Hybrid Video Coding

- Principle of Hybrid Video Coding
- Motion-Compensated Prediction
- Bit Allocation
- Motion Models
- Motion Estimation
- Efficiency of Hybrid Video Coding

“It has been customary in the past to transmit successive complete images of the transmitted picture.”

[...]

“In accordance with this invention, this difficulty is avoided by transmitting only the difference between successive images of the object.”
Motion-Compensated Prediction

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

$$\hat{s}(x,y,t) = s'(x-d_x,y-d_y,t-\Delta t)$$

Motion-Compensated Prediction: Example

Frame 1 $s(x,y,t-1)$ (previous)  Frame 2 $s(x,y,t)$ (current)  Partition of frame 2 into blocks (schematic)

Frame 2 with displacement vectors  Difference between motion-compensated prediction and current frame $u(x,y,t)$  Referenced blocks in frame 1
Rate-constrained Motion Estimation

- Optimum trade-off: \( \frac{dD}{dR_m} = \frac{dD}{dR_u} \)

- Displacement error variance can be influenced via:
  - Block-size, quantization of motion parameters
  - Choice of motion model

Rate-Constrained Coder Control

- Efficiency increase via adding coding modes

- Problem can be posed as: minimize distortion \( D \) subject to a rate constraint \( R_c \)
  \[
  \min \{ D \} \quad \text{s.t.} \quad R < R_c
  \]

- Coder Control using Lagrangian optimization:
  \[\text{Solve an unconstrained minimization problem} \]
  \[
  \min \{ D + \lambda R \}
  \]

[Everett III 1963], [Shoham, Gersho 1988],
[Chou, Lookabaugh, Gray 1989]
Lagrangian Optimization in Video Coding

- A number of interactions are often neglected
  - Temporal dependency due DPCM loop
  - Spatial dependency of coding decisions
  - Conditional entropy coding

- Rate-Constrained Motion Estimation [Sullivan, Baker 1991]:

  \[
  \min \{ D_m + \lambda R_m \}
  \]

  Distortion after motion compensation  Lagrange parameter  Number of bits for motion vector

- Rate-Constrained Mode Decision [Wiegand, et al. 1996]:

  \[
  \min \{ D + \lambda R \}
  \]

  Distortion after reconstruction  Lagrange parameter  Number of bits for coding mode

Camera Models

- Natural camera-view scenes are projected from the 3-D world into the 2-D image plane
- A camera model can be used to describe projection, lens and sampling

**Perspective projection model**

\[
F
\]

Focal point

3D Position

Image plane

\[
x = \frac{F}{Z} X
\]

\[
y = \frac{F}{Z} Y
\]

**Orthographic projection model**

\[
X = X
\]

\[
y = Y
\]

Can be used when \(F=Z\) for all considered points

\[
x = \frac{F}{Z} X
\]

\[
y = \frac{F}{Z} Y
\]
Motion Models

- Motion in 3-D space corresponds to displacements in the image plane
- Motion compensation in the image plane is conducted to provide a prediction signal for efficient video compression
- Efficient motion-compensated prediction often uses side information to transmit the displacements
- Displacements must be efficiently represented for video compression
- Motion models relate 3-D motion to displacements assuming reasonable restrictions of the motion and objects in the 3-D world

\[
\begin{align*}
\text{Motion Model} \\
& d_x = x' - x = f(a, x, y), \quad d_y = y' - y = f(b, x, y) \\
& x, y : \text{location in previous image} \\
& x', y' : \text{location in current image} \\
& a, b : \text{vector of motion coefficients} \\
& d_x, d_y : \text{displacements}
\end{align*}
\]

Perspective Motion Model

- Mathematical model:

\[
\begin{align*}
& d_x = \frac{a_1 + a_2x + a_3y}{1 + c_1x + c_2y}, \quad d_y = \frac{b_1 + b_2x + b_3y}{1 + c_1x + c_2y}
\end{align*}
\]

- Restrictions:
  - Rotation and scaling of a rigid body in 3-D space, but no translation
  - Translation, rotation, and scaling of a planar patch in 3-D
- Advantage: corresponds to perspective projection model
- Disadvantage: hyperbolic motion function
Orthographic Motion Models

- Translational motion model: \( d_x = a_1 \), \( d_y = b_1 \)
- 4-Parameter motion model: translation, zoom (isotropic Scaling), rotation in image plane
  \[
  d_x = a_1 + a_2 x + a_3 y \\
  d_y = b_1 - a_3 x + a_2 y
  \]
- Affine motion model:
  \[
  d_x = a_1 + a_2 x + a_3 y \\
  d_y = b_1 + b_2 x + b_3 y
  \]
- Parabolic motion model
  \[
  d_x = a_1 + a_2 x + a_3 y + a_4 x^2 + a_5 y^2 + a_6 xy \\
  d_y = b_1 + b_2 x + b_3 y + b_4 x^2 + b_5 y^2 + b_6 xy
  \]
- Can also be viewed as Taylor expansions of perspective motion model

Impact of the Affine Parameters

- \( d_x = a_1 \rightarrow \) translation
- \( d_x = a_2 x \rightarrow \) scaling
- \( d_x = a_3 y \rightarrow \) shearing
Impact of the Parabolic Parameters

\[ d_x = a_x x^2 \]

\[ d_y = a_y y^2 \]

\[ d_x = a_x xy \]

Impact of the Perspective Parameters

\[ d_x = \frac{1}{1 + c_x x} \]

\[ d_y = \frac{1}{1 + c_y y} \]
**Differential Motion Estimation**

- Assume small displacements $d_x, d_y$:

$$u(x, y, t) = s(x, y, t) - \hat{s}(x, y, t, d_x, d_y)$$

$$= s(x, y, t) - s'(x, y, t - \Delta t) - \frac{\partial s'}{\partial x} d_x - \frac{\partial s'}{\partial y} d_y$$

Displacement frame difference

- Aperture problem: several observations required
- Inaccurate for displacements > 0.5 pel
  - multigrid methods, iteration
- Minimize

$$\min \sum_{x} \sum_{y} \hat{u}^2(x, y, t)$$

**Gradient-Based Affine Refinement**

- Displacement vector field is represented as

$$x' = a_1 + a_2 x + a_3 y$$

$$y' = b_1 + b_2 x + b_3 y$$

- Combination

$$u(x, y, t) = s(x, y, t) - s'(x, y, t - \Delta t) + \frac{\partial s'}{\partial x} (a_1 + a_2 x + a_3 y) + \frac{\partial s'}{\partial y} (b_1 + b_2 x + b_3 y)$$

yields a system of linear equations:

$$u = s - s' - \begin{pmatrix} \frac{\partial s'}{\partial x} & \frac{\partial s'}{\partial x} & \frac{\partial s'}{\partial x} \\ \frac{\partial s'}{\partial y} & \frac{\partial s'}{\partial y} & \frac{\partial s'}{\partial y} \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ a_3 \\ b_1 \\ b_2 \\ b_3 \end{pmatrix}$$

- System can be solved using, e.g., pseudo-inverse, minimizing

$$\arg \min_{a_1, a_2, a_3, b_1, b_2, b_3} \sum_{x} \sum_{y} \hat{u}^2(x, y, t)$$
**Principle of Blockmatching**

- Subdivide every image into square blocks
- Find one displacement vector for each block
- Within a search range, find a best "match" that minimizes an error measure.

**Blockmatching: Error Measure**

- Mean squared error (sum of squared errors)
  \[ \text{SSE}(d_x, d_y) = \sum_{x=1}^{Bx} \sum_{y=1}^{By} [s(x, y, t) - s'(x - d_x, y - d_y, t - \Delta t)]^2 \]

- Sum of absolute differences
  \[ \text{SAD}(d_x, d_y) = \sum_{x=1}^{Bx} \sum_{y=1}^{By} |s(x, y, t) - s'(x - d_x, y - d_y, t - \Delta t)| \]

- Approximately same performance
  \[ \Rightarrow \text{SAD less complex for some architectures} \]
Blockmatching: Search Strategies I

Full search
- All possible displacements within the search range are compared.

\[ \begin{align*}
&d_x \\
&d_y
\end{align*} \]

- Computationally expensive
- Highly regular, parallelizable

Block matching: 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

\[ \begin{align*}
&d_x \\
&d_y
\end{align*} \]

from: Girod
Block matching: Search Strategies III

Computational Complexity

<table>
<thead>
<tr>
<th>Method</th>
<th>Block comparisons</th>
</tr>
</thead>
<tbody>
<tr>
<td>2D logarithmic</td>
<td>18, 21</td>
</tr>
<tr>
<td>full search</td>
<td>169, 169</td>
</tr>
</tbody>
</table>

Example: max. horizontal, vertical displacement = 6
integer-pel accuracy

- a - for special vector (2,6)
- b - worst case

Block Comparison Speed-Ups

- Triangle inequalities for SAD and SSE

\[
\sum_{\text{block}} |S_x - S_{x,1}| \geq \frac{1}{N} \sum_{\text{block}} (S_x - S_{x,1})^2 \geq \frac{1}{N} \sum_{\text{block}} S_x - \sum_{\text{block}} S_{x,1}^2
\]

- Strategy:
  1. Compute partial sums for blocks in current and previous frame
  2. Compare blocks based on partial sums
  3. 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
  1. Sum over 16x16 block
  2. Row wise block projection
Sub-pel Translational Motion

- Motion vectors are often not restricted to only point into the integer-pel grid of the reference frame
- Typical sub-pel accuracies: half-pel and quarter-pel
- Sub-pel positions are often estimated by "refinement"

```plaintext
d x

\[ \begin{array}{cccc}
  & 1 & 2 & 3 \\
 1 & & & \\
 2 & & & \\
 3 & & & \\
\end{array} \]
```

History of Motion Compensation

- **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 (16x16 & 8x8) motion compensation**
  - H.263 [1996], MPEG-4 [1999]
- **Variable block-size (16x16 – 4x4) and multi-frame motion compensation**
  - H.26L [2001]

Complexity increase
Milestones in Video Coding

Variable block size (16x16 – 4x4) + quarter-pel + multi-frame motion compensation (H.26L, 2001)

Variable block size (16x16 – 8x8) (H.263, 1996) + quarter-pel motion compensation (MPEG-4, 1998)

Half-pel motion compensation (MPEG-1 1993 MPEG-2 1994)

Frame Difference coding (H.120 1988)

Intraframe DCT coding (JPEG, 1990)

Foreman
10 Hz, QCIF
100 frames

Intraframe
DCT coding
(JPEG, 1990)

Half-pel motion compensation (MPEG-1 1993 MPEG-2 1994)

Conditional Replenishment (H.120)

Integer-pel motion compensation (H.261, 1991)

Integer-pel motion compensation (H.261, 1991)

Visual Comparison

Bit-rate Reduction: 75%

Frame
Difference
coding
(H.120 1988)

Foreman
10 Hz, QCIF
100 frames

Intraframe
DCT coding
(JPEG, 1990)

Half-pel motion compensation (MPEG-1 1993 MPEG-2 1994)

Conditional Replenishment (H.120)

Integer-pel motion compensation (H.261, 1991)
Summary

- Video coding as a hybrid of motion compensation and prediction error coding
- Lagrangian bit-allocation rules specify constant slope allocation to motion coefficients and prediction error
- Motion models can represent various kinds of motions
- In practice: affine or 8-parameter model for camera motion, translational model for small blocks
- Differential methods calculate displacement from spatial and temporal differences in the image signal.
- Block matching computes error measure for candidate displacements and finds best match.
- Speed up block matching by search methods and by clever application of triangle inequality
- Hybrid video coding has been drastically improved by enhanced motion compensation capabilities