Contents

9 Video Compression

9.1 Perception of motion

Perception of motion: Human visual system is specifically sensitive to motion. Eyes follow motion automatically. Some distortions are not as perceivable as in image coding (would be if we froze frame). No good psycho-visual model avaivable. Vusal perception is limited to < 24 Hz. Asuccession of images will be perceived as continuous if frequency is sufficiely high. Cinema 2424 Hz, TV 25 Hz or 50 Hz. We still nee to avoid aliasing (wheel effect). High-rendering framerates desired in computer games (needed due to absence of motion blur). Flicker can be perceived up to > 60 Hz in particular in periphery. Issue addressed by 100 Hz TV.

9.2 Interlaced video format

Two temporarlly shifted half images, increase of frequency 25 Hz → 50 Hz. Reduction of spatial resolution. Full image representation: progressive.

9.3 Why compress video?

Raw HD TV signal $720p@50$ Hz:

$$1280 \cdot 720 \cdot 50 \cdot 24 \text{ bits/s}$$
$$= 1\,105\,920\,000 \text{ bits/s} > 1 \text{ Gb/s}$$

Only 20 Mb/s HDTV channel bandwidth requires compression of factor of 60 (0.4 bits/pixel on average)

9.4 Lossy video compression

Take advantage of redundancy. Spatial correlation between neighboring pixels. Temporal correlation between frames. Drop perceptuall unimportant details.

Temporal Redundancy Take advantage of similarity between successive frames

Temporal processing
Usually high frame rate: Significant temporal redundancy. Possible representations along temporal dimension:
1. Transform/subband methods: Good for textbook case of constant velocity uniform global motion. Inefficient for nonuniform motion, i.e. real-world motion. Requires large number of frame stores which leads to delay. (Memory cost may also be an issue.) Is ineffective for many scene changes or high motion.
2. Predictive methods: Good performance using only 2 frame stores. However, simple frame differencing is not enough…

Goal Exploit the temporal redundancy

Predict current frame based on previously coded frames

Types of coded frames:
1. I-frame: Intra-coded frame, coded independently of all other frames.
2. P-frame: Predictively coded frame, coded based on previously coded frame I or P. Can send motion vector plus changes.
3. B-frame: Bi-directionally predicted frame, coded based on both previous and future coded frames I and P. In case something is uncovered.

Motion-compensated prediction
Simple frame differencing *fails* when there is motion. Must account for motion. → Motion-compensated (MC) prediction. MC-prediction generally provides significant improvements. Questions: How can we estimate motion? How can we form MC-prediction?

Ideal situation
1. Partition video into moving objects
2. describe object motion → Generally very difficult

Practical approach Block-Matching Motion Estimation:
1. Partition each frame into blocks, e.g. 16×16 pixels
2. Describe motion of each block

→ No object identification required and good, robust performance.

9.5 Block-matching motion estimation

Assumptions: 1. Translational motion within block:
$f(n_1, n_2, k_{\text{cur}})$
$= f(n_1 - mv_1, n_2 - mv_2, k_{\text{ref}})$.

ME Algorithm 1. Divide current frame into non-overlapping $N_1 \times N_2$ blocks.
2. For each block, find the best matching block in reference frame.

9.5.1 Determining the best matching block

For each block in the current frame, search for best matching block in the reference frame.

Metrics for determining "best match":

MSE
$$= \sum_{n_1, n_2 \in \text{Block}} [f(n_1, n_2, k_{\text{cur}}) - f(n_1 - mv_1, n_2 - mv_2, k_{\text{ref}})]^2 .$$

MAE
$$= \sum_{n_1, n_2 \in \text{Block}} |f(n_1, n_2, k_{\text{cur}}) - f(n_1 - mv_1, n_2 - mv_2, k_{\text{ref}})| .$$

Candidate blocks All blocks in, e.g. $(\pm 32, \pm 32)$ pixel area

Strategies for searching candidate blocks for best match.
1. Full search: Examine all candidate blocks
2. Partial (fast) search: Examine a carefully selected subset.

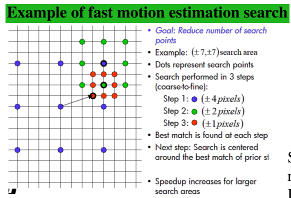
Motion vector Estimate of motion for best matching block.

9.6 Motion vector and motion vector field

Motion vector Expresses the *relative horizontal and vertical offsets* (mv_1, mv_2), or motion, of a given block from one frame to another.

Motion vector field Collection of motion vectors for all the blocks in a frame.

Example of fast motion estimation search



Motion Vector Presision
- Motivation: Motion is not limited to integer-pixel offsets. However, video is only known at discrete pixel locations. To estimate sub-pixel motion, frames must be spatially interpolated.
- Fractional MVs are used to represent the sub-pixel motion.
- Improved performance (extra complexity is worthwhile)
- Half-pixel ME used in most standards: MPEG-1/2/4
- Why are half-pixel motion vectors better? They can capture half-pixel motion. Averaging effect (from spatial interpolation) reduces prediction error → Improved prediction. For noisy sequences, averaging effect reduces noise → Improved compression.

9.6.1 Practical Half-Pixel Motion Estimation Algorithm

Half-pixel ME (coarse-fine) algorithm:
1. Coarse step: Perform integer motion estimation on blocks; find best integer-pixel MV
2. Fine step: Refine estimate to find best half-pixel MV
   a. Spatially interpolate the selected region in reference frame
   b. Compare current block to interpolated reference frame block.
   c. Choose the integer or half-pixel offset that provides best match Typically, bilinear interpolation is used for spatial interpolation

9.7 Block Matching Algorithm

Issues Block size, search range, motion vector accuracy

Estimate Done typically only from luminance

Advantages 1. Good, robust performance for compression.
2. Resulting motion vector field is easy to represent (one MV per block) and useful for compression.
3. Simple, periodic structure, easy VLSI implementations

Disadvantages
1. Assumes translational motion model → Breaks down for more complex motion.
2. Ofter produces blocking artifacts (OK for coding with Block DCT)

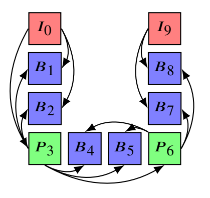
Bidirectional MC prediction is uset to estimate a block in the current frame from a block in:
1. Previous frame
2. Future frame
3. Average of ablock from the previous frame and a block from the future frame
4. Neither, i.e. code current block without prediction

Example: Prediction with P- and B-frames
1. Motion compensated prediction: Predict the current frame based on reference frame(s) while compensating for the motion.
2. Examples of block-based motion-compensated prediction (P-frame) and bi-directional prediction (B-frame).

9.8 Frame types

Main addition over image compression: Exploit the temporal redundancy. Predict current frame based on previously coded frames. Three types of coded frames:
1. *I-frame*: Intra-coded frame, coded independently of all other frames
2. *P-frame*: Predictively coded frame, coded based on previously coded frame
3. *B-frame*: Bi-directionally predicted frame, coded based on both previous and future coded frames.

MPEG Group of Pictures (GOP)



Starts with an I-frame, ends with frame right before next I-frame. "Open" ends in B-frame, "closed" in P-frame. MPEG Encoding a parameter, but "typical":

$$IBBPBBPBBI,$$
$$IBBPBBPBBPBBI.$$

Periodic I-frames enable random access into the coded bitstream. Parameters:
1. Spacing between I frames
2. Number of B frames between I and P frames

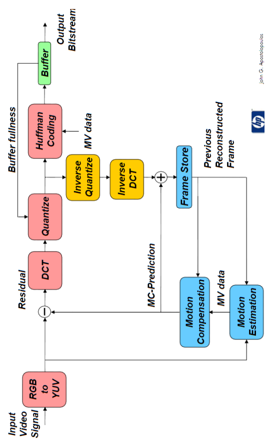
Example compression performance
I: $\frac{1}{7}$, P: $\frac{1}{20}$, B: $\frac{1}{50}$, Average: $\frac{1}{27}$.

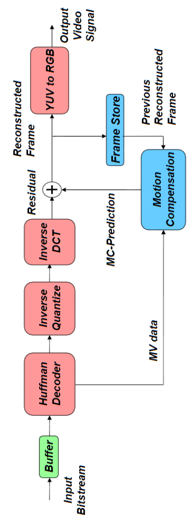
9.9 Summary of Temporal Processing

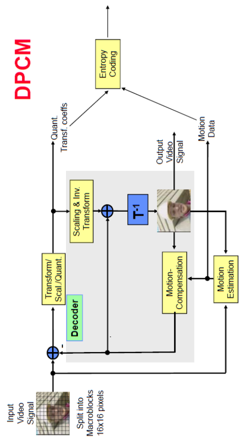
1. Use MC-prediction (P and P frames) to reduce temporal redundancy.
2. MC-prediction usually performs well; In compression have a second changce to recover when it performs badly.
3. MC-prediction yields
   a. Motion vectors
   b. MC-prediction error or residual → Code error with conventional image coder
4. Sometimes MC-prediction may *perform badly*
   a. Examples: complex motion, new imagery (occlusions)
   b. Approach: *1.* Identify frame or individual blocks where prediction fails *2.* Code without prediction

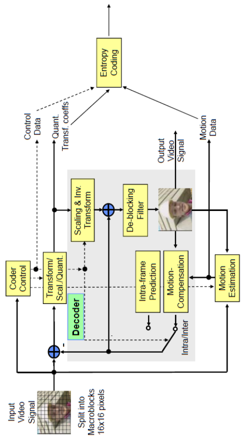
9.10 Basic Video Compression Architecture

1. Exploiting the reduncancies:
   a. Temporal: MC-prediction (P and B frames)
   b. Spatial: Block DCT
   c. Color: color space conversion
2. Scalar quantization of DCT coefficients
3. Zigzag scanning, runlength and Huffman coding of the nonzero quantized DCT coefficients









Transimission chain Content Creation → Transmission → Consumption. ⇒ Variable needs!

9.11 Scalable Video Coding

Produces *different layers with prioritized importance. Prioritized importance is key* for a variety of applications:
1. *Adapting* to different bandwidths, or client resources such as spatial or temporal resolution or computational power.
2. *Facilililates error-resilience* by explicitly identifying most important and less important bits.

Procedure:
1. Decompose video into *multiple layers of prioritized importance*
2. Code layers into *base and enhancement* bitstreams
3. Progressively combine *one or more bitstreams* to produce *different levels of video quality.*

Example of scalable coding with base and two enhancement layers: Can produce three different qualities: 1. Base layer 2. Base + Enh1 layers 3. Base + Enh1 + Enh2 layers Scalability with respect to: Spatial or temporal resolution, bit rate, computation, memory.

Example • Encode image/video into three layers: Base, Enh1, Enh2 • Low-bandwidth recoiver: Send only Base layer. • Medium-bandwidth recoiver: Send Base & Enh1 layers • High-bandwidth receiver: Send all three layers Base, Enh1, Enh2. • Can adapt to different clients and network situations • Three basic types of scalability (refine video quality along three different dimensions):
   a. Temporal scalability → Temporal resolution
   b. Spatial scalability → Spatial resolution
   c. SNR (quality) scalability → Amplitude resolution

- Each type of scalable coding provides scalability of one dimension of the video signal. Can cambine multiple types of scalability to provide scalability along multiple dimensions

Temporal Scalability Based on the use of *B-frames* to refine the *temporal resolution* B-frames are dependent on other frames. Howere, no other frame depends on a B-frame. Each B-frame may be discarded without affecting other frames.

Spatial scalability Based on refining the *spatial resolution Base layer* is *low resolution* version of video. *Enh1* contains coded *difference* between upsampled base layer and original video. Also called pyramid coding.

SNR scalability Based on refining the *amplitude resolution Base layer* uses a *coarse quantizer. Enh1* applies a *finer quantizer* to the difference between the original DCT coefficients and the coarsely quantized base layer coefficients.

9.12 Standards

Goal *Ensuring interoperability*: Enbabling communication between devices made by different manufacturers. Promoting a technology or industry. Reducing costs.

Scope of standardization Not the encoder, not the decoder. Just the *bitstream syntax* and the *decoding process* (e.g. use IDCT but not how to implement the IDCT) This enables improved encoding and decoding strategies to be employed in a standard-compatible manner.

9.13 Quality measure

objective: PSNR • Error for one pixel, difference between original and decoded value
$e(v, h) = \tilde{x}(v, h) - x(v, h)$
- Mean-squared-Error, MSE e.g. over an image

$$e_{\text{MSE}} = \sqrt{\frac{1}{N \cdot M} \sum_{v,h=1}^{v=N,h=M} e^2(v, h)}$$

- Peak-Signal-to-Noise-Ratio
PSNR $= [\max x]^2 / e^2_{\text{MSE}}$

E.g. $x = 2^K$ or 255. One can use a log-scale like dB.

10 Questions

- $I_{\text{comp}} = I_\alpha I_a + (1 - I_\alpha) I_b$
- MAP, Maximum a posteriori detector.
- graph cuts
- Solve MRFs with graph cuts
- impulse response $t(-x, -y)$
- Canny nonmaxima suppression
- Entropy Coding (Huffman code)
- Aperture problem: normal flow
- Lucas-Kanade: Iterative refinement/local gradient method
- Coarse-to-fine-estimation
- SNR scalability EI, EP frame

A Big equations

$$\mathcal{F}[h](u, v) = \frac{1}{2\ell} \int_{-\ell}^{\ell} \mathrm{d}x_1 \exp(-i2\pi u x_1) \cdot \underbrace{\int_{-\infty}^{\infty} \mathrm{d}x_2 \, \delta(x_2) \exp(-i2\pi v x_2)}_{=1}$$
$$= \operatorname{sinc}(2\pi u \ell) \tag{1}$$

$$E = \iint \mathrm{d}x\mathrm{d}y \left[\left(\frac{\partial I}{\partial x}\frac{\mathrm{d}x}{\mathrm{d}t} + \frac{\partial I}{\partial y}\frac{\partial y}{\partial t} + \frac{\partial I}{\partial t} \right)^2 + \alpha^2 (\|\nabla \dot{x}\| + \|\nabla \dot{y}\|)^2 \right] \tag{2}$$

$$\mathbf{v} = \begin{pmatrix} \sum_i w_i I_x(q_i)^2 & \sum_i w_i I_x(q_i) I_y(q_i) \\ \sum_i w_i I_x(q_i) I_y(q_i) & \sum_i w_i I_y(q_i)^2 \end{pmatrix}^{-1} \cdot \begin{pmatrix} -\sum_i w_i I_x(q_i) I_t(q_i) \\ -\sum_i w_i I_y(q_i) I_t(q_i) \end{pmatrix} \tag{3}$$