Video Coding
Outline

• Block-Based Hybrid Video Coding
  • Overview: putting previous lectures together
    • Representation; temporal prediction; spatial prediction; transform coding; quantization; variable bit-rate compression
  • Coding mode selection and rate control
  • Rate-distortion optimization
  • Loop filtering
Key Ideas in Video Compression

• Prediction errors have smaller energy and can be coded with fewer bits
  • Predict new frame from “previous” frames --- Inter prediction
  • Predict current block from previous blocks in the same frame --- Intra prediction

• Prediction error is coded using Transform coding

• When prediction fails, don’t use it!
  • Regions that cannot be predicted well are coded directly

• Work on each macroblock (MB) independently
  • Motion compensation done at the MB level
  • DCT coding of error at the block level (8x8 pixels)
Representation: Macroblock (MB) Structure

4 8x8 Y blocks 1 8x8 Cb blocks 1 8x8 Cr blocks

4:2:0 Color Format

In HEVC: Coding Tree Unit is a generalization of a macroblock (square, up to 64*64 pixels) (more later)
Temporal Compression

• Adjacent frames are similar and changes are due to object or camera motion

• In H.261 to H.264/AVC, motion compensation occurs at the Macroblock (16*16) level

• In H.265/HEVC, motion compensation occurs at the “Prediction Unit” (variable size)
Temporal Compression: Theory vs. Practice (1)

• Theory: $\hat{f}_t = \alpha \hat{f}_{t-1}$ where $\alpha$ is chosen to obtain the best prediction that minimizes the expected error

• Problems with theory:
  • Finding the best $\alpha$ is difficult
  • $\alpha$ changes over time
  • Decoder and encoder need to use the same $\alpha$
  • Implementation complexity: $\alpha$ should be limited to be some function of a power of two
Temporal Compression: Theory vs. Practice (2)

• More practical requirements
  • Every pixel must be predicted
  • Sometimes prediction works well; other times it does not
  • Some pixels are well predicted from a past frame
  • Some pixels are well predicted from a future frame

• Result: block-based motion compensation

\[ \hat{f}_t = (\hat{f}_{t-1} + \hat{f}_{t+1}) / 2 \]

• Signal to decoder which option is used for each block
• Divide by 2 is a simple shift-right
• (Some options for more general weights in later standards)
Group of Pictures
Group of Pictures

Display order: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
Bitstream order: 0 2 3 1 5 4 8 9 7 11 12 10 14 15 13
Group-of-Picture Structure

- I-frames coded without reference to other frames
- P-frames coded with reference to previous frames
- B-frames coded with reference to previous and future frames
  - Requires extra delay!

- Typically, an I-frame every 15 frames (0.5 seconds)
  - Fast random access (AKA channel change)

- Typically, two B frames between each P frame
  - Compromise between compression and delay
Hierarchical Temporal Prediction

In H.264 and beyond, B frames can be used for prediction

Display order: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
Block-Based Temporal Prediction

• No Motion Compensation
  • Works well in stationary regions
    \[ \hat{f}(t,m,n) = f(t-1,m,n) \]

• Uni-directional Motion Compensation
  • Does not work well for uncovered regions by object motion
    \[ \hat{f}(t,m,n) = f(t-1,m-d_x,n-d_y) \]

• Bi-directional Motion Compensation
  • Can handle better uncovered regions
    \[ \hat{f}(t,m,n) = w_b f(t-1,m-d_{b,x},n-d_{b,y}) \]
    \[ + w_f f(t+1,m-d_{f,x},n-d_{f,y}) \]
MPEG-2: Motion Compensation

Past frame

Current frame

All objects except this area have already been sent to decoder in “past frame”
Motion Compensated Prediction (P-Frame)

• Assumes rigid bodies move translationally; uniform illumination; no occlusion, no uncovered objects

• Big win: Improves compression by factor of 5-10
MPEG-2: Motion Compensation

Past frame

Current frame

Future frame

This area can now be predicted using “future frame”
Motion Compensated Prediction (B-Frame)

- Helps when there is occlusion or uncovered objects
- Vector into the future need not be the same as vector into the past
Multiple Reference Frame Temporal Prediction

When multiple references are combined, the best weighting coefficients can be determined using ideas similar to minimal mean square error predictor.

Figure 9. Motion-compensated prediction with multiple reference images. In addition to the motion vector, also an image reference parameter $d_t$ is transmitted.
Temporal Prediction Options

• Predict using one frame or two
• Save this frame for subsequent predictions (or not)
• Some limited ability to use prediction coefficients other than 1 or $\frac{1}{2}$

• Lots of flexibility for frame types to be chosen for best compression, or low delay, or error resilience
Spatial Compression: Theory vs. Practice (1)

• Theory: Karhunen Loeve Transform is best possible block-based transform

• Problems with theory:
  • Finding an accurate model of the source is difficult
  • Model and KLT change over time and in different regions
  • Decoder and encoder need to use same KLT
  • Implementation complexity: a full matrix multiply is necessary to implement KLT

• Practice: Discrete Cosine Transform
  • Also, approximations to DCT and also DST option
Spatial Compression: Theory vs. Practice (2)

• Theory: Larger transform blocks (using more pixels) are more efficient

• Problem with theory:
  • Hard to get an accurate model of the correlation of distant pixels
  • In the limit as the inter-pixel correlation approaches one, the KLT approaches the DCT; however, the inter-pixel correlation of distant pixels is not close to one

• Practice:
  • Small block transforms – usually 8x8 pixels, although in more recent systems we can use 4x4 blocks or 16x16 blocks
  • *There is still correlation between adjacent blocks*
Spatial Prediction

• General idea:
  • A pixel in the new block is predicted from previously coded pixels in the same frame
  • What neighbors? What weighting coefficients?

• Content-adaptive prediction
  • No edges: use all neighbors
  • With edges: use neighbors along the same direction
  • The best possible prediction pattern can be chosen from a set of candidates, similar to search for best matching block for inter-prediction
    • H.264 (and HEVC) have many possible intra-prediction patterns
H.264 Intra-Prediction

HEVC Intra-Prediction

Example: Directional mode 29

Boundary samples from decoded PUs

Current PU
Encoder Block Diagram of a Typical Block-Based Video Coder
(Assuming No Intra Prediction)

Hybrid: both prediction (temporal) and transform (spatial)
Decoder Block Diagram
Macroblock Coding in I-Mode (assuming no intra-prediction)

DCT transform each 8x8 DCT block

Quantize the DCT coefficients with properly chosen quantization matrices

The quantized DCT coefficients are zig-zag ordered and run-length coded

With intra-prediction, after the best intra-prediction pattern is found, the prediction error block is coded using DCT as above.
Macroblock Coding in P-Mode

Estimate one MV for each macroblock (16x16)

Depending on the motion compensation error, determine the coding mode (intra, inter-with-no-MC, inter-with-MC, etc.)

The original values (for intra mode) or motion compensation errors (for inter mode) in each of the DCT blocks (8x8) are DCT transformed, quantized, zig-zag/alternate scanned, and run-length coded
Macroblock Coding in B-Mode

- Same as for the P-mode, except a macroblock can be predicted from a previous picture, a following one, or both.
Encoding Strategies

• Frame-type selection
• Motion estimation
• Mode selection
• Quantization
• Loop filtering
Overlapped Block Motion Compensation (OBMC)

- Conventional block motion compensation
  - One best matching block is found from a reference frame
  - The current block is replaced by the best matching block

- OBMC
  - Each pixel in the current block is predicted by a weighted average of several corresponding pixels in the reference frame
  - The corresponding pixels are determined by the MVs of the current as well as adjacent MBs
  - The weights for each corresponding pixel depends on the expected accuracy of the associated MV
OBMC Using 4 Neighboring MBs

\[ \psi_p(x) = \sum_{k \in \mathcal{K}} h_k(x) \psi_r(x + d_{m,k}), \quad x \in B_m. \]

\( h_k(x) \) should be inversely proportional to the distance between \( x \) and the center of \( B_{m,k} \).
Weighting Coefficients Used in H.263

Figure 9.16  (a) The weighting coefficients for OBMC, specified in the H.263 video coding standard [17]: (a) for prediction with the motion vector of the current block; (b) for prediction with motion vectors of the blocks on top or bottom of the current block; (c) for prediction with motion vectors of the blocks to the left or right of the current block. The numbers given are 8 × the actual weights. For example, to predict the pixel at the top left corner of the block, the weights associated with the MVs of the current MB, the top MB, and the left MB are 4/8, 2/8, and 2/8, respectively. For the pixel on the first row and the second column, the weights are 5/8, 2/8, 1/8, respectively.
Optimal Weighting Design

• Convert to an optimization problem:

\[
\text{Minimize } \quad E \left\{ \left| \psi(x) - \sum_{k \in \mathcal{K}} h_k(x) \psi_{r}(x + d_{m,k}) \right|^2 \right\}
\]

• Subject to \( \sum_{k \in \mathcal{K}} h_k(x) = 1 \)

• Optimal weighting functions:

\[
h(x) = [R(x)]^{-1} \left( r(x) - \frac{i^T[R(x)]^{-1} r(x) - 1}{i^T[R(x)]^{-1}i} i \right),
\]

\[
R_{k,l}(x) = E\{\psi_r(x + d_{m,k}) \psi_r(x + d_{m,l})\}, \quad k, l \in \mathcal{K}
\]

\[
r_k(x) = E\{\psi(x) \psi_r(x + d_{m,k})\}, \quad k \in \mathcal{K}.
\]
How to Determine MVs with OBMC

• Option 1: Use conventional BMA, minimize the prediction error (MAD) within each MB independently

• Option 2: Minimize the prediction error assuming OBMC
  • Solve the MV for the current MB while keeping the MVs for the neighboring MBs found in the previous iterations

\[ \sum_{x \in B_m} \left| \psi(x) - \sum_{k \in K} h_k(x) \psi_r(x + d_{m,k}) \right|^p. \]

• Option 3: Using a weighted error criterion over a larger block

\[ \sum_{k \in K} \sum_{x \in B_{mk}} |\psi(x) - \psi_r(x + d_m)|^p h_k(x) \rightarrow \sum_{x \in B_m} |\psi(x) - \psi_r(x + d_m)|^p \tilde{h}(x) \]
Window Function Corresponding to H.263 Weights for OBMC
Operational Control of a Video Coder

• Typical video sequences contain varying content and motion
• Different content is compressed well with different techniques
• Encoders should match coding techniques to content

• Coding parameters
  • Macroblock type (coding mode)
  • Motion vectors
  • Quantizer step size

• Each leads to different rate and distortion
Coding Mode Selection

• Coding modes:
  • Intra vs. Inter, QP for each MB, Motion estimation methods and parameters (e.g., with or without overlapping, block size, search range)
  • Each combination will lead to different trade-off between overall rate and distortion of coded video

• Rate-distortion optimized selection, given target rate:
  • Minimize the distortion, subject to the target rate constraint

\[
\text{minimize } \sum_n D_n(m_n), \\
\text{subject to } \sum_n R_n(m_k, \forall k) \leq R_d.
\]

\[
\text{minimize } J(m_n, \forall n) = \sum_n D_n(m_n) + \lambda \sum_n R_n(m_k, \forall k)
\]

Simplified version

\[
J_n(m_n) = D_n(m_n) + \lambda R_n(m_n).
\]
Coding Mode Selection

• The optimal mode is chosen by coding the block with all candidates modes and taking the mode that yields the least cost

• Note that one can think of each candidate MV (and reference frame) as a possible mode, and determine the optimal MV (and reference frame) using this frame work --- Rate-distortion optimized motion estimation
Rate Control: Problem Definition

• The coding method necessarily yields variable bit rate
  • More active periods use more bits – prediction is less accurate
  • An I-frame uses many more bits than a P-frame or B-frame
  • Variable length coding

• Video is almost always either:
  • Sent over a constant bit rate (CBR) channel, where the rate when averaged over a short period should be constant
  • Sent over a variable bit-rate (VBR) channel that does NOT have the same variability as the bitstream

• The bit-rate fluctuation can be smoothed by a buffer at the encoder output
Rate Control: Problem Solution

• Adjust bit-rate of compressed video to avoid encoder buffer overflow

• Step 1) Determine the target rate at the frame, GOB, and MB level based on the current buffer fullness

• Step 2) Satisfy the target rate by varying frame rate (skip frames when necessary) and QP

• Step 3) Satisfy GOB/MB level target rate by varying the coding mode and QP at each MB
  = Rate-distortion optimized mode selection
Loop Filtering

- Errors in previously reconstructed frames accumulate over time with motion compensated temporal prediction
  - Reduce prediction accuracy
  - Increase bit rate for coding new frames

- Loop filtering
  - Filter the reference frame before using it for prediction
  - A side-effect of non-integer motion compensation
  - Explicit deblocking filtering: removing blocking artifacts after decoding each frame

- Loop filtering can significantly improve coding efficiency