ao invés de codificar o valor de cada amostra, codifica a diferença entre seu valor e o anterior*
 - Utiliza menos bits e obtém o mesmo erro
 - *Ex.:*
 - DPCM (*Differential Pulse Code Modulation*)
 - ADPCM (*Adaptive Differential Pulse Code Modulation*)

Codificação por Sub-Bandas

✓ Idéia básica

- *Divisão da banda passante do sinal em várias sub-bandas codificadas de forma distinta*
- *Trata com maior precisão as sub-bandas mais importantes do sinal*

✓ Por exemplo, sinal de voz:

- *300-800Hz – qualidade e timbre*
 - 8 bits por amostra
- *800-1400Hz – pouca informação*
 - 2 bits por amostra
- *1400-2400Hz – reconhecimento e intelegibilidade*
 - 8 bits por amostra
 - Maior informação de conteúdo (ex.: voz metálica)
- *2400-3400Hz – pouca informação*
 - 3 bits por amostra

Codificação por Transformadas

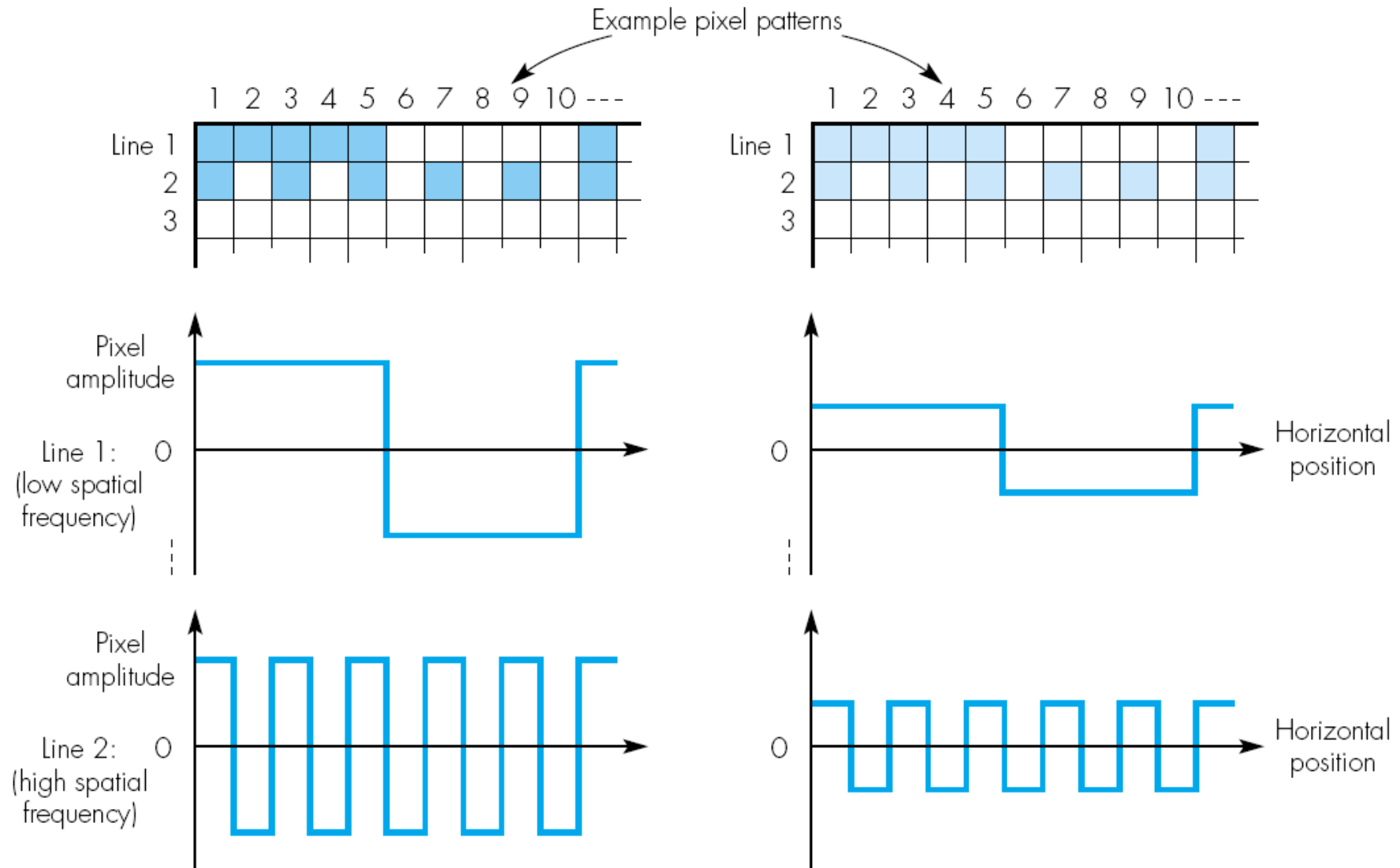
✓ Idéia básica

- *Transformar os dados para outro domínio matemático onde uma técnica de compressão seja melhor aplicada*
 - Ex.: Transformada de Fourier => transforma sinais no domínio do tempo para o domínio da frequência
- *Deve existir uma transformada inversa*
- *Transformadas eficazes para redução de dados são:*
 - DCT – Discrete Cosine Transform (JPEG)
 - FFT – Fast Fourier Transform (MPEG-áudio)

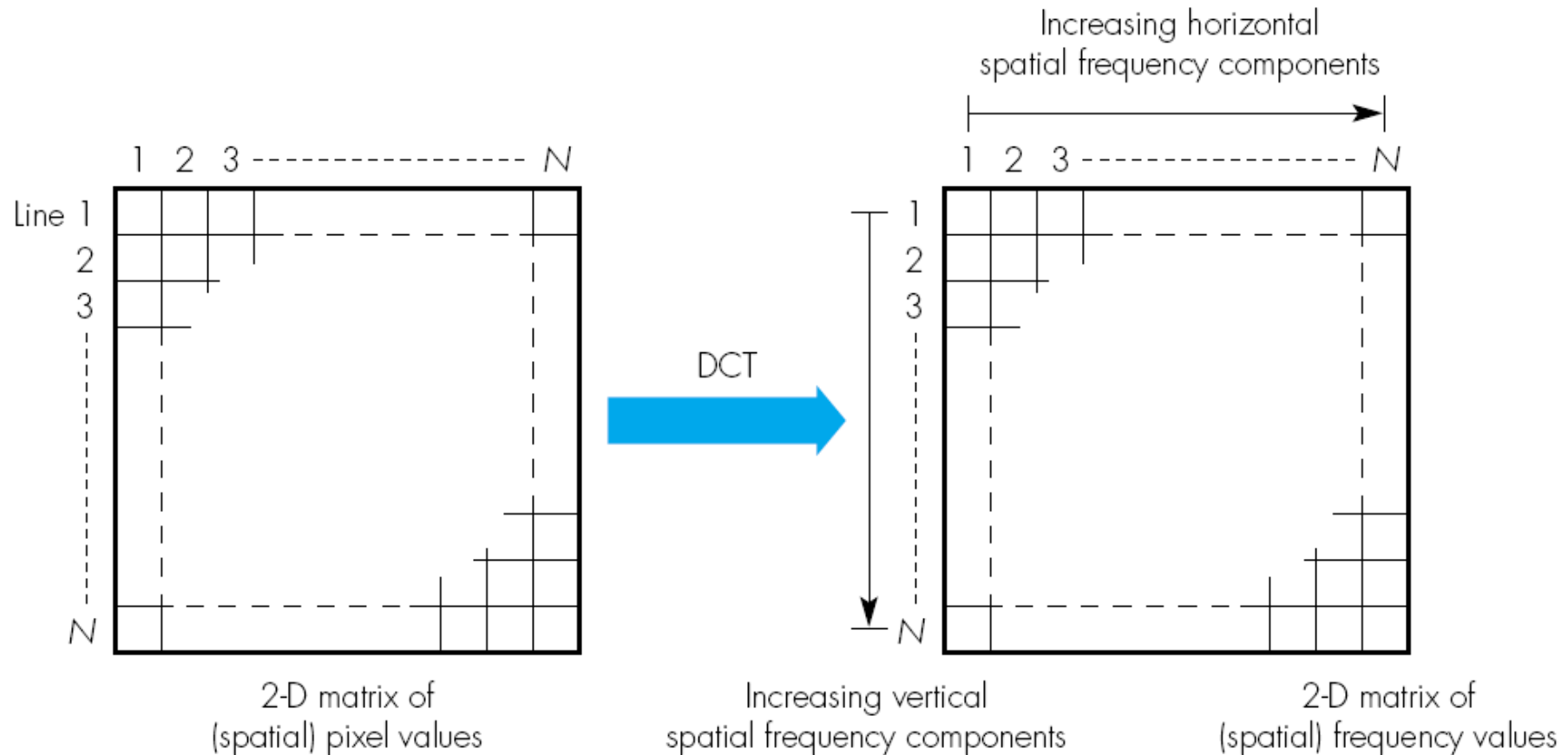
Transformadas em Imagens

- ✓ **Existem várias similaridades entre amostras (no tempo) e pixels. Na verdade, podemos considerar os pixels como se fossem amostras do “sinal imagem”, só que amostras obtidas não no tempo, mas no espaço.**
- ✓ **É exatamente por isso que podemos aplicar todas as técnicas aplicadas em sinais contínuos nas imagens estáticas.**

Codificação por Transformadas



Codificação por Transformadas



DCT = discrete cosine transform

Quantização Vetorial

- ✓ Fluxo de dados é dividido em blocos
- ✓ Constrói uma tabela (ou usa uma predefinida) contendo o conjunto de valores que mais aparecem (valores padrões)
- ✓ Para cada bloco, a tabela é consultada para achar o padrão mais parecido
- ✓ Então cada bloco é codificado com o índice do vetor
- ✓ O decodificador deve conhecer a mesma tabela e usa os índices para gerar uma aproximação do fluxo de dados original

...11438565567896313...

...1143 8565 5678 9631 3...

v1	8456
v2	6102
v3	0034
...	
Vn	5688

Dicionário de Vetores

...1143 8565 5678 9631 3... → V3 V1 Vn V4