Πως περιγράφεται το bitstream

- H.264, Κεφάλαιο 7.1, σελ. 37

Η γλώσσα Flavor

- Formal Language for Audio-Visual Object Representation

What does it do?

Automatic mapping of media information represented in arbitrary ways to program data structures

Technical Approach

- Principle of separation
  - Same core decoding algorithm can be used with different syntax
- Declarative, not procedural
  - Power of abstraction
  - Efficiency (opportunities for optimization)
- Extension of typing system of C++ and Java
  - Seamless integration with users' programs
  - Rich typing system
  - Easy to translate to regular C++ and Java

Note: Where C++ and Java differ, we select the common denominator

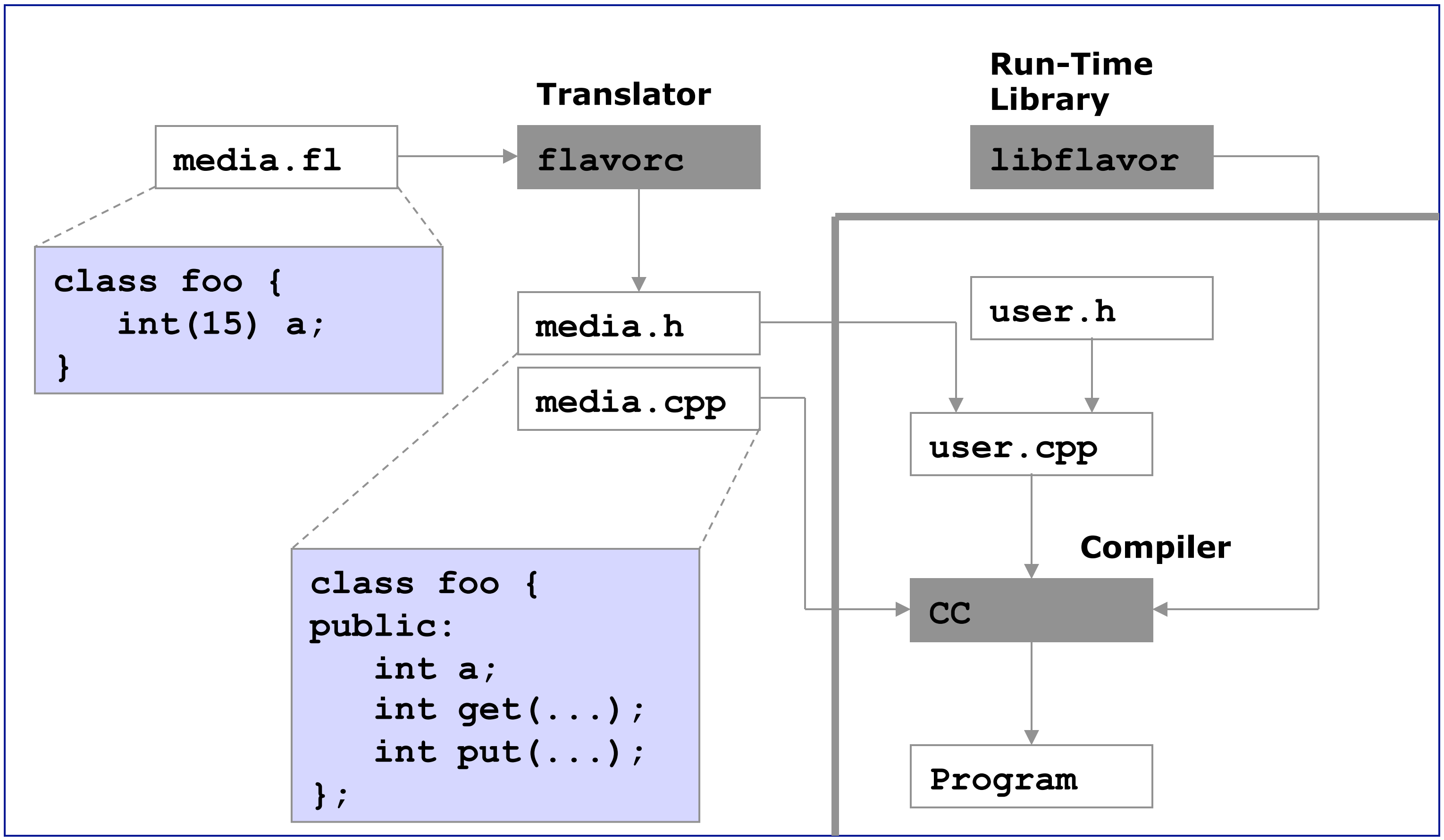
“Hello Bits!”

- “Hello World!” in Flavor

C++/Java	Flavor

<pre>class HelloBits {
    char data;
    void get(void){
       data=getint(8);
    }
};</pre>
	<pre>class HelloBits {
	    char(8) data;
	}</pre>

Using Flavor in Programs



Status

- Used for syntax description in MPEG-4 (Systems, Structured Audio)
- About 1,000 users in diverse fields (even VLSI design)
- Open-sourced since 10/2001
- XFlavor – XML support (with D. Hong)
- Winner – 1st ACM Multimedia Open Source Software Competition (ACM MM-2004)

Flavor Web Site:

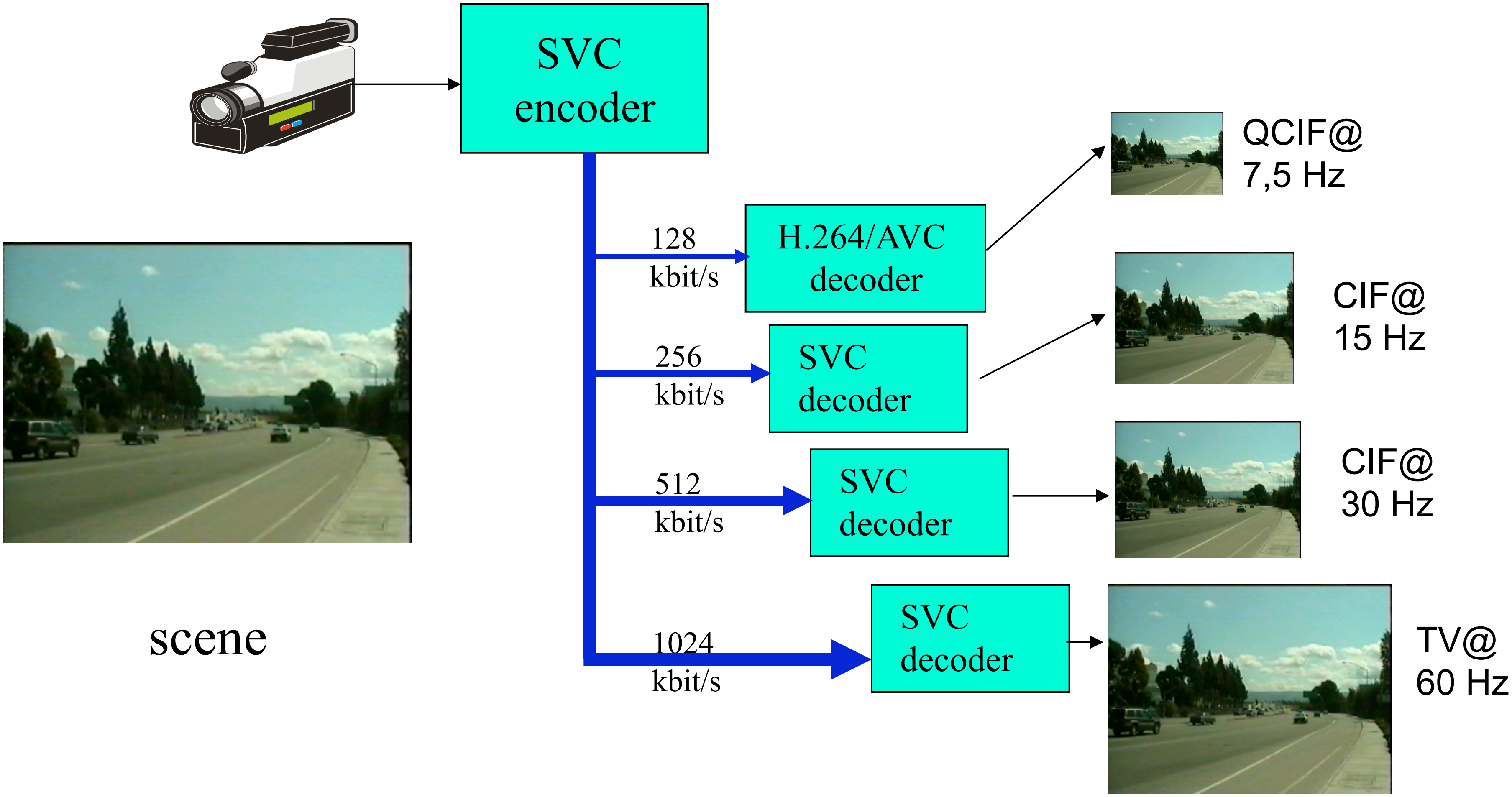
http://flavor.sourceforge.net

- Download Version 5 software
(source/binaries for Win32 and all UNIX)
- Extensive Documentation and Papers

H.264 SVC (Scalable Video Coding)

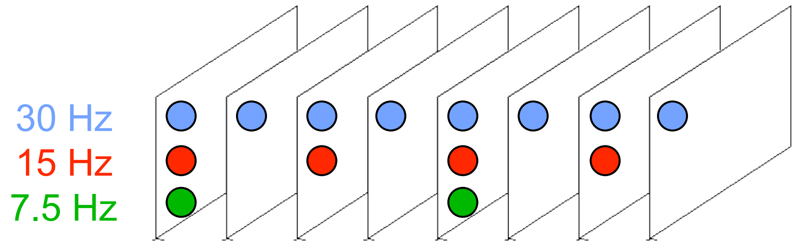
Σημ.: Μερικές διαφάνειες και διαγράμματα είναι από τον Thomas Wiegand, HHI

Scalable Video Coding (SVC) Principle



Scalability of Video - Modalities

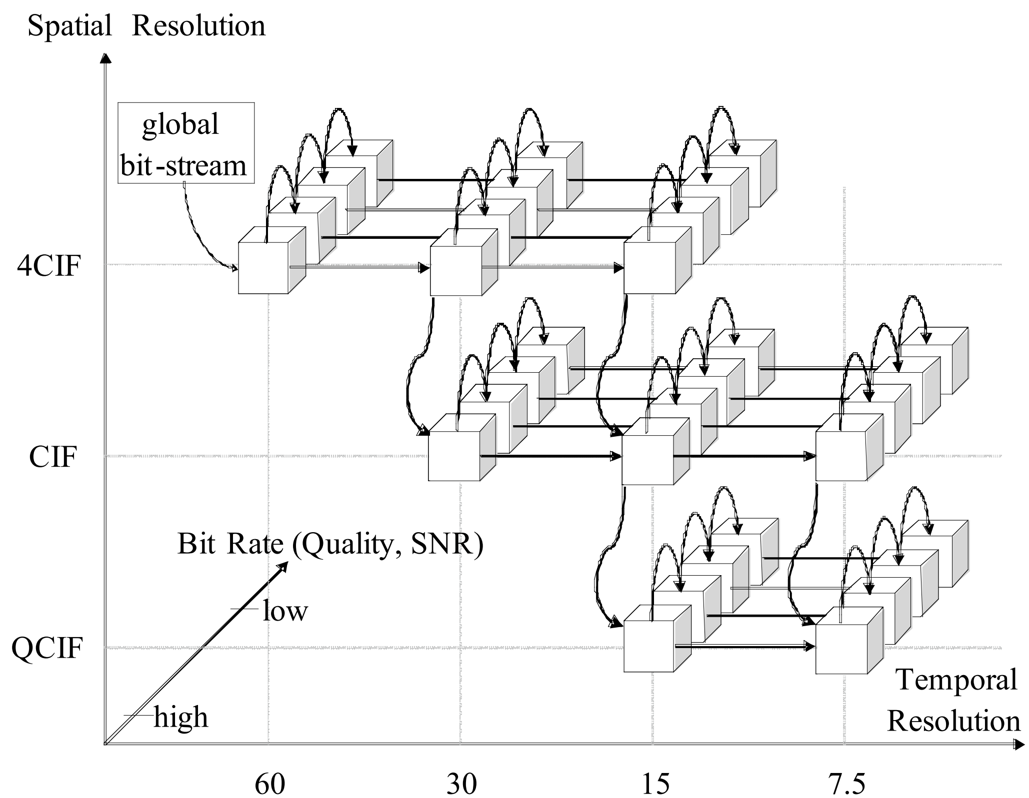
- Temporal: change of frame rate



- Spatial: change of frame size

- Fidelity: change of quality (e.g. SNR)

SVC Dependency Structure

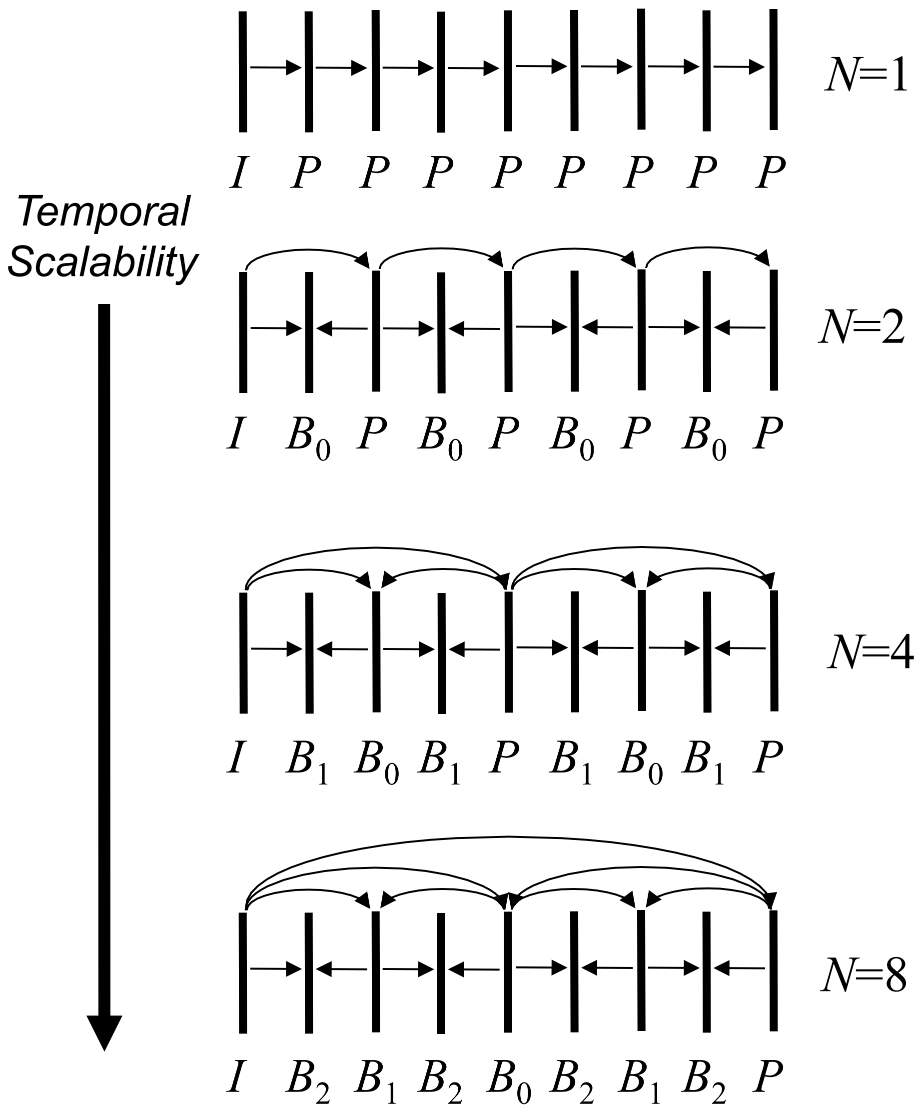


- In scalable video coding there are multiple scalability and dependency dimensions

New Scalability Ecosystem

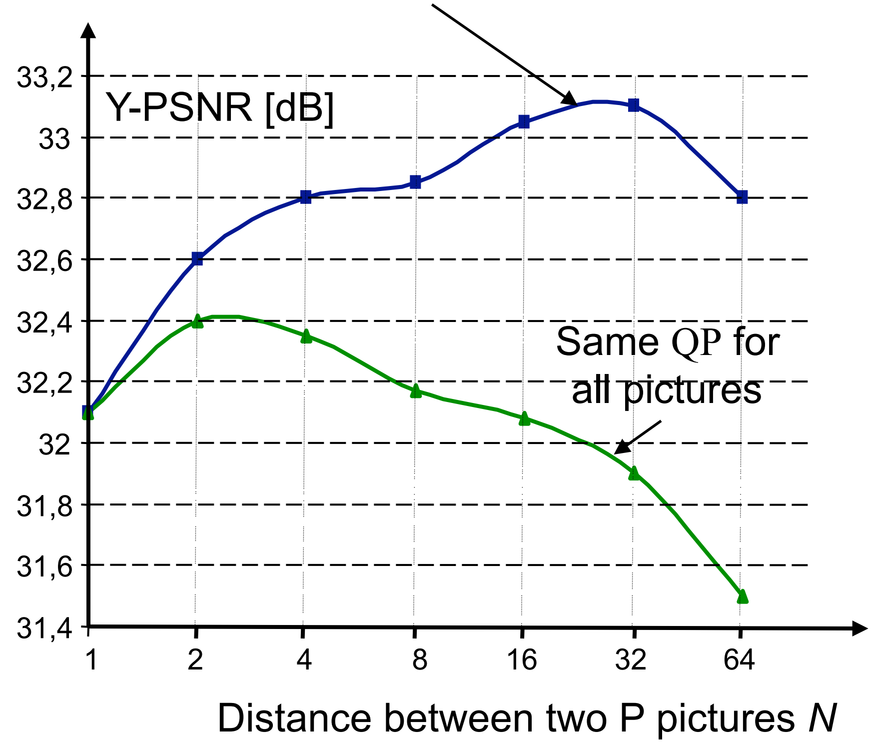
- **Prior scalable video coding standards: MPEG-2, H.263, MPEG-4 failed because of technical shortcomings and**
  - Graceful degradation not needed: transmission channel works or not
  - Bit rate adaptation not needed: transmission channel has fixed throughput
  - Power adaptation not needed: no handheld devices
  - Format adaptation not needed: just one single format – standard definition
- **Today, the situation has changed**
  - Internet and mobile transmission are becoming primary distribution mechanisms
    - Shared resource systems with varying throughput and errors
    - *Graceful degradation, bit rate adaptation, power adaptation*
  - A variety of terminals and displays from QCIF, QVGA, VGA, SD, to HD
    - Backwards compatible extension
    - *Format adaptation*: QVGA ➔ VGA, 720p/1080i ➔ 1080p

Temporal Decomposition of Video

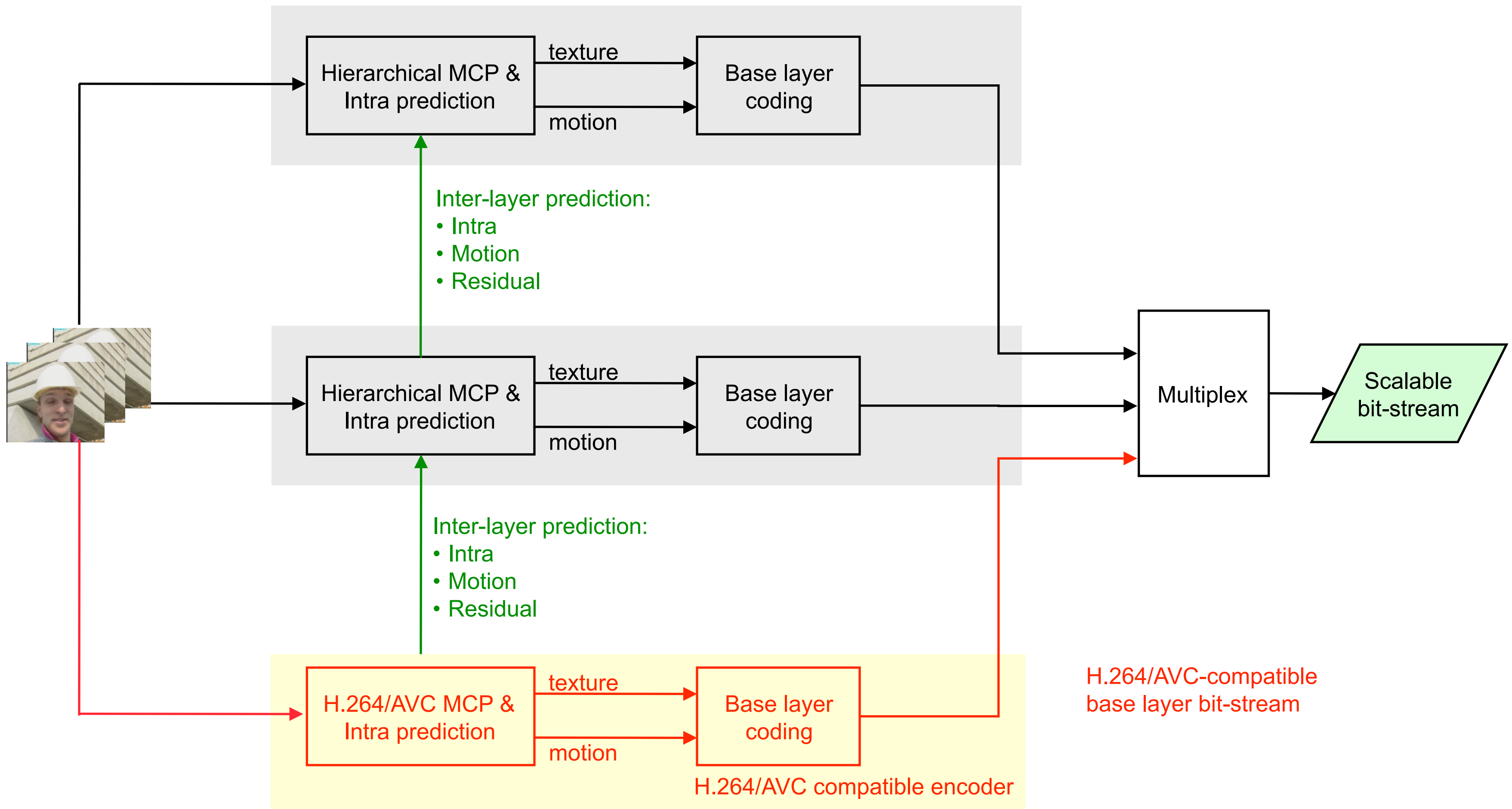


Video Coding Experiment with H.264/AVC:
Foreman, CIF 30Hz @ 132 kbit/s

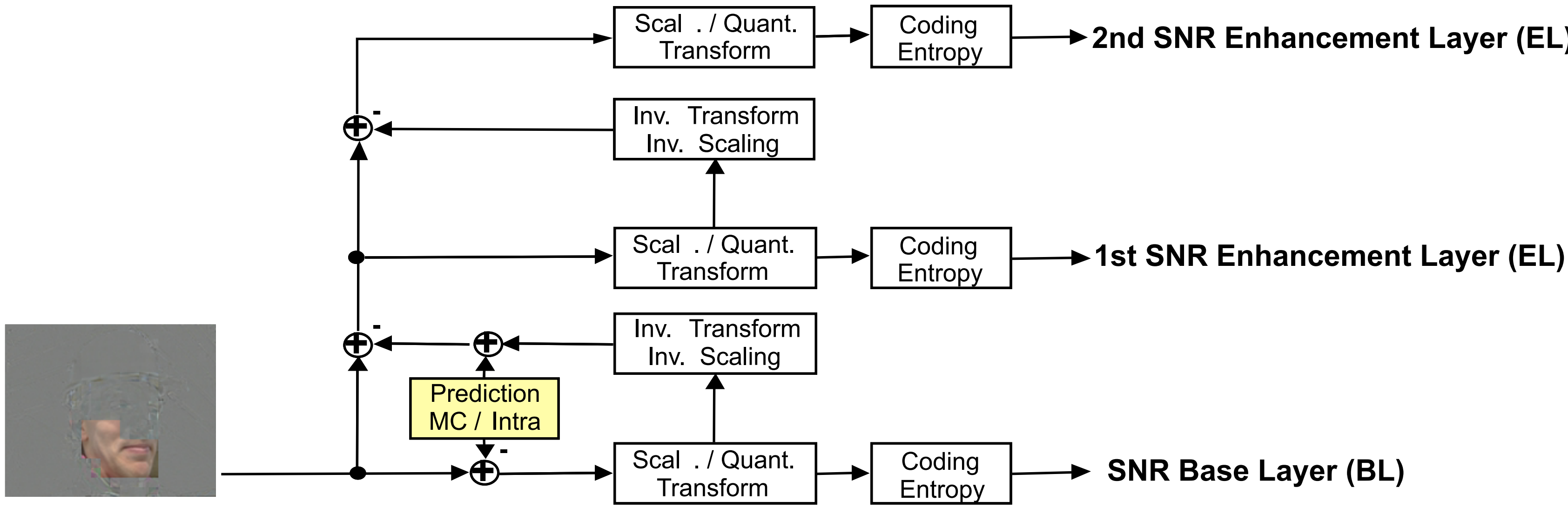
Cascaded QP assignment
I/*P*: best quality,
B_0: much worse quality than *I*/*P*,
B_1: slightly worse quality than B_0, …



SNR Scalability: Typical Encoding



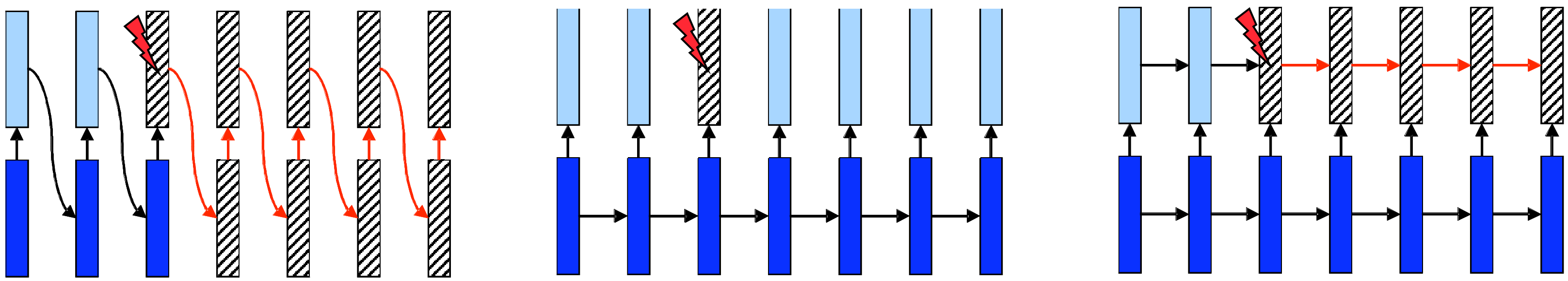
Layered Coding for SNR Scalability



- Same resolution of pictures over all layers
- Residual coding of the quantization error between original pictures and their lower layer reconstruction
- Can be converted to H.264/AVC without transcoding at any bit rate

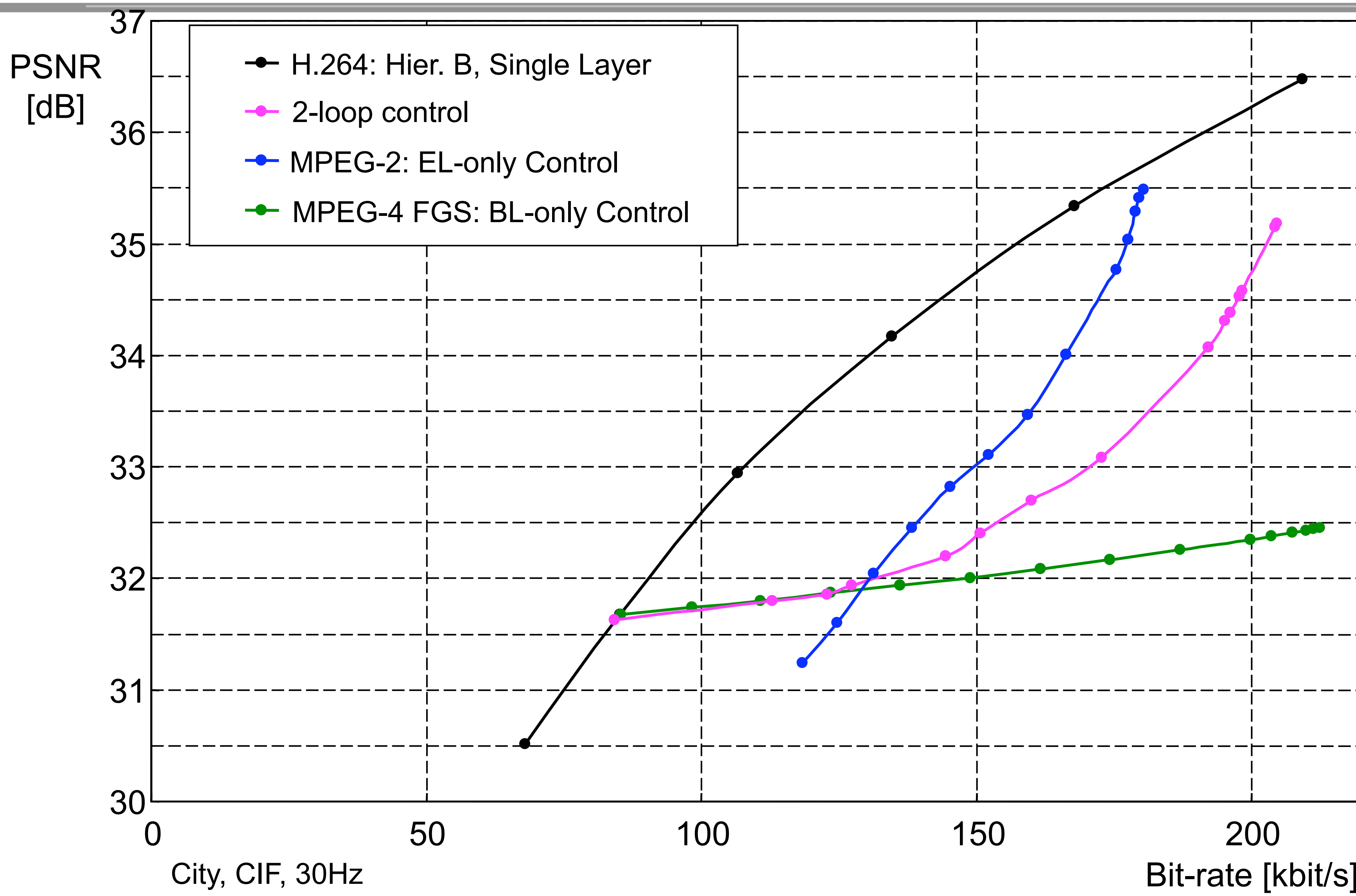
Drift in Past SNR Scalable Coding

- Source of drift:
  Non-synchronized motion compensation loops in encoder and decoder due to loss of enhancement layer information



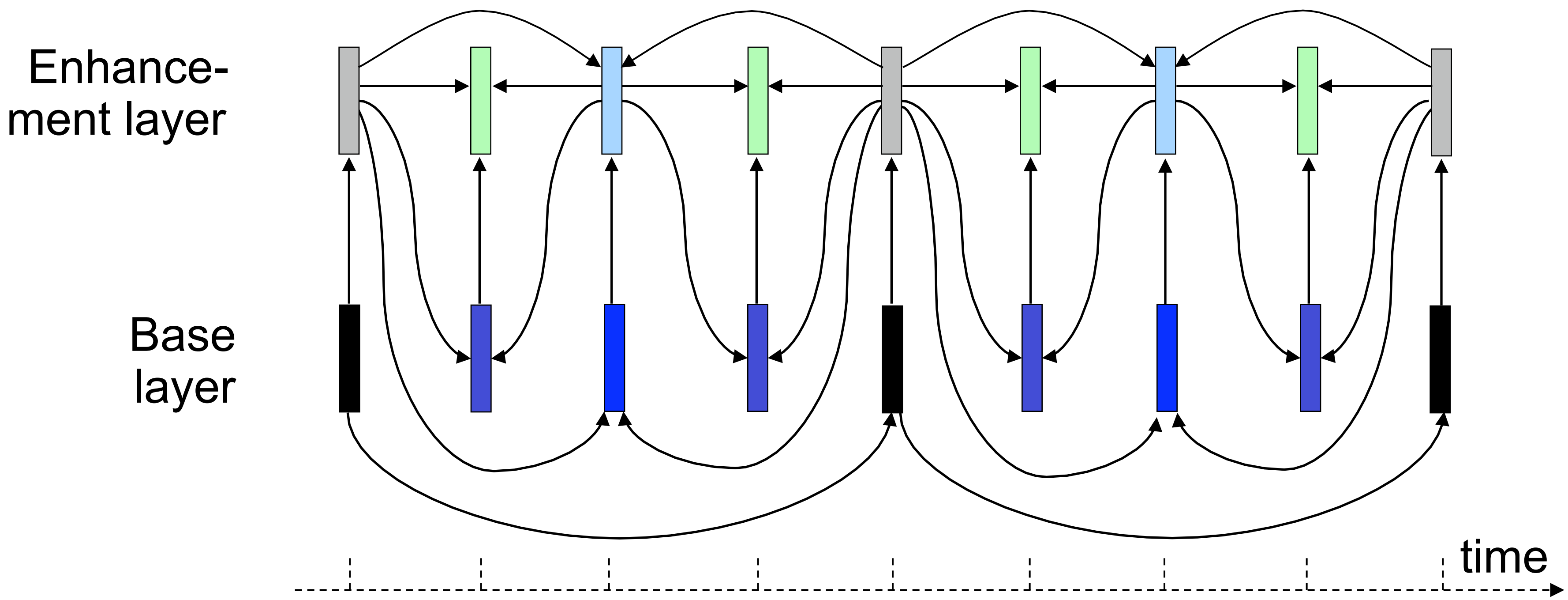
- **EL only control**
  - MPEG-2 SNR
  - Drift in both BL and EL
  - Low complexity
  - Inefficient BL and efficient EL coding (when latter complete)
- **BL only control**
  - MPEG-4 FGS
  - No drift
  - Low complexity
  - Efficient BL and inefficient EL coding
- **2-loop control**
  - No drift in BL
  - Drift in EL
  - High complexity
  - Efficient BL and medium efficient EL coding

SNR Scalability Results: Past Codecs

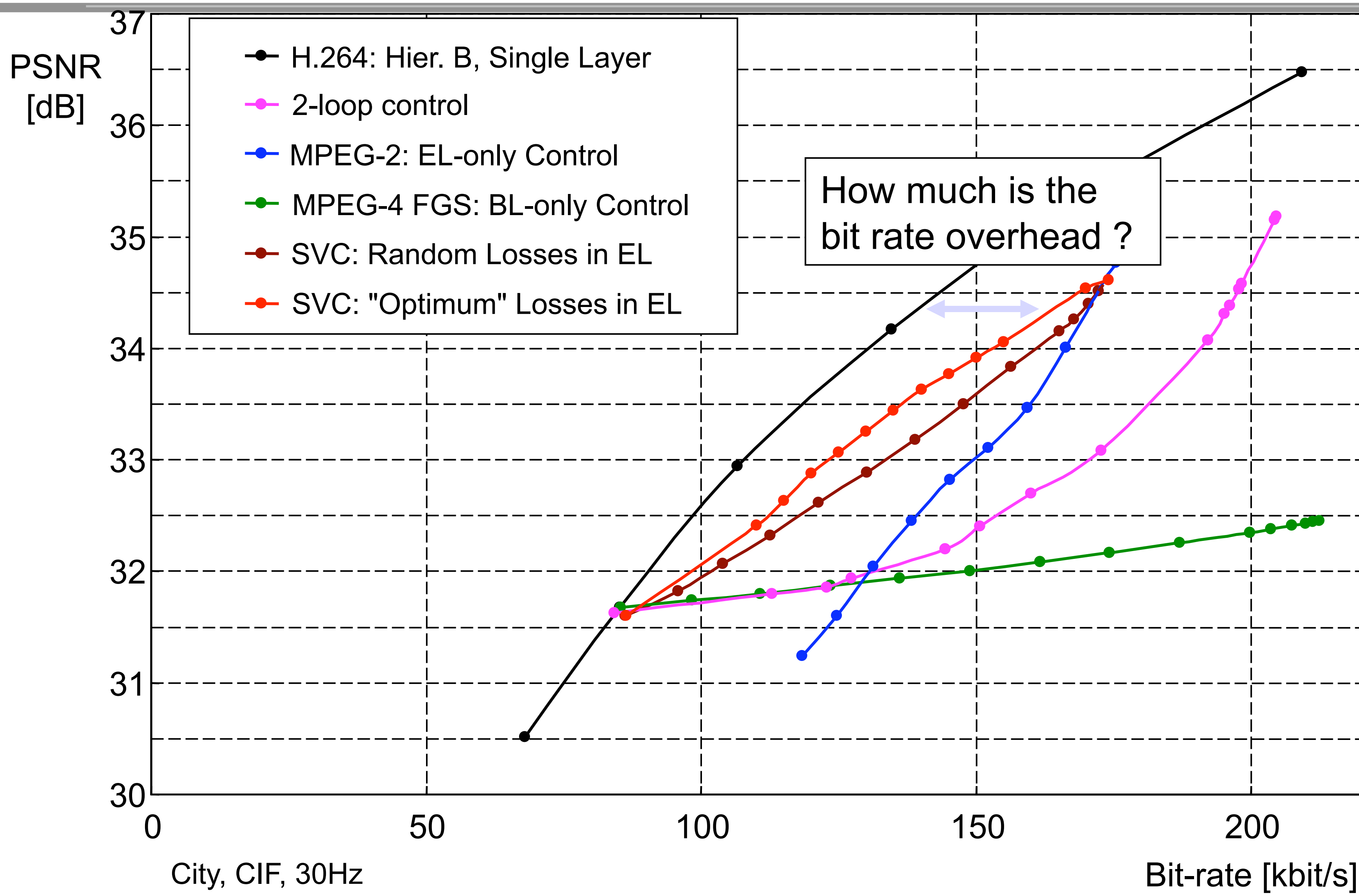


Drift Control in SVC

- Adaptive BL/EL only encoder control for hierarchical prediction structures
- Single motion compensation loop decoding



SNR Scalability Results: New SVC



Coder Control

- **Constrained problem**:

$$\min_{\boldsymbol{p}} D(\boldsymbol{p}) \quad \text{s.t.} \quad R(\boldsymbol{p}) \le R_T$$

D Distortion
R Rate
R_T Target rate
$\boldsymbol{p}$ Parameter Vector

- **Unconstrained Lagrangian formulation**:

$$\boldsymbol{p}' = \arg\min_{\boldsymbol{p}} \left\{ D(\boldsymbol{p}) + \lambda \cdot R(\boldsymbol{p}) \right\}$$

with *l* controlling the rate-distortion trade-off

- **Rate-Constrained Mode Decision**

$$D_2(M|QP)+\lambda_M R(M|QP)$$

M Evaluated macroblock mode out of a set of possible modes
QP Value of quantizer for transform coefficients
l_M Lagrange parameter for mode decision
D_2 Sum of squared differences (luminance & chrominance)
R Number of bits associated with header, motion, transform coefficients

- **Rate-Constrained Motion Estimation**

$$D_1(\boldsymbol{m},\mathrm{D})+\lambda_D R(\boldsymbol{m},D)$$

$\boldsymbol{m}$ Motion vector containing spatial displacement
D Picture reference parameter
l_D Lagrange parameter for motion estimation
D_1 Sum of absolute differences (luminance)
R Number of bits associated with motion information

Encoder Optimization: JSVM encoder control

- Lagrangian bit-allocation techniques: bottom-up encoding process
- Encode base layer:
  - Coding parameters are optimized for the base layer

$$\boldsymbol{p}_0' = \arg\min_{\{\boldsymbol{p}_0\}} \left[D_0(\boldsymbol{p}_0) + \lambda_0 \cdot R_0(\boldsymbol{p}_0) \right]$$

- Encode enhancement layer
  - Coding parameters are optimized for the enhancement layer

$$\boldsymbol{p}_1' = \arg\min_{\{\boldsymbol{p}_1 | \boldsymbol{p}_0\}} \left[D_1(\boldsymbol{p}_1 | \boldsymbol{p}_0) + \lambda_1 \cdot R_1(\boldsymbol{p}_1 | \boldsymbol{p}_0) \right]$$

- All enhancement layer decisions are conditioned on already determined base layer coding parameters

Encoder Optimization: Joint Control

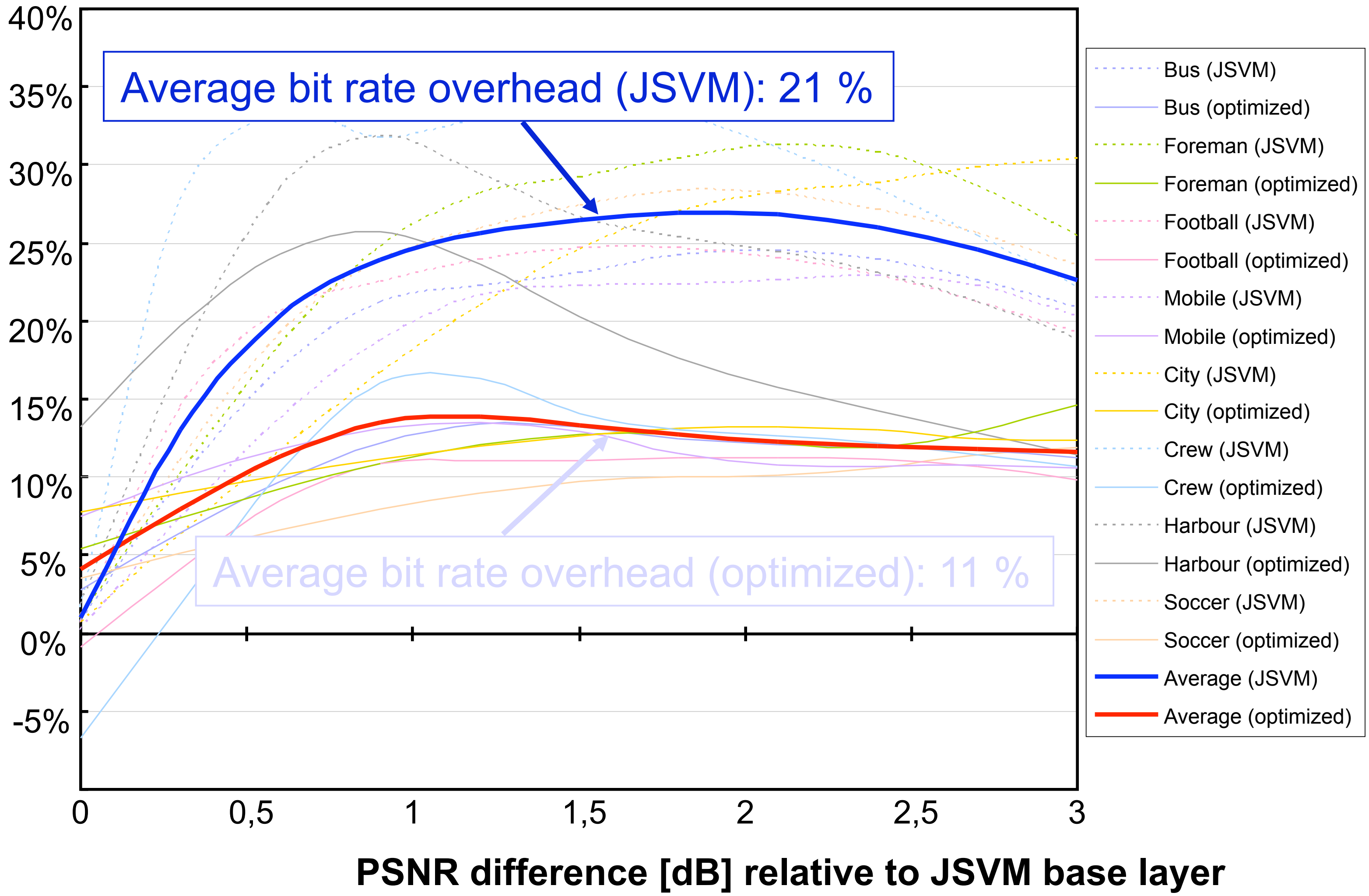
- Jointly optimize base and enhancement layer coding parameters
- Consider enhancement layer during base layer encoding
- Modified Lagrangian approach for base layer encoding

$$\boldsymbol{p}_0 = \arg\min_{\{\boldsymbol{p}_0, \boldsymbol{p}_1 | \boldsymbol{p}_0\}} \left[(1-w) \cdot \left(D_0(\boldsymbol{p}_0) + \lambda_0 \cdot R_0(\boldsymbol{p}_0) \right) + w \cdot \left(D_1(\boldsymbol{p}_1 | \boldsymbol{p}_0) + \lambda_1 \cdot R_1(\boldsymbol{p}_1 | \boldsymbol{p}_0) \right) \right]$$

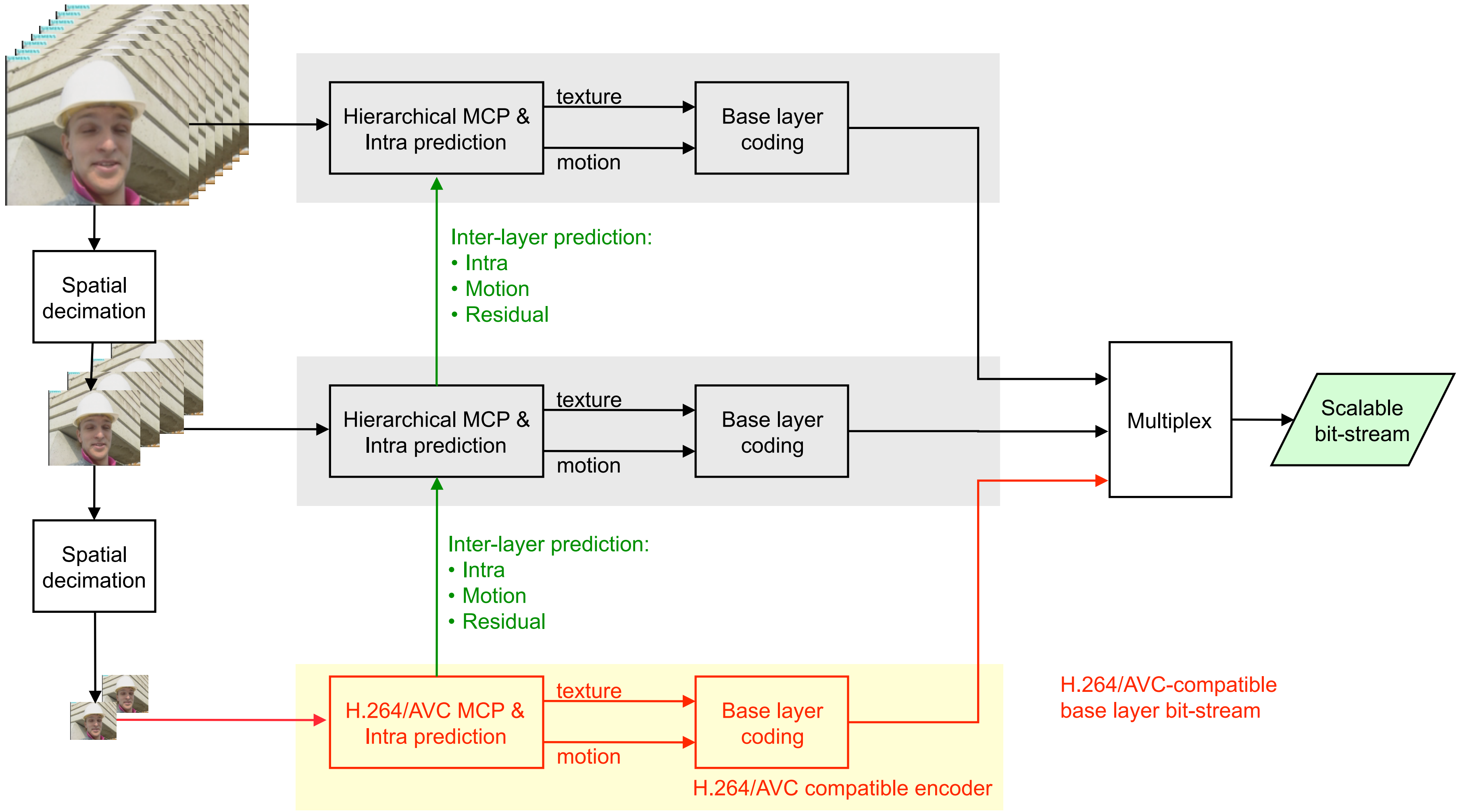
- Weighting factor w
  - $w = 0$: Current JSVM encoder control
  - $w = 1$: Single-loop encoder control (base layer is not controlled)
  - $0 < w < 1$: Partly consider enhancement layer during base layer encoding
- Enhancement layer encoding is not modified

Results for SNR Scalability

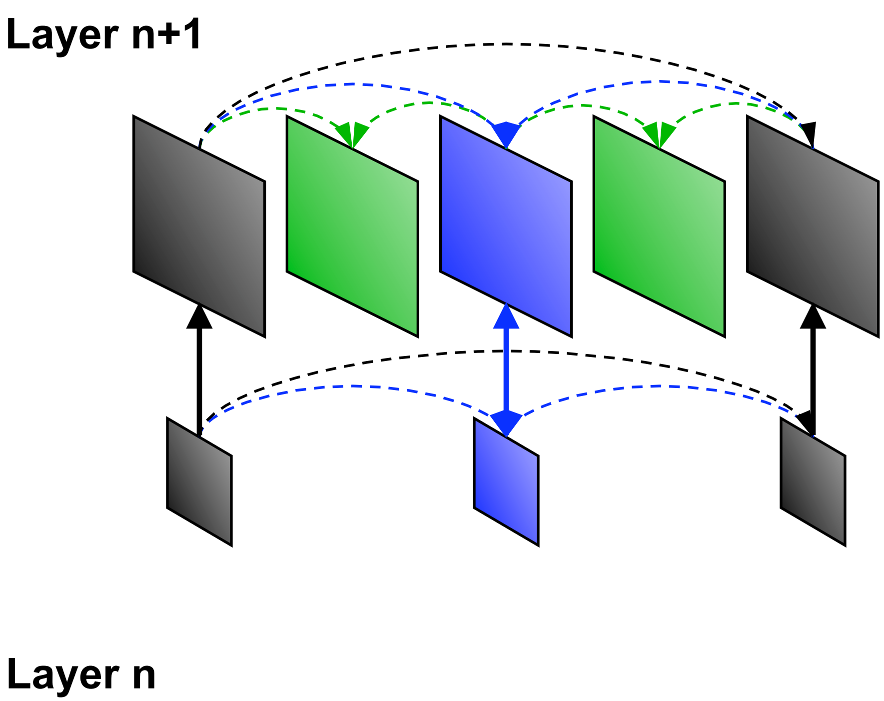
Average bit rate overhead relative to single layer coding



Spatial Scalability: Typical Encoding



Layered Coding for Spatial Scalability



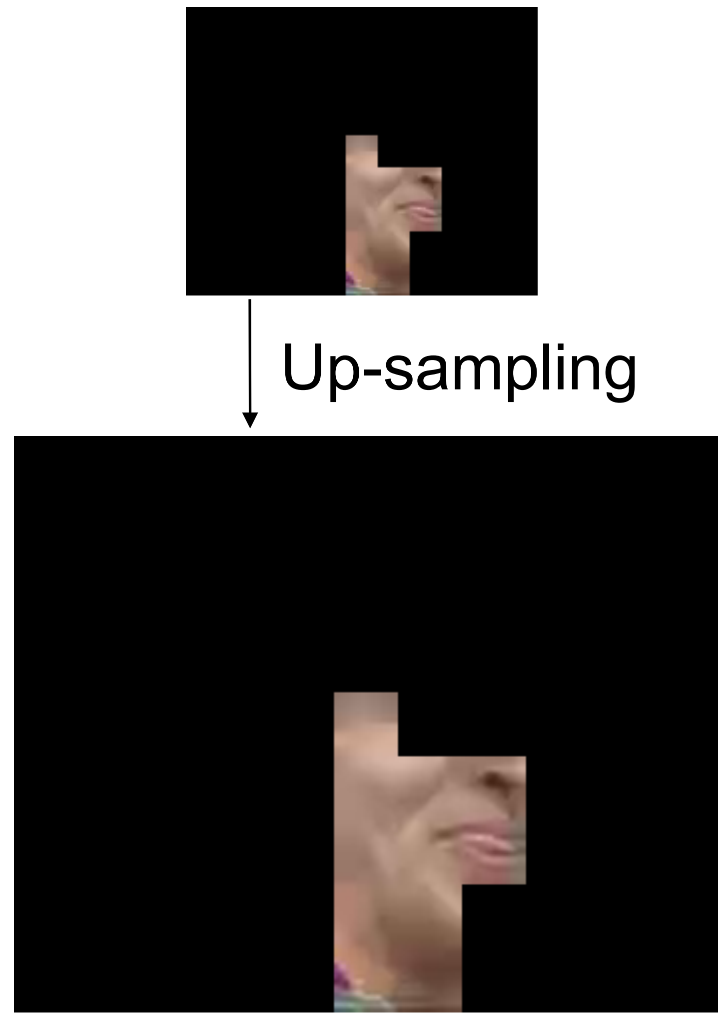
- Layered coding
  - Oversampled pyramid for each resolution: QCIF, CIF, 4CIF, 16CIF
  - Independent MC prediction structure in each layer
- Inter-layer prediction: Switchable prediction (with upsampling)
  - Prediction of intra macroblocks
  - Prediction of partitioning and motion information - NEW
  - Prediction of residual data (prediction errors) - NEW

- Single motion-compensation loop for decoding - NEW
  - Reconstruction signal for motion-compensated macroblock cannot be used for inter-layer prediction
  - Loss between 0 – 0.5 dB

Inter-Layer Prediction: Intra Prediction

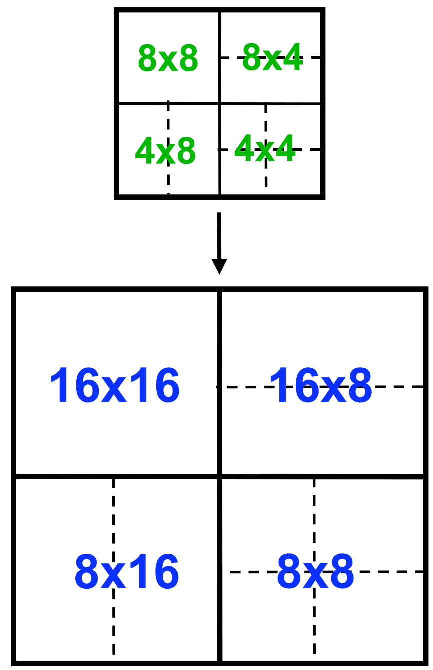
- **Similar to MPEG-2 / H.262, H. 263, and MPEG-4**

- **Additional intra macroblock type**
  - Switch between existing Intra_4x4, Intra_8x8, or Intra_16x16 and new Intra_BL mode
  - Intra_BL: prediction signal is generated by up-sampling the co-located signal in the base layer
    - Luma: 4-tap filter
    - Chroma: bi-linear filter



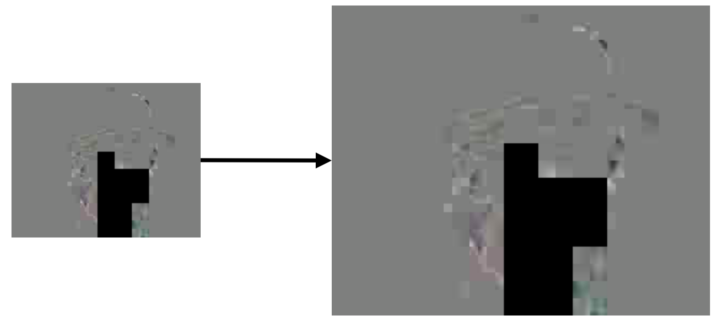
Inter-Layer Prediction: Motion (New)

- Additional inter macroblock type: data are derived from base-layer
  - **base_mode_flag**
  - Macroblock partitioning
  - Reference indices
  - Motion vectors
- Additional motion vector prediction mode from base layer for conventional macroblock types
  - **motion_pred_flag**
  - inter-layer motion predictor
  - existing spatial predictor



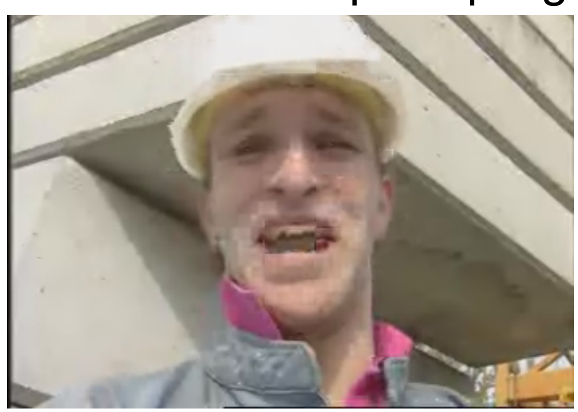
Inter-Layer Prediction: Residual (New)

- Flag for inter macroblock types
- Signalling that residual is predicted by up-sampled base layer residual signal

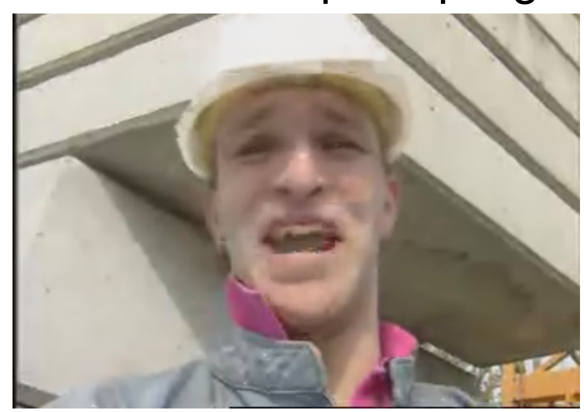


- Block-wise, bi-linear up-sampling

Not blockwise upsampling



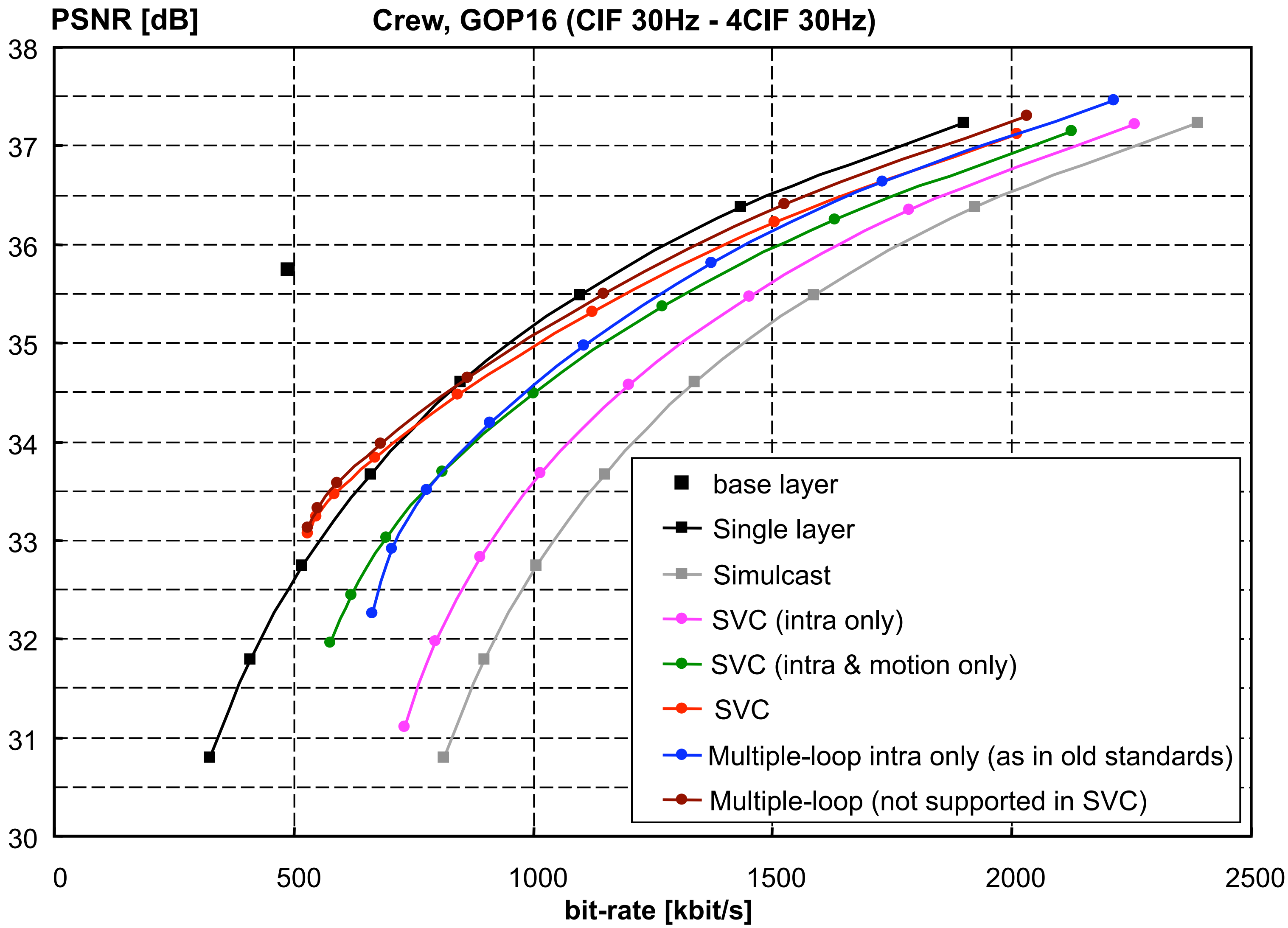
Blockwise upsampling



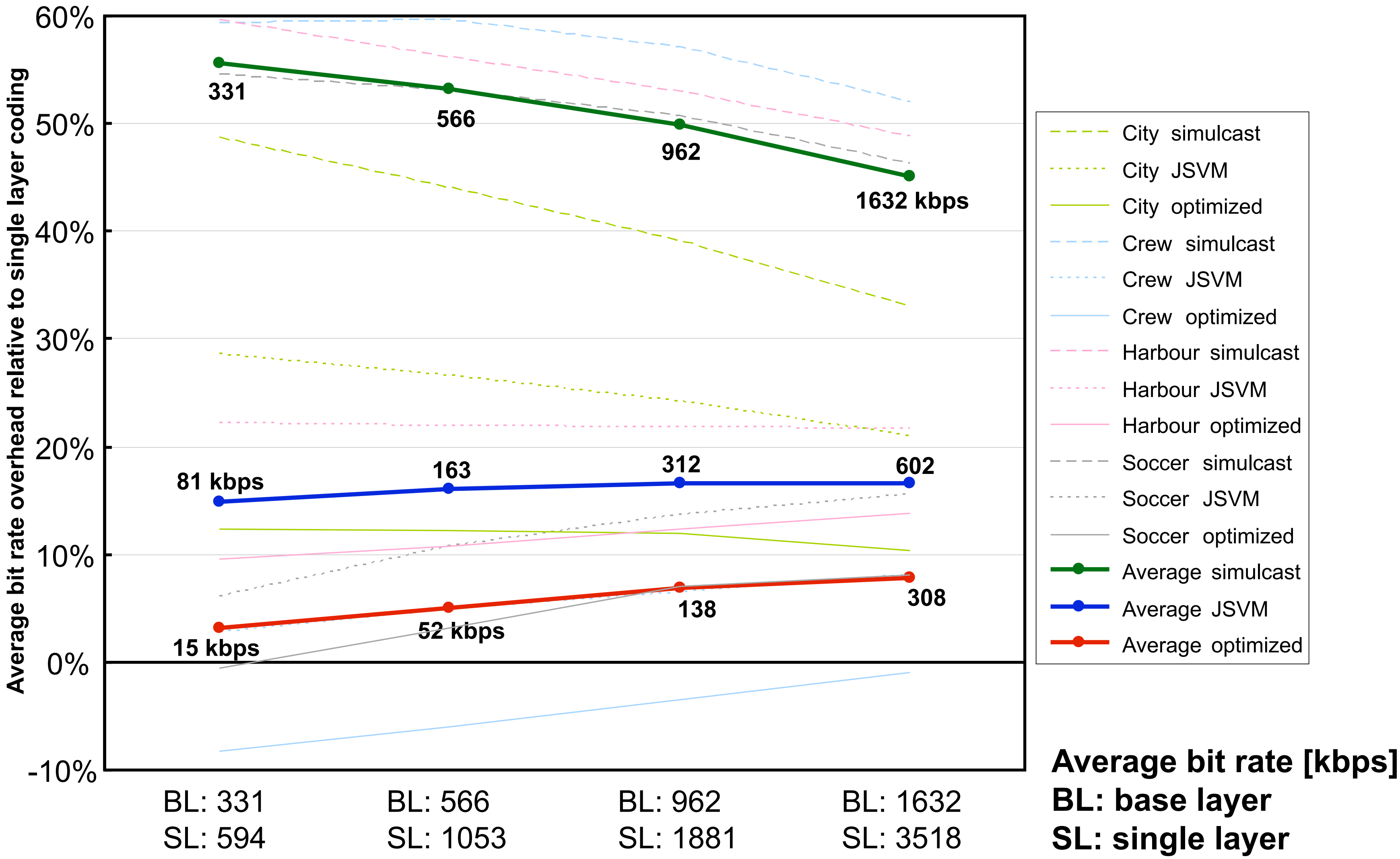
Single-loop Decoding (New)

- Past spatial scalable video:
  - Inter-layer intra prediction requires that base layer is completely reconstructed
  - Decode multiple motion compensation loops (and deblocking filter)
  - Complexity larger than simulcast: full decoding plus inter-layer prediction
- In SVC: Inter-layer intra prediction is restricted to macroblocks for which the co-located base layer signal is intra-coded
  - No need to decode multiple motion compensation loops
  - Complexity overhead comparable to single-layer coding

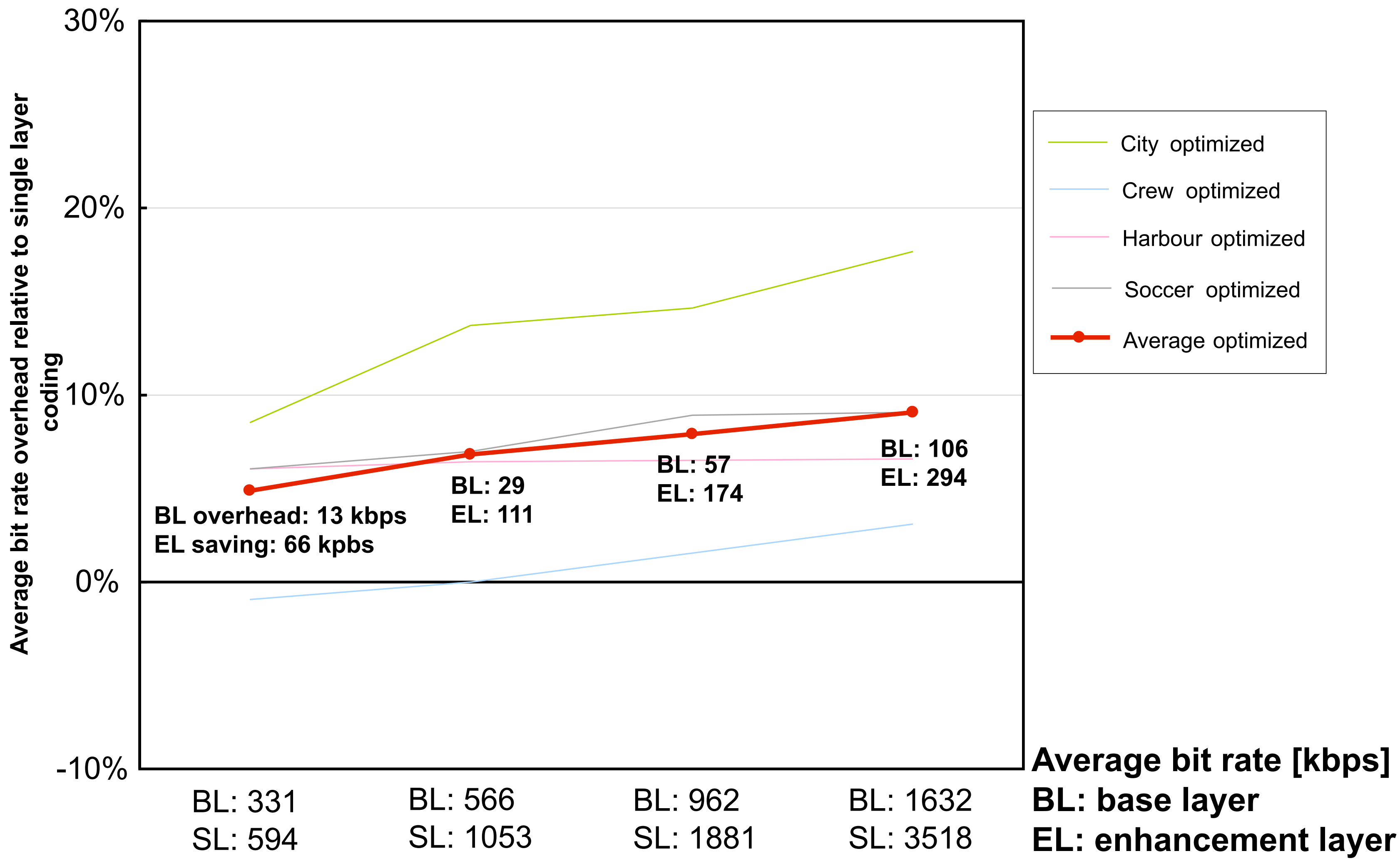
Impact on Coding Efficiency



Enhancement Layer Bit rate Overhead for CIF/30Hz ➔ 4CIF/30Hz Spatial Scalability



Base Layer Bit rate Overhead in Joint Optimization for CIF/30Hz ➔ 4CIF/30Hz Spatial Scalability



Summary of SVC

- H.264/AVC hierarchical B pictures allow better coding efficiency and temporal scalability
- Scalable Video Coding is the next step video applications
  - SVC realized through layered extension of H.264/AVC
  - Power adaptation: decode appropriate part of bit-stream
  - Graceful degradation when the “right” parts of the bitstream get lost
  - Format adaptation: backwards compatible extension in mobile TV: QVGA⇨VGA
  - Interlaced and other ratios than 2:1 are also specified, e.g.. 1.5:1 for 720p to 1080p
- What’s next in SVC standardization?
  - Bit-depth scalability (8bit 4:2:0 -> 10bit 4:2:0)
  - Color format scalability (4:2:0 -> 4:4:4)