

Processamento de Áudio e Vídeo

Prof. Leonardo Araújo



<https://sites.google.com/site/leolca/teaching/multimedia-signal-processing>

1 Introdução

2 Técnicas Básicas de Compressão

- RLE

Processamento de Áudio e Vídeo

- ▶ Compressão
- ▶ Com/Sem perdas
- ▶ mp3, jpeg, mpeg, flac, zip, gif, png, etc
- ▶ sinais de áudio, fala, imagens e vídeo
- ▶ qualidade, taxa de compressão, custo

- Compressão
- Com/Sem perdas
- mp3, jpeg, mpeg, flac, dvb, gif, png, etc
- dados de áudio, vídeo, imagens e vídeo
- qualidade, taxa de compressão, tempo

- O conceito de compressão surge naturalmente quando estamos lidando com comunicação.
- Compressão de dados é o processo de converter dados provenientes de uma fonte em outros dados com menor tamanho.
- Armazenamento e transmissão (no fundo, ambos são formas de comunicação).
 - linha telefônica analógica
 - comunicação digital através desta linha telefônica analógica
 - link de comunicação de rádio entre a sonda espacial Galileu orbitando Júpiter e a Terra
 - armazenamento e reprodução de áudio ou vídeo (ou dados) em um CD, DVD ou disco rígido
 - reprodução celular em que a informação sobre as células é contida no DNA

- Compressão
- Com/Sem perdas
- mp3, jpeg, mpeg, flac, dv, gH, png, etc
- Áudio de áudio, ía b, imagens e vídeo
- Qualidade, taxa de compressão, etc

Métodos de compressão sem perda (alguns são vistos na disciplina Teoria da Informação) possuem como limite a entropia. Reconstrução exata da mensagem produzida pela fonte. Remover redundância.

Métodos de compressão com perda utilizam-se do fato de que muita informação pode ser perdida sem ser percebida ou aceita-se uma distorção do sinal em prol de uma maior compressão.

- ▶ Compressão
- ▶ Com/Sem perdas
- ▶ mp3, jpeg, mpeg, flac, dv, gif, png, etc
- ▶ sinais de áudio, falas, imagens e vídeo
- ▶ qualidade, taxa de compressão, tempo

- Áudio, fala, imagens e vídeo são originalmente sinais analógicos.
- Conversão em sinais digitais: amostragem, quantização, codificação.

- ▶ Compressão
- ▶ Com/Sem perdas
- ▶ mp3, jpeg, mpeg, flac, dv, gif, png, etc
- ▶ Áudio de áudio, ía b, imagens e vídeo
- ▶ Qualidade, custo de compressão, tempo

A qualidade da compressão pode ser uma medida objetiva ou subjetiva. Na maioria das vezes, iremos realizar medidas objetivas pois realizar testes subjetivos é muito dispendioso. Podemos escolher medidas objetivas que sejam bem correlacionadas com medidas subjetivas.

O custo de compressão e descompressão podem, em geral, serem diferentes. Descompressão deve ser privilegiada pois é realizada diversas vezes e geralmente por terminais com menor poder computacional.

Imagem digital

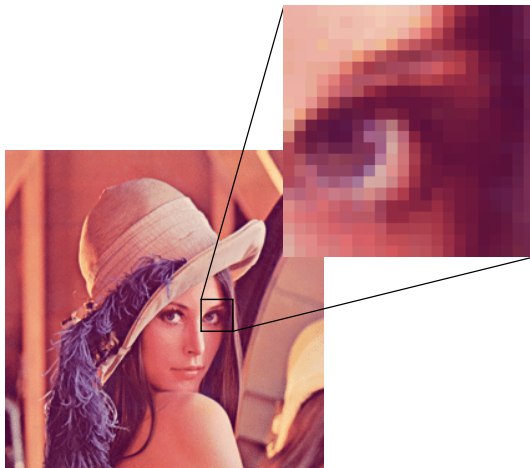


Figura 1: Lena - detalhe.

Espaço necessário para armazenar uma foto

- ▶ câmera 10 Mpixel
- ▶ 3 bytes por pixel (RGB)
- ▶ cada foto requer 30 Mbyte
- ▶ um cartão de memória de 2 Gbytes é capaz de armazenar 66 fotos

Espaço necessário para armazenar um vídeo

- ▶ 480 x 720, 30 fps
- ▶ 345.600 pixels por frame
- ▶ RGB 3 bytes por pixel
- ▶ 1.036.800 byte, aprox. 1 Mbyte por frame
- ▶ 30 frames requerem 31.104.000 bytes, aprox. 31 Mbyte por segundo
- ▶ um CD de 650 Mbytes é capaz de armazenar apenas 21 segundos de vídeo e um DVD de 4.7 GB apenas 155 segundos de vídeo.

Dilema de compressão

Quando devemos parar a busca por uma **melhor** compressão?

melhor:

- ▶ menor tamanho da representação digital resultante
- ▶ eficiência computacional (compressão e/ou descompressão)
- ▶ simplicidade do algoritmo

Qual é o limite de compressão para um determinado dado?

Processamento de Áudio e Vídeo

└─ Introdução

└─ Dilema de compressão

Modificar um algoritmo para melhorar a taxa de compressão em 1% pode acarretar um aumento de 10% no tempo de execução do algoritmo e ainda mais sobre a complexidade do programa.

Quando devemos parar a busca por uma **melhor** compressão?

resultos:

- ▶ menor tamanho da representação digital resultante
- ▶ eficiência computacional (compressão e/ou decompressão)
- ▶ simplicidade de algoritmos

Qual é o limite de compressão para um determinado dado?

Quando devemos parar a busca por uma **melhor** compressão?

resposta:

- ▶ menor tamanho da representação digital resultante
- ▶ eficiência computacional (compressão e/ou decompressão)
- ▶ simplicidade de algoritmos

Qual é o limite de compressão para um determinado dado?

Conjecturas¹.

- Compressão de dados pode ser interpretada como o processo de remover complexidades (redundâncias) desnecessárias na informação, e desta forma, maximizando a simplicidade enquanto preserva o máximo possível do poder discricionário dos dados.
- Todo tipo de computação e racionalização formal pode ser compreendida como compressão de informação através do processo de identificar padrões, busca e unificação das instâncias destes padrões.

Termos I

compressor ou codificador é o programa que comprime os dados crus na entrada e cria uma saída de dados comprimida (com baixa redundância).

decompressor ou decodificador converte os dados na direção oposta.

fluxo é o dado a ser comprimido, armazenado como um arquivo ou transmitido.

dado não-codificado, cru, ou original é o fluxo de dados da entrada.

dado codificado ou comprimido é o fluxo de saída.

método de compressão não-adaptativo é rígido e não modifica sua operação ou seus parâmetros em resposta aos dados em particular que estão sendo comprimidos.

método adaptativo analisa os dados crus e modifica sua operação e/ou parâmetros de acordo com os dados em mãos.

método semi-adaptativo utiliza 2 passagens aonde, na primeira, realiza a leitura dos dados e contabiliza estatísticas dos dados a serem comprimidos; na segunda passagem, realiza de fato a compressão utilizados parâmetros determinados na primeira varredura.

Termos II

método localmente adaptativo se adapta às condições locais do fluxo de dados e varia à medida que move ao longo dos dados.

compressão com perdas/sem perdas : Para atingirem maior compressão, os métodos de compressão com perda perdem informação. Os métodos de compressão sem perda não admitem perder informação alguma.

Compressão em cascata ocorre quando diferentes métodos de compressão são utilizados um em seguida do outro.

Compressão perceptiva ocorre quando apenas a informação imperceptível pelos nossos sentidos é removida.

Compressão simétrica é o caso em que o compressor e descompressor utilizam basicamente o mesmo algoritmo, porém em direções opostas.

Complacente é o codificador/decodificador que gera/lê de forma correta um fluxo de dados (Qualquer pessoa é livre para implementar seu próprio algoritmo).

Universal é o método de compressão de dados que não depende da estatística dos dados.

Termos III

Razão de Compressão = tamanho do dado de saída / tamanho do dado de entrada.

Fator de Compressão = tamanho do dado de entrada / tamanho do dado de saída = (razão de compressão) $^{-1}$.

Ganho de Compressão = $100 \log_e$ (tamanho de referência / tamanho comprimido), aonde o tamanho de referência é o tamanho dos dados de entrada ou o tamanho do dado de saída comprimido por algum algoritmo padrão.

Erro médio quadrático (MSE) e relação sinal ruído de sinal (PSNR) são utilizados para medir a distorção causada por uma compressão com perdas.

Termos

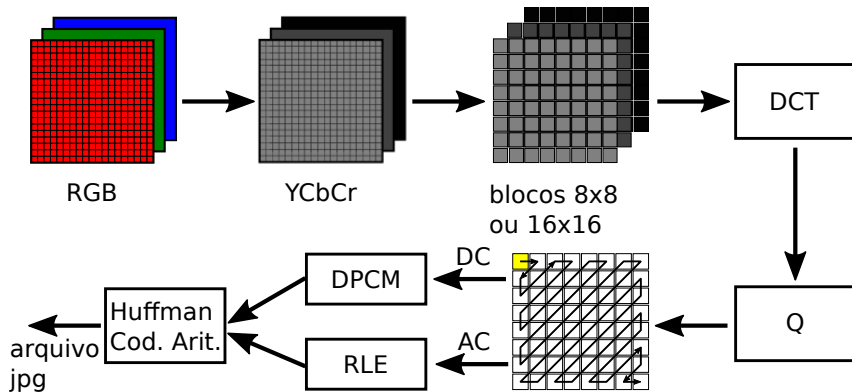


Figura 2: Esquema de compressão JPEG.

Slides- introdução ao GNU Octave



https://drive.google.com/open?id=1ew5fl9v_0Iybsy3KdEgIvohLTcuwuru_

Notebook - introdução



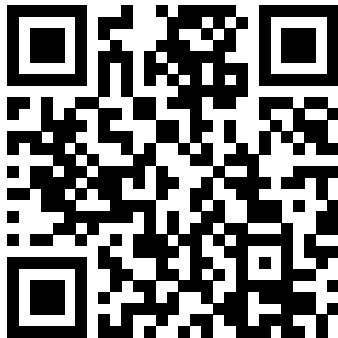
`https://nbviewer.jupyter.org/github/leolca/notebooks/blob/master/aev/
introducao.ipynb`

Notebook - imagem colorida



[https://nbviewer.jupyter.org/github/leolca/notebooks/blob/master/aev/
introdocao_imagem_colorida.ipynb](https://nbviewer.jupyter.org/github/leolca/notebooks/blob/master/aev/introdocao_imagem_colorida.ipynb)

Leitura



David Salomon, Giovanni Motta - *Handbook of Data Compression*, 2010

<https://books.google.com.br/books?id=LHCY4VbiFqAC>

Introduction, Basic Techniques (Salomon et al., 2010)

Compressão RLE

Exemplo:

string: '2. all is too well'

codificação: '2. a@2l is t@2o we@2l'

Método MNP5 era utilizado nos modems antigos.

Processamento de Áudio e Vídeo

└ Tecnicas Básicas de Compressão

└ RLE

└ Compressão RLE

Exemplo:
original: "2, a016 m 0 2"
codificação: "2, a016 0 2s m021"

Método MNP era utilizado nos modems antigos.

MNP : Microcom Networking Protocol

"The MNP5 method is a two-stage process that starts with run-length encoding, followed by adaptive frequency encoding."(Salomon, 2000)

"With MNP 5, the data received from the computer are first compressed with a simple algorithm, and then passed into the MNP 4 packetizing system for transmission. On best-case data the system offered about 2:1 compression, but in general terms about 1.6:1 was typical, at least on text. As a result a 2400 bit/s modem would appear to transfer text at 4000 bit/s, even though the modem was still running at the same 600 baud * 4 bits per symbol rate.

This dramatic increase in throughput allowed Microcom modems to remain somewhat competitive with models from other companies that were otherwise nominally much faster. For instance, Microcom generally produced 1200 and 2400 bit/s modems using commodity parts, while companies like USRobotics and Telebit offered models with speeds up to 19200 bit/s."(https://en.wikipedia.org/wiki/Microcom_Networking_Protocol)

Compressão RLE

Exemplo: uma imagem em tons de cinza com 8-bit de profundidade começa com os seguintes valores

12, 12, 12, 12, 12, 12, 12, 12, 12, 35, 76, 112, 67, 87, 87, 87, 5, 5, 5, 5, 5, 5, 1, ...

será comprimida como 9,12,35,76,112,67,3,87,6,5,1, ...

Se utilizarmos como *flag* o valor 255, então a sequência acima será expressa por
255, 9, 12, 35, 76, 112, 67, 255, 3, 87, 255, 6, 5, 1, ...

grupos de 8

10000010,9,12,35,76,112,67,3,87,100...,6,5,1, ...

Exemplo RLE - GNU Octave



https://nbviewer.jupyter.org/github/leolca/notebooks/blob/master/aev/rle_mario.ipynb

Move-to-Front Coding

Consideramos o alfabeto de símbolos \mathcal{A} como uma lista onde os símbolos mais frequentes estarão dispostos no início da lista.

O método é localmente adaptativo, já que ele se adapta à frequência dos símbolos em cada região do fluxo de dados.

Move-to-Front Coding - Exemplo (Salomon et al., 2010)

Exemplo: entrada a ser codificada: **abcddcbamnoppnm**

$C = (0, 1, 2, 3, 0, 1, 2, 3, 4, 5, 6, 7, 0, 1, 2, 3)$

– utilizando move-to-front

$C' = (0, 1, 2, 3, 3, 2, 1, 0, 4, 5, 6, 7, 7, 6, 5, 4)$

– sem utilizar move-to-front

a	abcdmnop	0	a	abcdmnop	0
b	abcdmnop	1	b	abcdmnop	1
c	baedmnop	2	c	abcdmnop	2
d	cbadmnop	3	d	abcdmnop	3
d	dcbamnop	0	d	abcdmnop	3
c	dcbamnop	1	c	abcdmnop	2
b	cdabmnop	2	b	abcdmnop	1
a	bedamnop	3	a	abcdmnop	0
m	abcdmnop	4	m	abcdmnop	4
n	mabednop	5	n	abcdmnop	5
o	nmabedop	6	o	abcdmnop	6
p	onmabedp	7	p	abcdmnop	7
p	ponmabed	0	p	abcdmnop	7
o	ponmabed	1	o	abcdmnop	6
n	opnmabed	2	n	abcdmnop	5
m	nopmabed	3	m	abcdmnop	4
	mnopabed				

Move-to-Front Coding - Exemplo (Salomon et al., 2010)

O resultado C obtido pelo move-to-front é tal que, na média, os valores em C são pequenos (os valores no início do dicionário são os mais prováveis). Isto faz com que a saída seja propícia para ser codificada através da codificação de Huffman ou codificação aritmética.

i	Code	Size
1	1	1
2	010	3
3	011	3
4	00100	5
5	00101	5
6	00110	5
7	00111	5
8	0001000	7
9	0001001	7
\vdots	\vdots	\vdots
15	0001111	7
16	000010000	9

Figura 3: Exemplo de código de tamanho variável.

Move-to-Front Coding

Variações:

- 1) Move-ahead-k: O elemento do alfabeto A que corresponde ao símbolo corrente será deslocado k posições para cima na lista ao invés de ir para o topo da lista.
- 2) Wait-c-and-move: O elemento do alfabeto A será deslocado para o início da lista apenas após aparecer c vezes durante a codificação. item Wait-c-and-ahead-k: Um combinação das duas variantes anteriores.

Exemplo Move-to-Front - GNU Octave



`https://nbviewer.jupyter.org/github/leolca/notebooks/blob/master/aev/
move-to-front.ipynb`

Salomon, D. (2000). *Data Compression: The Complete Reference*. Springer New York.

Salomon, D., Bryant, D., and Motta, G. (2010). *Handbook of Data Compression*. Springer London.