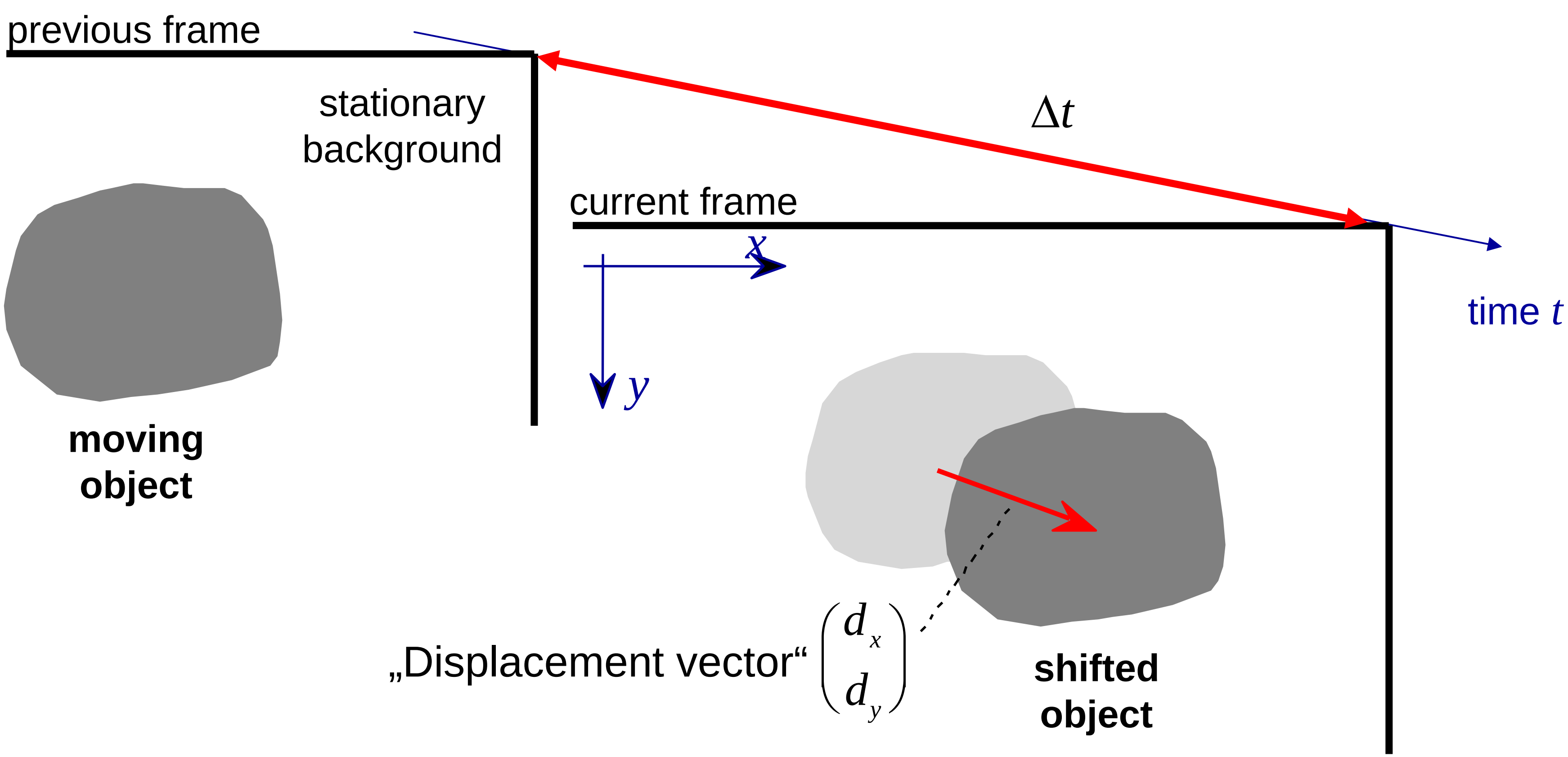
Overview: motion-compensated coding

- 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

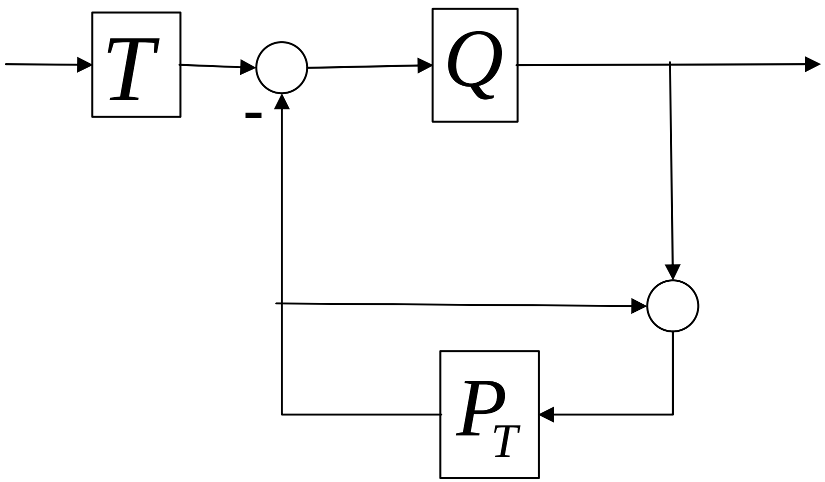


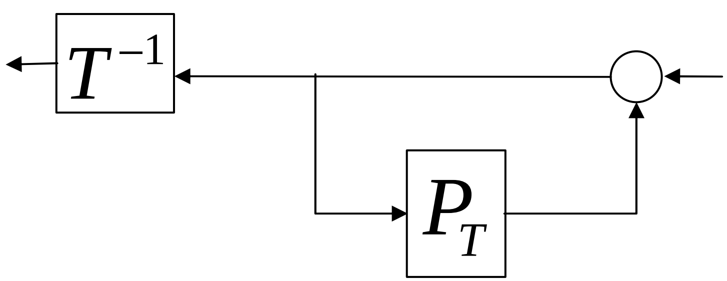
Prediction for the luminance signal $S(x,y,t)$ within the moving object:

$$\boxed{\hat{S}(x,y,t) = S(x-d_x\,,y-d_y\,,t-\Delta t)}$$

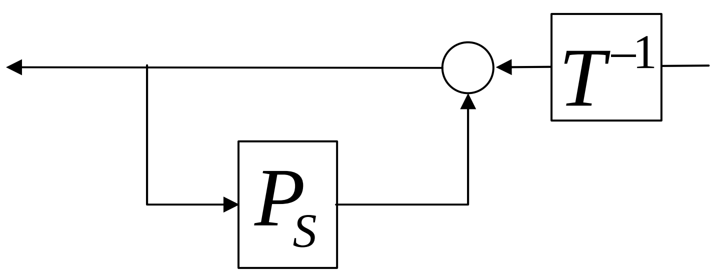
Combining transform coding and prediction

Transform domain prediction

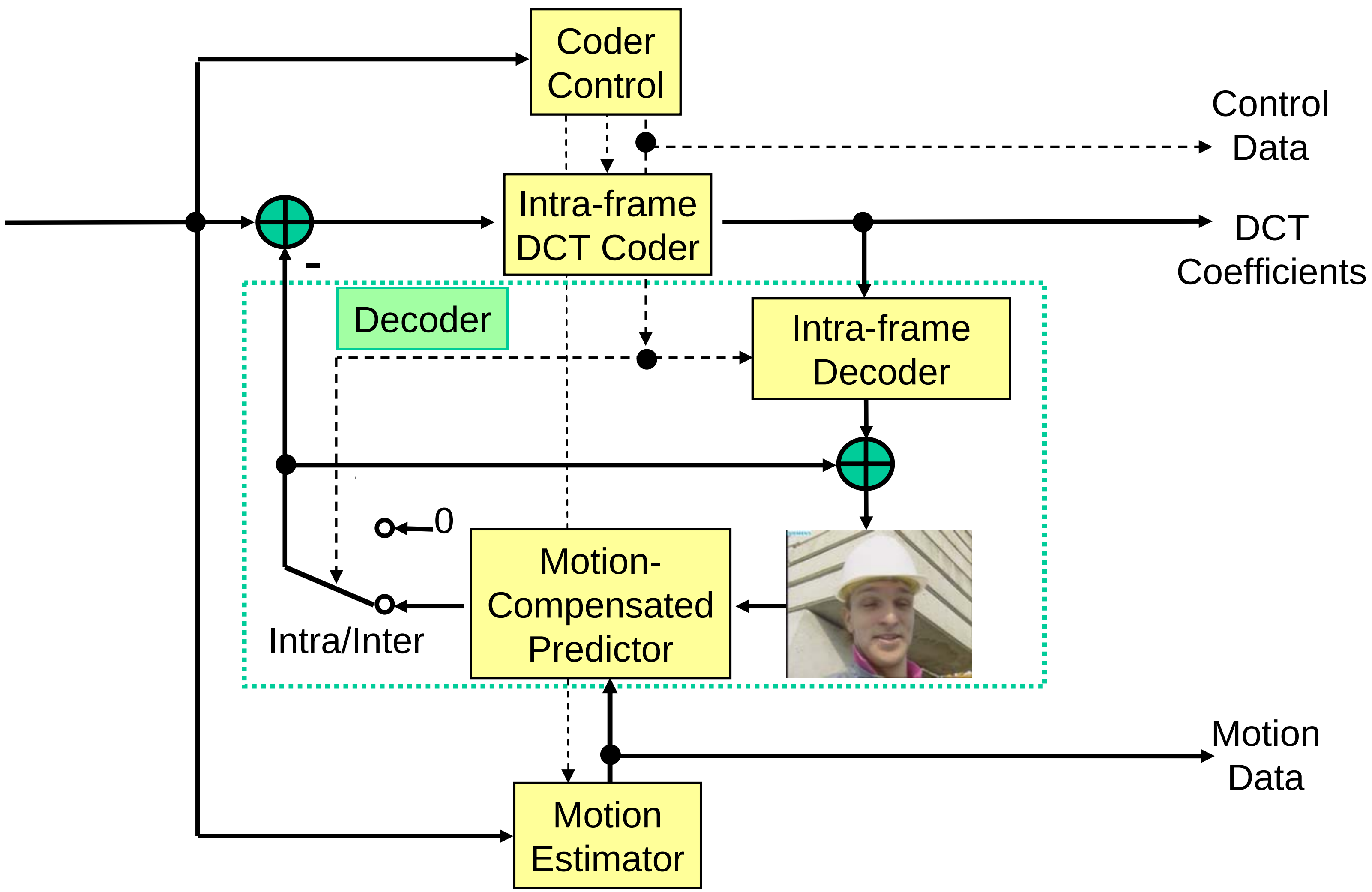




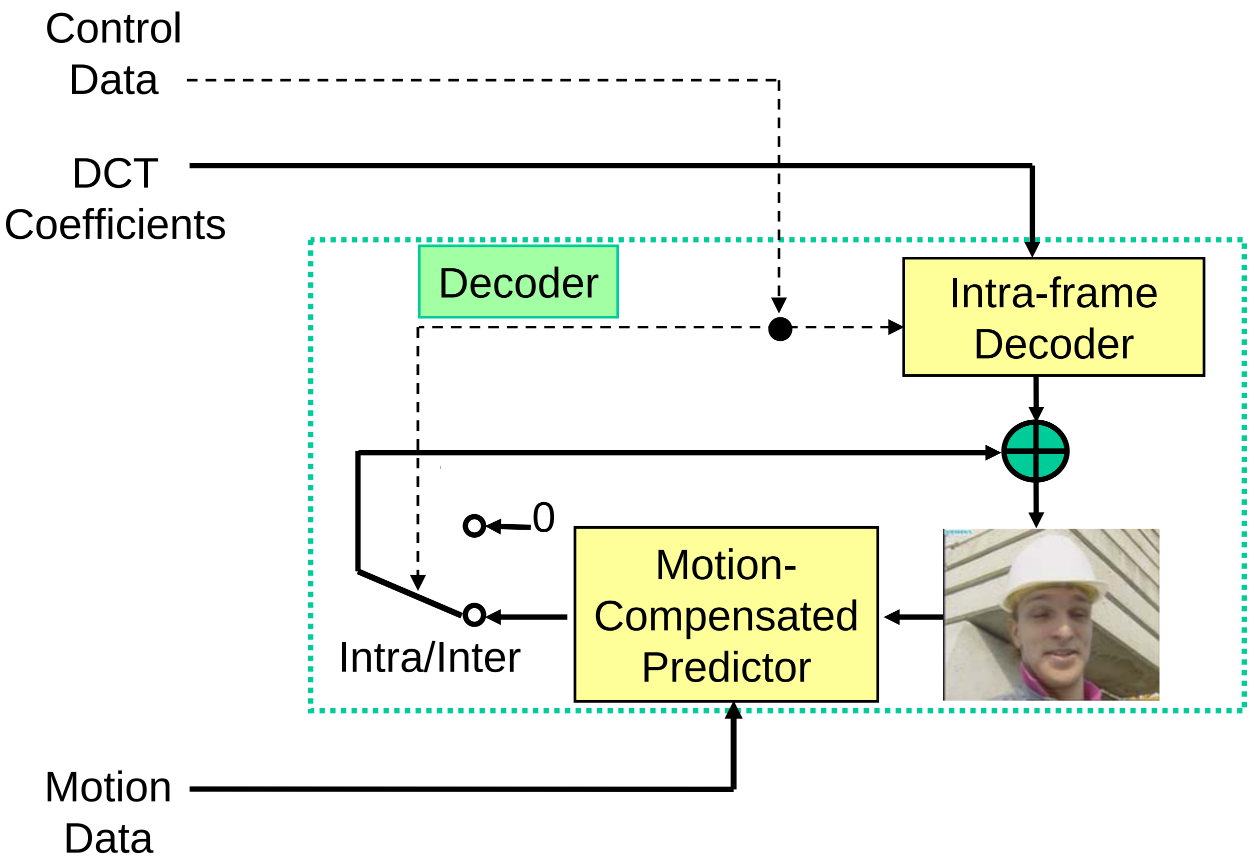
Space domain prediction



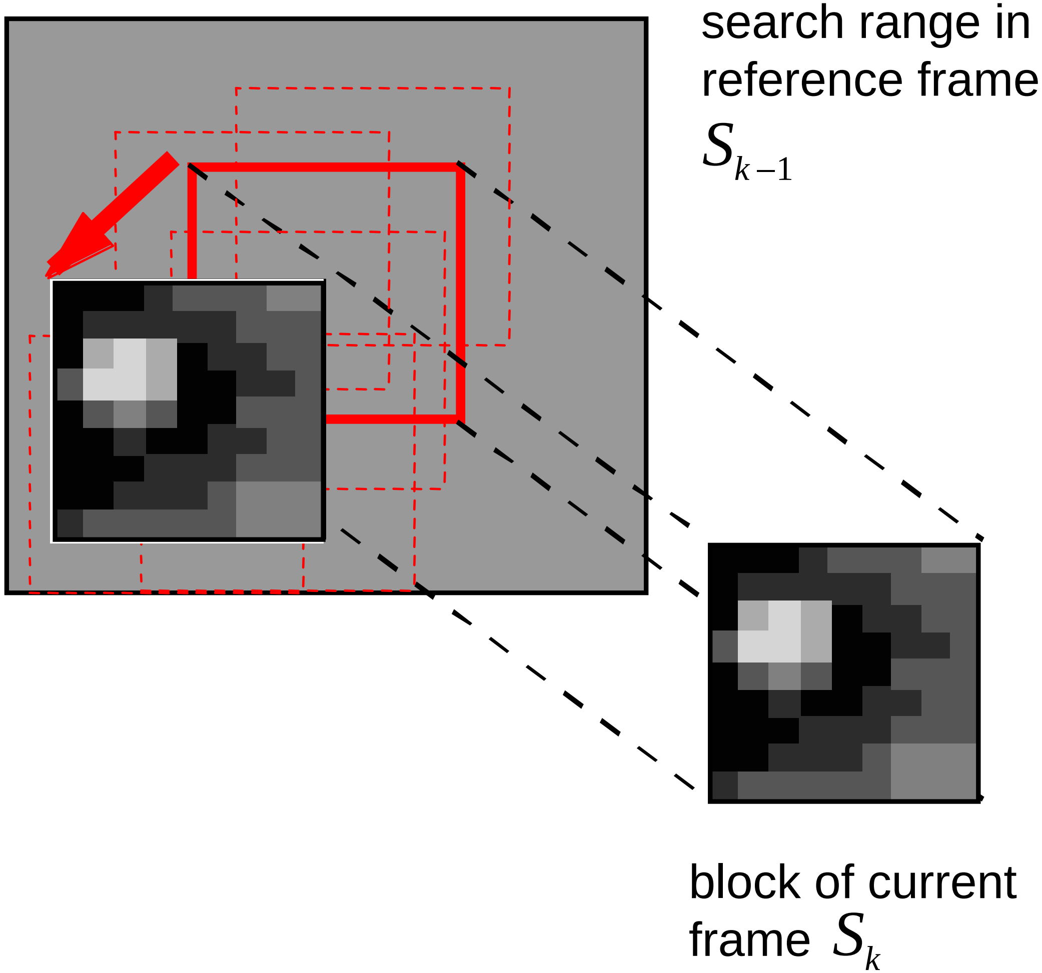
Motion-compensated hybrid coder



Motion-compensated hybrid decoder



Block-matching algorithm



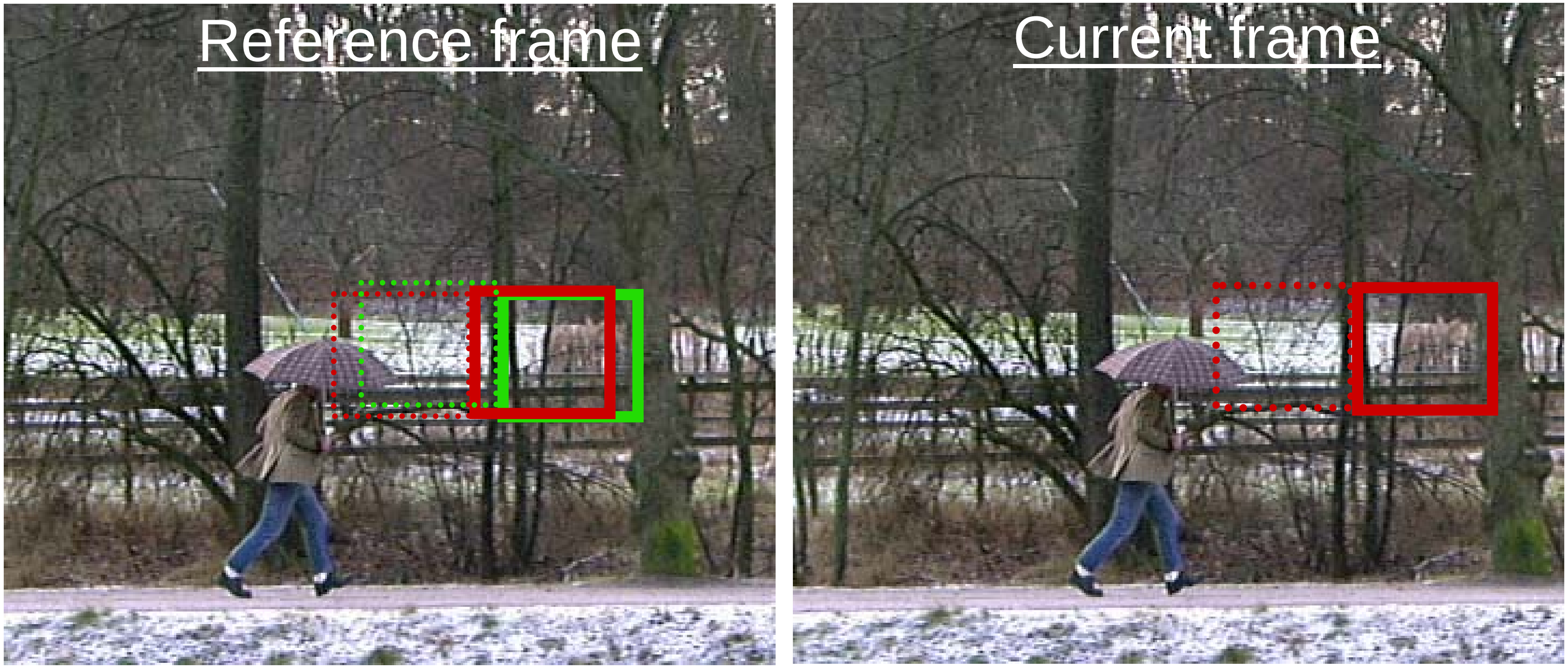
- Subdivide current frame into blocks.
- Find one displacement vector for each block.
- Within a search range, find a best „match“ that minimizes an error measure.
- Intelligent search strategies can reduce computation.

Block-matching algorithm

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



. . . process repeated for the next block

Blockmatching: Matching Criterion

- *Sum of Squared Differences* to determine similarity

Sum of all values in block

Current frame

Reference frame

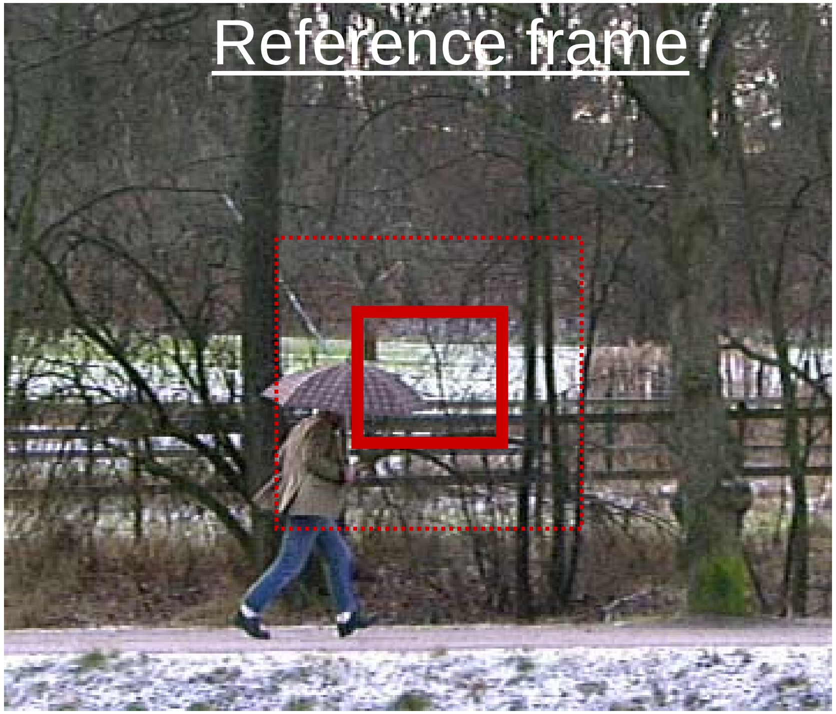
$$SSD(d_x, d_y) = \sum_{x,y \in \text{Block}} \left[S_k(x,y) - S_{k-1}(x + d_x, y + d_y) \right]^2$$

Horizontal shift

Vertical shift

- Alternative matching criteria: SAD (*Sum of Absolute Differences*), cross correlation, . . .
- Only integer pixel shifts (so far)

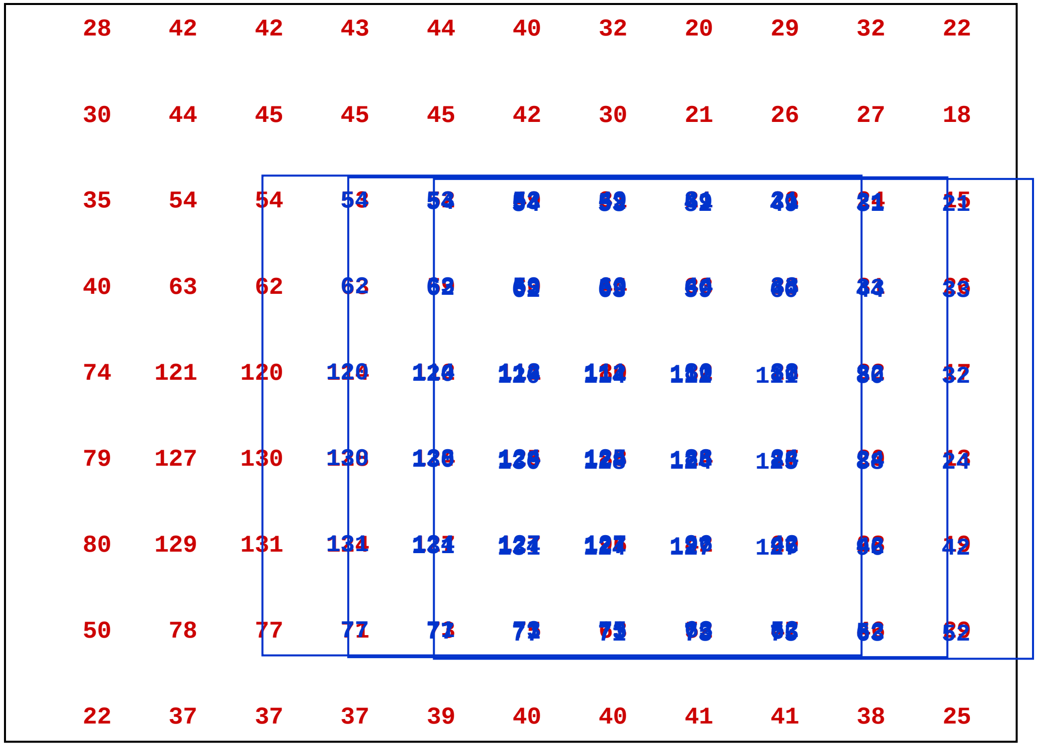
Integer Pixel Shifts



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

Block of pixels is considered

Integer Pixel Shifts

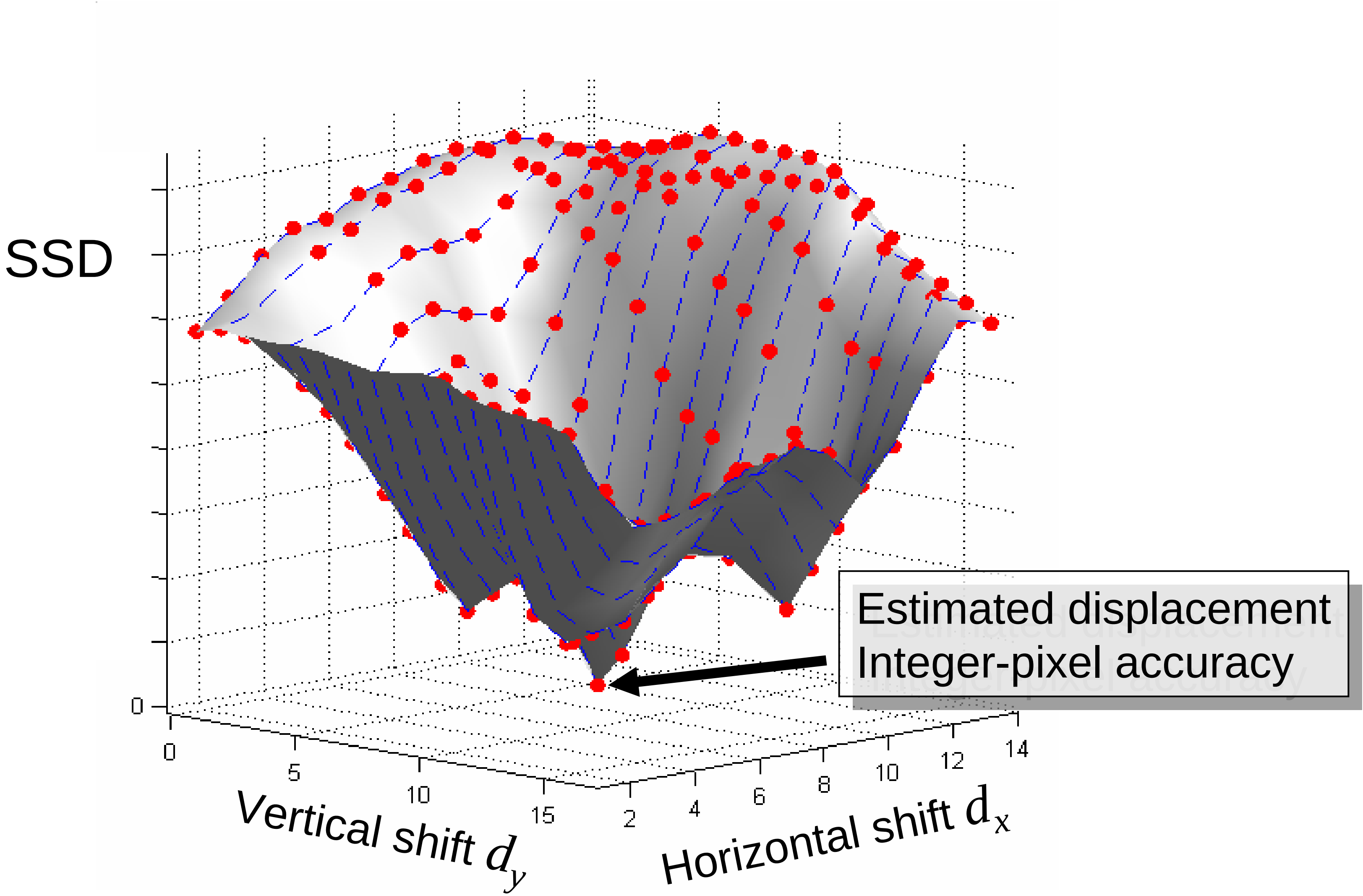


Block is compared with a shifted array of pixels in the reference frame to determine the best match

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

Block of pixels is considered

SSD Values Resulting from Blockmatching

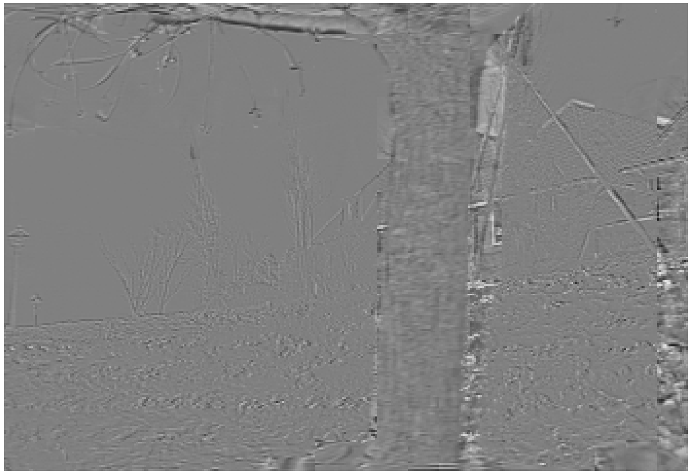


Motion-compensated prediction: example

Previous frame

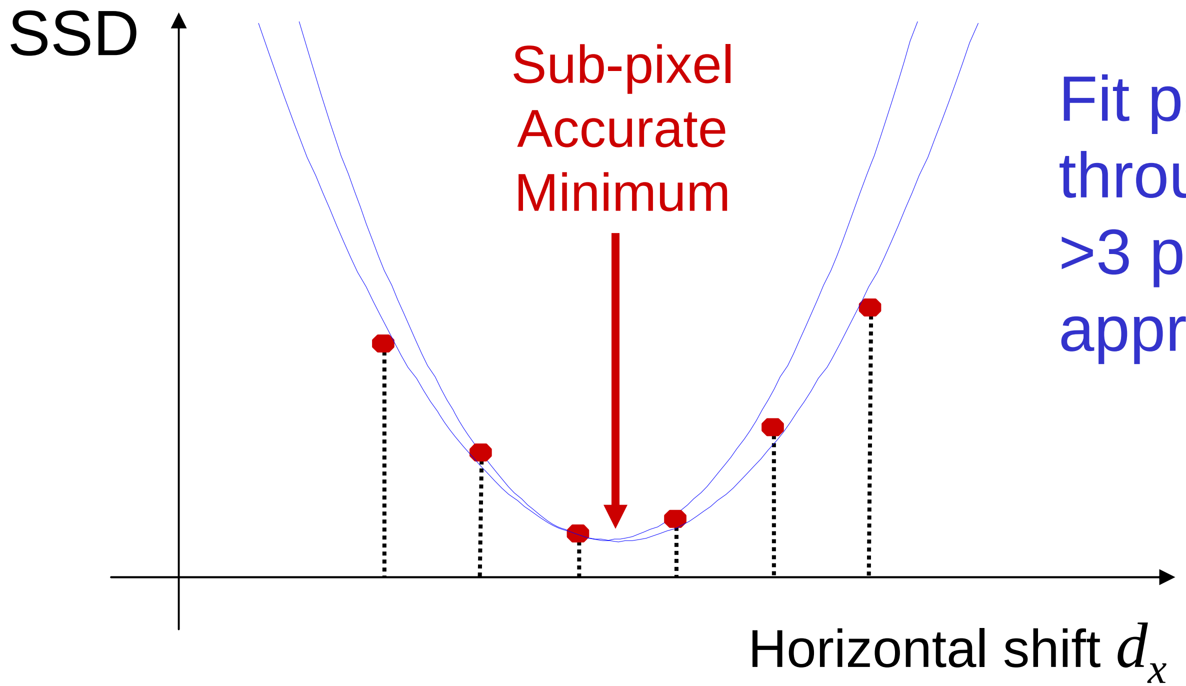
Current frame

Current frame with displacement vectors



Motion-compensated Prediction error

Interpolation of the SSD Minimum

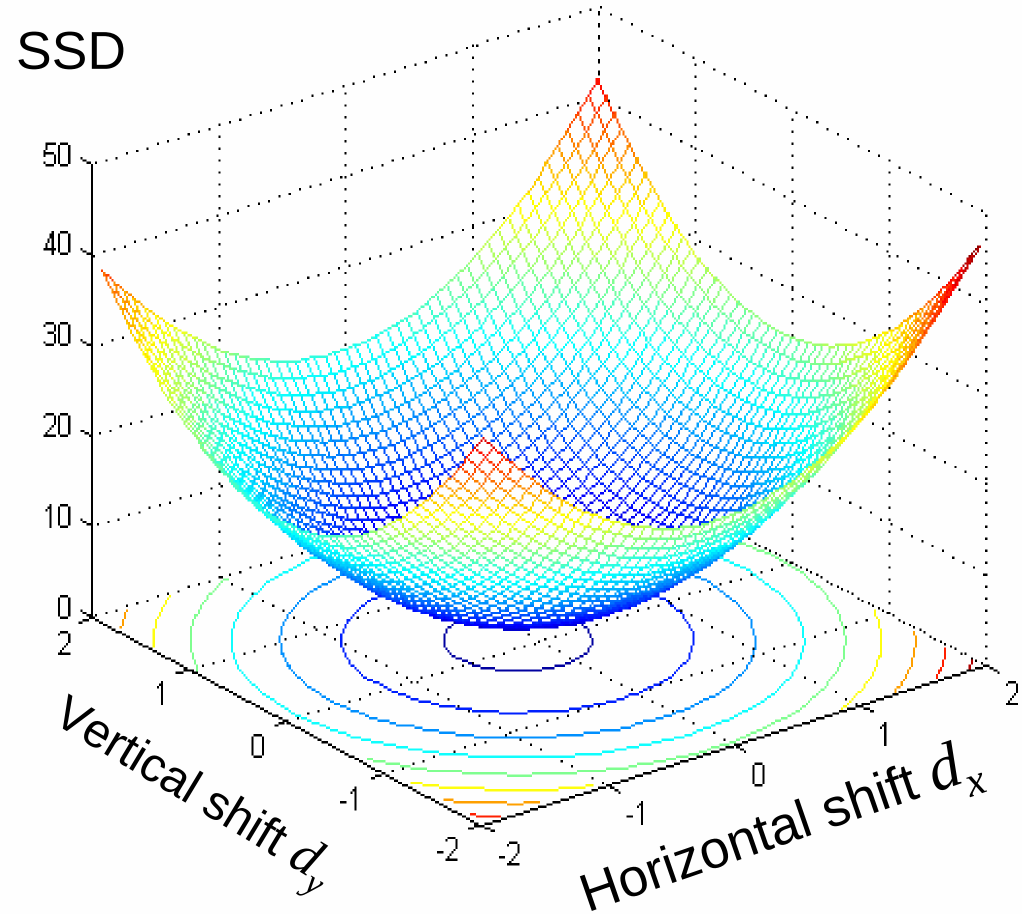


Fit parabola through >3 points approximately

2-d Interpolation of SSD Minimum

<u>Paraboloid</u>

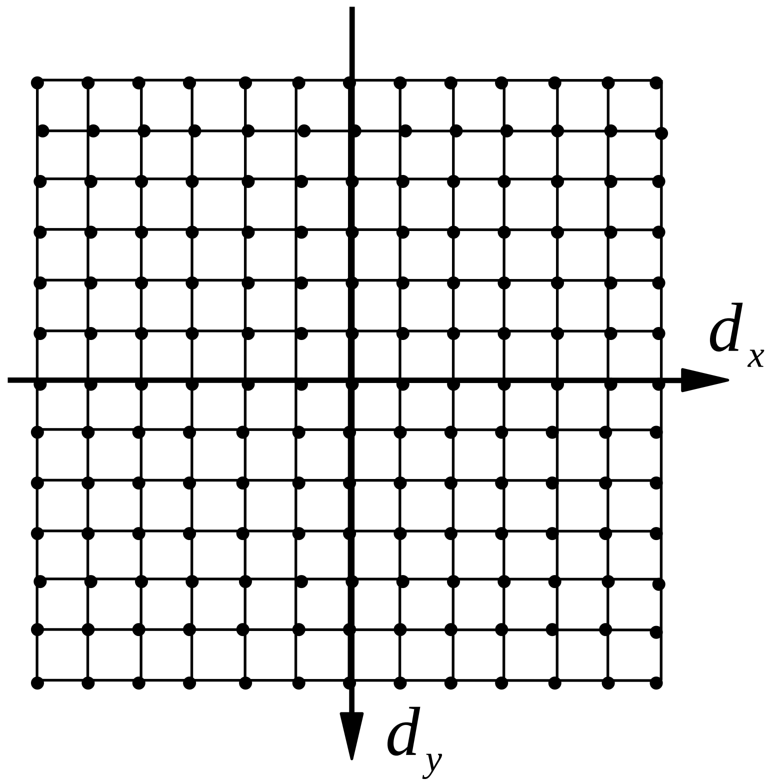
- Perfect fit through 6 points
- Approximate fit through >6 points



Blockmatching: search strategies I

Full search

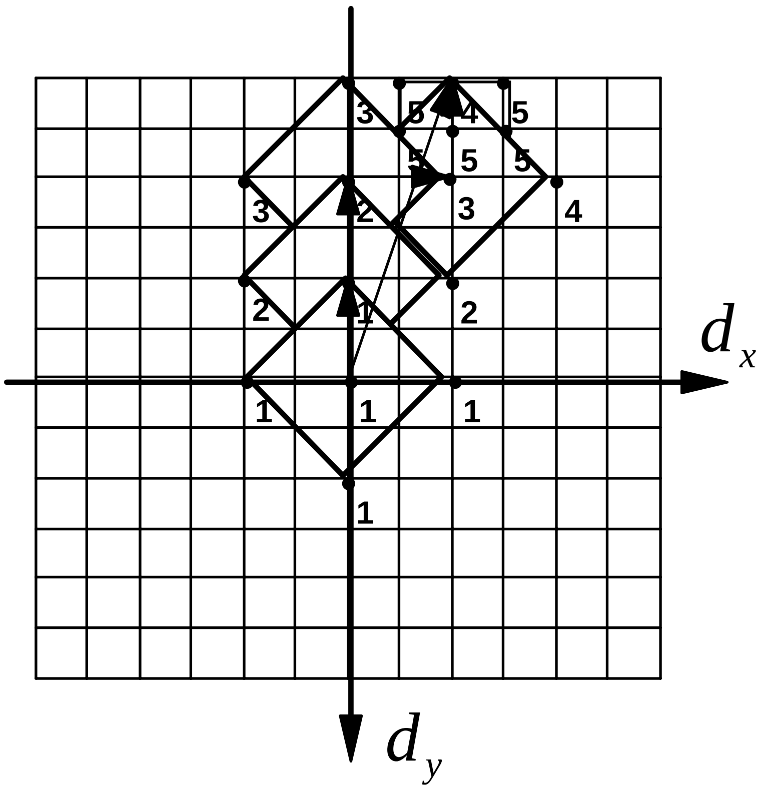
- All possible displacements within the search range are compared.
- Computationally expensive
- Highly regular, parallelizable



Blockmatching: search strategies II

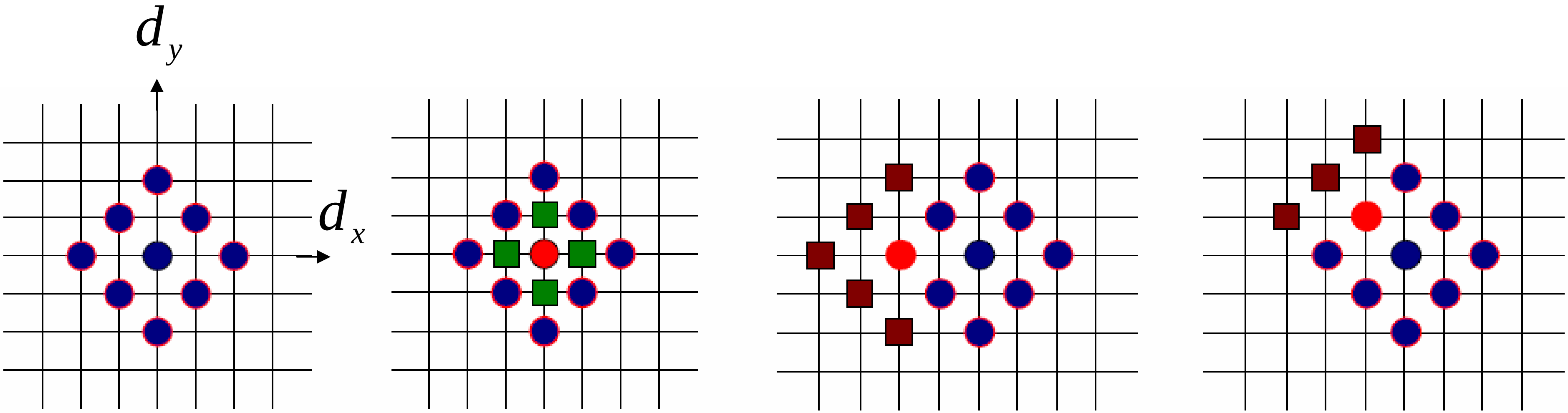
2D logarithmic search [Jain + Jain, 1981]

- 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

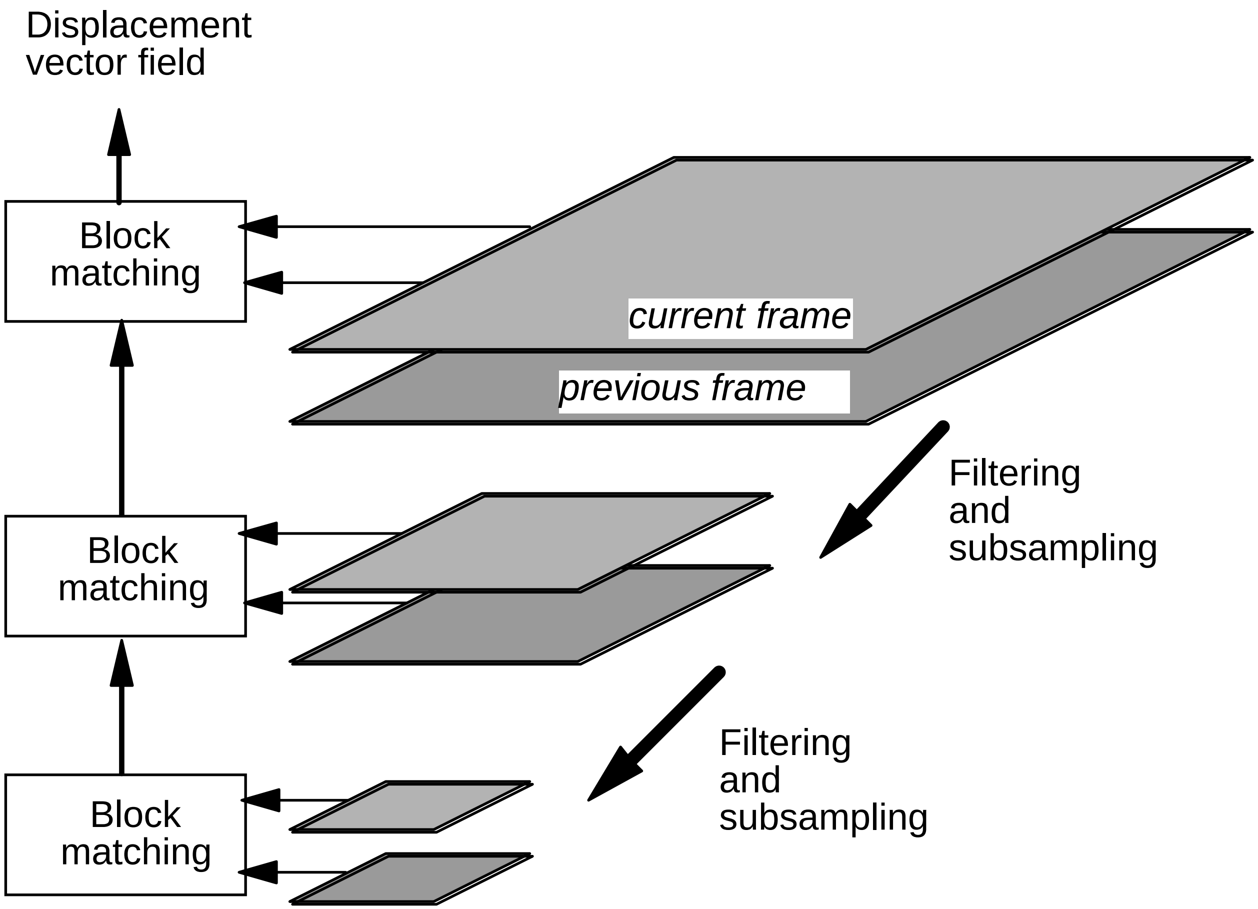
- Triangle and Cauchy-Schwarz inequality for SAD and SSE

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

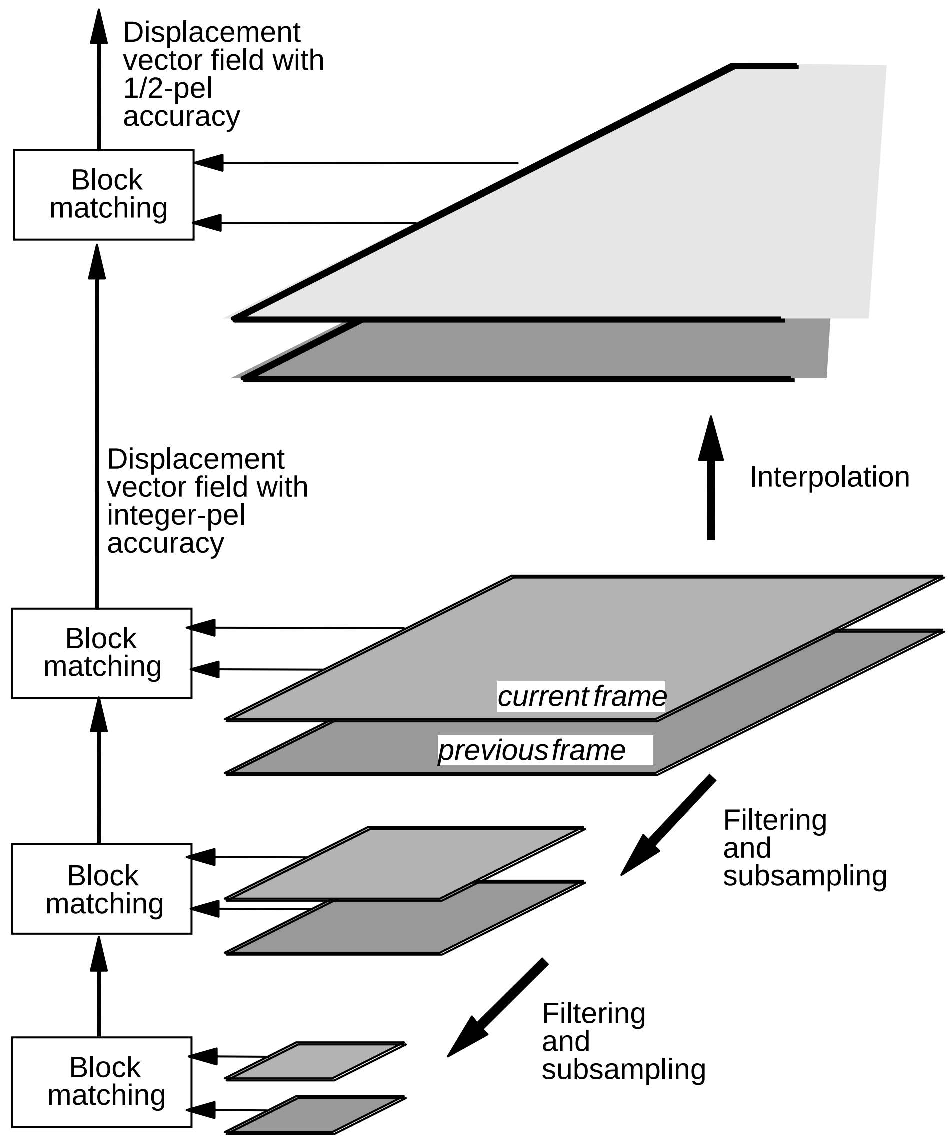
$$\sum_{\text{block}} \left(S_k - S_{k-1} \right)^2 \geq \frac{1}{\underset{\text{number of terms in sum}}{N}} \left(\sum_{\text{block}} S_k - S_{k-1} \right)^2 = \frac{1}{\underset{\text{number of terms in sum}}{N}} \left(\sum_{\text{block}} S_k - \sum_{\text{block}} S_{k-1} \right)^2$$

- 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 *[Lin + Tai, IEEE Tr. Commun., May 97]*
  - Sum over 16x16 block
  - Row wise block projection
  - Column wise block projection

Hierarchical blockmatching

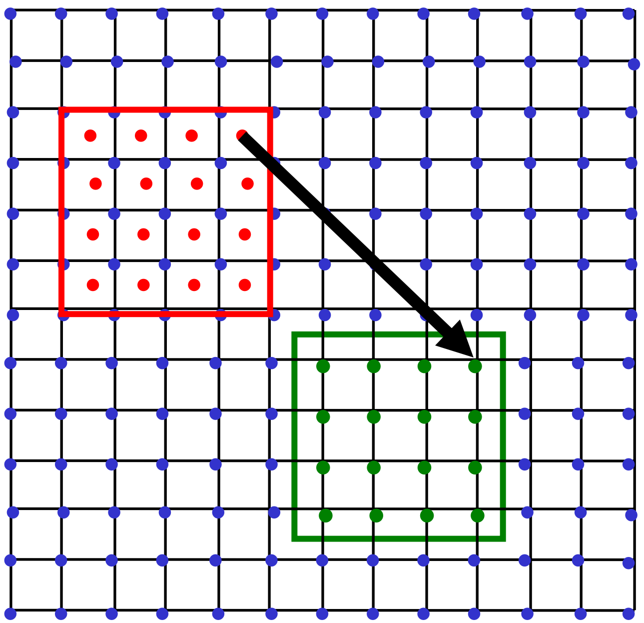


Sub-pel accuracy



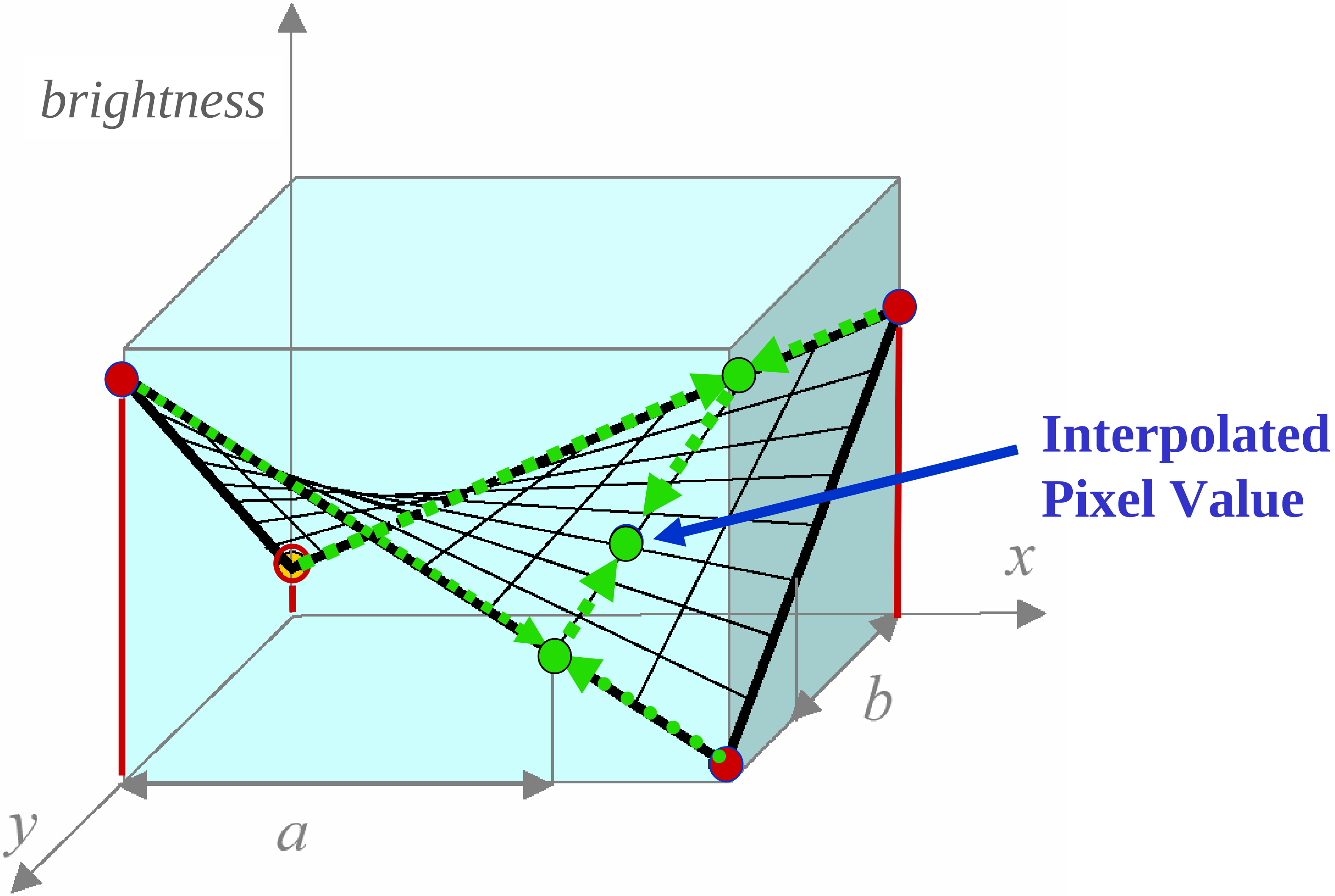
Sub-pel accuracy

- 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

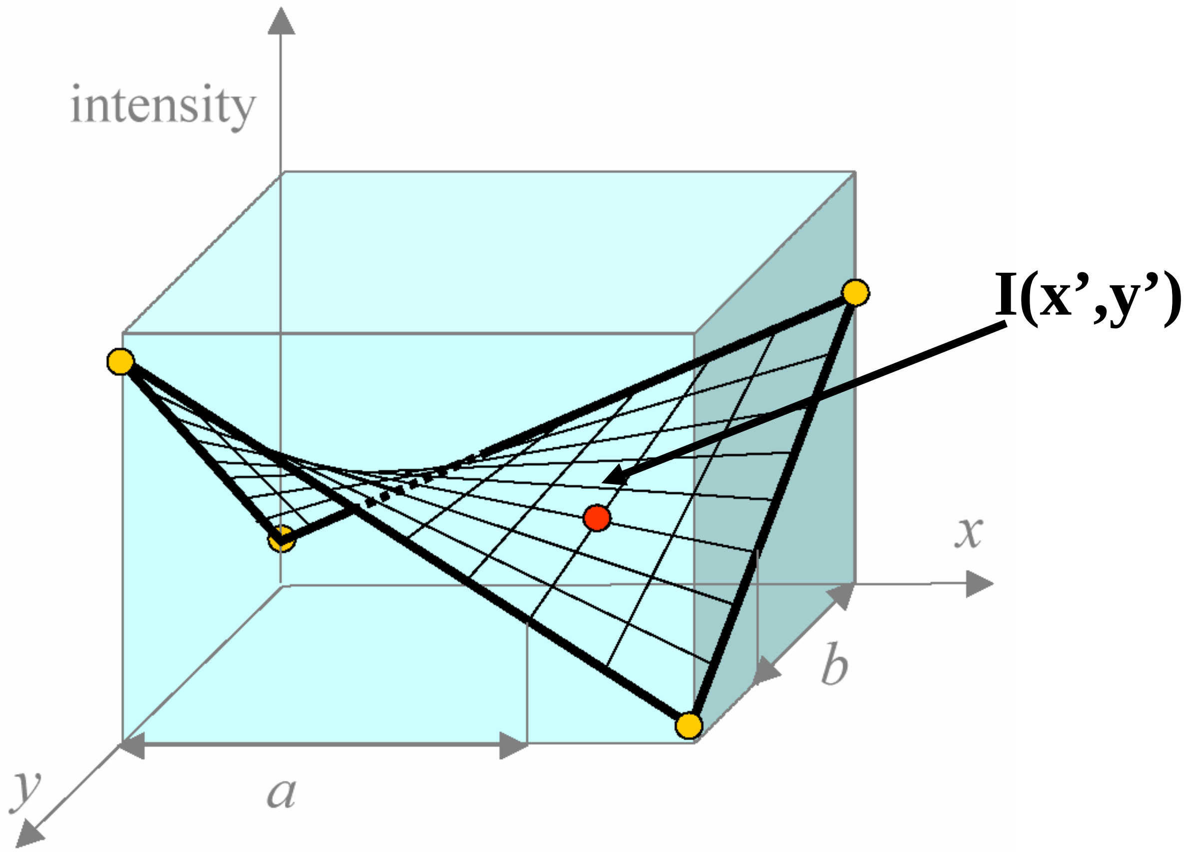


$$\begin{pmatrix} d_x \\ d_y \end{pmatrix} = \begin{pmatrix} 4.5 \\ 4.5 \end{pmatrix}$$

Bi-linear Interpolation

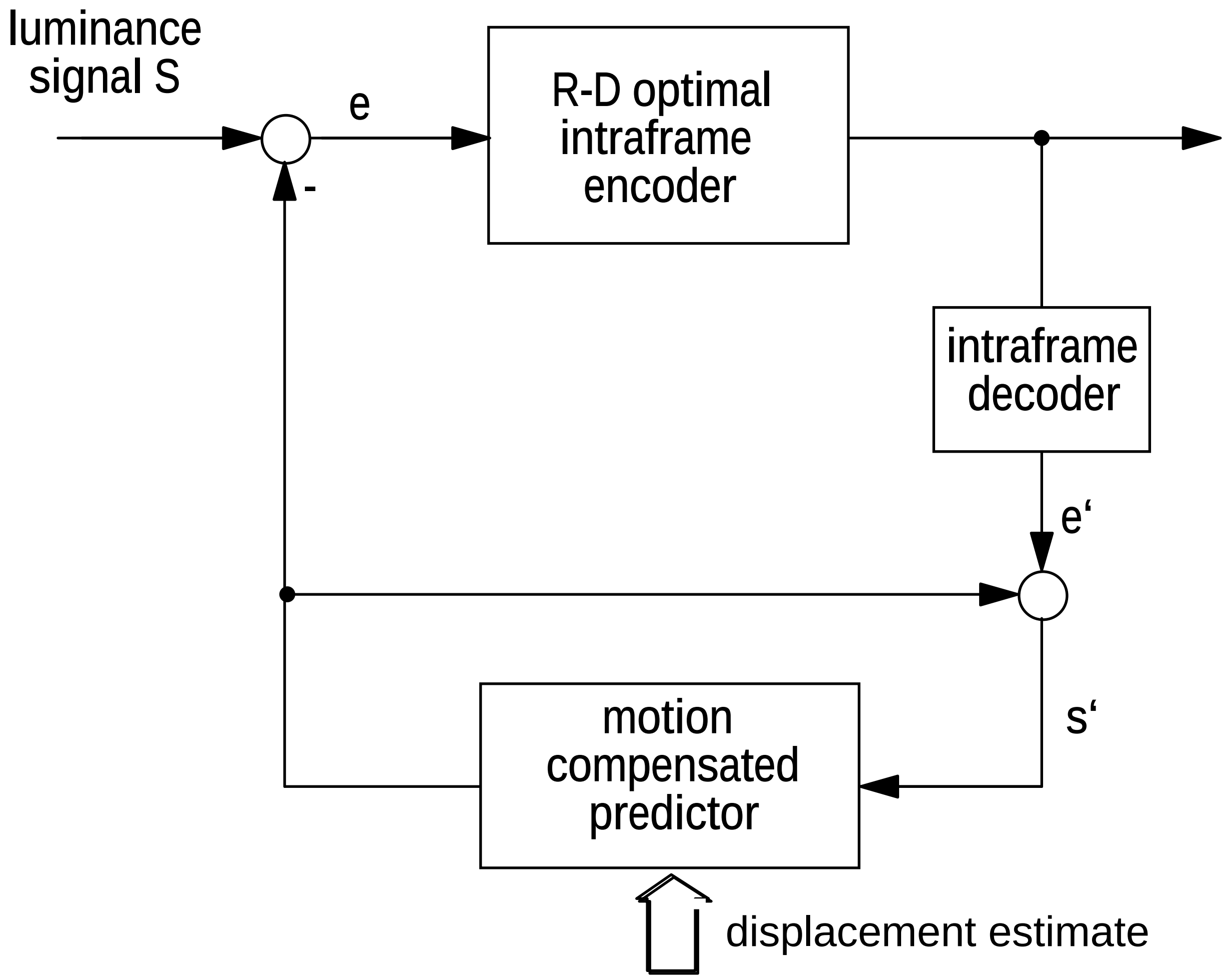


Bi-linear Interpolation (cont.)



$$\mathbf{I}(x',y') = \begin{bmatrix} 1-b & b \end{bmatrix} \begin{bmatrix} \mathbf{I}(x,y) & \mathbf{I}(x+1,y) \\ \mathbf{I}(x,y+1) & \mathbf{I}(x+1,y+1) \end{bmatrix} \begin{bmatrix} 1-a \\ a \end{bmatrix}$$

Model for performance analysis of an MCP hybrid coder



true displacement $\longrightarrow \begin{pmatrix} d_x \\ d_y \end{pmatrix} + \begin{pmatrix} \Delta_x \\ \Delta_y \end{pmatrix} \longleftarrow$ displacement error

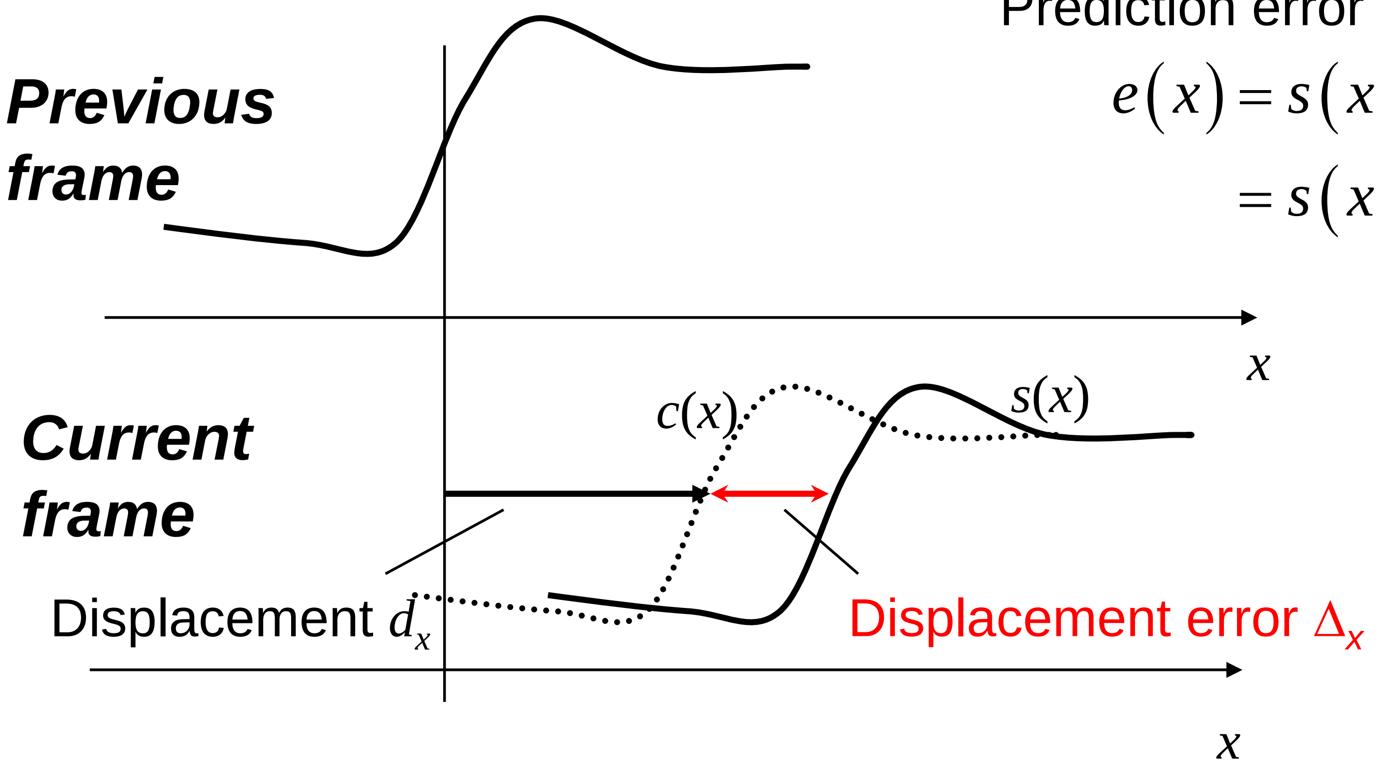
Analysis of the motion-compensated prediction error

Motion-compensated signal

$$c(x) = s(x - \Delta_x) - n(x)$$

Prediction error

$$e(x) = s(x) - c(x)$$
$$= s(x) - s(x - \Delta_x) + n(x)$$



Analysis of m.c. prediction error (cont.)

- 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 Δ_x, statistical independence of s and n

$$\begin{aligned}\Phi_{ee}(\omega)&=\Phi_{ss}(\omega)\left(1-e^{-j\omega\Delta_x}\right)\left(1-e^{j\omega\Delta_x}\right)+\Phi_{nn}(\omega)\\&=2\Phi_{ss}(\omega)\left(1-\mathrm{Re}\left\{e^{-j\omega\Delta_x}\right\}\right)+\Phi_{nn}(\omega)\end{aligned}$$

- Random displacement error Δ_x, statistically independent from s, n

$$\begin{aligned}\Phi_{ee}(\omega)&=E\left\{2\Phi_{ss}(\omega)\left(1-\mathrm{Re}\left\{e^{-j\omega\Delta_x}\right\}\right)+\Phi_{nn}(\omega)\right\}\\&=2\Phi_{ss}(\omega)\left(1-\mathrm{Re}\left\{E\left\{e^{-j\omega\Delta_x}\right\}\right\}\right)+\Phi_{nn}(\omega)\\&=2\Phi_{ss}(\omega)\left(1-\mathrm{Re}\left\{P(\omega)\right\}\right)+\Phi_{nn}(\omega)\end{aligned}$$

Analysis of m.c. prediction error (cont.)

- 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

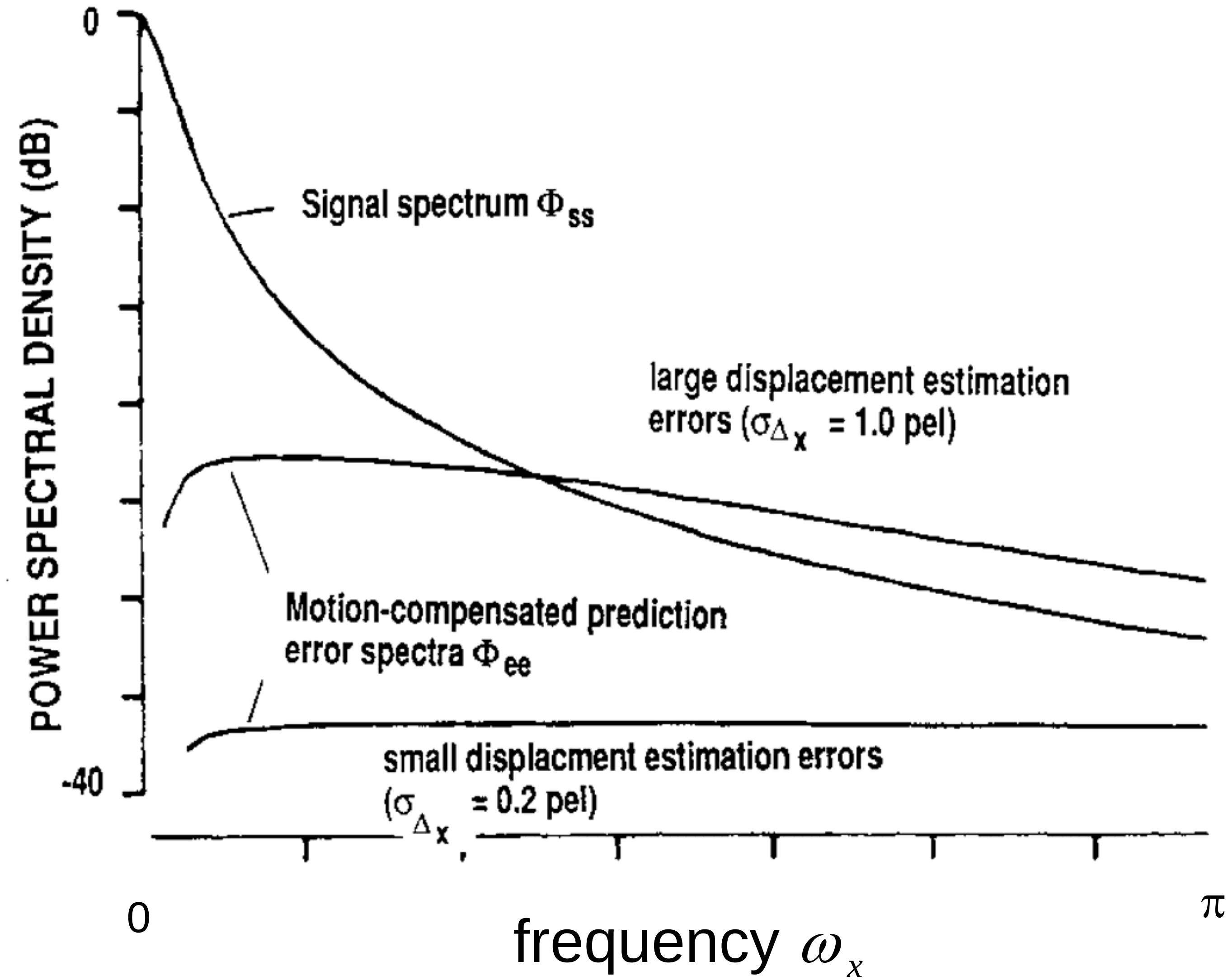
$$\boxed{\Phi_{ee}(\omega_x,\omega_y) = 2\Phi_{ss}(\omega_x,\omega_y)\left(1 - \mathrm{Re}\left\{P(\omega_x,\omega_y)\right\}\right) + \Phi_{nn}(\omega_x,\omega_y)}$$

power spectrum of luminance signal

Fourier transform of the displacement error pdf $p(\Delta_x,\Delta_y)$

noise spectrum

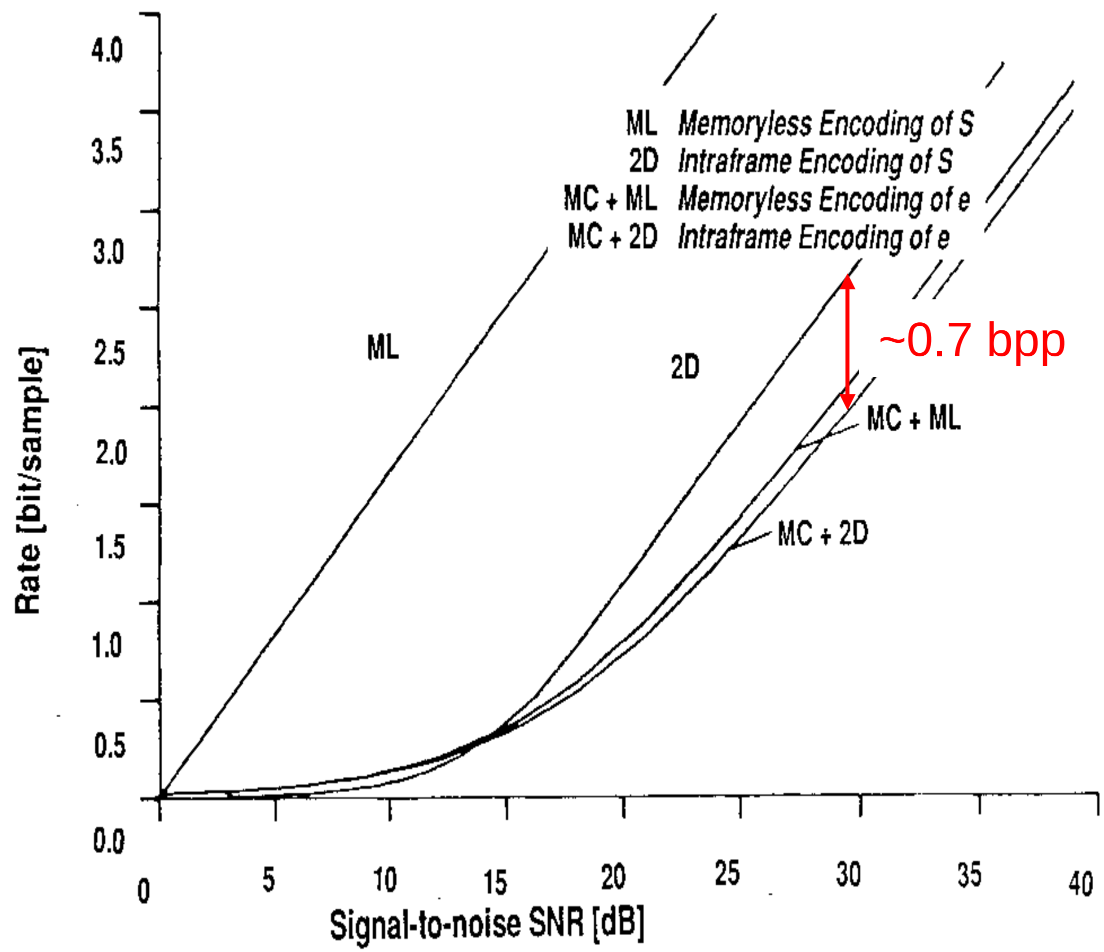
Power spectrum of motion-compensated prediction error



R-D function for MCP with integer-pixel accuracy

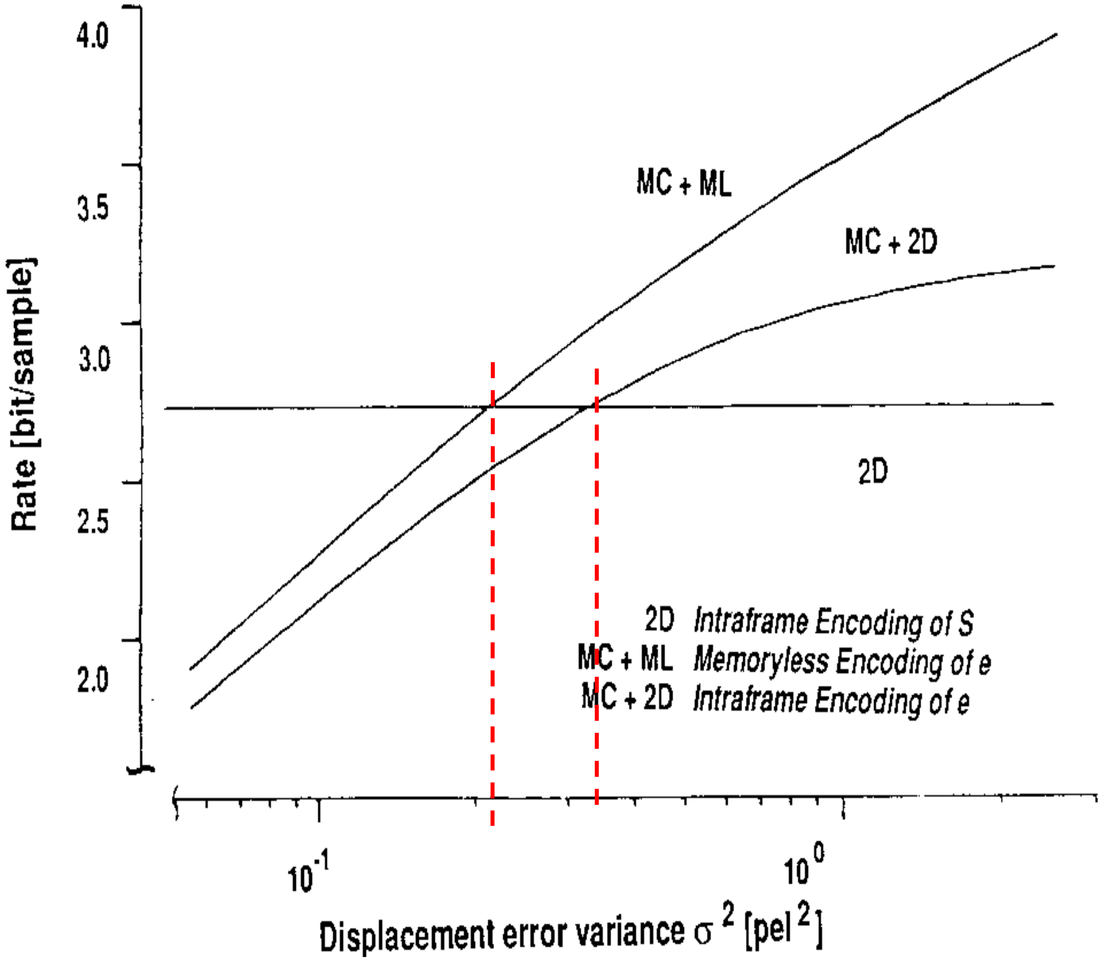
- $(\Delta_x, \Delta_y)^T$ assumed uniformly distributed between
$$\Delta_x = \pm\frac{1}{2} \text{ pel}$$
$$\Delta_y = \pm\frac{1}{2} \text{ line}$$
- Gaussian signal model
$$\Phi_{ss}(\omega_x, \omega_y) = A\left(1 + \frac{\omega_x^2 + \omega_y^2}{\omega_0^2}\right)^{-\frac{3}{2}}$$
- Typical parameters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

Minimum bit-rate for given SNR

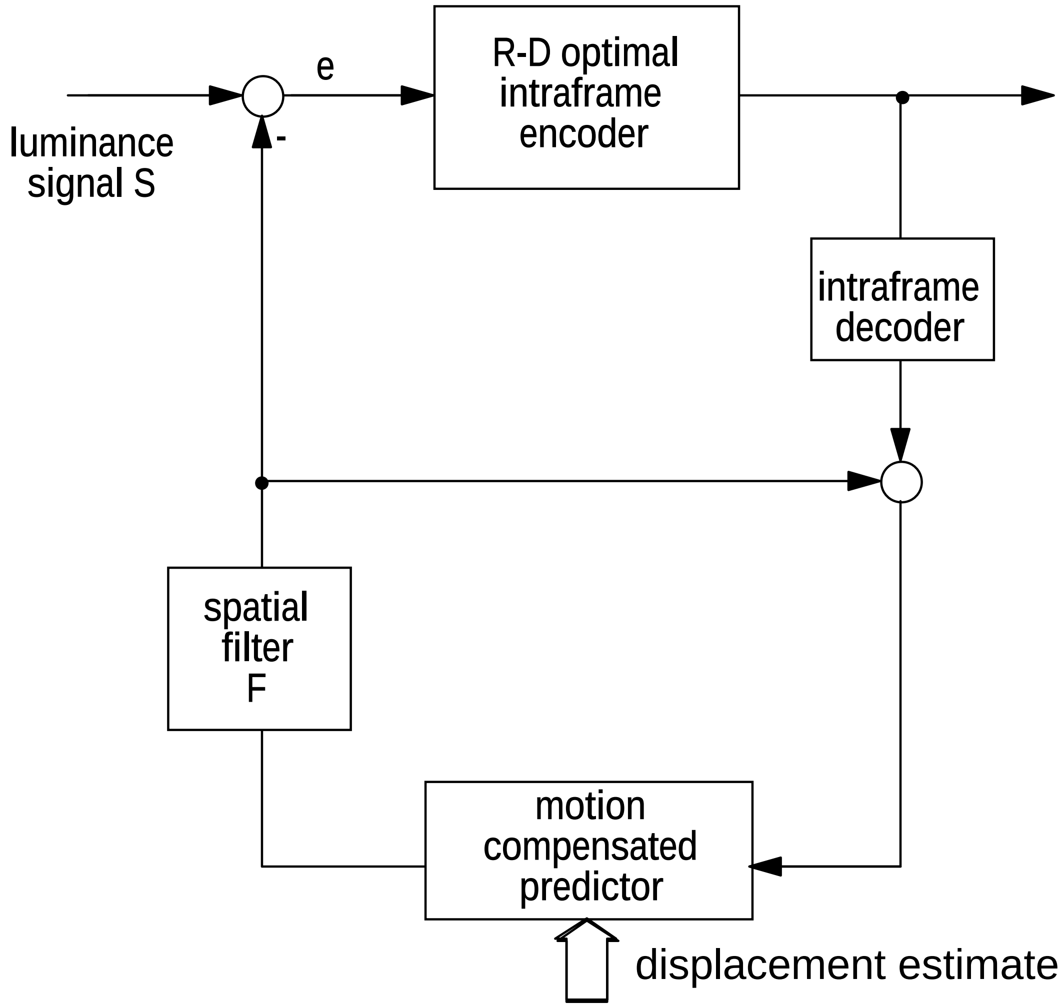


Required accuracy of motion compensation

- $p(\Delta_x, \Delta_y)$ isotropic Gaussian pdf with variance σ^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



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

Motion-compensated prediction error with loop filter

Motion-compensated signal

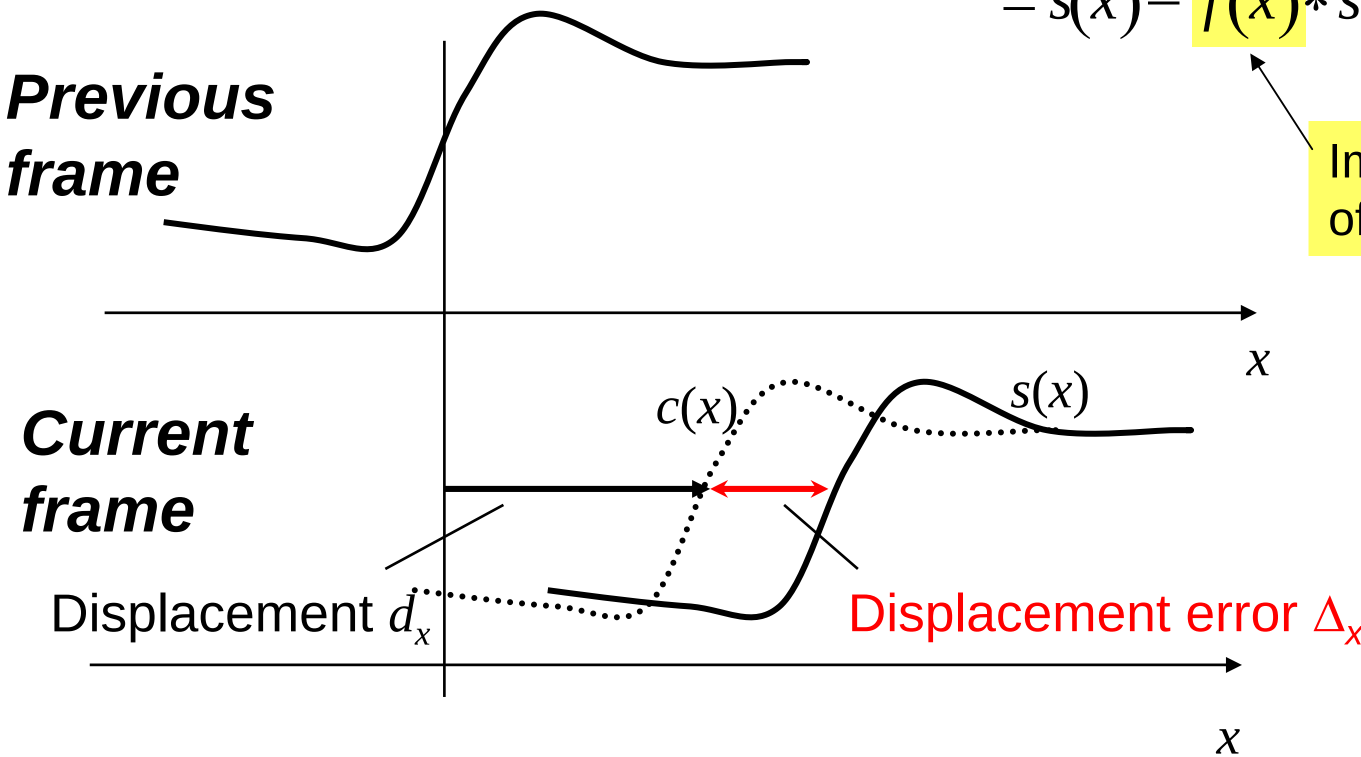
$$c(x) = s(x - \Delta_x) - n(x)$$

Prediction error

$$e(x) = s(x) - \boxed{f(x)} * c(x)$$

$$= s(x) - \boxed{f(x)} * s(x - \Delta_x) + \boxed{f(x)} * n(x)$$

Impulse response of loop filter



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

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

$P(\Lambda)$	2-D Fourier transform of displacement error pdf
$F(\Lambda)$	2-D Fourier transform of $f(x,y)$
Φ_{uu}	spatial spectral power density of signal u
Λ	vector of spatial frequencies (ω_x, ω_y)
$n(x,y)$	noise

Optimum loop filter

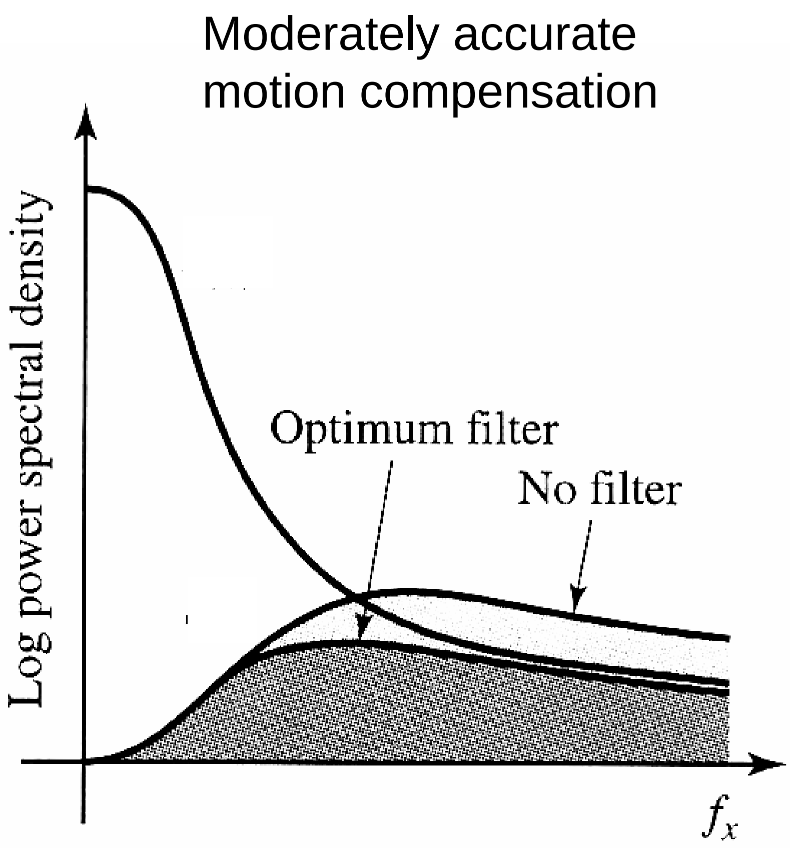
- Wiener filter minimizes prediction error variance

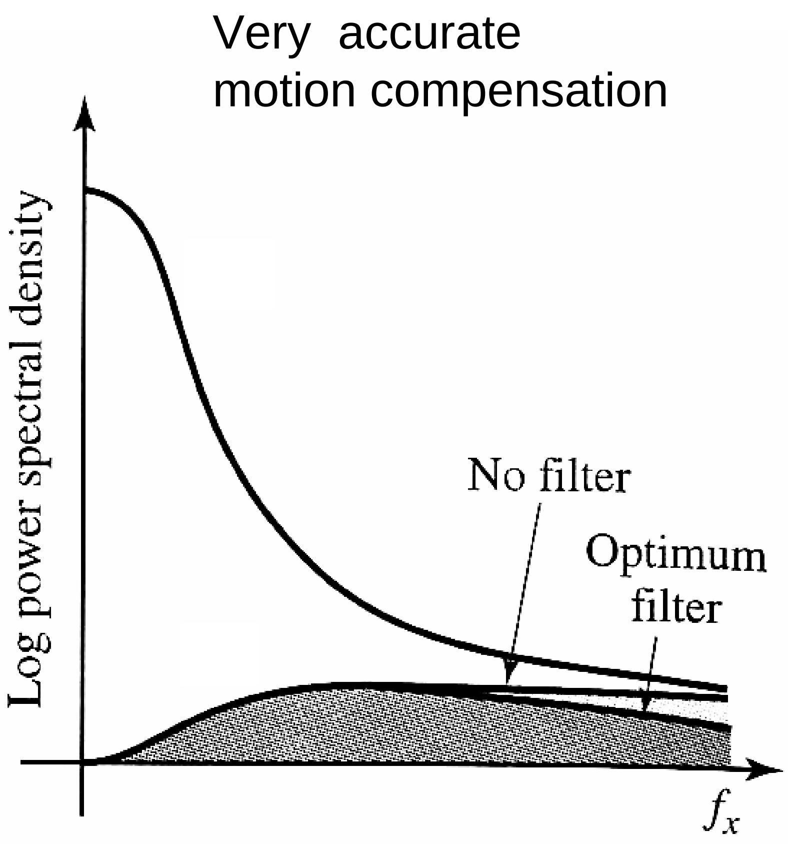
$$F_{\mathrm{opt}}(\Lambda) = \underbrace{P^*(\Lambda)}_{\text{accounts for accuracy of motion compensation}} \cdot \underbrace{\frac{\Phi_{ss}(\Lambda)}{\Phi_{ss}(\Lambda)+\Phi_{nn}(\Lambda)}}_{\text{accounts for noise}}$$

- Resulting minimum prediction error spectrum

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

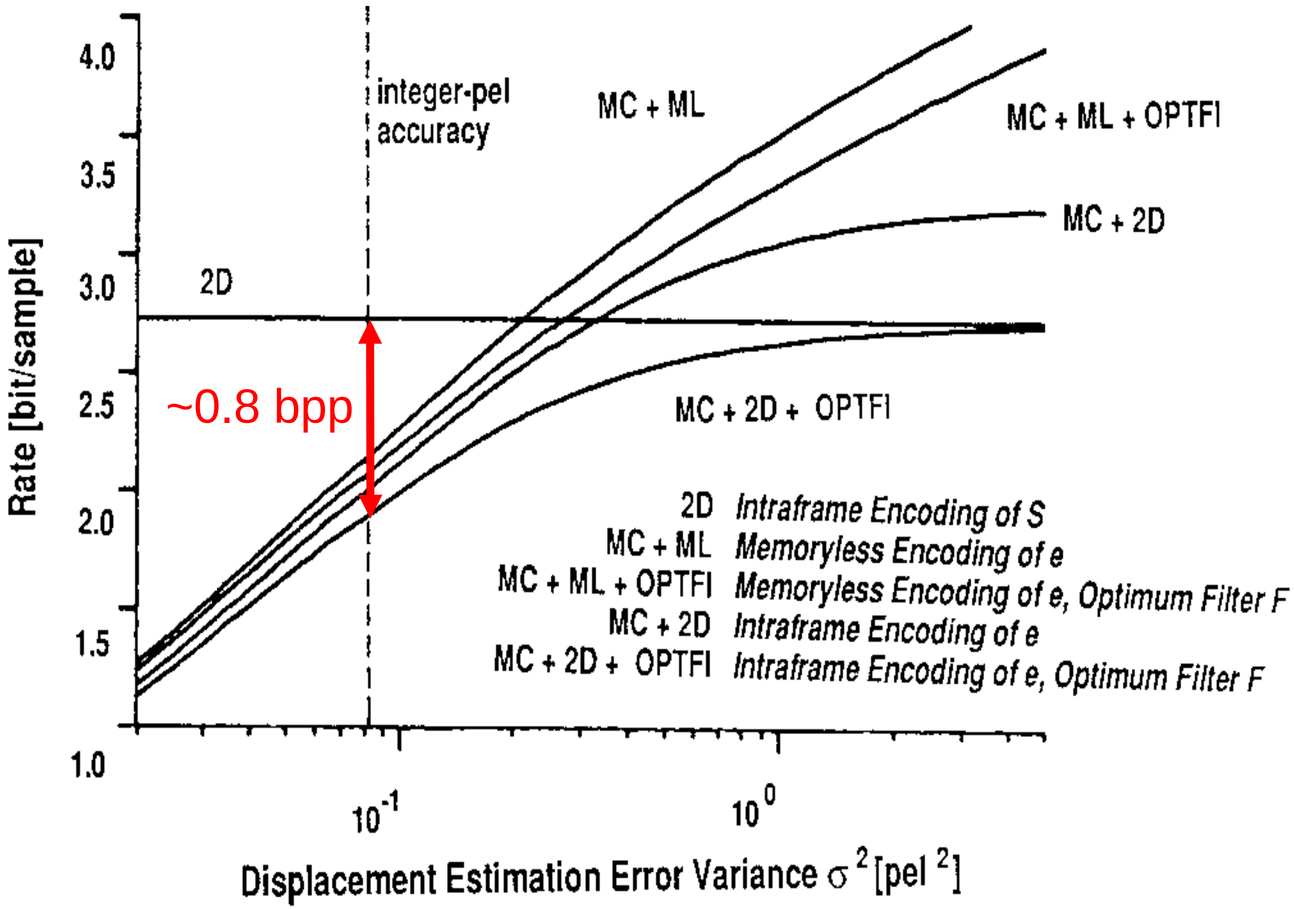
Effect of loop filter





Required accuracy of motion compensation with loop filter

- $p(\Delta_x, \Delta_y)$ isotropic Gaussian pdf with variance σ^2
- Minimum bit-rate for SNR = 30 dB



Practical optimum loop filter design

- Not practical for loop filter design

$$F_{\text{opt}}(\Lambda) = \underbrace{P^*(\Lambda)}_{\text{Motion compensation accuracy not known}} \cdot \frac{\Phi_{ss}(\Lambda)}{\underbrace{\Phi_{ss}(\Lambda) + \Phi_{nn}(\Lambda)}_{\text{“Noise” psd not known}}}$$

- To determine Wiener filter from measurements:

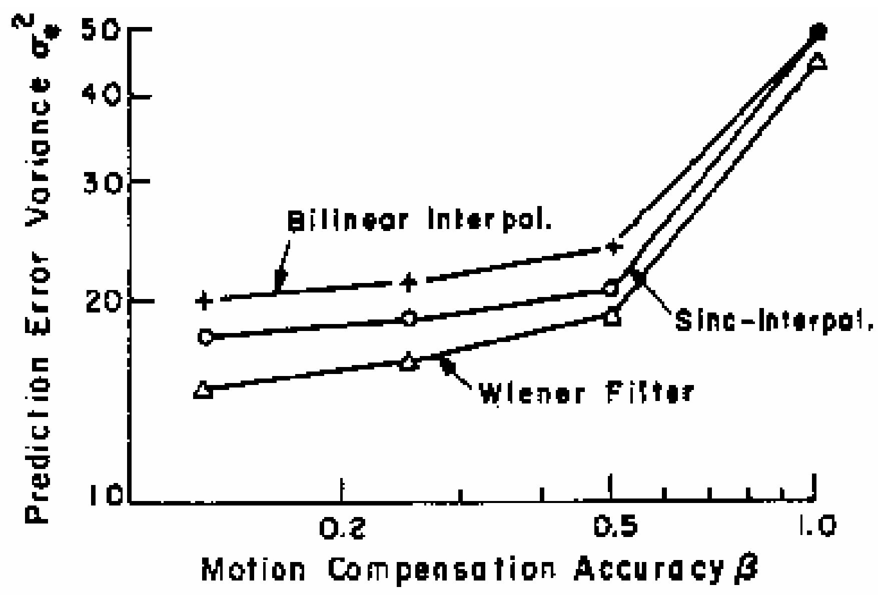
$$F_{\text{opt}}(\Lambda) = \frac{\Phi_{sc}(\Lambda)}{\Phi_{cc}(\Lambda)}$$

cross spectrum between $s(x,y)$ and the motion-compensated signal $c(x,y) = r(x - \hat{d}_x, y - \hat{d}_y)$

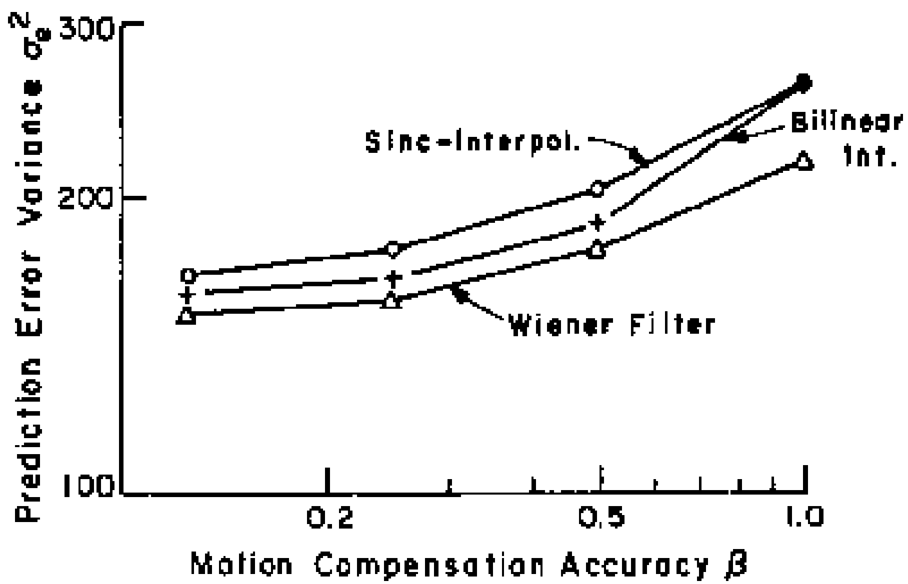
Experimental evaluation of fractional-pixel motion compensation

- ITU-R 601 TV signals, 13.5 MHz sampling rate, interlaced, blockwise motion compensation with blocksize16x16

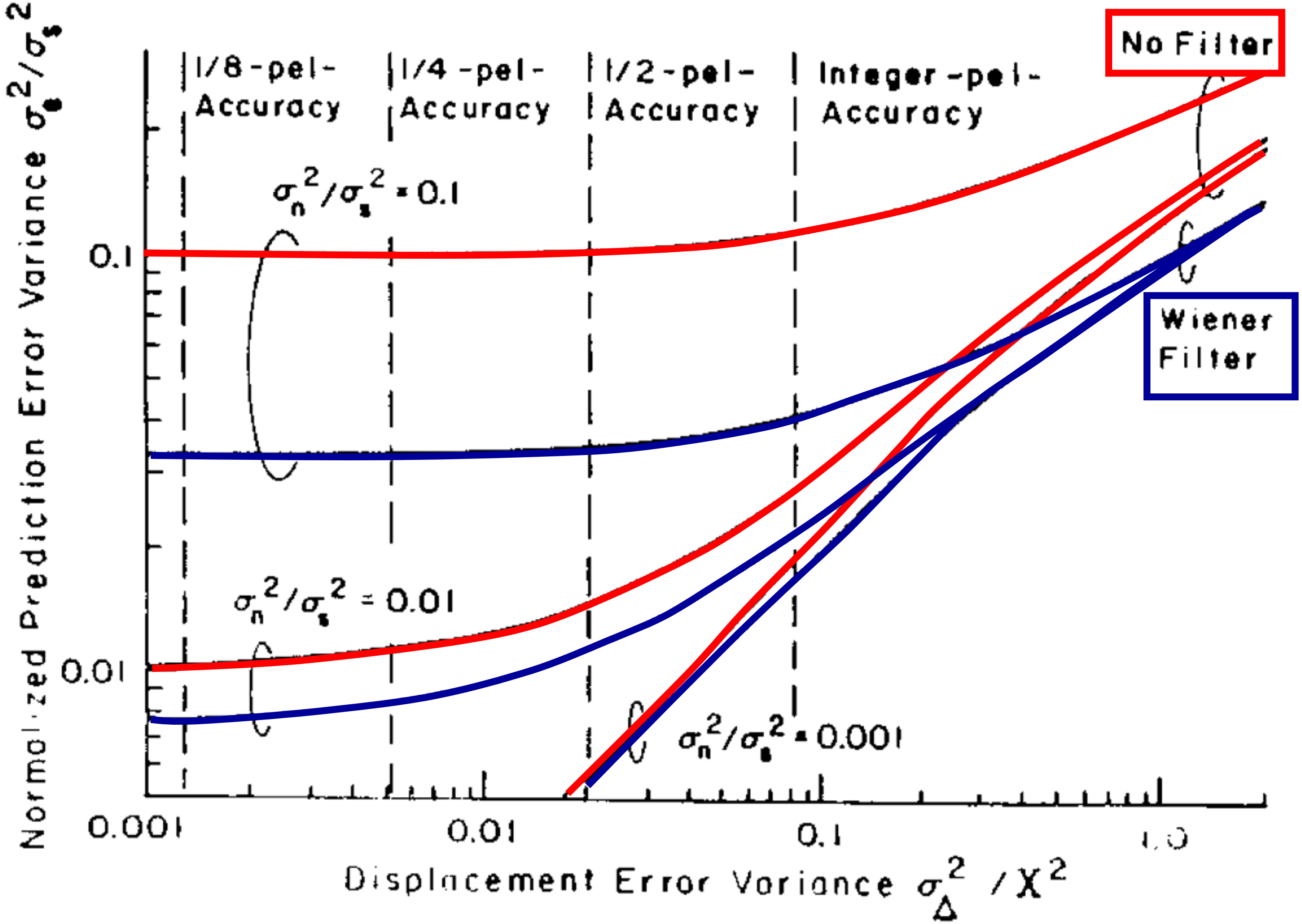
Zoom



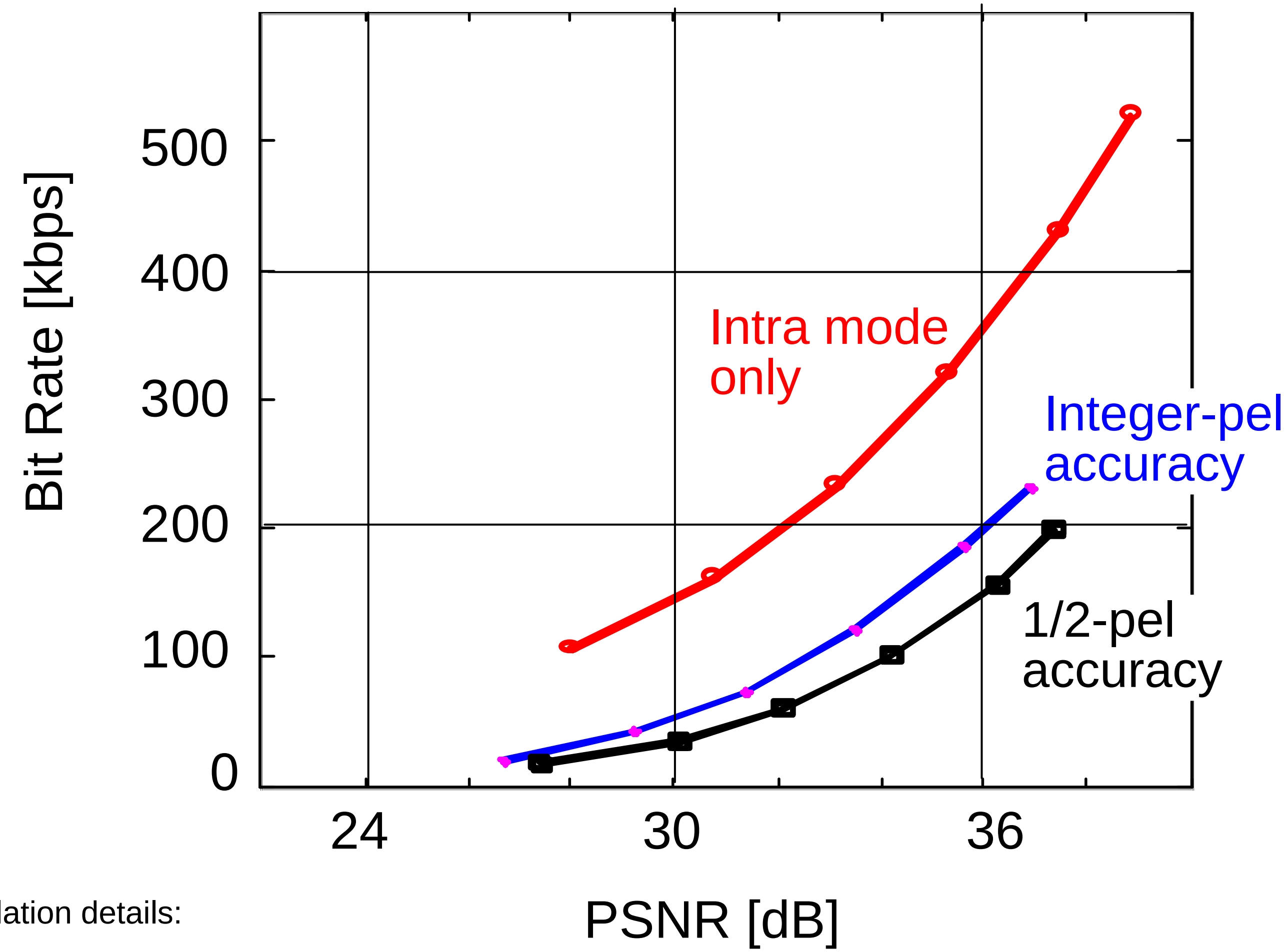
Voiture



Influence of noise on the performance of MCP



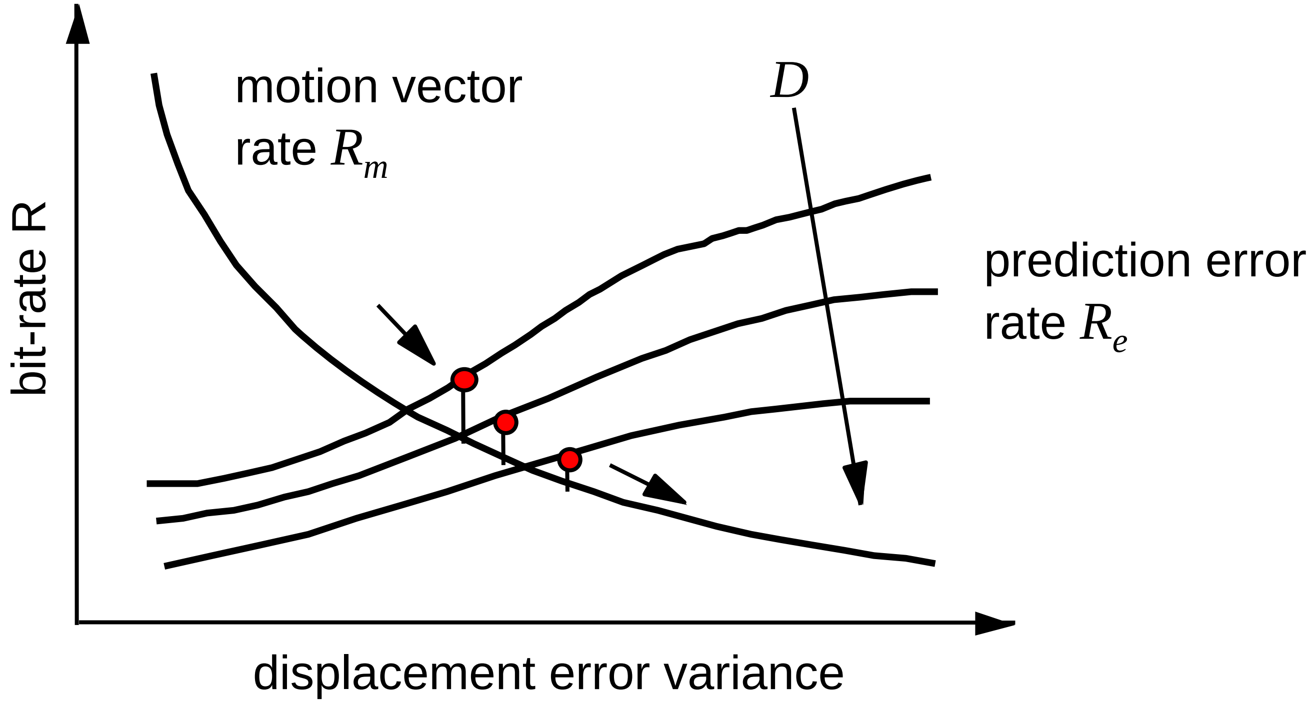
Motion Compensation Performance in H.263



Simulation details:

Foreman, QCIF, SKIP=2
Q=4,5,7,10,15,25

Rate-constrained motion estimation I



optimum trade-off:

$$\boxed{\frac{\partial D}{\partial R_m} = \frac{\partial D}{\partial R_e}}$$

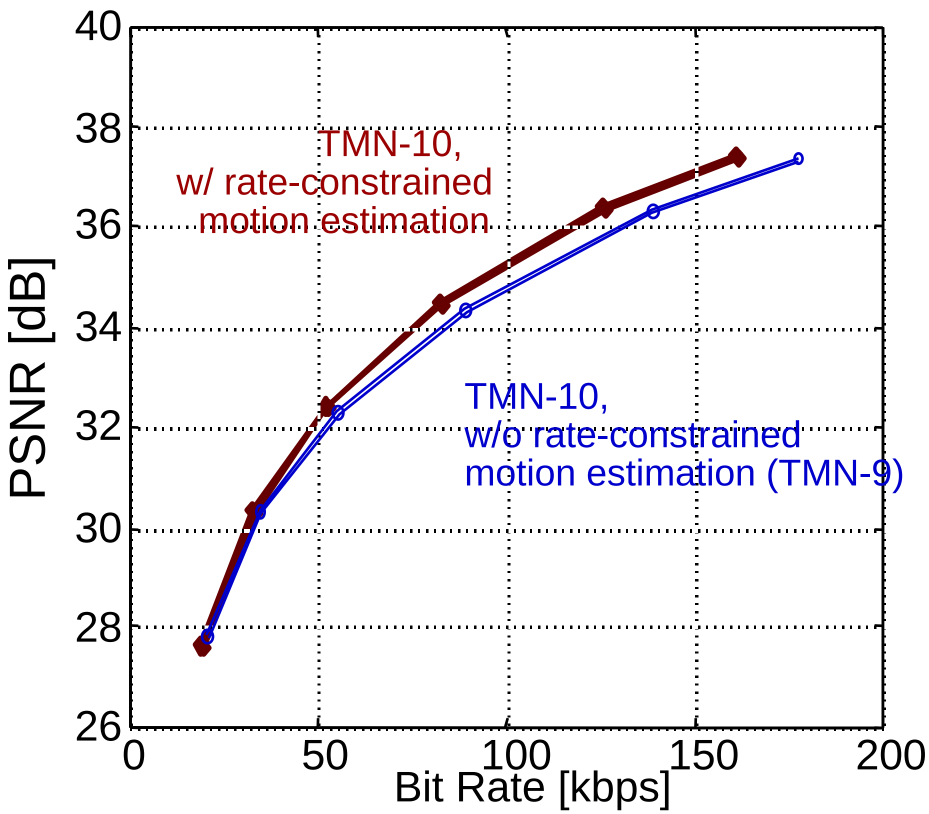
Rate-constrained motion estimation II

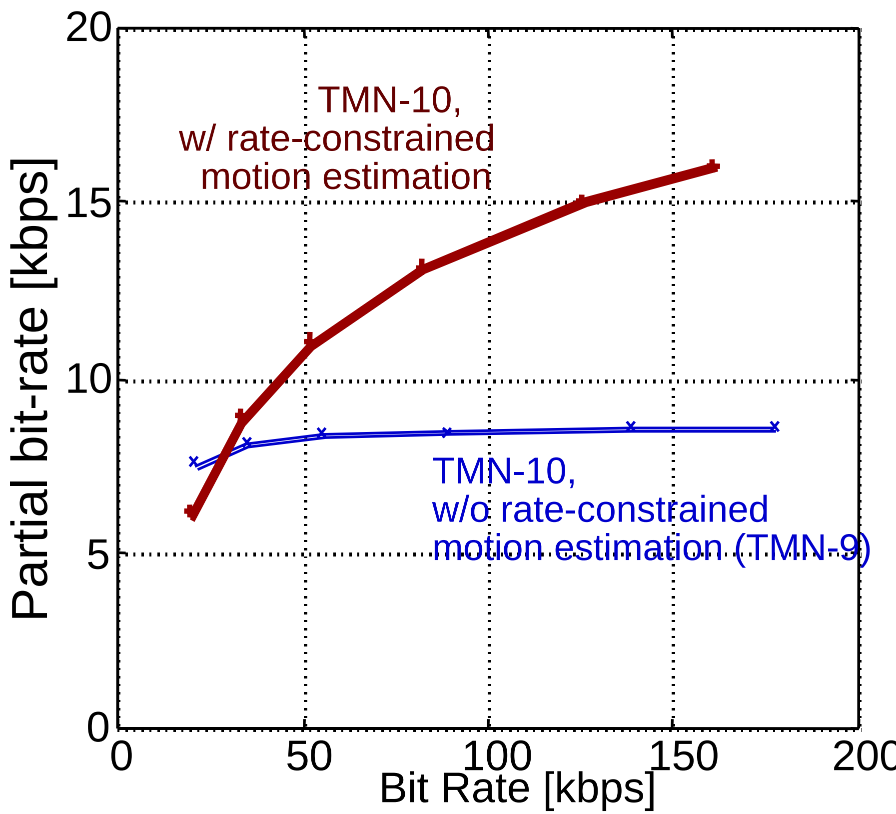
- How to find best motion vector subject to rate constraint?
- Lagrangian cost function: solve unconstrained problem rather than constrained problem

$$\boxed{\min\left(\underbrace{D}_{\text{error measure}} + \lambda \underbrace{R_m}_{\text{motion vector rate}}\right)}$$

$\Rightarrow$ Interpret motion search as ECVQ problem.

Rate-constrained Motion Estimation in H.263 Reference Model TMN-10



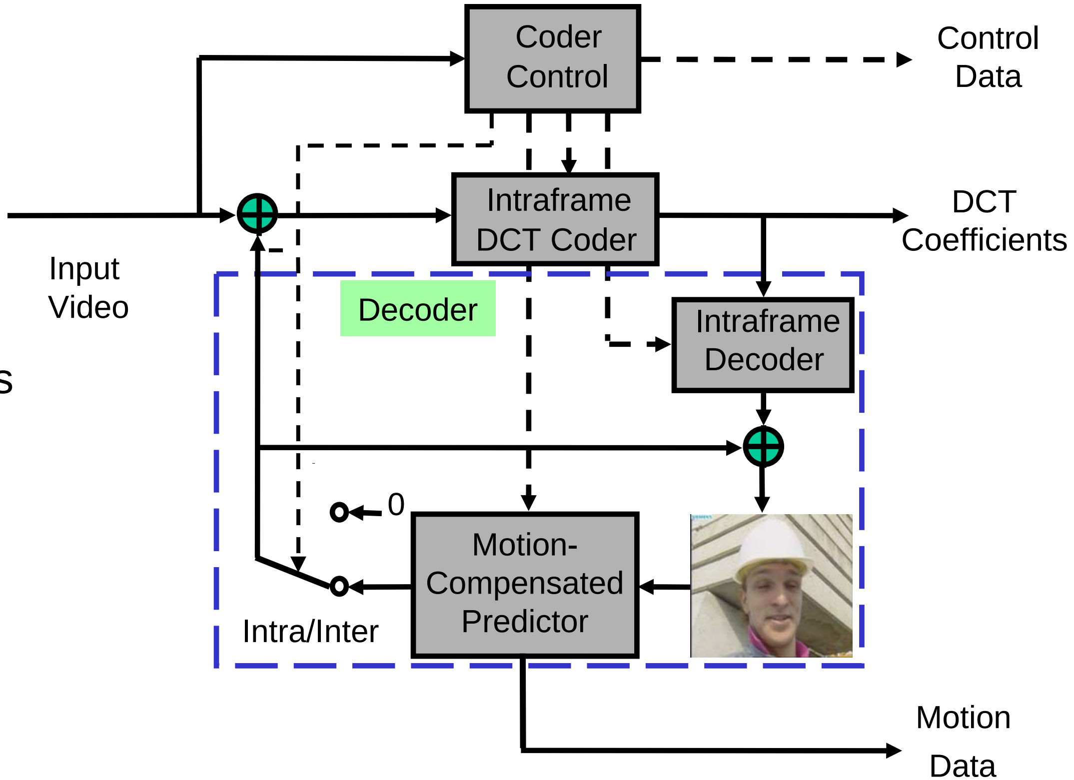


Simulation details:

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

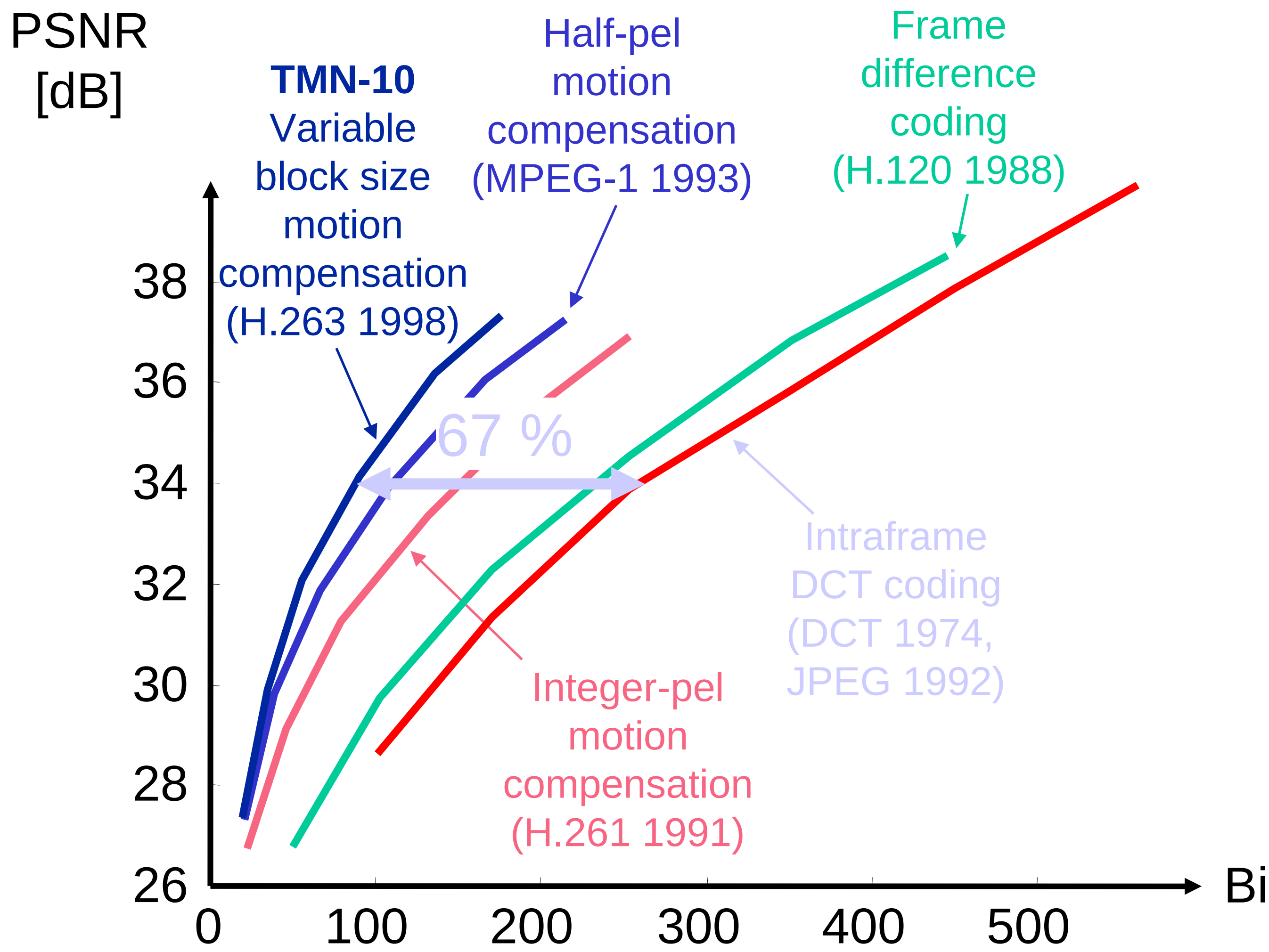


History of motion-compensated coding

Complexity increases

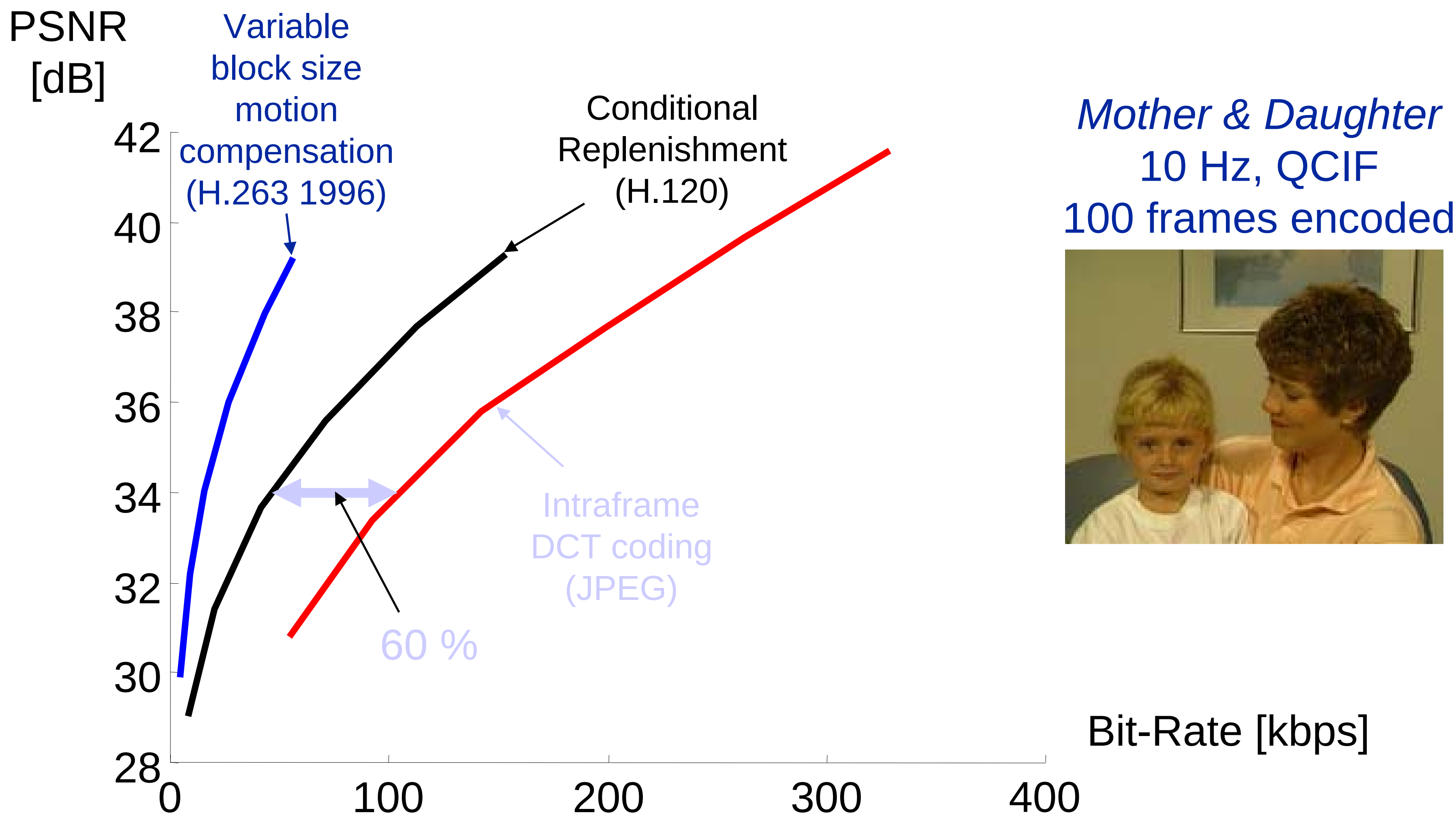
- *Intraframe coding:* only spatial correlation exploited
  ➔ DCT [Ahmed, Natarajan, Rao 1974], JPEG [1992]
- *Conditional replenishment*
  ➔ H.120 [1984] (*DPCM, scalar quantization*)
- *Frame difference coding*
  ➔ H.120 Version 2 [1988]
- *Motion compensation: integer-pel accurate displacements*
  ➔ H.261 [1991]
- *Half-pel accurate motion compensation*
  ➔ MPEG-1 [1993], MPEG-2/H.262 [1994]
- *Variable block-size motion compensation*
  ➔ H.263 [1996], MPEG-4 [1999]

Efficiency of motion-compensated coding



Foreman
10 Hz, QCIF
100 frames encoded

Efficiency of motion-compensated coding



Efficiency of motion-compensated coding