Video Coding Standards and Scalable Coding

Yao Wang
Polytechnic School of Engineering, New York University
Outline

• Role of standards
• H.264/AVC
• HEVC
• Scalable coding and H.264/SVC
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?

Encoder → Bitstream → Decoder

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)

 enabled 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”
History of Video Coding Standards

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

~2x Improvement in compression ratio every decade!
## International Image and Video Compression Standards

<table>
<thead>
<tr>
<th>Standard</th>
<th>Application</th>
<th>Bit Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>JPEG</td>
<td>Continuous-tone still-image compression</td>
<td>Variable</td>
</tr>
<tr>
<td>H.261</td>
<td>Video telephony and teleconferencing over ISDN</td>
<td>p x 64 kb/s</td>
</tr>
<tr>
<td>MPEG-1</td>
<td>Video on digital storage media (CD-ROM)</td>
<td>1.5 Mb/s</td>
</tr>
<tr>
<td>MPEG-2</td>
<td>Digital Television</td>
<td>2-20 Mb/s</td>
</tr>
<tr>
<td>H.263</td>
<td>Video telephony over PSDN and Internet</td>
<td>33.6-? kb/s</td>
</tr>
<tr>
<td>MPEG-4</td>
<td>Object-based coding, synthetic content, interactivity</td>
<td>Variable</td>
</tr>
<tr>
<td>JPEG-2000</td>
<td>Improved still image compression</td>
<td>Variable</td>
</tr>
<tr>
<td>H.264/AVC</td>
<td>Improved video compression</td>
<td>10’s kb/s to Mb/s</td>
</tr>
<tr>
<td>H.265/HEVC</td>
<td>Improved video compression, especially for ultra HD</td>
<td>10’s kb/s to Mb/s</td>
</tr>
</tbody>
</table>

Modified from John G. Apostolopoulos
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 bfps (modems or internet access, desktop conferencing)
  – Half-pel MC and other improvement (Variable block sizes)

• **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)**
  – TV/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)
  - 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
From [Sze2014]
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
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-, unidirectional-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

8 possible directions
Sample Intra Prediction Modes

Mode 0: Vertical
Mode 1: Horizontal
Mode 2: DC

A → M : Neighboring samples that are already reconstructed at the encoder and at the decoder side
☐ : Samples to be predicted

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

![Diagram showing motion-compensated prediction with multiple reference images.](image)

*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.
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.

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 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)
- 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

- Whether filtering will be turned on depends on the pixel differences involving pixels $p_0, \ldots, q_0, \ldots$, 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.

From [Ostermann04]
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
Coding efficiency for video streaming

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

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

<table>
<thead>
<tr>
<th>Coder</th>
<th>Average Bit Rate Savings Relative To:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MPEG-4 ASP</td>
</tr>
<tr>
<td>H.264/AVC MP</td>
<td>37.44%</td>
</tr>
<tr>
<td>MPEG-4 ASP</td>
<td>–</td>
</tr>
<tr>
<td>H.263 HLP</td>
<td>–</td>
</tr>
</tbody>
</table>

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]).

<table>
<thead>
<tr>
<th>Coder</th>
<th>Average Bit Rate Savings Relative To:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>H.263 CHC</td>
</tr>
<tr>
<td>H.264/AVC BP</td>
<td>27.69%</td>
</tr>
<tr>
<td>H.263 CHC</td>
<td>-</td>
</tr>
<tr>
<td>MPEG-4 SP</td>
<td>-</td>
</tr>
</tbody>
</table>

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)
The latest video coding standard

- 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
• Debloating 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

- 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)

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


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]

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

- 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]
From [Sze2014]

Coding Efficiency Based on PSNR

From [Sze2014]
Subjective Tests for Entertainment Applications (Random Access)

<table>
<thead>
<tr>
<th>Sequences</th>
<th>Bit-rate Savings</th>
</tr>
</thead>
<tbody>
<tr>
<td>BQ Terrace</td>
<td>63.1%</td>
</tr>
<tr>
<td>Basketball Drive</td>
<td>66.6%</td>
</tr>
<tr>
<td>Kimono1</td>
<td>55.2%</td>
</tr>
<tr>
<td>Park Scene</td>
<td>49.7%</td>
</tr>
<tr>
<td>Cactus</td>
<td>50.2%</td>
</tr>
<tr>
<td>BQ Mall</td>
<td>41.6%</td>
</tr>
<tr>
<td>Basketball Drill</td>
<td>44.9%</td>
</tr>
<tr>
<td>Party Scene</td>
<td>29.8%</td>
</tr>
<tr>
<td>Race Horse</td>
<td>42.7%</td>
</tr>
<tr>
<td><strong>Average</strong></td>
<td><strong>49.3%</strong></td>
</tr>
</tbody>
</table>

From [Sze2014]
Intra-Frame Coding Efficiency

- HEVC also provides improved compression for still images

<table>
<thead>
<tr>
<th></th>
<th>BD-Rate Reduction</th>
</tr>
</thead>
<tbody>
<tr>
<td>H.264/AVC (intra only)</td>
<td>15.8%</td>
</tr>
<tr>
<td>JPEG 2000</td>
<td>22.6%</td>
</tr>
<tr>
<td>JPEG XR</td>
<td>30.0%</td>
</tr>
<tr>
<td>Web P</td>
<td>31.0%</td>
</tr>
<tr>
<td>JPEG</td>
<td>43.0%</td>
</tr>
</tbody>
</table>

From [Sze2014]
Other Related Standards

• Other MPEG standards
  – MPEG-7
    • 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
  – AVS (A Chinese video coding standard, roughly similar to H264)
  – VP8 (Google’s version of H264)
  – VP9 (Google’s version of HEVC)
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

Switched infrastructure, not transcode
d

Video is standard H. 264

Tablet/Smart Phone user

Room system

Legacy System

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.
Scalable Video Coding and Distribution

From Wainhouse Research, LLC.
Scalable Video Coding

• Definition
  – Ability to recover acceptable image/video by decoding only parts of the bitstream

• Ideal goal is an *embedded bitstream*
  – Truncate at any arbitrary rate

• Practical video coder
  – Layered coder: base layer provides basic quality, successive layers refine the quality incrementally
  – Fine granularity (FGS): each layer is very thin

• To be useful, a scalable solution needs to be more efficient than Simulcast or Transcoding
Illustration of Scalable Coding

- Amplitude (SNR or quality) scalability
- Spatial scalability

- 6.5 kbps
- 21.6 kbps
- 133.9 kbps
- 436.3 kbps
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 backwards compatible extensions
- Power adaptation for a trade-off between decoding time (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!
H.264/SVC (Scalable Video Coding)

• An scalable extension of H.264 / MPEG-AVC
• Using H.264/AVC as base layer
• A good trade-off between efficiency and error-propagation/drift
• Encode essentially uses multiple encodings to generate different layers
• Decoding complexity is similar to single-layer H.264 decoding
  – Uses only a single motion-compensation loop at the decoder
• Predicts not only residual (DCT) information, but also predict motion information and macroblock modes
SVC scalability modes

• Temporal scalability: using hierarchical B or hierarchical P structure.
  – No loss of coding efficiency when using hierarchical B

• Spatial scalability:
  – Using down/up sampling combined with switching between intra-layer and inter-layer prediction (CGS and MGS)

• Amplitude (quality) scalability
  – Same as spatial scalability where each layer has the same spatial resolution, but different QP

• QP cascading:
  – Using lower QP for lower spatial/temporal layers, increasing QP for higher spatial/temporal layers incrementally
Scalability Modes

- Temporal scalability (frame rate)
- Spatial scalability (picture size)
- Amplitude (AKA SNR or Quality) scalability (quantization stepsize or QP)
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, Layer 1+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

Foreman, CIF 30 Hz

Average Y-PSNR [dB]

- IPPPP
- IBPBP
- IBBP
- GOP04
- GOP08
- GOP16
- GOP32
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

(a) Raw video

ME at Base

ME at Enhance

Spatial/temporal down-sampling
Spatial/temporal up-sampling

Base-layer compressed bit stream

(b) Base-layer compressed bit stream

Enhanced-layer compressed bit stream
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!
Recommended Readings (1)

- H.264:
- HEVC
  - Vç, HEVC tutorial at ISCAS2014: 
Recommended Readings (2)

- **SVC:**
  - [http://iphome.hhi.de/wiegand/assets/pdfs/DIC_SVC_07.pdf](http://iphome.hhi.de/wiegand/assets/pdfs/DIC_SVC_07.pdf)

- **AVS**
    (King Ngan, Chinese University of Hong Kong)
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 H.265/AVC, 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.

3. One major innovation in HEVC beyond H.264/AVC is the coding tree structure, where a 64x64 block may be divided into coding units (CU) of various size. Each coding unit (CU) may be further divided into Prediction Units (PU) of different sizes. Finally the prediction error over each CU may be further divided into transform units (TU) of different sizes. Explain in more detail how does this work and why such partition may benefit the compression.

4. 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.

5. Compare temporal scalability through Hierarchical B and Hierarchical P structures. What are the pros and cons of each?
6. 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 (1Mbps), 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 bitstream, 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?