Institut
Mines-Telecom

Video coding principles

Marco Cagnazzo

MN907 Compression

Plan

Plan

Video compression principles

- Spatial redundancy
  - Images are made up of homogeneous regions
- Time redundancy
  - Successive images in a video are similar each to the other
- A video compression algorithm must exploit both kind of redundancy

Video compression principles

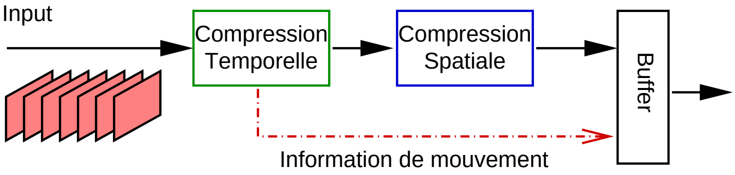
Spatial redundancy

Video compression principles

Time redundancy

Video compression principles

General scheme of a video coder



Plan

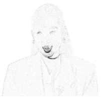
TELECOM ParisTech

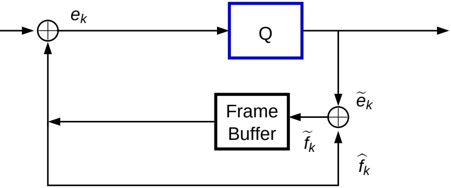
Prediction in video coding: DPCM

- Successive images are very similar
- Prediction: $\widehat{f}_{n,m,k} = \widetilde{f}_{n,m,k-1}$

Current image

Error





Previous image

Conditional replenishment

- Prediction:

$$\widehat{f}_{n,m,k} = \begin{cases} f_{n,m,k-1} & \text{if } |f_{n,m,k} - f_{n,m,k-1}| < \gamma \\ 0 & \text{otherwise} \end{cases}$$

Problem:

- *Side information*: one bit per pixel
- Using blocks of pixels the SI can be reduced

Conditional replenishment

Block similarity measure:

$$d(B_1, B_2) = \sum_{\mathbf{p}} |B_1(\mathbf{p}) - B_2(\mathbf{p})|^k$$

If $d(B_k^{(\mathbf{p})}, B_h^{(\mathbf{p})}) < \gamma$

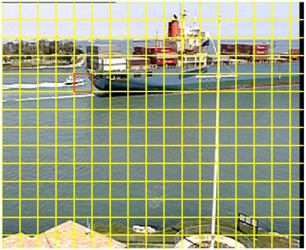
- *refine*: prediction error is transmitted
- *skip*: no bit is transmitted

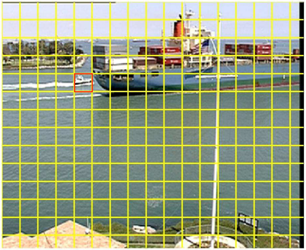
If vi $d(B_k^{(\mathbf{p})}, B_h^{(\mathbf{p})}) \geq \gamma$

- *new*: the block is transmitted without prediction

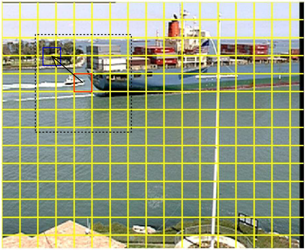
How to set γ and the block size?

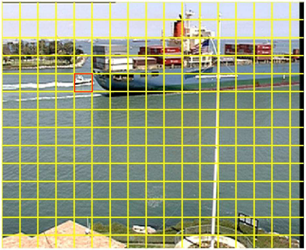
Motion estimation



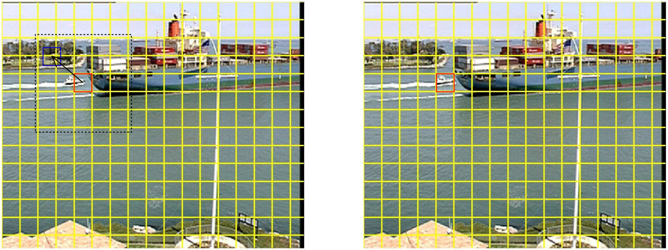


Motion estimation





Motion estimation



We compare $B_k^{(\mathbf{p})}$ and $B_h^{(\mathbf{p}+\mathbf{v})}$

Motion estimation

- ME Test:

$$d(\mathbf{v}) = d(B_k^{(\mathbf{p})}, B_h^{(\mathbf{p}+\mathbf{v})})$$

- Estimated vector:

$$\mathbf{v}^* = \arg\min_{\mathbf{v}} d(\mathbf{v})$$

- Transmitted info:

$$B_k^{(\mathbf{p})} - B_h^{(\mathbf{p}+\mathbf{v})}$$

- The decoder reconstructs the prediction of $B_k^{(\mathbf{p})}$ using the motion vectors and the reference image: this is the *motion compensation*.

Motion estimation

Cost function

Several choices are possible for $d(\cdot,\cdot)$:

- SAD (Sum of Absolute Differences)

$$d(B_1,B_2)=\sum_{n,m}|B_1(n,m)-B_2(n,m)|$$

- SSD (Sum of Squared Differences)

$$d(B_1,B_2)=\sum_{n,m}[B_1(n,m)-B_2(n,m)]^2$$

- ZN-SSD (Zero-mean Normalized SSD)

$$d(B_1,B_2)=\frac{\sum_{n,m}\left[\overline{B}_1(n,m)-\overline{B}_2(n,m)\right]^2}{\sum_{n,m}\overline{B}_1^2(n,m)\sum_{n,m}\overline{B}_2^2(n,m)}$$

Prediction in video coding

Motion estimation regularization

- Vectors in homogeneous areas are chaotic
- A regularization term is added

$$J(\mathbf{v}) = d(\mathbf{v}) + \lambda r(\mathbf{v})$$

- Estimated vector:

$$\mathbf{v}^* = \arg\min_{\mathbf{v}} J(\mathbf{v})$$

- λ defines the trade-off between fidelity and regularity
- $r(\mathbf{v})$: coding cost or smooth constraint

Prediction in video coding

Examples

Reference image

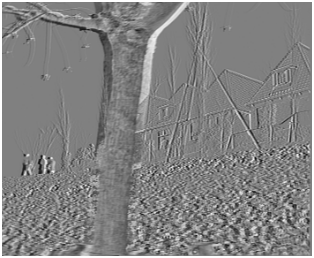
Current image

Prediction in video coding

Examples

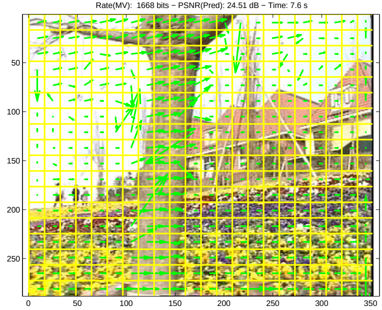
Current image

Difference image



Prediction in video coding

Estimated vectors

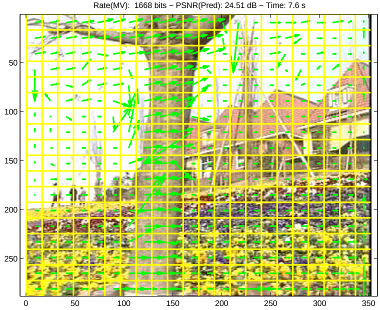


Non-regularized MVF

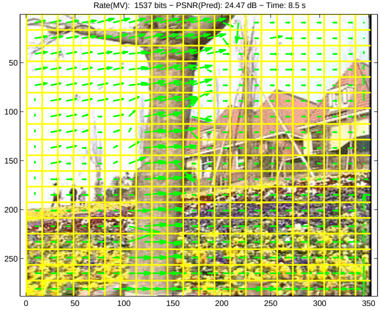
Regularized MVF

Prediction in video coding

Estimated vectors



Non-regularized MVF



Regularized MVF

Prediction in video coding

Estimated vectors

Motion−compesated image

Regularized MVF, motion-compensated image

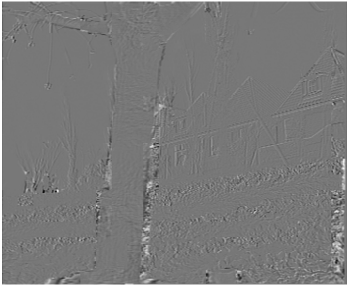
Regularized MVF, compensation error

Prediction in video coding

Estimated vectors

Motion−compesated image

MC error

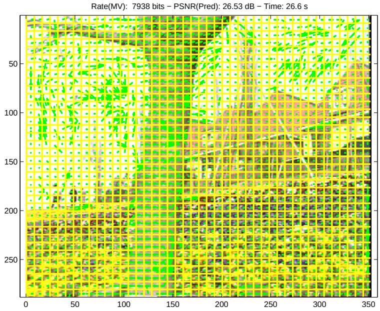


Regularized MVF, motion-compensated image

Regularized MVF, compensation error

Prediction in video coding

Estimated vectors

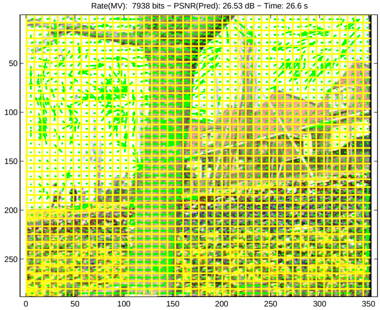


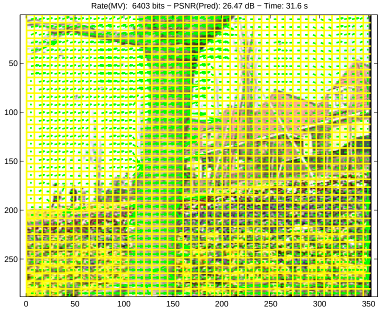
Non-regularized MVF

Regularized MVF

Prediction in video coding

Estimated vectors





Non-regularized MVF

Regularized MVF

Prediction in video coding

Estimated vectors

Motion−compesated image

Regularized MVF, motion-compensated image

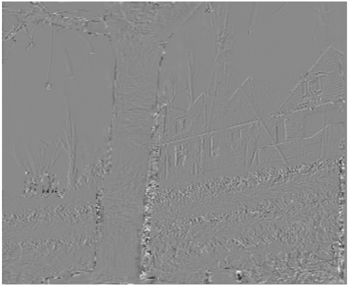
Regularized MVF, compensation error

Prediction in video coding

Estimated vectors

Motion−compesated image

MC error

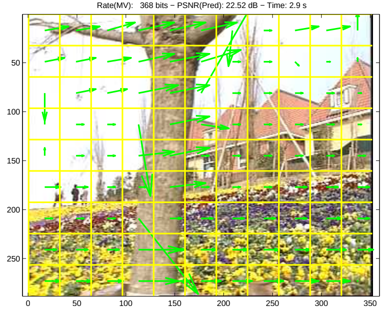


Regularized MVF, motion-compensated image

Regularized MVF, compensation error

Prediction in video coding

Estimated vectors

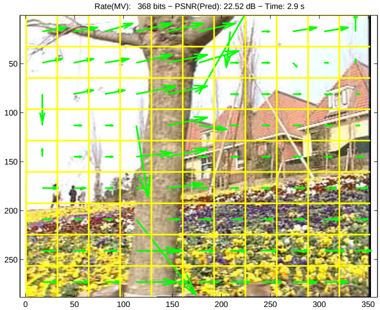


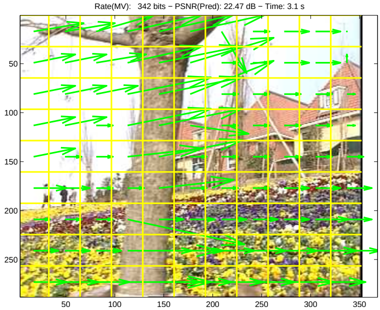
Non-regularized MVF

Regularized MVF

Prediction in video coding

Estimated vectors





Non-regularized MVF

Regularized MVF

Prediction in video coding

Estimated vectors

Motion−compesated image

Regularized MVF, motion-compensated image

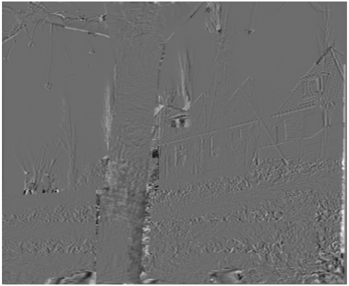
Regularized MVF, compensation error

Prediction in video coding

Estimated vectors

Motion−compesated image

MC error



Regularized MVF, motion-compensated image

Regularized MVF, compensation error

Prediction in video coding

Examples

Reference image

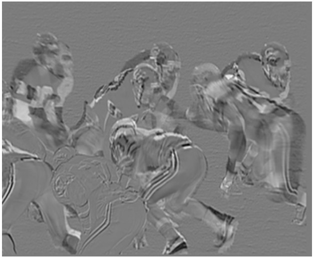
Current image

Prediction in video coding

Examples

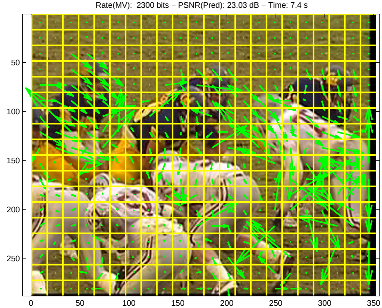
Current image

Difference image



Prediction in video coding

Estimated vectors

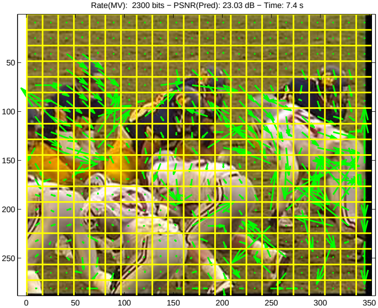


Non-regularized MVF

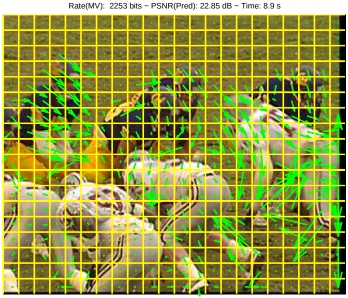
Regularized MVF

Prediction in video coding

Estimated vectors



Non-regularized MVF



Regularized MVF

Prediction in video coding

Estimated vectors

Motion−compesated image

Regularized MVF, motion-compensated image

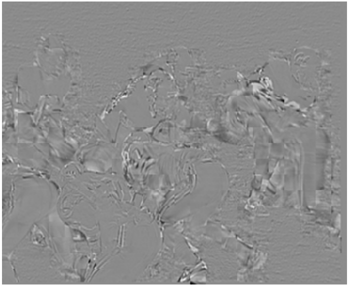
Regularized MVF, compensation error

Prediction in video coding

Estimated vectors

Motion−compesated image

MC error



Regularized MVF, motion-compensated image

Regularized MVF, compensation error

Motion estimation

Search strategy: complexity/effectiveness trade-off

Let n be the search window side

- *Full search* method: All the n^2 vectors are tested
- *Cross search* method: First horizontal vectors are tested; then the vertical component is found; total, $2n$ vectors
- *Log search* method: Nine positions $\{0, \pm(2^m - 1)\}^2$ are tested; the search window is then centered on the best position and the search step is halved to $2^{m-1} - 1$ pixels. The number of tests is $\approx 8 \log_2 n$
- *Diamond search* method: Eight directions are tested, but the step is reduced only when the center position has been chosen

Motion estimation

Summary

- Very effective for video temporal prediction
- Used in virtually all video encoders
- Trade-off: precision - coding cost - complexity
- Design choices:
  - Cost function (SAD, SSD, regularization, . . .)
  - Motion support (shape and size of blocks, search area, . . .)
  - Search strategy (Full-search, Log, Diamond, . . .)

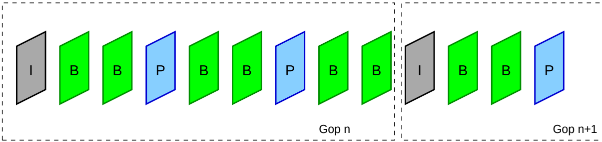
Frame types

- Frames I (Intra coded)
- Frames P (Predictive)
- Frames B (Bi-directional)

I and P Frames: Anchor Frames (AF)

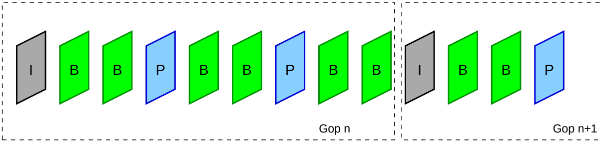
Group of Pictures

- Frames organized into GOP (Group of Pictures)
- First image : Intra
- Structure :
  - interval between I frames
  - interval between AFs



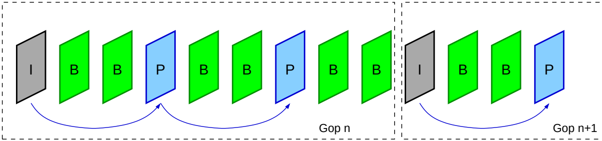
I Frames

- Encoded independently from others
- JPEG-like coding
- Low complexity, low coding rate
- Used for:
  - Fast forwards
  - Random access
  - Error robustness



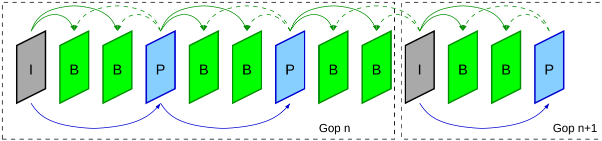
P Frames

- Prediction from previous AF
- High Complexity (ME)
- High compression ratio



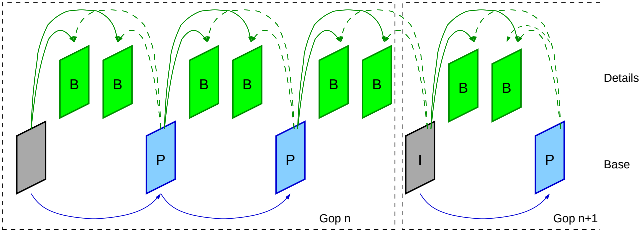
B Frames

- Predicted from both previous and next AF
- Very high complexity (double ME)
- Very high compression ratio



Frame coding order

I $\rightarrow$ AF $\rightarrow$ Frames B $\rightarrow$ AF $\rightarrow$ Frames B . . .
Delay?



The hybrid video encoder

- Macroblock-based coding
- Coding modes

  Intra: No temporal prediction, transform-based coding

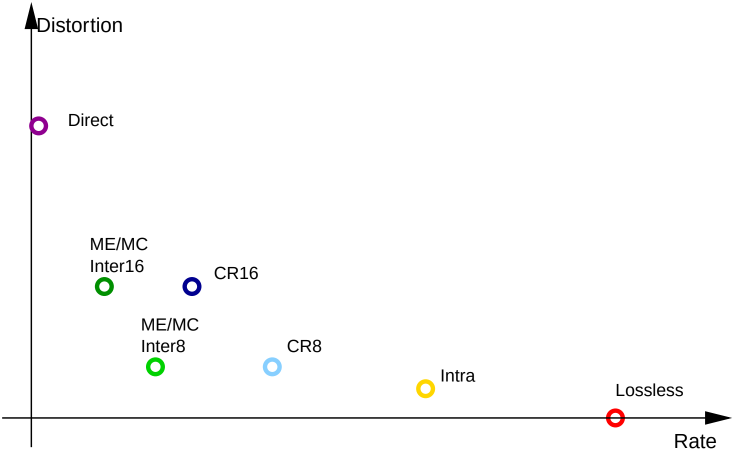
  Inter: ME/MC-based temporal prediction, transform coding

  Direct: Motion vector inferred from neighbors; no residual coding

  Lossless

The hybrid video encoder

Coding performance examples



The hybrid video encoder

Coding mode selection

- Goal: minimizing D for a given R:

$$D = \sum_{k=1}^{K} D_k(i_k, Q) \qquad R = \sum_{k=1}^{K} R_k(i_k, Q)$$

- The quantization step is given Q
- The set of modes $\mathbf{i} = \{i_k\}_{k=1}^{K}$ must be chosen such that we minimize:

$$J(\mathbf{i}, Q, \lambda) = \sum_{k=1}^{K} D_k(i_k, Q) + \lambda \sum_{k=1}^{K} R_k(i_k, Q)$$

The hybrid video encoder

Coding mode selection

- Joint minimization over **i** is way too complex
- A sub-optimal minimization is preferable
- For a MB k, we choose the mode such that we minimize:

$$J_k(i_k, Q, \lambda) = D_k(i_k, Q) + \lambda R_k(i_k, Q)$$

- That is, we minimize *separately* each term of the sum giving J
- The selected mode depends on Q and λ

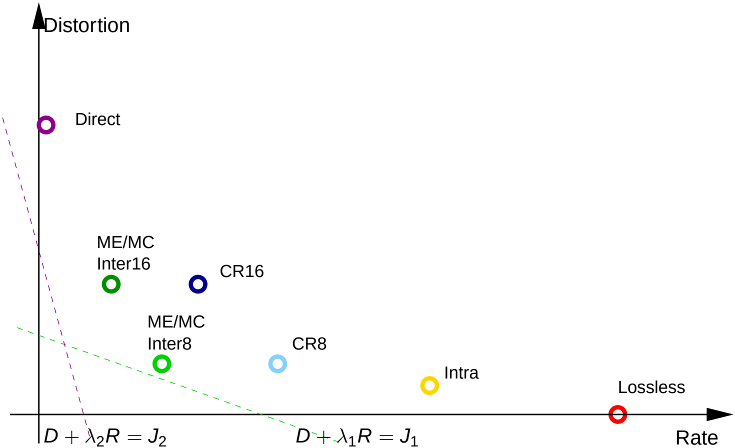
The hybrid video encoder

Coding mode selection

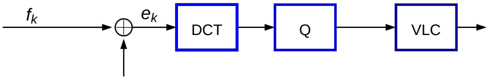
- quantization step Q is considered as an *input*
- For each Q (rate) there exists an optimal λ value, which is determined empirically
  - MPEG-2 : $\lambda = aQ^2 + b$
  - H.264 : $\lambda = c2^{dQ+e}$
- With this λ, minimizising J_k amounts to find a line in the RD plane

The hybrid video encoder

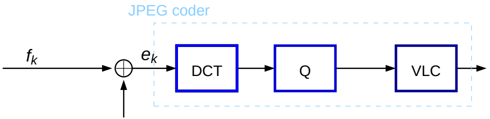
Example of performance



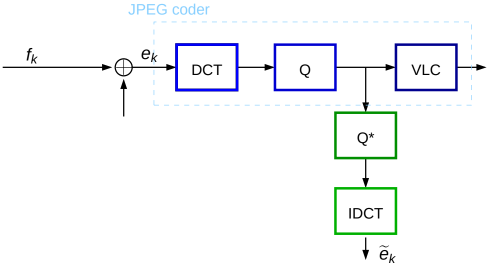
The hybrid video encoder



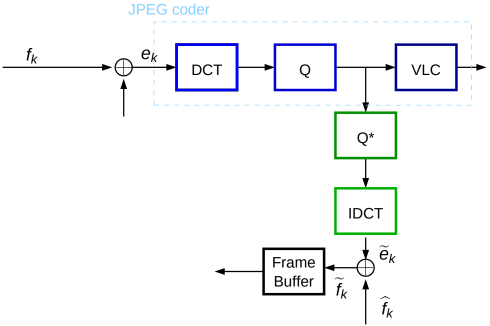
The hybrid video encoder



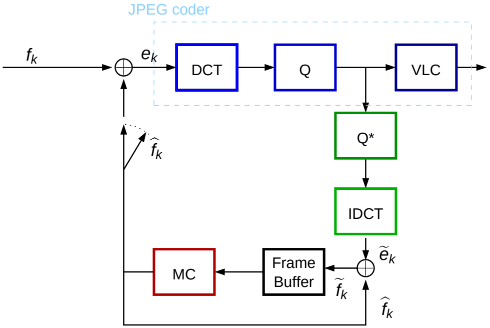
The hybrid video encoder



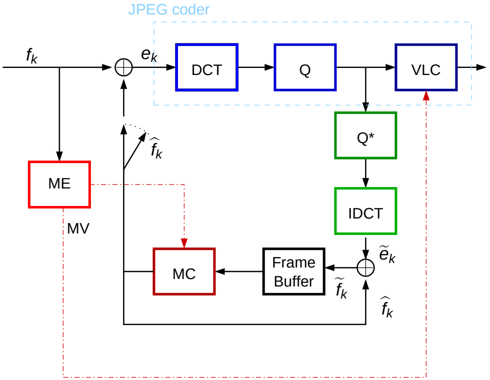
The hybrid video encoder



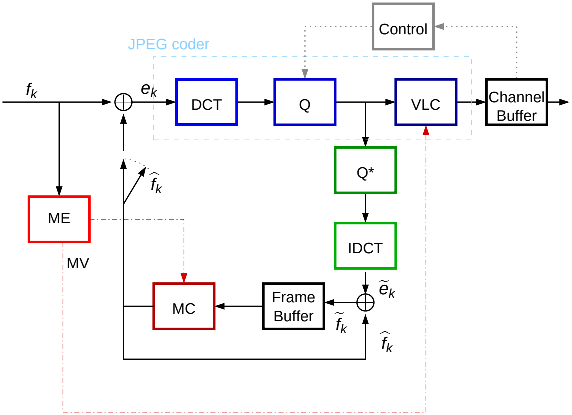
The hybrid video encoder



The hybrid video encoder

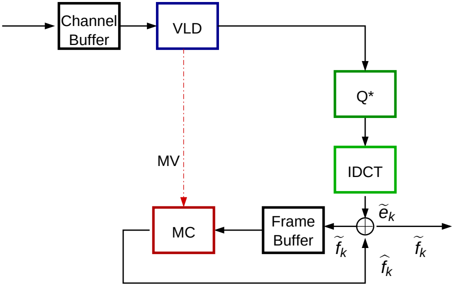


The hybrid video encoder



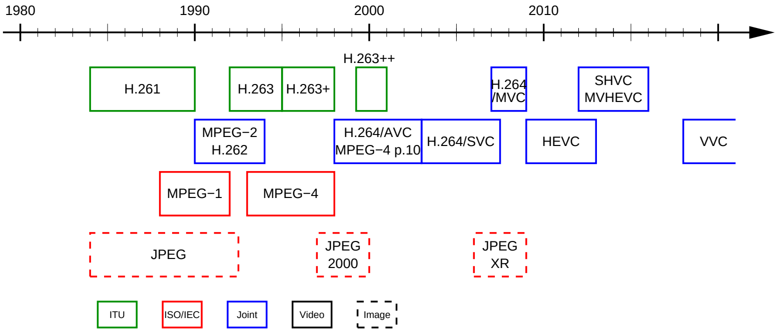
The hybrid decoder

Asymmetrical scheme!

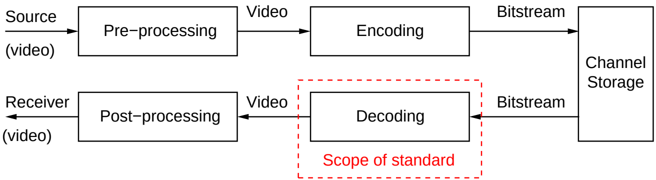


Plan

Video standards



Standard *Scope*



- The standard only defines the bitstream syntax and the decoder behavior
- Goal: interoperability, competition

MPEG-1 standard

- Developed in 1988-1992
- Parts
  1. Systems
  2. Video
  3. Audio
  4. Conformance test
  5. Software simulation

MPEG-1 standard

Part 2 (Video)

- Hybrid coder with ME/MC
- Input: max 720 $\times$ 576 pixel @ 30 fps
- Rate $\leq$ 1.86 Mbps (VHS quality)
- Asymmetric applications: VoD, video CD, videogames

Features

- Image types
- Sub-pixel ME/MC

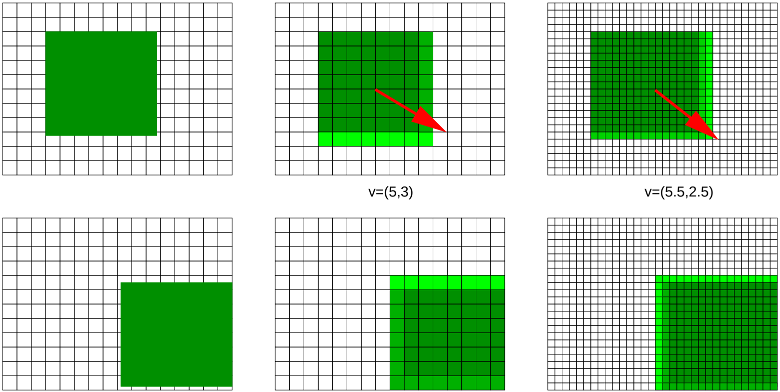
Standard MPEG-1

Sub-pixel ME/MC

- Physical motion does not necessarily correspond to pixel grid
- Interpolation to improve precision
- Further complexity increase
- Rate-distortion improvement

Standard MPEG-1

Sub-pixel ME/MC



MPEG-2 standard

- Developed in1990-1994
- Parts
  1. Systems
  2. Video
  3. Audio
  4. Conformance test
  5. Software simulation

MPEG-2 standard

- Hybrid coder
- Rate $\leq$ 15 Mbps (HDTV)
- Profiles and levels
- Interlaced video support
- Scalability support

MPEG-2 standard

Profiles and levels

Level	width [pixel]	height [pixel]	frame rate [frame/s]	bit rate [Mbps]
Low	352	288	30	4
Main	720	576	30	15
High-1440	1440	1152	60	60
High	1920	1152	60	80

Profile	Feature
Simple	No scalability; interlaced video; no B-frames
Main	Simple + B-frames
SNR scalable	Main + Two or three quality scalability levels
Spatial scalable	SNR + Two or three resolution scalability levels
High	Space + Oversampled chroma

MPEG-2 standard

Profiles and levels

Level	Profile				
	Simple	Main	SNR scalable	Spatial scalable	High
Low		●	●		
Main	●	●	●		●
High-1440		●		●	●
High		●			●

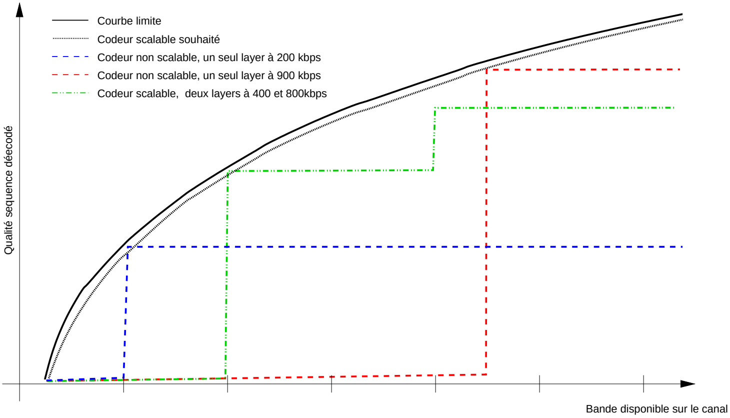
MPEG-2 standard

Scalability

Encode once, decode many!

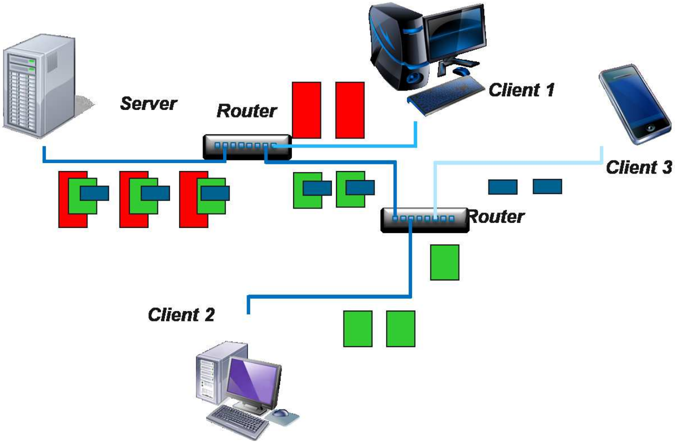
- Bitstream made up of:
  - One *base* layer
  - One or more *enhancement* layers
- The base layer can be decoded alone
- The enhancement improves quality or resolution...
- ... but cannot be decoded alone
- A client may demand the base layer only or base+enhancement

Scalability



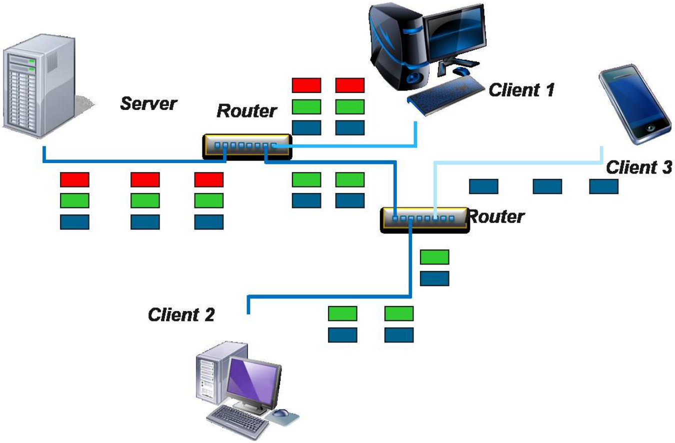
Scalability: example

Video distribution without scalability



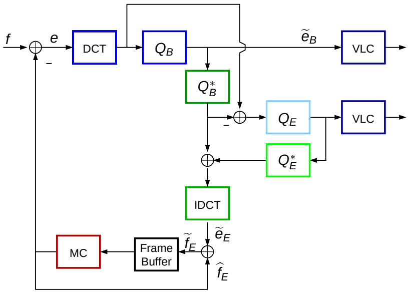
Scalability: example

Video distribution with scalability



MPEG-2 standard

SNR scalability: encoder

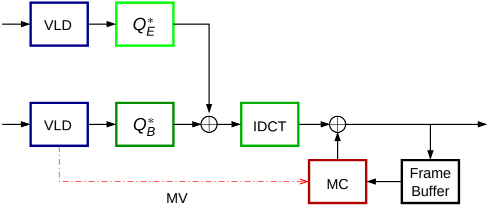


- DCT coefficients refinement
- *Drift* of the base layer
- Good quality of the enhanced layer

TELECOM ParisTech

MPEG-2 standard

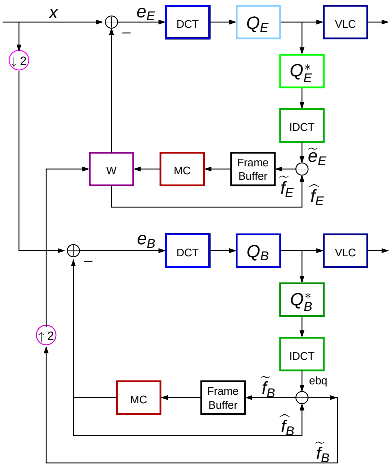
SNR scalability: decoder



- No drift control
- The same MVF is used at both layers

MPEG-2 standard

Resolution scalability: encoder



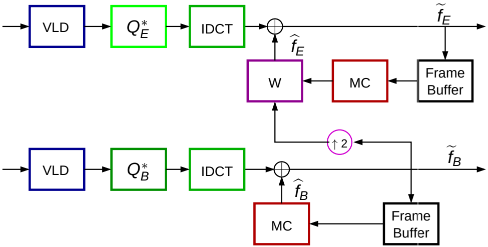
MPEG-2 standard

Resolution scalability: encoder

- Double loop: no drift
- Input video is filtered and subsampled
- Enhanced level prediction is a weighted sum of:
  - The interpolated base-layer image
  - ME/MC prediction
- The weight is changed per-MB, and its value is encoded in the bitstream

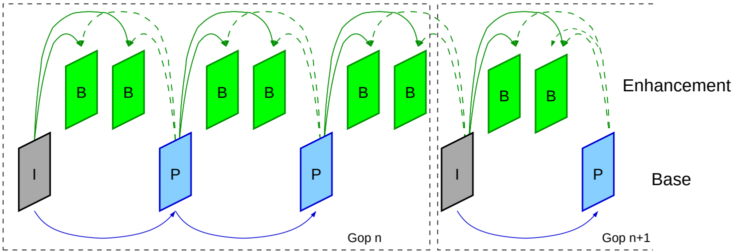
MPEG-2 standard

Resolution scalability: decoder



MPEG-2 standard

Time scalability



MPEG-2 standard

Hybrid scalability, 1/2

- SNR + spatial
  1. SDTV/CIF, low quality
  2. HDTV/SDTV, low quality
  3. HDTV/SDTV, high quality

MPEG-2 standard

Hybrid scalability, 2/2

- spatial + temporal
    1. SDTV interlaced
    2. HDTV interlaced
    3. HDTV progressive
- SNR + temporal
    1. HDTV interlaced, low quality
    2. HDTV interlaced, high quality
    3. HDTV progressive, high quality

Le standard MPEG-4

- Developed in 1993-1998
- Parts
  - 5 main parts (as MPEG-1 et 2)
  - 18 additional parts
  - E.g. MPEG4/part 10 is H.264/AVC

Standard MPEG-4

Features

- Hybrid coder
- Interactivity
  - Bitstream manipulation without transcoding
  - Hybrid coding of natural and synthetic data
  - Improved random access
- Compression
  - Improved RD performance
- Universal access
  - Error robustness
  - Object-based scalability

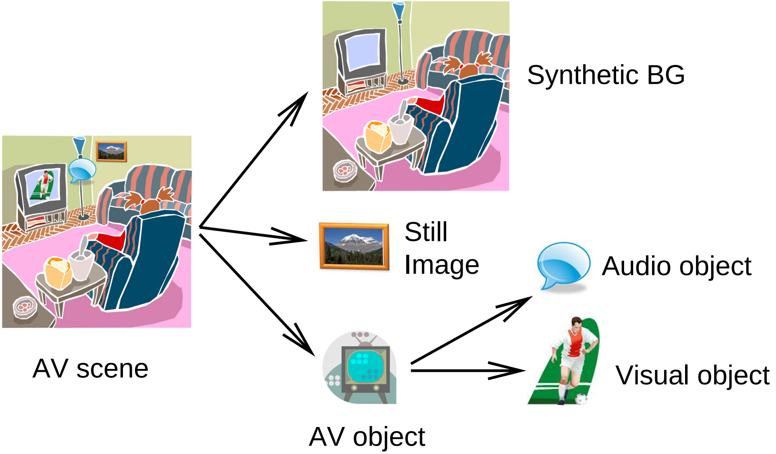
Standard MPEG-4

Object-based representation

- Audiovisual object (AVO)
  - Several AVOs encoded in different bitstreams
  - Audio (mono, stereo, synthetic, . . .) and/or video part (natural, synthetic, . . .)
- Several AVOs make a *AV scene*
- MPEG-4 defines syntax scene description

Standard MPEG-4

Audiovisual scene



Standard MPEG-4

Visual coding

- Video object coding
- Mesh object coding
- Model-based coding
- Still texture coding

Standard MPEG-4

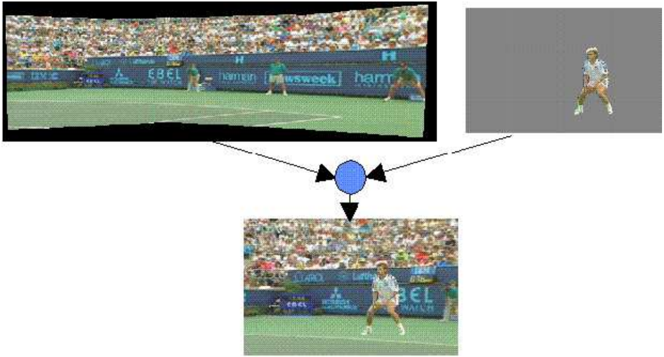
Video object coding

A *video object* (VO) is the succession of *video object planes* (VOP), made up of:

- Motion
- Texture
- Contours (Shape)

Standard MPEG-4

Sprite coding



Standard MPEG-4

Scalability

- Frame-rate and resolution: as MPEG-2
- Quality: *fine grain scalability* (bit-plane coding)
- Object scalability: the scene can be composed with a subset of available objects