Wavelet Scalable Video Codec
Part 1: image compression by JPEG2000

Aline Roumy
aline.roumy@inria.fr

May 2011
Motivation for Video Compression

- **Digital** video studio standard ITU-R Rec. 601

<table>
<thead>
<tr>
<th></th>
<th>Y luminance</th>
<th>Cb Chrominance</th>
<th>Cr Chrominance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sampling Rate</td>
<td>13.5 MHz</td>
<td>6.75 MHz</td>
<td>6.75 MHz</td>
</tr>
<tr>
<td>Quantization</td>
<td>8 bit</td>
<td>8 bit</td>
<td>8 bit</td>
</tr>
<tr>
<td>Raw bit rate</td>
<td></td>
<td>216 Mbps</td>
<td></td>
</tr>
</tbody>
</table>

Human eyes are less sensitive to the chrominance than to the luminance
Plan

1. Still-image compression
   1. Typical coding scheme: transform coding
   2. Information theoretical justification for the transform coding scheme

2. From Transform coding to JPEG-2000
   Wavelet based still-image compression

3. From motion compensation to WSVC (Ronan)
Typical Image Coder and Decoder

- **Transform** $T(x)$ usually invertible
- **Quantization** $Q(y)$ not invertible, introduces distortion
- **Lossless** encoder $C(q)$
Lossless source coding

Random process (source) \( \{X_n\} \) Example: model for the pixels

Minimum average length (in bits) to represent a source sample:

- **Discrete-valued i.i.d.** source [Shannon]
  \[ H(X) \triangleq H(X_1) \] bits/source sample

- **finite-valued** stationary **ergodic** source [Shannon-McMillan-Breiman]
  \[ H(X) \triangleq \lim_{n \to +\infty} \frac{1}{n} H(X_1, \ldots, X_n) \] bits/source sample
Entropy coding for i.i.d. source

The way to achieve $H(X)$ is entropy coding, aka Variable length code VLC

Example: Huffman

For any distribution of $X$, a VLC code can be found, whose rate $R$ satisfies

$$H(X) \leq R < H(X) + 1$$

very inefficient for $H(X) << 1$ bit/symbol

Remedy: combine $m$ successive symbols to a new “block-symbol”

$$H(X) \leq R < H(X) + 1/m$$

- exponentially growing alphabet size

Example: block-Huffman, arithmetic
Sufficiency of binary encoding

• Consider the entropy coding of the r.v. $X \in \{0, 1, ..., 2^{K-1}\}$

Example: $2^8=256$ gray levels

• Assume that $X$ is equivalent to a r.vector

• No loss by **conditionally** encoding bits (in any order)

\[ H(X) = H(B) \]
\[ H(X) = H(B_0) + H(B_1|B_0) + ... + H(B_{K-1}|B_0, ..., B_{K-2}) \]

Definition: a **context** is the set of r.v. taken to condition in the entropy encoder.

Example: $B_0, ..., B_{K-2}$ is the context for $B_{K-1}$
What if the process is not i.i.d.?
Joint entropy and statistical dependence

Theorem (Independence bound on joint entropy)

\[ H(X_1, \ldots, X_n) \leq H(X_1) + \ldots + H(X_n) \]

equality iff \( X_1, \ldots, X_n \) are statistically independent

- Exploiting statistical dependence lowers bit-rate \( \text{idea 1} \)
- Statistically independent components can be coded \textit{separately} without loss of efficiency \( \text{idea 2} \)
idea 1... toward context adaptive coding

Exploiting statistical dependence lowers bit-rate

Break image into blocks
• combine $m$ neighbor pixels to a new “block-pixel” (≈ seen before)
• Block-by-block entropy coding
  • Statistical dependencies across block boundaries neglected

Sliding window across image
• Interpret image as source with memory
• Perform conditional entropy coding (spatial context)
  
Context adaptive coding [JPEG-2000]
idea 2... toward transform coding

Independent symbols can be coded *separately*

Transform $x$ into an equivalent array of coefficients $y$

$$H(x_1, \ldots, x_n) = H(y_1, \ldots, y_n) \leq H(y_1) + \ldots + H(y_n)$$

invertible transform

same sampling rate

Min $1^{st}$ order entropy

$\approx \min$ variance

transform coding [JPEG-2000]

Goal = transform allows to use *scalar* entropy coding
Summary on lossless compression

• **ergodic** stationary process:
  • transform and decorrelate the samples:
    make the entropy coding task easier (scalar is optimal if iid process)

• exploit **residual** statistical context = spatial neighborhood
  limited context size due to transform

\[
\text{image } x \quad \xrightarrow{\text{Transform}} \quad y = T(x) \quad \xrightarrow{\text{coefficients}} \quad y \quad \xrightarrow{\text{Entropy coder}} \quad c = C(y)
\]
Lossy compression

Lower the bit-rate $R$ by introducing some (acceptable) distortion $D$ of $X$.

Rate Distortion theory gives the optimal curve for a given distribution… in theory…
Scalar quantization
Vector quantization
Uniform vector quantization
High resolution analysis

Uniform vector Quantization + entropy coding = optimal
(dimension→infinity)

Uniform scalar Quantization + scalar entropy coding = small loss
(1.53 dB Mean square error distortion)
High resolution regime

- Valid for moderate to high rate
- Valid for all rates by adding a dither (uniform noise in Quantization cell)

[Zamir92]
Typical Image Coder and Decoder

- **Transform** $T(x)$ to decorrelate the pixels (to allow scalar $C$)
- **Quantization** $Q(y)$: scalar uniform
- **Lossless** encoder $C(q)$: context adaptive: bit+spatial (to deal with remaining redundancy)
2. JPEG 2000
Features of JPEG2000

ISO/IEC JTC1/SC29/WG1


What's new?

• Improved compression efficiency (vs JPEG);
• Multiple resolution representation;
• Progressive decoding;
• Scalability;
• Error resilience.
JPEG scheme?

- Transform
  - $y = T(x)$
  - $q = Q(y)$
  - $c = C(q)$

- Inverse transform
  - $T^{-1}$
  - $Q^{-1}$
  - $C^{-1}$

- Decoder

- Encoder bistream $c$

- Image $x$
  - Coefficients $y$
  - Indices $q$
  - Reconstructed image $x$
2-d Discrete Wavelet Transform

- Optimal orthonormal transform = KLT. Not used: source statistics dependent
- DCT (JPEG) blocking artefacts ⇒ Subband transform
- Spatial scalability ⇒ DWT
Are the DWT-coefficients independent?

Not really… need for context adaptive entropy coding to remove residual dependencies.
Partitioning of image subbands into code-blocks
JPEG scheme?

- **Quantizer?** Scalar uniform?
Embedded deadzone scalar uniform quantizer
JPEG scheme?

- **Entropy coder?**

  Bit-plane coding + R-D code-block truncation = EBCOT

  Embedded Block Coding with Optimized Truncation
Bit-plane coding

bitplane = a set of bits having the same position. 1 bit/coefficient
Achievable rates through bitplane encoding

Bitplane encoding leads too few Rate levels

Generate additional feasible truncation points for each block
Fractional bit-planes

coefficient = \text{sign} / \begin{array}{c} 000 \text{ insignificant} \, 1 \, 010011 \text{ significant} \end{array}

Significance Propagation Pass
insignificant coefficient with 1 significant neighbor

Magnitude Refinement Pass
significant coefficient

Cleanup Pass
remaining coefficients
Entropy coding

**Independent** entropy coding for each pass

Context-dimension reduction from $2^8=256$ to 18 contexts by using “primitives” = almost sufficient statistics of the neighborhood.
Optimal bit-stream truncation

PCRD, Post Compression Rate-Distortion

Minimize Distortion (of the image) under a global rate constraint
JPEG2000 compression results

Original Carol Image (512 x 512 Pixels, 24-Bit RGB, Size 786K)
JPEG2000 compression results

75:1, 10.6 Kbyte
JPEG2000 compression results

150:1, 5.3 Kbyte
JPEG2000 compression results

300:1, 5.3 Kbyte
Comparison JPEG vs. JPEG2000

Lenna, 256x256 RGB
Baseline JPEG: 4572 bytes

Lenna, 256x256 RGB
JPEG-2000: 4572 bytes
Sufficiency of binary encoding

- Consider a r.v. $X \in \{0, 1, ..., 2^K-1\}$ equivalent to a r.v. $B = \begin{pmatrix} B_0 \\ B_1 \\ \vdots \\ B_{K-1} \end{pmatrix}$
- No loss by conditionally encoding bits (in any order)
  \[
  H(X) = H(B) \\
  H(X) = H(B_0) + H(B_1|B_0) + ... + H(B_{K-1}|B_0, ..., B_{K-2})
  \]
- Supply entropy encoder with symbol/probability pairs
  \[
  (b_0, p_0), (b_1, p_1), ..., (b_{K-1}, p_K - 1) \text{ with } p_k = f_{B_k|B_{0:k-1}}(0, b_{0:k-1})
  \]
- Total number of (conditional) probabilities to be estimated per symbol $x$
  \[
  1 + 2 + ... + 2^{K-1} = 2^K - 1 \quad \text{... same as with $K$-ary encoding of $x$}
  \]

Definition: a context is the set of r.v. taken to condition in the entropy encoder.
Fractional bit-planes

coefficient = 000 insignificant \text{ 1010011 significant}

Partition the bits in the bitplane in 3 classes and encode them separately

1. “Significance Propagation Pass”
insignificant bits with significant neighbors

2. “Magnitude Refinement Pass”
coefficient that were already significant in the previous bit-plane

3. Cleanup Pass: all remaining (insignificant) coefficients

Context-dimension reduction

by using “primitives” = interpretation of the coefficient