Special Module on Media Processing and Communication

Dayalbagh Educational Institute (DEI)  
Dayalbagh Agra  
PHM 961

Indian Institute of Technology Delhi (IITD)  
New Delhi  
SIV 864
Recap

- Lecture 1
  - Overview
  - Digital Representation
    - Audio
    - Image
    - Video
    - Geometry
  - Need of Compression
Image Compression

Compression Ratio

\[ C_r = \frac{n_o}{n_c} \]

\( n_o \) = Number of carrying units (bits) in the original data (image)

\( n_r \) = Number of carrying units (bits) in the compressed data (image)

Also,

\[ R_d = 1 - \frac{1}{C_r} \]

\( R_d \) = Relative data redundancy
Image Compression

Fidelity Criteria

Measure of loss or degradation

- Mean Square Error (MSE)

\[
MSE = \frac{1}{MN} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} [f(i, j) - f'(i, j)]^2
\]

- Signal to Noise Ratio (SNR)
- Subjective Voting
Compression

Compression Techniques

• Loss-less Compression
  Information can be compressed and restored without any loss of information

• Lossy Compression
  Large compression, perfect recovery is not possible
Compression

Compression Techniques

Symmetric
• Same time for compression (coding) and decompression (decoding)
• Used for dialog (interactive) mode applications

Asymmetric
• Compression is done once so can take longer
• Decompression is done frequently so should be fast
• Used for retrieval model applications
Image Compression

Data Redundancy

- **Coding**
  Variable length coding with shorter codes for frequent symbols

- **Interpixel**
  Neighboring pixels are similar

- **Psychovisual**
  Human visual perception - limited
Image Compression

Coding Redundancy

**Example**: (from Digital Image Processing by Gonzalez and Woods)

<table>
<thead>
<tr>
<th>$r_k$</th>
<th>$p_i(r_k)$</th>
<th>Code 1</th>
<th>$I_i(r_k)$</th>
<th>Code 2</th>
<th>$I_2(r_k)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$r_0 = 0$</td>
<td>0.19</td>
<td>000</td>
<td>3</td>
<td>11</td>
<td>2</td>
</tr>
<tr>
<td>$r_1 = 1/7$</td>
<td>0.25</td>
<td>001</td>
<td>3</td>
<td>01</td>
<td>2</td>
</tr>
<tr>
<td>$r_2 = 2/7$</td>
<td>0.21</td>
<td>010</td>
<td>3</td>
<td>10</td>
<td>2</td>
</tr>
<tr>
<td>$r_3 = 3/7$</td>
<td>0.16</td>
<td>011</td>
<td>3</td>
<td>001</td>
<td>3</td>
</tr>
<tr>
<td>$r_4 = 4/7$</td>
<td>0.08</td>
<td>100</td>
<td>3</td>
<td>0001</td>
<td>4</td>
</tr>
<tr>
<td>$r_5 = 5/7$</td>
<td>0.06</td>
<td>101</td>
<td>3</td>
<td>00001</td>
<td>5</td>
</tr>
<tr>
<td>$r_6 = 6/7$</td>
<td>0.03</td>
<td>110</td>
<td>3</td>
<td>000001</td>
<td>6</td>
</tr>
<tr>
<td>$r_7 = 1$</td>
<td>0.02</td>
<td>111</td>
<td>3</td>
<td>000000</td>
<td>6</td>
</tr>
</tbody>
</table>

**fixed length coding**
Avg length=3 bits

**variable length coding**
Avg length=2.7 bits
Image Compression

Interpixel Redundancy

Example: (from Digital Image Processing by Gonzalez and Woods)

Image

Histogram
Image Compression

Interpixel Redundancy

Example: (from Digital Image Processing by Gonzalez and Woods)

Image

High interpixel correlation

Histogram

Special Module on Media Processing and Communication

http://www.cse.iitd.ac.in/~pkalra/sil864
Image Compression

Psychovisual Redundancy

Example: (from Digital Image Processing by Gonzalez and Woods)

Original 256 levels  16 level quantization  IGS quantization
Image Compression

Loss-less Techniques

• Coding redundancy
  Variable length coding

• Interpixel redundancy
  Run length coding
  Predictive coding
Image Compression

Variable Length Coding (Huffman Coding)

Sequence of symbols (a1, a2, a3, a4, a5) with associated probabilities (p1, p2, p3, p4, p5)

• Start with two symbols of the least probability
  a1:p1
  a2:p2
• Combine (a1 or a2) with probability (p1+p2)
• Do it recursively (sort and combine)
• A binary tree construction
Image Compression

Variable Length Coding (Huffman Coding)

**Example:**
Symbols and their probabilities of occurrence
a1 (0.2), a2 (0.4), a3 (0.2), a4 (0.1), a5 (0.1)

Sort in probability

- a2 (0.4)
- a1 (0.2)
- a3 (0.2)
- a4 (0.1)
- a5 (0.1)
Image Compression

Variable Length Coding (Huffman Coding)

Example:

Sort

a2 (0.4)
a1 (0.2)
a3 (0.2)
a4 (0.1)
a5 (0.1)
Image Compression

Variable Length Coding (Huffman Coding)

Example:
Sort combine

\[ a2 (0.4) \]
\[ a1 (0.2) \]
\[ a3 (0.2) \]
\[ a4 (0.1) \]
\[ a5 (0.1) \]
Image Compression

Variable Length Coding (Huffman Coding)

Example:

<table>
<thead>
<tr>
<th>Sort</th>
<th>combine</th>
<th>Sort</th>
</tr>
</thead>
<tbody>
<tr>
<td>a2 (0.4)</td>
<td>0.4</td>
<td></td>
</tr>
<tr>
<td>a1(0.2)</td>
<td>0.2</td>
<td></td>
</tr>
<tr>
<td>a3(0.2)</td>
<td>0.2</td>
<td></td>
</tr>
<tr>
<td>a4(0.1) 0.2</td>
<td>0.2</td>
<td></td>
</tr>
<tr>
<td>a5(0.1)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Special Module on Media Processing and Communication  http://www.cse.iitd.ac.in/~pkalra/sil864
Image Compression

Variable Length Coding (Huffman Coding)

Example:

```
Sort  combine  Sort  combine  Sort  combine  Sort  combine
a2 (0.4)  0.4  0.4  0.6  1
a1 (0.2)  0.2  0.4  0.4  0.6
a3 (0.2)  0.2  0.2  0.6  0.4
a4 (0.1)  0.2  0.2
a5 (0.1)  0.2
```

Variable Length Coding (Huffman Coding)
Image Compression

Variable Length Coding (Huffman Coding)

Example:

<table>
<thead>
<tr>
<th>Sort</th>
<th>combine</th>
<th>Sort</th>
<th>combine</th>
<th>Sort</th>
<th>combine</th>
<th>Sort</th>
<th>combine</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>a2 (0.4)</td>
<td>0.4</td>
<td>1</td>
<td>0.4</td>
<td>1</td>
<td>0.6</td>
<td>0</td>
</tr>
<tr>
<td>01</td>
<td>a1 (0.2)</td>
<td>0.2</td>
<td>01</td>
<td>0.4</td>
<td>0</td>
<td>0.4</td>
<td>1</td>
</tr>
<tr>
<td>000</td>
<td>a3 (0.2)</td>
<td>0.2</td>
<td>000</td>
<td>0.2</td>
<td>0</td>
<td>0.4</td>
<td>1</td>
</tr>
<tr>
<td>0010</td>
<td>a4 (0.1)</td>
<td>0.2</td>
<td>0.2</td>
<td>0.2</td>
<td>0</td>
<td>0.4</td>
<td>1</td>
</tr>
<tr>
<td>0011</td>
<td>a5 (0.1)</td>
<td>0.2</td>
<td>001</td>
<td>0.1</td>
<td>0</td>
<td>0.6</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Assign code

Variable Length Coding (Huffman Coding)

Example:
Image Compression

Variable Length Coding (Huffman Coding)

Example:

Avg length code:

\[0.4 \times 1 + 0.2 \times 2 + 0.2 \times 3 + 0.1 \times 4 + 0.1 \times 4\]

= 2.2 bits
Image Compression

Variable Length Coding (Huffman Coding)

Example:

Avg length code:

\[0.4x1 + 0.2x2 + 0.2x3 + 0.1x4 + 0.1x4\]

\[= 2.2 \text{ bits}\]

Entropy

A measure of information that captures uncertainty

\[I(e) = \log \left( \frac{1}{p(e)} \right)\]

\[H = - \sum_{i=0}^{L-1} p(a_i) \log_2 p(a_i) \text{ bits / symbol}\]
Image Compression

Decoding

Example:

00111010001
Image Compression

Decoding

Example:

00111010001
Image Compression

Decoding

Example:

```
00111010001
```

Diagram:

```
Root

a5

a2

a1

a3

a2

0 1

0 1

0 1

a4

a5
```

http://www.cse.iitd.ac.in/~pkalra/sil864
A Method for the Construction of Minimum-Redundancy Codes

DAVID A. HUFFMAN*, ASSOCIATE, IRE

Summary—An optimum method of coding an ensemble of messages consisting of a finite number of members is developed. A minimum-redundancy code is one constructed in such a way that the average number of coding digits per message is minimized.

INTRODUCTION

ONE IMPORTANT METHOD of transmitting messages is to transmit in their place sequences of symbols. If there are more messages which might be sent than there are kinds of symbols available, then some of the messages must use more than one symbol. If it is assumed that each symbol requires the same time for transmission, then the time for transmission (length) of a message is directly proportional to the number of symbols associated with it. In this paper, the symbol or sequence of symbols associated with a given message will be called the “message code.” The entire will be defined here as an ensemble code which, for a message ensemble consisting of a finite number of members, N, and for a given number of coding digits, D, yields the lowest possible average message length. In order to avoid the use of the lengthy term “minimum-redundancy,” this term will be replaced here by “optimum.” It will be understood then that, in this paper, “optimum code” means “minimum-redundancy code.”

The following basic restrictions will be imposed on an ensemble code:

(a) No two messages will consist of identical arrangements of coding digits.
(b) The message codes will be constructed in such a way that no additional indication is necessary to specify where a message code begins and ends once the starting point of a sequence of messages
Audio

Digital Representation
Audio (Sound): continuous signal (wave form) in time
1D function f(x)
2D function $f(x,y)$

Sampling: Discretization in $x$ and $y$
Video

Video is a sequence of images in time
Geometry Data: Meshes

- Points
- Connectivity