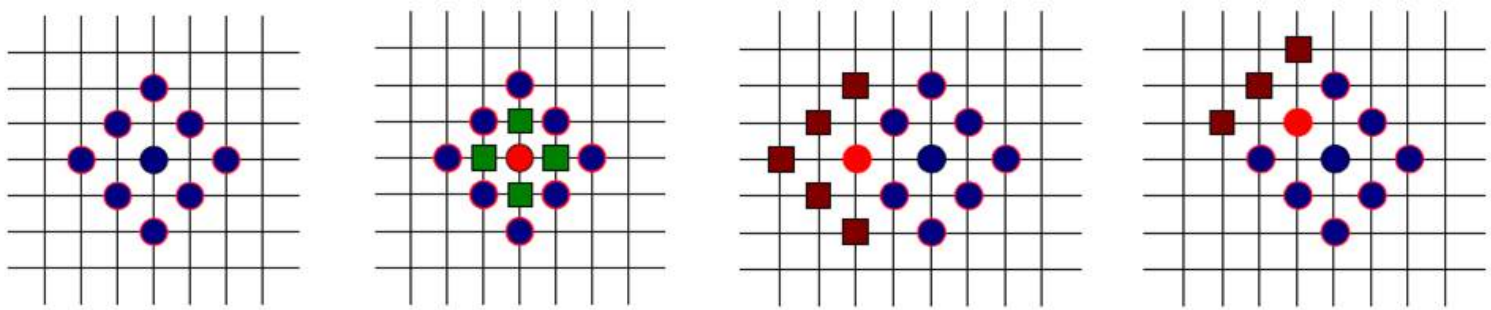
Motion Parameter Coding and Estimation



Coding of Motion Parameters

Coding parameters for inter-picture coding modes

- Number of motion hypotheses or prediction type
- Reference index for each hypothesis
- Motion vector for each hypothesis

Conventional inter coding modes

- Transmit prediction type and reference index (unless it can be inferred)
- Transmit motion vector difference

$$(\Delta m_x, \Delta m_y) = (m_x, m_y) - (\widehat{m}_x, \widehat{m}_y)$$

➡ Coding efficiency depends on choice of motion vector predictor $(\widehat{m}_x, \widehat{m}_y)$

Coding modes with inferred motion parameters

- All motion parameters are derived at the decoder side
- Exploit data of already coded blocks (spatial & temporal neighbors)

➡ Suitable for consistently moving regions

Motion Vector Prediction

Simple variant

- H.262 | MPEG-2 Video
- Use motion vector of left block as predictor

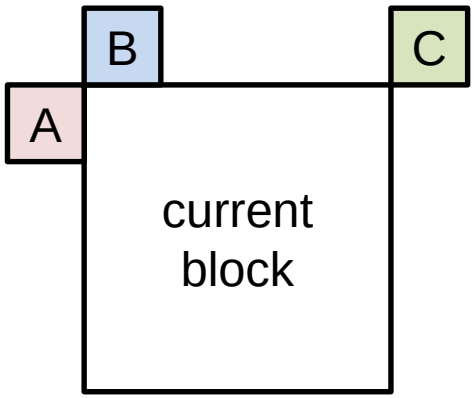
$$\widehat{m}_x = m_x^{(\text{left})}$$
$$\widehat{m}_y = m_y^{(\text{left})}$$

➡ Large motion vector differences at object boundaries

Median prediction

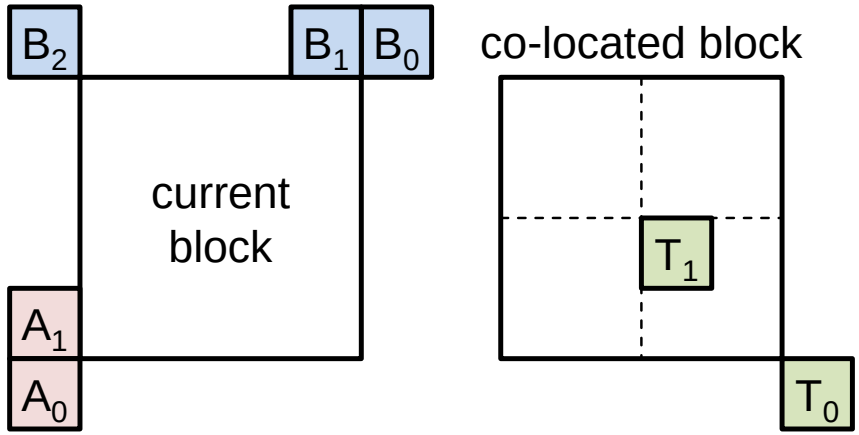
- H.263, MPEG-4 Visual, H.264 | MPEG-4 AVC
- Component-wise median of three neighboring blocks

$$\widehat{m}_x = \text{median}\left(m_x^{\text{A}}, m_x^{\text{B}}, m_x^{\text{C}}\right)$$
$$\widehat{m}_y = \text{median}\left(m_y^{\text{A}}, m_y^{\text{B}}, m_y^{\text{C}}\right)$$



➡ On average, smaller motion vector differences

Motion Vector Prediction



Switched motion vector prediction (H.265 | MPEG-H HEVC)

- Idea: Adaptively choose most suitable neighboring motion vector
- Construct list of spatially and temporal neighboring motion vectors
- Select best candidate predictor
- Transmit index into candidate list in addition to motion vector difference
- H.265 | MPEG-H HEVC: Two candidate predictors

➡ Bit rate saving for motion vector differences is typically larger than bit rate required for signaling the chosen predictor

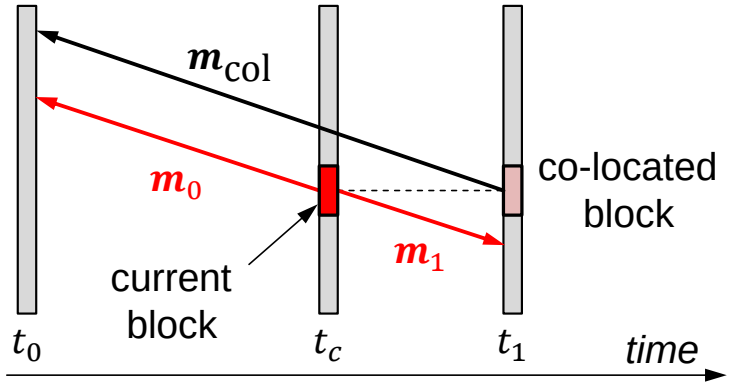
Coding Modes with Inferred Motion Parameters

Temporal direct mode

- Motion parameters for bi-prediction
- Scale motion vector $\boldsymbol{m}_{\mathrm{col}}$ of co-located block according to time differences

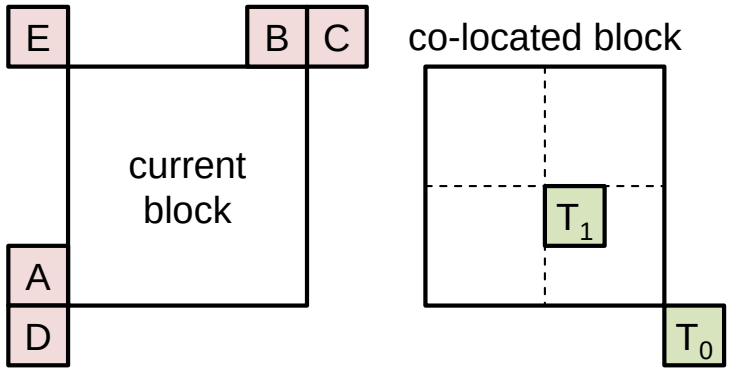
$$\boldsymbol{m}_0 = \frac{t_c - t_0}{t_1 - t_0} \cdot \boldsymbol{m}_{\mathrm{col}}$$

$$\boldsymbol{m}_1 = \frac{t_c - t_1}{t_1 - t_0} \cdot \boldsymbol{m}_{\mathrm{col}}$$

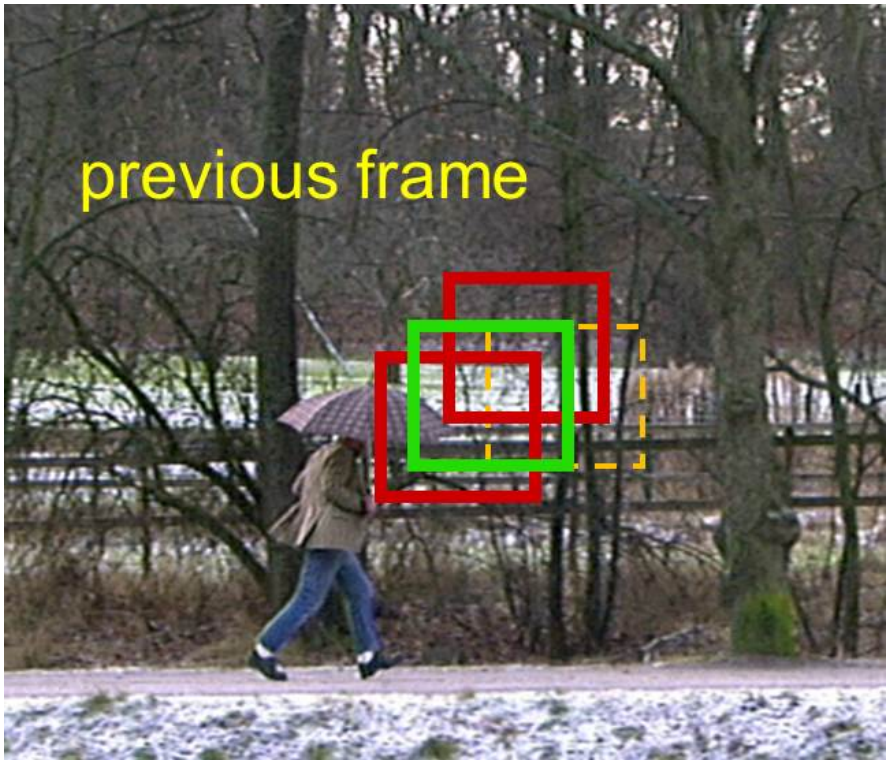


Merge mode (H.265 | MPEG-H HEVC)

- Similar concept as switched motion vector prediction
- Candidate list with motion parameters of spatial and temporal neighbors
- Select best candidate
- Transmit index into candidate list



Motion Estimation — Block Matching Algorithm



The measurement window is compared with **different shifted blocks** in the reference frame and the **best match** is determined

The considered block of samples in the current frame is selected as a measurement window

Lagrangian Motion Estimation

Lagrangian Cost Measure

- Mode decision concept (with transform coding of residual) is too complex for hundreds or thousands of motion vector candidates

➡ Assume that reconstructed prediction error signal is equal to zero

➡ Select motion vector $\boldsymbol{m}$ in search range $\mathcal{M}$ according to

$$\boldsymbol{m}_k^* = \arg\min_{\boldsymbol{m}\in\mathcal{M}_k} D_k(\boldsymbol{m}) + \lambda_{\mathrm{M}}\, R_k(\boldsymbol{m})$$

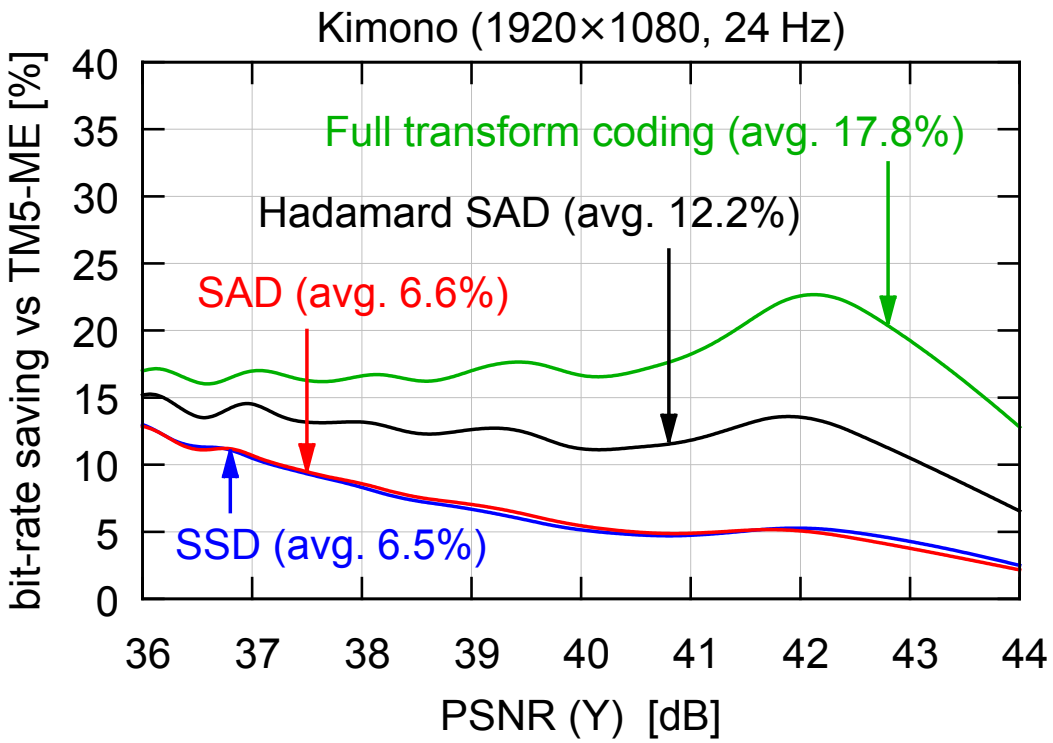
with

$D_k(\boldsymbol{m})$ – Distortion between original block $\boldsymbol{s}_k$ and prediction signal $\hat{\boldsymbol{s}}_k$
$R_k(\boldsymbol{m})$ – Number of bits for transmitting the motion vector $\boldsymbol{m}$
λ_M – Lagrange multiplier (depends on chosen distortion measure)

Distortion measure

- SSD ($\lambda_M = \lambda$) or SAD ($\lambda_M = \sqrt{\lambda}$) $\Longrightarrow$ SAD is often faster to compute
- SAD after Hadamard transform (better approximation of real RD costs)

Lagrangian Motion Estimation — Cost Measure



Compare exhaustive Lagrangian motion search with TM5 approach

- Full transform coding (similar to mode decision) $\Longrightarrow$ very complex
- SSD between original and prediction signal
- SAD between original and prediction signal
- Hadamard SAD between original and prediction signal

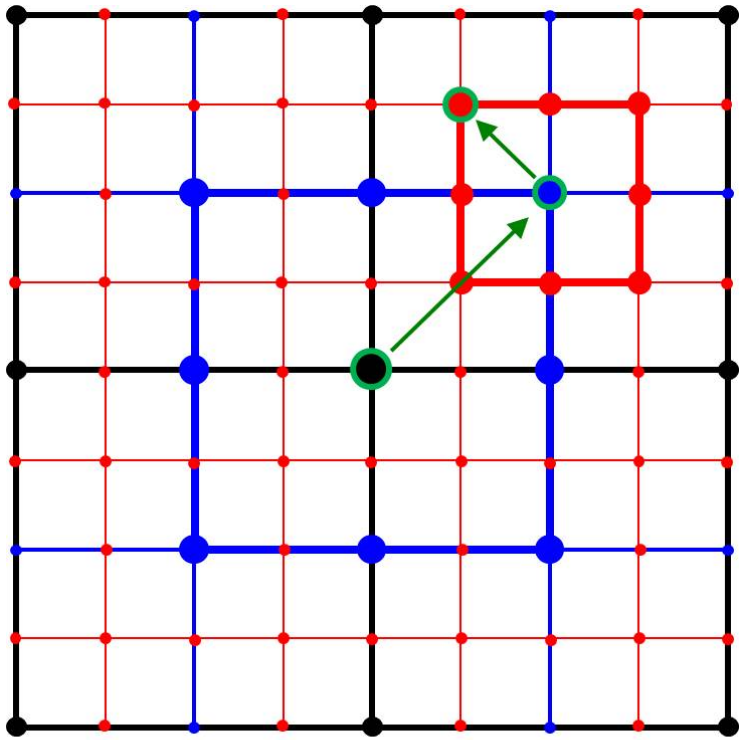
Search Strategy

Sub-sample accurate motion vectors

- Sub-sample locations: Interpolation
- Split search into integer-sample search and sub-sample refinement(s)
- Use simpler distortion measure for integer-sample search

Fast integer-sample search

- Reduce number of tested candidates
- Multiple strategies possible
  - ➡ Three-step search
  - ➡ Logarithmic search
  - ➡ Conjugate directional search
  - ➡ Enhanced predictive zonal search
  - ➡ ...

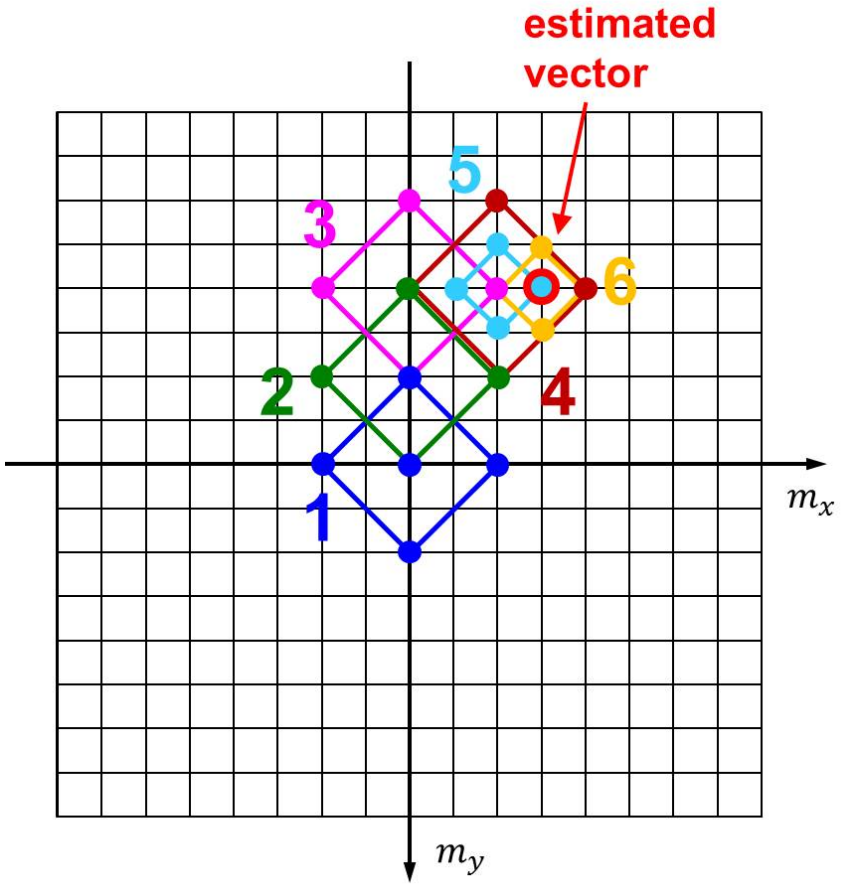


- integer-sample positions
- half-sample positions
- quarter-sample positions

Fast Search Strategies: Logarithmic Search

Logarithmic search [Jain, Jain, 1981]

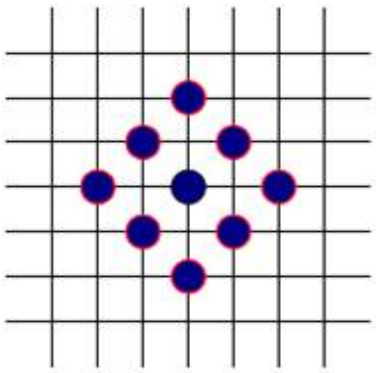
- Iterative comparison of the cost measures at 5 points (corners and center) of a diamond-shaped pattern
- Move pattern so that pattern is centered around best match
  - ➡ No more than 3 new candidates
- Logarithmic refinement of search pattern (4 new candidates) if
  - ➡ Best match is in center of pattern
  - ➡ Or best match is at the border of the search range
- Motion search is terminated if
  - ➡ Best match is in center of pattern
  - ➡ And smallest pattern size is used



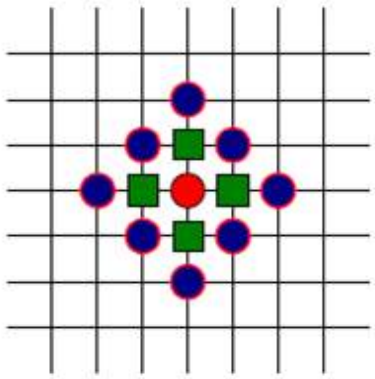
Fast Search Strategies: Diamond Search

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

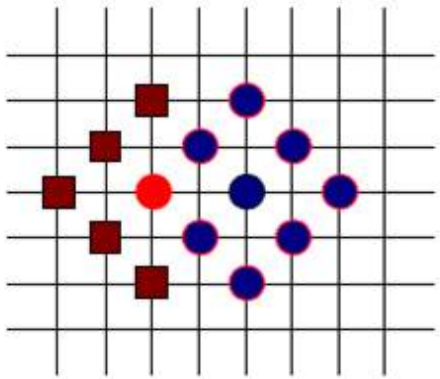
- Iterative search with 9 points of a diamond pattern
- Similar search strategy as logarithmic search



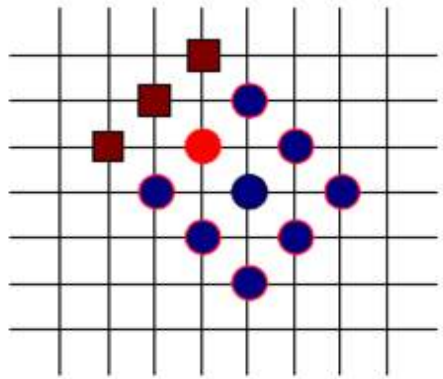
Start with large diamond pattern at motion vector (0,0) or at a predicted vector



If best match is in the center of a large diamond, then proceed with a smaller diamond



If best match does not lie in the center of the diamond pattern, center next diamond pattern at the best match



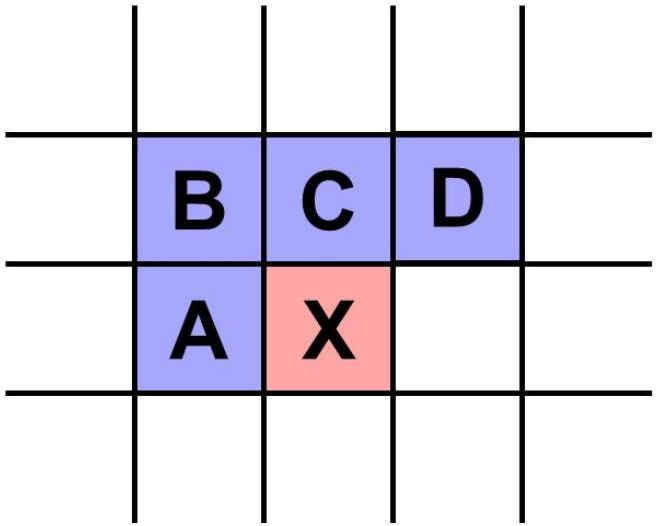
Fast Search Strategies: Choosing of Start Point

Non-adaptive choices of start point

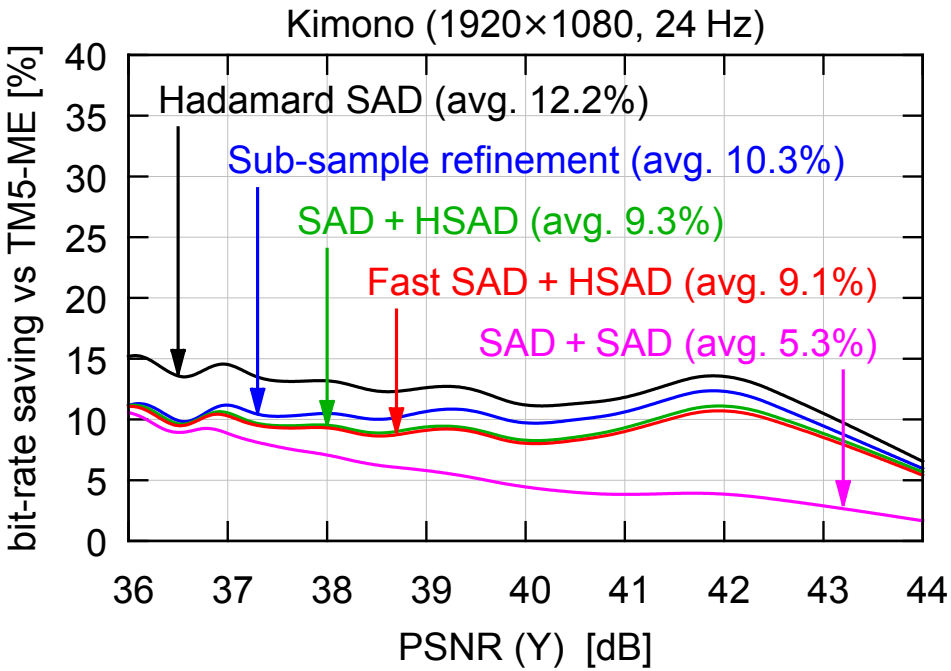
- Use motion vector (0,0) as start point of motion search
  - ➜ Suitable for applications like video conferencing
- Use motion vector predictor as start point for motion search
  - ➜ Typically results in faster termination of motion search

Adaptive choice of start point

- General idea: Motion of a block is similar to at least one of the neighboring blocks
- First evaluate the motion vectors of the already estimated neighboring blocks
  - Example: Blocks A, B, C and D
  - Candidates can also include a temporally predicted motion vector
- Choose best match among the candidates as start point of the motion search



Motion Estimation — Search Strategy



TM5 approach:

- Integer search with SAD
- Sub-sample refinement with SAD
- Cost measure: Distortion (without a rate term)

Compare different search strategies with TM5 approach

- Exhaustive search (all sub-sample locations) with Hadamard SAD
- Exh. integer search + sub-sample refinement (both with Hadamard SAD)
- Exh. integer search with SAD + sub-sample refinement with Hadamard SAD
- Exh. integer search with SAD + sub-sample refinement with SAD
- Fast integer search with SAD + sub-sample refinement with Hadamard SAD

Motion Estimation — Summary

Require significant complexity reduction compared to mode decision

- Neglect dependencies between motion vectors and transform coefficient levels
- ➡ Assume residual signal equal to zero during motion estimation
- Split motion search into integer-sample search and sub-sample refinement
- Apply fast search strategies

Configuration with reasonable trade-off between coding efficiency and complexity

- Fast integer-sample search with Lagrangian cost
  - Distortion: SAD between original and prediction signal
  - Rate: Number of bits required for transmitting motion vectors
  - Fast search strategy: HM approach (combination of different concepts)
- Sub-sample refinement with Lagrangian cost
  - Distortion: Hadamard SAD between original and prediction signal
  - Rate: Number of bits required for transmitting motion vectors

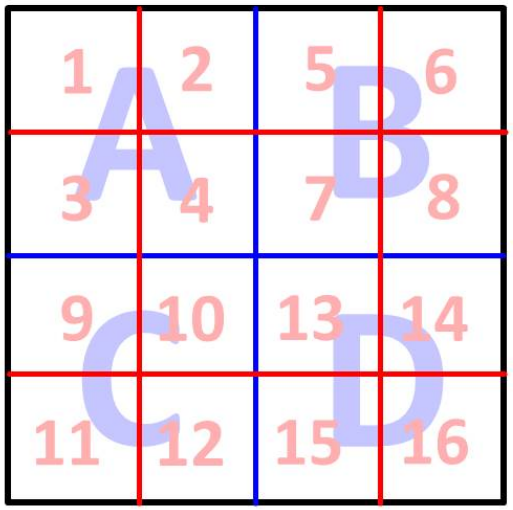
Block Size Selection

Lagrangian Mode Decision

- Determine Lagrangian costs $D + \lambda R$ for different partitioning options
- Choose option with minimum cost
- Quadtree partitioning: Depth-first order

Example: Two quadtree levels for a 16×16 block

1. Select best partitioning for first 8×8 blocks A
2. Select best partitioning for second 8×8 blocks B
3. Select best partitioning for third 8×8 blocks C
4. Select best partitioning for fourth 8×8 blocks D
5. Choose between 16×16 block and sub-division



Fast Mode Decision

- General idea: Skip evaluation of unlikely partitionings
- Typical: Top-down approach with stop criterion
- Stop when certain "quality criterion" is met

Part Summary

Motion parameter coding

- Large impact on coding efficiency: Motion vector prediction
- Median prediction
- Switched prediction improves coding efficiency

Inferred motion parameters

- Special modes with reduced number of bits for motion parameters
- Temporal direct mode
- Merge mode

Encoder control

- Lagrangian motion estimation
- Fast motion search strategies
- Lagrangian mode decision for block size selection
- Lagrangian mode decision for inter/intra selection