CSEP 521  
Applied Algorithms  
Spring 2005  
Lossy Image Compression

Lossy Image Compression Methods

- Scalar quantization (SQ).
- Vector quantization (VQ).
- DCT Compression
  - JPEG
- Wavelet Compression
  - SPIHT
  - UWIC (University of Washington Image Coder)
  - EBCOT (JPEG 2000)

JPEG Standard

- JPEG - Joint Photographic Experts Group
- JPEG 2000 uses wavelet compression.

Barbara

32:1 compression ratio  
25 bits/pixel (8 bits)

Original  JPEG  VQ  Wavelet SPIHT  UWIC

JPEG  VQ
Images and the Eye

- Images are meant to be viewed by the human eye (usually).
- The eye is very good at “interpolation”, that is, the eye can tolerate some distortion. So lossy compression is not necessarily bad. The eye has more acuity for luminance (gray scale) than chrominance (color).
  - Gray scale is more important than color.
  - Compression is usually done in the YUV color coordinates. Y for luminance and U and V for color.
  - U and V should be compressed more than Y
  - This is why we will concentrate on compressing gray scale (8 bits per pixel) images.

Distortion

- Lossy compression: \( x \neq \hat{x} \)
- Measure of distortion is commonly mean squared error (MSE). Assume \( x \) has \( n \) real components (pixels).
\[
\text{MSE} = \frac{1}{n} \sum_{i=1}^{n} (x_i - \hat{x}_i)^2
\]

PSNR

- Peak Signal to Noise Ratio (PSNR) is the standard way to measure fidelity.
\[
\text{PSNR} = 10 \log_{10} \left( \frac{m^2}{\text{MSE}} \right)
\]

where \( m \) is the maximum value of a pixel possible. For gray scale images (8 bits per pixel) \( m = 255 \).

- PSNR is measured in decibels (dB).
  - .5 to 1 dB is said to be a perceptible difference.
  - Decent images start at about 30 dB
Rate-Fidelity Curve

PSNR is not Everything

Wavelet Transform

Wavelet Transformed Barbara (Actual)

Wavelet Transform Compression
**Bit Planes of Coefficients**

Coefficients are normalized between -1 and 1

**Why Wavelet Compression Works**

- Wavelet coefficients are transmitted in bit-plane order.
  - In most significant bit planes most coefficients are 0 so they can be coded efficiently.
  - Only some of the bit planes are transmitted. This is where fidelity is lost when compression is gained.
- Natural progressive transmission

**Rate-Fidelity Curve**

More bit planes of the wavelet transformed image that is sent the higher the fidelity.

**Wavelet Coding Methods**

- EZW - Shapiro, 1993
  - Embedded Zerotree coding.
- SPIHT - Said and Pearlman, 1996
  - Set Partitioning in Hierarchical Trees coding. Also uses “zerotrees”.
- EECOW - Wu, 1997
  - Uses arithmetic coding with context.
- EBCOT – Taubman, 2000
  - Uses arithmetic coding with different context.
- JPEG 2000 – new standard based largely on EBCOT
- GTW – Hong, Ladner 2000
  - Uses group testing which is closely related to Golomb codes
- UWIC - Ladner, Askew, Barney 2003
  - Like GTW but uses arithmetic coding

**Wavelet Transform**

A wavelet transform decomposes the image into a low resolution version and details. The details are typically very small so they can be coded in very few bits.

**One-Dimensional Average Transform (1)**

How do we represent two data points at lower resolution?
One-Dimensional Average Transform

\[ (x+y)/2 = L \]
\[ (y-x)/2 = H \]

Inverse Transform

\[ x = L - H \]
\[ y = L + H \]

Note that the low resolution version and the detail together have the same number of values as the original.

Two Dimensional Transform (1)

Transform each row

Horizontal transform

Transform each column in L and H

3 detail subbands

2 levels of transform gives 7 subbands.

k levels of transform gives \(3k + 1\) subbands.
Two Dimensional Average Transform

Wavelet Transformed Image

Wavelet Transform Details
- Conversion to reals.
  - Convert gray scale to floating point.
  - Convert color to Y U V and then convert each to band to floating point. Compress separately.
- After several levels (3-8) of transform we have a matrix of floating point numbers called the wavelet transformed image (coefficients).

Wavelet Transforms
- Technically wavelet transforms are special kinds of linear transformations. Easiest to think of them as filters.
  - The filters depend only on a constant number of values. (bounded support)
  - Preserve energy (norm of the pixels = norm of the coefficients)
  - Inverse filters also have bounded support.
- Well-known wavelet transforms
  - Haar – like the average but orthogonal to preserve energy. Not used in practice.
  - Daubechies 9/7 – biorthogonal (inverse is not the transpose). Most commonly used in practice.

Haar Filters
- Low pass
  \[ B[i] = \frac{1}{\sqrt{2}} A[2i] + \frac{1}{\sqrt{2}} A[2i+1], \quad 0 \leq i < \frac{n}{2} \]
- High pass
  \[ B[n/2 + i] = -\frac{1}{\sqrt{2}} A[2i] + \frac{1}{\sqrt{2}} A[2i+1], \quad 0 \leq i < \frac{n}{2} \]
- Want the sum of squares of the filter coefficients = 1

Daubechies 9/7 Filters
- Low pass
  \[ B[i] = \sum_{j=0}^{4} h_{2j} A[2i + j], \quad 0 \leq i < \frac{n}{2} \]
- High pass
  \[ B[n/2 + i] = \sum_{j=0}^{4} g_{2j} A[2i + j], \quad 0 \leq i < \frac{n}{2} \]
- Reflection used near boundaries
Linear Time Complexity of 2D Wavelet Transform

- Let \( n \) = number of pixels and let \( b \) be the number of coefficients in the filters.
- One level of transform takes time
  - \( O(bn) \)
- \( k \) levels of transform takes time proportional to
  - \( bn + bn/4 + \ldots + bn/4^{k-1} \leq (4/3)bn \).
- The wavelet transform is linear time when the filters have constant size.
  - The point of wavelets is to use constant size filters unlike many other transforms.

Wavelet Transform

- Encoder
  - Image (pixels)
  - Wavelet transform (coefficients)
  - Wavelet coding
- Decoder
  - Transformed image (approx coefficients)
  - Inverse wavelet transform
  - Distorted image

Wavelet coder transmits wavelet transformed image in bit plane order with the most significant bits first.

Wavelet Coding

- Normalize the coefficients to be between \(-1\) and \(1\)
- Transmit one bit-plane at a time
- For each bit-plane
  - Significance pass: Find the newly significant coefficients, transmit their signs.
  - Refinement pass: transmit the bits of the known significant coefficients.

Significant Coefficients

- Significance & Refinement Passes
  - Code a bit-plane in two passes
    - Significance pass
      - codes previously insignificant coefficients
      - also codes sign bit
    - Refinement pass
      - refines values for previously significant coefficients
  - Main idea:
    - Significance-pass bits likely to be 0;
    - Refinement-pass bit are not

Coefficient List

- \[ \begin{array}{c|c}
  x & value \\
  \hline
  1 & 101010110 \\
  2 & 101110110 \\
  3 & 100110101 \\
  4 & 100010101 \\
  5 & 100001110 \\
  6 & 1000000011 \\
  7 & 1001100011 \\
  8 & 1001110101 \\
  9 & 1001110101 \\
  10 & 10001000101 \\
\end{array} \]
Lossy Image Compression

### Lecture 7 - Lossy Image Compression

#### UWIC

- A simple image coder based on
  - Bit-plane coding
    - Significance pass
    - Refinement pass
  - Arithmetic coding
  - Careful selection of contexts based on statistical studies
  - Priority queue for selecting contexts to code
- Implemented by undergraduates Amanda Askew and Dane Barney in Summer 2003.

#### UWIC Block Diagram

- Encoder
  - Image pixels
  - Wavelet transform
  - Transformed image coefficients
  - Subtract LL Avg
  - Divide into bit-planes
  - Bit plane encoding using AC
  - Bit stream

- Decoder
  - Bit plane decoding using AC
  - Recombine bit-planes
  - Add LL Avg
  - Inverse wavelet transform
  - Restored image

#### Arithmetic Coding in UWIC

- Performed on each individual bit plane.
  - Alphabet is \( \Sigma = \{0, 1\} \)
- Uses integer implementation with 32-bit integers. (Initialize \( L = 0, R = 2^{32} - 1 \))
- Uses scaling and adaptation.
- Uses contexts based on statistical studies.

---

**Table:**

<table>
<thead>
<tr>
<th>Bit Planes</th>
<th>bpp</th>
<th>Ratio</th>
<th>PSNR</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 – 7</td>
<td>.303</td>
<td>26 : 1</td>
<td>28.7</td>
</tr>
<tr>
<td>1 – 8</td>
<td>.619</td>
<td>13 : 1</td>
<td>32.9</td>
</tr>
<tr>
<td>1 – 9</td>
<td>1.116</td>
<td>7 : 1</td>
<td>37.5</td>
</tr>
</tbody>
</table>
Coding the Bit-Planes

- Code most significant bit-planes first
- Significance pass for a bit-plane
  - First code those coefficients that were insignificant in the previous bit-plane.
  - Code coefficients most likely to be significant first (priority queue).
  - If a coefficient becomes significant then code its sign.
- Refinement pass for a bit-plane
  - Code the refinement bit for each coefficient that is significant in a previous bit-plane.

Contexts (per bit plane)

- Significance pass contexts:
  - Contexts based on
    - Subband level
    - Number of significant neighbors
  - Sign context
- Refinement contexts
  - 1st refinement bit has a context
  - All other refinement bits have one context
- Context Principles
  - Bits in a given context have a probability distribution
  - Bits in different contexts have different probability distributions

Subband Level

- Image is divided into subbands until LL band (subband level 0) is less than 16x16
- Barbara image has 7 subband levels

Statistics for Subband Levels

<table>
<thead>
<tr>
<th>Subband Level</th>
<th># significant</th>
<th># insignificant</th>
<th>% significant</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>144</td>
<td>364</td>
<td>28.3%</td>
</tr>
<tr>
<td>1</td>
<td>272</td>
<td>1048</td>
<td>20.6%</td>
</tr>
<tr>
<td>2</td>
<td>848</td>
<td>4592</td>
<td>15.6%</td>
</tr>
<tr>
<td>3</td>
<td>3134</td>
<td>23568</td>
<td>11.7%</td>
</tr>
<tr>
<td>4</td>
<td>13268</td>
<td>113886</td>
<td>9.7%</td>
</tr>
<tr>
<td>5</td>
<td>48282</td>
<td>504633</td>
<td>8.7%</td>
</tr>
<tr>
<td>6</td>
<td>190000</td>
<td>2269040</td>
<td>7.8%</td>
</tr>
</tbody>
</table>

Significant Neighbor Metric

- Count # of significant neighbors
  - children count for at most 1
  - 0,1,2,3+

Number of Significant Neighbors

<table>
<thead>
<tr>
<th>Significant neighbors</th>
<th># significant</th>
<th># insignificant</th>
<th>% significant</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>4849</td>
<td>2252468</td>
<td>2.2%</td>
</tr>
<tr>
<td>1</td>
<td>13319</td>
<td>210695</td>
<td>5.9%</td>
</tr>
<tr>
<td>2</td>
<td>22276</td>
<td>104252</td>
<td>17.6%</td>
</tr>
<tr>
<td>3</td>
<td>30206</td>
<td>78899</td>
<td>27.7%</td>
</tr>
<tr>
<td>4</td>
<td>33244</td>
<td>55841</td>
<td>37.3%</td>
</tr>
<tr>
<td>5</td>
<td>27354</td>
<td>39189</td>
<td>41.1%</td>
</tr>
<tr>
<td>6</td>
<td>36482</td>
<td>44225</td>
<td>45.2%</td>
</tr>
<tr>
<td>7</td>
<td>87566</td>
<td>91760</td>
<td>48.8%</td>
</tr>
</tbody>
</table>
### Lecture 7 - Lossy Image Compression

#### Refinement Bit Context Statistics

<table>
<thead>
<tr>
<th></th>
<th>0's</th>
<th>1's</th>
<th>% 0's</th>
</tr>
</thead>
<tbody>
<tr>
<td>2nd Refinement Bits</td>
<td>146,293</td>
<td>100,521</td>
<td>59.3%</td>
</tr>
<tr>
<td>Other Refinement Bits</td>
<td>475,941</td>
<td>433,982</td>
<td>53.3%</td>
</tr>
<tr>
<td>Sign Bits</td>
<td>128,145</td>
<td>130,100</td>
<td>49.6%</td>
</tr>
</tbody>
</table>

- Barbara at 2bpp: 2nd Refinement bit % 0’s = 65.8%

#### Context Details

- Significance pass contexts per bit-plane:
  - Max neighbors * num subband levels contexts
  - For Barbara: contexts for sig neighbor counts of 0 - 3 and subband levels of 0-6 = 477 contexts
  - Index of a context:
    - Max neighbors * subband level + num sig neighbors
    - Example num sig neighbors = 2, subband level = 3, index = 4 * 3 * 2 = 24
  - Sign context
    - 1 contexts
  - 2 Refinement contexts
    - 1st refinement bit is always 1 not transmitted
    - 2nd refinement bit has a context
  - All other refinement bits have a context
  - Number of contexts per bit-plane for Barbara = 28 + 1 + 2 = 31

#### Priority Queue

- Used in significance pass to decide which coefficient to code next
  - Goal code coefficients most likely to become significant
- All non-empty contexts are kept in a max heap
- Priority is determined by:
  - # sig coefficients coded / total coefficients coded

#### Reconstruction of Coefficients

- Coefficients are decoded to a certain number of bit planes
  - .101110XXXXX. What should X’s be?
    - .101110000... < .101110XXXXX < .101110111...< .10111010000 is half-way
  - Handled the same as SPIHT and GTW
    - if coefficient is still insignificant, do no interpolation
    - if newly significant, add on .38 to scale
    - if significant, add on .5 to scale

#### Original Barbara Image

![Original Barbara Image](image)

#### Barbara at .5 bpp (PSNR = 31.68)

![Barbara at .5 bpp](image)
Barbara at .25 bpp (PSNR = 27.75)

Barbara at .1 bpp (PSNR = 24.53)

Results
Compression of Barbara

Results
Compression of Lena

Results
Compression of RoughWall

UWIC Notes
• UWIC competitive with JPEG 2000, SPIHT-AC, and GTW.
• Developed in Java from scratch by two undergraduates in 2 months.