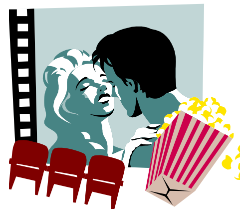
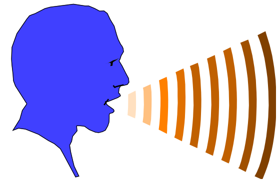





Image and Video Processing

Video Coding Standards and Video Streaming

Yao Wang
Tandon School of Engineering, New York University

Outline

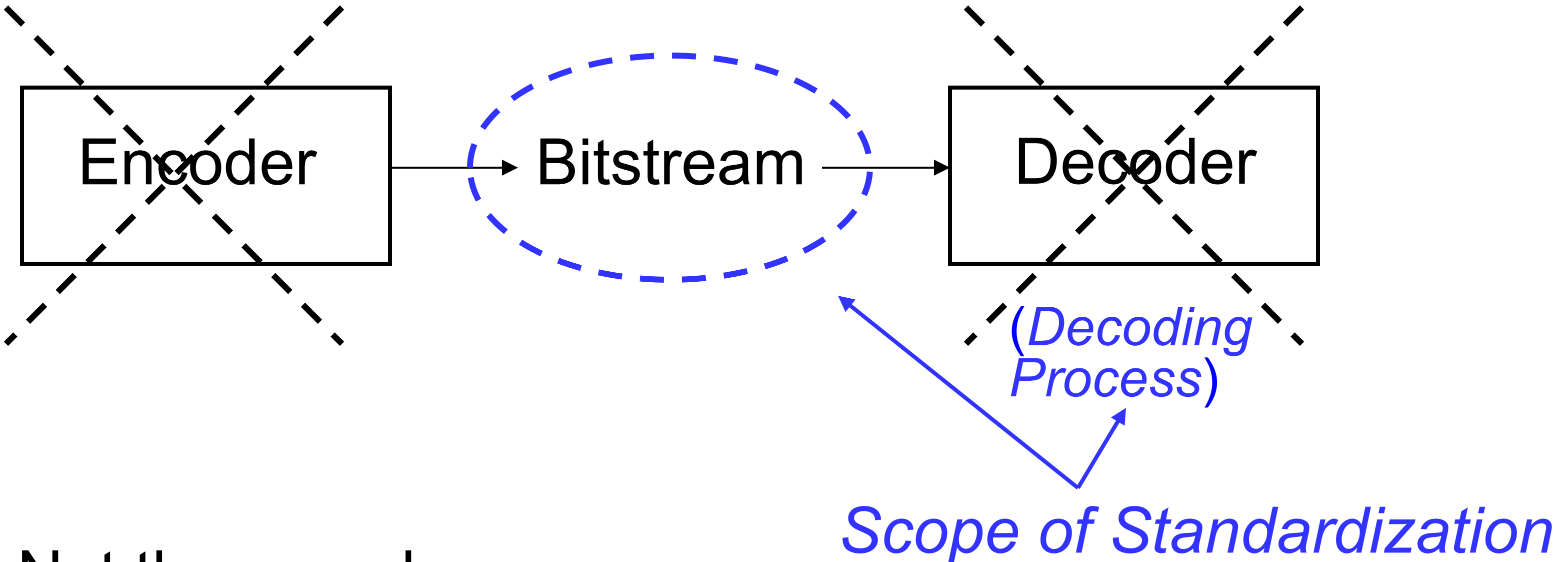
- Role of standards and summary of standards
- From H.264/AVC to H.265/HEVC
- Other standards
- Video streaming challenges
- Scalable coding

Why do we need standards?

- Goal of standards:
  - *Ensuring interoperability:* Enabling communication between devices made by different manufacturers
  - Promoting a technology or industry
  - Reducing costs

From John Apostolopoulos

What do the Standards Specify?



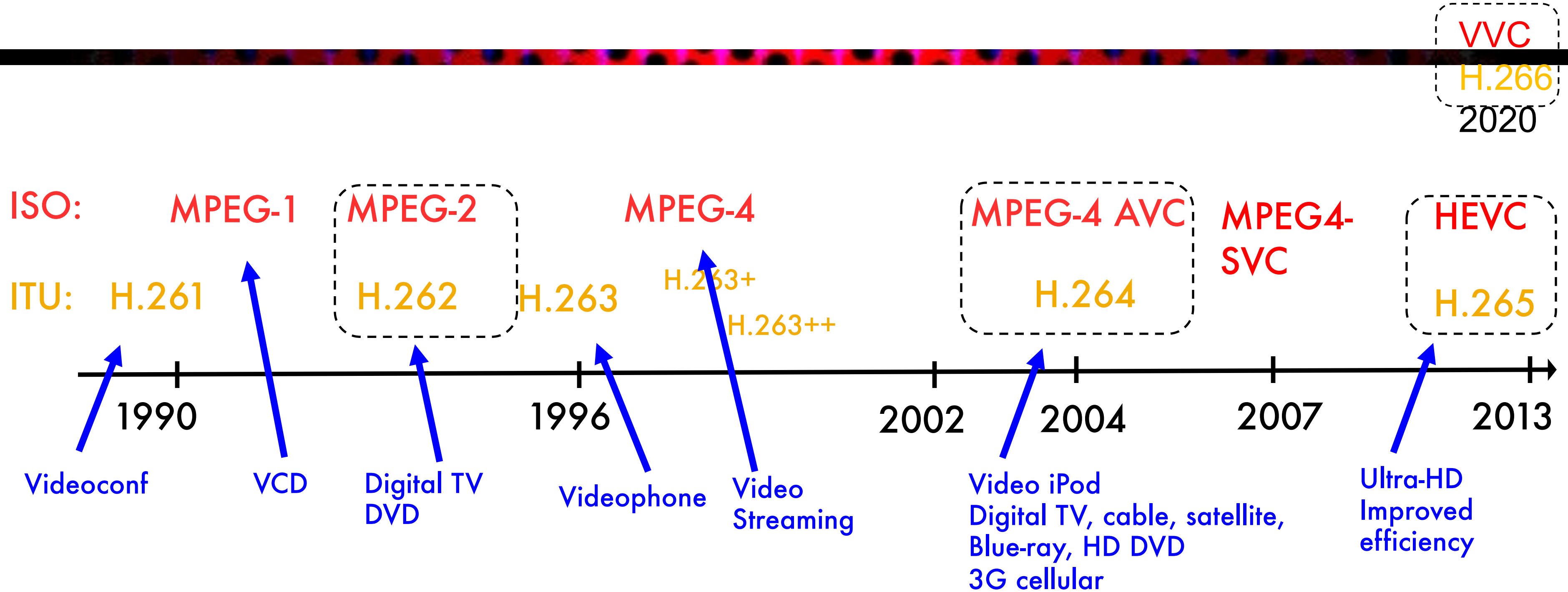
- Not the encoder
- Not the decoder
- Just the *bitstream syntax* and the *decoding process* (e.g., use IDCT, but not how to implement the IDCT)
  → Enables improved encoding & decoding strategies to be employed in a standard-compatible manner

From John Apostolopoulos

Video coding standards

- Video coding standards define the operation of a decoder given a correct bitstream
- They do NOT describe an encoder
- Video coding standards typically define a toolkit
- Not all pieces of the toolkit need to be implemented to create a conforming bitstream
- Decoders must implement some subset of the toolkit to be declared "conforming"
- Standard bodies:
  - MPEG: Motion Picture Expert Group of International Standard Organization, each standard includes parts for system, video, audio, etc.
  - H.26x: Video coding group under International Telcommunications Union (ITU)
  - JVT: Joint video team of MPEG and ITU-T. Standards starting with H.264/MPEG4-AVC were developed by this group jointly

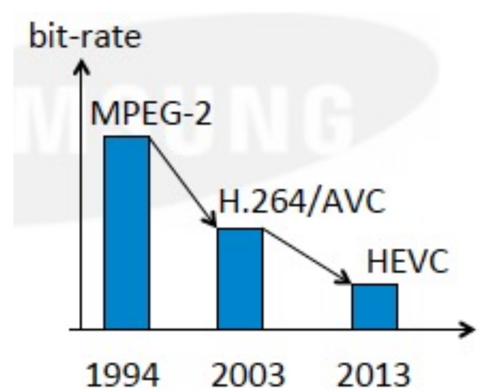
History of Video Coding Standards



- Above figure modified from Amy Reibman
- Right figure from SzeBudagavi[2014]

~2x Improvement in compression ratio every decade!

Most recent effort: Versatile Video Coding (VVC)/H.266, finalized July 2020



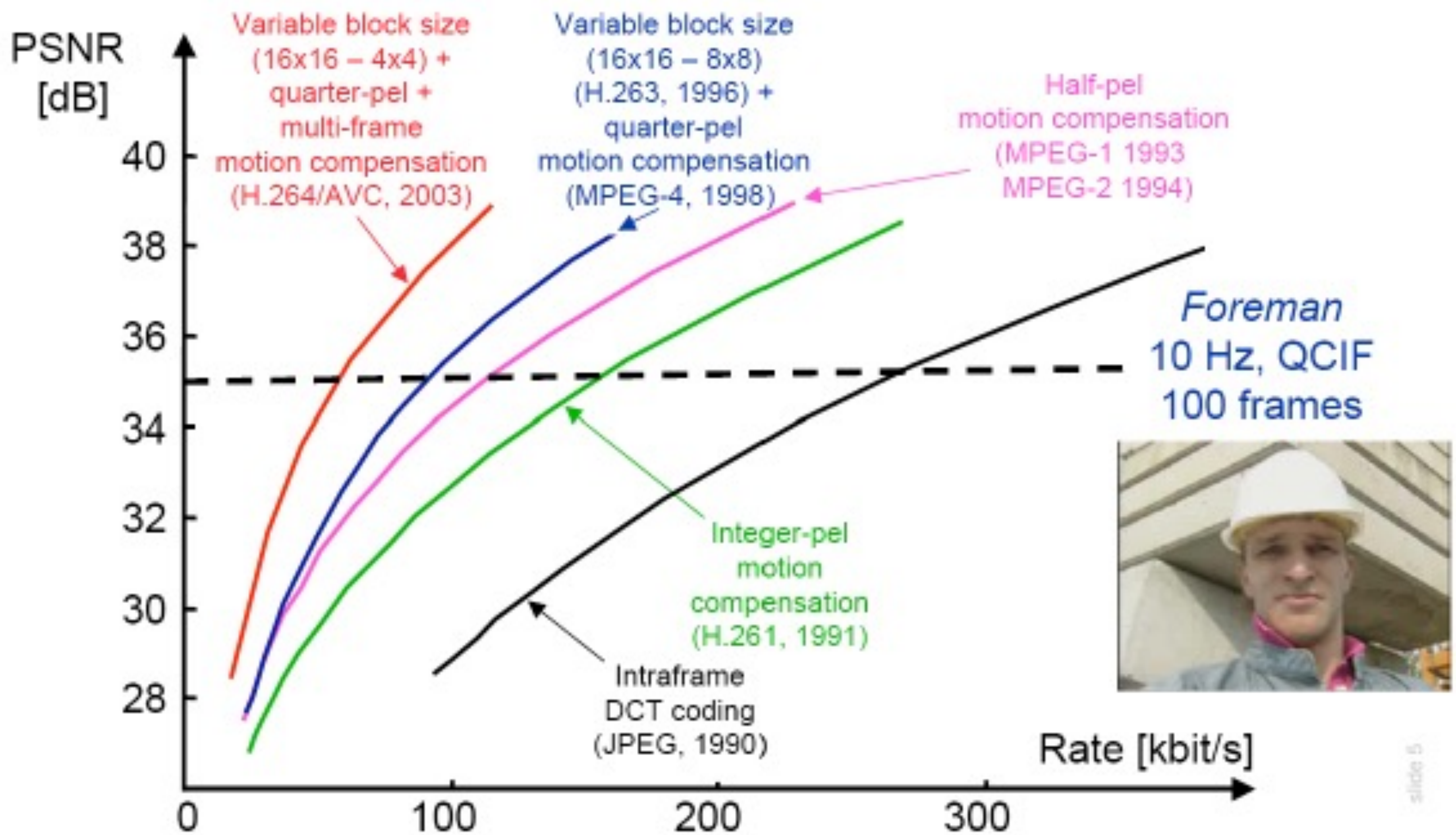
From [Sze2014]

Summary of Standards (1)

- H.261 (1990):
  - First video coding standard, targeted for video conferencing over ISDN
  - Uses block-based hybrid coding framework with integer-pel MC, no intra-prediction, fixed block size
- H.263:
  - Improved quality at lower bit rate, to enable video conferencing/telephony below 54 bkps (modems or internet access, desktop conferencing)
  - Half-pel MC and other improvement (Variable block sizes)
  - H.263 (1995) ->H.263+ (1997) -> H.263++ (2000)
- MPEG-1 video (1992)
  - Video on CD (good quality at 1.5 mbps)
  - Video streaming on the Internet
  - Half-pel MC and bidirectional MC
- MPEG-2 video (1996)
  - Digital SDTV/HDTV/DVD (4-15 mbps)
  - Extended from MPEG-1
  - Additional MC modes for handling interlaced video
  - First standard considering scalability
  - Supersedes MPEG-3 planned for HD

Summary of Standards (2)

- MPEG-4 video (MPEG4-part 2) (1999)
  - Video over internet in addition to broadcasting/DVD
  - Object-oriented coding: to enable manipulation of individual objects
    - Coding of shapes
  - Coding of synthetic audio and video (animations)
  - Fine granularity scalability (FGS)
- MPEG4/AVC (MPEG4-part 10) / H.264 (2003)
  - Improved coding efficiency (approx. doubling) over MPEG4
- H.264/SVC
  - Improved scalable coding on top of H.264/AVC
- HEVC/H.265 (2013)
  - Improved coding efficiency (approx. doubling) over AVC/H.264
- VVC/H.266 (2020)
  - Improved coding efficiency over HEVC/H.265
  - Support UHD (4K, 8K video), HDR, 360 degree video, multi-channel (depth, Multiview)
  - Expected to be finalized in 2020



Source: T. Wiegand, JVT-W132, 2007

From [Sze2014]

Outline

- Role of standards and summary of standards
- HEVC/H.265
- Other standards
- Video streaming challenges
- Scalable coding

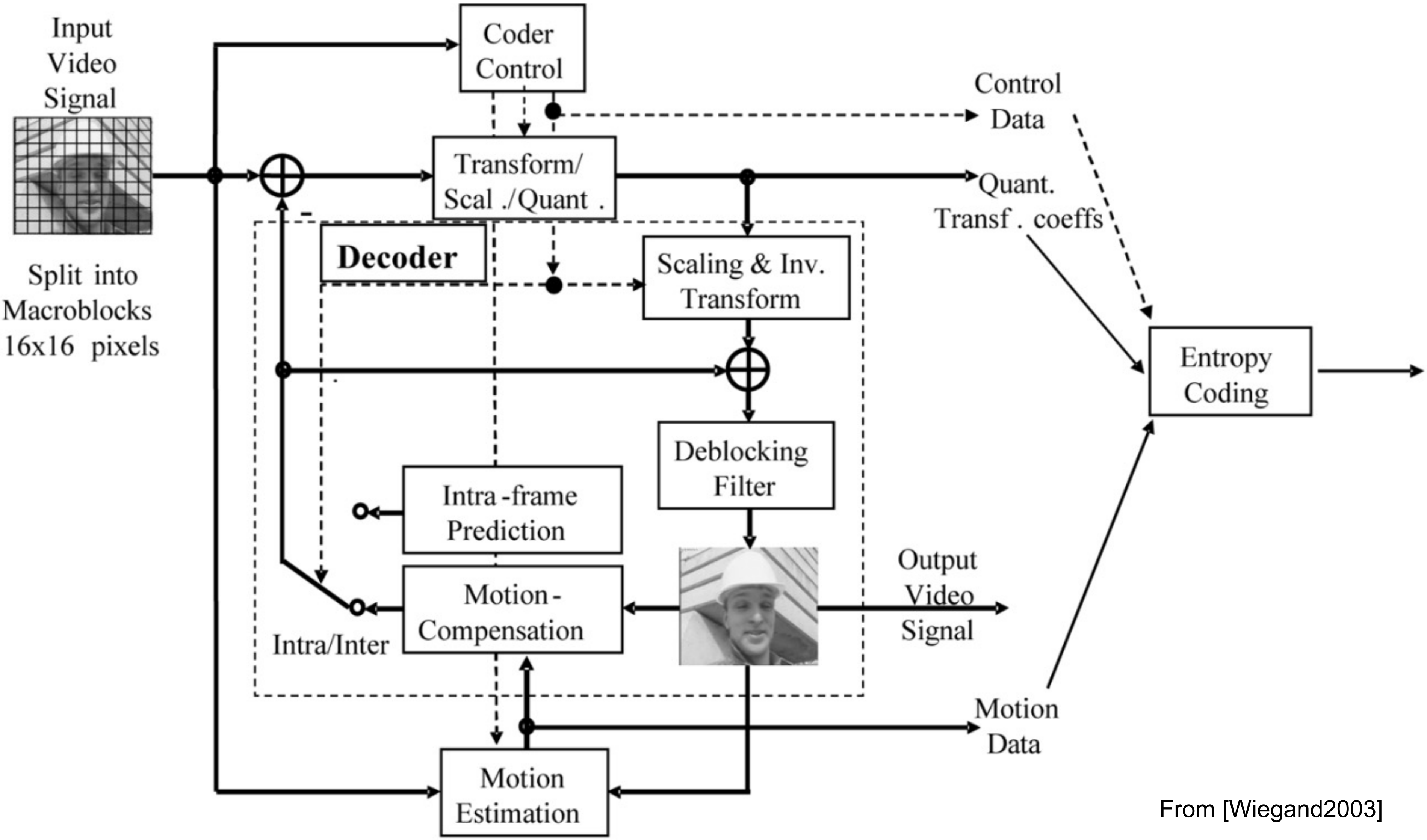
H.264/AVC Standards

- Developed by the joint video team (JVT) including video coding experts from the ITU-T and the ISO MPEG
- Finalized March 2003
- Improved video coding efficiency, up to 50% over H.263++/MPEG4
  - Half the bit rate for similar quality
  - Significantly better quality for the same bit rate
- Reference & figures for this section are from
  - *Ostermann et al., Video coding with H.264/AVC: Tools, performance, and complexity, IEEE Circuits and Systems Magazine, First Quarter, 2004*

Key Idea in All Video Coding Standards

- Divide a frame into non-overlapping blocks
- Predict each block using different modes (intra-, unidirection-inter, bidirectional-inter)
- Choose the best prediction mode (the one leading to least prediction error or best rate-distortion trade-off)
- Quantize and code prediction error using transform coding
- Code (losslessly) the mode and motion info
- Hybrid coding: predictive coding+transform coding

Block Diagram of H.264 Encoder



From [Wiegand2003]

High Efficiency Video Coding (HEVC)

- Targeting for high resolution videos: HD (1920x1080) to ultra HD (7680x4320), progressive only (60p)
- Two targeted applications
  - Random access
  - Low delay
- Two categories of profile
  - High efficiency (HE)
  - Low complexity (LC)
- Performance: 2x better video compression performance compared to H.264/AVC.
  - Half the bit rate for similar quality

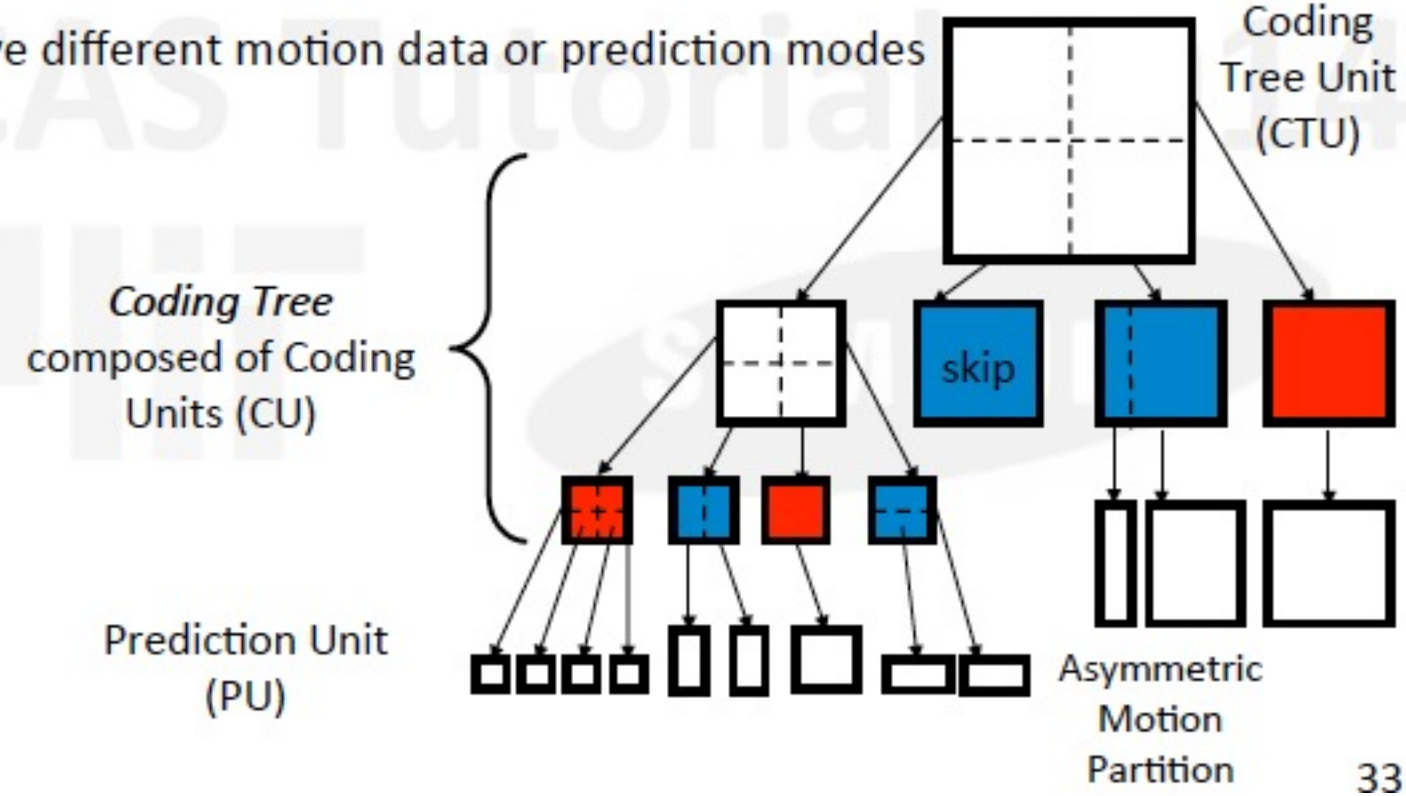
- Committee draft: Feb 2012.
- Standardization: Early 2013

New Coding Tools in HEVC

- Quadtree partition in 64x64 blocks: Block sizes from 8x8 to 64x64
- Up to 34 directions for intra-prediction
- For sub-pel motion estimation (down to ¼ pel), use 6- or 12-tap interpolation filter
- Advanced motion vector prediction
- CABAC or Low Complexity Entropy Coding
- Deblocking filter or Adaptive Loop Filter
- Extended precision options

Tree Structure for block partition

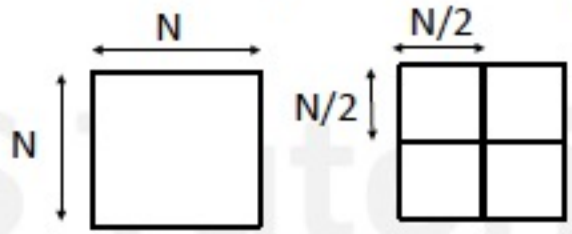
- Better adaptation to different video content
- CTU divided into Coding Units (CU) with Quad tree
- Coding units divided into prediction units (PU)
- PU have different motion data or prediction modes



From [Sze2014]

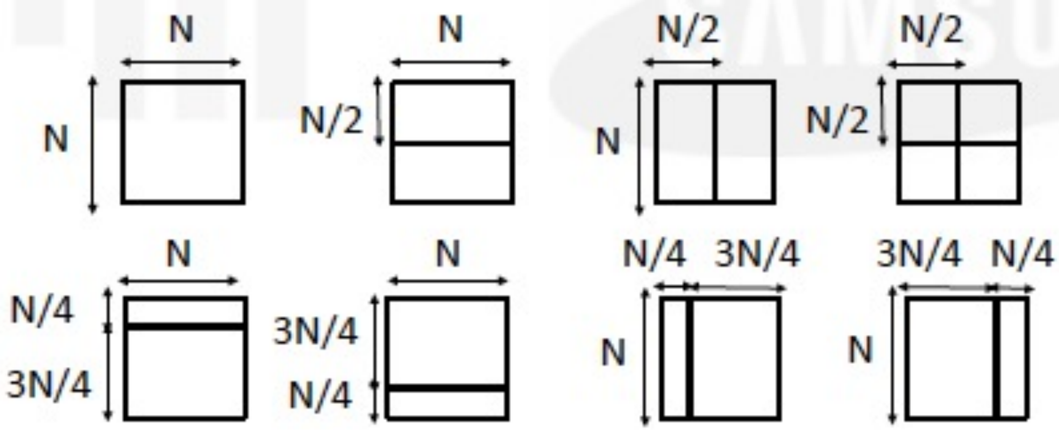
Prediction Units

- Intra-Coded CU can only be divided into square partition units
  - For a CU, make decision to split into four PU (8x8 CUs only) or single PU



Two methods of partitioning for intra-coded CU

- Inter-Coded CU can be divide into square and non-square PU as long as one side is at least 4 pixels wide (note: no 4x4 PU)



Eight methods of partitioning for inter-coded CU

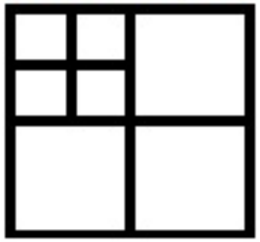
From [Sze2014]

Variable Size Transforms

Prediction residual of each coding unit may be further partitioned in a quad tree structure for transform coding

- HEVC supports 4x4, 8x8, 16x16, 32x32 integer transforms
  - Two types of 4x4 transforms (IDST-based for Intra, IDCT-based for Inter); IDCT-based transform for 8x8, 16x16, 32x32 block sizes
  - Integer transform avoids encoder-decoder mismatch and drift caused by slightly different floating point representations.
  - Parallel friendly matrix multiplication/partial butterfly implementation
  - Transform size signaled using Residual Quad Tree
- Achieves 5 to 10% increase in coding efficiency
- Increased complexity compared to H.264/AVC
  - 8x more computations per coefficient
  - 16x larger transpose memory

Represent residual of CU with TU quad tree

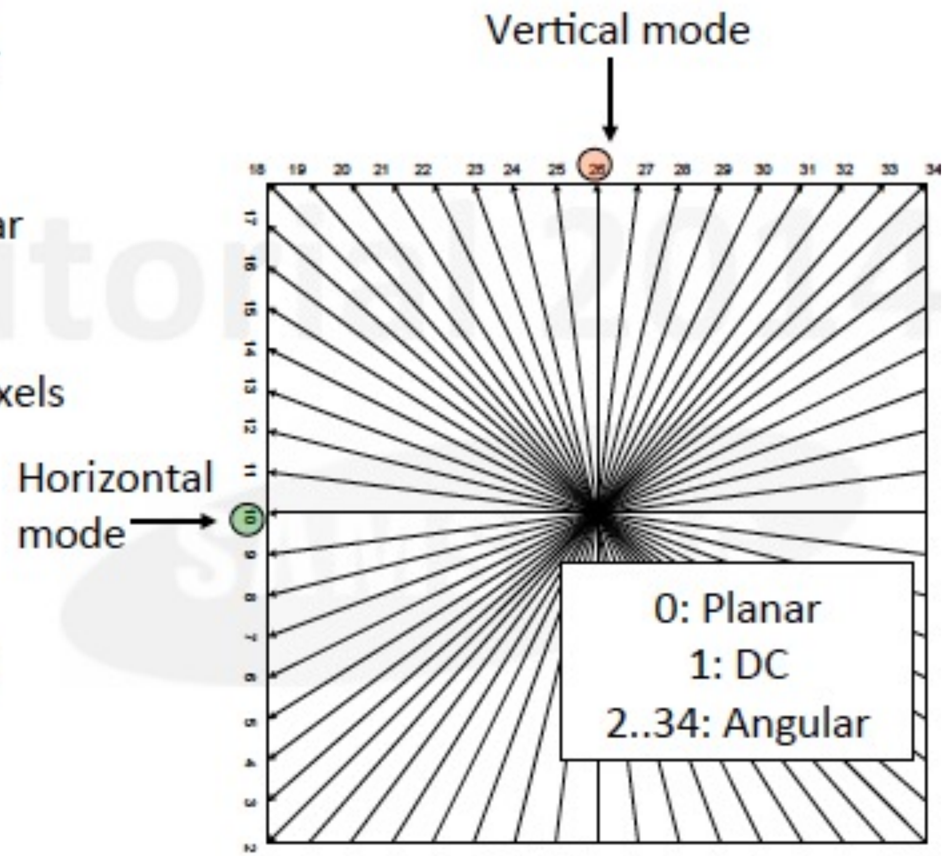


M. Budagavi et al., "Core Transform Design in the High Efficiency Video Coding (HEVC) Standard," IEEE JSTSP, 2013

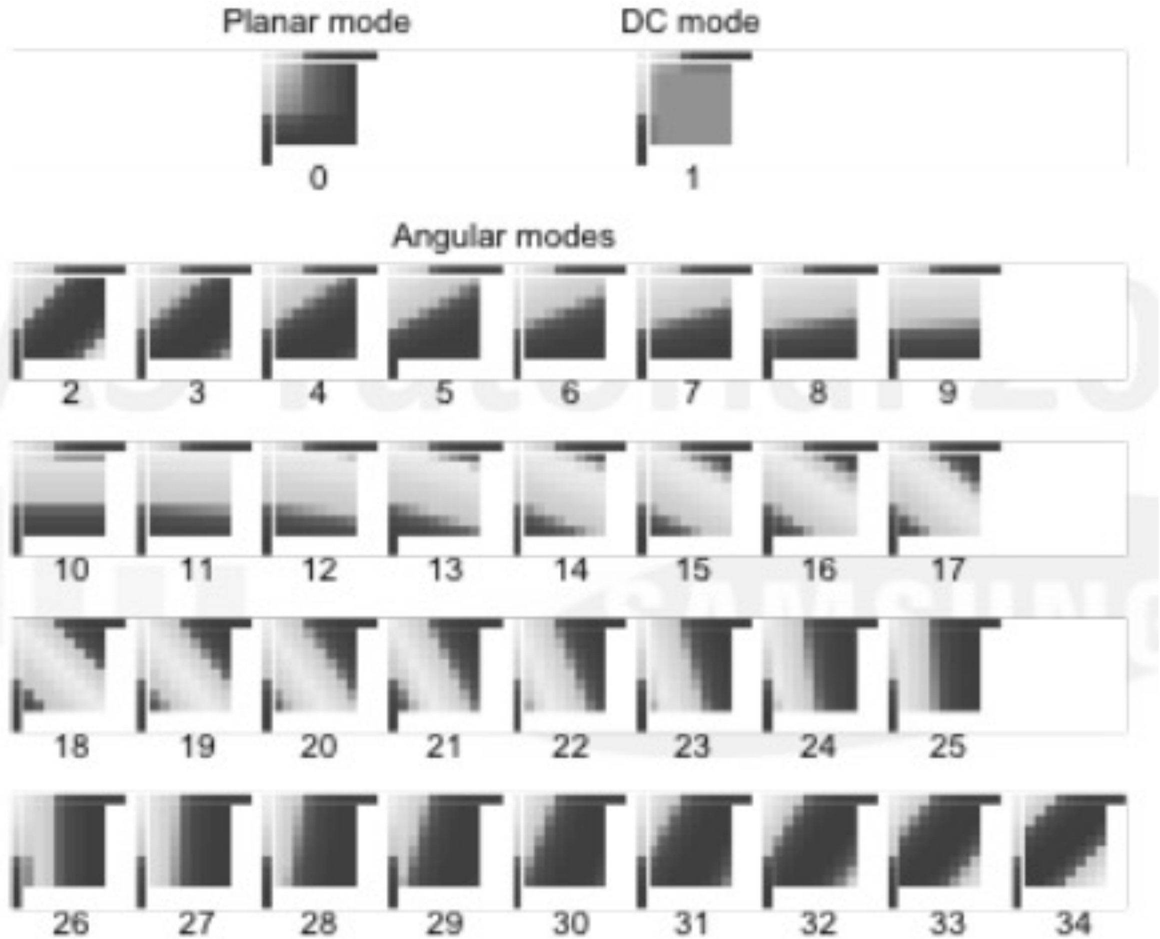
From [Sze2014]

Intra-Prediction Modes

- H.264/AVC has 10 modes
  - angular (8 modes), DC, planar
- HEVC has 35 modes
  - angular (33 modes), DC, planar
- Angular prediction
  - Interpolate from reference pixels at locations based on angle
- DC
  - Constant value which is an average of neighboring pixels (reference samples)
- Planar
  - Average of horizontal and vertical prediction



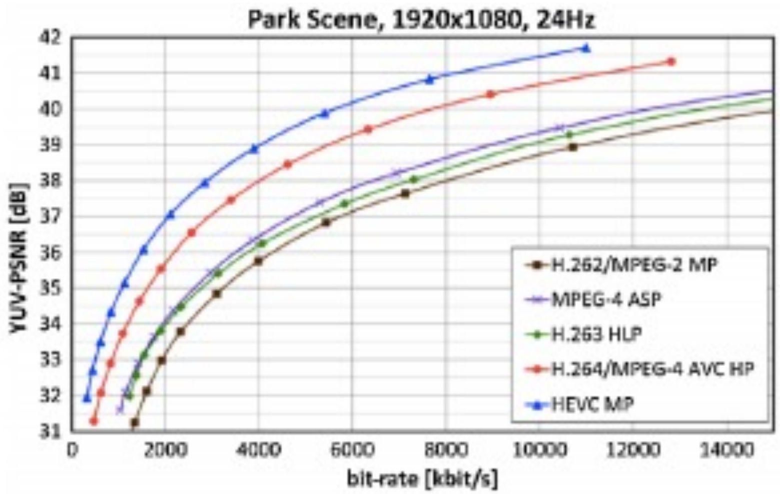
From [Sze2014]



From [Sze2014]

J. Lainema, W.-J. Han, "Intra Prediction in HEVC," ***High Efficiency Video Coding (HEVC): Algorithms and Architectures***, Springer, 2014.

Coding Efficiency Based on PSNR



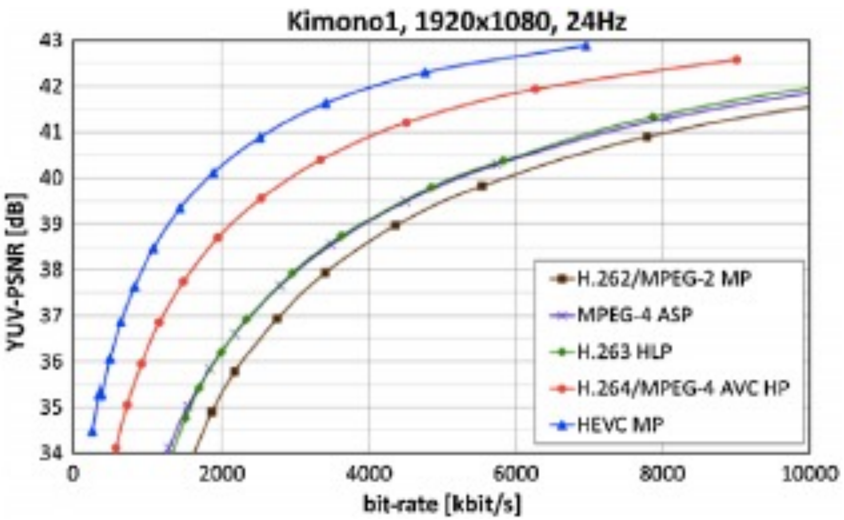


TABLE VI
AVERAGE BIT-RATE SAVINGS FOR EQUAL PSNR FOR ENTERTAINMENT APPLICATIONS

Encoding	Bit-Rate Savings Relative to			
	H.264/MPEG-4 AVC HP	MPEG-4 ASP	H.263 HLP	MPEG-2/ H.262 MP
HEVC MP	35.4%	63.7%	65.1%	70.8%
H.264/MPEG-4 AVC HP	–	44.5%	46.6%	55.4%
MPEG-4 ASP	–	–	3.9%	19.7%
H.263 HLP	–	–	–	16.2%

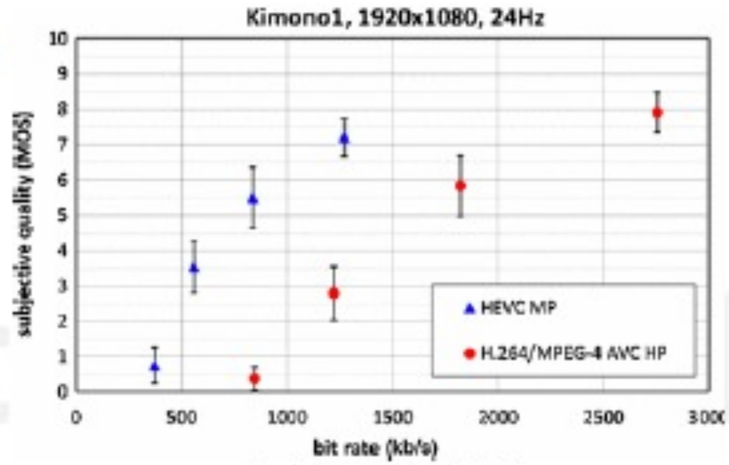
J. R. Ohm et al., "Comparison of the Coding Efficiency of Video Coding Standards—Including High Efficiency Video Coding (HEVC),"*IEEE Transactions on Circuits and Systems for Video Technology*, 2012

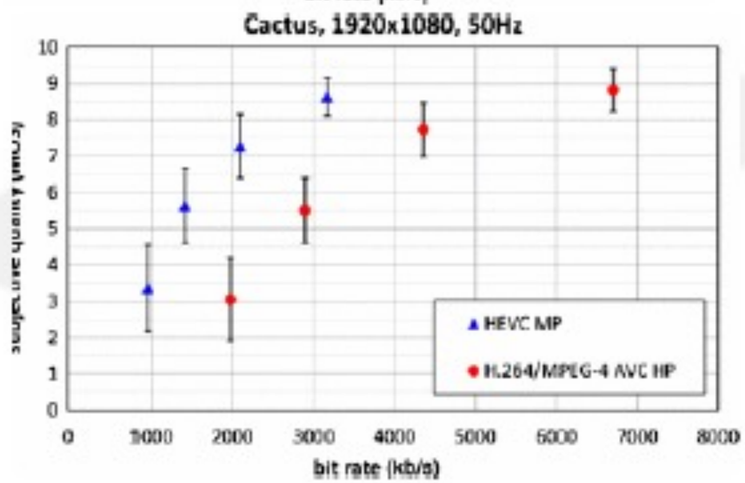
From [Sze2014]

Coding Efficiency Based on Perceptual Quality

Subjective Tests for Entertainment Applications (Random Access)

Sequences	Bit-rate Savings
BQ Terrace	63.1%
Basketball Drive	66.6%
Kimono1	55.2%
Park Scene	49.7%
Cactus	50.2%
BQ Mall	41.6%
Basketball Drill	44.9%
Party Scene	29.8%
Race Horse	42.7%
Average	**49.3%**





J. Ohm et al., "Comparison of the Coding Efficiency of Video Coding Standards—Including High Efficiency Video Coding (HEVC)," *IEEE Transactions on Circuits and Systems for Video Technology*, 2012

From [Sze2014]

Intra-Frame Coding Efficiency

- HEVC also provides improved compression for still images

	BD-Rate Reduction
H.264/AVC (intra only)	15.8%
JPEG 2000	22.6%
JPEG XR	30.0%
Web P	31.0%
JPEG	43.0%

T. Nguyen, D. Marpe, “Performance Comparison of HM 6.0 with Existing Still Image Compression Schemes Using a Test Set of Popular Still Images” JCTVC-I0595, 2012

From [Sze2014]

Outline

- Role of standards and summary of standards
- HEVC/H.265
- Other standards
- Video streaming challenges
- Scalable coding

Other Related Standards

- Other MPEG standards
  - MPEG-7
    - Standards on meta data to enable search and browsing of multimedia documents
  - MPEG-21
    - beyond MPEG-7, considering intellectual property protection, etc.
- Digital TV
  - US Grand Alliance (Using MPEG2 video)
  - European DTV (Using MPEG2 video and audio)
- Other non-international video coding standards
  - AVS1 (A Chinese video coding standard, roughly similar to H264)
  - AVS2 (roughly similar to H.265/HEVC
  - VP8 (Google's version of H264, royalty free!))
  - VP9 (Google's version of HEVC, in your web browser!)
  - VP10 …
  - Thor (CISCO)
  - Daala (Xiph)
- Kim, et al, "Coding efficiency comparison of new video coding standards: HEVC vs VP9 vs AVS2 video," *2014 IEEE International Conference on Multimedia and Expo Workshops (ICMEW)*, URL: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6890700&isnumber=6890528

More Recent Standardization Effort

- Beyond HEVC (Effort by ITU/MPEG)
  - VVC (versatile video coding), started in April 2018, committee draft July 2020
  - Add support for 360 video, point cloud, etc.
  - https://news.itu.int/versatile-video-coding-project-starts-strongly/
- AV1 (Effort by by Alliance for Open Media or AOmedia)
  - Merge different industry efforts to a new royalty free standard
  - The alliance includes all major tech/media companies including

  Amazon, Apple, ARM, Cisco, Facebook, Google, IBM, Intel, Microsoft, Mozilla, Netflix, Nvidia, Samsung Electronics and Tencent.

  AV1 competes with HEVC

  AV2 competes with VVC

  https://aomedia.org/
- VP9 and AV1 are used in YouTube and Netflix

On-Going MPEG Standardization Efforts

- Video coding for machines:
  - compression of features of popular deep networks
- AI-based video coding
- Point cloud compression

  - https://www.mpeg.org/

Outline

- Role of standards and summary of standards
- HEVC/H.265
- Other standards
- Video streaming challenges
- Scalable coding

Heterogeneity of Clients and Network Links

- Many heterogeneous clients
  - Different bandwidth requirements
  - Different decoding complexity and power constraints
  - Different screen sizes

- Heterogeneous networks
  - Different rates on different networks
    - Mobile phone
    - Corporate LAN
  - Dynamically varying rates
    - Congestion in the network
    - Distance to base station

Simulcast and Transcoding

- Simulcast
  - Compress a video into multiple versions at different rates
  - Transmit the version whose rate matches with the user's sustainable bandwidth
  - To support a range of possible clients requires compressing and saving at each possible rate
- Transcoding at a gateway/relay
  - Compress video once; transcode to a lower bit-rate based on client capability
  - Simplest scenario: decode and re-encode
  - Also possible to reduce complexity by careful design; however, it almost always involves more than VLC
  - To support a range of possible clients requires transcoding to each possible rate

Simulcast for video conferencing and streaming

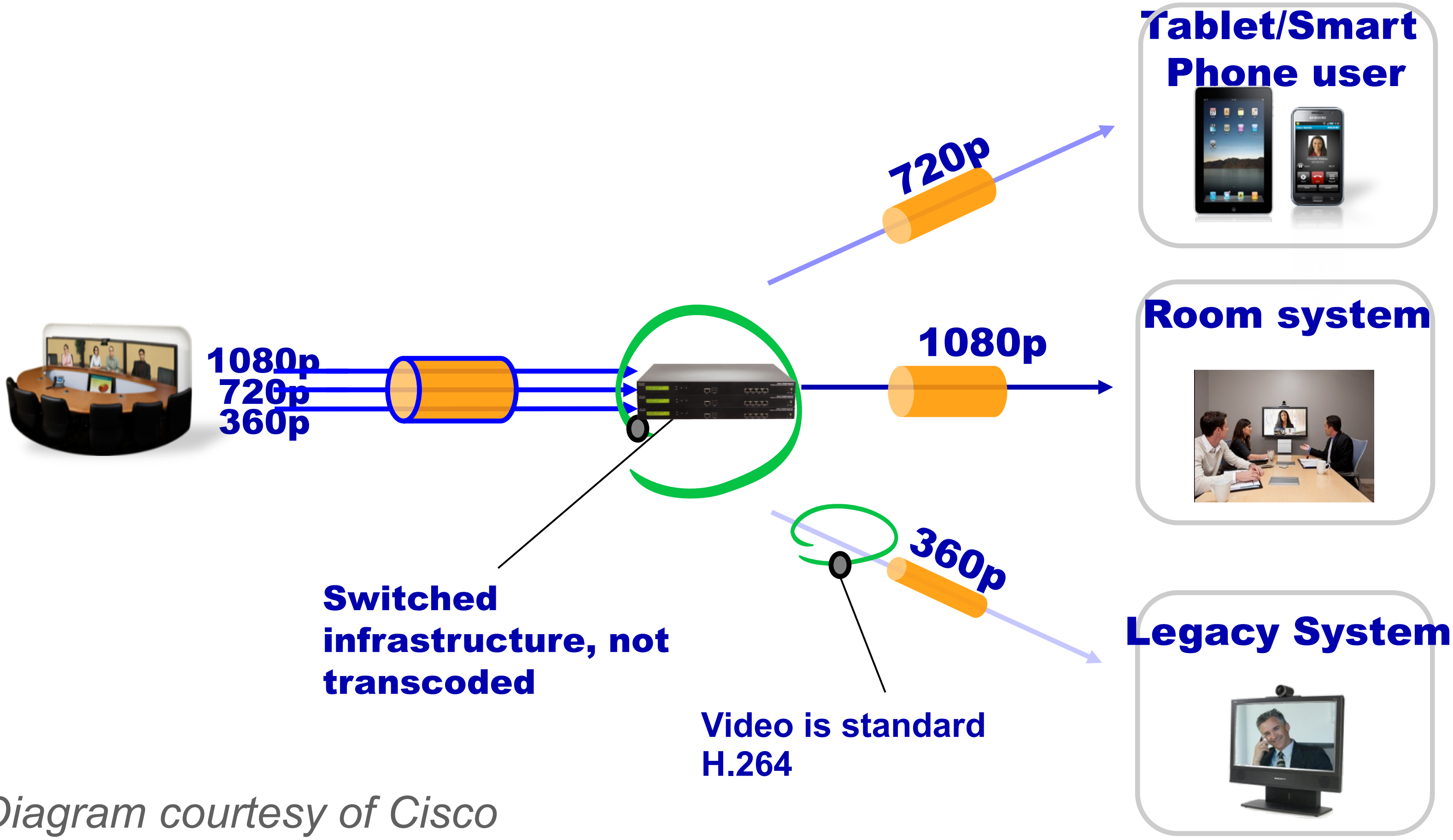


Diagram courtesy of Cisco

Note that simulcast is also used for video streaming, where the same video is coded into multiple rate /resolution versions and each client receives one particular version.

DASH: Dynamic adaptive streaming over HTTP

- Developed to accommodate temporal variation of available bandwidth at the receiver
- A video is divided into segments (1s-10s long)
- Each seg is coded into multiple representations with different rates and stored in the server
- The streaming client request the representation for the next segment based on the estimated available bandwidth for the next time duration and the current buffer status
- Widely used in today's video streaming applications: Netflix, YouTube, …
- MPEG standard for streaming

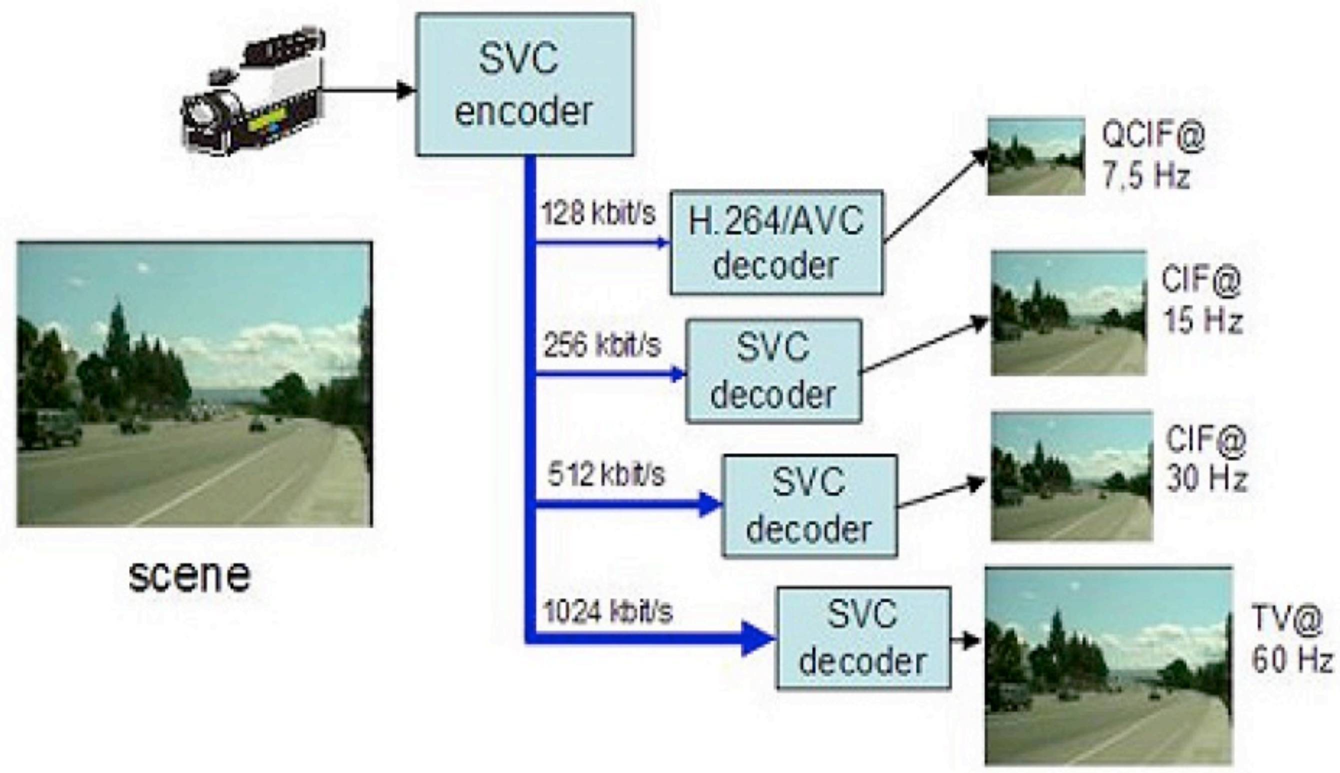
HLS (HTTP Live Streaming)

- Similar to DASH
- Developed by Apple

Outline

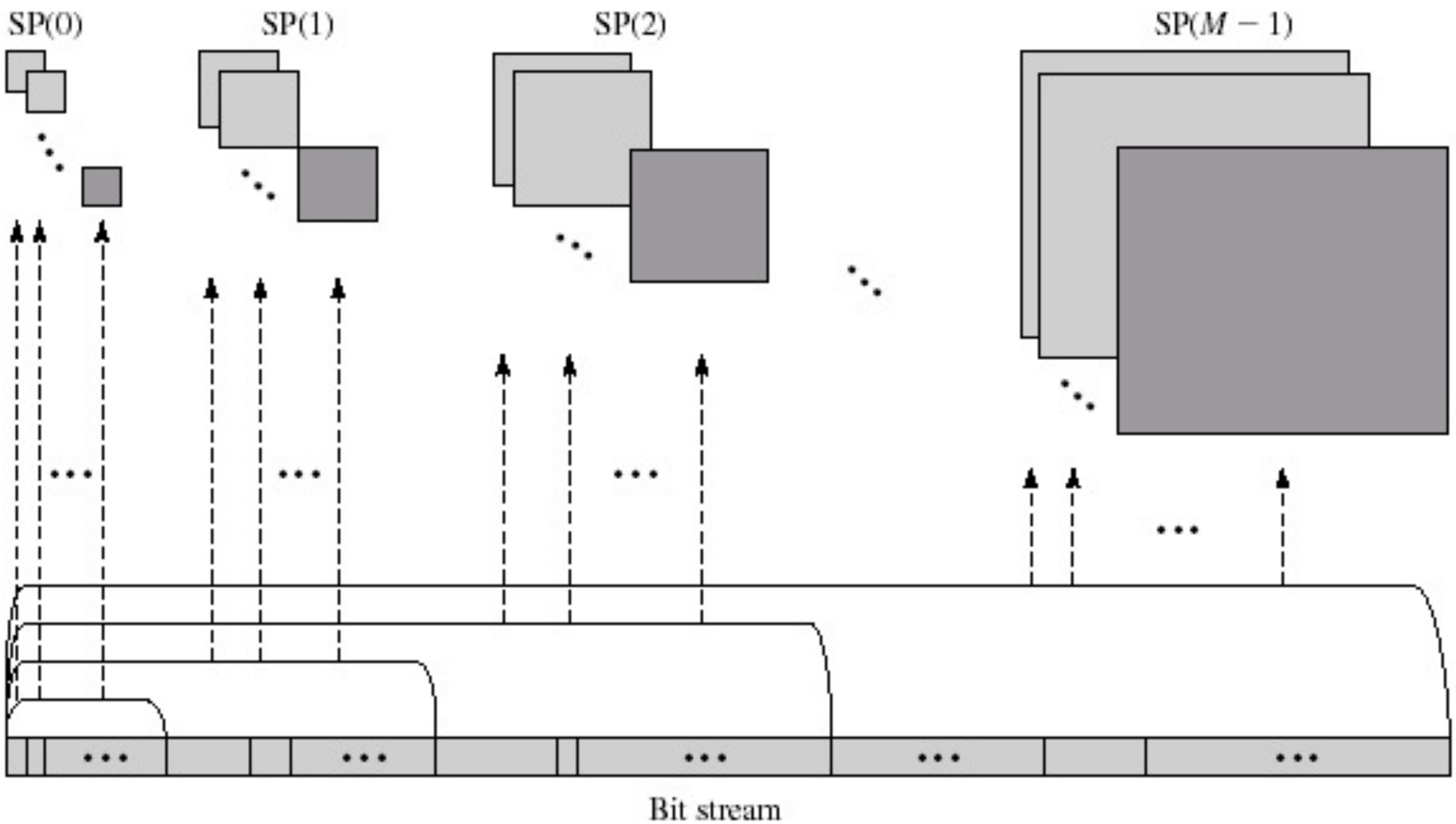
- Role of standards and summary of standards
- HEVC/H.265
- Other standards
- Video streaming challenges
- Scalable coding

Scalable Video Coding and Distribution



From Wainhouse Research, LLC.

Scalable (Embedded) Bit Stream



Functionality Provided by Scalability

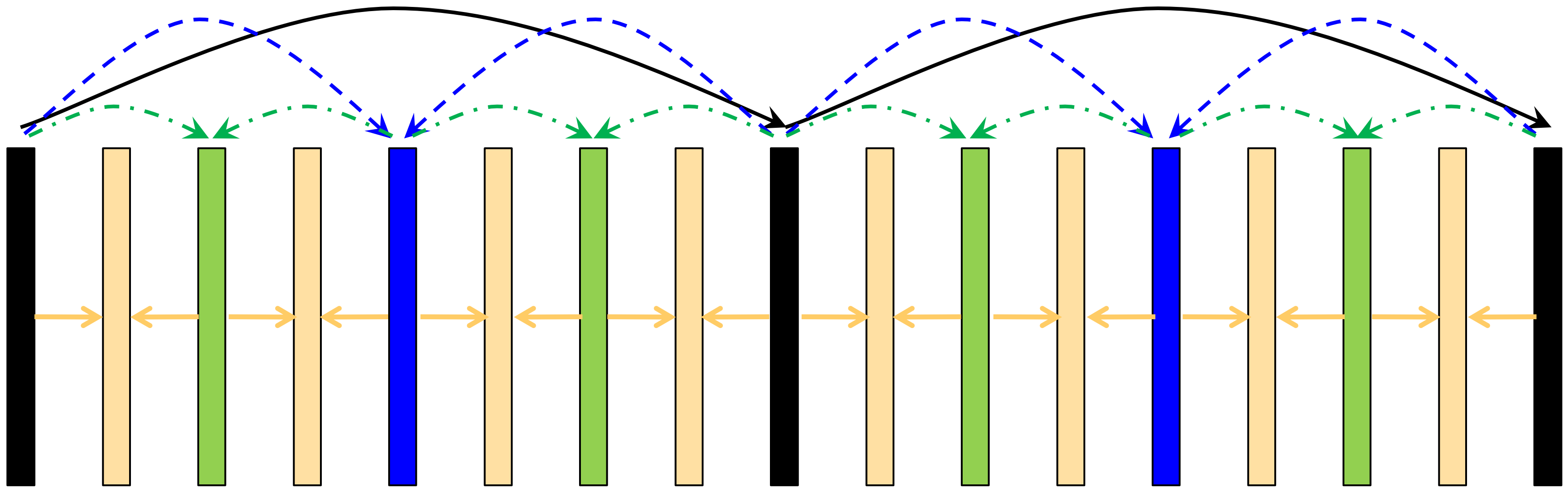
- Graceful degradation if the less important parts of the bitstream are not delivered or received or decoded (lost, discarded)
- Bit-rate adaptation at the sender or intermediate nodes to match the channel throughput
- Format adaptation for backward compatible extensions
- Power adaptation for a trade-off between decoding power consumption and quality
- Transport module can provide more protection against packet losses to lower layers (unequal error protection or UEP)
- Overall robustness to bandwidth fluctuation and packet losses

Design Considerations for Scalability

- Compression efficiency
- Encoder and decoder complexity
- Resilience to losses
- Flexible partitioning for rate adaptation
  - Range of rate partitioning (ratio of base rate to total rate)
  - Number of partitions (finely granular, or a few discrete levels)
- Compatibility with standards
- Ease of prioritization

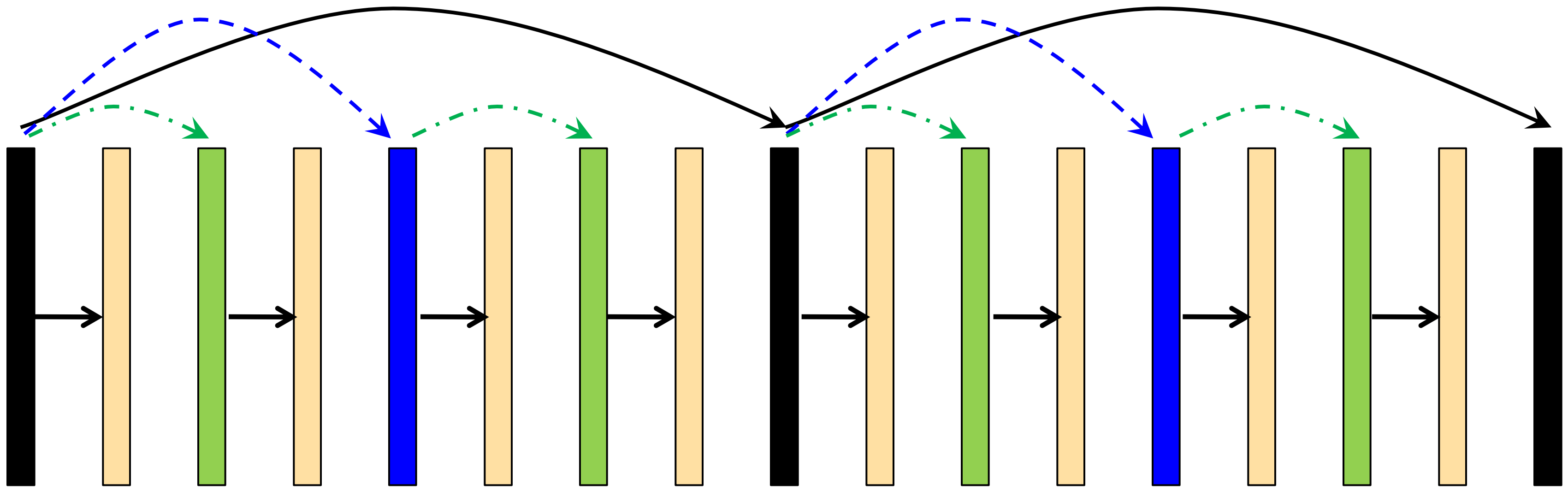
- Prediction structure controls most of these!

Temporal Scalability with Hierarchical B pictures



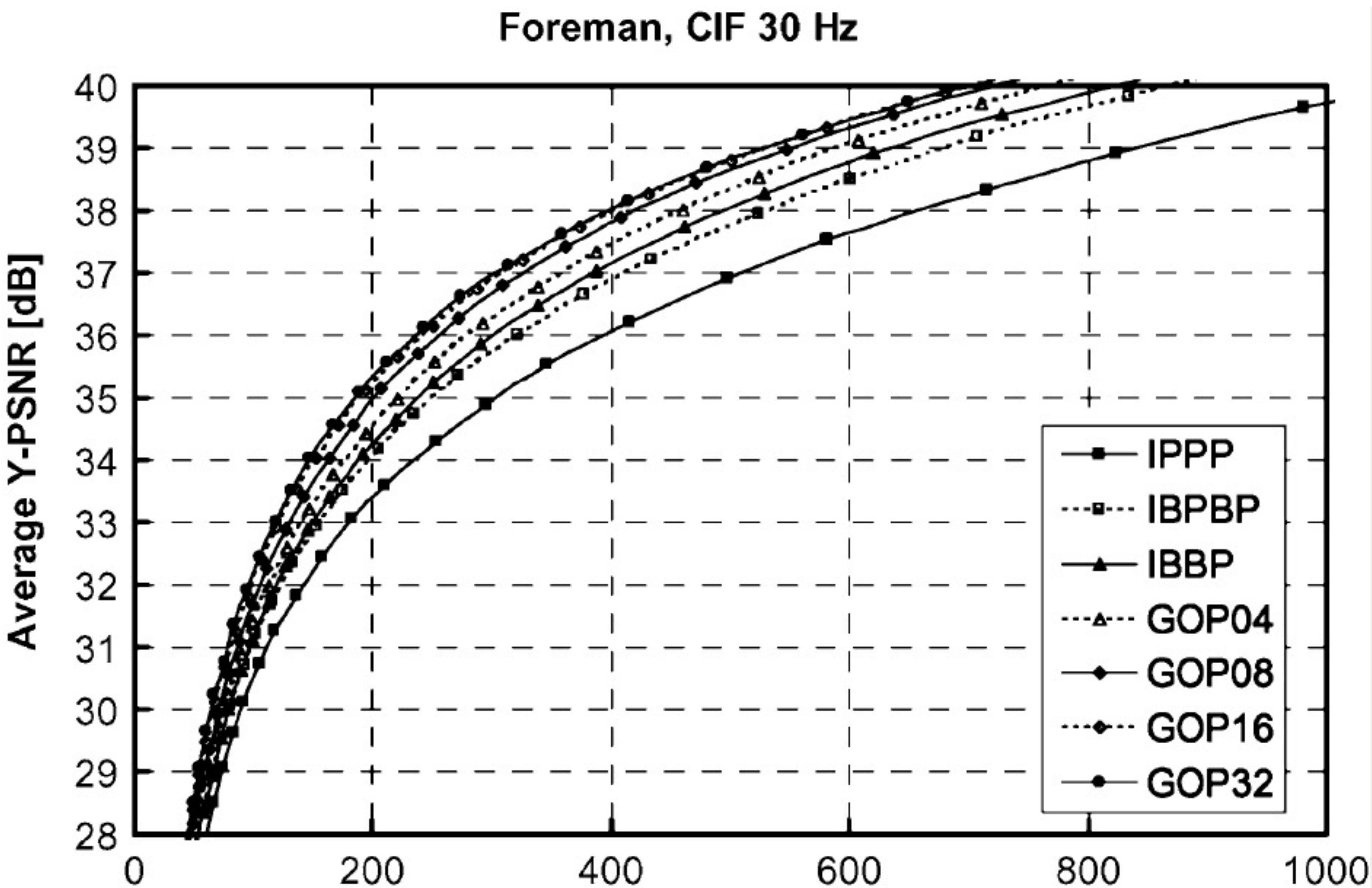
- Base layer (layer 1): black frames ; layer 2: blue frames; layer 3: green frames; layer 4: yellow frames. Layer 1 only: 30/8=3.75 Hz, Layer1+2: 30/4=7.5 Hz, Layer 1+2+3: 30/2=15 Hz, All layers: 30 Hz.
- Base layer (black frames) coded as a single layer video.
- Enhancement layer (e.g. green) frames predicted from frames of lower layers (black and blue).
- Problem: encoding delay = number of frames in a GOP (between black frames)
- OK for non-realtime applications: live streaming, video-on-demand

Temporal Scalability with Hierarchical prediction and Zero delay (Hierarchical P)

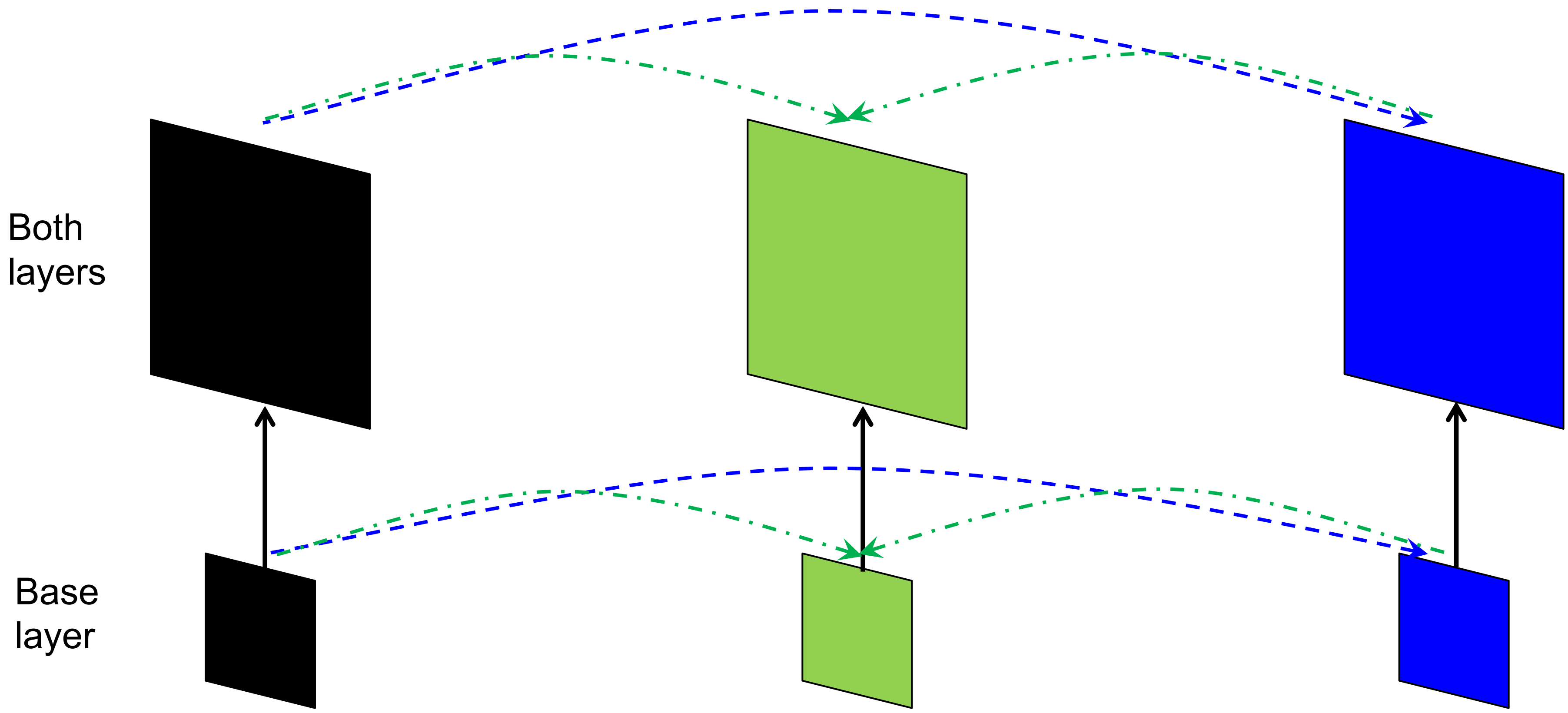


Good for realtime applications: chat or conferencing

Efficiency of H.264 Temporal Scalability (no loss in efficiency with Hierarchical-B)

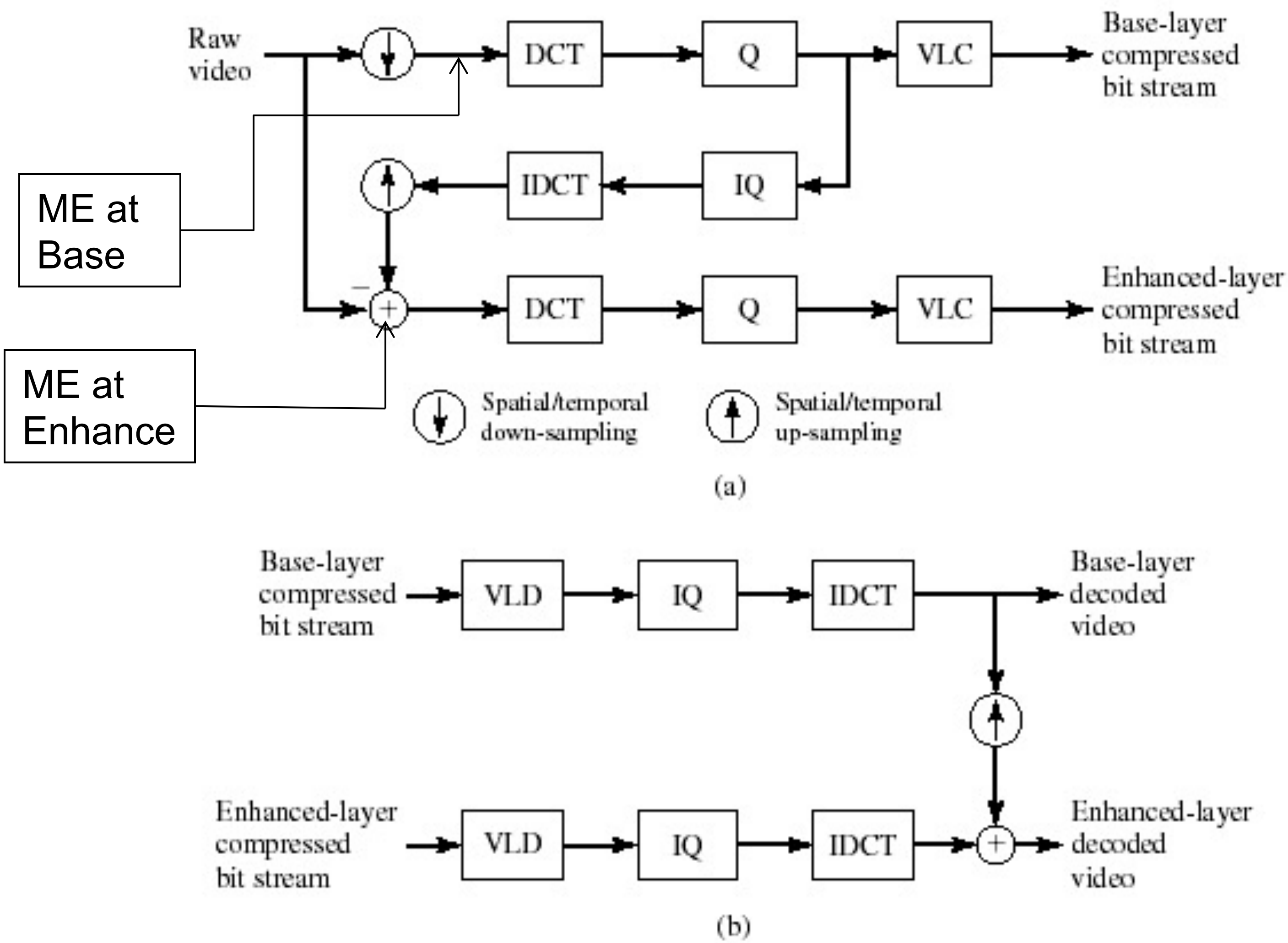


Spatial Scalability

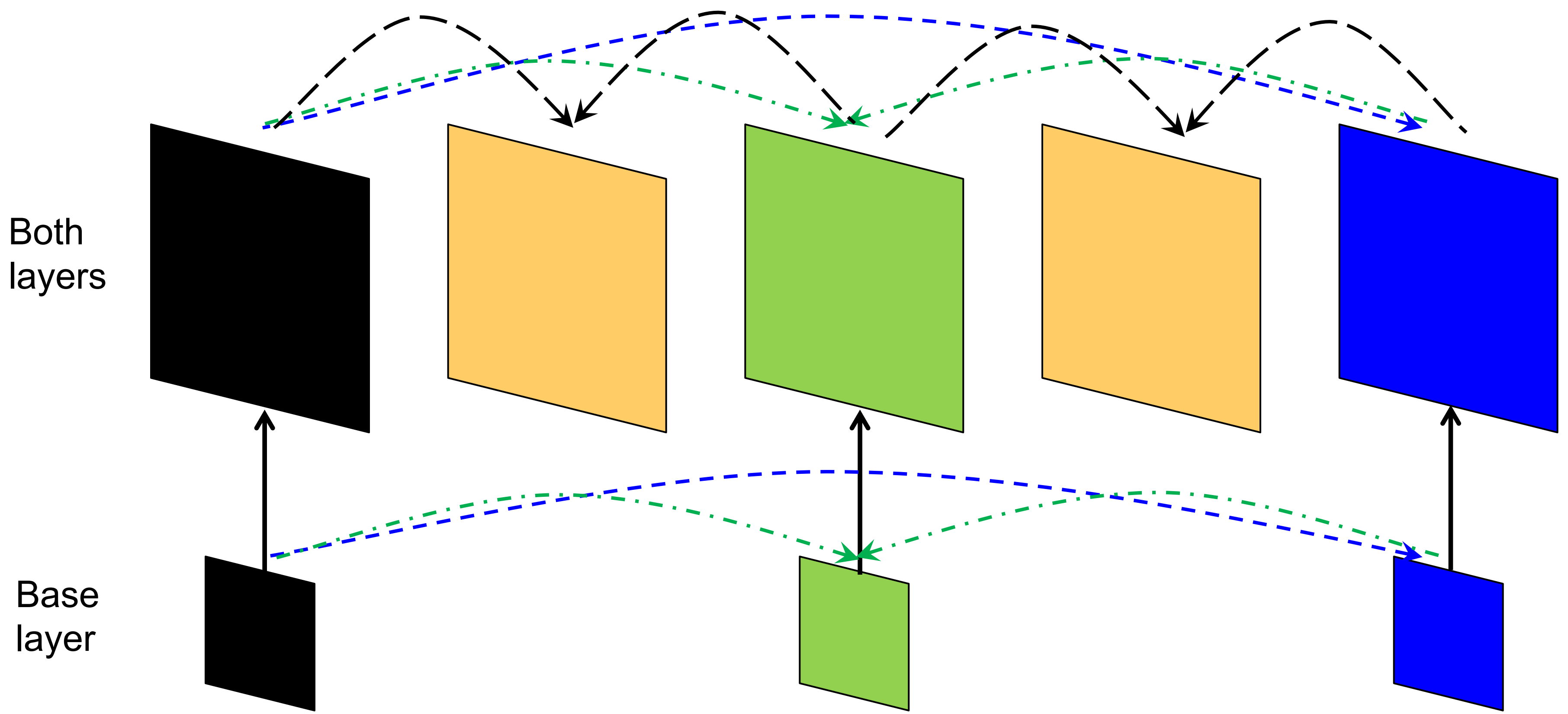


- Produce different size representations of each frame through filtering and down sampling (Gaussian pyramid of each frame)
- Base layer (smallest size) coded as a single layer video.
- Enhancement layer (larger size) frames can be predicted from other frame of the same layer, or upsampled version of the lower layer for the same frame.

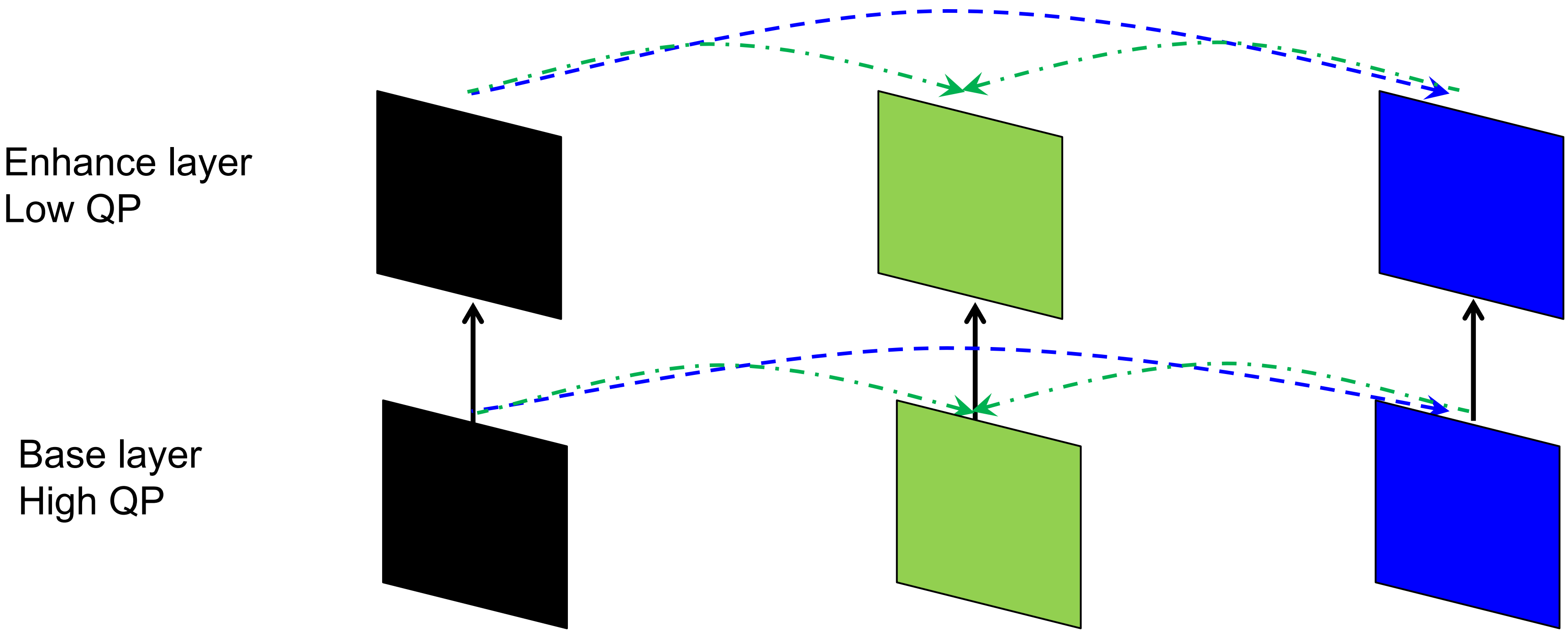
Spatial Scalability Through Down/Up Sampling



Spatial and Temporal Scalability

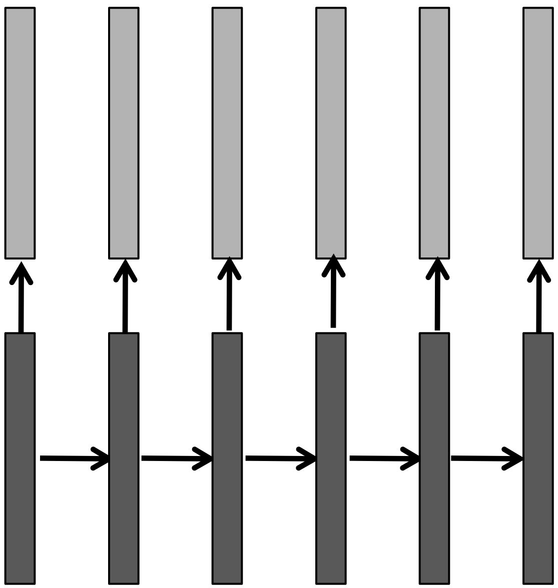


Amplitude Scalability



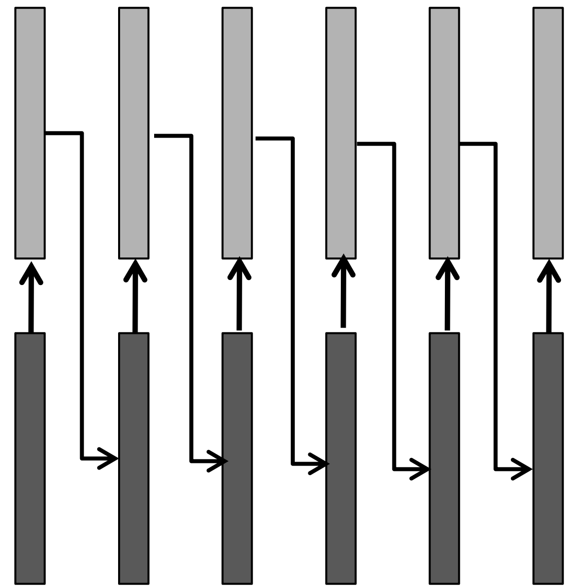
- Amplitude resolution in each layer differs because of the quantization level
- Base layer coded as a single layer video with a high QP
- Enhancement layer frames can be predicted from previous frames of the current layer or the lower layer of the current frame

Prediction structures for spatial/amplitude scalability (Options 1 and 2)



Enhancement layer is predicted only from same frame in base layer

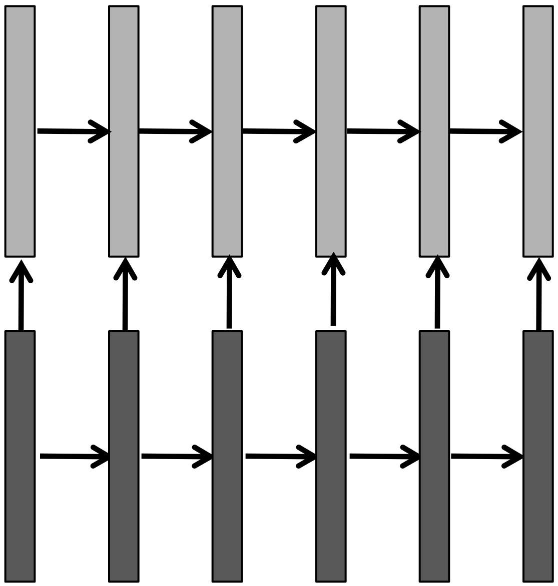
MPEG-2 Spatial Scalability (1)
MPEG-4 FGS
VERY INEFFICIENT!!
No drift in base layer



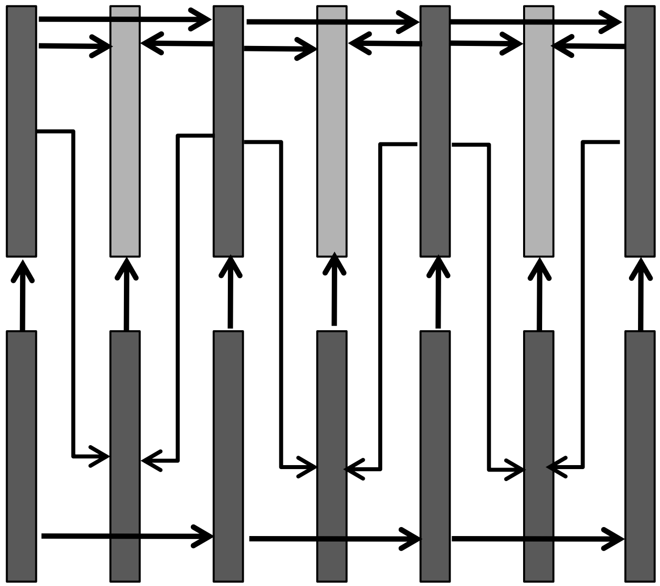
Enhancement layer is used to predict base layer

MPEG-2 SNR scalability
Errors propagate into base layer
More efficient

Prediction structures for spatial/amplitude scalability (Options 3 and 4)



H.264 CGS (coarse grain scalability):
Base: predict from base only
Enhancement: predict from base and enhancement
No drift in base layer
reasonably efficient



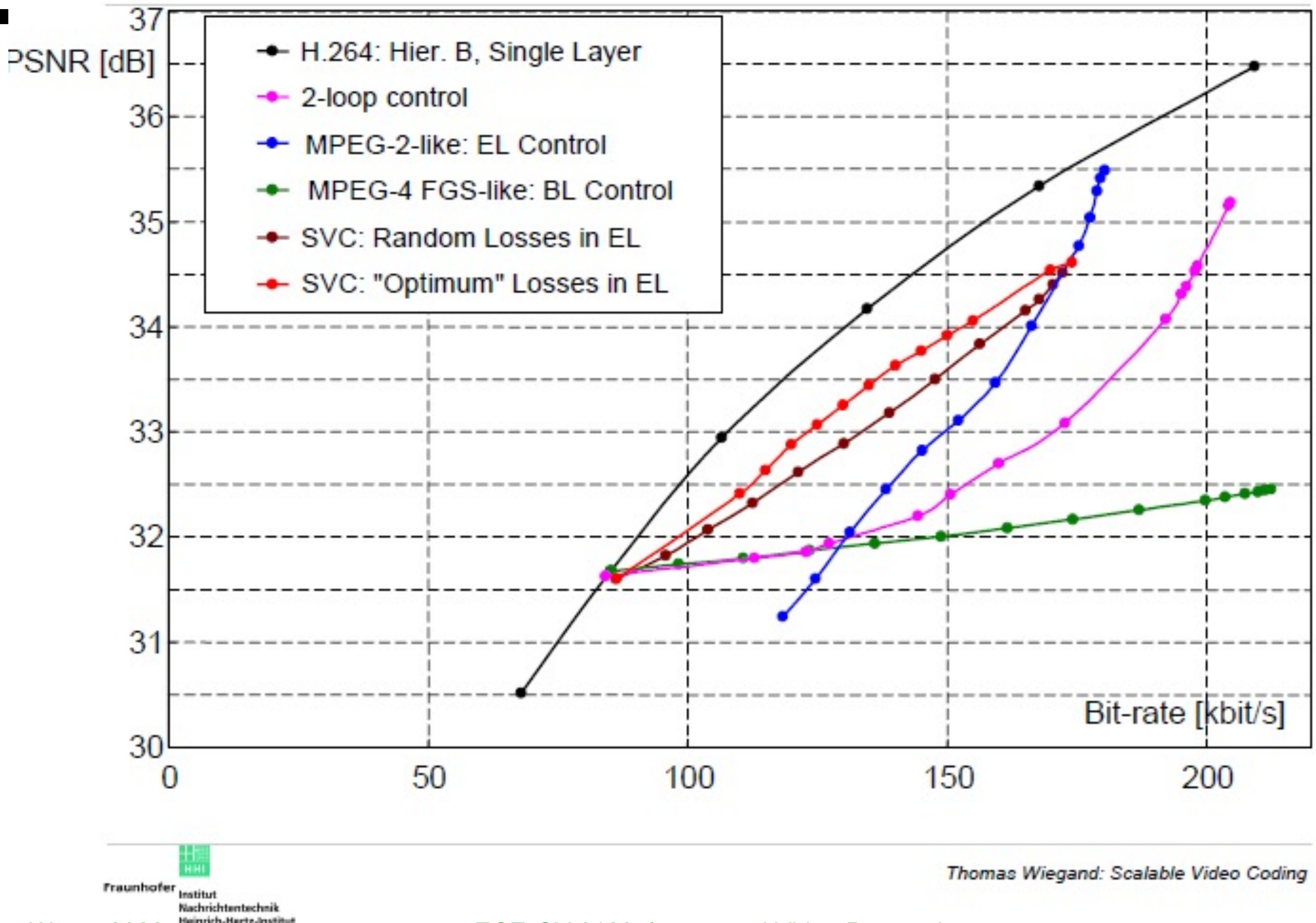
H.264 MGS (medium grain scalability):
Base: non-key frames predict using enhancement; key frames from base layer key frames
Enhancement: predict from base and enhancement
Tradeoff between efficiency and robustness

Options 3 &4: Allow both intra-layer and inter-layer prediction

- Inter-layer prediction
  - Predict from the same frame of the lower layer (higher Q), quantize the error using lower Q
- Intra-layer prediction
  - Predict from previous frame (or previous blocks of the current frame) of the current layer (lower Q), quantize the error using the same lower Q

- Choose which ever is better in RD sense (H.264/SVC quality scalability)

SNR scalability: with H.264 SVC



Scalable Video Coding Using Wavelet Transforms

- Wavelet-based image coding:
  - Full frame image transform (as opposed to block-based transform)
  - Bit plane coding of the transform coefficients can lead to embedded bitstreams
  - EZW → SPIHT → JPEG2000
- Wavelet-based video coding
  - Temporal filtering with and without motion compensation
    - Using MC limits the range of scalability
  - Can achieve temporal, spatial, and quality scalability simultaneously
  - So far has not outperformed block-based approach!
- Ref:

  - Luo, Lin, Jin Li, Shipeng Li, Zhenquan Zhuang, and Ya-Qin Zhang. "Motion compensated lifting wavelet and its application in video coding." In *IEEE International Conference on Multimedia and Expo, 2001. ICME 2001.*, pp. 93-93. IEEE Computer Society, 2001.
  - Hsiang, Shih-Ta, and John W. Woods. "Embedded video coding using invertible motion compensated 3-D subband/wavelet filter bank." *Signal Processing: Image Communication* 16, no. 8 (2001): 705-724.
  - Flierl, Markus, and Bernd Girod. "Video coding with motion-compensated lifted wavelet transforms." *Signal Processing: Image Communication* 19, no. 7 (2004): 561-575.

Reading Assignments

- [Wang2002] Chap. 13 (standards), Chap 11.1 (scalable coding)
- HEVC
  - G. J., Sullivan, J.-R. Ohm, W.-J. Han, T. Wiegand,, “Overview of the High Efficiency Video Coding (HEVC) Standard,” IEEE Trans. Circuits and Systems for Video Technology, Special Section on the Joint Call for Proposals on High Efficiency Video Coding (HEVC) Standardization. Dec. 2012
  - V. Sze and M. Budagavi, HEVC tutorial at ISCAS2014: http://www.rle.mit.edu/eems/wp-content/uploads/2014/06/H.265-HEVC-Tutorial-2014-ISCAS.pdf (include information on software and hardware implementation)

Recommended Readings

- H264/SVC:
  - H. Schwarz, D. Marpe, T. Wiegand, "Overview of the Scalable Video Coding Extension of the H.264/AVC Standard", IEEE Trans. CSVT, September 2007
  - H. Schwarz, D. Marpe, T. Wiegand, "Overview of the Scalable Video Coding Extension of the H.264/AVC Standard", IEEE Trans. CSVT, September 2007
- AVS (http://www.avs.org.cn/english/)
  - http://vspc.ee.cuhk.edu.hk/~ele5431/AVS.pdf (King Ngan, Chinese University of Hong Kong)
- Alliance for Open Media or AOmedia
  - http://aomedia.org/

Written Assignment (1)

1. What does video coding standard specify and how does it enable interoperability and yet encourage innovations and competitions?
2. Now that you have learnt about basics of video coding, imagine that you would like to tell your friend how does it work. Write down what would you say to make it easier for them to understand. You can assume that a fixed block size is used.
3. What are the different types of scalability modes supported in SVC? Describe briefly how each mode works. Can these different modes be combined? Give an example on how would you combine two scalability, e.g. temporal and amplitude scalability.
4. Compare temporal scalability through Hierarchical B and Hierarchical P structures. What are the pros and cons of each?

Written Assignment (2)

5. Suppose that you are asked to design a video streaming server that has to serve clients with different downlink capacities. You have to choose between simulcast vs. scalable coding strategies. First describe how the system will work with each strategy. Then describe the benefit and downside of each approach in terms of computation cost, storage requirements and bandwidth utilization. To make it easier to consider, assume that the clients can be categorized into 3 groups, with low (250kbps), medium (1Mpbs), and high (2 Mbps) downlink capacities. Also assume that coding a scalable bitstream with 3 layers and with base layer at 250kbps will take 50% more computation power than generating a single layer bistream, and the redundancy of the scalable coder is roughly 30% (or 1dB loss in the decoded video PSNR). That is, the decoded video consisting of base layer and one enhancement layer (with total bit rate roughly 1Mbps) will have a PSNR that is 1dB lower than the single layer video at bit rate of 1Mbps, and similarly, the video consisting of the base layer and two enhancement layers (with a total rate of roughly 2Mbps) will have a PSNR that is 1dB lower than the single layer video at bit rate of 2Mbps. Overall, based your list of pros and cons of each strategy, which approach will you recommend? How would you convince your boss that your choice is a good one?

Additional Material

- H.264/AVC in more detail
- H.265/HEVC in more detail

H.264/AVC Standards

- Developed by the joint video team (JVT) including video coding experts from the ITU-T and the ISO MPEG
- Finalized March 2003
- Improved video coding efficiency, up to 50% over H.263++/MPEG4
  - Half the bit rate for similar quality
  - Significantly better quality for the same bit rate
- Reference & figures for this section are from
  - *Ostermann et al., Video coding with H.264/AVC: Tools, performance, and complexity, IEEE Circuits and Systems Magazine, First Quarter, 2004*

H.264/MPEG4-AVC

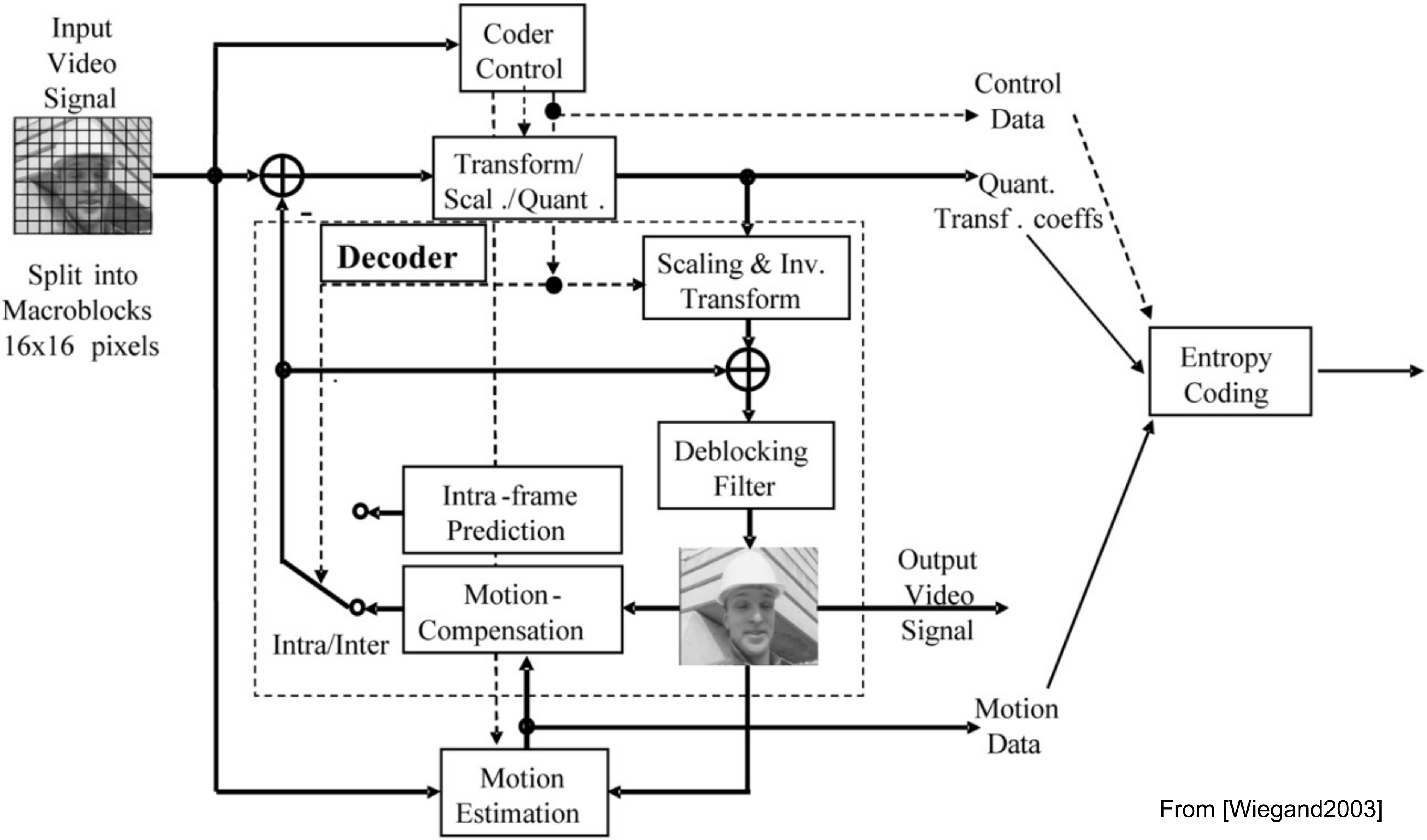
- Completed (version 1) in May 2003
- H.264/AVC is the most popular video standard in market
  - 80% of video on the internet is encoded with H.264/AVC
- Applications include
  - HDTV broadcast satellite, cable, and terrestrial
  - video content acquisition and editing
  - camcorders, security applications, Internet and mobile network video, Blu-ray Discs
  - real-time video chat, video conferencing, and telepresence
- ~50% higher coding efficiency than MPEG-2 (used in DVD, US terrestrial broadcast)

From [Sze2014]

Key Idea in All Video Coding Standards

- Divide a frame into non-overlapping blocks
- Predict each block using different modes (intra-, unidirection-inter, bidirectional-inter)
- Choose the best prediction mode (the one leading to least prediction error or best rate-distortion trade-off)
- Quantize and code prediction error using transform coding
- Code (losslessly) the mode and motion info
- Hybrid coding: predictive coding+transform coding

Block Diagram of H.264 Encoder



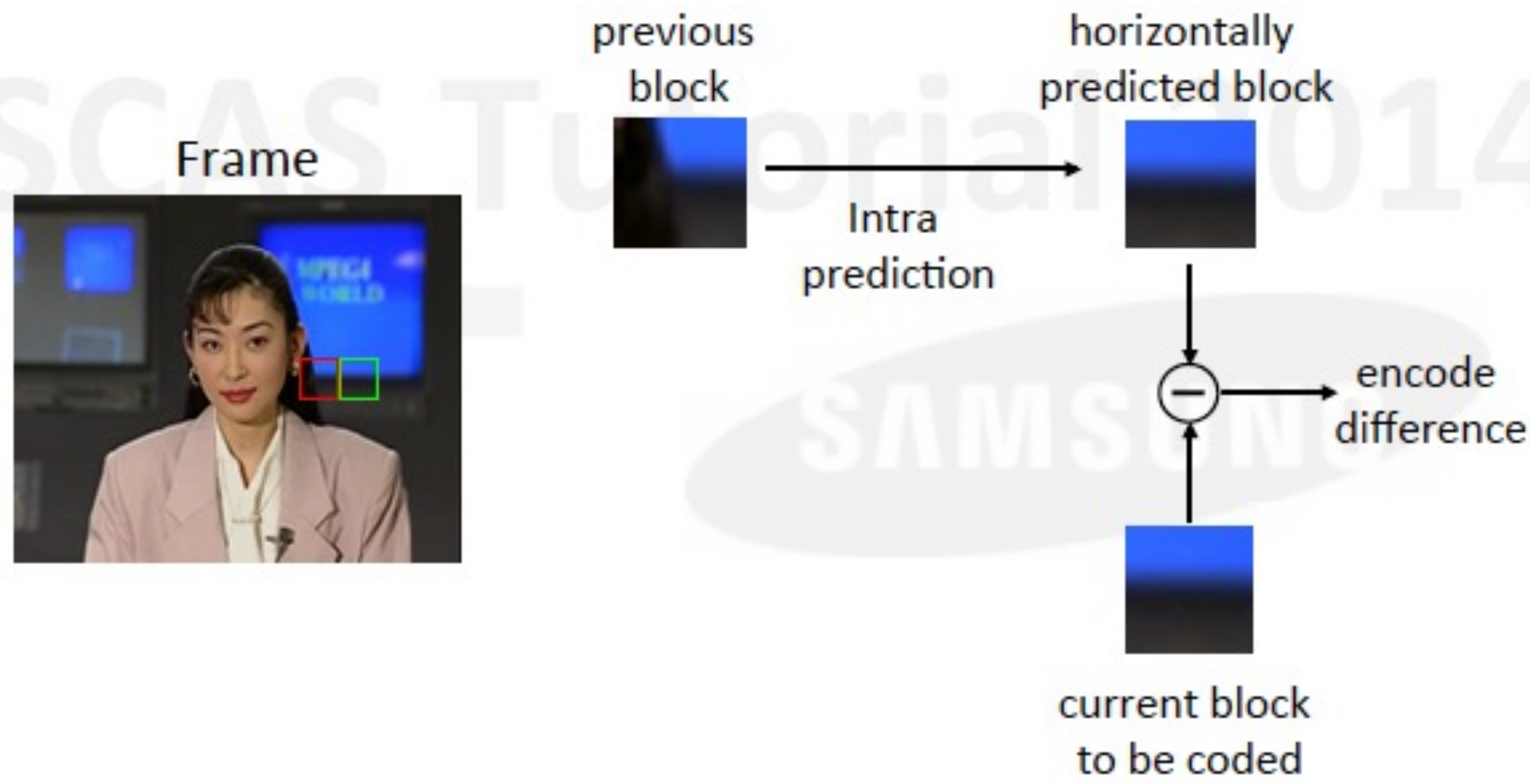
From [Wiegand2003]

New Video Coding Tools Introduced Beyond H.263/MPEG4

- Intra-prediction
- Integer DCT with variable block sizes
- Adaptive deblocking filtering
- Multiple reference frame prediction

Spatial prediction

- H.261
  - Motion vector prediction using previously encoded MV
- MPEG-1
  - DC coefficients coded predictively
- H.263
  - MV prediction using the median of three neighbors
  - Optional: Intra DC prediction (10-15% improvement)
- MPEG-4
  - DC prediction: can predict DC coefficient from *either* the previous block or the block above
  - AC prediction: can predict one column/row of AC coefficients from *either* the previous block or the block above
- H.264
  - Pixel domain directional intra prediction



From [Sze2014]

H.264 Intra prediction

- Instead of the simple DC coefficient prediction to exploit the correlation between nearby pixels in the same frame, more sophisticated spatial prediction is used
- Apply prediction to the entire 16*16 block (INTRA_16x16), or apply prediction separately to sixteen 4*4 blocks (INTRA_4x4)
- Adaptive directional prediction

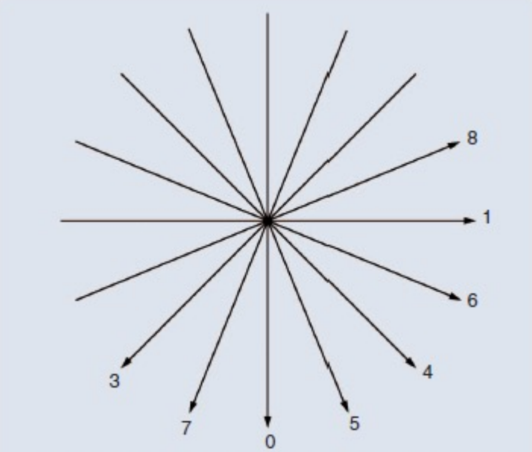


Figure 7. Possible prediction directions for INTRA_4×4 mode.

8 possible directions

Sample Intra Prediction Modes

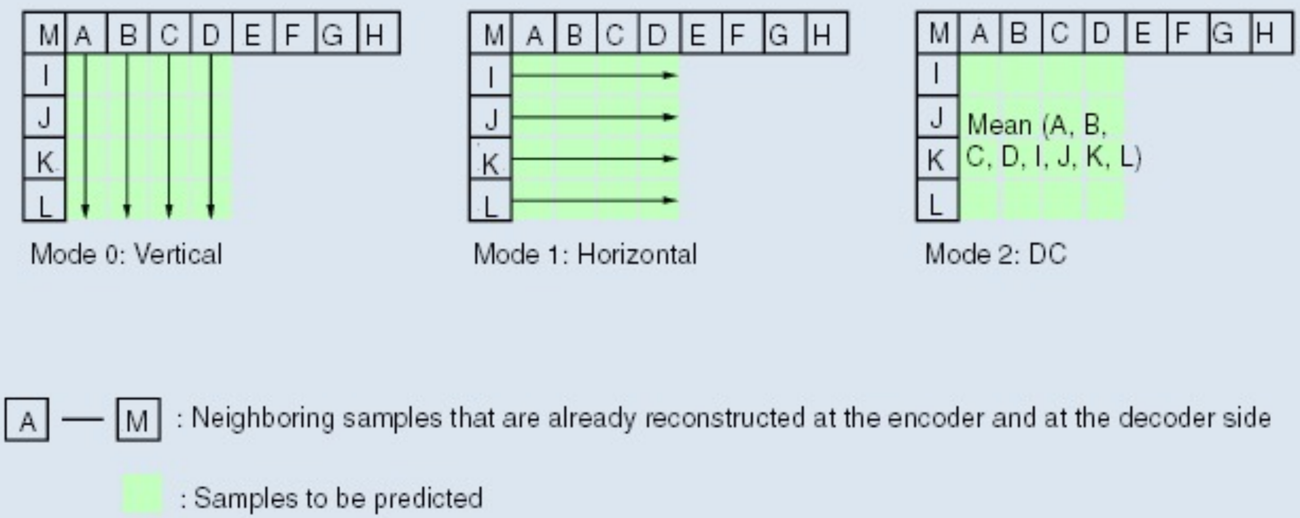


Figure 6. Three out of nine possible intra prediction modes for the intra prediction type INTRA_4×4.

From [Ostermann04]

Motion Compensation

- Quarter-pel accuracy
- Variable block size
- Multiple reference frames
  - Generalized B-picture
- Weighted prediction (fade in, fade out, etc)

Variable Blocksize Motion Compensation

- Use variable size block-based motion compensation
  - 16x16, 16x8, 8x16, 8x8, 8x4, 4x8, 4x4
  - H.263/MPEG4 use only 16x16 and 8x8

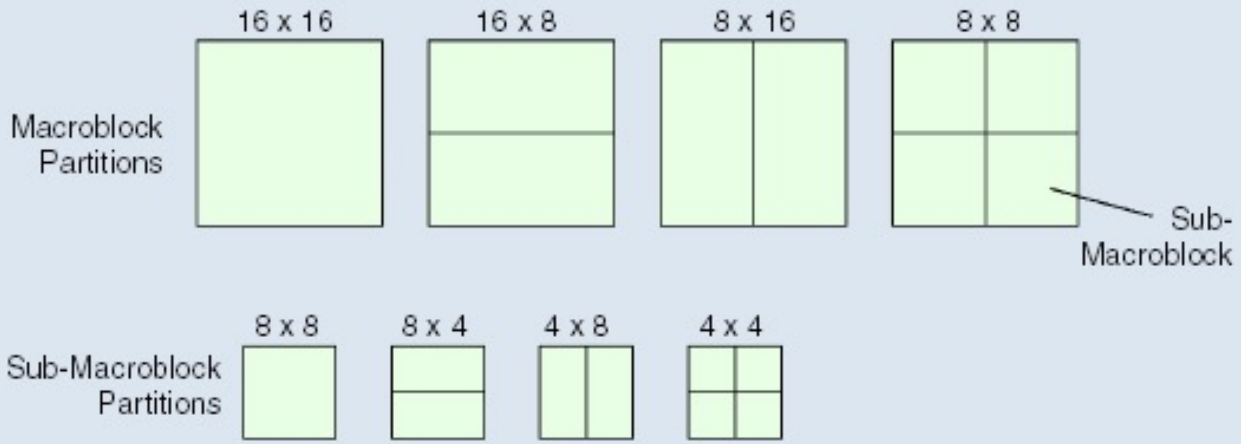


Figure 8. Partitioning of a macroblock and a sub-macroblock for motion compensated prediction.

From [Ostermann04]

Multiple Reference Frames for Motion Compensation

- Can use one or two from several possible reference frames
- When two reference frames are used, arbitrary weights can be used to combine them – Generalized B-picture

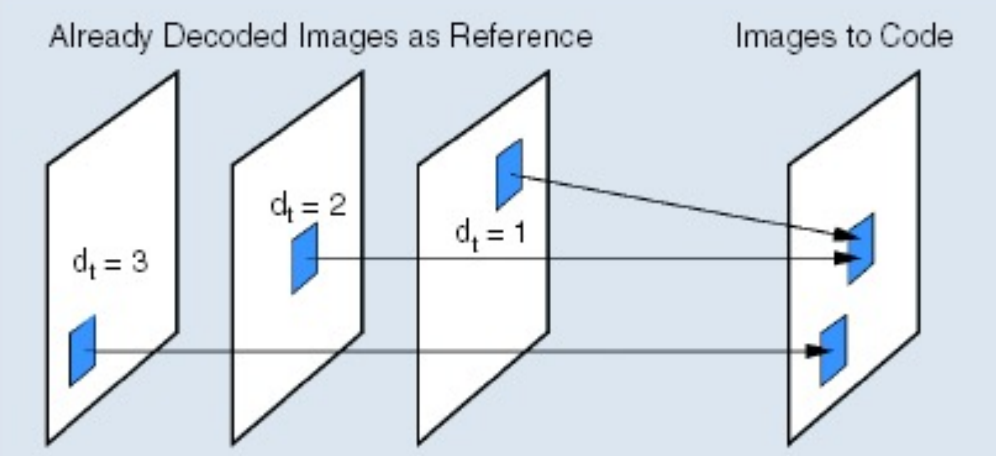
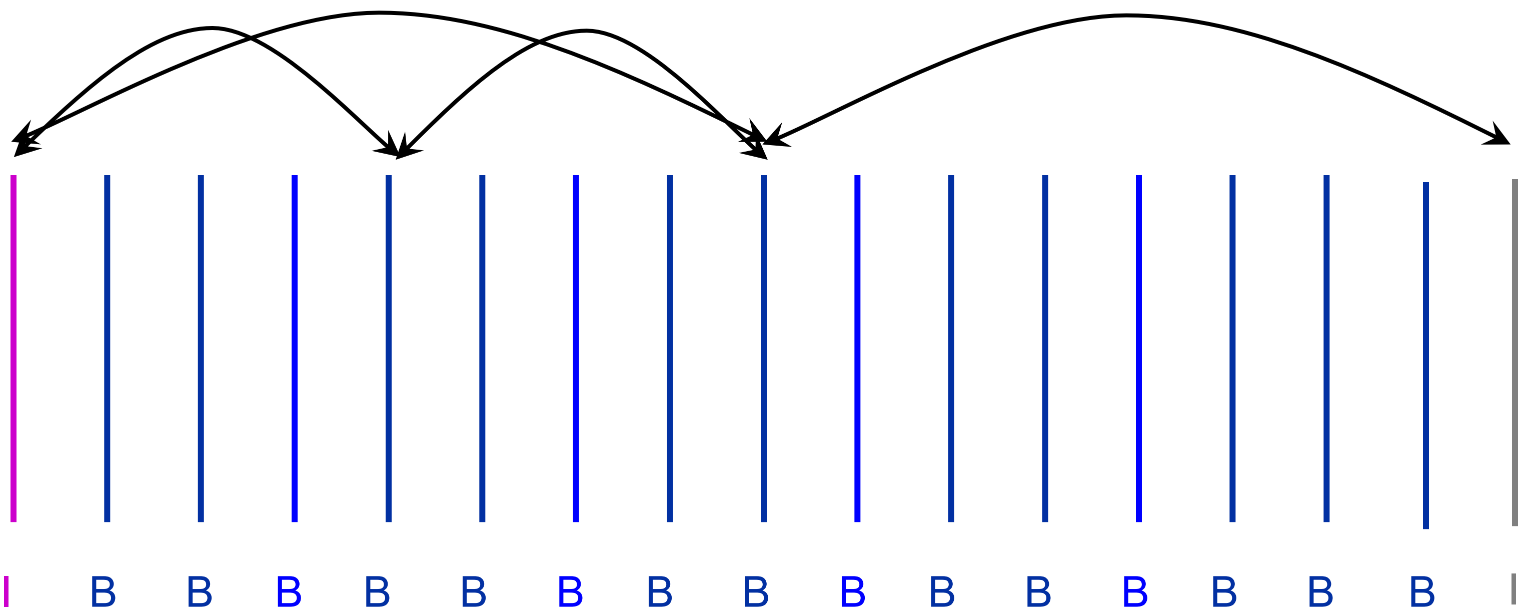


Figure 9. Motion-compensated prediction with multiple reference images. In addition to the motion vector, also an image reference parameter d_t is transmitted.

From [Ostermann04]

Generalized B-frames

In H.264, B frames can be used for prediction



Display order: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
16
Encoding order: 0 16 8 4 2 1 3 6 5 7 12 10 9 11 14 13
15

Transform

- 8x8 DCT
  - H.261
  - MPEG-1
  - H.263
  - MPEG-2
  - MPEG-4
  - DCT is non-integer; the result depends on the implementation details
- H.264:
  - Integer transforms, variable size (2x2, 4x4, 16x16)

Integer Transform

- Smaller block size (4x4 or 2x2) can better represent boundaries of moving objects, and match prediction errors generated by smaller block size motion compensation
- Integer transform can be implemented more efficiently and no mismatch problem between encoder and decoder

Primary transform

$$H_1 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{bmatrix} \quad H_2 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix} \quad H_3 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

Figure 10. Matrices H_1, H_2 and H_3 of the three different transforms applied in H.264/AVC.

From [Ostermann04]

Variable Length Coding

- H.261
  - DCT coefficients are converted into runlength representations and then coded using VLC (Huffman coding for each pair of symbols)
    - Symbol: (Zero run-length, non-zero value range)
  - Other information are also coded using VLC (Huffman coding)
- H.263
  - 3-D VLC for DCT coefficients (runlength, value, EOB)
  - Syntax-based arithmetic coding (option)
    - 4% savings in bit rate for P-mode, 10% saving for I-mode, at 50% more computations
- MPEG-4
  - 3-D VLC similar to H.263

H.264 Entropy Coding

- Baseline technique: CAVLC (context adaptively switched sets of variable length Huffman codes)
- A more complex technique called CABAC: context-based adaptive binary arithmetic coding
- Both offer significant improvement over Huffman coding which uses pre-designed coding tables based on some assumed statistics

In-Loop Filter (Deblocking Filter)

- In-Loop filtering can be applied to suppress propagation of coding noise temporally
- H.261
  - Separable filter [1/4,1/2,1/4]
  - Loop filter can be turned on or off
- MPEG-1
  - No loop filter (half-pel motion compensation provides some filtering)
- H.263
  - Optional deblocking filter included in H.263+
  - Overlapped block motion effectively smoothes block boundaries
  - Decoder can choose to implement out-of-loop deblocking filter
- H.264
  - Deblocking filter adapts to the strength of the blocking artifact

H.264 Adaptive Deblocking

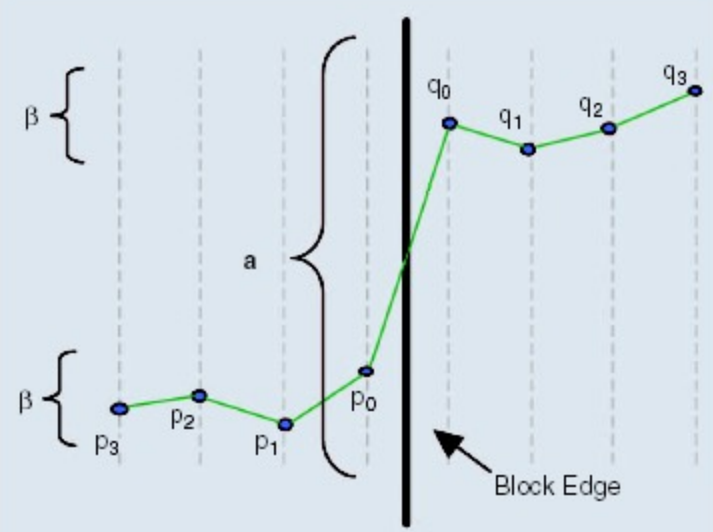


Figure 14. One-dimensional visualization of a block edge in a typical situation where the filter would be turned on.

From [Ostermann04]

- Whether filtering will be turned on depends on the pixel differences involving pixels p0,…, q0,…, and the filter depends on block characteristics and coding mode.
- Deblocking results in bit rate savings of 6-9% at medium qualities, and more remarkable subjective improvements,

Profiles and Levels

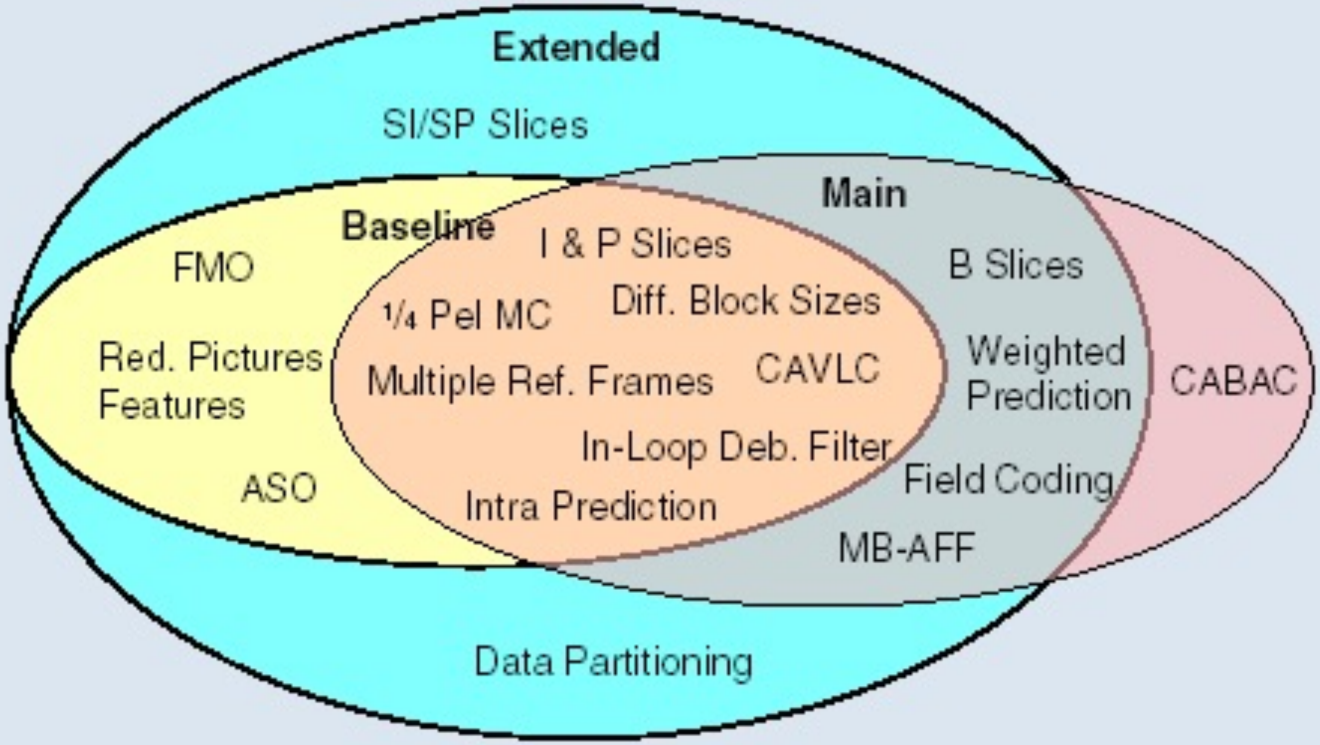


Figure 20. H.264/AVC profiles and corresponding tools.

From [Ostermann04]

Comparison with Previous Standards

- Coding efficiency: in terms of achievable rates for target video quality (PSNR)
  - Video streaming application
  - Video conferencing application

- Complexity:
  - Encoder
  - Decoder

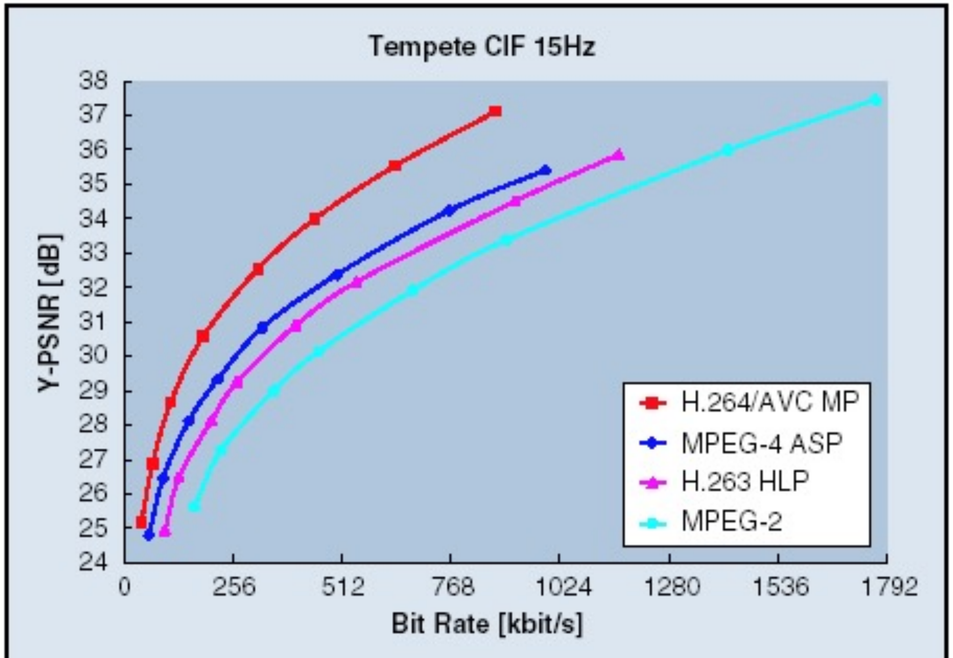


Figure 21. Luminance PSNR versus average bit rate for different coding standards, measured for the test sequence *Tempete* for video streaming applications (from [36]).

Coding efficiency for video streaming

Table 1.
Average bit rate savings for video streaming applications (from [10]).

	Average Bit Rate Savings Relative To:		
Coder	MPEG-4 ASP	H.263 HLP	MPEG-2
H.264/AVC MP	37.44%	47.58%	63.57%
MPEG-4 ASP	–	16.65%	42.95%
H.263 HLP	–	–	30.61%

From [Ostermann02]

Coding efficiency for conferencing

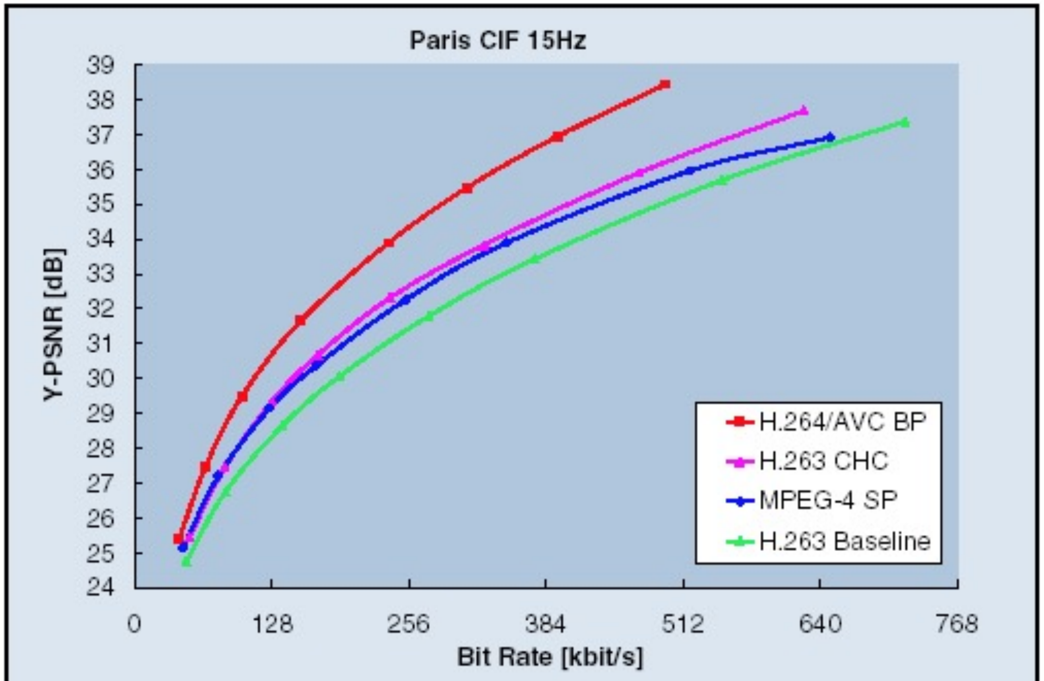


Figure 22. Luminance PSNR versus average bit rate for different coding standards, measured for the test sequence *Paris* for video conferencing applications (from [36]).

Table 2.
Average bit rate savings for video conferencing applications (from [10]).

	Average Bit Rate Savings Relative To:		
Coder	H.263 CHC	MPEG-4 SP	H.263 Base
H.264/AVC BP	27.69%	29.37%	40.59%
H.263 CHC	–	2.04%	17.63%
MPEG-4 SP	–	–	15.69%

From [Ostermann02]

What about complexity ?

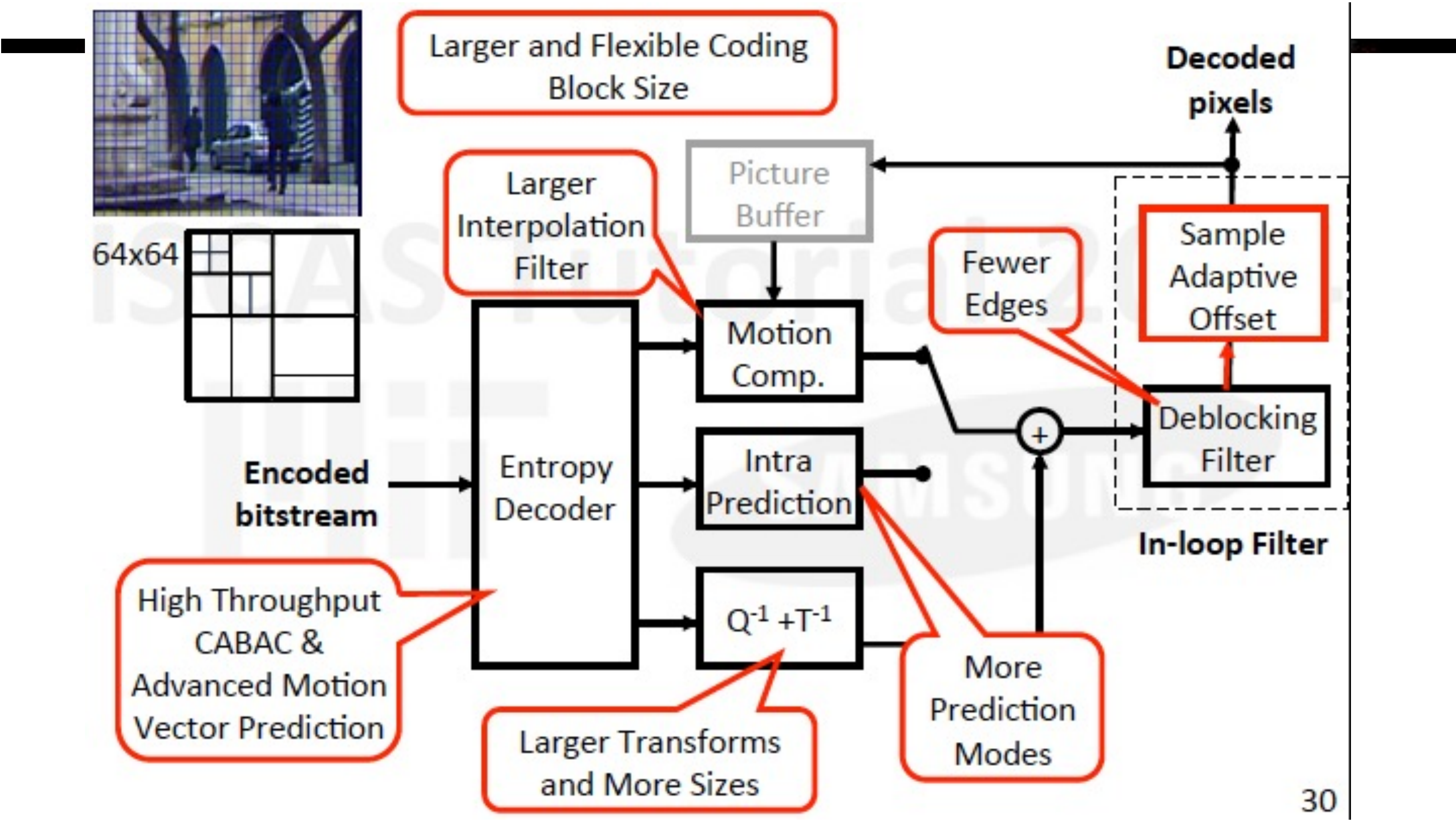
- H.264 decoder is about 2 times as complex as an MPEG-4 Visual decoder for the Simple profile
- H.264 encoder is about 10 times as complex as a corresponding MPEG-4 Visual encoder for the Simple profile
- The H.264/AVC main profile decoder suitable for entertainment applications is about 4 times more complex than MPEG-2

High Efficiency Video Coding (HEVC)

- Targeting for high resolution videos: HD (1920x1080) to ultra HD (7680x4320), progressive only (60p)
- Two targeted applications
  - Random access
  - Low delay
- Two categories of profile
  - High efficiency (HE)
  - Low complexity (LC)
- Performance: 2x better video compression performance compared to H.264/AVC.
  - Half the bit rate for similar quality

- Committee draft: Feb 2012.
- Standardization: Early 2013

Block Diagram of HEVC



Red boxes indicate changes from H.264/AVC

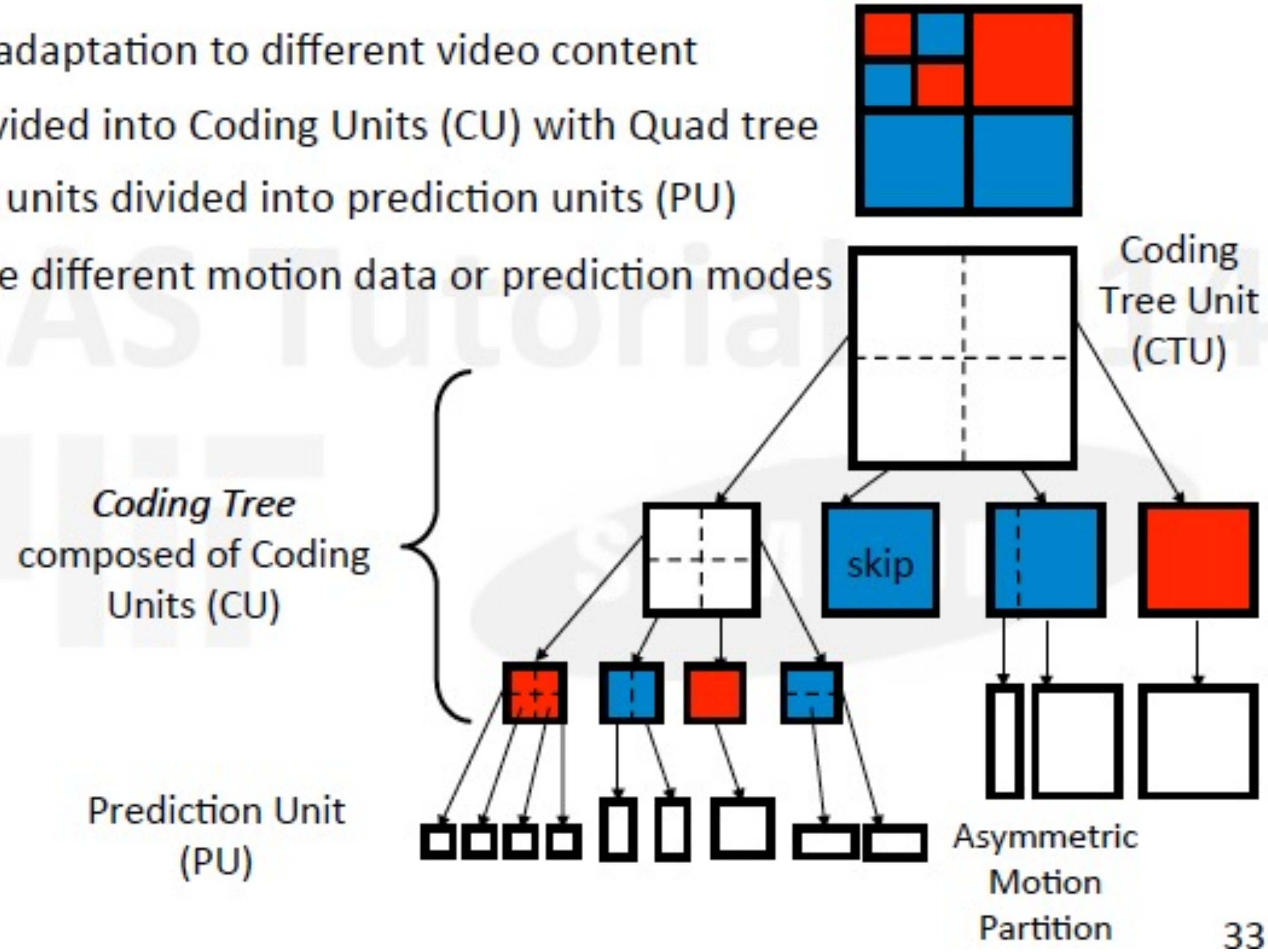
From [Sze2014]

New Coding Tools in HEVC

- Quadtree partition in 64x64 blocks: Block sizes from 8x8 to 64x64
- Up to 34 directions for intra-prediction
- For sub-pel motion estimation (down to ¼ pel), use 6- or 12-tap interpolation filter
- Advanced motion vector prediction
- CABAC or Low Complexity Entropy Coding
- Deblocking filter or Adaptive Loop Filter
- Extended precision options

Tree Structure for block partition

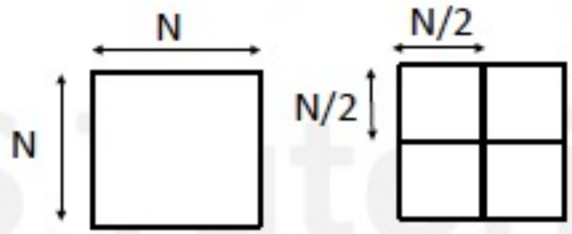
- Better adaptation to different video content
- CTU divided into Coding Units (CU) with Quad tree
- Coding units divided into prediction units (PU)
- PU have different motion data or prediction modes



From [Sze2014]

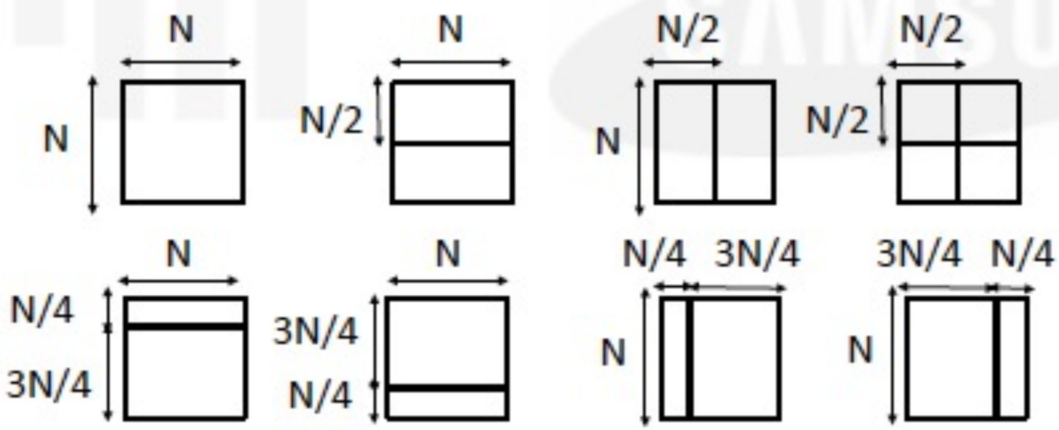
Prediction Units

- Intra-Coded CU can only be divided into square partition units
  - For a CU, make decision to split into four PU (8x8 CUs only) or single PU



Two methods of partitioning for intra-coded CU

- Inter-Coded CU can be divide into square and non-square PU as long as one side is at least 4 pixels wide (note: no 4x4 PU)



Eight methods of partitioning for inter-coded CU

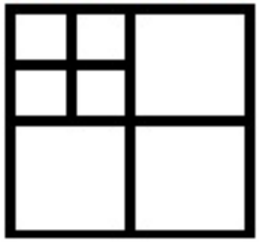
From [Sze2014]

Variable Size Transforms

Prediction residual of each coding unit may be further partitioned in a quad tree structure for transform coding

- HEVC supports 4x4, 8x8, 16x16, 32x32 integer transforms
  - Two types of 4x4 transforms (IDST-based for Intra, IDCT-based for Inter); IDCT-based transform for 8x8, 16x16, 32x32 block sizes
  - Integer transform avoids encoder-decoder mismatch and drift caused by slightly different floating point representations.
  - Parallel friendly matrix multiplication/partial butterfly implementation
  - Transform size signaled using Residual Quad Tree
- Achieves 5 to 10% increase in coding efficiency
- Increased complexity compared to H.264/AVC
  - 8x more computations per coefficient
  - 16x larger transpose memory

Represent residual of CU with TU quad tree

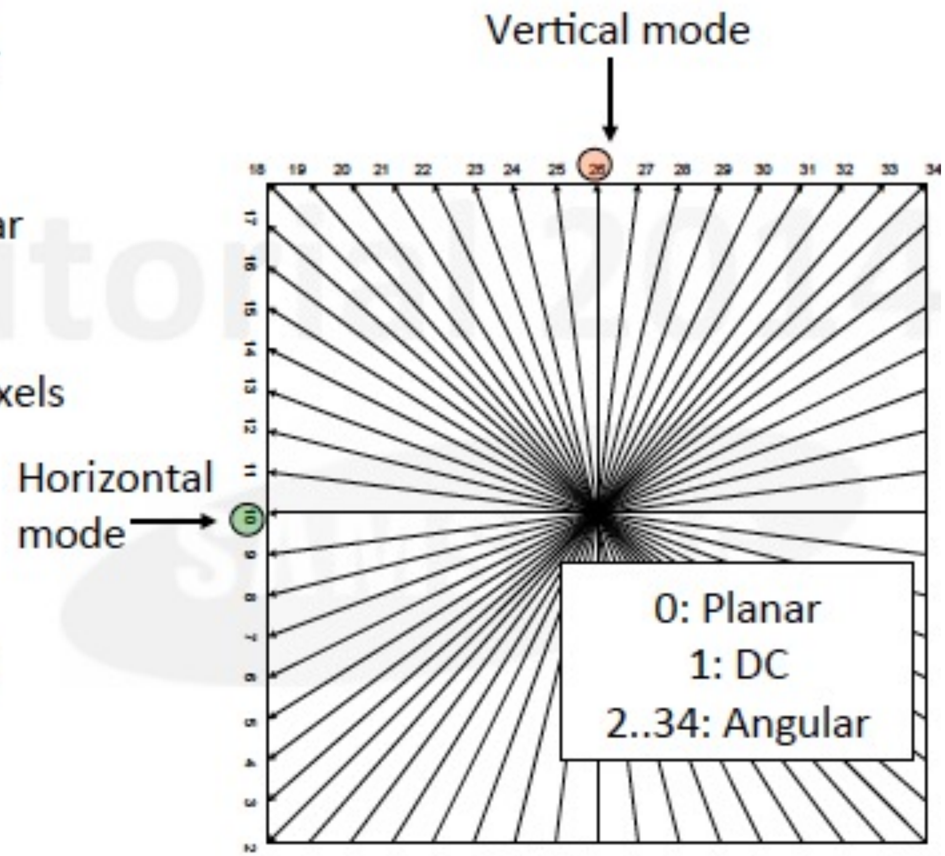


M. Budagavi et al., “Core Transform Design in the High Efficiency Video Coding (HEVC) Standard,” IEEE JSTSP, 2013

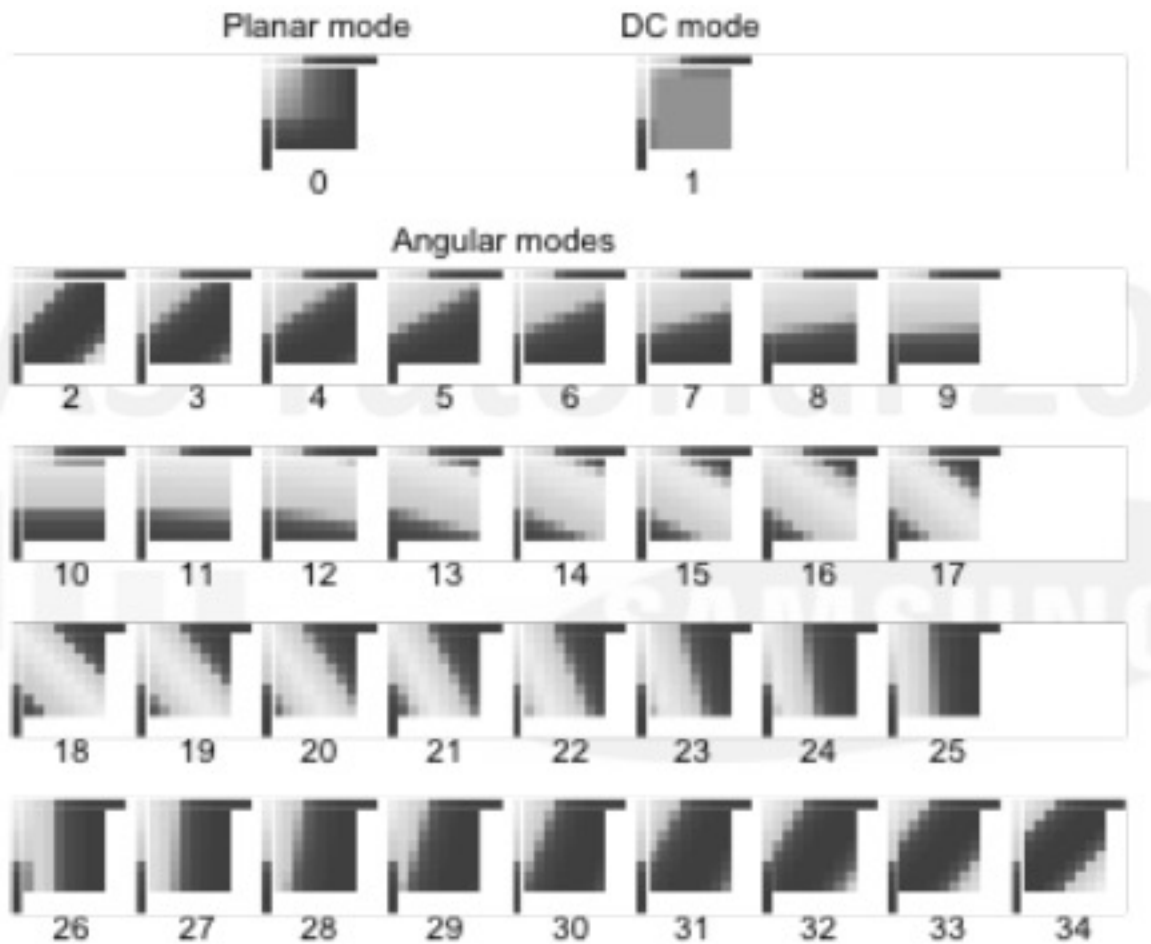
From [Sze2014]

Intra-Prediction Modes

- H.264/AVC has 10 modes
  - angular (8 modes), DC, planar
- HEVC has 35 modes
  - angular (33 modes), DC, planar
- Angular prediction
  - Interpolate from reference pixels at locations based on angle
- DC
  - Constant value which is an average of neighboring pixels (reference samples)
- Planar
  - Average of horizontal and vertical prediction



From [Sze2014]

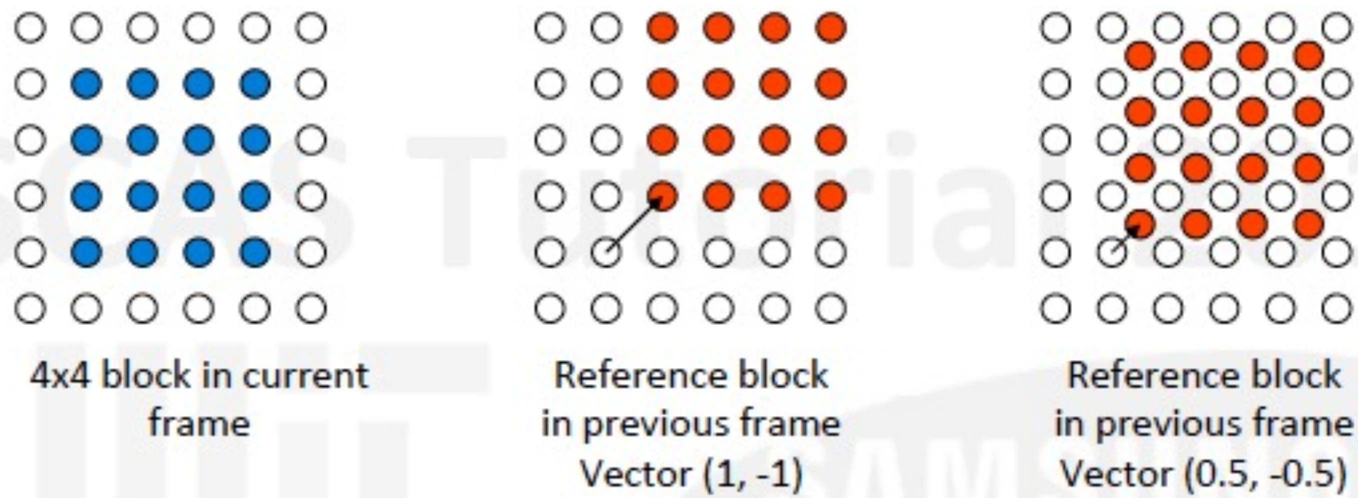


From [Sze2014]

J. Lainema, W.-J. Han, “Intra Prediction in HEVC,” ***High Efficiency Video Coding (HEVC): Algorithms and Architectures***, Springer, 2014.

Motion Compensated Inter-Prediction

- Motion vectors can have up to ¼ pixel accuracy (interpolation required)



- In H.264/AVC, luma uses 6-tap filter, and chroma uses bilinear filter
- In HEVC, luma uses 8/7-tap and chroma uses 4-tap
  - Different coefficients for ¼ and ½ positions
- Restricted prediction on small PU sizes

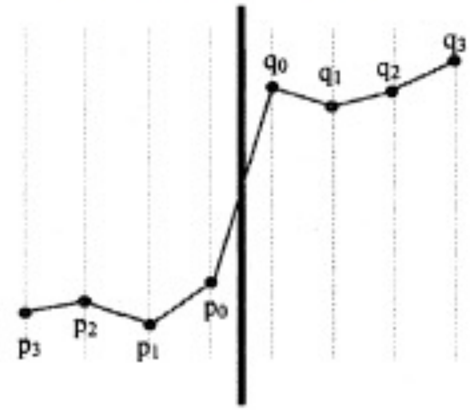
From [Sze2014]

Deblocking Filtering

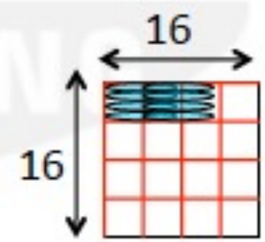
- Removes blocking artifacts due to block based processing
  - Computationally intensive in H.264/AVC

w/o deblocking

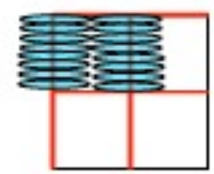
w/ deblocking



- In H.264/AVC, performed on every 4x4 block edge
  - Each macroblock has 128 pixel edges, 32 edge calculations
  - Each 4x4 depends on neighboring 4x4



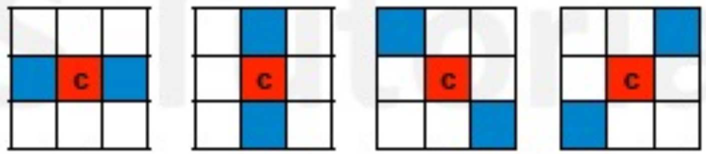
- In HEVC, performed on every 8x8 block edge
  - Each 16x16 CTU has 64 pixel edges, 8 edge calculations
  - All 8x8 are independent (can be processed in parallel)



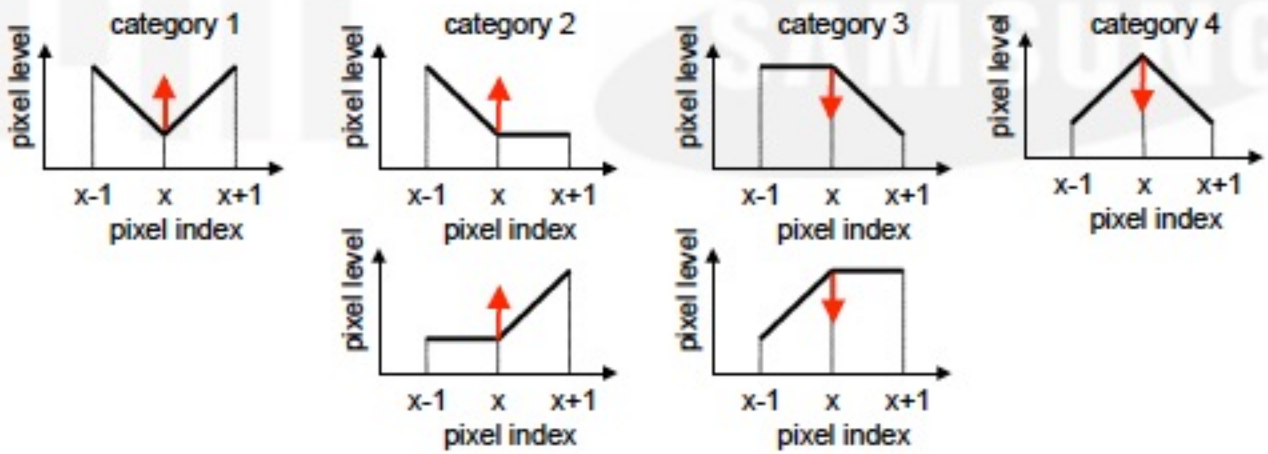
From [Sze2014]

Deblocking Filtering: Sample Adaptive Offset (SAO)

- Filter to address local discontinuities
  - Edge Offset and Band Offset
- Check neighbors in one of 4 directions (0, 90, 135, 45 degrees)



- Based on the values of the neighbors, apply one of 4 offsets



From [Sze2014]

With SAO Without SAO

C.-M. Fu et al., "Sample Adaptive Offset in the HEVC Standard," *IEEE Transactions on Circuits and Systems for Video Technology*, 2012

From [Sze2014]