Video Coding

Correlation in Video Sequence

- **Spatial correlation**
  - Similar pixels seem to group together – just like spatial correlation in images

- **Temporal correlation**
  - Similar background with a few moving objects in the foreground

There is usually much more **temporal** correlation then spatial.
Temporal activity in a “Talking Head” video sequence
Video Coding

Motion Model in Video Sequence

- **Natural motion**
  - Moving people & moving objects
  - Translational, rotational, scaling, shear mapping

- **Camera motion**
  - Camera panning, camera zooming, fading

Shear Mapping

en.wikipedia.org/wiki/Shear_mapping
Motion-JPEG

Encoder 1

- **Intraframe** compression
- Encodes every frame independently

**Pros:**
- Very simple and fast
- No latency (frame delay)
- Easy frame access for video editing

**Cons:**
- Does not take into account any temporal correlation
- Low coding performance
Motion-JPEG

Encoder 2

- **Interframe compression**
  - Takes advantage of minimal changes from one frame to the next to achieve dramatic compression. Instead of storing complete information about each frame only the difference information between frames is stored.

- **Encodes frame difference**

- **Pros:**
  - Improves compression
  - Low latency

- **Cons:**
  - Requires very stationary background to be effective
  - Still low coding performance (too simple motion model)
  - To prevent quantization error accumulation from coder/decoder mismatch, has to be closed-loop
Motion-Compensated Framework

Introduction

- **Motion Estimation and Motion Compensation (MEMC)**
- **Transform-based coding** on motion-compensated prediction error (residue)
- All international video coding standards are based on this coding framework
  - Video teleconferencing: H.261, H.263, H.263++, H.26L/H.264
  - Video archive & play-back: MPEG-1, MPEG-2 (in DVDs) MPEG-4
Motion-Compensated Framework

Motion Estimation

- Motion estimation is the process of determining **motion vectors** that describe the transformation from one 2D image to another.
- **Block Matching Algorithm (BMA):**
  - Subdivide current frame into blocks.
  - Find one displacement vector for each block (Within a search window, find a “best match” that **minimizes an error measure**).
- Intelligent search strategies can reduce computation.

![Motion Estimation Diagram](image-url)
Motion-Compensated Framework

Block Matching Algorithm, Ex. 1

Measurement window is compared with a shifted block of pixels in the other image, to determine the best match.

Block of pixels is selected as a measurement window.
Motion-Compensated Framework

Block Matching Algorithm, Ex. 2

Previous Frame

Current Frame

... process repeated for another block.
Block Matching Algorithm

Optical Flow, Ex. 1

Suzie QCIF animation: The sequence shows a woman that faces the camera while speaking on the telephone.

Frames 47 to 52

The circles in the plot marks the center of a block in the current frame and the lines goes out from the center of a circle to the position of where this block was located in the previous frame. This sequence mainly has two moving parts that move in different directions. These are the head that goes to the right while turning and the telephone that goes down. The frames also consist of a rather large portion that is, except from some noise, constant from frame to frame. A typical motion between frames in this sequence is 5 pixels for the telephone and 0 for the background.

In these frames the woman is simply listening to the phone and the frames contains virtually no motion at all.
Block Matching Algorithm

Optical Flow, Ex. 2

The frames 308 to 313 of the foreman QCIF movie. These frames show a pan of the camera across a construction yard. The sequence consists of large diagonal motion.

Large portions of new image are presented in the lower and right edges of each frame. The size of the motion between frames is somewhere between 8 and 9 pixels. All motion of this sequence is translatory.
Block Matching Algorithm

From Current to Previous Frame
Motion-Compensated Framework

Motion Compensation

- Applying the motion vectors to an image to synthesize the transformation to the next image is called motion compensation.

Original frame

Differences between the original frame and the next frame, shifted right by 2 pixels.

Shifting the frame compensates for the panning of the camera, thus there is greater overlap between the two frames.
Motion-Compensated Framework

Encoder and Decoder

Diagram:

- Input Frame
- Transform, Quantization, Entropy Coding
- Encoded Residual (To Channel)
- Decoder
- Motion Compensated Prediction
- Entropy Decoding, Inverse Q, Inverse Transform
- Approximated Input Frame (To Display)
- Frame Buffer (Delay)
- Motion Comp. Predictor
- Motion Estimation
- Motion Vector and Block Mode Data (Side-Info, To Channel)
Motion-Compensated Framework

MEMC, Example

![Previous frame](image1)

![Current frame](image2)

![Prediction with displacement vectors](image3)

![Motion-compensated Prediction error](image4)
MEMC, Example
MEMC, Example

Previous Frame  Current Frame  Motion Vector Field

Frame Difference  Motion-Compensated Difference
Block Matching Algorithm

Summary

- **Issues:**
  - Block size?
  - Search range?, Search strategy?
  - Motion vector accuracy? (Motion model? Matching criterion?)
    - Motion typically estimated only from luminance

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

- **Disadvantages:**
  - Assumes translational motion model →Breaks down for more complex motion.
  - Often produces blocking artifacts (OK for coding with Block DCT).
Motion-Compensated Framework

Motion Model for Estimation

- **Affine motion**
  - Translation, rotation, scaling, shear mapping
  - 6 parameters in 2-D space
- **Complex**
- **Simplified translation model**
  - Motion vectors (2 parameters)

In geometry, an **affine** transformation is a transformation which **preserves straight lines** (i.e., all points lying on a line initially still lie on a line after transformation) and ratios of distances between points lying on a straight line (e.g., the midpoint of a line segment remains the midpoint after transformation). It does not necessarily preserve angles or lengths, but does have the property that sets of parallel lines will remain parallel to each other after an affine transformation.

[en.wikipedia.org/wiki/Affine_transformations](en.wikipedia.org/wiki/Affine_transformations)
Motion-Compensated Framework

Matching Criterion

Mean Square Error (MSE)

\[
MSE(u, v) = \frac{1}{16^2} \sum_{i=0}^{15} \sum_{j=0}^{15} \left[ f_t(i, j) - f_{t-1}(i + u, j + v) \right]^2,
\]

\( u: \) Horizontal shift

\( v: \) Vertical Shift

\(-15 \leq u, v \leq 15\)
Motion-Compensated Framework

Matching Criterion

Sum of Squared Differences (SSD)

\[
SSD(u, v) = \sum_{i=0}^{15} \sum_{j=0}^{15} \left[ f_t(i, j) - f_{t-1}(i+u, j+v) \right]^2,
\]

- \( u \): Horizontal shift
- \( v \): Vertical Shift

\(-15 \leq u, v \leq 15\)
Motion-Compensated Framework

Matching Criterion

- **Sum of Absolute Differences (SAD)**

\[
SAD(u, v) = \sum_{i=0}^{15} \sum_{j=0}^{15} |f_t(i, j) - f_{t-1}(i + u, j + v)|,
\]

where:
- \(f_t\): Current frame
- \(f_{t-1}\): Previous frame
- \(u\): Horizontal shift
- \(v\): Vertical Shift

\(-15 \leq u, v \leq 15\)
Motion-Compensated Framework

Matching Criterion

Estimated displacement Integer-pixel accuracy
Block Matching

Search Strategies, Full Search

- All possible displacements within the search range are compared.
- Computationally expensive.
- Highly regular, parallelizable.
Block Matching

Search Strategies, Fast methods

- **Divide and Conquer**
  - Sampling the search space
  - Divide search space into regions
  - Pick a center point for each region
  - Perform more elaborate search on the region with best center

- **Gradient Search**
  - Sampling the search space
  - At every stage, investigate a few candidates
  - Move in the direction of the best match
  - Can adaptively reduce the displacement size for each stage
Block Matching

Search Strategies, 2D Log Search

- Algorithm: Gradient Search
  - Use + pattern at each stage (5 candidates)
  - Move center to best match
  - Reduce step size at each stage
- Fast, reasonable quality
- Similar algorithms:
  - Three-step-search
  - Four-step-search
  - Cross search...
Block Matching, 3 Step Search
Block Matching, 3 Step Search

Block Based Gradient Descent Search
Block Matching

Search Strategies, Binary Search

- Algorithm: Divide-and-conquer
- Move search to region with best match
- Finer search at later stages
Block Matching

Diamond search [Li, Zeng, Liou, 1994] [Zhu, Ma, 1997]

- Start with large diamond pattern at (0,0)
- If best match lies in the center of large diamond, proceed with small diamond
- If best match does not lie in the center of large diamond, center large diamond pattern at new best match
Block Matching

Search Strategies, Acceleration

- **Predictive motion search**
  - Use *median of motion vectors* in causal neighborhood as starting point for search.
  - Additionally test zero-vector as a starting point

- **Early termination**
  - Interrupt summation to calculate SSD or SAD, if value *grows too quickly* (relative to previous best match)
  - Stop search, if match is “*good enough*” (SSD, SAD < threshold)

- Tradeoff between the *quality of the motion vector* versus the *complexity of the motion estimation* process is left to the implementer.
Block Matching

Block Size

- **Small** Blocks: It is possible in this case that a match may be erroneously established between blocks containing similar gray-level patterns which are otherwise unrelated in the sense of motion.

- **Large** Blocks: If the motion varies locally (possibly within the measurement block), for instance due to independently moving smaller size structures within the block, it is obvious that block matching will provide inaccurate motion estimates when large blocks are used.
Video Coding

Temporal picture structures

- sequence
- group of pictures
- picture
- slice
- macroblock
- block
Video Coding

Picture Types

- **I-frames** are the least compressible but don't require other video frames to decode (a.k.a Key frames).
- **P-frames** can use data from previous frames to decompress and are more compressible than I-frames.
- **B-frames** can use both previous and forward frames for data reference to get the highest amount of data compression.

MPEG-1 has a unique frame type not found in later video standards. **D-frames** or DC-pictures are independent images (intra-frames) that have been encoded DC-only (AC coefficients are removed) and hence are very low quality.
Video Coding

B-frames

Frame 1

Frame 2

Frame 3

Previous frame doesn’t contain this information

Next frame does. Can we code from this?

Often an object moves in front of a background. P frames code the object fine, but can’t effectively code the revealed background.
Video Coding

B-frames

- Coded from one previous frame, one future frame, or a combination of both.
  - 1. Do motion vector search separately in past reference frame and future reference frame.
  - 2. Compare:
    - Difference from past frame.
    - Difference from future frame.
    - Difference from average of past and future frame.
  - 3. Encode the version with the least difference.
Video Coding

Group of Pictures (GOP)

- A GOP always begins with an I-frame. Afterwards several P-frames follow, in each case with some frames distance. In the remaining gaps are B-frames. A few video codecs allow for more than one I-frame in a GOP.

The more I-frames the video stream has, the more editable it is. However, having more I-frames increases the stream size. In order to save bandwidth and disk space, videos prepared for internet broadcast often have only one I-frame per GOP.

The GOP structure is often referred by two numbers, for example M=3, N=12. The first one tells the distance between two anchor frames (I or P). The second one tells the distance between two full images (I-frames): it is the GOP length. For the example M=3 N=12, the GOP structure is IBPBBPBBPBBBI. Instead of the M parameter one can use the maximal count of B-frames between two consecutive anchor frames.

GOP, M=3, N=9.
Video Coding

How Many I, P, or B

- A few hints:
  - Only use motion vector if a “close” match can be found.
  - Evaluate “closeness” with MSE or other metric.
  - Can’t search all possible blocks, so need a smart algorithm.
  - If no suitable match found, just code the macroblock as an I-block.
  - If a scene change is detected, start fresh.
- Don’t want too many P or B frames in a row
  - Predictive error will keep propagating until next I frame.
  - Delay in decoding.
Video Coding

Codec Information Appliances

http://avidemux.sourceforge.net/
(Video editor)

http://mediainfo.sourceforge.net/en/
(Shows video codec info.)

http://www.headbands.com/gspot/
(Shows video codec info.)

Avidemux is a free video editor designed for simple cutting, filtering and encoding tasks. It supports many file types, including AVI, DVD compatible MPEG files, MP4 and ASF, using a variety of codecs.
Thank You

Next Session: Video III (Video Coding Standards)

FIND OUT MORE AT...

1. http://ce.sharif.edu/~m_amiri/