Wavelet Based Image Compression

Ahmed Abu-Hajar, Ph.D.
March 10, 2011
University of Massachusetts, Lowell.
Outline

• Introduce Digital Image
• Objective of Image Compression
• Basic Idea of Compression
• Discrete Cosine Transform (DCT)
• DCT-Based Image Compression
• Basic Idea of Entropy Coding
• Wavelet Transform
• Basic Idea of Wavelet Image Compression
• Our developed Image Compression Algorithm
• Results & Conclusion
• Summary.
What is an Image?

- **Recall**: 1-D signal $f(t)$.

- **Image** is a 2-D signal
  - $I(x,y)$ indicates the intensity of the reflected light at each spatial location $(x,y)$
  - *Primary colors*: Red, Blue, Green
Digitizing an Image

- **Recall**: 1-D analog signal that is sampled.
  \[ f(n) = \sum_{m=0}^{\infty} a_m \delta(n - m) \]
  - Rounded to finite \( n \)-bit representation
- **Digital Image**: An analog image is sampled at each location with a rectangle called pixel

\[ I(n, m) = \sum_{l=0}^{L} \sum_{m=0}^{M} d_{l,k} \delta(n - l, m - k) \]
Why Image Compression?

• Uncompressed images require *large bandwidth*
  – Example: BW of STD TV
    \[ \text{BW} = 720 \times 480 \times 3 \times 8 \times 30 = 248.8 \text{ Mbps} \]

• *Solution:* Image Compression
  – Reduce the amount of data required to transmit or store the image without degrading from the quality of the image below an acceptable level.
  – The compressed image is an approximation of the actual image

\[
\hat{I}(n,m) \approx I(n,m)
\]

\[
\hat{I}(n,m) = I(n,m) + e(n,m)
\]

• Where:
  – \( \hat{I}(n,m) \) is the compressed image
  – \( e(n,m) \) is the noise due to compression
Observation About Natural Images

- Frequency Content of 1-D Signals
- Images are locally smooth
  - Most of the energy (information) is located in low frequency
- The high frequency content
  - Contains the details (edges, textures,....)
Obtaining the Frequency Content

– Frequency content of a signal is obtained by DFT (FFT)

– 1-D

\[
F(k) = \sum_{n=0}^{N-1} f(n) e^{-j\omega_k kn}
\]

\[
\omega_k = \frac{2\pi}{N}
\]

– 2-D

\[
F(u, v) = \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} I(n, m) e^{-j(\omega_u un + \omega_v vn)}
\]

\[
\omega_u = \frac{2\pi}{N}; \quad \omega_v = \frac{2\pi}{V}
\]
The Basic Idea Behind Image Compression

Convert the image into the frequency domain using DFT or (FFT)

• Truncate the high frequency coefficients
• You will end up with many high frequency coefficients to be zero

• **Disadvantages:**
  – Each coefficient is complex value
  – we may truncate important high frequency values.
  – **Solution:** use Discrete Cosine Transform (DCT)
Discrete Cosine Transform (DCT)

- Basic Idea:
  \[ e^{jx} + e^{-jx} = 2\cos(x) \]
  Let \( f(n) \) be a real value signal

- Let's also add \( f(-n) \) to the signal
  - We end up with a symmetric signal

- Take FFT of the combined signals

\[
F(k) = \sum_{n=-\frac{N-1}{2}}^{\frac{N-1}{2}} f(n) e^{-j\omega_k kn}
\]

- The result is DCT

\[
F(k) = \sum_{n=0}^{N-1} f(n) \cos(\omega_k kn)
\]
Discrete Cosine Transform DCT

- 2-D DCT (8x8) Block
  \[
  F(u,v) = \frac{C(u)c(v)}{4} \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} I(n,m) \cos \frac{(2n+1)u\pi}{16} \cos \frac{(2m+1)v\pi}{16}
  \]
  \[
  C(x) = \begin{cases} 
  \frac{1}{\sqrt{2}} & x = 0 \\
  1 & x \neq 0
  \end{cases}
  \]

- Advantage:
  - The high energy is compacted in the low frequency coefficients.

- Use 8x8 blocks
  - Uses 8x8 matrixes
  - Simpler hardware

\[
T = \begin{bmatrix}
6.1917 & -0.8411 & 1.2418 & 0.1492 & 0.1683 & 0.2742 & -0.0724 & 0.0661 \\
0.2206 & 0.0214 & 0.4603 & 0.3947 & -0.7846 & -0.8991 & 0.1001 & -0.2564 \\
1.0423 & 0.2214 & -1.0017 & -0.2720 & 0.9789 & -0.1962 & 0.2801 & 0.4713 \\
-0.2340 & -0.0392 & -0.2617 & -0.2886 & 0.8361 & 0.3501 & -0.1438 & 0.8550 \\
0.2750 & 0.0226 & 0.1220 & 0.2183 & -0.2638 & -0.0742 & -0.2042 & -0.5906 \\
0.0853 & 0.0428 & -0.4721 & -0.2096 & 0.4746 & 0.2875 & -0.0284 & -0.1311 \\
0.3169 & 0.0541 & -0.1033 & -0.0226 & -0.0066 & 0.1017 & -0.1850 & -0.1600 \\
-0.2970 & -0.0627 & 0.1909 & 0.0644 & -0.1130 & -0.1081 & 0.1887 & 0.1444
\end{bmatrix}
\]
Basic DCT Image Compression

Transform() → Quantize() & Sort() → Entropy Coding

Entropy Decoding → Inverse Quantize() & Sort() → Inverse Transform()
Basic DCT Image Compression

- Entropy Coding: The minimum bits required to code a sequence of symbols (values)
- The most probable values are coded with the least bit size.
- Example: let's say we have four symbols A, B, C, and D.
  - Fixed length symbols will code each symbol with two bits.
    - AAABAAABABAC (2 bits X 10 symbols = 20 bits)
      Assign A = 1, B = 01, C = 001.
      11101110111001 = 14 bits.
- Image Decompression (Decoding)
  - Mimic the opposite
Wavelet Transform

• **Basic Idea:** use subband Coding
  – **Analysis Side:** Decomposes the signal into two bands
    • low frequency & High frequency.
    • The same number of coefficients as the number of the sample original signal.
  – **Synthesis Side:** Forces the condition of perfectly reconstruct the original signal.
  – DSP & Compression is executed before the synthesis Side
2-D Wavelet Transform of Images

- Steps for evaluating Wavelet transform
  - Decompose each row into two subbands (L & H)
  - Decompose each column into two subbands (LL, LH, HL, HH)
  - Repeat the process on the LL level. (here we have three Levels.
  - Image Algorithms have five levels.
Wavelet Zero Tree
Bit-Plane Representation of Wavelet Coefficients

- Each coefficient is represented using $n$-bit and a sign bit ($SB_{n-1} B_{n-2} ... B_1 B_0$)
- Bit Plane: only one bit is placed on the tree for all the coefficients.
- Progressive Coding: each coefficient is coded by progressively refine it value from the most significant bit to the least significant bit.
Wavelet Based Image Compression
Set Partition in Hierarchical Tree (SPIHT)

- For each bit plane
  - For each subband
    - Start from the top level to the least level.
      - If the 2x2 node or any of its descendants has a “1” (significant bit) code the four bits of the at the node.
      - If the coefficient codes the first “1” bit code its sign with it.
  - Stop when
    - bandwidth requirement is met
    - or when PSNR is met
    - or when all bits are coded (lossless)
Algorithm of (P-SPIHT)

1. Image
2. DWT
3. Bit Plane
4. \( P(B_1 = 1) \)
   - \( P_1 < 0.2 \)
   - \( 0.2 < P_1 < 0.3 \)
   - \( P_1 > 0.3 \)
5. Mode 1
6. Mode 2
7. Mode 3
8. Sort Data
9. Organize Stream
10. End
11. Bit Stream
Results for Lossless Compression

<table>
<thead>
<tr>
<th>Image</th>
<th>SPIHT</th>
<th>P-SPIHT_1</th>
<th>P-SPIHT_2</th>
<th>P-SPIHT_3</th>
<th>P-SPIHT_4</th>
<th>JPEG2000</th>
</tr>
</thead>
<tbody>
<tr>
<td>8-bit Set</td>
<td>5.2858</td>
<td>4.8752</td>
<td>4.792</td>
<td>4.775</td>
<td>4.7454</td>
<td>4.74196</td>
</tr>
</tbody>
</table>

– 16 bits superior to SPIHT & JPEG2000
Results for Lossy Compression vs. Quality

<table>
<thead>
<tr>
<th>Rate</th>
<th>SPIHT</th>
<th>P-SPIHT_2</th>
<th>JPEG2000</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.0</td>
<td>39.108267</td>
<td>39.841256</td>
<td>40.869171</td>
</tr>
<tr>
<td>1.0</td>
<td>34.868311</td>
<td>35.459456</td>
<td>36.316183</td>
</tr>
<tr>
<td>0.5</td>
<td>31.716822</td>
<td>32.362089</td>
<td>32.848599</td>
</tr>
<tr>
<td>0.2</td>
<td>28.102156</td>
<td>28.487033</td>
<td>29.090472</td>
</tr>
<tr>
<td>0.1</td>
<td>26.066144</td>
<td>26.350722</td>
<td>26.730983</td>
</tr>
</tbody>
</table>

- 8-bit resolution superior to SPIHT comparable to JPEG2000
Results for Lossy Compression vs. Quality

<table>
<thead>
<tr>
<th>Rate</th>
<th>SPIHT</th>
<th>P-SPIHT_2</th>
<th>JPEG2000</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>69.18103</td>
<td>73.82173</td>
<td>-</td>
</tr>
<tr>
<td>3</td>
<td>55.96763</td>
<td>59.05263</td>
<td>59.97</td>
</tr>
<tr>
<td>2</td>
<td>53.28655</td>
<td>54.47028</td>
<td>55.68238</td>
</tr>
<tr>
<td>1</td>
<td>49.25195</td>
<td>49.6945</td>
<td>50.73325</td>
</tr>
<tr>
<td>0.5</td>
<td>46.56175</td>
<td>46.6425</td>
<td>47.51543</td>
</tr>
</tbody>
</table>

- 16-bit high resolution superior to SPIHT comparable to JPEG2000
Performance of the Arithmetic Coder

- Results for the arithmetic coder.
  - The efficiency of the arithmetic coder doubles with P-SPIHT.
Summary

- The concept of digital image and the need for image compression was introduced. Then we looked at DCT based and wavelet based compression algorithms. We have shown some results for high resolution images.