Visual Computing:
Advanced Optical Flow & Video Compression

Prof. Marc Pollefeys

ETH

inf | Informatik Computer Science

Last lecture: Optical Flow

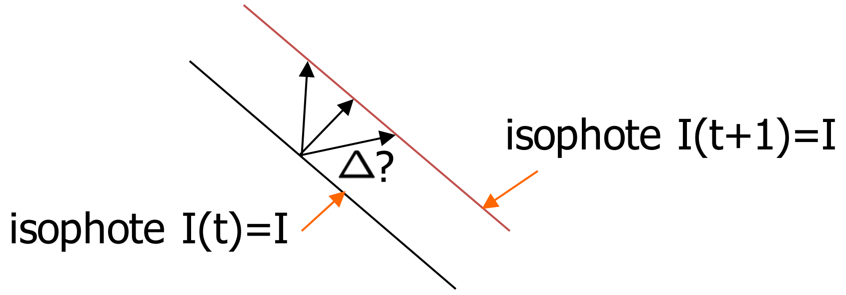
- Brightness constancy equation

$$\boxed{I_x u + I_y v + I_t = 0}$$

- Aperture problem

$I_x u + I_y v + I_t = 0$ (1 constraint)

u, v (2 unknowns)

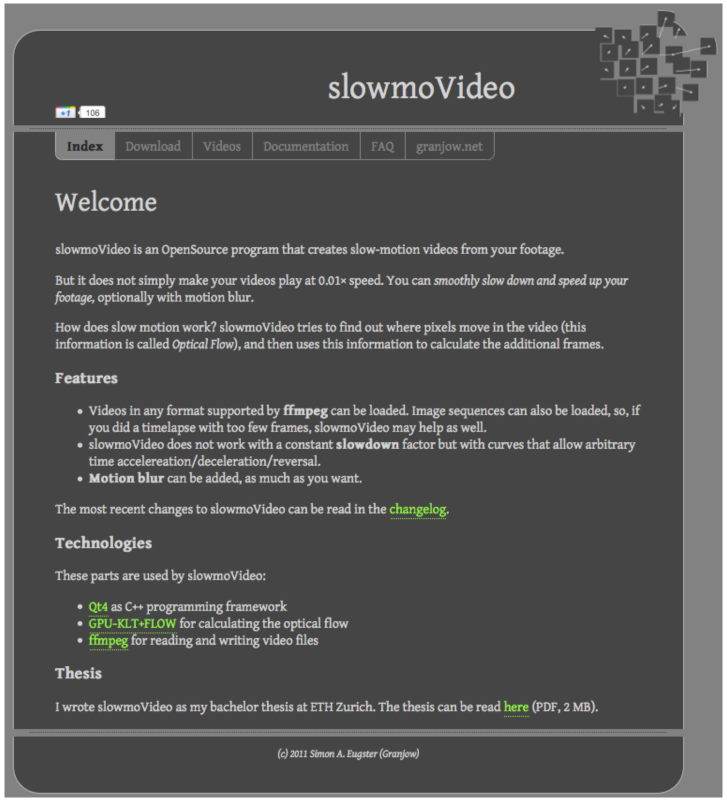


- Solution:
  - regularize (trade-off brightness constancy and smoothness)
  - And many more (***today***)

$$E(\mathbf{h}) = \sum_{\mathbf{x} \in R} [I(\mathbf{x}+\mathbf{h}) - I_0(\mathbf{x})]^2$$

SlowMoVideo

Bachelor thesis Simon Eugster



slowmoVideo

+1 106

Index | Download | Videos | Documentation | FAQ | granjow.net

Welcome

slowmoVideo is an OpenSource program that creates slow-motion videos from your footage.

But it does not simply make your videos play at 0.01× speed. You can *smoothly slow down and speed up your footage*, optionally with motion blur.

How does slow motion work? slowmoVideo tries to find out where pixels move in the video (this information is called *Optical Flow*), and then uses this information to calculate the additional frames.

Features

- Videos in any format supported by **ffmpeg** can be loaded. Image sequences can also be loaded, so, if you did a timelapse with too few frames, slowmoVideo may help as well.
- slowmoVideo does not work with a constant **slowdown** factor but with curves that allow arbitrary time acceleration/deceleration/reversal.
- **Motion blur** can be added, as much as you want.

The most recent changes to slowmoVideo can be read in the changelog.

Technologies

These parts are used by slowmoVideo:

- Qt4 as C++ programming framework
- GPU-KLT+FLOW for calculating the optical flow
- ffmpeg for reading and writing video files

Thesis

I wrote slowmoVideo as my bachelor thesis at ETH Zurich. The thesis can be read here (PDF, 2 MB).

(c) 2011 Simon A. Eugster (Granjow)

http://slowmovideo.granjow.net/

Optical Flow

- Brightness Constancy
- The Aperture problem
- Regularization
- Lucas-Kanade
- Coarse-to-fine
- Parametric motion models
- SSD tracking
- Bayesian flow

Parametric (Global) Motion Models

Global motion models offer

- more constrained solutions than smoothness (Horn-Schunck)
- integration over a larger area than a translation-only model can accommodate (Lucas-Kanade)

Parametric (Global) Motion Models

2D Models:

(Translation)

Affine

Quadratic

Planar projective transform (Homography)

3D Models:

Instantaneous camera motion models

Homography+epipole

Plane+Parallax

$$E(\mathbf{h}) = \sum_{\mathbf{x} \in R} \left[I(\mathbf{x} + \mathbf{h}) - I_0(\mathbf{x}) \right]^2$$

- Transformations/warping of image

$$E(\mathbf{h}) = \sum_{\mathbf{x} \in R} \left[I(\mathbf{x} + \mathbf{h}) - I_0(\mathbf{x}) \right]^2$$

Translations:

$$\mathbf{h} = \begin{bmatrix} \delta x \\ \delta y \end{bmatrix}$$

What about other types of motion?

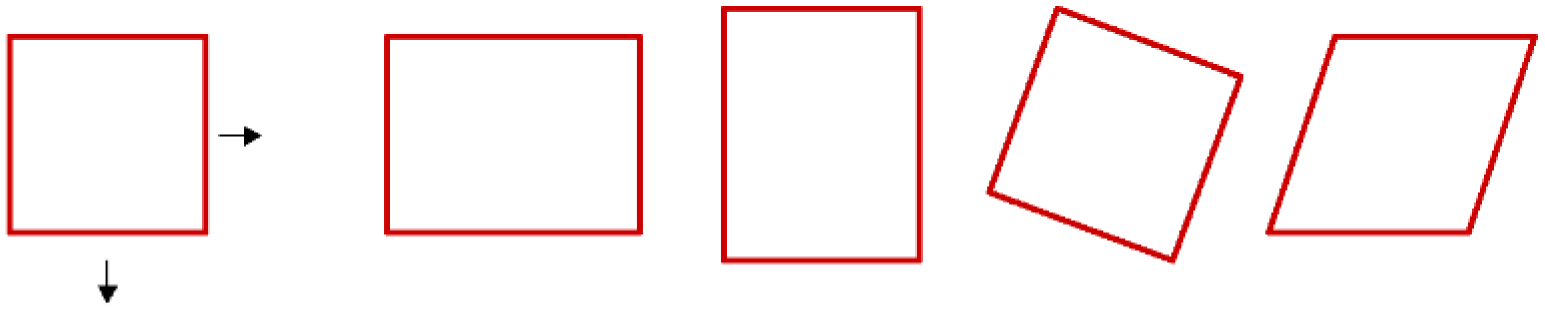
Generalization

- Transformations/warping of image

$$E(\mathbf{A}, \mathbf{h}) = \sum_{\mathbf{x} \in R} \left[I(\mathbf{A}\mathbf{x} + \mathbf{h}) - I_0(\mathbf{x}) \right]^2$$

Affine: $\mathbf{A} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \quad \mathbf{h} = \begin{bmatrix} \delta x \\ \delta y \end{bmatrix}$

Generalization



Affine: $\mathbf{A} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \quad \mathbf{h} = \begin{bmatrix} \delta x \\ \delta y \end{bmatrix}$

Example: Affine Motion

$$\boxed{\begin{aligned} u(x,y) &= a_1 + a_2 x + a_3 y \\ v(x,y) &= a_4 + a_5 x + a_6 y \end{aligned}}$$

Substituting into the B.C. Equation:

$$I_x \cdot u + I_y \cdot v + I_t \approx 0$$

$$\boxed{I_x (a_1 + a_2 x + a_3 y) + I_y (a_4 + a_5 x + a_6 y) + I_t \approx 0}$$

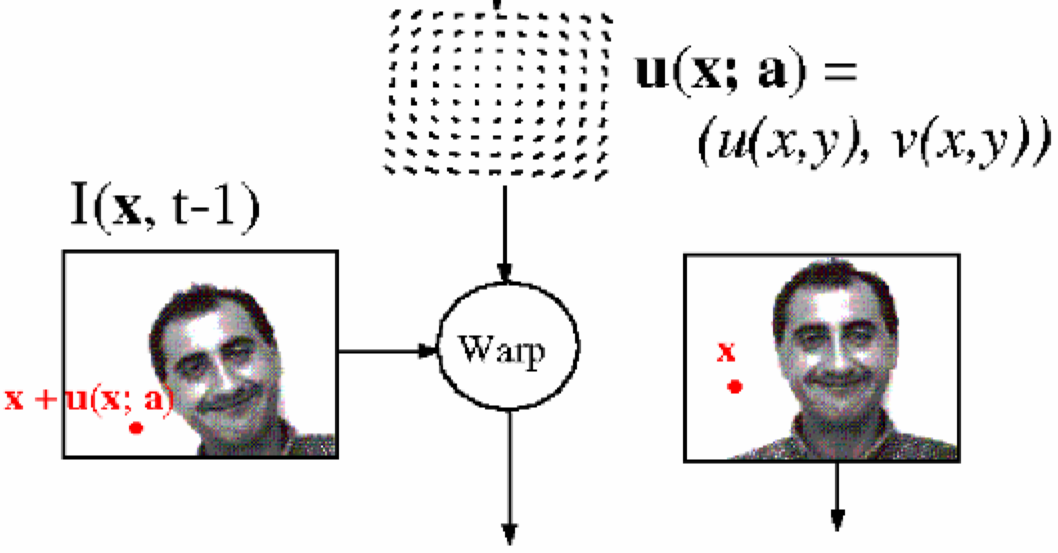
Each pixel provides 1 linear constraint in 6 *global* unknowns
(minimum 6 pixels necessary)

Least Square Minimization (over all pixels):

$$Err(\vec{a}) = \sum \left[I_x (a_1 + a_2 x + a_3 y) + I_y (a_4 + a_5 x + a_6 y) + I_t \right]^2$$

$$u(x, y) = a_0 + a_1 x + a_2 y$$

$$v(x, y) = a_3 + a_4 x + a_5 y$$


$\mathbf{u}(\mathbf{x};\, \mathbf{a}) =$
$(u(x,y),\, v(x,y))$
$I(\mathbf{x}, t\text{-}1)$
$\mathbf{x} + \mathbf{u}(\mathbf{x};\, \mathbf{a})$
Warp
$\mathbf{x}$

$$I(\mathbf{x}+\mathbf{u}(\mathbf{x};\, \mathbf{a}),\, t-1) = I(\mathbf{x}, t)$$

(Brightness Constancy Assumption)

KLT: Good features to keep tracking

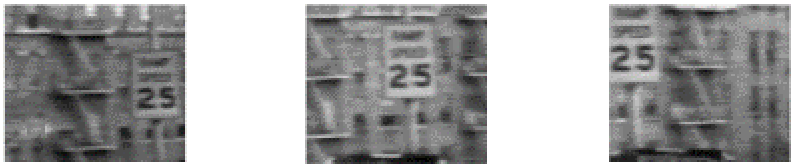
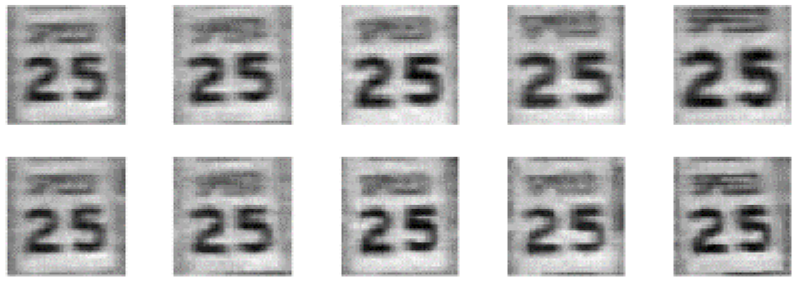
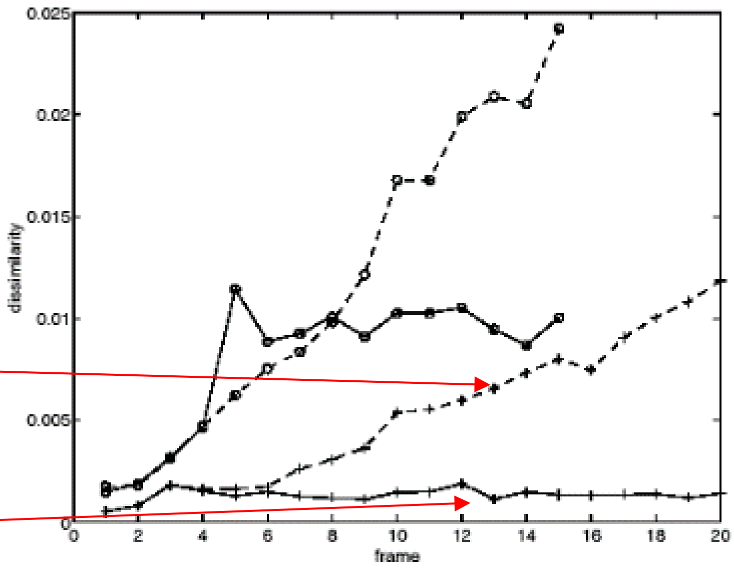


Figure 1: Three frame details from Woody Allen's *Manhattan*. The details are from the 1st, 11th, and 21st frames of a subsequence from the movie.





Simple displacement is sufficient between consecutive frames, but not to compare to reference template

Generalization

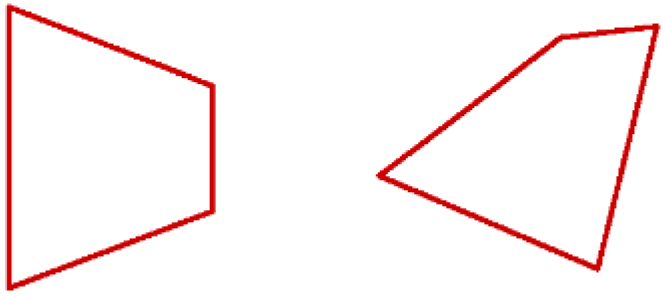
- Transformations/warping of image

$$E(\mathbf{A}) = \sum_{\mathbf{x} \in R} [I(\mathbf{A}\mathbf{x}) - I_0(\mathbf{x})]^2$$

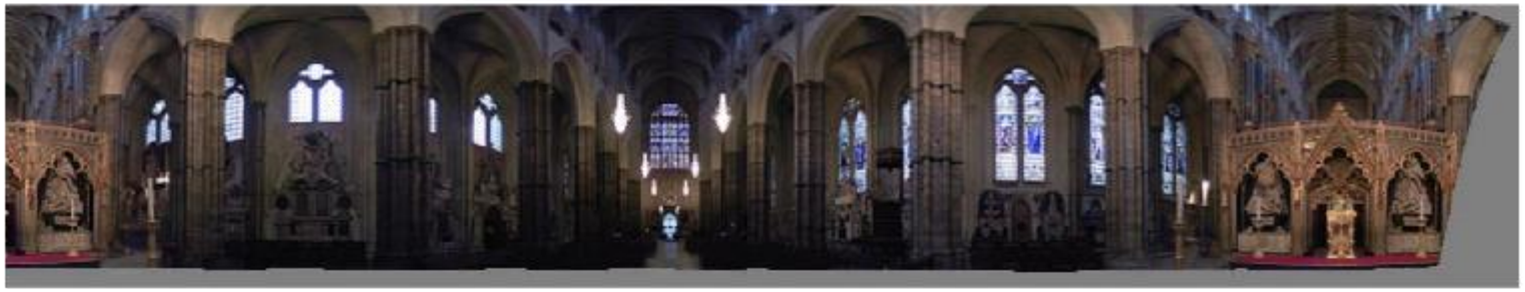
Planar perspective: $\mathbf{A} = \begin{bmatrix} a_1 & a_2 & a_3 \\ a_4 & a_5 & a_6 \\ a_7 & a_8 & 1 \end{bmatrix}$

Generalization

Affine $+$



Planar perspective: $\mathbf{A} = \begin{bmatrix} a_1 & a_2 & a_3 \\ a_4 & a_5 & a_6 \\ a_7 & a_8 & 1 \end{bmatrix}$



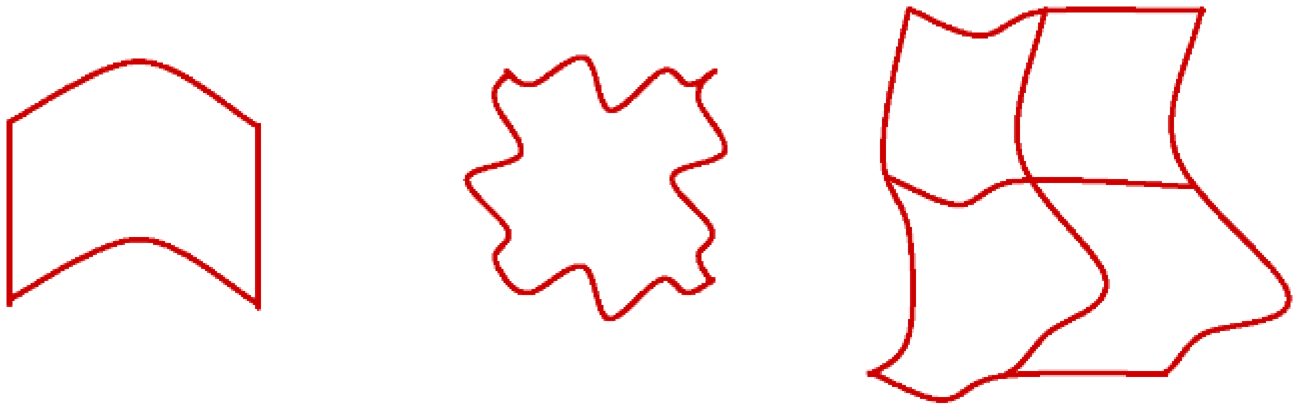
Generalization

- Transformations/warping of image

$$E(\mathbf{h}) = \sum_{\mathbf{x} \in R} \left[I(\mathbf{f}(\mathbf{x}, \mathbf{h})) - I_0(\mathbf{x}) \right]^2$$

Other parametrized transformations

Generalization



Other parametrized transformations

2D Motion Models summary

Quadratic – instantaneous approximation to planar motion

$$u = q_1 + q_2 x + q_3 y + q_7 x^2 + q_8 xy$$
$$v = q_4 + q_5 x + q_6 y + q_7 xy + q_8 y^2$$

Projective – exact planar motion

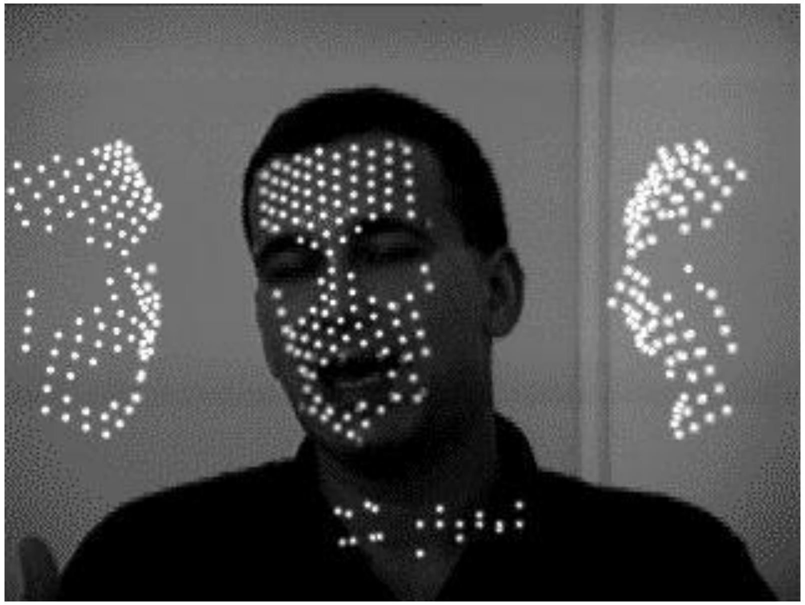
$$x' = \frac{h_1 + h_2 x + h_3 y}{h_7 + h_8 x + h_9 y}$$
$$y' = \frac{h_4 + h_5 x + h_6 y}{h_7 + h_8 x + h_9 y}$$

and

$$u = x' - x, \quad v = y' - y$$

Advanced parametric model

- Optical flow constrained by non-rigid face model



Flexible flow for 3D nonrigid tracking and shape recovery,
Brand and Bhotika, CVPR2001.

3D Motion Models summary

Instantaneous camera motion:

Global parameters: $\Omega_X, \Omega_Y, \Omega_Z, T_X, T_Y, T_Z$

Local Parameter: $Z(x,y)$

$$u = -xy\Omega_X + (1+x^2)\Omega_Y - y\Omega_Z + (T_X - T_Z x)/Z$$
$$v = -(1+y^2)\Omega_X + xy\Omega_Y - x\Omega_Z + (T_Y - T_Z x)/Z$$

Homography+Epipole

Global parameters: $h_1, \ldots, h_9, t_1, t_2, t_3$

Local Parameter: $\gamma(x,y)$

$$x' = \frac{h_1 x + h_2 y + h_3 + \gamma t_1}{h_7 x + h_8 y + h_9 + \gamma t_3}$$
$$y' = \frac{h_4 x + h_5 y + h_6 + \gamma t_1}{h_7 x + h_8 y + h_9 + \gamma t_3}$$
and : $u = x' - x, \quad v = y' - y$

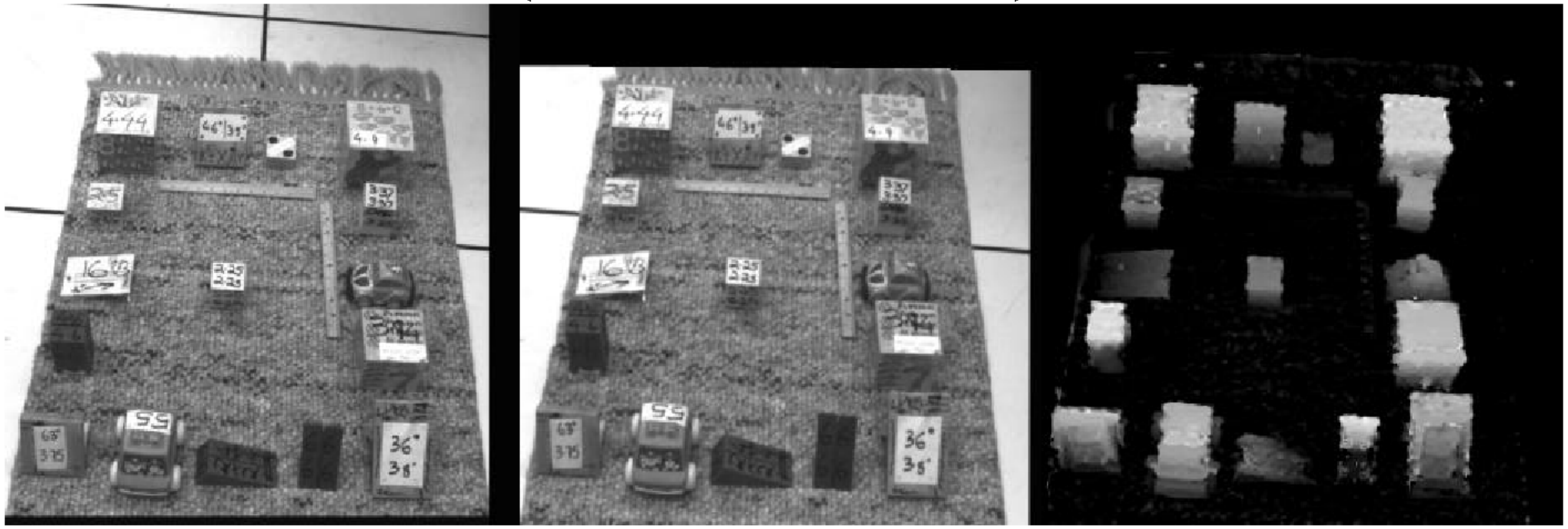
Residual Planar Parallax Motion

Global parameters: t_1, t_2, t_3

Local Parameter: $\gamma(x,y)$

$$u = x^w - x = \frac{\gamma}{1+\gamma t_3}(t_3 x - t_1)$$
$$v = y^w - x = \frac{\gamma}{1+\gamma t_3}(t_3 y - t_2)$$

Residual Planar Parallax Motion (Plane+Parallax)

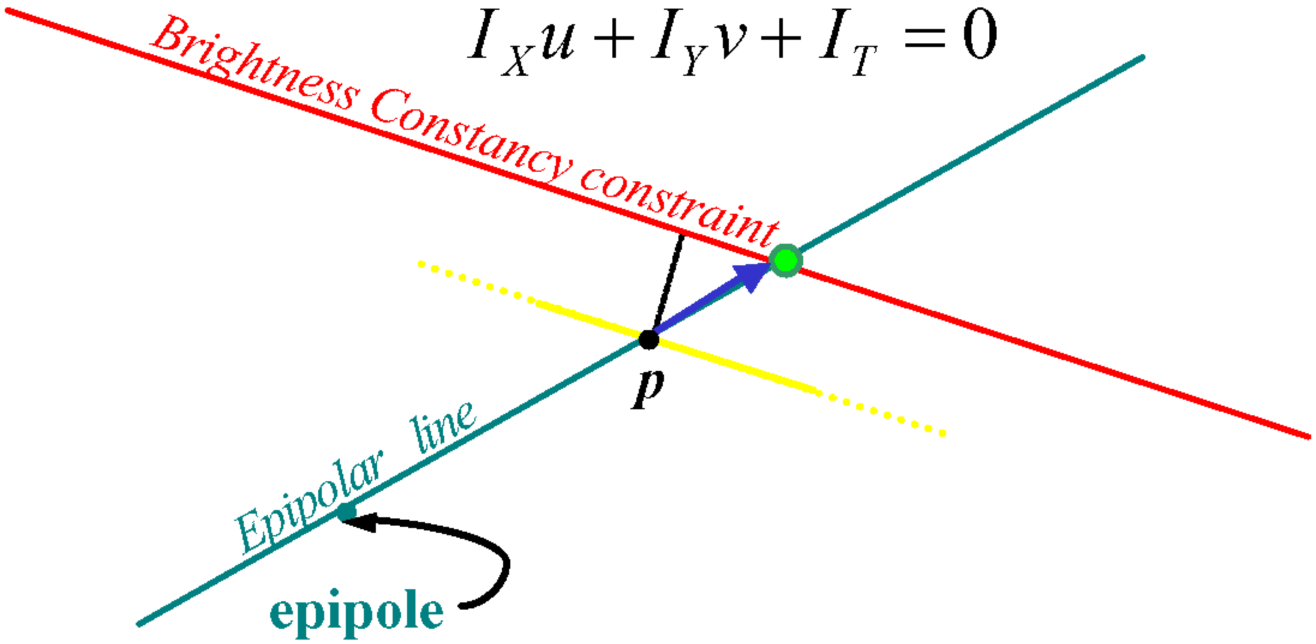


Original sequence **Plane-aligned sequence** **Recovered shape**

Block sequence from [Kumar-Anandan-Hanna'94]

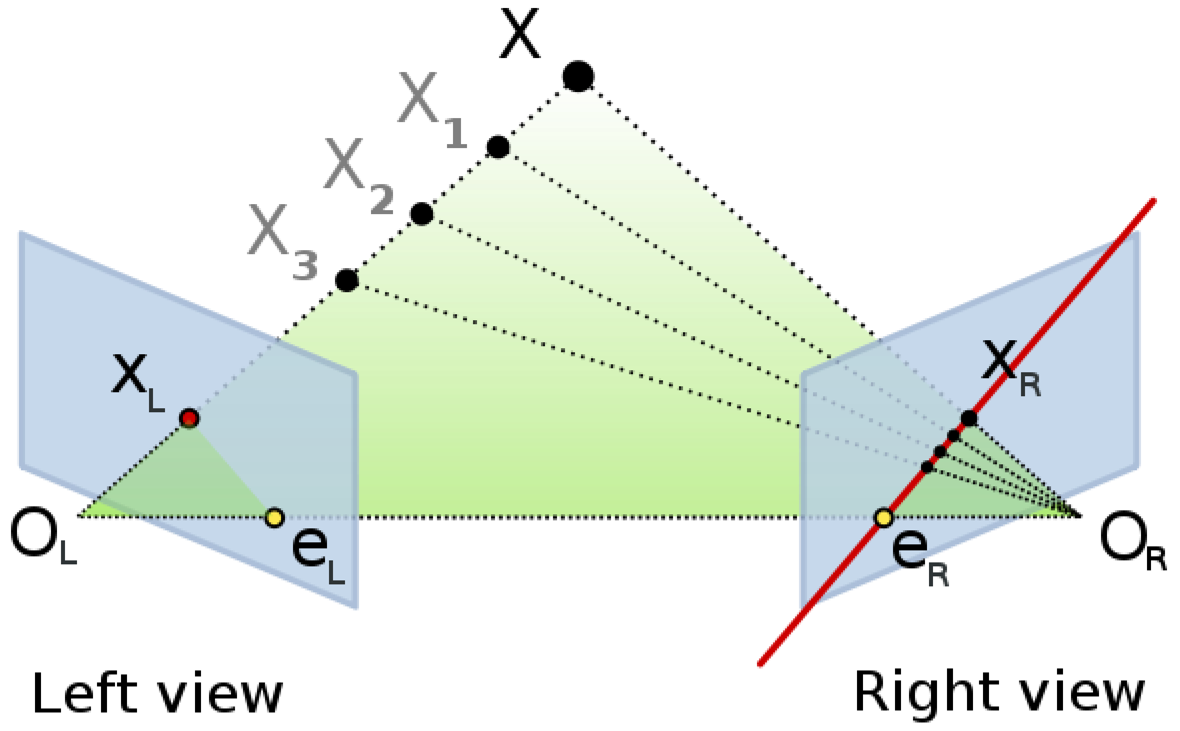
"Given two views where motion of points on a parametric surface has been compensated, the residual parallax is an epipolar field"

Residual Planar Parallax Motion

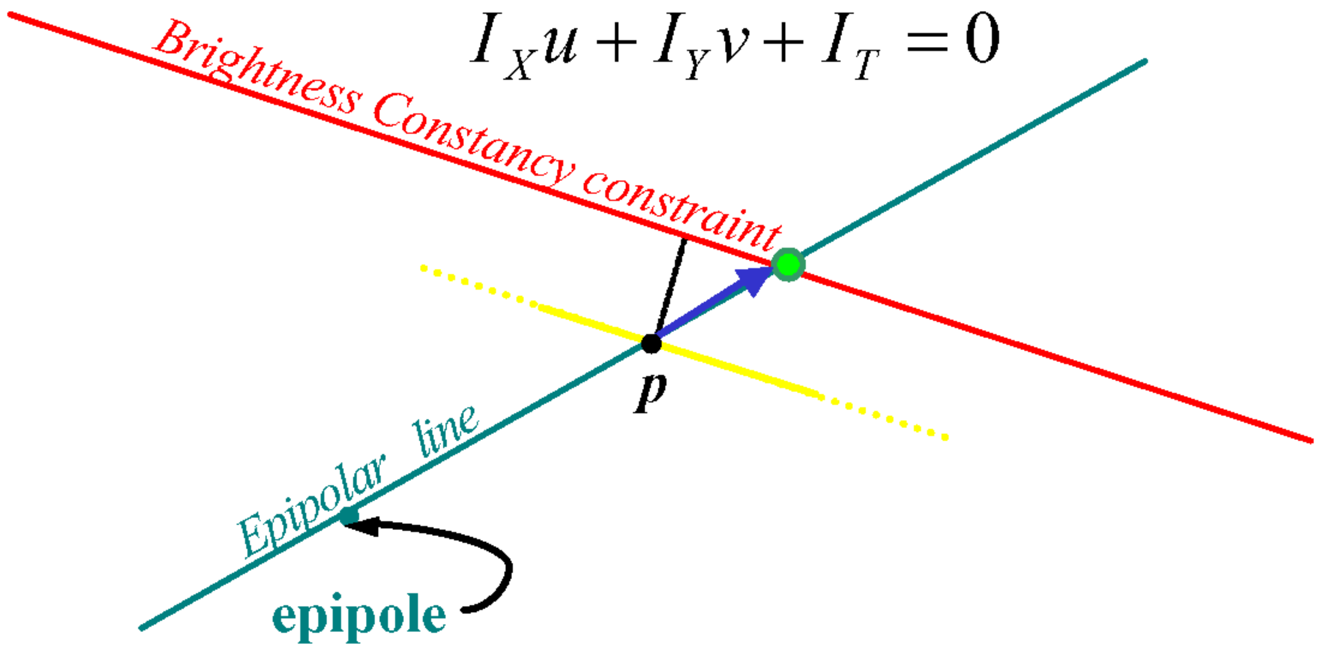


The intersection of the two line constraints <u>uniquely</u> defines the displacement.

Epipolar Geometry



Residual Planar Parallax Motion



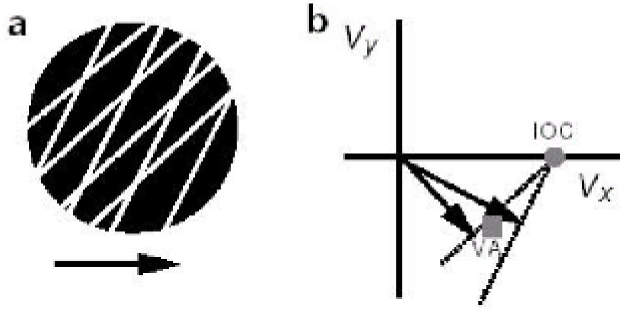
The intersection of the two line constraints <u>uniquely</u> defines the displacement.

Optical Flow

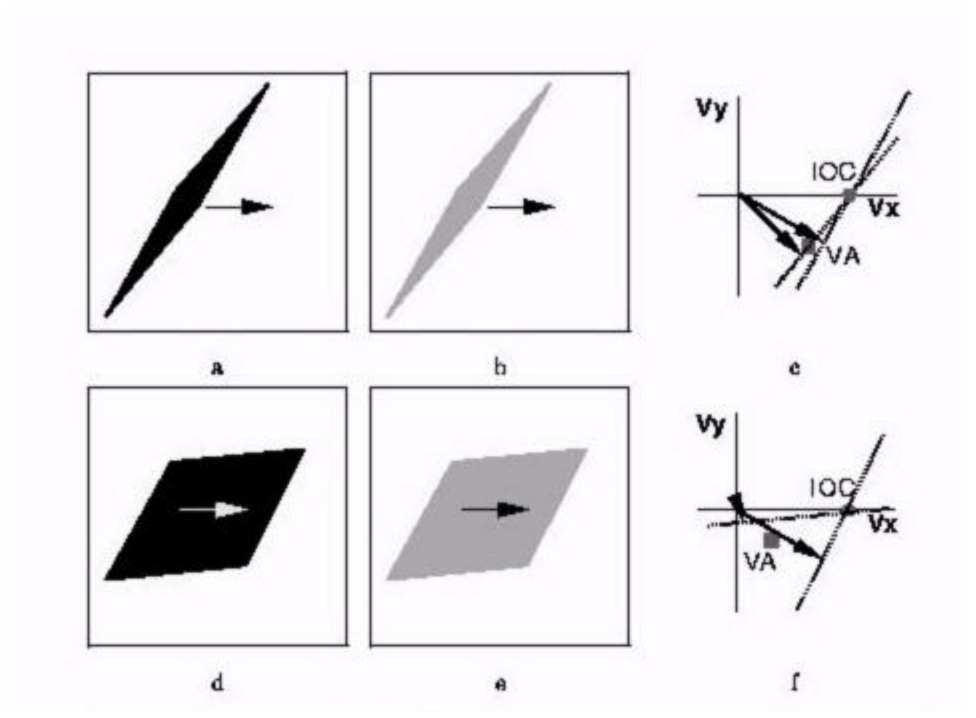
- Brightness Constancy
- The Aperture problem
- Regularization
- Lucas-Kanade
- Coarse-to-fine
- Parametric motion models
- SSD tracking
- Bayesian flow

Bayesian Optic Flow

- Some low-level human motion illusions can be explained by adding an uncertianty model to Lucas-Kanade tracking
- Theories from Psychology about normal flow fusion:
  - (VA) vector average (of normal motions)
  - (IOC) intersection of constraints (e.g., Lucas-Kanade):



Rhombus Displays



http://www.cs.huji.ac.il/~yweiss/Rhombus/

Brightness constancy with noise:

$$I(x,y,t) = I(x + v_x\Delta t,\ y + v_y\Delta t,\ t + \Delta t) + \eta$$

Assume Gaussian noise, smooth surfaces, locally constant; take first order linear approximation:

$$P(I(x_i,y_i,t)|v_i) \propto$$

$$\exp\left(-\frac{1}{2\sigma^2} \int_{x,y} w_i(x,y)\ (I_x(x,y,t)v_x + I_y(x,y,t)v_y + I_t(x,y,t))^2\, dx\, dy \right)$$

Prior favoring slow speeds:

$$P(v) \propto \exp(-\|v\|^2/2\sigma_p^{\ 2}).$$

Assume noise is independent across location; apply Bayes:

$$P(v|I) \propto P(v) \prod_i P(I(x_i,y_i,t)\ |v),$$

With constant window w=1,

$$P(v|I) \propto \exp\left(-\|v\|^2/2\sigma_p^{\ 2} - \frac{1}{2\sigma^2} \int_{x,y} (I(x,y)\ v_x + I_y(x,y)v_y + I_t)^2\, dx\, dy \right)$$

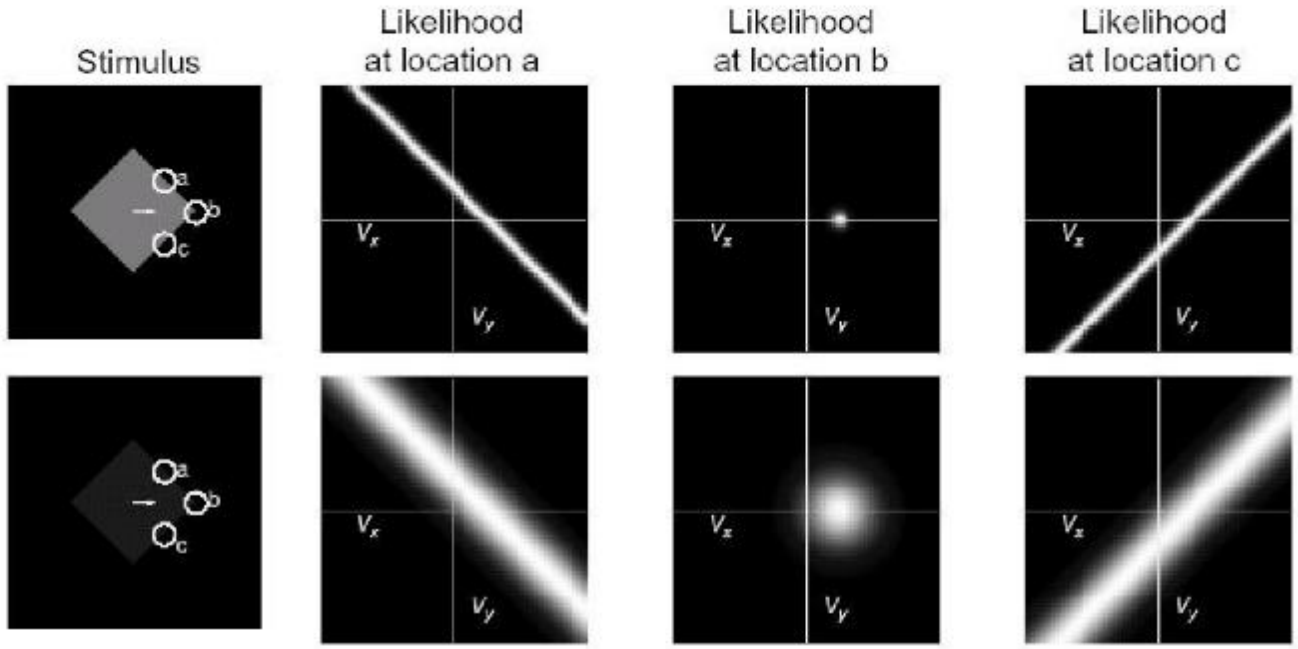
Form ‘normal equations’ to arrive at….

Lucas-Kanade with uncertainty:

$$v^* = -\begin{pmatrix} \sum I_x^2 + \frac{\sigma^2}{\sigma_p^2} & \sum I_x I_y \\ \sum I_x I_y & \sum I_y^2 + \frac{\sigma^2}{\sigma_p^2} \end{pmatrix}^{-1} \begin{pmatrix} \sum I_x I_t \\ \sum I_y I_t \end{pmatrix}$$

One parameter: ratio of observation and prior gaussian spread.

http://www.cs.huji.ac.il/~yweiss/Rhombus
[Weiss, Simoncelli, Adelson Nature Neuroscience 2002]


Stimulus
Likelihood at location a
Likelihood at location b
Likelihood at location c
a
b
c
V_x
V_y

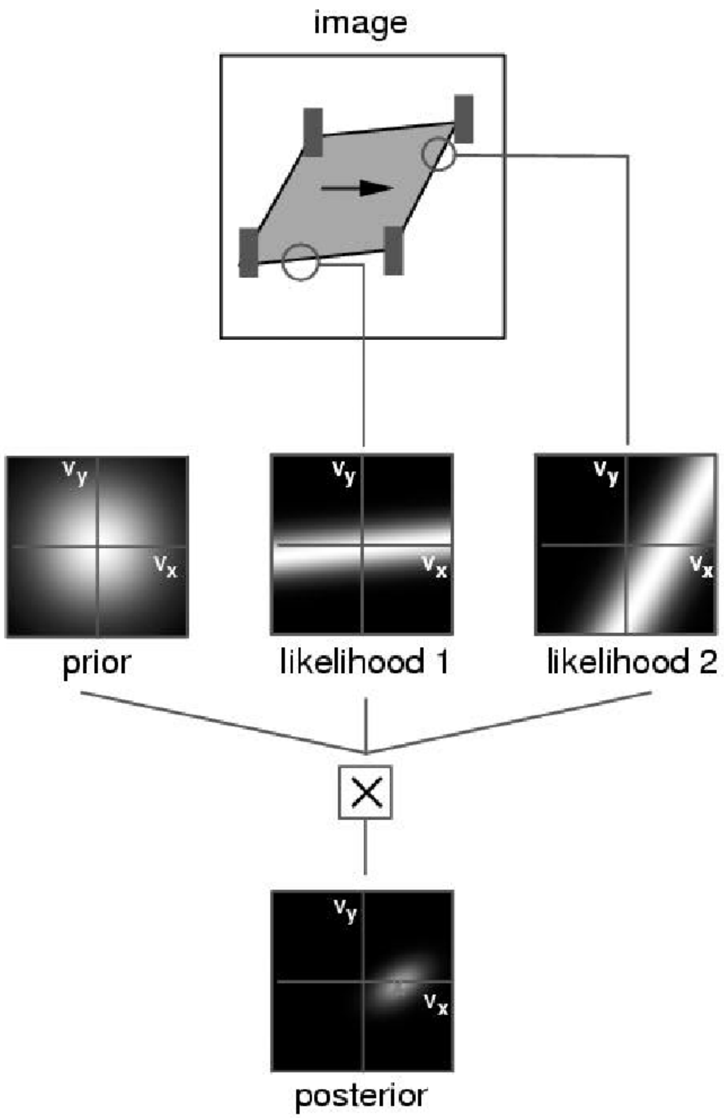


Figure 4: The response of the Bayesian estimator to a fat rhombus. (replotted from Weiss and Adelson 98)

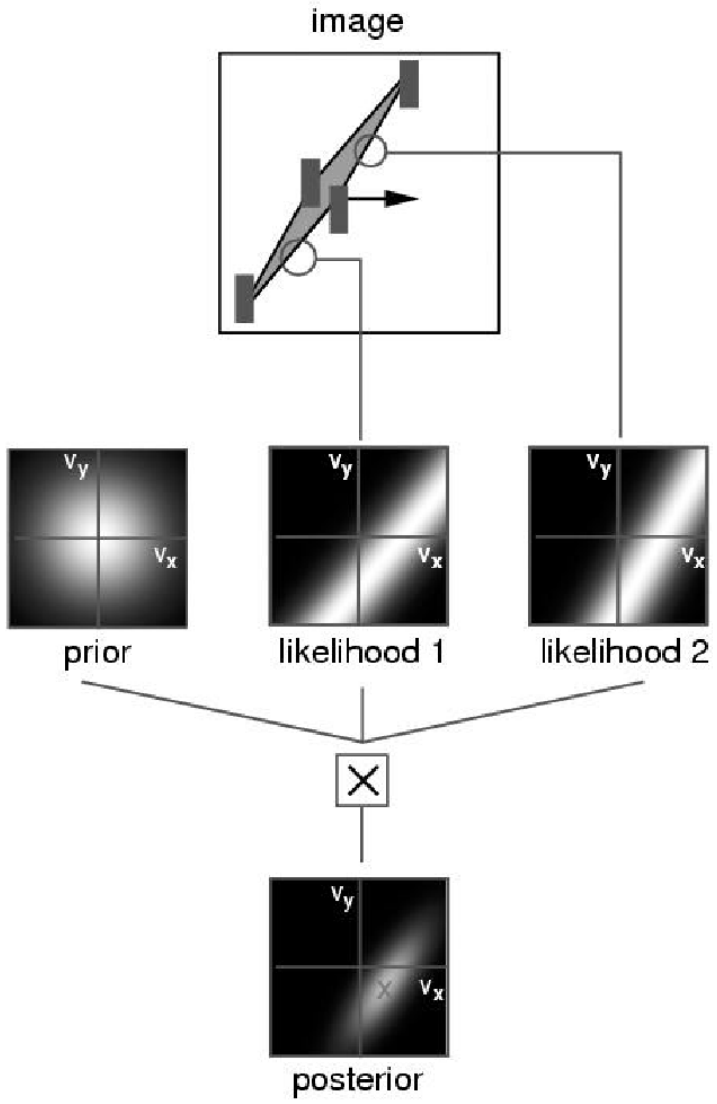
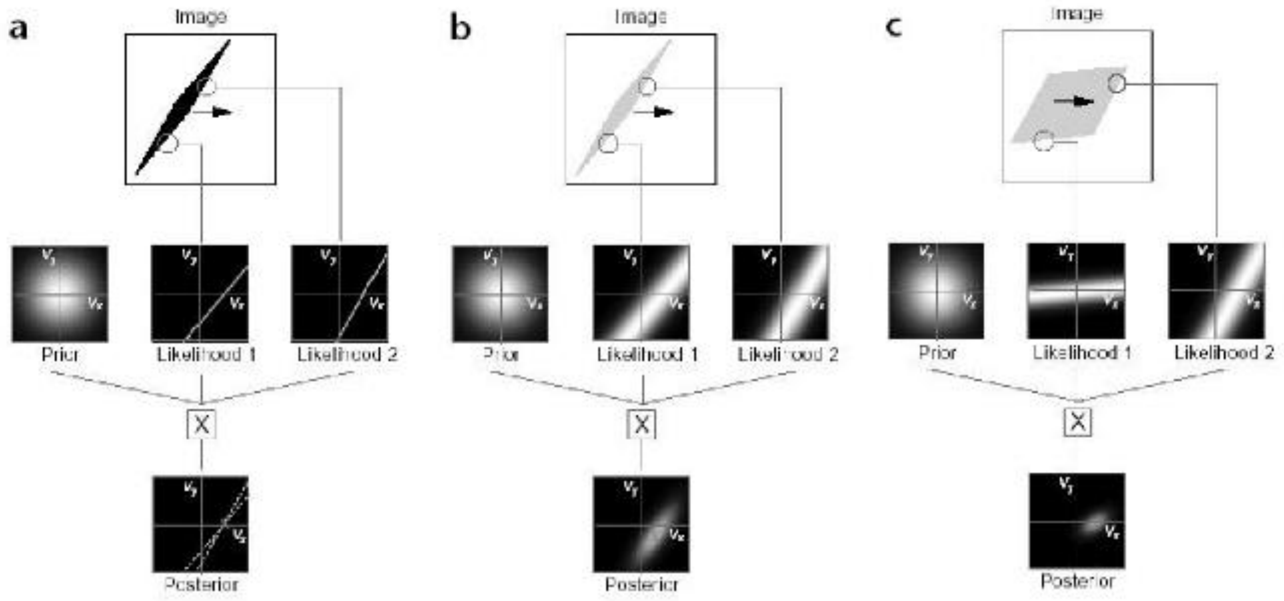


Figure 3: The response of the Bayesian estimator to a narrow rhombus. (replotted from Weiss and Adelson 98)

Effect of contrast



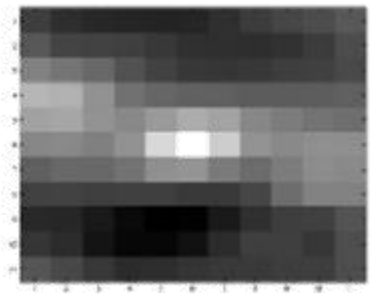
Optical Flow

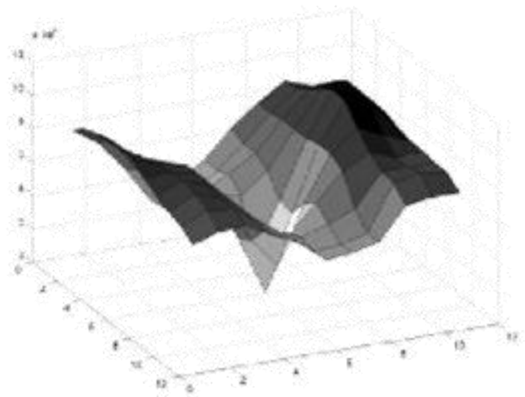
- Brightness Constancy
- The Aperture problem
- Regularization
- Lucas-Kanade
- Coarse-to-fine
- Parametric motion models
- SSD tracking
- Bayesian flow

Correlation and SSD

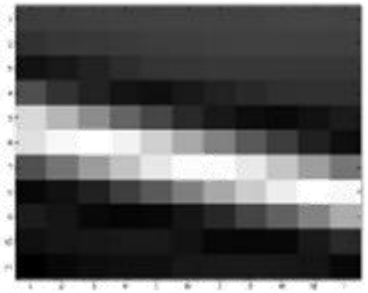
- For large displacements, do template matching as was used in stereo disparity search.
  - Define a small area around a pixel as the template
  - Match the template against each pixel within a search area in next image.
  - Use a match measure such as correlation, normalized correlation, or sum-of-squares difference
  - Choose the maximum (or minimum) as the match
  - Sub-pixel interpolation also possible

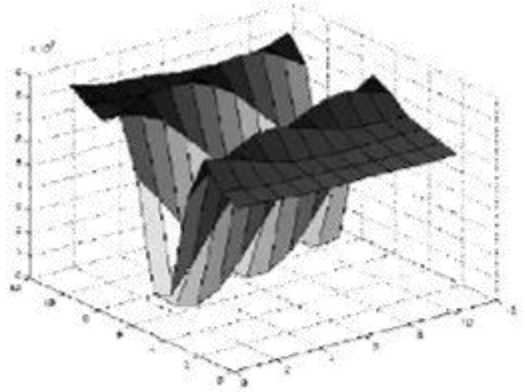
SSD Surface – Textured area



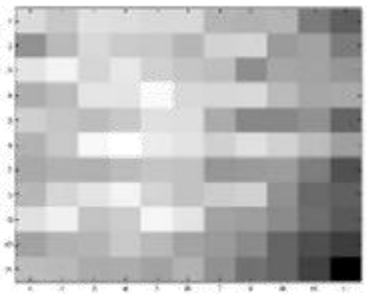


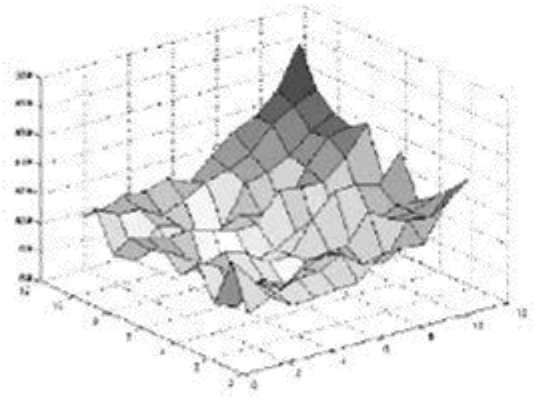
SSD Surface -- Edge





SSD Surface – homogeneous area





Discrete Search vs. Gradient Based Estimation

Consider image I translated by u_0, v_0

$$\boxed{\begin{aligned} I_0(x,y) &= I(x,y) \\ I_1(x+u_0, y+v_0) &= I(x,y) + \eta_1(x,y) \end{aligned}}$$

$$\boxed{\begin{aligned} E(u,v) &= \sum_{x,y} (I(x,y) - I_1(x+u, y+v))^2 \\ &= \sum_{x,y} (I(x,y) - I(x-u_0+u, y-v_0+v) - \eta_1(x,y))^2 \end{aligned}}$$

Discrete search simply searches for the best estimate.
Gradient method linearizes the intensity function and solves for the estimate

Visual Computing: Video Compression

Prof. Marc Pollefeys

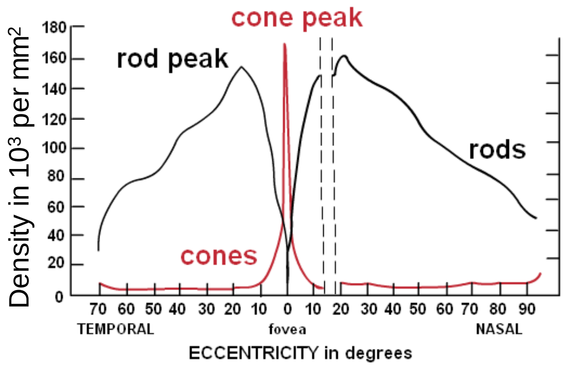
ETH

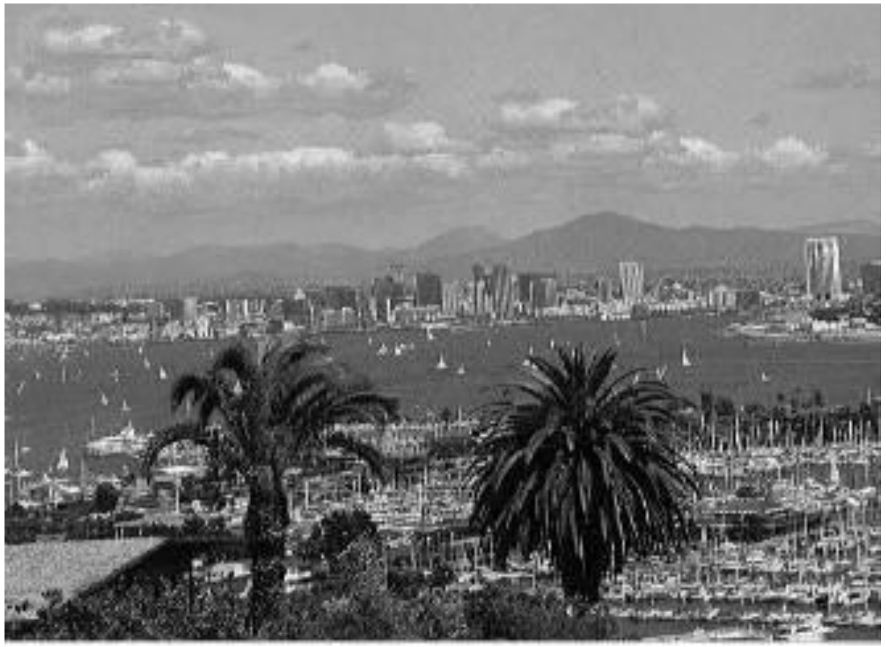
inf Informatik Computer Science

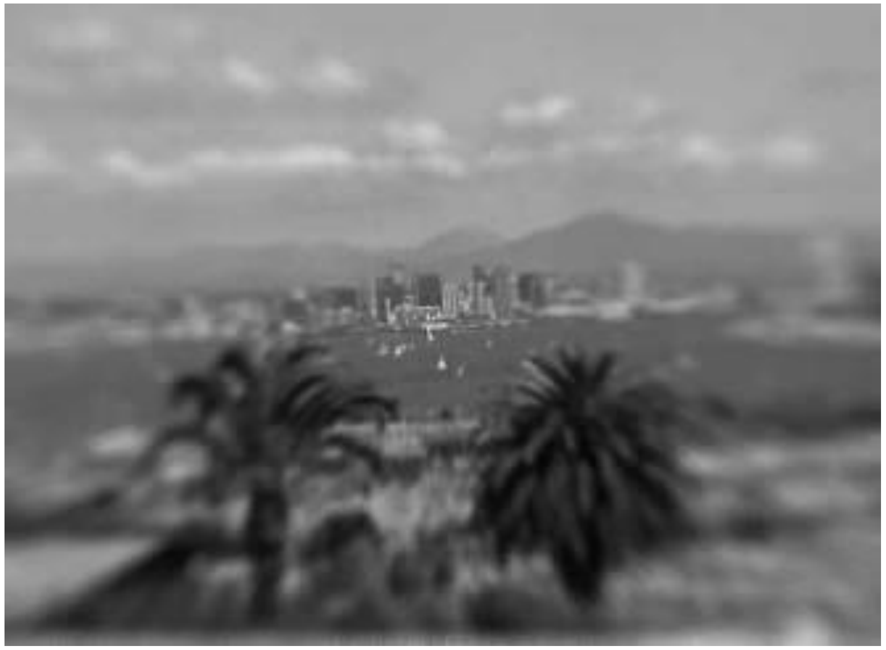
Perception of motion

- Human visual system is specifically sensitive to motion
- Eyes follow motion automatically
- Some distortions are not as perceivable as in image coding (would be if we froze frame)
- No good psycho-visual model available

Acuity of the Visual Field (small digression)



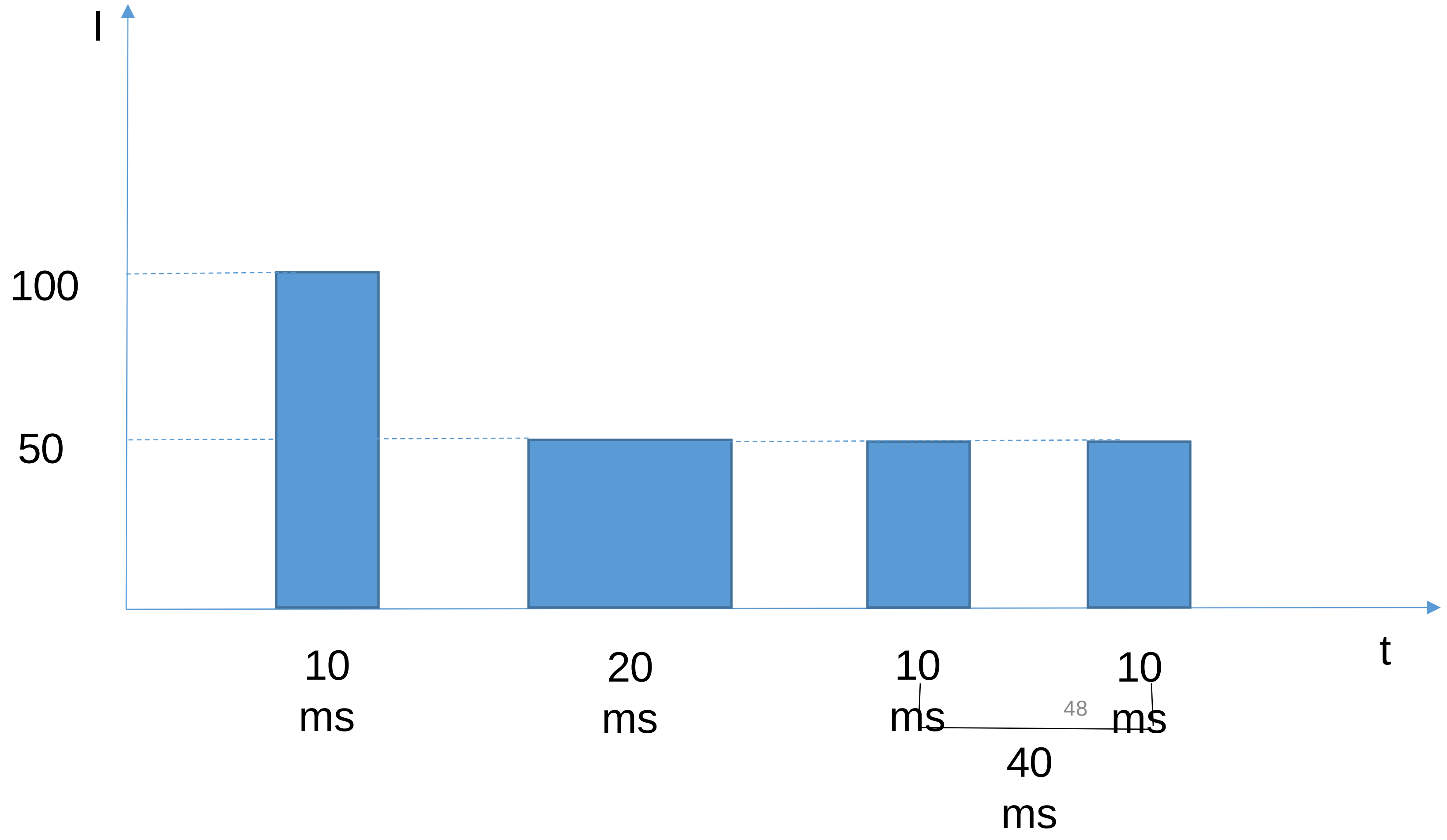




Perception of motion

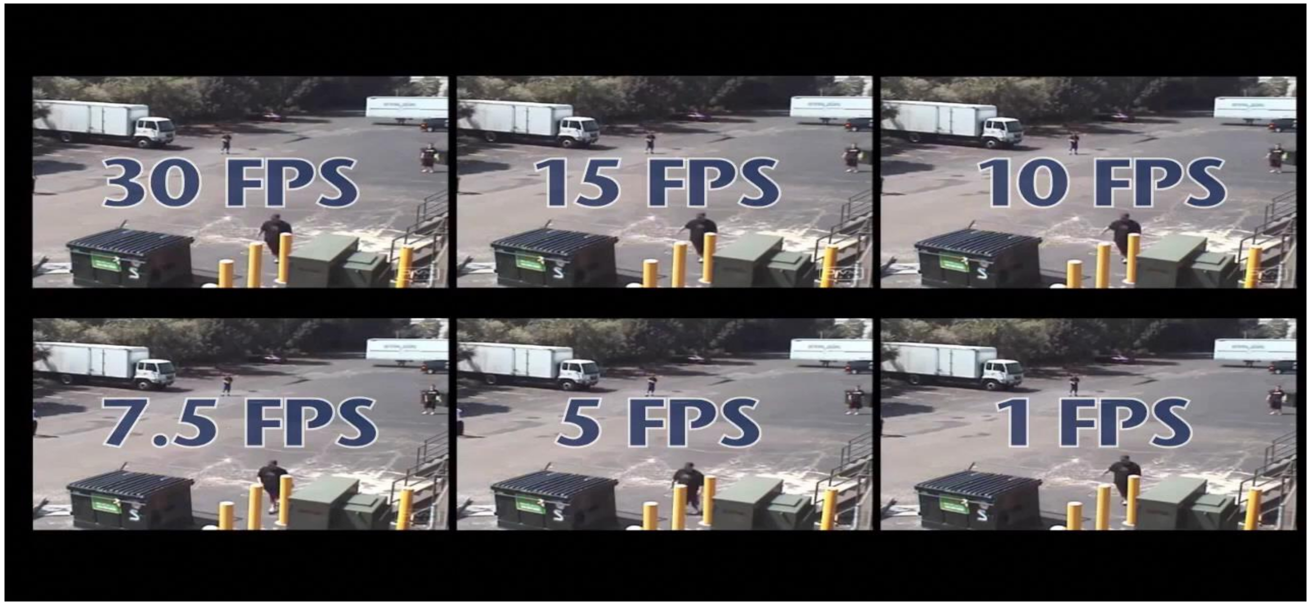
- Visual perception is limited to <24Hz
  - A succession of images will be perceived as continuous if frequency is sufficiently high
  - Cinema 24Hz, TV 25(50) Hz
- Still need to avoid aliasing (wheel effect)
  - High-rendering frame-rates desired in computer games (needed due to absence of motion blur)
- Flicker can be perceived up to >60Hz in particular in periphery
  - Issue addressed by 100Hz TV

Bloch's Law



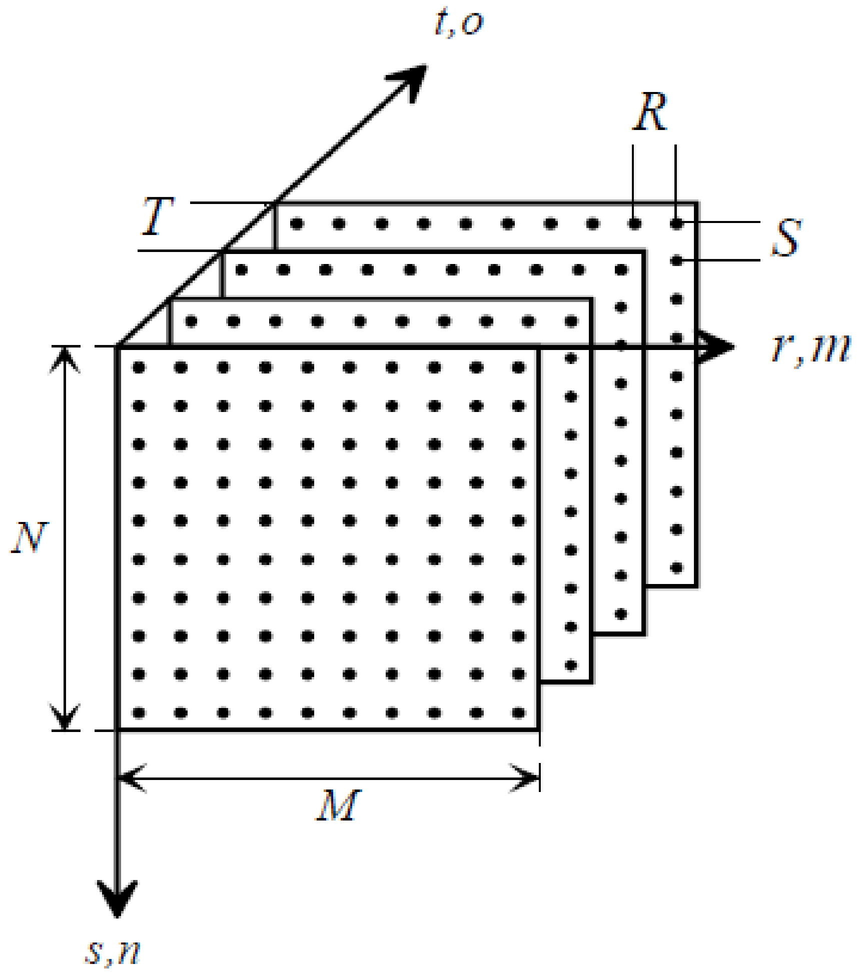
Bloch's Law - Implications

- Enforces limits on framerate for animations and videos (min 10 Hz)

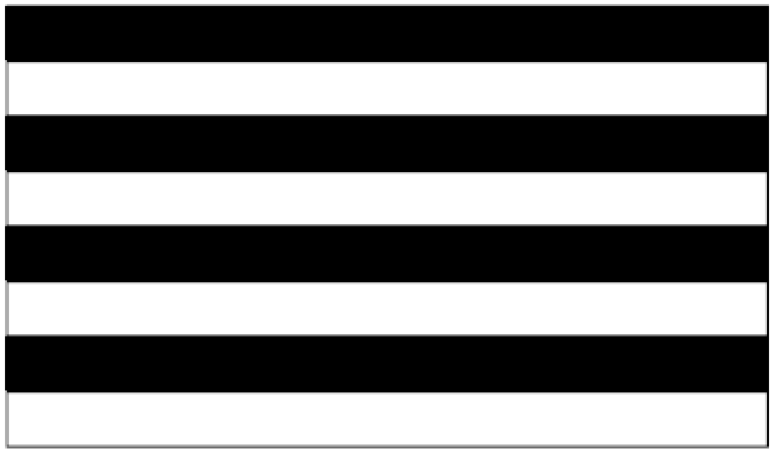


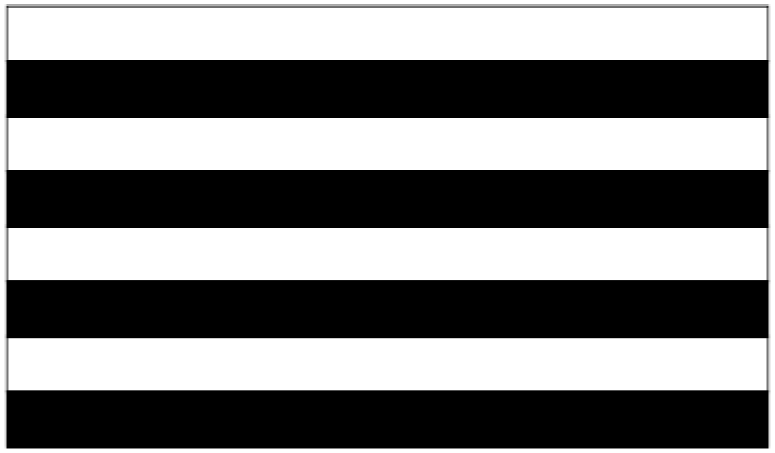
Video Format

- **2D+t :** Video sequence



Interlaced video format


top field


bottom field

- 2 temporally shifted half images, increase of frequency 25 -> 50 Hz
- Reduction of spatial resolution
- Full image representation: progressive

Why compress video?

- Raw HD TV signal 720p@50Hz

  1280x720x50x24bits/s = 1.105.920.000bits/s

  >1Gb/s

(e.g.1920x1080@60Hz>1Gb/s)

Only 20Mb/s HDTV channel bandwidth

Required compression factor of 60

(0.40bits/pixel on average)

Video Compression

XviD Configuration

About

Brightness

Postprocessing
- [✓] Deblocking (Y)
- [✓] Dering (Y)
- [ ] Film Effect
- [✓] Deblocking (UV)
- [✓] Dering (UV)

Output
- [ ] Rip video
- [ ] Compatibility Renderer

Output Colourspace: No Force

FourCC support
- [✓] DIVX
- [✓] DX50
- [✓] mp4v

Aspect Ratio

After restarting player, use this AR: Auto (mpeg-4 firs

Lossy video compression

- Take advantage of redundancy
  - Spatial correlation between neighboring pixels
  - Temporal correlation between frames
- Drop perceptually unimportant details

Temporal Redundancy

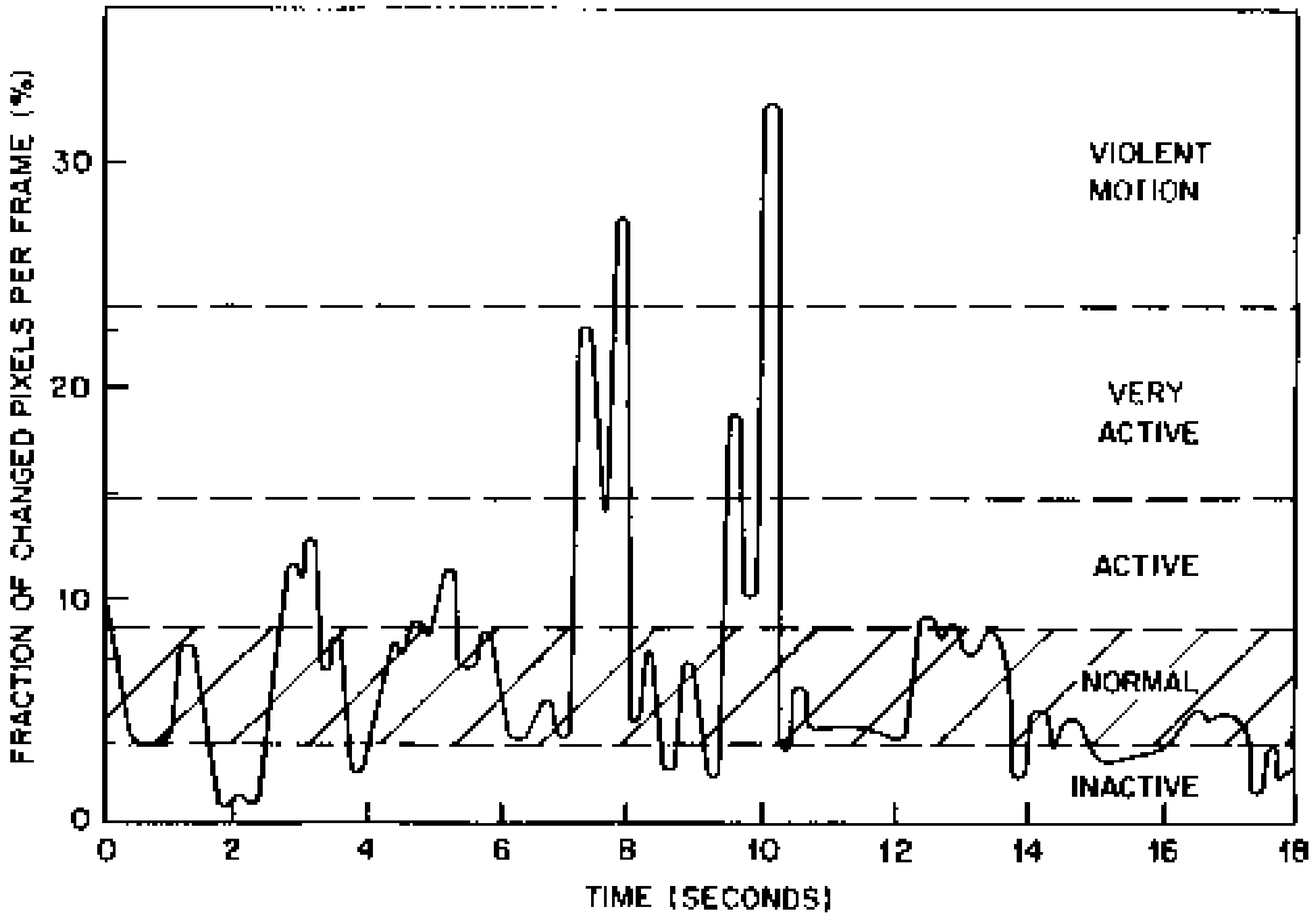
- Take advantage of similarity between successive frames

950

951

952

Temporal Activity



“Talking Head”

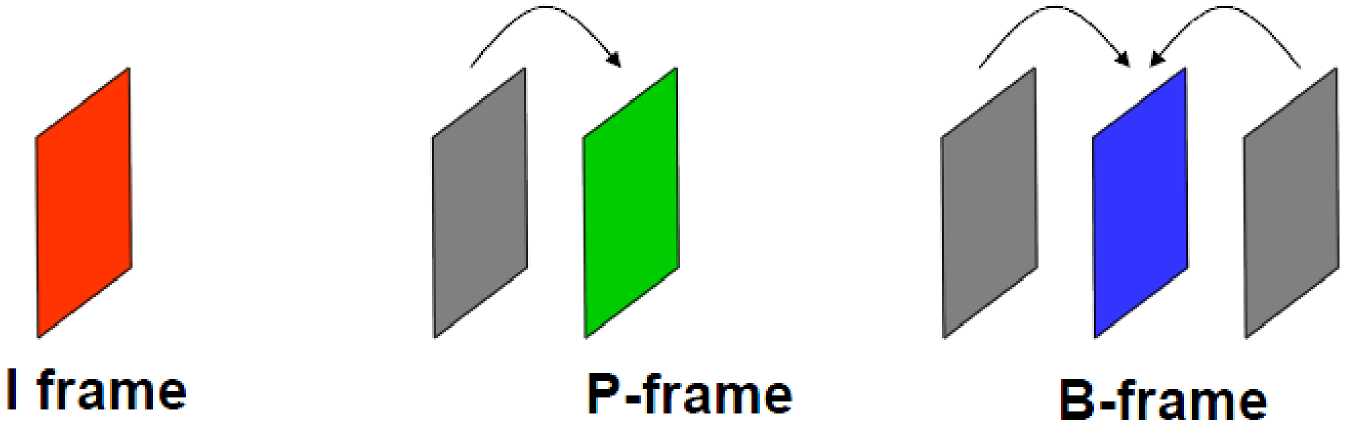
Temporal processing

- Usually high frame rate: *Significant temporal redundancy*
- Possible representations along temporal dimension:
  - *Transform/subband methods*
    - Good for textbook case of constant velocity uniform global motion
    - Inefficient for nonuniform motion, I.e. real-world motion
    - Requires large number of frame stores
      - Leads to delay (Memory cost may also be an issue)
  - *Predictive methods*
    - Good performance using only 2 frame stores
    - However, simple frame differencing in not enough…

Slide from John G. Apostolopoulos

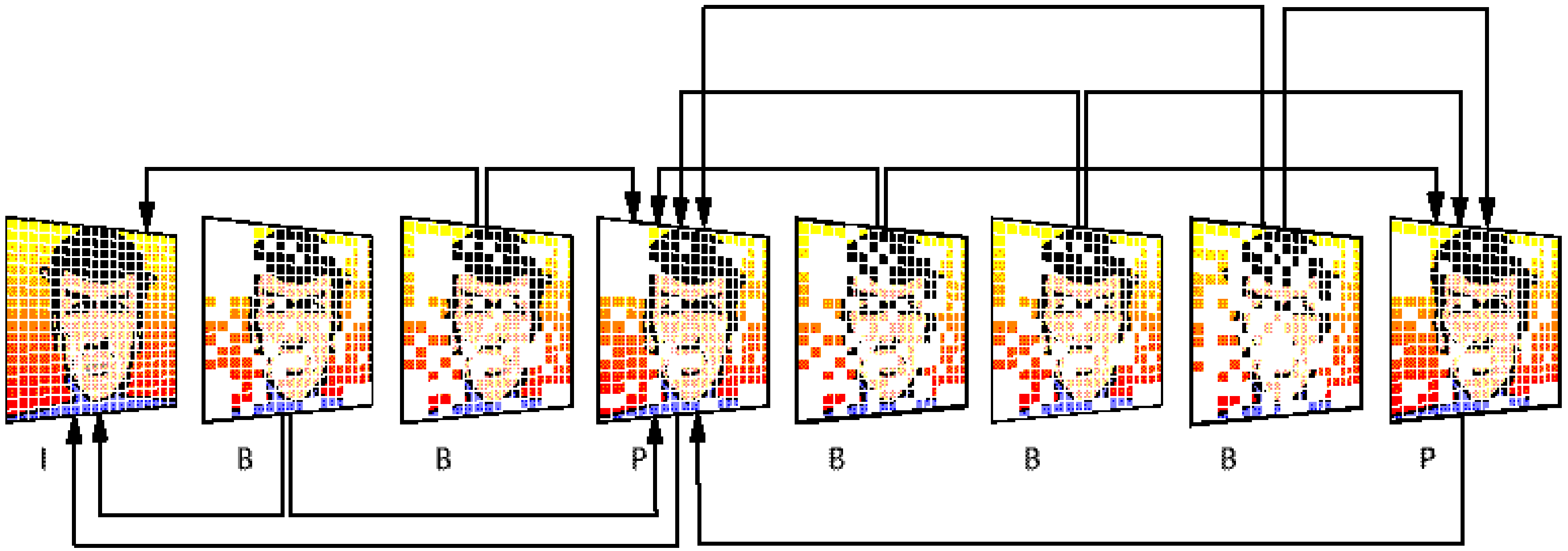
Video Compression

- Goal: Exploit the temporal redundancy
- *Predict current frame* based on previously coded frames
- Three types of coded frames:
  - ***I-frame:*** Intra-coded frame, coded independently of all other frames
  - ***P-frame:*** Predictively coded frame, coded based on previously coded frame
  - ***B-frame:*** Bi-directionally predicted frame, coded based on both previous and future coded frames



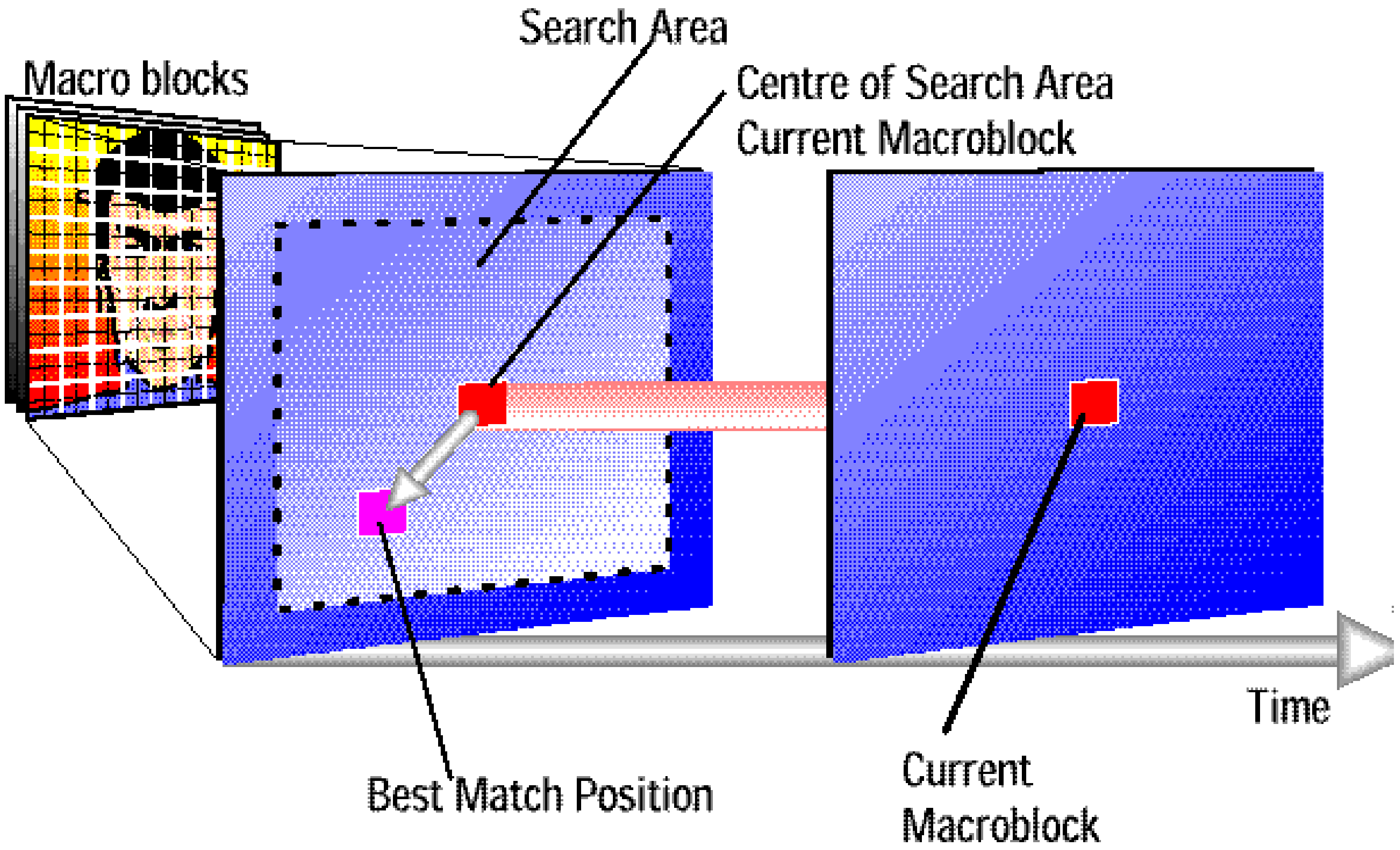
Slide from John G. Apostolopoulos

Temporal Redundancy Reduction

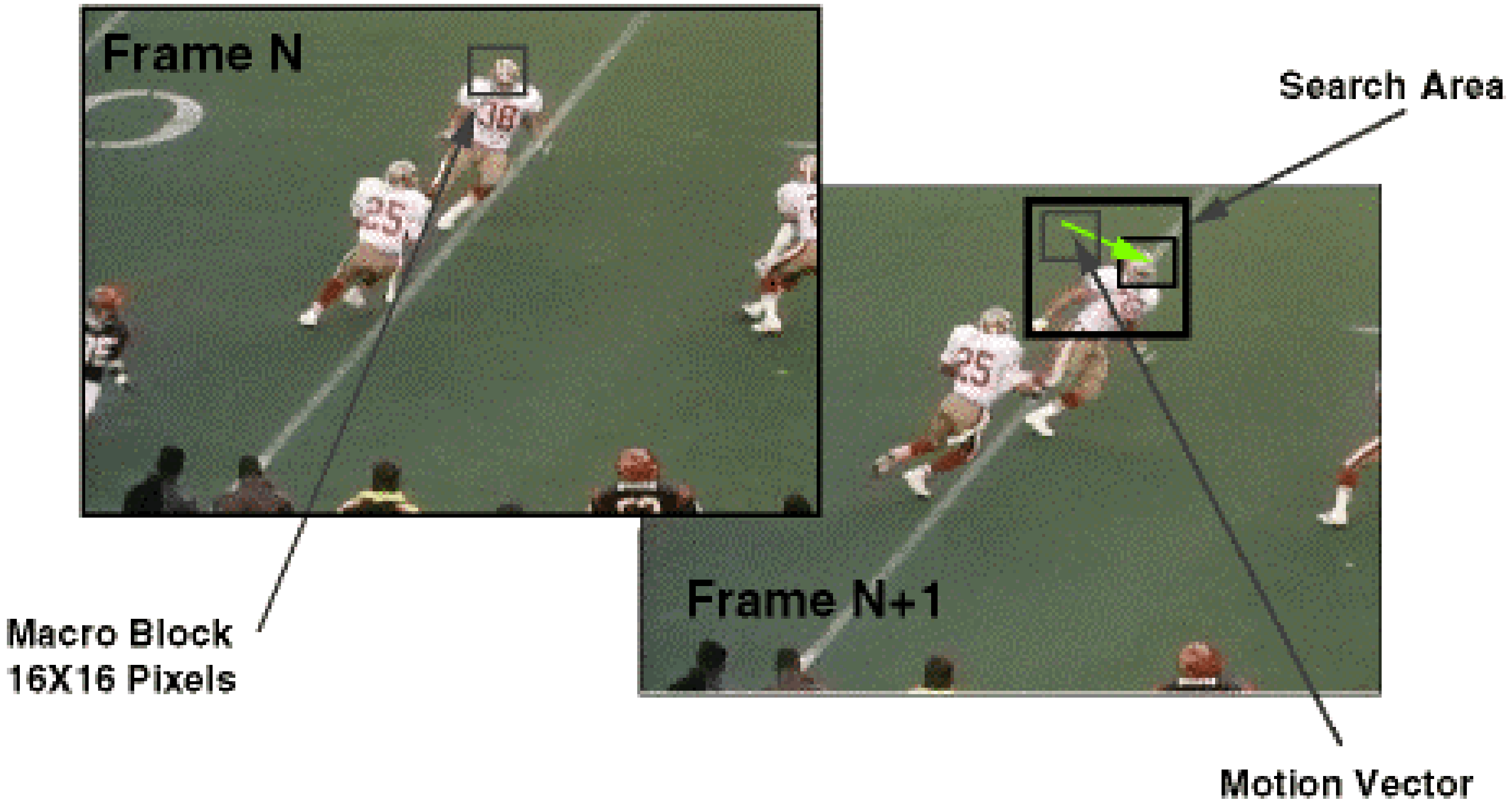


- *I* frames are independently encoded
- *P* frames are based on previous I, P frames
  - Can send motion vector plus changes
- *B* frames are based on previous and following I and P frames
  - In case something is uncovered

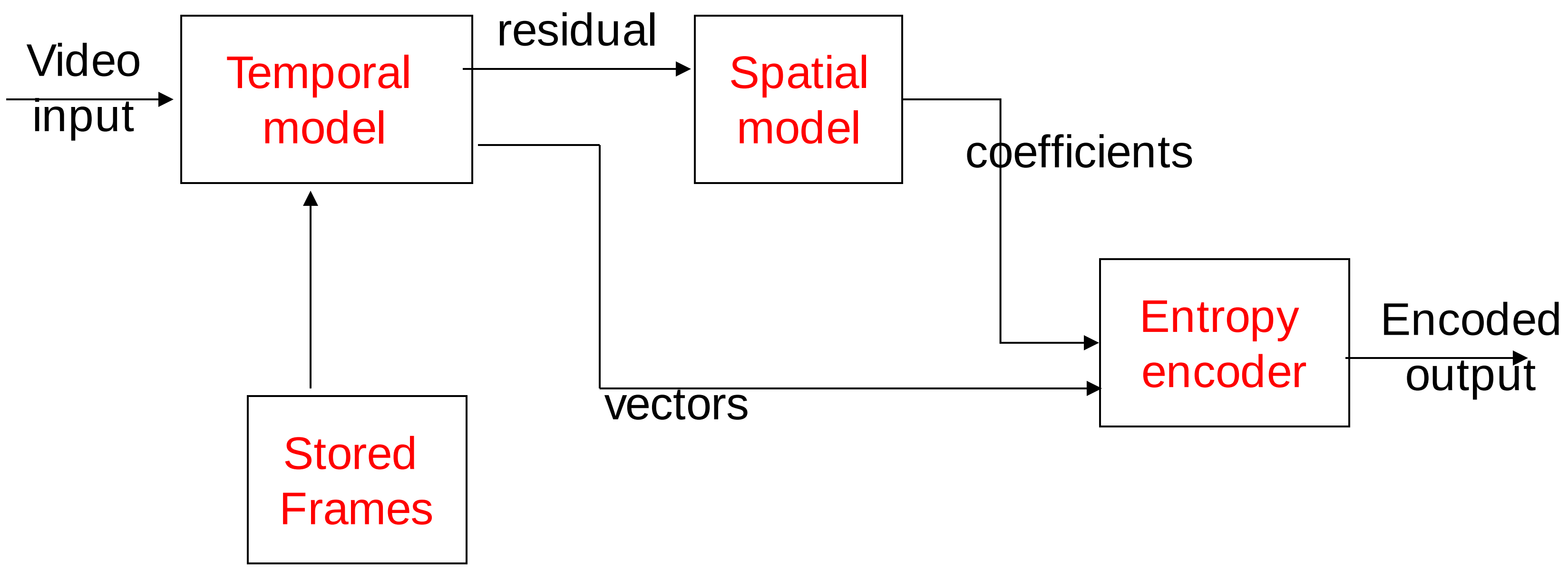
Temporal Redundancy Reduction



Temporal Redundancy Reduction



Video compressor diagram



Question

- When may temporal redundancy reduction be ineffective?

Answer

- *When may temporal redundancy reduction be ineffective?*
  - Many scene changes
  - High motion

Non-Temporal Redundancy

- Many scene changes

Non-Temporal Redundancy

- Sometimes high motion

Temporal processing: Motion-compensated prediction

- Simple frame differencing *fails* when there is motion
- *Must account for motion*

  $\rightarrow$ *Motion-compensated (MC) prediction*
- MC-prediction generally provides significant improvements
- Questions:
  - How can we estimate motion?
  - How can we form MC-prediction?

Slide from John G. Apostolopoulos

Temporal processing: Motion estimation

- Ideal situation:
  - Partition video into moving objects
  - Describe object motion

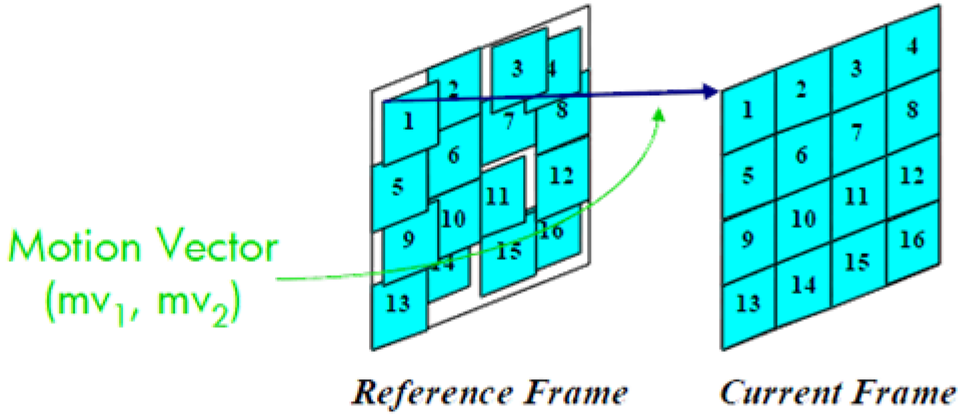
  → Generally very difficult
- Practical approach: *Block-Matching Motion Estimation*
  - Partition each frame into blocks, e.g. 16x16 pixels
  - Describe motion of each block

  → No object identification required

  → Good, robust performance

Slide from John G. Apostolopoulos

Block-matching motion estimation



- *Assumptions:*
  - Translational motion within block:

$$f(n_1, n_2, k_{cur}) = f(n_1 - mv_1, n_2 - mv_2, k_{ref})$$

  - All pixels within each block have the same motion
- *ME Algorithm:*
  1) Divide current frame into non-overlapping $N_1 x N_2$ blocks
  2) For each block, find the *best matching block* in reference frame
- *MC-Prediction Algorithm:*
  - Use best matching blocks of reference frame as prediction of blocks in current frame

ETH

Slide from John G. Apostolopoulos

Block-matching: determining the best matching block

- For each block in the current frame search for best matching block in the reference frame
  - *Metrics* for determining "best match":

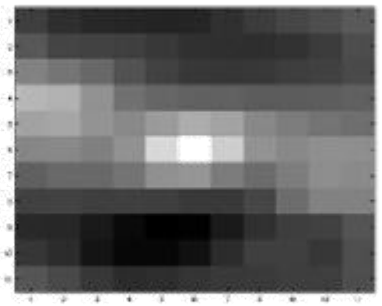
$$MSE = \sum_{(n_1,n_2)\in Block} \sum \left[f(n_1, n_2, k_{cur}) - f(n_1 - mv_1, n_2 - mv_2, k_{ref}) \right]^2$$

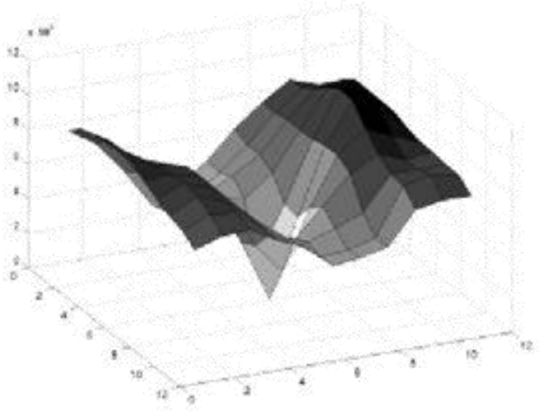
$$MAE = \sum_{(n_1,n_2)\in Block} \sum \left| f(n_1, n_2, k_{cur}) - f(n_1 - mv_1, n_2 - mv_2, k_{ref}) \right|$$

  - *Candidate blocks*: All blocks in, e.g., $(\pm 32, \pm 32)$ pixel area
  - *Strategies for searching* candidate blocks for best match
    - Full search: Examine all candidate blocks
    - Partial (fast) search: Examine a carefully selected subset
- Estimate of motion for best matching block: *"motion vector"*

Slide from John G. Apostolopoulos

SSD Surface – Textured area





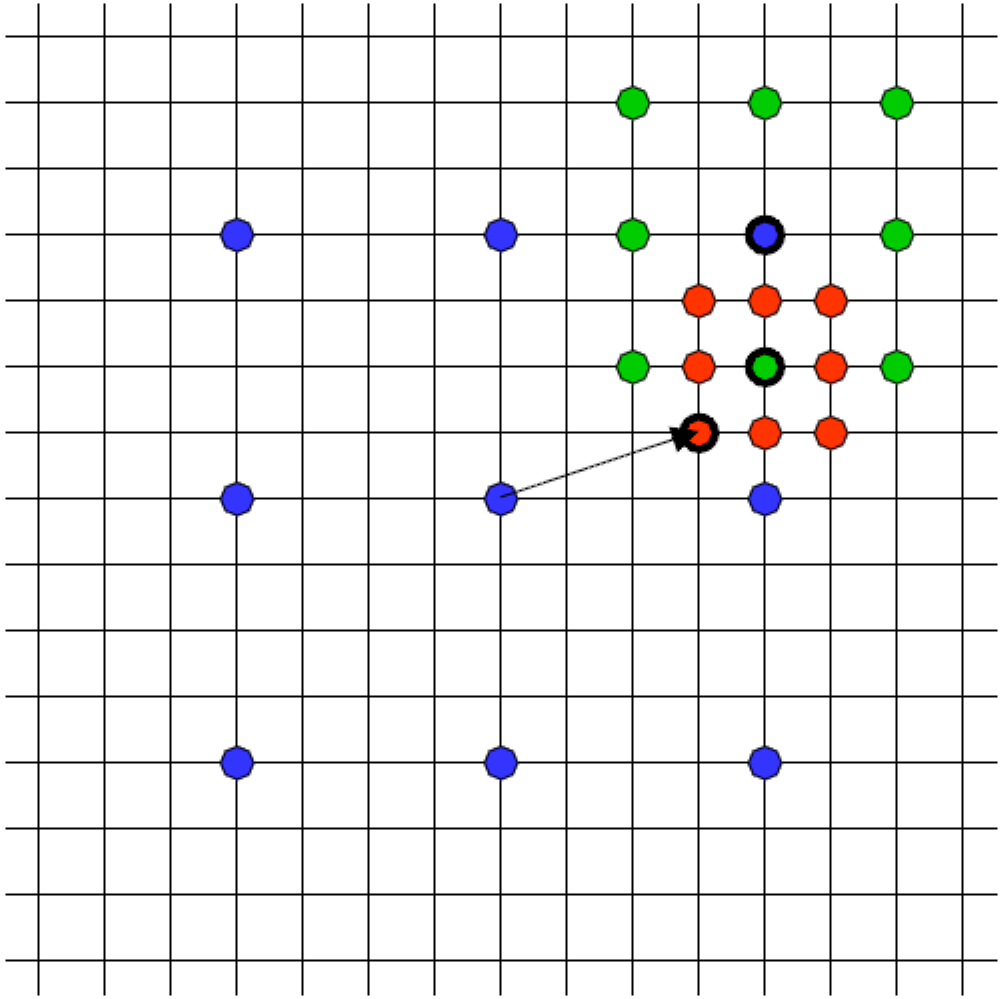
(from last lecture)

Motion vector and motion vector field

- *Motion vector*
  - Expresses the *relative horizontal and vertical offsets* *(mv_1,mv_2)*, or motion, of a given block from one frame to another
  - Each block has its own motion vector
- *Motion vector field*
  - Collection of motion vectors for all the blocks in a frame

Slide from John G. Apostolopoulos

Example of Fast Motion Estimation Search: 3-Step (Log) Search



- *Goal: Reduce number of search points*
- Example: $(\pm 7, \pm 7)$ search area
- Dots represent search points
- Search performed in 3 steps (coarse-to-fine):
  - Step 1: ● $(\pm 4\, pixels)$
  - Step 2: ● $(\pm 2\, pixels)$
  - Step 3: ● $(\pm 1\, pixels)$
- Best match is found at each step
- Next step: Search is centered around the best match of prior step
- Speedup increases for larger search areas

Slide from John G. Apostolopoulos

Motion Vector Precision?

- Motivation:
  - *Motion is not limited to integer-pixel offsets*
  - However, video only known at discrete pixel locations
  - To estimate sub-pixel motion, frames must be *spatially interpolated*
- Fractional MVs are used to represent the sub-pixel motion
- Improved performance (extra complexity is worthwhile)
- Half-pixel ME used in most standards: MPEG-1/2/4
- *Why are half-pixel motion vectors better?*
  - Can capture half-pixel motion
  - Averaging effect (from spatial interpolation) reduces prediction error → Improved prediction
  - For noisy sequences, averaging effect reduces noise → Improved compression

ETH

Slide from John G. Apostolopoulos

Practical Half-Pixel Motion Estimation Algorithm

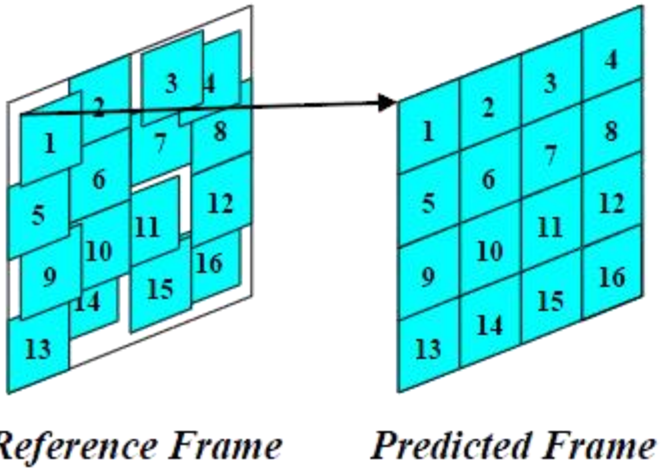
- *Half-pixel ME (coarse-fine) algorithm:*
  1) *Coarse step:* Perform integer motion estimation on blocks; find best integer-pixel MV
  2) *Fine step:* Refine estimate to find best half-pixel MV
     a) Spatially interpolate the selected region in reference frame
     b) Compare current block to interpolated reference frame block
     c) Choose the integer or half-pixel offset that provides best match
- Typically, bilinear interpolation is used for spatial interpolation

Slide from John G. Apostolopoulos

Example: MC-Prediction for Two Consecutive Frames

Previous Frame
(Reference Frame)

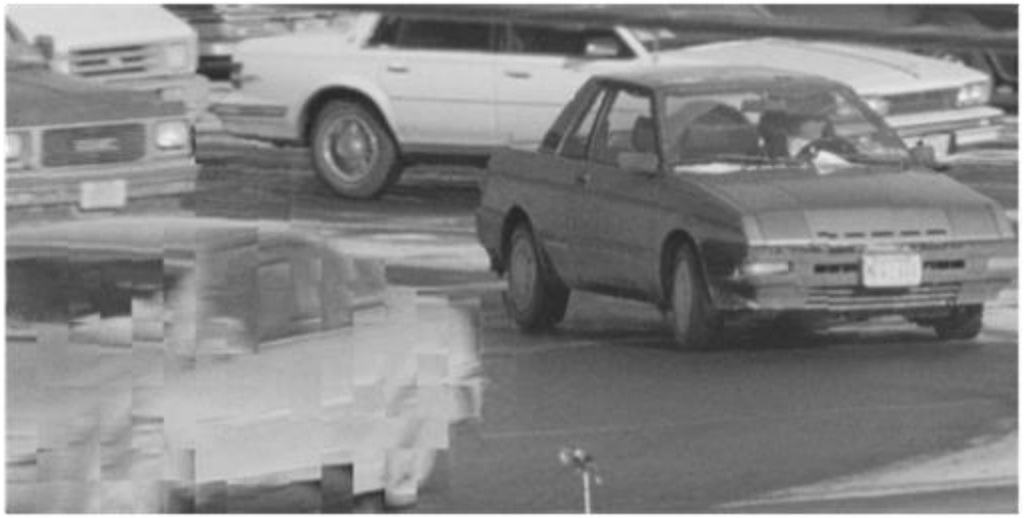
Current Frame
(To be Predicted)



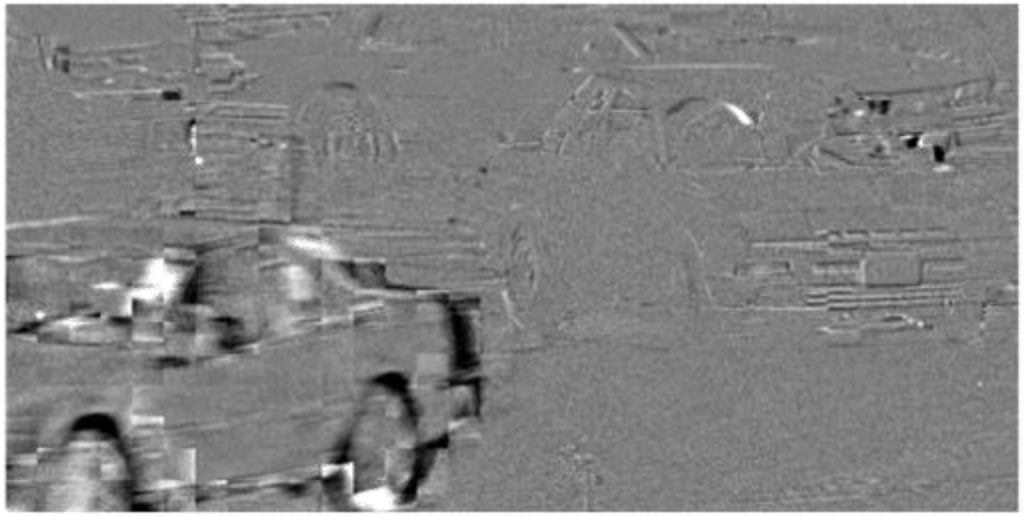
Slide from John G. Apostolopoulos

Example: MC-Prediction for Two Consecutive Frames

Prediction of Current Frame



Prediction Error (Residual)

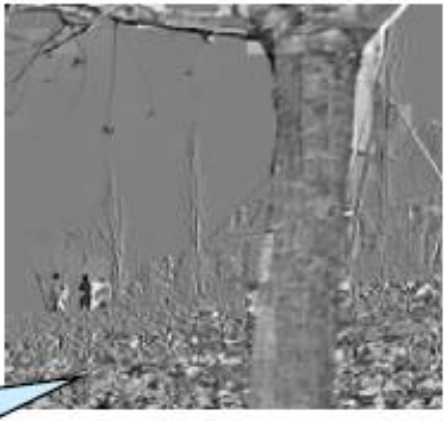


Slide from John G. Apostolopoulos

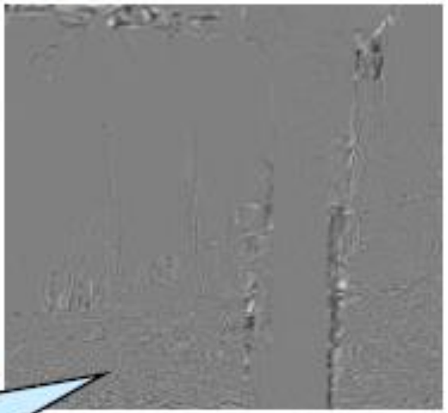
MC-prediction

Image of a video sequence

Difference to previous image



Without Motion compensation

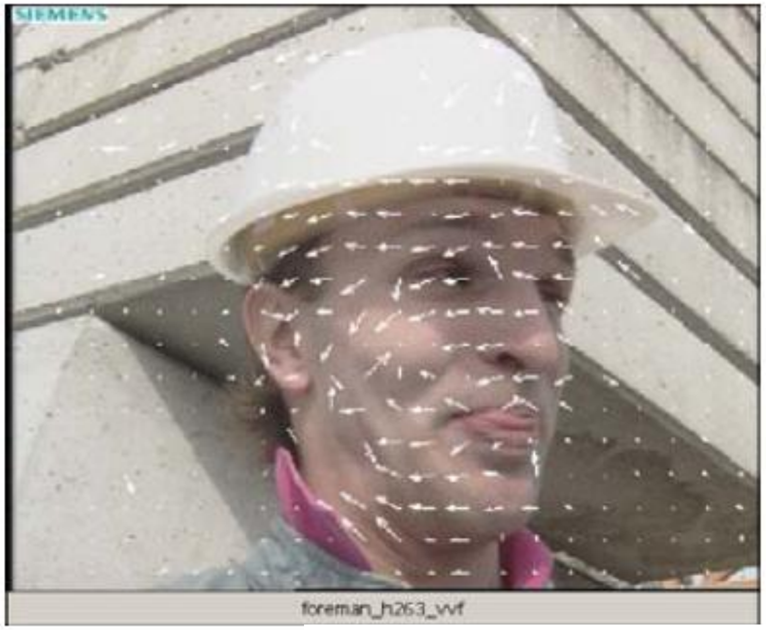


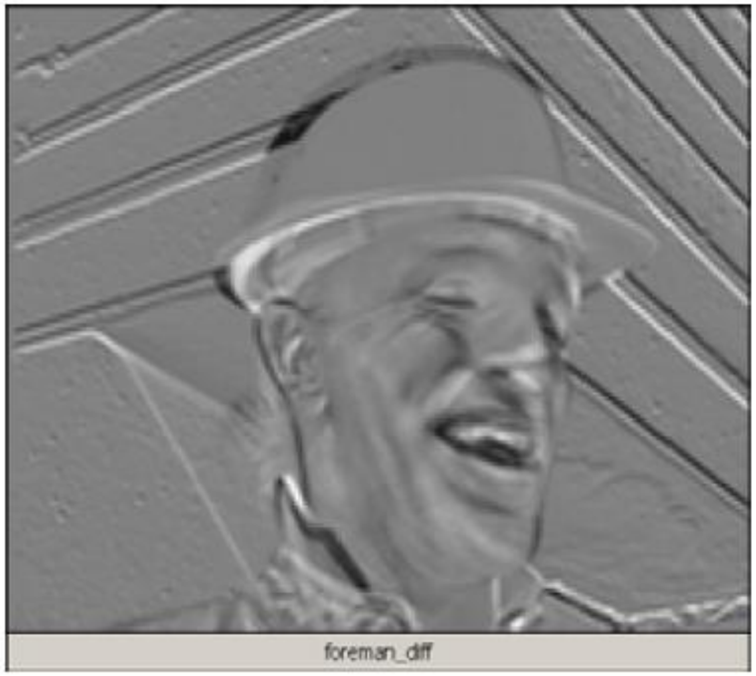
With Motion compensation

ETH

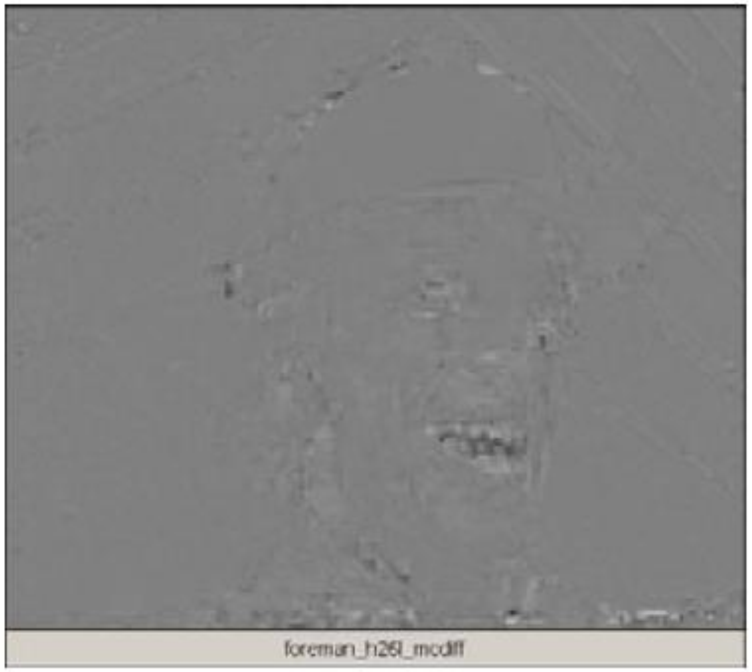
Slide from Aljoscha Smolic

Example MC





Without motion compensation



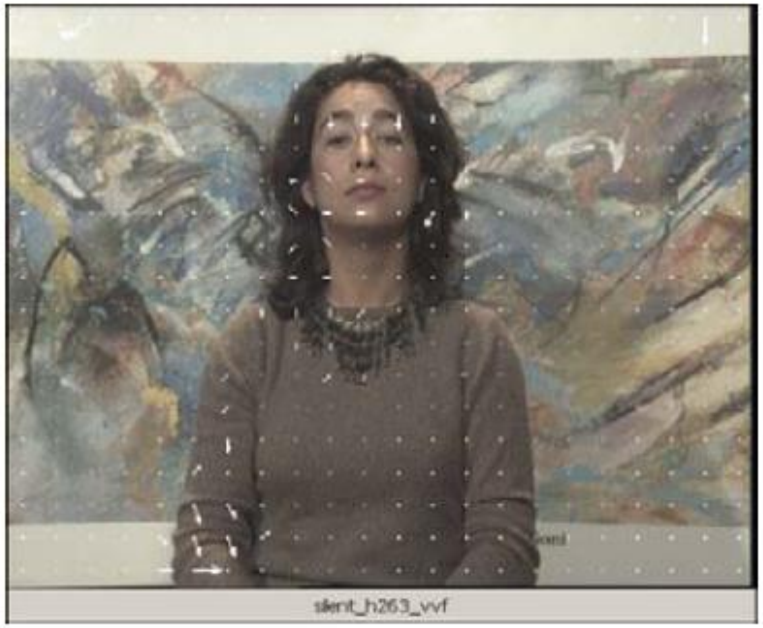
With motion compensation

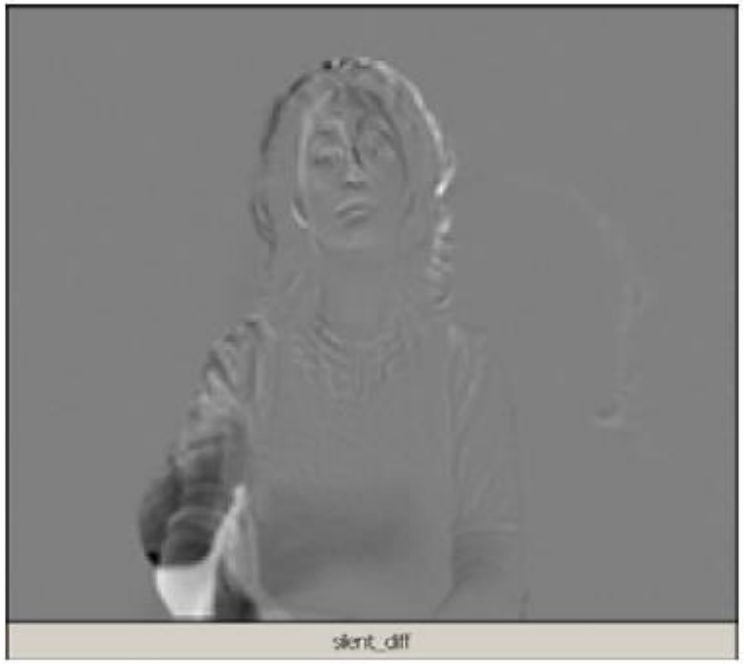
CIF Format (352x288)

ETH

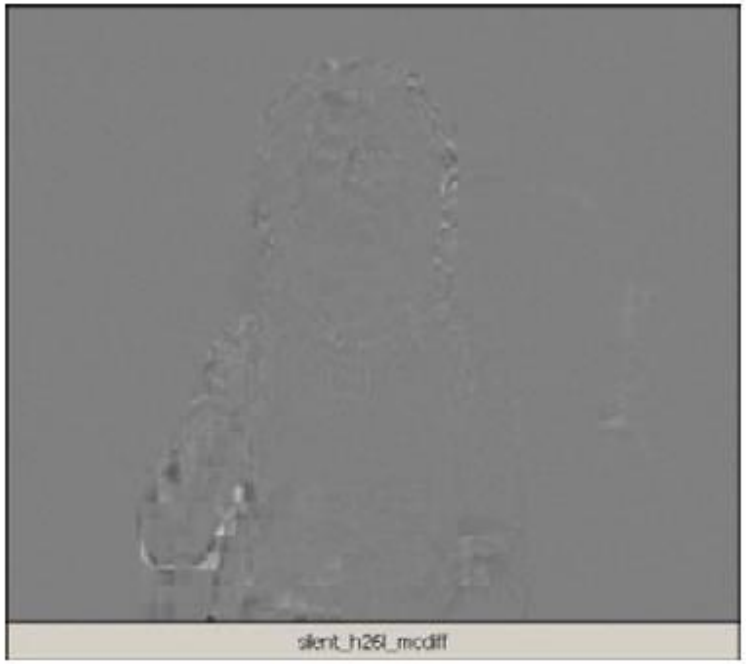
Slide from Aljoscha Smolic

Example MC





Without motion compensation



With motion compensation

QCIF Format
(176x144)

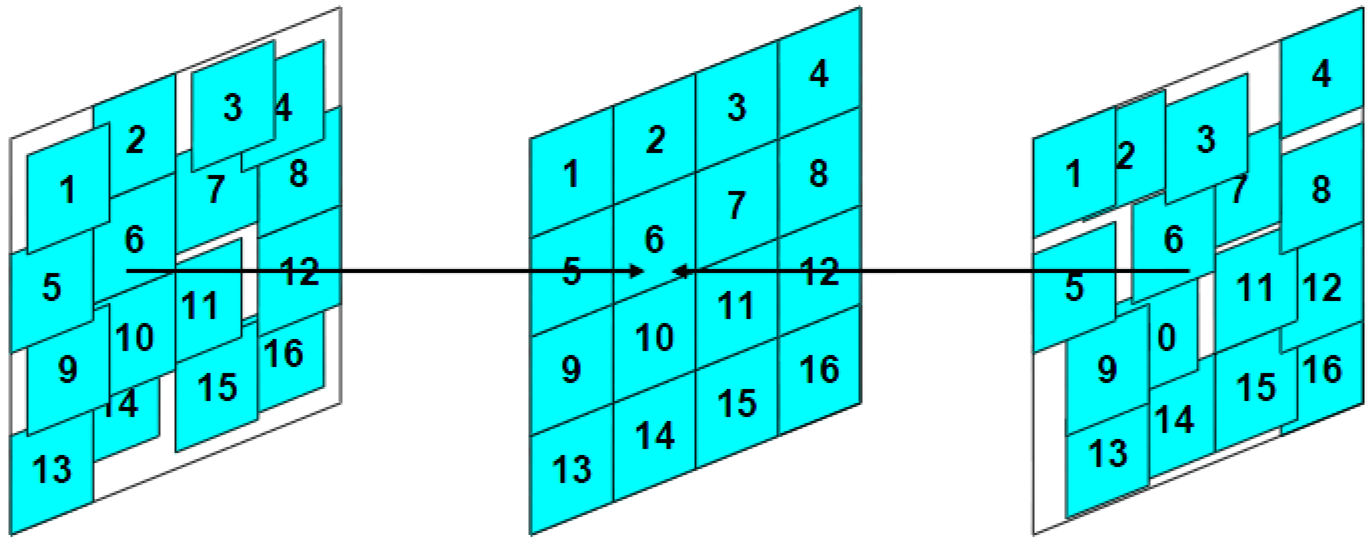
Slide from
Aljoscha Smolic

Block Matching Algorithm: Summary

- *Issues:*
  - Block size?
  - Search range?
  - Motion vector accuracy?
- 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 $\rightarrow$ Breaks down for more complex motion
  - Often produces blocking artifacts (OK for coding with Block DCT)

Slide from John G. Apostolopoulos

Bidirectional MC prediction



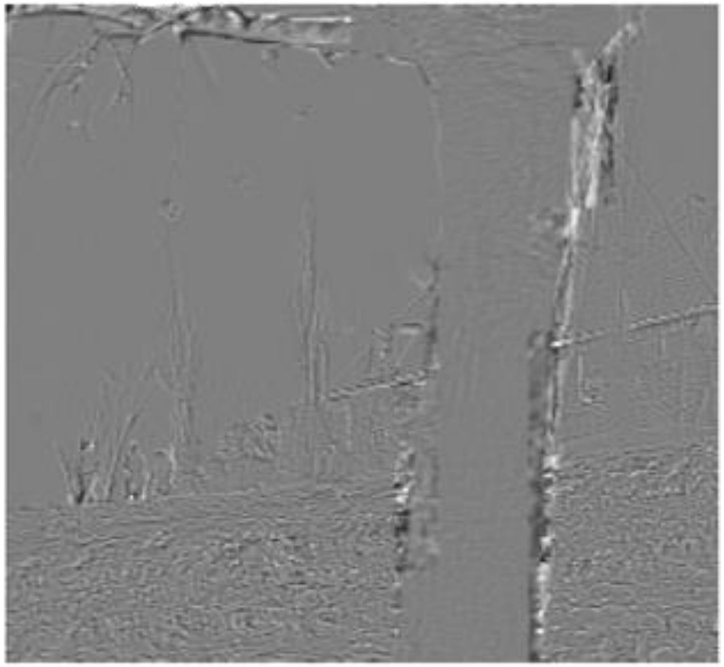
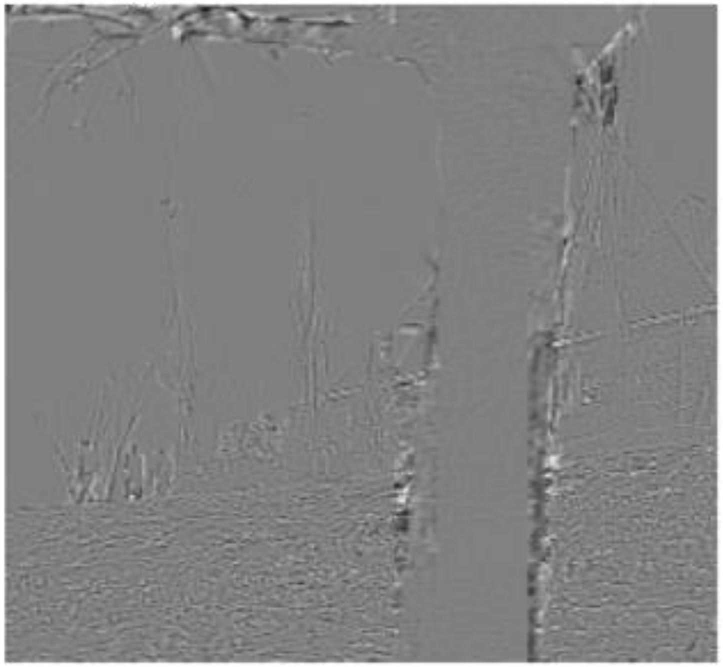
Previous Frame ***Current Frame*** ***Future Frame***

- *Bi-Directional MC-Prediction* is used to estimate a block in the current frame from a block in:
  1) Previous frame
  2) Future frame
  3) Average of a block from the previous frame and a block from the future frame
  4) Neither, i.e. code current block without prediction

Slide from John G. Apostolopoulos

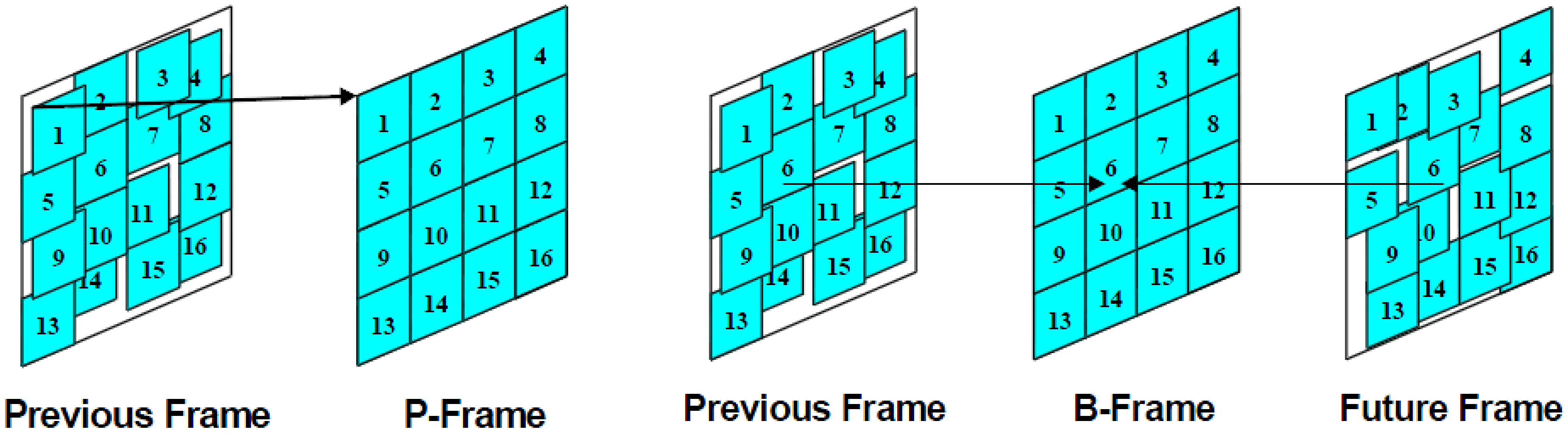
Example bidirectional prediction

- Prediction error with unidirectional (left) and bidirectional (right) prediction

MC-Prediction and Bi-Directional MC-Prediction (P-and B-frames)

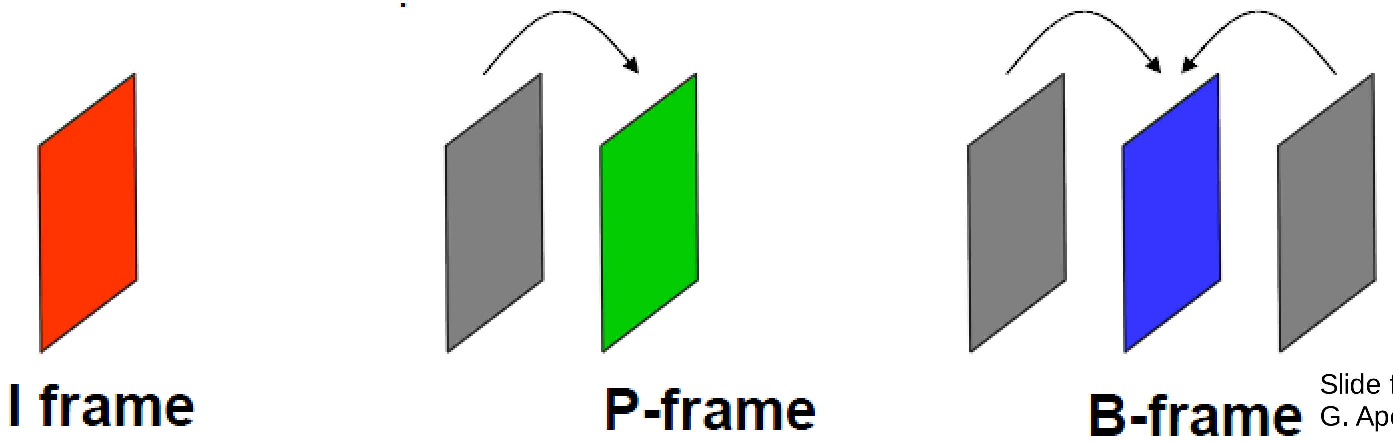
- Motion compensated prediction: Predict the current frame based on reference frame(s) while compensating for the motion
- Examples of block-based motion-compensated prediction (*P-frame*) and bi-directional prediction (*B-frame*):



Slide from John G. Apostolopoulos

Video compression

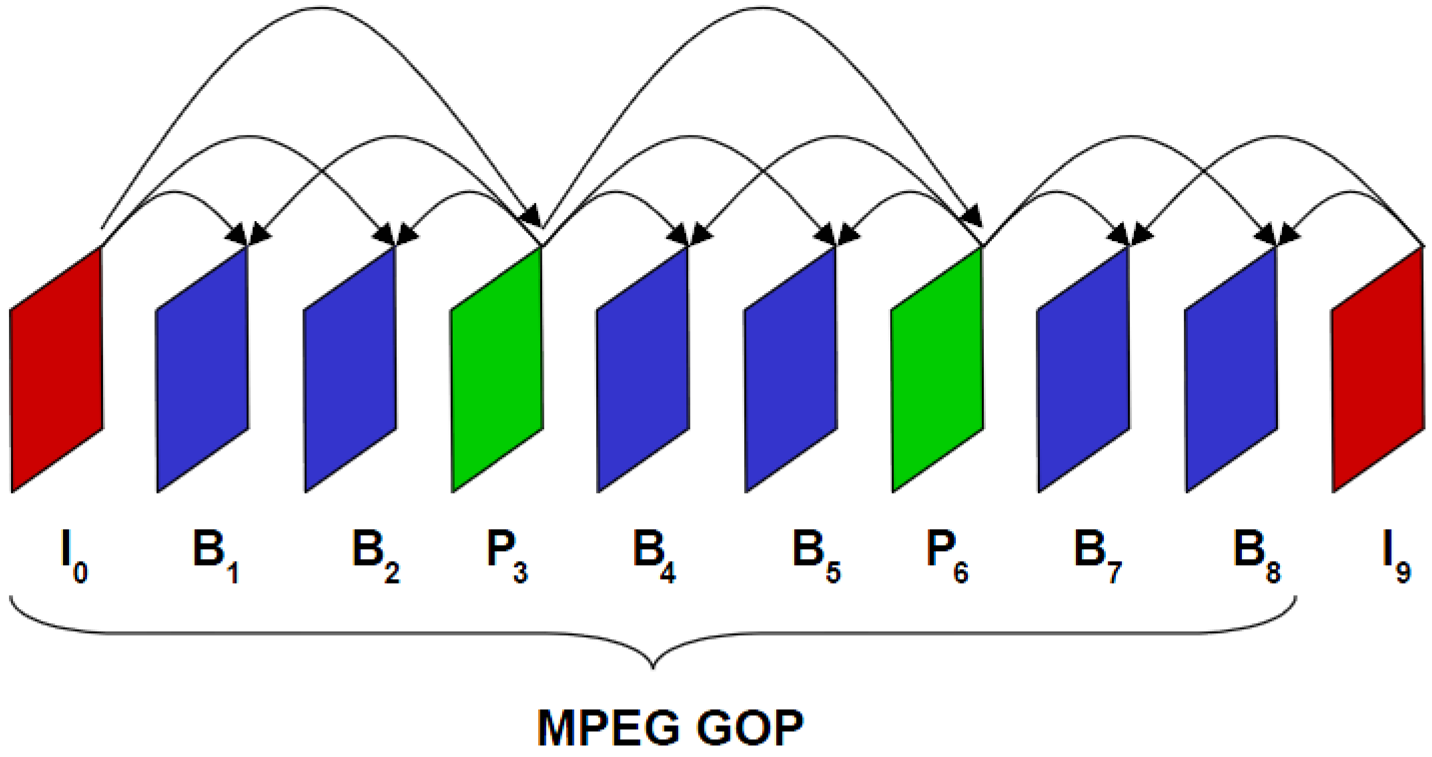
- Main addition over image compression:
  - Exploit the temporal redundancy
- Predict current frame based on previously coded frames
- Three types of coded frames:
  - ***I-frame:*** Intra-coded frame, coded independently of all other frames
  - ***P-frame:*** Predictively coded frame, coded based on previously coded frame
  - ***B-frame:*** Bi-directionally predicted frame, coded based on both previous and future coded frames



Slide from John G. Apostolopoulos

Example Use of I-,P-,B-frames: MPEG Group of Pictures (GOP)

- Arrows show prediction dependencies between frames



ETH

Slide from John G. Apostolopoulos

Group of Pictures (GOP)

- Starts with an I-frame
- Ends with frame right before next I-frame
- "Open" ends in B-frame, "Closed" in P-frame
  - (What is the difference?)
- MPEG Encoding a parameter, but 'typical':
  - I B B P B B P B B I
  - I B B P B B P B B P B B I
- *Why not have all P and B frames after initial I?*

Example Compress. Performance

Type	***Size***	***Compression***
I	18 KB	7:1
P	6 KB	20:1
B	2.5 KB	50:1
Avg	4.8 KB	27:1

Note, results are Variable Bit Rate, even if frame rate is constant

ETH

Summary of Temporal Processing

- Use MC-prediction (P and B frames) to reduce temporal redundancy
- MC-prediction usually performs well; In compression have a second chance to recover when it performs badly
- *MC-prediction yields:*
  - *Motion vectors*
  - *MC-prediction error or residual* $\rightarrow$ *Code error with conventional image coder*
- Sometimes MC-prediction may *perform badly*
  - Examples: Complex motion, new imagery (occlusions)
  - Approach:
    1. Identify frame or individual blocks where prediction fails
    2. Code *without prediction*

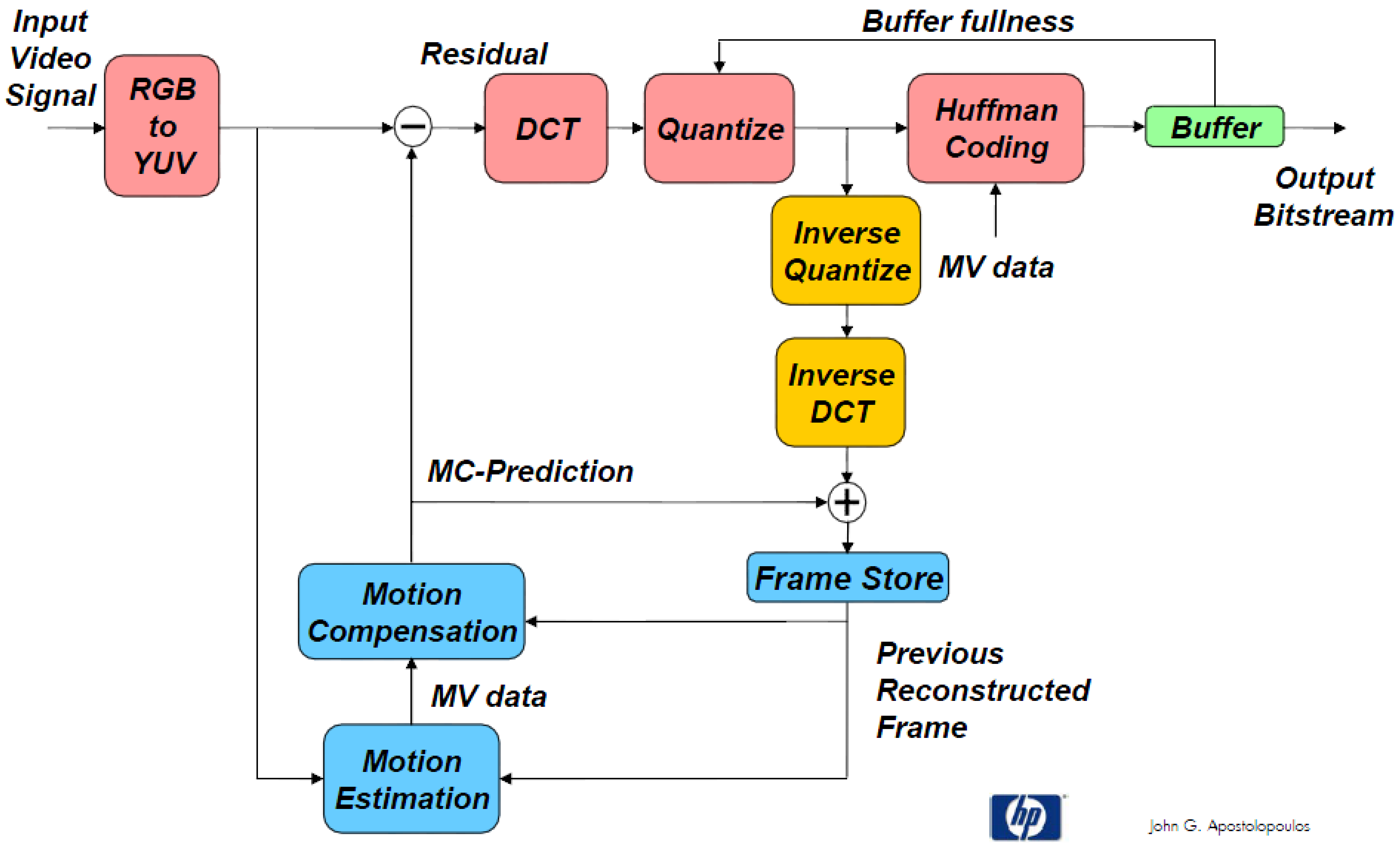
Slide from John G. Apostolopoulos

Basic Video Compression Architecture

- Exploiting the redundancies:
  - Temporal: MC-prediction (P and B frames)
  - Spatial: Block DCT
  - Color: Color space conversion
- Scalar quantization of DCT coefficients
- Zigzag scanning, runlength and Huffman coding of the nonzero quantized DCT coefficients

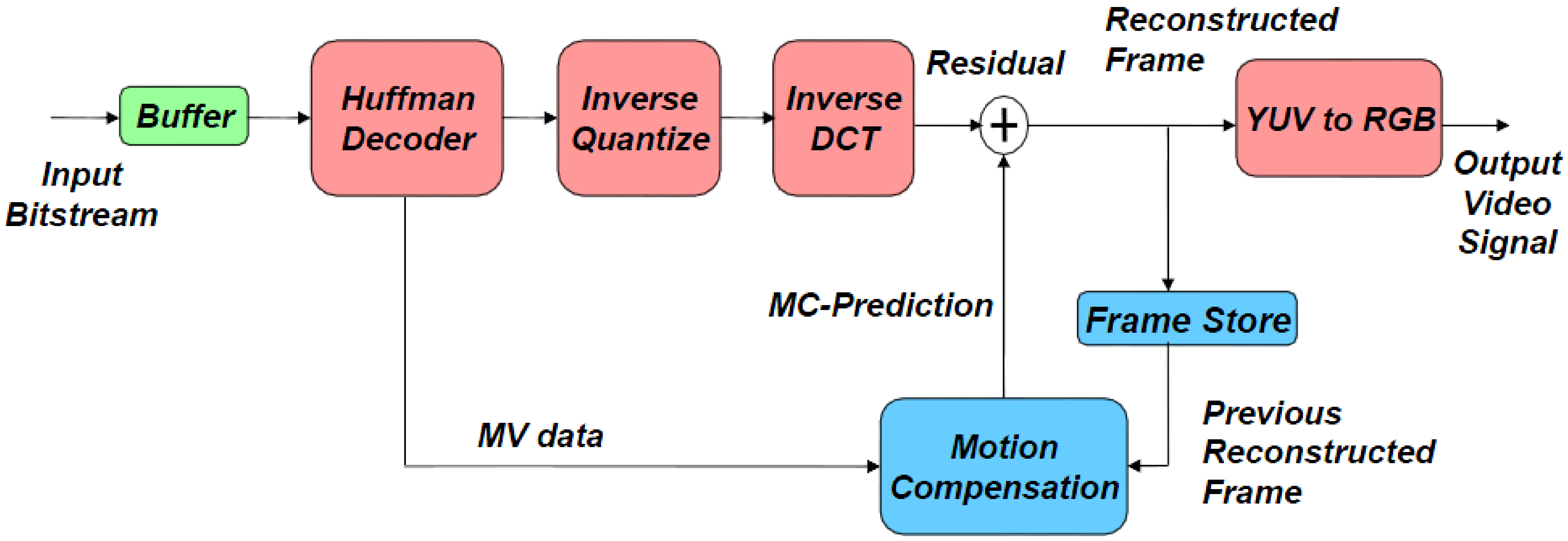
Slide from John G. Apostolopoulos

Example Video Encoder



John G. Apostolopoulos

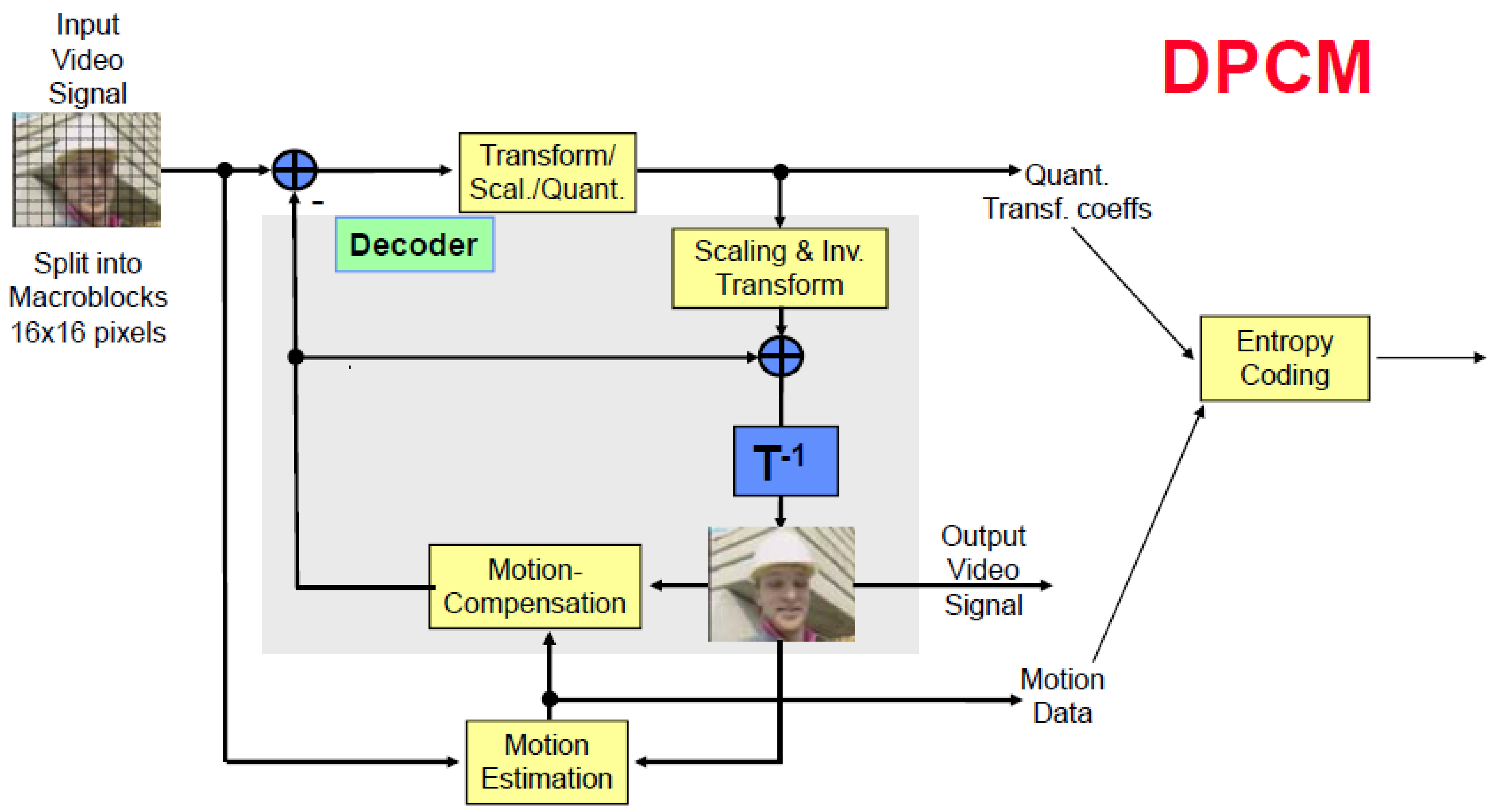
Example Video Decoder



ETH

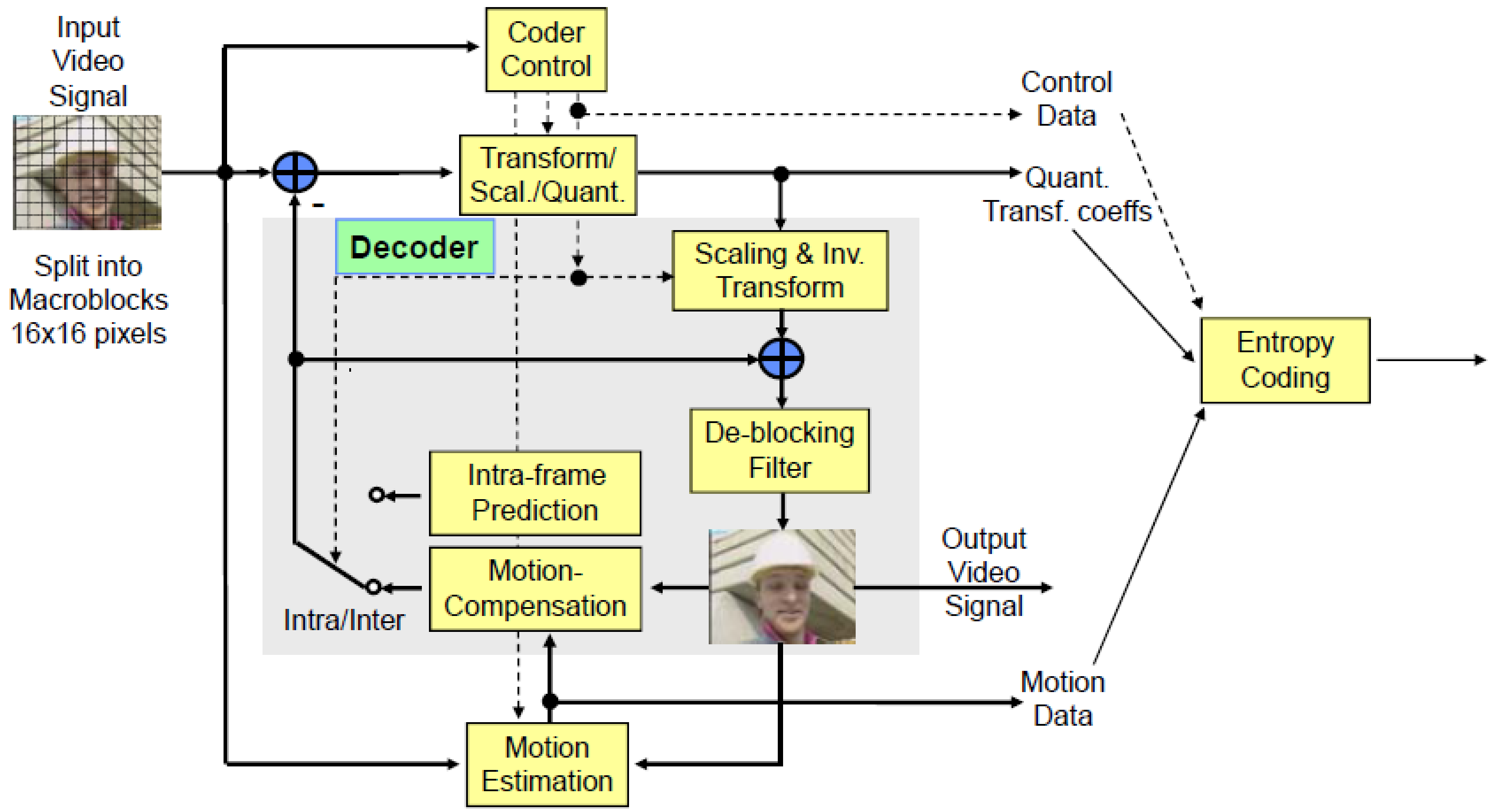
Slide from John G. Apostolopoulos

Hybrid Coding (MC+DCT)



ETH

MPEG-4 part 10 aka H.264



Current Video Compression Standards

- International Telecommunication Union (ITU-T)
  - H.261 : 1991
  - H.262 : 1994
  - H.263 : 1995 Version 1, versch. Erweiterungen "H.263+/++"
  - H.264 : Ende 2002

- International Standardization Organization (ISO) : Moving Picture Experts Group (MPEG)
  - MPEG-1 : ISO 11572 : 1992
  - MPEG-2 : ISO 13818 : 1994 (identisch mit H.262)
  - MPEG-4 : ISO 14496 : 1999 (teilweise identisch mit H.263 V1)
  - MPEG-4 Part 10 (AVC): 2003 (identisch mit H.264)

"Joint Video Team" (JVT)

Actual Standard H.264/AVC

- Still hybrid video codec (MC-DCT) as other standards before
- Significant over all improvement by improvement of all parts
  - Motion compensation
    - Variable block sizes (16x16, 16x8, 8x8, 8x4, 4x4)
    - ¼ pixel motion compensation
    - Multiple reference frames
    - Intra prediction
  - Integer 4x4 transform
  - In-loop deblockingfilter
  - Entropycoding
    - Arithmetic coding CABAC

Objective quality measure: PSNR

- Error for one pixel, difference between original and decoded value

$$e(v,h) = \tilde{x}(v,h) - x(v,h)$$

- Mean-Squared-Error, MSE e.g. over an image

$$e_{\text{mse}} = \sqrt{\frac{1}{N \cdot M} \sum_{v=1}^{N} \sum_{h=1}^{M} e^2(v,h)}$$

- Peak-Signal-to-Noise-Ratio

$$\text{PSNR} = \frac{[\text{maximum value of } x]^2}{e_{\text{mse}}^2}$$

$$\text{PSNR} = 10 \cdot \log_{10}\left(\frac{[2^K]^2}{e_{\text{mse}}^2}\right) \text{dB}$$

$$\text{PSNR} = 10 \cdot \log_{10}\left(\frac{255^2}{e_{\text{mse}}^2}\right) \text{dB}$$

Slides Aljoscha Smolic

Subjective evaluation

- MOS: Mean opinion score

5	Imperceptible
4	Perceptible, but not annoying
3	Slightly annoying
2	Annoying
1	Very annoying

- Subjective Tests with many participants and statistical evaluation
- Representative test material
- Careful selection of test conditions (display, light, disturbances, ...)

Rate-Distortion Curve

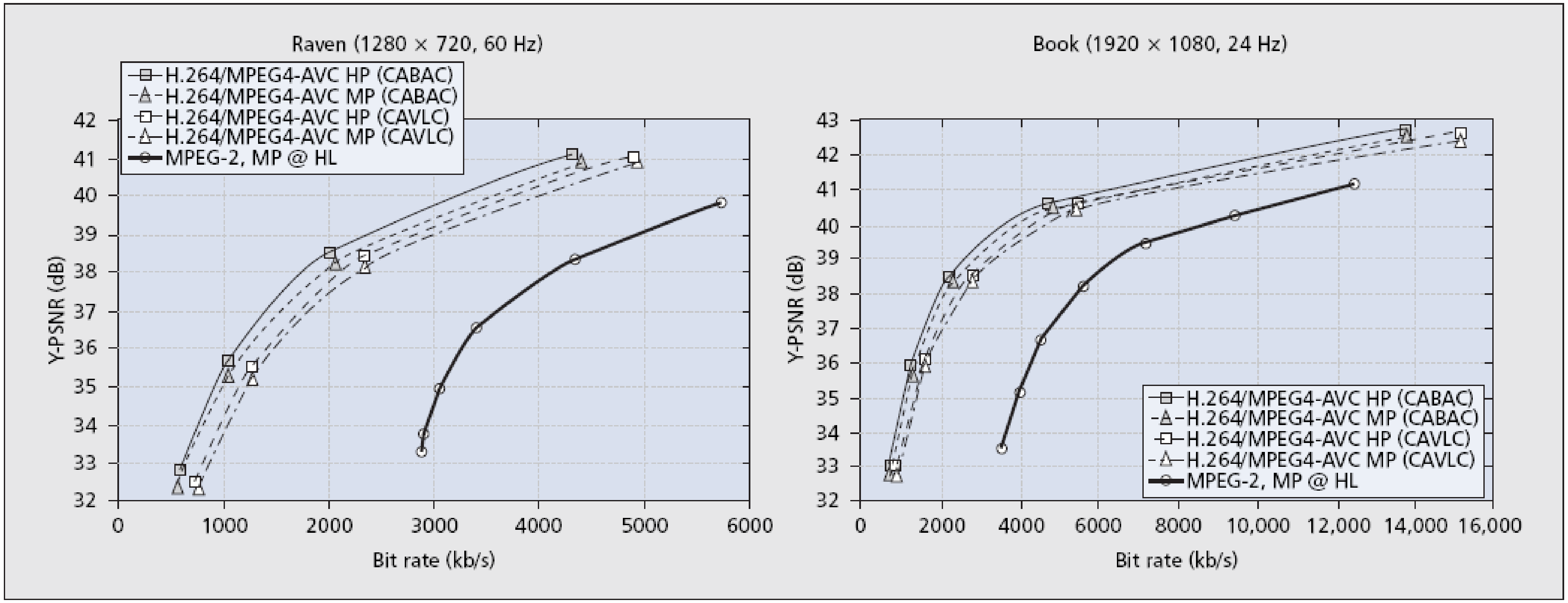


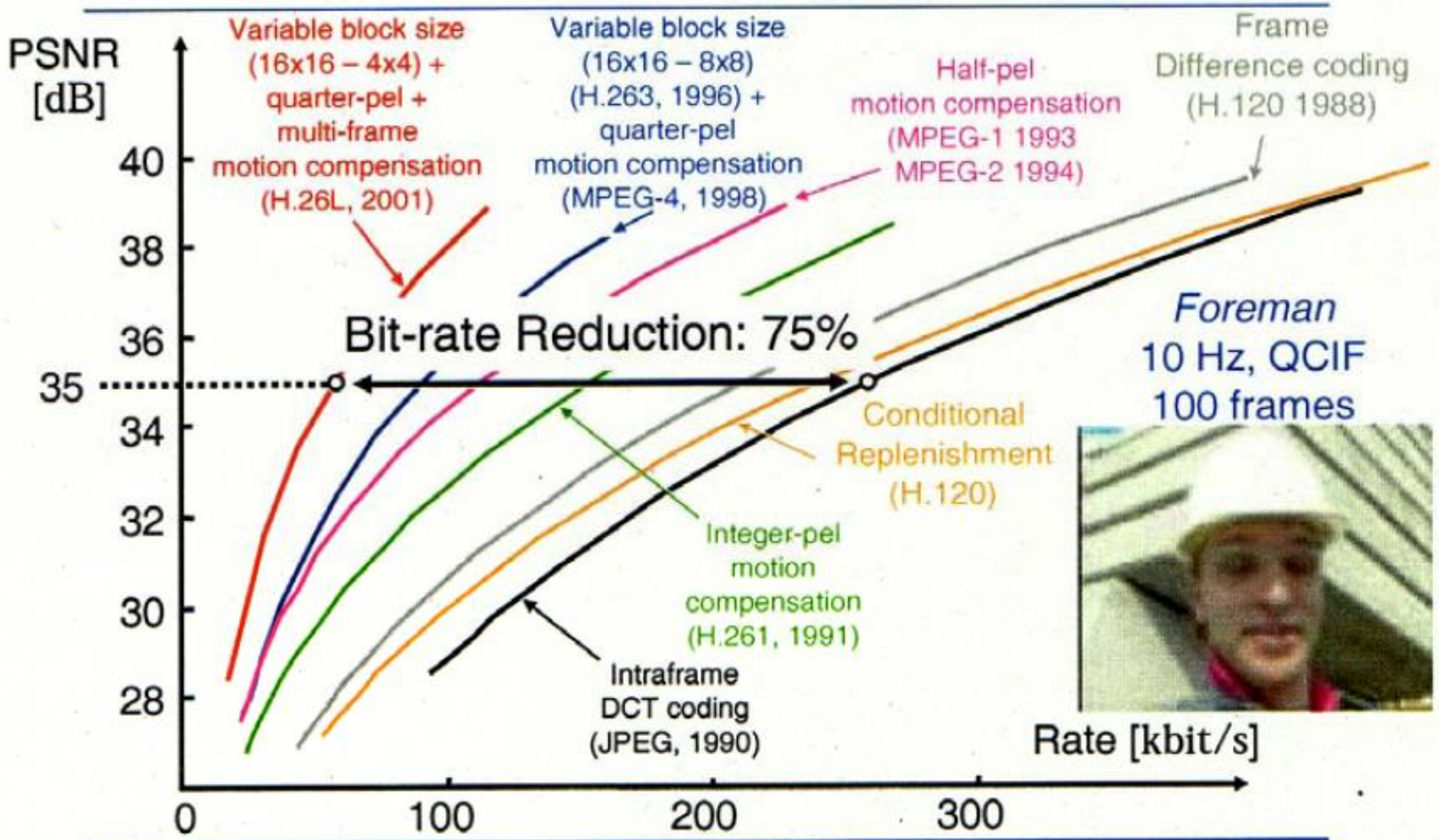
Figure 7. *Left: Objective performance for the "Raven" sequence (left) and "Book" sequence (right) comparing H.264/MPEG4-AVC HP, MP (both using CABAC and CAVLC), and MPEG2 MP@HL.*

$$PSNR = 10 \cdot \log_{10}\left(\frac{MAX_I^2}{MSE}\right) = 20 \cdot \log_{10}\left(\frac{MAX_I}{\sqrt{MSE}}\right)$$

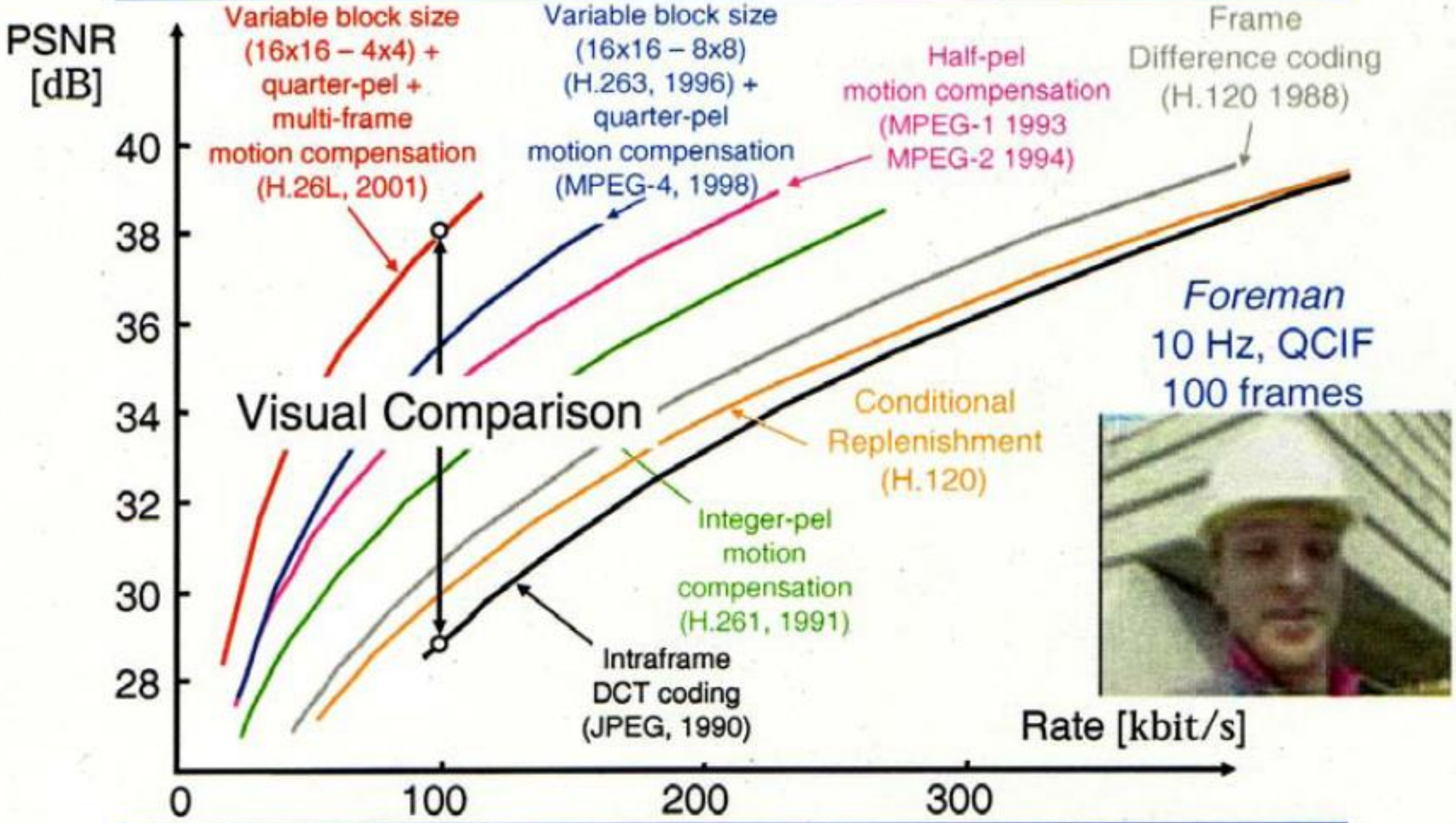
$$MSE = \frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}||I(i,j) - K(i,j)||^2$$

($PSNR$ = Peak Signal to Noise Ratio) (MSE = Mean Square Error)

Milestones in Video Coding

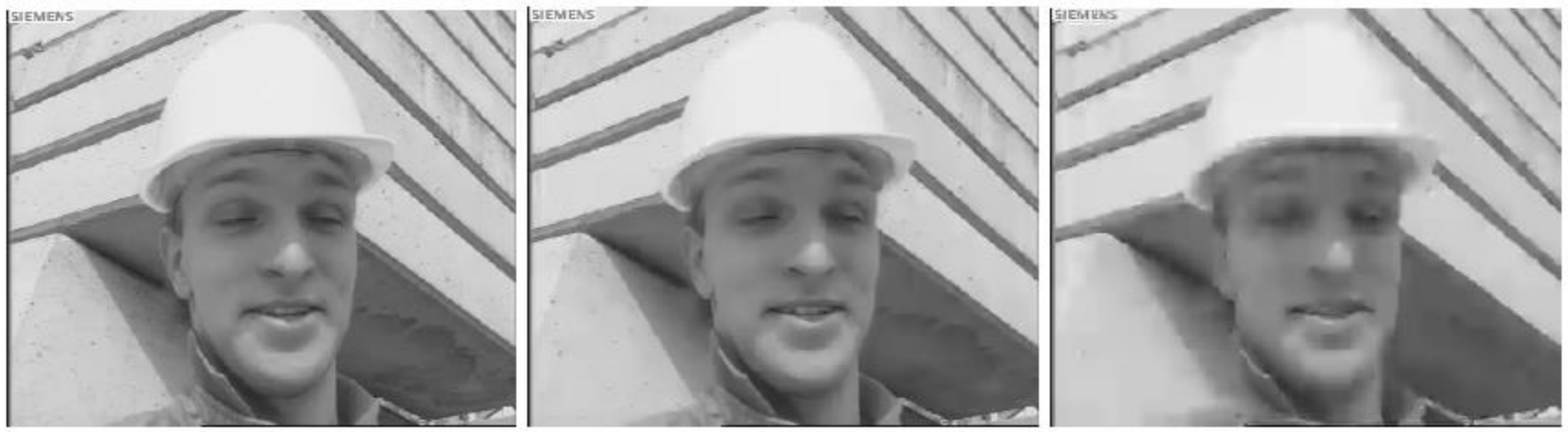


Milestones in Video Coding



ETH

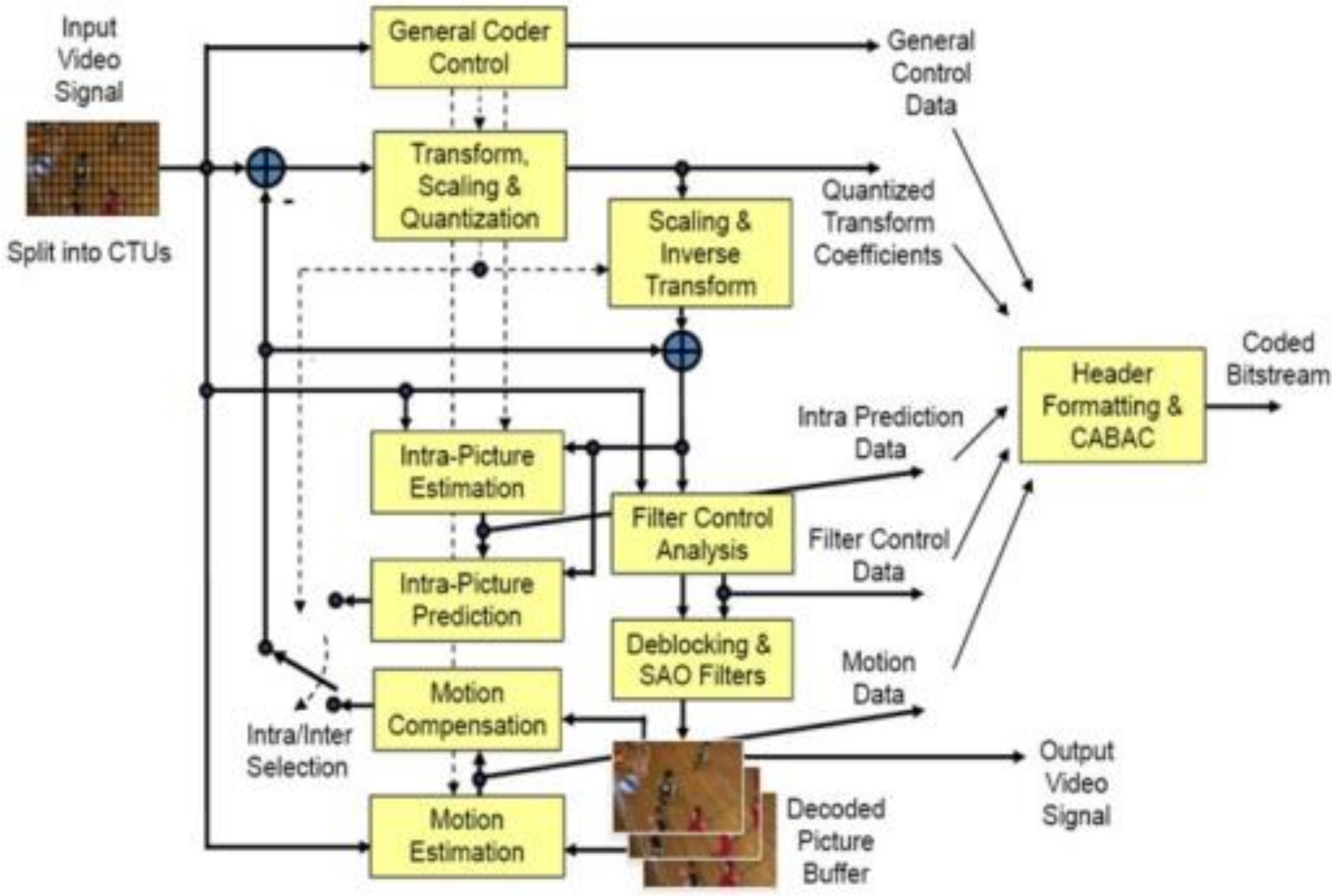
Examples H.264/AVC

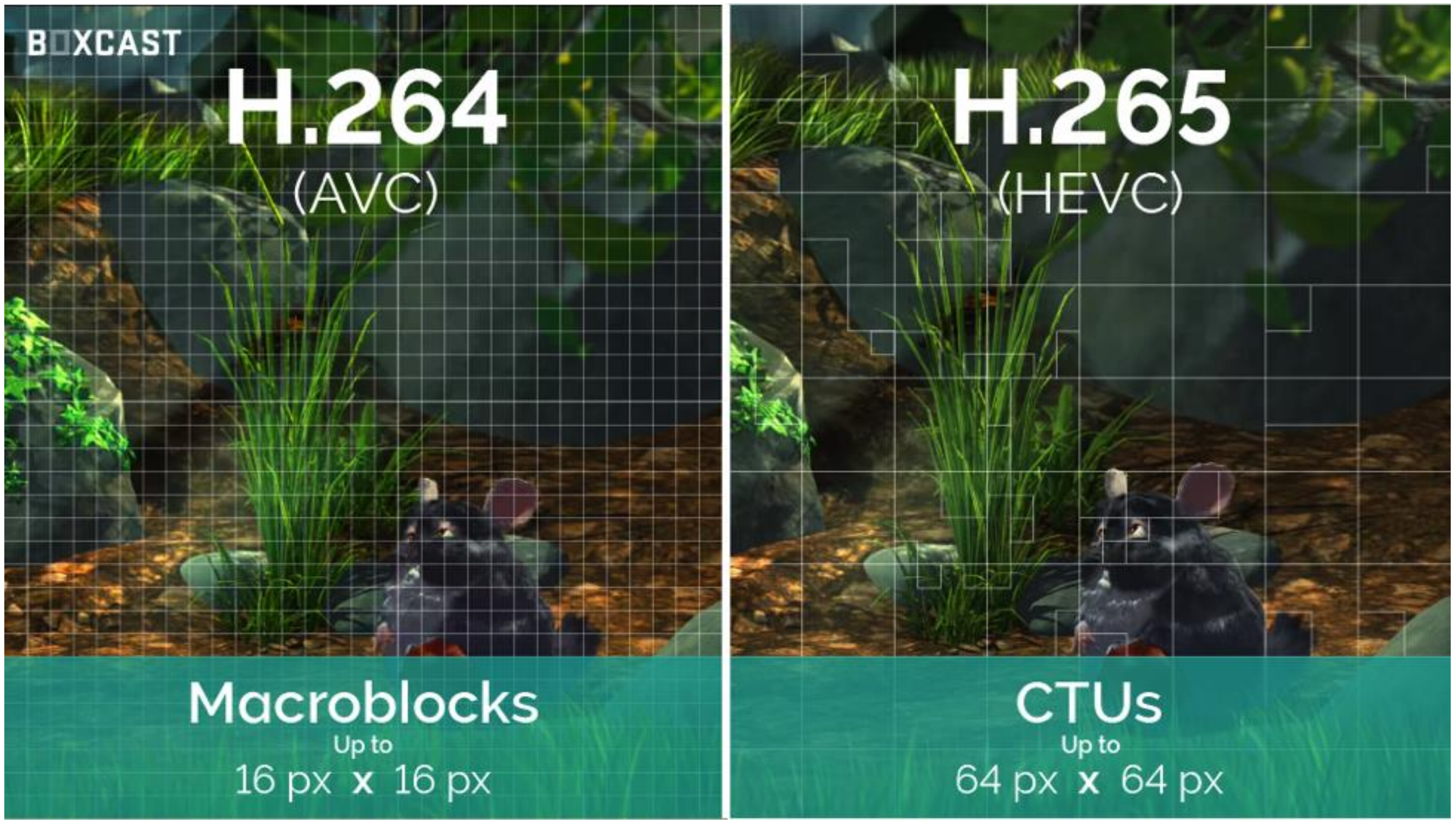


1:47 **1:165** **1:495**

Compression ratio

- http://en.wikipedia.org/wiki/High_Efficiency_Video_Coding





From https://www.thebroadcastbridge.com/content/entry/10029/h.264-versus-hevc-understanding-the-differences#:~:text=265%20is%20more%20advanced%20than,Unlike%20H .

Next week:

Selected topics:
CNN & Radon transform