Content Insertion into Compressed Video in the Coding Domain

Nimrod Peleg

http://sipl.technion.ac.il

Tel-Hai Conference on Signal Processing and Multimedia Applications, June 2007
The Staff

- Prof. David Malah
- Yair Moshe
- Naama Hait
- Tamar Shoham – Thanks for the slides!

- 12 Undergraduate students
  Algorithms, software implementations, real-time implementations
Outline

• Content insertion: Motivation
• Classic video compression
• Video-in-video algorithms

• If time allows: H.264 compression novelties
Motivation

– There is a need for content insertion in the communication industry (logos, subtitles, advertisements, etc.)
– The insertion must support logos that change in time as well as in space
– The insertion should be fast and low complexity

Broadcasting Vs. Personal video
Introduction to video compression
Redundancy ⇔ Irrelevancy in video clips

• Redundancy
  ✓ **Spatial**: pixel-to-pixel or spectral correlation
  ✓ **Temporal**: frame-to-frame similarity
  ✓ **Statistical**: non-uniform distribution of data

• Irrelevancy relates to an observer viewing an image

Redundancy + Irrelevancy ⇒ high compression ratio
Spatial Redundancy & Irrelevancy

- What is the value of the **missing pixel**? It is 39.
- How critical is it to correctly reproduce it?
Irrelevancy: CSF

The **Contrast Sensitivity Function** illustrates the **limited perceptual sensitivity** to high spatial frequencies.
Irrelevancy: Visual Masking
Irrelevancy: Visual Masking
Video Compression enablers

Video clips are:
- Spatially redundant
- Temporally redundant
- Statistically redundant

Human eyes are:
- Less sensitive to high spatial frequencies
- Less sensitive to chromatic resolution
- Less sensitive to distortions in “busy” areas
The video coding scheme

Each block removes some redundancy / irrelevancy element
The video coding scheme

We illustrate the encoder,
Although all standards define the decoder.

- In addition, various pre/post processing operations may be applied to the input/decoded frames.
What is “YUV 4:2:0”? 

YUV color representation

![YUV color representation diagram](image)

= 

Chromatic down-sampling

![Chromatic down-sampling diagram](image)

× Represent luminance samples
○ Represent chrominance samples
Blocks and macroblocks

Macroblock

Y

U

V

8x8 pixel blocks
DCT transform

- The **Discrete Cosine Transform** is an energy-preserving, reversible transform.
- For natural images, DCT helps remove local spatial redundancy.

\[
F(u, v) = \frac{2}{n} \cdot C(u) \cdot C(v) \cdot \sum_{k=0}^{n-1} \sum_{l=0}^{n-1} f(k, l) \cdot \cos \left[ \frac{(2k + 1) \cdot u \pi}{2n} \right] \cdot \cos \left[ \frac{(2l + 1) \cdot v \pi}{2n} \right]
\]

\[
f(k, l) = \frac{2}{n} \cdot \sum_{u=0}^{n-1} \sum_{v=0}^{n-1} C(u) \cdot C(v) \cdot F(u, v) \cdot \cos \left[ \frac{(2k + 1) \cdot u \pi}{2n} \right] \cdot \cos \left[ \frac{(2l + 1) \cdot v \pi}{2n} \right]
\]

where:

\[
C(w) = \begin{cases} 
\frac{1}{\sqrt{2}} & \text{for } w = 0 \\
1 & \text{otherwise}
\end{cases}
\]
DCT basis functions

Original image block

DC (flat) basis function

AC basis functions

\[ = F(0,0) \times + F(1,0) \times + F(0,1) \times + \ldots \times + F(7,7) \times \]
Quantization

• A many-to-one mapping.
• Reduces the number of possible signal values at the cost of introducing errors.
• Quantizer step size selection controls the trade off between image quality and bit rate.

Uniform quantizer function:
Motion estimation

- Estimates the displacement of image structures from one frame to another, not necessarily true motion

![Motion estimation diagram]

- Search Window
- Motion Vector
- Matching block
- Reference frame
- Current Frame
- Residual
- Motion Vector (dx, dy)
- SAD example
Motion Estimation - Example

Goal: search for minimum Sum of Absolute Differences

Current block

Reference region

SAD map

Best match
Frame types

- The selection of macroblock and frame types is performed by the encoder rate control module.

<table>
<thead>
<tr>
<th></th>
<th>INTRA MB</th>
<th>INTER MB</th>
<th>SKIPPED MB</th>
</tr>
</thead>
<tbody>
<tr>
<td>I frame</td>
<td>√</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>P/B frame</td>
<td>√</td>
<td>√</td>
<td>√</td>
</tr>
</tbody>
</table>
video coding - summary

- The loss is introduced by the quantizer.
- The top branch removes spatial & statistical redundancies and irrelevancies.
- The bottom branch removes temporal redundancy.
- The decoder performs the inverse operations to receive the reconstructed video sequence.
Content Insertion Algorithms
Content Insertion Algorithm

Video Insertion System
Content Insertion Challenges

• **Segmentation**: Distinguishing between **affected** (‘variable’) areas and **unaffected** (‘constant’) areas.

• Efficient handling of **unaffected areas** (non trivial due to predictions & entropy coding).

• **Seamless content insertion in affected area**.

Important: The “logo” is NOT aligned to the MB grid!
‘Constant’ and ‘Variable’ Blocks

**INTRA:**

Reference frame

Current frame

Reference frame

Current frame

**INTER:** (additional Variable blocks due to motion)

27
Example: - 'constant' and 'variable' blocks map

White – ‘variable’ blocks
Black – ‘Constant’ blocks
Coding Domain Logo Insertion

Encoded frame (without logo)

Partial decoder

DCT coeff.

Coding domain content insertion unit

DCT coeff. (with “logo”)

Partial encoder

Encoded frame (with logo)

Side information (MVs, Quant)
MC-DCT Motivation

MC-DCT: Motion Compensation in the DCT domain

Divide into blocks and macroblocks

Motion Estimation

Frequency Domain

DCT

Quantization

Coefficient predictions

Entropy Coding

input frame (YUV 4:2:0)

output bitstream

“Not natural” in the frequency domain
MC-DCT properties

• Saves the IDCT-DCT operations. 😊

• Pixel and DCT motion compensation results are equivalent.

• MC-DCT complexity is higher but may only be required in affected areas.

• Enables other DCT domain operations such as DCT domain resize. 😊
Pixel motion compensation

Retrieve reference block, pointed to by motion vector, from reference frame:
Retrieval of Un-aligned block

Original blocks

B1

B2

B3

B4

Relevant sub-blocks from each of the four blocks

Windowed & shifted sub-blocks

Desired reference block
Pixel MC: formalization

• A sub-block is obtained from an original block by pre & post multiplication with window/shift matrices, as given by: $S_i = H_i B_i V_i, \quad i = 1, \ldots, 4$

• The windowing/shifting matrices are defined as:

$$H_1 = \begin{bmatrix} 0 & I_h \\ 0 & 0 \end{bmatrix}, \quad V_1 = \begin{bmatrix} 0 & 0 \\ I_w & 0 \end{bmatrix}$$

$$H_2 = \begin{bmatrix} 0 & I_h \\ 0 & 0 \end{bmatrix}, \quad V_2 = \begin{bmatrix} 0 & I_{8-w} \\ 0 & 0 \end{bmatrix}$$

$$H_3 = \begin{bmatrix} 0 & 0 \\ I_{8-h} & 0 \end{bmatrix}, \quad V_3 = \begin{bmatrix} 0 & 0 \\ I_w & 0 \end{bmatrix}$$

$$H_4 = \begin{bmatrix} 0 & 0 \\ I_{8-h} & 0 \end{bmatrix}, \quad V_4 = \begin{bmatrix} 0 & I_{8-w} \\ 0 & 0 \end{bmatrix}$$

• The required block is: $\hat{B} = \sum_{i=1}^{4} S_i$
Pixel MC - example

example of extracting the bottom right 3x2 area from a 4x4 matrix:

\[
\begin{bmatrix}
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 \\
\end{bmatrix}
\begin{bmatrix}
11 & 12 & 13 & 14 \\
21 & 22 & 23 & 24 \\
31 & 32 & 33 & 34 \\
41 & 42 & 43 & 44 \\
\end{bmatrix}
\begin{bmatrix}
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
\end{bmatrix}
= \\
\begin{bmatrix}
21 & 22 & 23 & 24 \\
31 & 32 & 33 & 34 \\
41 & 42 & 43 & 44 \\
0 & 0 & 0 & 0 \\
\end{bmatrix}
\begin{bmatrix}
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
\end{bmatrix}
= \\
\begin{bmatrix}
23 & 24 & 0 & 0 \\
33 & 34 & 0 & 0 \\
43 & 44 & 0 & 0 \\
0 & 0 & 0 & 0 \\
\end{bmatrix}
\]
DCT domain MC

DCT, an orthogonal transform, is distributive to matrix multiplications: \( DCT(AB) = DCT(A)DCT(B) \)

Therefore:

\[
DCT(\hat{B}) = \sum_{i=1}^{4} DCT(H_i B_i V_i) = \sum_{i=1}^{4} DCT(H_i) DCT(B_i) DCT(V_i)
\]

- DCT of the manipulation matrices is performed off-line, and stored for each possible offset.
Performing DCT domain MC

1. Calculate w and h from the motion vector.

2. If the block is aligned (w=h=0), the desired DCT coefficients are immediately available.

3. Otherwise, for:
   - (w = 0) & (h ≠ 0) get S1 and S3,
   - (w ≠ 0) & (h = 0) get S1 and S2,
   - (w ≠ 0) & (h ≠ 0) get S1, S2, S3 and S4.

4. Perform matrix multiplications & summations.
Half-pixel resolution motion vectors require retrieval of an interpolated block from the reference frame.

Horizontal Interpolated pixel
\[ h = \frac{(A+B)}{2} \]

Vertical interpolated pixel
\[ v = \frac{(A+C)}{2} \]

Central interpolated pixel
\[ c = \frac{(A+B+C+D)}{4} \]
The windowing/shifting/interpolation matrices for retrieving a half-pel resolution reference block are of the form:

\[ Hhp_1 = \frac{1}{2} \left\{ \begin{bmatrix} 0 & I_h \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & I_{h-1} \\ 0 & 0 \end{bmatrix} \right\}, \quad Vhp_1 = \frac{1}{2} \left\{ \begin{bmatrix} 0 & 0 \\ I_w & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ I_{w-1} & 0 \end{bmatrix} \right\} \]

Where, \( w \) & \( h \) are \( \text{ceil}() \) of the half pixel motion vectors, (8 possible values).
Half pel resolution MC-DCT

- The required interpolated motion compensated block in the DCT domain is given by:

\[
DCT(\hat{B}_{hp}) = \sum_{i=1}^{4} DCT(H_{hp_i})DCT(B_{i})DCT(V_{hp_i})
\]

- The calculation steps are identical to those of the full-pel case.
Compressed Domain Logo Insertion

Encoded frame (without logo) → Partial decoder → DCT coeff. → Coding domain logo insertion unit → DCT coeff. (with logo) → Partial encoder → Encoded frame (with logo)

Side information (MVs, Quant)
THANK you
H.264 compression novelties
H.264 main innovations - 1

• 4x4 basic block size, with integer transforms (ICT).

• Tools for improving temporal prediction:
  – Multiple reference frame mechanism.
  – Large variety of block size and shapes for motion compensation.
  – 1/4 pixel resolution motion vectors combined with high quality interpolation filters.
  – Advanced prediction of MVs between adjacent blocks.
H.264 main innovations – 2

Spatial prediction

- INTRA block content is predicted from neighbors
- Removes redundancies between adjacent blocks
- Performed in the pixel domain
- MANY different prediction modes supported
4x4 luma prediction modes

Figure 3 4x4 luma prediction modes
INTRA prediction example

Vertical Prediction

Horizontal Prediction

DC (mean) Prediction

Diagonal Down Left Prediction

Lena
INTRA prediction example – cont.

Vertical Prediction Error

Horizontal Prediction Error

DC (mean) Prediction Error

Diagonal Down Left Prediction Error

Lena
H.264 main innovations – 3

- Efficient context-adaptive entropy coding.
- In loop deblocking filter

Figure 4
Performance of the deblocking filter for highly compressed pictures. *Left:* without the deblocking filter. *Right:* with the deblocking filter.
Additional Video in video Challenges in H.264
Challenges - 1

• Advanced spatial predictions complicate affected/unaffected segmentation.
  – INTRA frames: Logo area prorogates via intra prediction to the entire frame.
  – INTER frames: MVs are spatially predicted, therefore local MV changes propagate through the frame.
Challenges - 2

• MC-ICT, the integer transform equivalent to MC-DCT must be developed and evaluated.

• The many motion modes and ¼ pel resolution complicate the MC-ICT and require re-evaluation of its profitability.

• Effect of in-loop deblocking filter must be evaluated.

• Sophisticated context-adaptive entropy coding causes any change to affect the coding of the entire frame.
Video in Video – H.264

- Motion estimation remains the most “expensive” part of encoding.
- Macroblock mode selection requires evaluation of many different modes and consumes a significant part of encoding resources.
- Transform complexity is no longer a video-in-video bottleneck.

⇒ H.264 Partial encoder should operate in the coding domain
Coding Domain Logo Insertion

- **Encoded frame (without logo)**
- **Frame**
- **Frame (with logo)**
- **Partial encoder**
- **Encoded frame (with logo)**

The diagram illustrates the process of coding domain logo insertion. The flow starts with an encoded frame (without logo) from the decoder. This frame is then passed to the logo insertion unit, which adds the logo information. The output is a frame (with logo) that is passed to the partial encoder. The coding domain info (prediction modes, motion modes, MVs, Quant) is included in this process. Finally, the encoded frame (with logo) is produced.
Progress Report

PROGRESS REPORT ON THE WAR IN IRAQ...

I THINK WE'VE GOT THEM ON THE RUN!
Progress Report

• MPEG-2 static logo insertion – completed. (~70-80% reduction in run-time compared to naive solution)

• Static logo insertion into H.264 INTRA Baseline frames – completed. (~50% gain in initial tests on Nokia Baseline software)

• Static logo insertion into H.264 INTER Baseline frames – in progress.

• DSP implementation – in progress.