Overview: motion-compensated coding

- Motion-compensated prediction
- Motion-compensated hybrid coding
- Motion estimation by block-matching
- Motion estimation with sub-pixel accuracy
- Power spectral density of the motion-compensated prediction error
- Rate-distortion analysis
- Loop filter
- Motion compensated coding with sub-pixel accuracy
- Rate-constrained motion estimation
Motion-compensated prediction

Prediction for the luminance signal $S(x,y,t)$ within the moving object:

$$\hat{S}(x,y,t) = S(x - d_x, y - d_y, t - \Delta t)$$
Combining transform coding and prediction

Transform domain prediction

\[ T \rightarrow Q \rightarrow P_T \rightarrow T^{-1} \]

Space domain prediction

\[ T^{-1} \rightarrow P_S \rightarrow T^{-1} \]
Motion-compensated hybrid coder

Diagram showing the components of a motion-compensated hybrid coder, including:
- Coder Control
- Intra-frame DCT Coder
- Decoder
- Intra/Inter
- Motion-COMPENSATED Predictor
- Motion Estimator
- Intra-frame Decoder

Flow of data includes:
- Control Data
- DCT Coefficients
- Motion Data
Motion-compensated hybrid decoder
Block-matching algorithm

- Subdivide current frame into blocks.
- Find one displacement vector for each block.
- Within a search range, find a best „match“ that minimizes an error measure.
- Intelligent search strategies can reduce computation.
Block-matching algorithm

Reference frame

Current frame

Block is compared with a shifted array of pixels in the reference frame to determine the best match

Block of pixels is considered
Block-matching algorithm

Reference frame

Current frame

... process repeated for the next block
Sum of Squared Differences to determine similarity

\[
SSD(d_x, d_y) = \sum_{x,y \in \text{Block}} \left[ S_k(x, y) - S_{k-1}(x + d_x, y + d_y) \right]^2
\]

Alternative matching criteria: SAD (Sum of Absolute Differences), cross correlation, . . .

Only integer pixel shifts (so far)
Integer Pixel Shifts

Block is compared with a shifted array of pixels in the reference frame to determine the best match.
### Integer Pixel Shifts

A block of pixels is compared with a shifted array of pixels in the reference frame to determine the best match.

**Block of pixels is considered**

| 28 42 42 43 44 40 32 20 29 32 22  |
| 30 44 45 45 42 30 21 26 27 18  |
| 35 54 53 52 52 52 52 52 52 52 52  |
| 74 121 120 120 120 120 120 120 120 120 120  |
| 79 127 110 110 110 110 110 110 110 110 110  |
| 80 129 111 111 111 111 111 111 111 111 111  |
| 50 78 77 77 77 77 77 77 77 77 77  |
| 22 37 37 37 37 37 37 37 37 37 37  |

| 54 53 52 49 31 21  |
| 62 63 59 60 44 33  |
| 120 114 112 111 80 32  |
| 130 128 124 125 88 24  |
| 131 124 127 127 96 42  |
| 77 71 73 75 63 52  |
SSD Values Resulting from Blockmatching

![SSD Values Diagram]

Estimated displacement
Integer-pixel accuracy
Motion-compensated prediction: example

Previous frame

Current frame

Current frame with displacement vectors

Motion-compensated Prediction error
Interpolation of the SSD Minimum

Fit parabola through >3 points approximately

Horizontal shift $d_x$
2-d Interpolation of SSD Minimum

Paraboloid
- Perfect fit through 6 points
- Approximate fit through >6 points
Blockmatching: search strategies I

Full search

- All possible displacements within the search range are compared.
- Computationally expensive
- Highly regular, parallelizable
Blockmatching: search strategies II

2D logarithmic search [Jain + Jain, 1981]

- Iterative comparison of error measure values at 5 neighboring points
- Logarithmic refinement of the search pattern if
  - best match is in the center of the 5-point pattern
  - center of search pattern touches the border of the search range
Blockmatching: search strategies III

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
Blockmatching: search strategies IV

Most search strategies can be further accelerated by . . .

- **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)
Block comparison speed-ups

- Triangle and Cauchy-Schwarz inequality for SAD and SSE

\[
\sum_{\text{block}} |S_k - S_{k-1}| \geq \left| \sum_{\text{block}} S_k - S_{k-1} \right| = \left| \sum_{\text{block}} S_k - \sum_{\text{block}} S_{k-1} \right|
\]

\[
\sum_{\text{block}} (S_k - S_{k-1})^2 \geq \frac{1}{N} \left( \sum_{\text{block}} S_k - S_{k-1} \right)^2 = \frac{1}{N} \left( \sum_{\text{block}} S_k - \sum_{\text{block}} S_{k-1} \right)^2
\]

- Strategy:
  - Compute partial sums for blocks in current and previous frame
  - Compare blocks based on partial sums
  - Omit full block comparison, if partial sums indicate worse error measure than previous best result

- Performance: > 20x speed-up of full search block matching reported by employing
  
  
  - Sum over 16x16 block
  - Row wise block projection
  - Column wise block projection
Hierarchical blockmatching

Displacement vector field

Block matching

current frame

previous frame

Filtering and subsampling

Filtering and subsampling

Block matching

Block matching

Block matching
Sub-pel accuracy

Displacement vector field with 1/2-pel accuracy

Block matching

Displacement vector field with integer-pel accuracy

Interpolation

Filtering and subsampling

current frame

previous frame

Filtering and subsampling
Sub-pel accuracy

- Interpolate pixel raster of the reference frame to desired fractional pel accuracy (e.g., by bi-linear interpolation)
- Straightforward extension of displacement vector search to fractional accuracy
- Example: half-pel accurate displacements

\[
\begin{bmatrix}
\frac{1}{2} \\
\frac{1}{2}
\end{bmatrix} = \begin{bmatrix}
4.5 \\
4.5
\end{bmatrix}
\]
Bi-linear Interpolation

Interpolated Pixel Value
Bi-linear Interpolation (cont.)

\[ I(x', y') = \begin{bmatrix} 1 - b & b \\ I(x, y) & I(x + 1, y) \\ I(x, y + 1) & I(x + 1, y + 1) \end{bmatrix} \begin{bmatrix} 1 - a \\ a \end{bmatrix} \]
Model for performance analysis of an MCP hybrid coder

luminance signal $S$

- $e$

R-D optimal intraframe encoder

intraframe decoder

motion compensated predictor

$e'$

$s'$

true displacement

$$\begin{pmatrix} d_x \\ d_y \end{pmatrix} + \begin{pmatrix} \Delta_x \\ \Delta_y \end{pmatrix}$$

displacement estimate

displacement error
Analysis of the motion-compensated prediction error

Motion-compensated signal
\[ c(x) = s(x - \Delta_x) - n(x) \]

Prediction error
\[ e(x) = s(x) - c(x) \]
\[ = s(x) - s(x - \Delta_x) + n(x) \]
Motion-compensated prediction error

\[ e(x) = s(x) - c(x) = s(x) - s(x - \Delta_x) + n(x) = \left( \delta(x) - \delta(x - \Delta_x) \right) * s(x) + n(x) \]

Power spectrum of prediction error, assuming constant displacement error \( \Delta_x \), statistical independence of \( s \) and \( n \)

\[
\Phi_{ee}(\omega) = \Phi_{ss}(\omega) \left( 1 - e^{-j\omega\Delta_x} \right) \left( 1 - e^{j\omega\Delta_x} \right) + \Phi_{nn}(\omega) \\
= 2\Phi_{ss}(\omega) \left( 1 - \text{Re} \left\{ e^{-j\omega\Delta_x} \right\} \right) + \Phi_{nn}(\omega)
\]

Random displacement error \( \Delta_x \), statistically independent from \( s, n \)

\[
\Phi_{ee}(\omega) = E \left\{ 2\Phi_{ss}(\omega) \left( 1 - \text{Re} \left\{ e^{-j\omega\Delta_x} \right\} \right) + \Phi_{nn}(\omega) \right\} \\
= 2\Phi_{ss}(\omega) \left( 1 - \text{Re} \left\{ E \left\{ e^{-j\omega\Delta_x} \right\} \right\} \right) + \Phi_{nn}(\omega) \\
= 2\Phi_{ss}(\omega) \left( 1 - \text{Re} \left\{ P(\omega) \right\} \right) + \Phi_{nn}(\omega)
\]
What is $P(\omega)$?

$$P(\omega) = E \left\{ e^{-j\omega \Delta_x} \right\}$$

$$= \int_{-\infty}^{\infty} p_{\Delta_x}(\Delta) e^{-j\omega \Delta} d\Delta = F \left\{ p_{\Delta_x}(\Delta_x) \right\}$$

Fourier transform of the displacement error pdf!

- Same as characteristic function of displacement error, except for sign
- Extension to 2-d

$$\Phi_{ee}(\omega_x, \omega_y) = 2\Phi_{ss}(\omega_x, \omega_y) \left(1 - \text{Re}\left\{ P(\omega_x, \omega_y) \right\} \right) + \Phi_{nn}(\omega_x, \omega_y)$$

- Fourier transform of the displacement error pdf
- Noise spectrum
- Power spectrum of luminance signal

$p(\Delta_x, \Delta_y)$
Power spectrum of motion-compensated prediction error

Signal spectrum $\Phi_{ss}$

large displacement estimation errors ($\sigma_{\Delta x} = 1.0$ pel)

Motion-compensated prediction error spectra $\Phi_{ee}$

small displacement estimation errors ($\sigma_{\Delta x} = 0.2$ pel)

POWER SPECTRAL DENSITY (dB)

frequency $\omega_x$
R-D function for MCP with integer-pixel accuracy

- \((\Delta_x, \Delta_y)^T\) assumed uniformly distributed between
  - \(\Delta_x = \pm \frac{1}{2}\) pel
  - \(\Delta_y = \pm \frac{1}{2}\) line
- Gaussian signal model
  - \(\Phi_{ss}(\omega_x, \omega_y) = A \left( 1 + \frac{\omega_x^2 + \omega_y^2}{\omega_0^2} \right)^{-\frac{3}{2}}\)
- Typical parameters for CIF resolution (352 x 288 pixels)

\[ \text{Minimum bit-rate for given SNR} \]
Required accuracy of motion compensation

- \( p(\Delta x, \Delta y) \) isotropic Gaussian pdf with variance \( \sigma^2 \)
- \( \Phi_{ss}(\omega_x, \omega_y) = A \left( 1 + \frac{\omega_x^2 + \omega_y^2}{\omega_0^2} \right)^{-\frac{3}{2}} \)
- Typical parameters for CIF resolution (352 x 288 pixels)
- Minimum bit-rate for SNR = 30 dB
Model of MCP hybrid coder with loop filter

- Luminance signal $S$
- Spatial filter $F$
- R-D optimal intraframe encoder
- Intraframe decoder
- Motion compensated predictor
- True displacement
- Displacement estimate
- Displacement error

$$\begin{pmatrix} d_x \\ d_y \end{pmatrix} + \begin{pmatrix} \Delta_x \\ \Delta_y \end{pmatrix}$$
Motion-compensated prediction error with loop filter

Motion-compensated signal:
\[ c(x) = s(x - \Delta_x) - n(x) \]

Prediction error:
\[ e(x) = s(x) - f(x) \ast c(x) = s(x) - f(x) \ast s(x - \Delta_x) + f(x) \ast n(x) \]

Previous frame

Current frame

Displacement error \( \Delta_x \)

Displacement \( d_x \)

Impulse response of loop filter
Spatial power spectrum of m.c. prediction error with loop filter

$$\Phi_{ee}(\Lambda) = \Phi_{ss}(\Lambda) \left( 1 + |F(\Lambda)|^2 - 2 \text{Re} \{ F(\Lambda)P(\Lambda) \} \right) + \Phi_{nn}(\Lambda) |F(\Lambda)|^2$$

- $P(\Lambda)$: 2-D Fourier transform of displacement error pdf
- $F(\Lambda)$: 2-D Fourier transform of $f(x, y)$
- $\Phi_{uu}$: spatial spectral power density of signal $u$
- $\Lambda$: vector of spatial frequencies $(\omega_x, \omega_y)$
- $n(x, y)$: noise
Optimum loop filter

- Wiener filter minimizes prediction error variance

\[ F_{\text{opt}}(\Lambda) = P^*(\Lambda) \cdot \frac{\Phi_{ss}(\Lambda)}{\Phi_{ss}(\Lambda) + \Phi_{nn}(\Lambda)} \]

accounts for accuracy of motion compensation

accounts for noise

- Resulting minimum prediction error spectrum

\[ \Phi_{ee}(\Lambda) = \Phi_{ss}(\Lambda) \left( 1 - |P(\Lambda)|^2 \frac{\Phi_{ss}(\Lambda)}{\Phi_{ss}(\Lambda) + \Phi_{nn}(\Lambda)} \right) \]
Effect of loop filter

- Moderately accurate motion compensation
- Very accurate motion compensation

Log power spectral density vs. $f_x$
Required accuracy of motion compensation with loop filter

- $p(\Delta x, \Delta y)$ isotropic Gaussian pdf with variance $\sigma^2$
- Minimum bit-rate for SNR = 30 dB

![Graph showing rate vs. displacement estimation error variance for different motion compensation methods. The graph indicates ~0.8 bpp at a certain variance level.](image-url)
Practical optimum loop filter design

- Not practical for loop filter design

\[
F_{\text{opt}}(\Lambda) = P^*(\Lambda) \cdot \frac{\Phi_{ss}(\Lambda)}{\Phi_{ss}(\Lambda) + \Phi_{nn}(\Lambda)}
\]

Motion compensation accuracy not known

“Noise” psd not known

- To determine Wiener filter from measurements:

\[
F_{\text{opt}}(\Lambda) = \frac{\Phi_{sc}(\Lambda)}{\Phi_{cc}(\Lambda)}
\]

cross spectrum between \(s(x,y)\) and the motion-compensated signal

\[
c(x,y) = r(x - \hat{d}_x, y - \hat{d}_y)
\]
Experimental evaluation of fractional-pixel motion compensation

- ITU-R 601 TV signals, 13.5 MHz sampling rate, interlaced, blockwise motion compensation with blocksize 16x16

![Graphs showing prediction error variance vs. motion compensation accuracy for Zoom and Voiture]
Influence of noise on the performance of MCP
Motion Compensation Performance in H.263

Simulation details:
Foreman, QCIF, SKIP=2
Q=4,5,7,10,15,25
Rate-constrained motion estimation I

\[
\frac{\partial D}{\partial R_m} = \frac{\partial D}{\partial R_e}
\]

optimum trade-off:
Rate-constrained motion estimation II

- How to find best motion vector subject to rate constraint?
- Lagrangian cost function: solve unconstrained problem rather than constrained problem

\[ \min(D + \lambda R_m) \]

- Interpret motion search as ECVQ problem.

⇒ Interpret motion search as ECVQ problem.
Rate-constrained Motion Estimation in H.263 Reference Model TMN-10

Simulation details:
Foreman, QCIF, SKIP=2
Q=4,5,7,10,15,25
Annexes D+F
Video coder control

- Encoding decisions
  - Coding modes (intra/inter/motion comp.)
  - Block size
  - Motion vectors
  - Quantizer step size
  - Suppression of DCT coefficients

- Solution
  - Embed rate-constrained motion estimation into mode decision with Lagrangian cost function
  - Couple Lagrange multiplier to quantizer step size

- Difficulties
  - Joint entropy coding of side information
  - Temporal dependencies due to DPCM structure
History of motion-compensated coding

- **Intraframe coding**: only spatial correlation exploited

- **Conditional replenishment**
  - H.120 [1984] (*DPCM, scalar quantization*)

- **Frame difference coding**
  - H.120 Version 2 [1988]

- **Motion compensation: integer-pel accurate displacements**
  - H.261 [1991]

- **Half-pel accurate motion compensation**

- **Variable block-size motion compensation**
  - H.263 [1996], MPEG-4 [1999]
Efficiency of motion-compensated coding

- **TMN-10** Variable block size motion compensation (H.263 1998)
- **Half-pel motion compensation** (MPEG-1 1993)
- **Frame difference coding** (H.120 1988)
- **Integer-pel motion compensation** (H.261 1991)
- **Intraframe DCT coding** (DCT 1974, JPEG 1992)

**Foreman**
10 Hz, QCIF
100 frames encoded

PSNR [dB] vs Bit-Rate [kbps] graph.
Efficiency of motion-compensated coding

- Variable block size motion compensation (H.263 1996)
- Conditional Replenishment (H.120)
- Intraframe DCT coding (JPEG)
- 60 %

Mother & Daughter
10 Hz, QCIF
100 frames encoded
Efficiency of motion-compensated coding

- **Variable block size motion compensation (H.263 1996)**: 35%
- **Integer-pel motion compensation (H.261 1991)**
- **Intraframe DCT coding (JPEG)**: 40%

**Mobile & Calendar**
- 10 Hz, QCIF
- 100 frames encoded

PSNR [dB]

Bit-Rate [kbps]

0 500 1000 1500

Bernd Girod: EE398B Image Communication II

Motion Compensated Coding no. 50