Image Compression and Watermarking

Image Processing
CSE 166
Lecture 14
Announcements

• Assignment 4 is due May 22, 11:59 PM
• Reading
  – Chapter 8: Image Compression and Watermarking
    • Sections 8.1, 8.9, 8.10, and 8.12
Data compression

- Data redundancy

\[ R = 1 - \frac{1}{C} \]

where compression ratio

\[ C = \frac{b}{b'} \]

where

\( b \) and \( b' \) are the number of bits in two different representations of the same information.
Data redundancy in images

- **Coding redundancy**: Does not need all 8 bits.
- **Spatial redundancy**: Information is unnecessarily replicated.
- **Irrelevant information**: Information is not useful.
Image information

• Entropy

\[
\tilde{H} = - \sum_{k=0}^{L-1} p_r(r_k) \log_2(p_r(r_k))
\]

where

- \( L \) is the number of intensity or gray levels
- \( r_k \) is input image intensity or gray level value \( k \)
- \( p_r(r_k) \) is normalized histogram of input image

- It is not possible to encode input image with fewer than \( \tilde{H} \) bits/pixel
Fidelity criteria, objective (quantitative)

• Total error

\[
\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} \left( \hat{f}(x, y) - f(x, y) \right)
\]

• Root-mean-square error

\[
e_{\text{rms}} = \left( \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} \left( \hat{f}(x, y) - f(x, y) \right)^2 \right)^{1/2}
\]

• Mean-square signal to noise ratio (SNR)

\[
\text{SNR}_{\text{ms}} = \frac{\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} \hat{f}(x, y)^2}{\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} \left( \hat{f}(x, y) - f(x, y) \right)^2}
\]
Fidelity criteria, subjective (qualitative)

<table>
<thead>
<tr>
<th>Value</th>
<th>Rating</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Excellent</td>
<td>An image of extremely high quality, as good as you could desire.</td>
</tr>
<tr>
<td>2</td>
<td>Fine</td>
<td>An image of high quality, providing enjoyable viewing. Interference is not objectionable.</td>
</tr>
<tr>
<td>3</td>
<td>Passable</td>
<td>An image of acceptable quality. Interference is not objectionable.</td>
</tr>
<tr>
<td>4</td>
<td>Marginal</td>
<td>An image of poor quality; you wish you could improve it. Interference is somewhat objectionable.</td>
</tr>
<tr>
<td>5</td>
<td>Inferior</td>
<td>A very poor image, but you could watch it. Objectionable interference is definitely present.</td>
</tr>
<tr>
<td>6</td>
<td>Unusable</td>
<td>An image so bad that you could not watch it.</td>
</tr>
</tbody>
</table>
Approximations

Objective (quantitative) quality
rms error (in intensity levels)

5.17
15.67
14.17

(a) is better than (b).
(b) is better than (c)

Subjective (qualitative) quality, relative

Lower is better
Compression system

Encoder

$\hat{f}(x, y)$ or $\hat{f}(x, y, t)$

Decoder
Compression methods

- Huffman coding
- Golomb coding
- Arithmetic coding
- Lempel-Ziv-Welch (LZW) coding
- Run-length coding
- Symbol-based coding
- Bit-plane coding
- Block transform coding
- Predictive coding
- Wavelet coding
Symbol-based coding

(0,2) (3,10) ...

<table>
<thead>
<tr>
<th>Token</th>
<th>Symbol</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>![Symbol Image]</td>
</tr>
<tr>
<td>1</td>
<td>![Symbol Image]</td>
</tr>
<tr>
<td>2</td>
<td>![Symbol Image]</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Triplet</th>
</tr>
</thead>
<tbody>
<tr>
<td>(0, 2, 0)</td>
</tr>
<tr>
<td>(3, 10, 1)</td>
</tr>
<tr>
<td>(3, 18, 2)</td>
</tr>
<tr>
<td>(3, 26, 1)</td>
</tr>
<tr>
<td>(3, 34, 2)</td>
</tr>
<tr>
<td>(3, 42, 1)</td>
</tr>
</tbody>
</table>
Block-transform coding

**Encoder**

1. **Input image** $(M \times N)$
2. Contract $n \times n$ subimage
3. Forward transform
4. Quantizer
5. Symbol encoder
6. Compressed image

**Decoder**

1. Compressed image
2. Symbol decoder
3. Inverse transform
4. Merge $n \times n$ subimage
5. Decompressed image
Block-transform coding

• Example: discrete cosine transform

\[ T(u, v) = \sum_{x=0}^{n-1} \sum_{y=0}^{n-1} g(x, y) \alpha(u) \alpha(v) \cos \left( \frac{(2x + 1)u\pi}{2n} \right) \cos \left( \frac{(2y + 1)v\pi}{2n} \right) \]

where

\[ \alpha(u) = \begin{cases} \sqrt{\frac{1}{n}} & \text{if } u = 0 \\ \sqrt{\frac{2}{n}} & \text{otherwise} \end{cases} \]

and \[ \alpha(v) = \begin{cases} \sqrt{\frac{1}{n}} & \text{if } v = 0 \\ \sqrt{\frac{2}{n}} & \text{otherwise} \end{cases} \]

• Inverse

\[ g(x, y) = \sum_{u=0}^{n-1} \sum_{v=0}^{n-1} T(u, v) \alpha(u) \alpha(v) \cos \left( \frac{(2x + 1)u\pi}{2n} \right) \cos \left( \frac{(2y + 1)v\pi}{2n} \right) \]
Block-transform coding

4x4 subimages (4x4 basis images)

Walsh-Hadamard transform

Discrete cosine transform
Block-transform coding

- **8x8 subimages**
- **Retain 32 largest coefficients**
- **Error image**

Fourier transform

Walsh-Hadamard transform

Cosine transform

- **Error image**
  - 8x8 subimages
  - Retain 32 largest coefficients
  - Error image

- **rms error**
  - Fourier transform: 2.32
  - Walsh-Hadamard transform: 1.78
  - Cosine transform: 1.13

Lower is better
Block-transform coding

Reconstruction error versus subimage size

DCT subimage size: 2x2 4x4 8x8
JPEG uses block DCT-based coding

Compression reconstruction  Scaled error image  Zoomed compression reconstruction

25:1

Compression ratio

52:1
Predictive coding model

Encoder

Decoder
Predictive coding

Example: previous pixel coding

Input image

Prediction error image

Histograms

Std. dev. = 45.60
Entropy = 7.25

Std. dev. = 15.58
Entropy = 3.99
Wavelet coding

Encoder

Input image → Wavelet transform → Quantizer → Symbol encoder → Compressed image

Decoder

Compressed image → Symbol decoder → Inverse wavelet transform → Decompressed image
Wavelet coding

Detail coefficients below 25 are truncated to zero

<table>
<thead>
<tr>
<th>Decomposition Level (Scales or Filter Bank Iterations)</th>
<th>Approximation Coefficient Image</th>
<th>Truncated Coefficients (%)</th>
<th>Reconstruction Error (rms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$256 \times 256$</td>
<td>74.7%</td>
<td>3.27</td>
</tr>
<tr>
<td>2</td>
<td>$128 \times 128$</td>
<td>91.7%</td>
<td>4.23</td>
</tr>
<tr>
<td>3</td>
<td>$64 \times 64$</td>
<td>95.1%</td>
<td>4.54</td>
</tr>
<tr>
<td>4</td>
<td>$32 \times 32$</td>
<td>95.6%</td>
<td>4.61</td>
</tr>
<tr>
<td>5</td>
<td>$16 \times 16$</td>
<td>95.5%</td>
<td>4.63</td>
</tr>
</tbody>
</table>
JPEG-2000 uses wavelet-based coding

Compression reconstruction  Scaled error image  Zoomed compression reconstruction

25:1

Compression ratio

52:1
JPEG-2000 uses wavelet-based coding

Compression ratio

- 75:1
- 105:1

Zoomed compression reconstruction

Scaled error image

Compression reconstruction
Image watermarking

• Visible watermarks
• Invisible watermarks
Visible watermark

\[ f_w = (1 - \alpha)f + \alpha w \]

Watermarked image

Original image minus watermark
Invisible image watermarking system

Encoder

Decoder
Invisible watermark

Example: watermarking using two least significant bits

- Original image
- JPEG compressed
- Extracted watermark
- Two least significant bits
- Fragile invisible watermark
Invisible watermark

Example: DCT-based watermarking

Watermarked images

Extracted robust invisible watermark
Next Lecture

• Morphological image processing

• Reading

  – Chapter 9: Morphological image processing

    • Sections 9.1, 9.2, 9.3, and 9.5 (through subsection connected components)