Advanced Multimedia Systems Design

Lecture 4 – Video Compression (MPEG-4/H.264/HEVC/H.265

Klara Nahrstedt

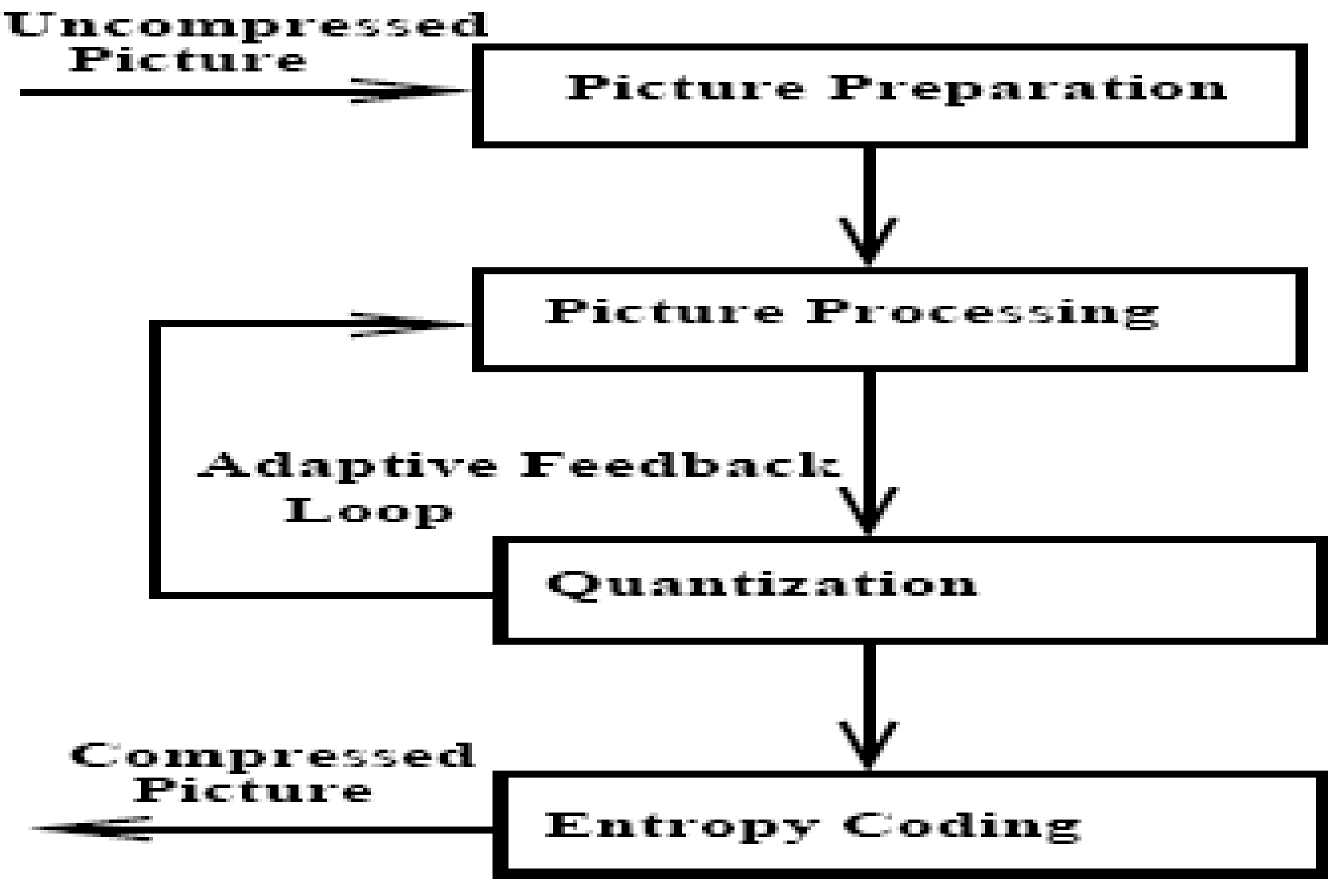
Fall 2018

Overview

- MPEG-4
- MPEG-7
- H.264
  - Coding Tools for H.264
    - I-slide, P-slide, B-slide
- Remarks on HEVC/H.265

MPEG-2 SUMMARY

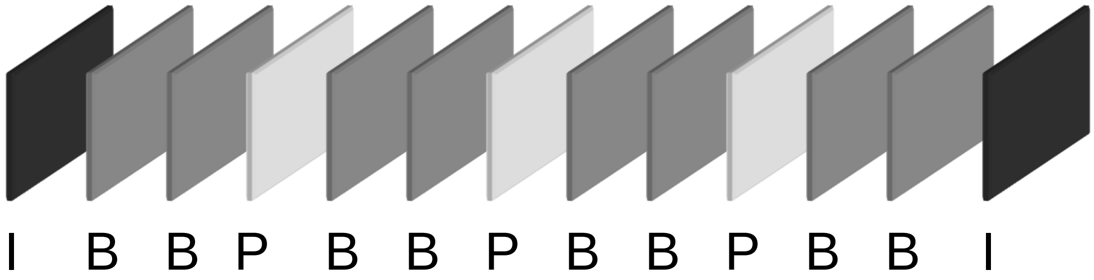
Video MPEG Compression is Hybrid Coding



RLE, Huffman
Arithmetic
Coding

MPEG Video Processing

- Intra frames (same as JPEG)
  - typically about 12 frames between I frames
- Predictive frames
  - encode from previous I or P reference frame
- Bi-directional frames
  - encode from previous and future I or P frames

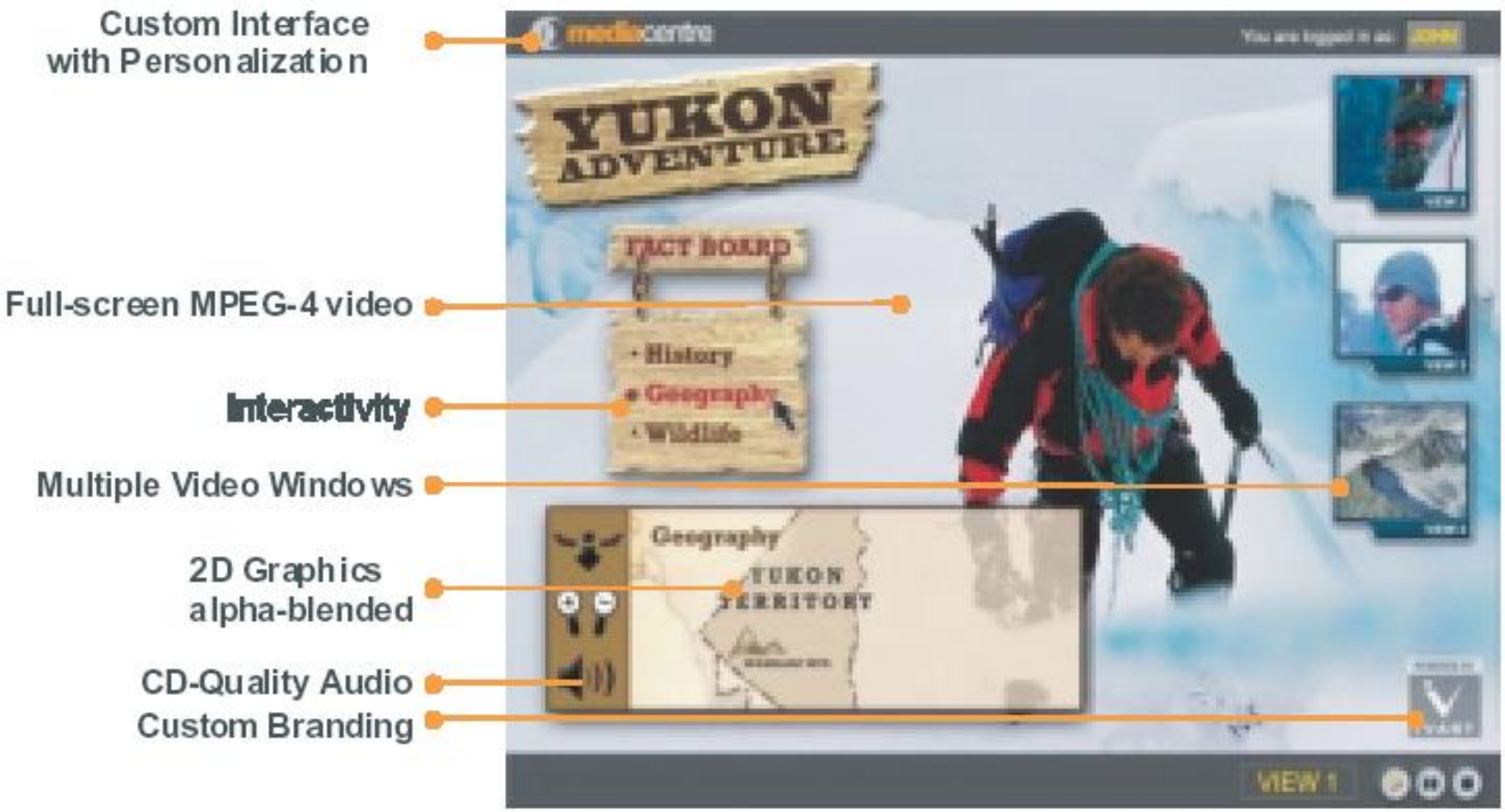


MPEG Video Quantization

- AC coefficients of B/P frames are usually large values, I frames have smaller values
  - Adjust quantization
- If data rate increases over threshold, then quantization enlarges step size (increase quantization factor Q)
- If data rate decreases below threshold, then quantization decreases Q

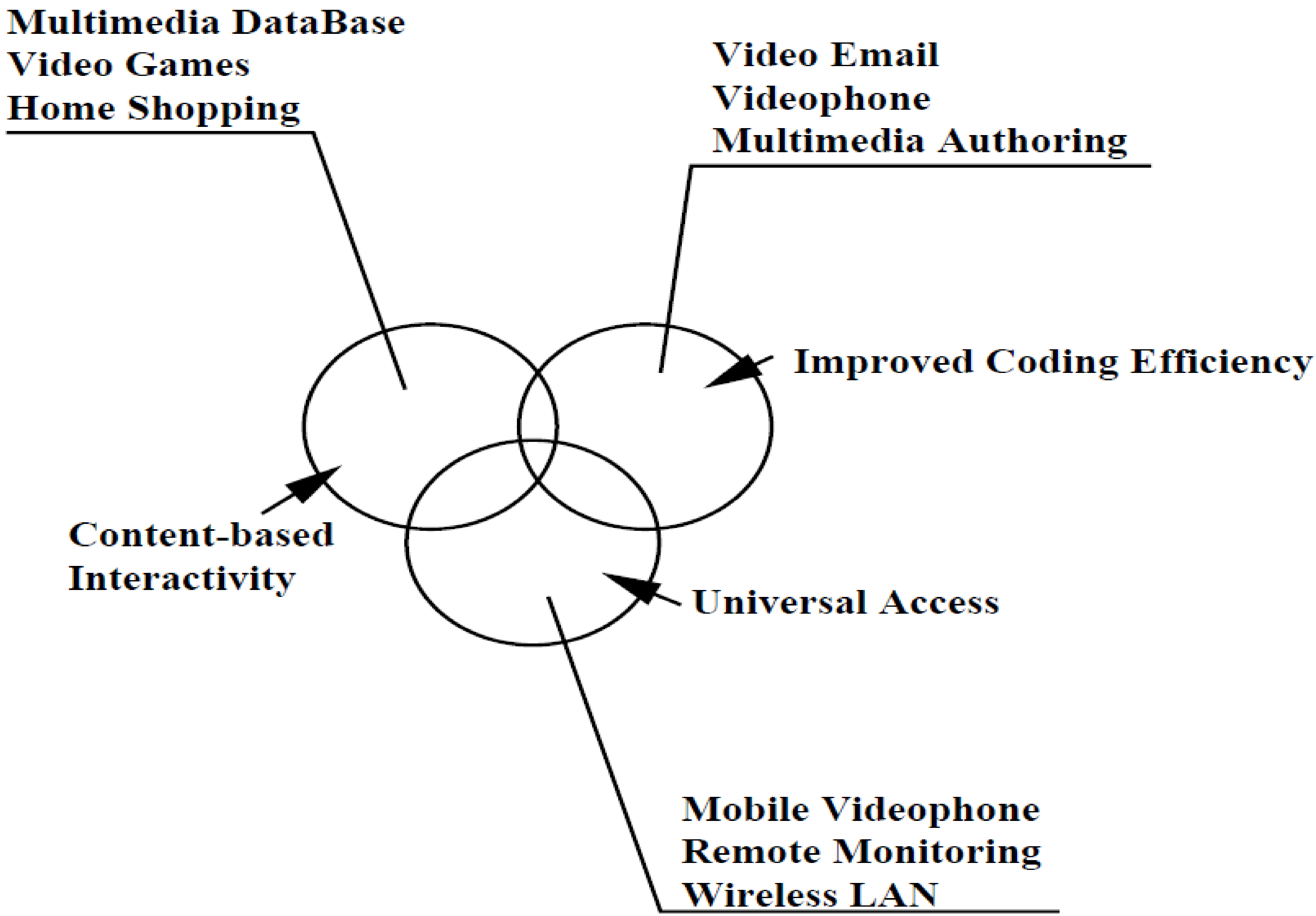
MPEG-4

MPEG-4 Example



ISO N3536 MPEG4

MPEG-4 Characteristics and Applications

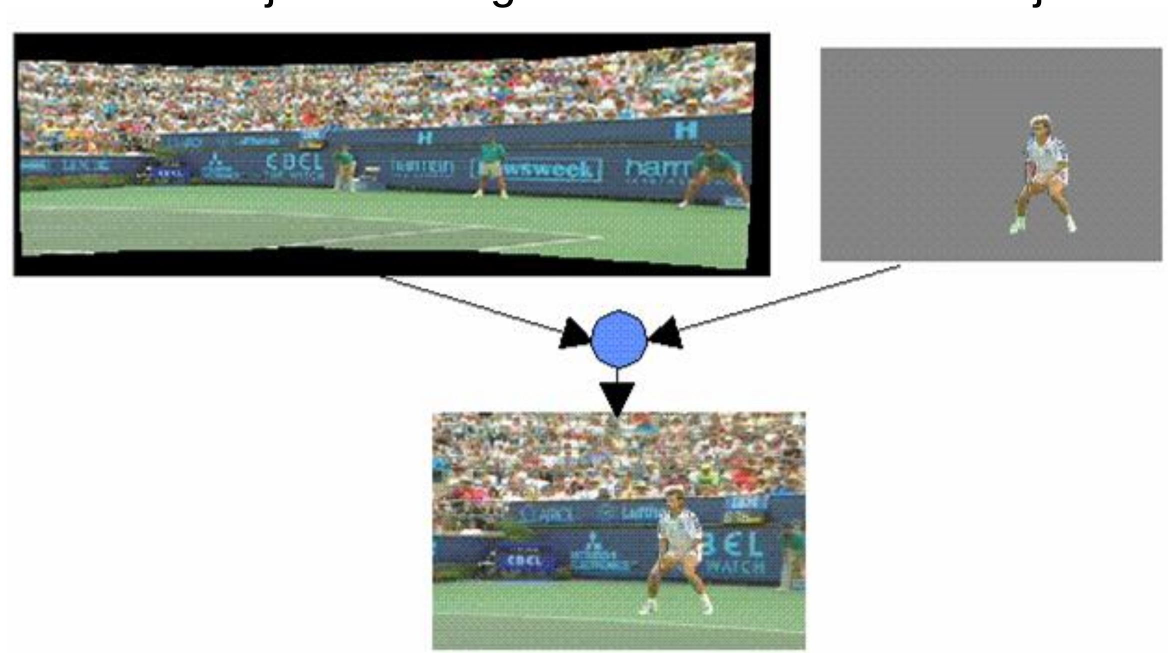


Media Objects

- An object is called a *media object*
  - real and synthetic images; analog and synthetic audio; animated faces; interaction
- Media objects have
  - Spatial relationships
  - Temporal relationships
- Compose media objects into a hierarchical representation
  - form compound, dynamic scenes

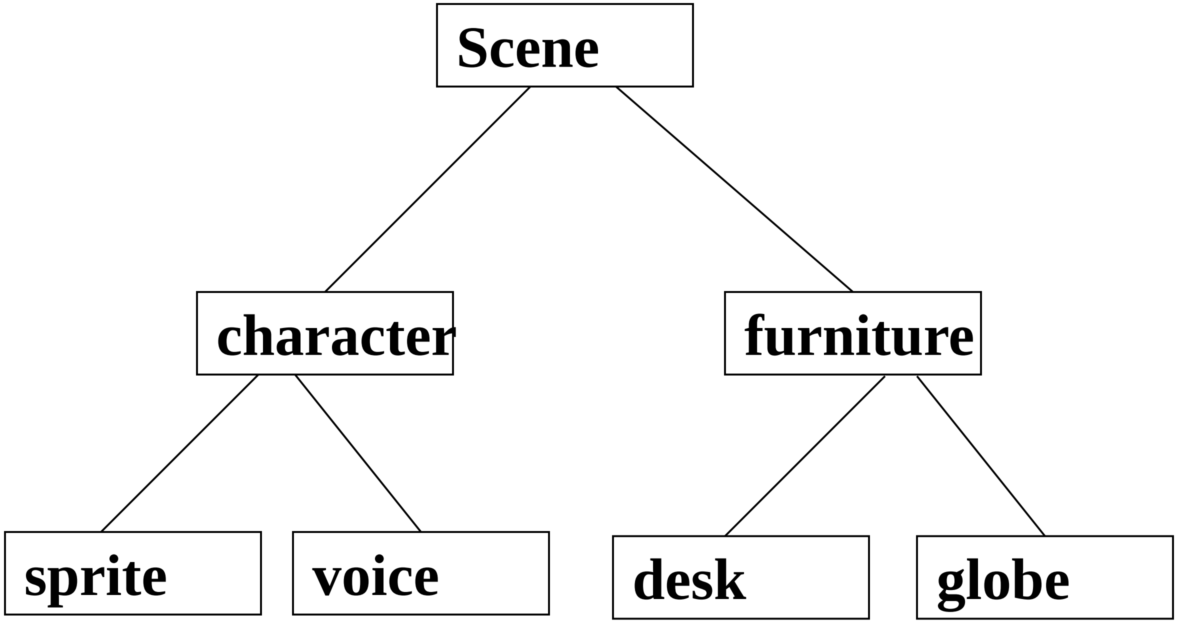
MPEG-4 Example

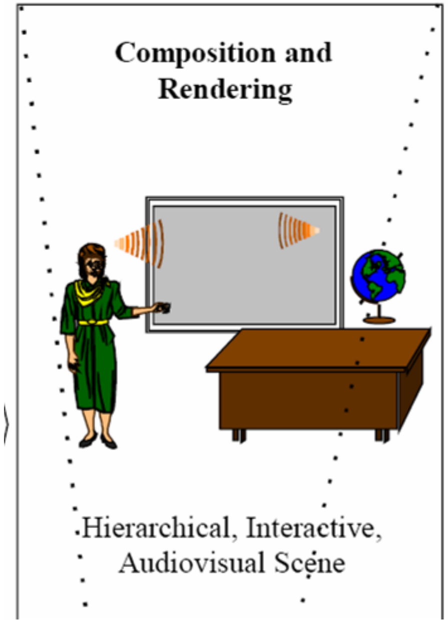
Media Object - Background　　　　Media Object -Player



ISO N3536 MPEG4

Spatial Relationship - Composition (Scene Graph)



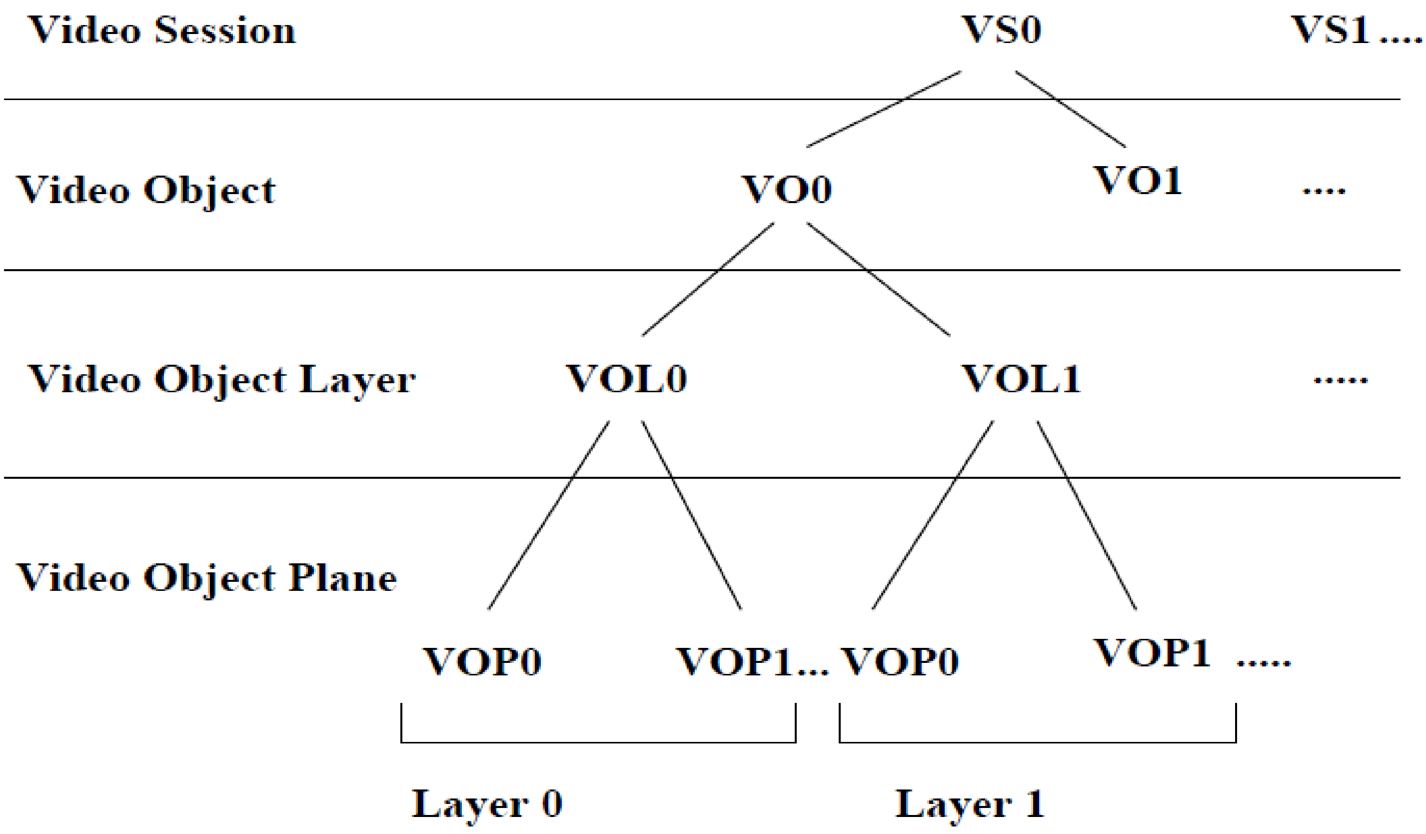


ISO N3536 MPEG4

Temporal Relationships

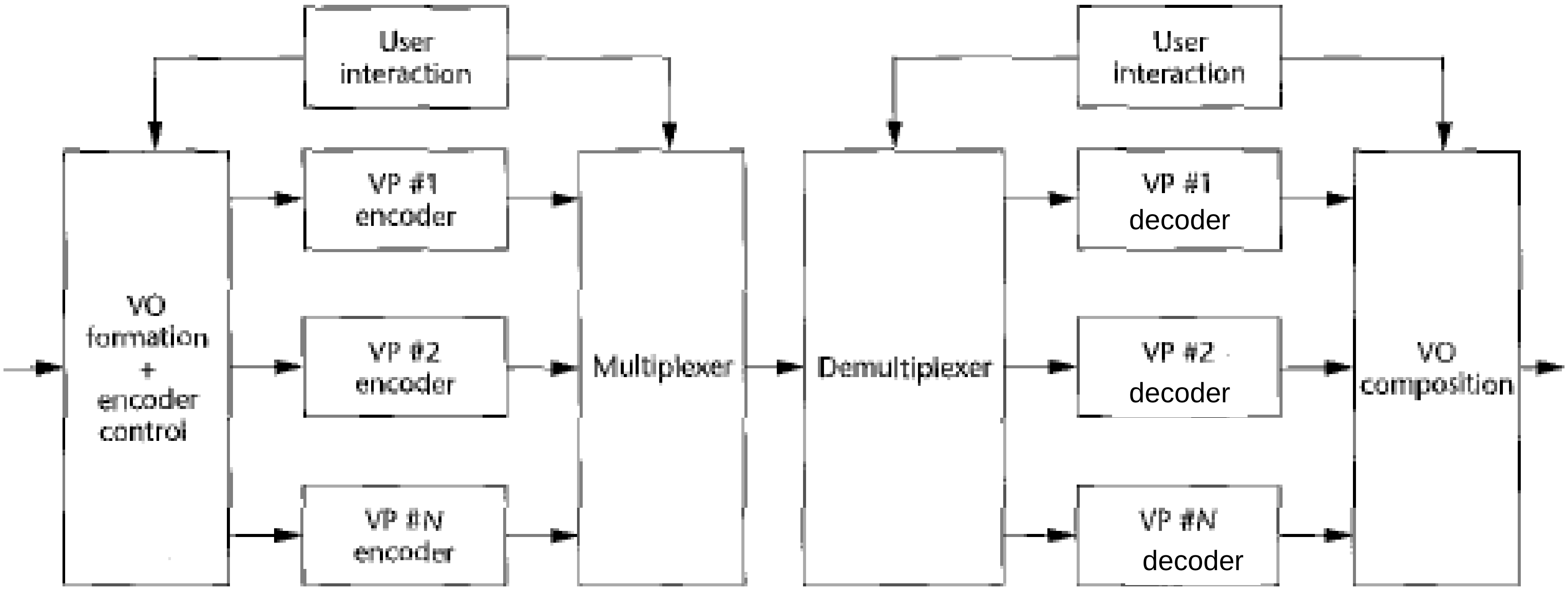
- Composition stream (BIFS) has its own associated time base
  - **Composition timestamps** specify at what time access units for composition must be ready at input of composition information decoder
- Timestamps are attached to each elementary stream
  - **Decoding timestamp (DTS)** specifies at which time the access unit for media object should be ready at decoder input
  - **Composition timestamp (CTS)** specifies time when object should be ready at the composition unit (compositor input).

Video Syntax Structure

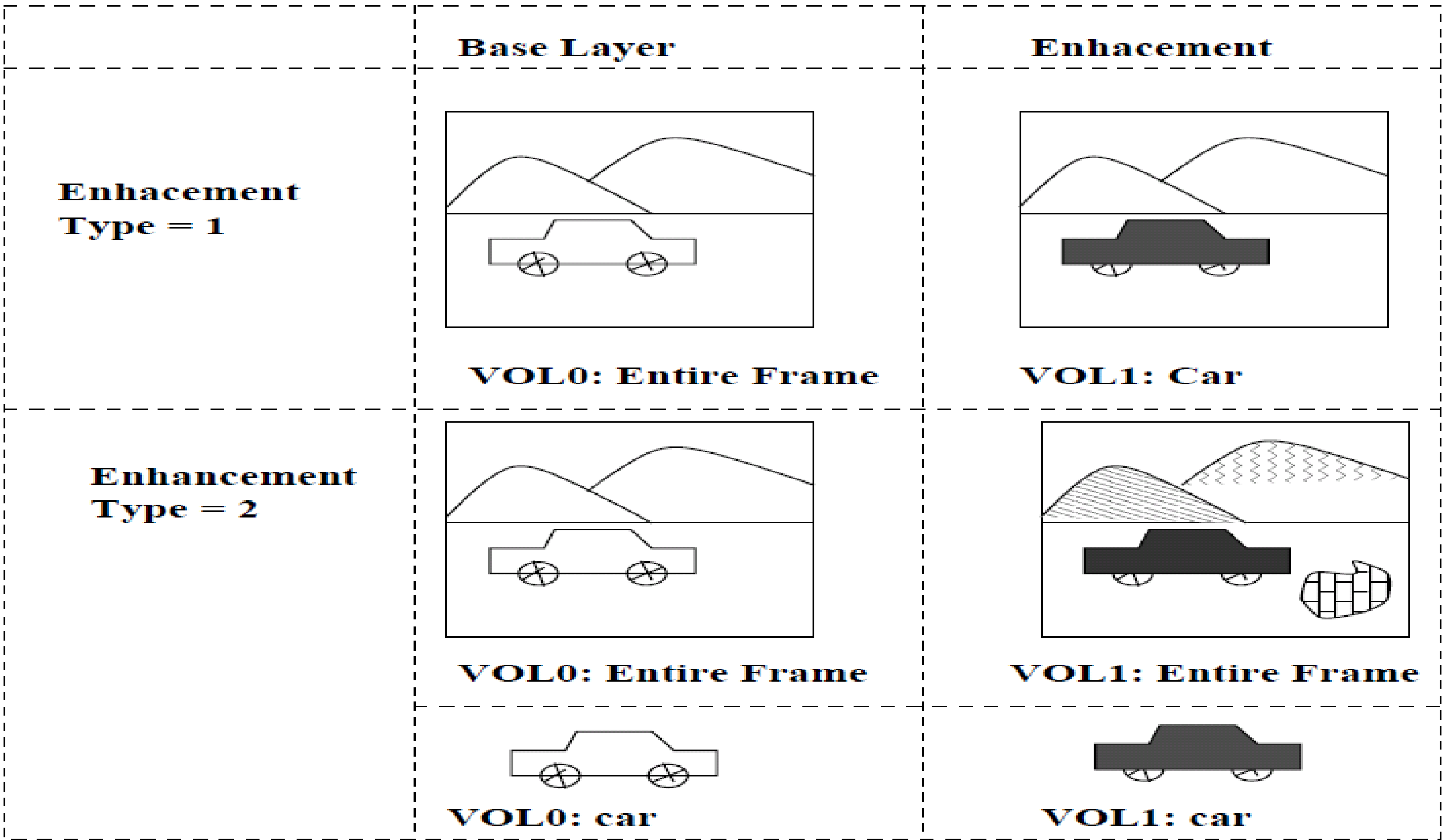


New MPEG-4 Aspect: Object-based layered syntactic structure

MPEG-4 Coding Architecture



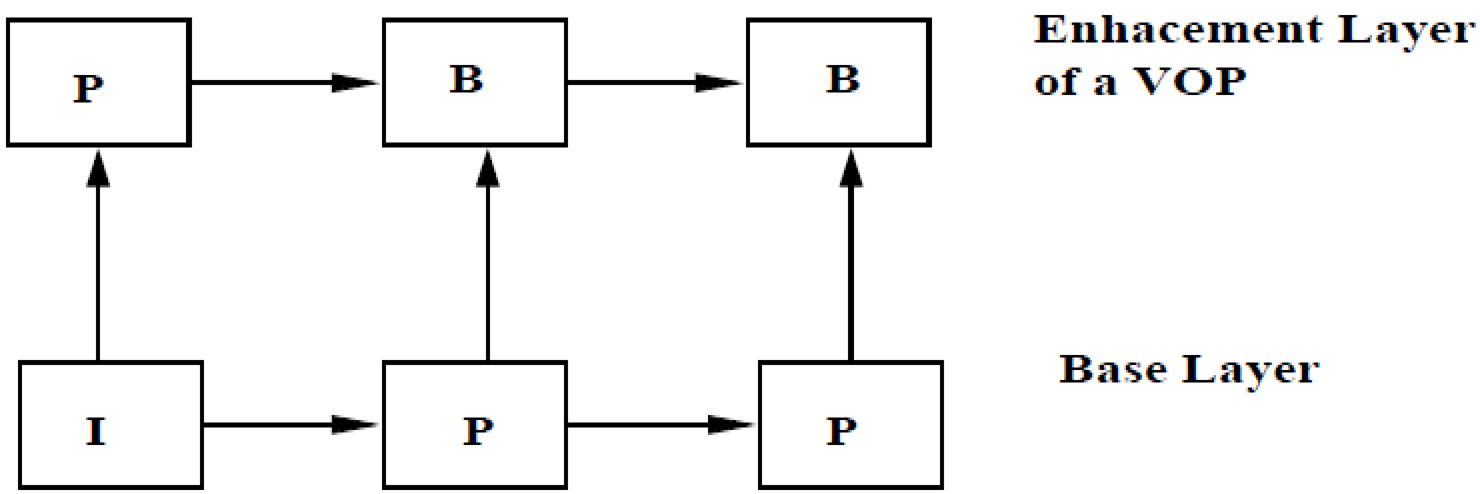
Examples of Base and Enhancement Layers



Coding of Objects

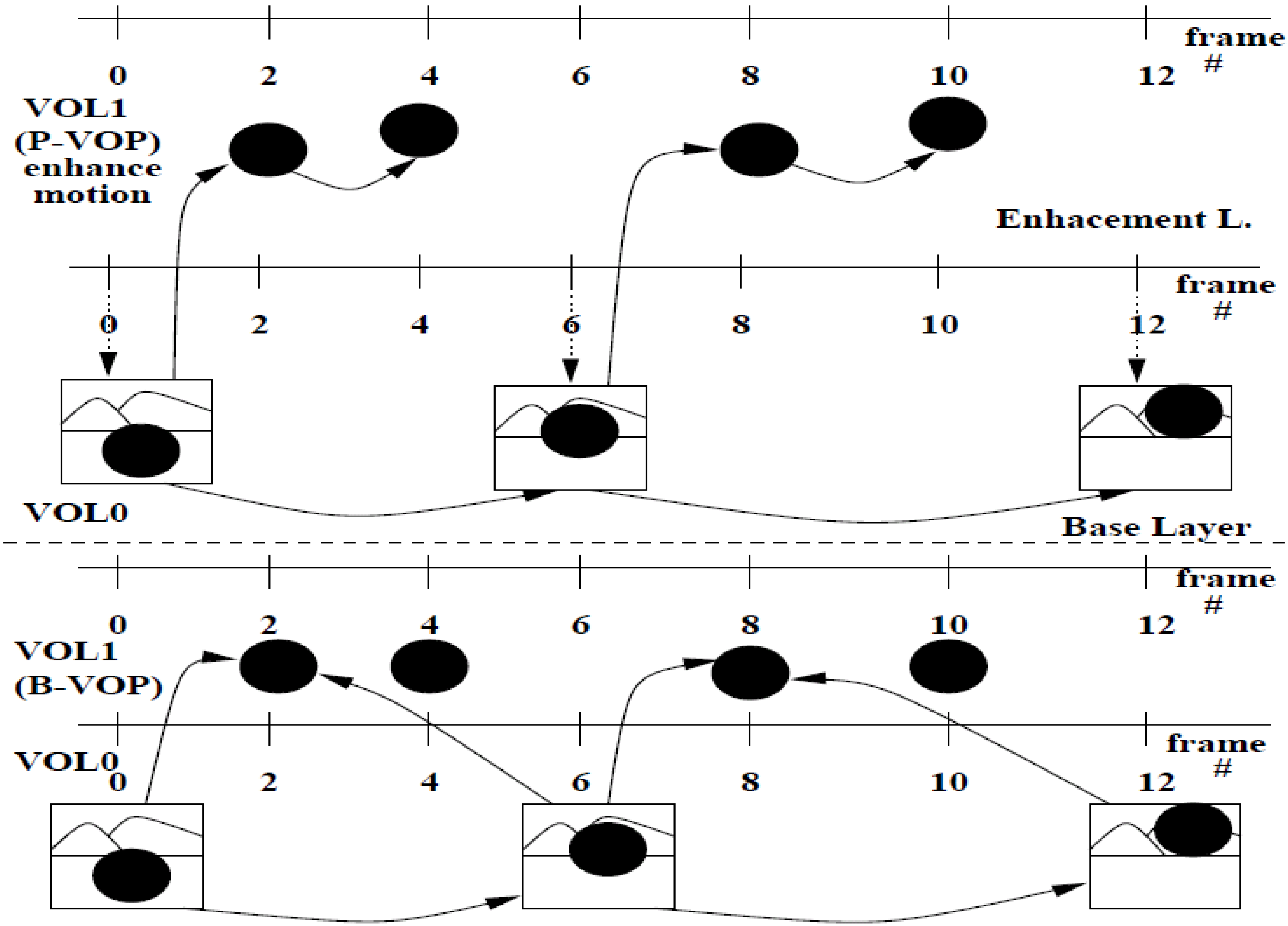
- Each VOP corresponds to an entity that after being coded is added to the bit stream
- Encoder sends together with VOP
  - Composition information where and when each VOP is to be displayed
- Users are allowed to change the composition of the entire scene displayed by interacting with the composition information

Spatial Scalability



VOP which is temporally coincident with **I-VOP** in the base layer, is encoded as **P-VOP in the enhancement layer**.
VOP which is temporally coincident with **P-VOP** in the base layer is encoded as **B-VOP in the enhancement layer.**

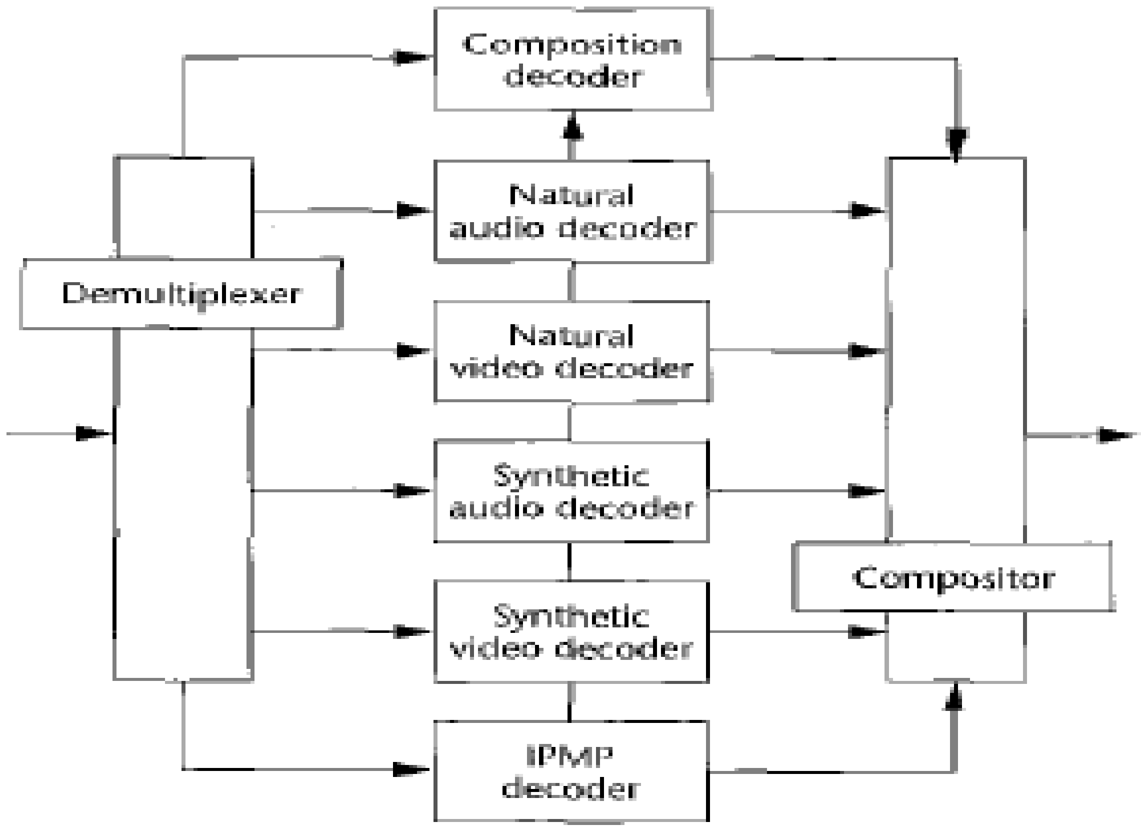
Temporal Scalability



Composition (cont.)

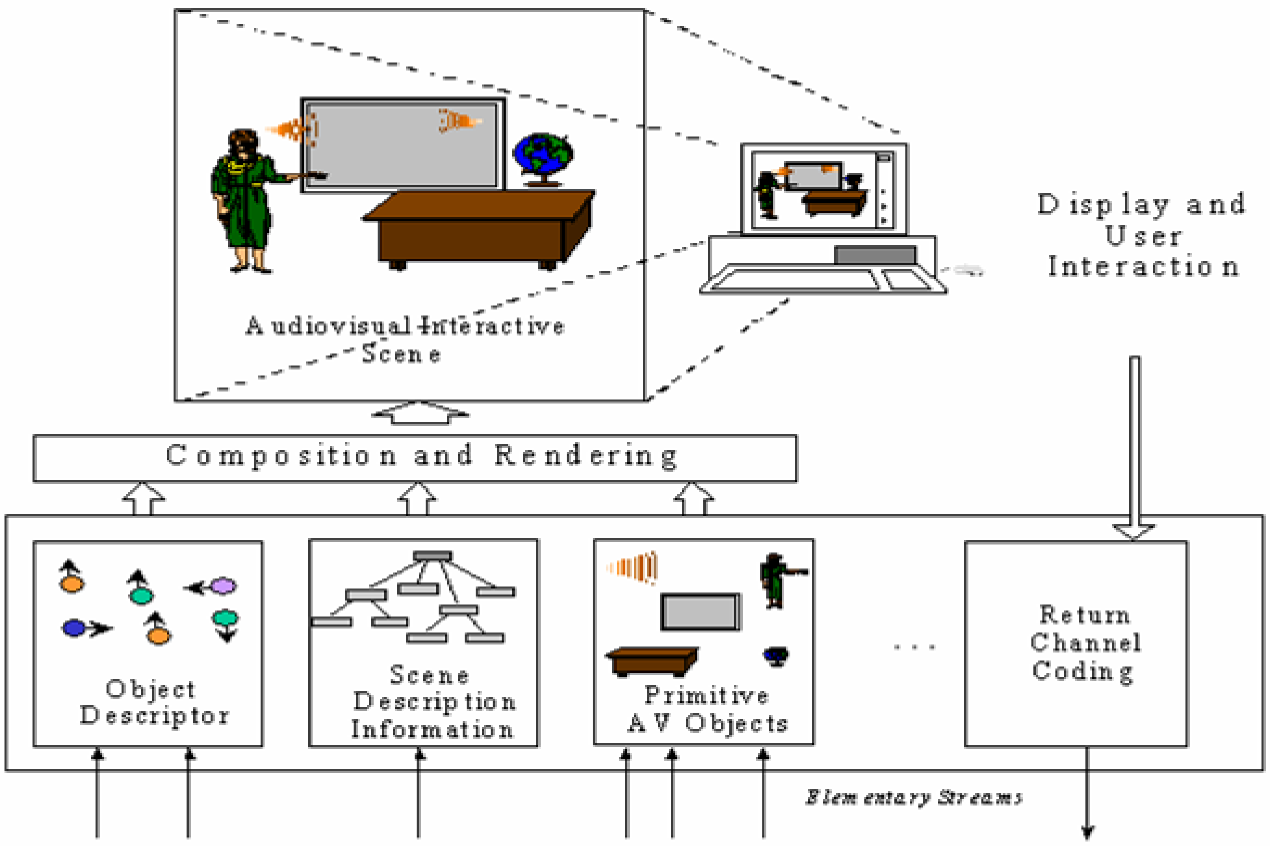
- Encode objects in separate channels
  - encode using most efficient mechanism
  - transmit each object in a separate *stream*
- Composition takes place at the decoder, rather than at the encoder
  - requires a binary scene description (BIFS)
- BIFS is low-level language for describing:
  - hierarchical, spatial, and temporal relations

MPEG-4 Decoder Architecture



Source: Batista et al, “MPEG-4:A multimedia standard for the third milenium: Part 1”, IEEE Multimedia, 1999

MPEG-4 Rendering



ISO N3536 MPEG4

Synchronization Layer (timing and synchronization)

- Synchronization of elementary streams
  - Elementary streams consist of access units which correspond to portions of stream with DTS and CTS
- Global timeline (high-resolution units)
  - e.g., 600 units/sec
- Each continuous track specifies relation
  - e.g., if a video is 30 fps, then a frame should be displayed every 33 ms.
- Others specify start/end time

MPEG-7

MPEG-7 Motivation

- Need for searching for audio-visual information
- Goal:
  - input a picture of motorbike from “Terminator II” and get search results
  - Efficiently search for video sequence where King Lear congratulates his assistants
  - Input a whistle of a melody and find a song

MPEG-7 Description

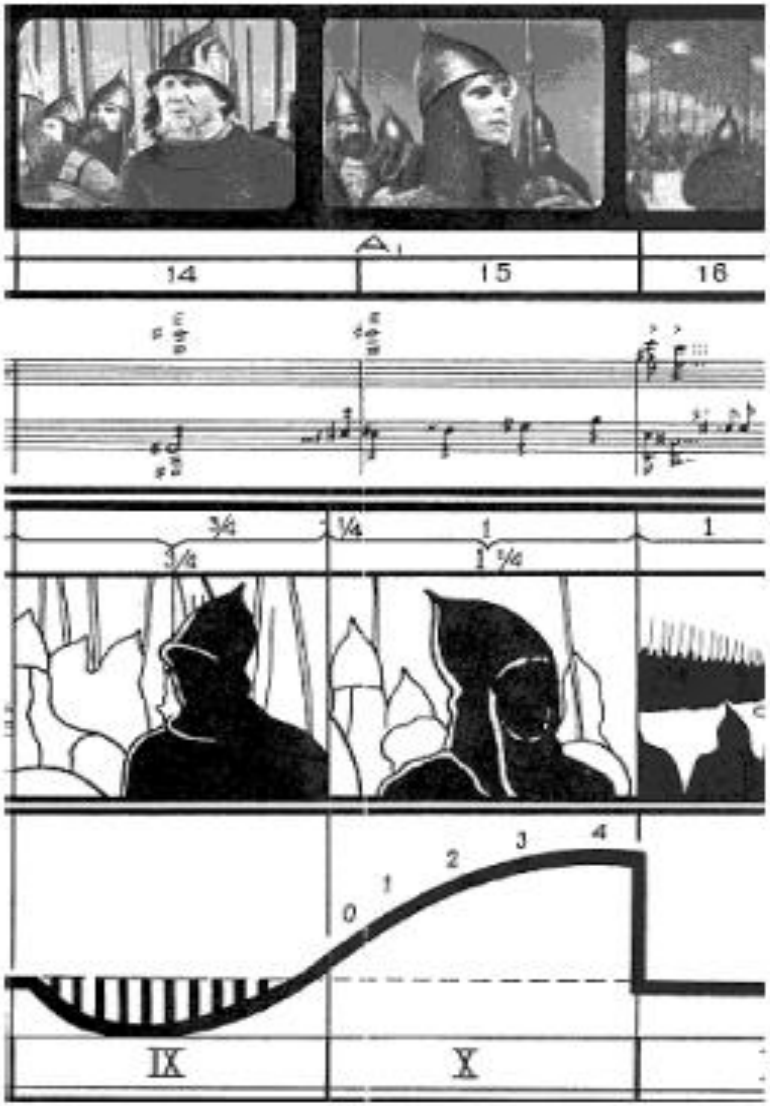
- Standardization of
  - core set of quantitative measures of audio-visual features, called Descriptors (D)
  - structures of descriptors and their relationships, called Description Schemes (DS)
  - A language, called Description Definition Language (DDL)
- Result: one can index and search for audio-visual material that has MPEG-7 metadata associated with it.

MPEG-7 Features

- **Multimedia content description**
  - Set of methods and tools for different classes of multimedia content description
  - Types of descriptions:
    - Medium-based: Sampling rate, location of shot cuts, camera’s focal length, etc
    - Physical: physical loudness of sound, ..
    - Perceptual: perceptual ‘pitch’
    - Transcription: dialog transcriptions
    - Architectural: domain of document structuralists
    - Annotative: domain of human annotations

MPEG-7 Features

- **Flexibility in data management**
  - Referencing - to parts of document, who document, series of documents
  - Multimodality – description of multimedia content in such a way as to allow queries based on visual descriptions to retrieve audio data and visa versa
  - Applications – description of stored, streamed, real-time, non-RT apps



Source: Nack et all, “MPEG-7”, IEEE Multimedia 1999

MPEG-7 Features

- **Globalization of data resources**
  - MPEG-7 resources can be
    - Co-located with audio-visual material or
    - Remote - live somewhere else
  - Need mechanisms to link audio-video material and MPEG-4 descriptions

MPEG-7 Features

- MPEG-7 does not extract descriptions/features automatically
- MPEG-7 concentrates on representations that can be used for descriptions
- MPEG-7 uses describing text documents such as SGML, XML (Extensible Markup Language), RDF (Resource Description Language)
- MPEG-4 and MPEG-7 are a pair of standards complementing each other

H.264 OVERVIEW

(SULLIVAN ET AL.”THE H.264/AVC ADVANCED VIDEO CODING STANDARD: OVERVIEW AND INTRODUCTION TO THE FIDELITY RANGE EXTENIONS”, SPIE CONFERENCE ON APPLICATIONS OF DIGITAL IMAGE PROCESSING, 2004)

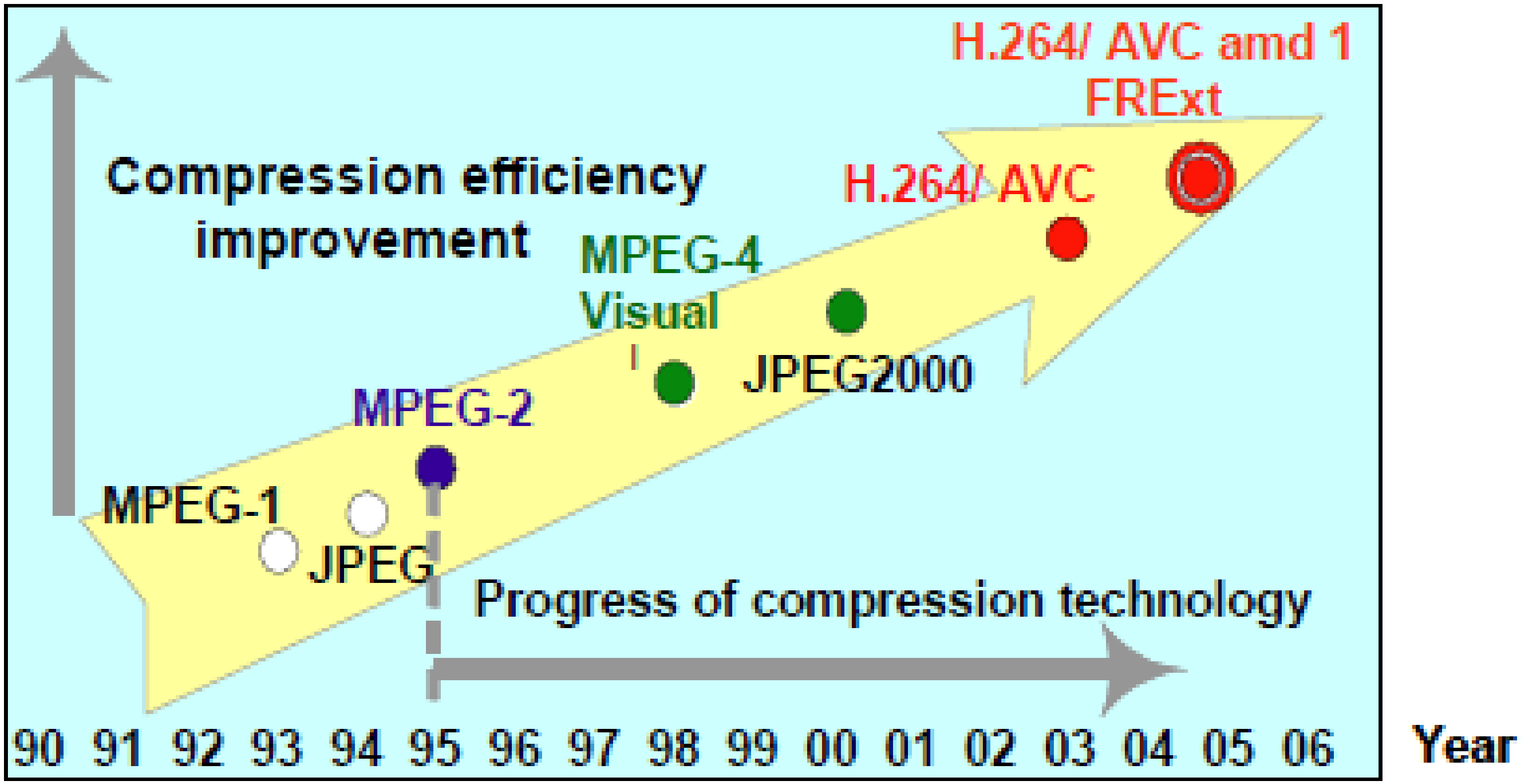
H.264 /AVC for Video Conferencing (Advanced Video Coding)

- Developed by Joint Video Team
  - ITU-T's Video Coding Experts Group (VCEG)
  - ISO/IEC's Moving Picture Experts Group (MPEG)
- Reference model developed in 2003
- Amendment added in 2004
  - Fidelity Range Extensions (FRExt, Amendment 1)
    - Demonstrates coding efficiency against MPEG-2 with potentially 3:1 compression ratio

H.264/AVC History

- In 1998 – ITU VCEG finished its video coding standard, H.26L
  - First Call for Proposals to improve coding
    - In 1999 - First proposed design of new standard
- In 2001 – ISO MPEG finished its own video coding standard, MPEG-4, Part 2
  - Call For Proposals (CFP) to improve coding efficiency
  - VCEG provided its own design to MPEG's CFP
- Joint Forces VCEG+MPEG => New standard
  - **MPEG-4/Part 10 = MPEG-4 AVC = H.264 AVC=JVT = H.26L = H.264**

Evolution of Compression Technology

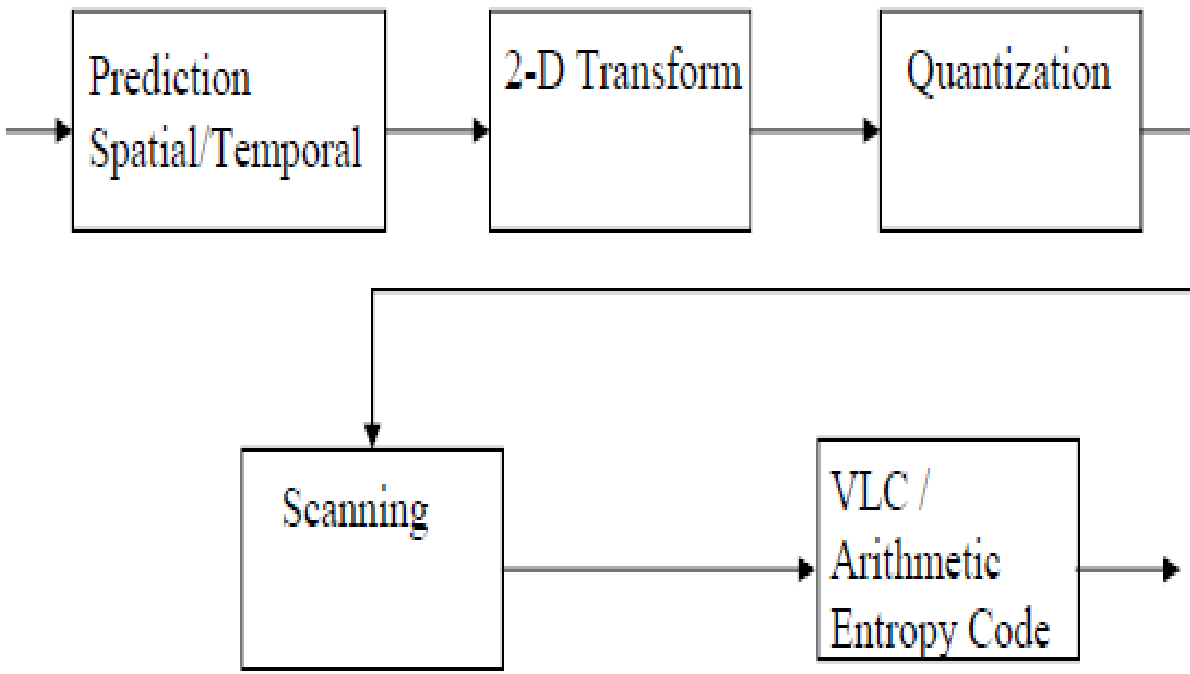


Source: Technology Overview, “AVC-Intra (H.264 Intra) Compression,” Panasonic Broadcast, 2007

H.264 Coding Structure

- At basic level, coding structure of H.264 similar to MPEG-1, MPEG-2, MPEG-4/Part 2.
- Hierarchy of video sequence
  - Sequence (pictures (slides (macroblocks partitions (sub-macroblock partitions (block (samples))))))
  - Bits associated with slide layer and below - Video Coding Layer (VCL)
  - Bits associated with higher layers – Network Abstraction Layer (NAL)

H.264 Encoding Block Diagram



VLC – Variable Length Coding (e.g., Huffman Coding)

H.264 Image Preparation

- First version had
  - 4:2:0 luma chroma relations
  - RGB-to-YCbCr color space translation
  - Subsampling chroma components by 2:1
  - 8 bit sample precision
- FRExt version has
  - 4:2:2, 4:4:4 formats
  - Higher than 8 bits precision

H.264 Image Preparation

- Basic unit of coding – macroblock
  - In 4:2:0 chroma format
    - macroblock 16x16 pixels in luma
    - Macroblock 8x8 pixels in chroma
  - In 4:2:2 chroma format
    - Macroblock in chroma 8x16 pixels
  - In 4:4:4 chroma format
    - Macroblock in chroma 16x16 pixels
- Major processing unit becomes **Slice** (group of macroblocks)

H.264 Image Coding Tools in Slices

(New in comparison to previous MPEG-1,2/H.26X standards)

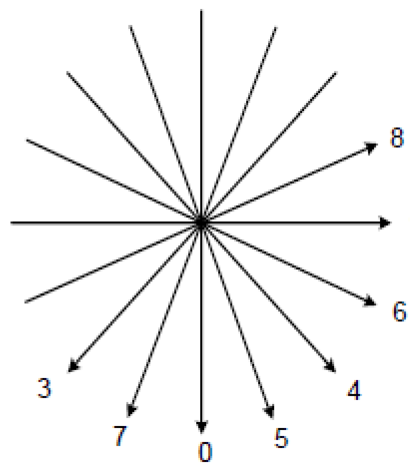
- Spatial and temporal prediction
- Lossless entropy coding
- Deblocking filter
- Residual color transform for efficient RGB coding
- Different slides use different coding tools
  - I-slice; P-slice, B-slice

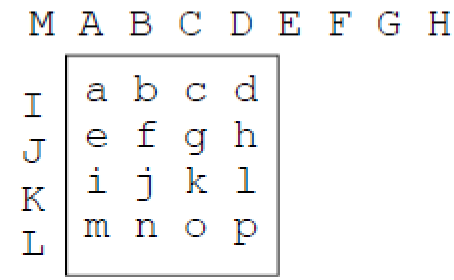
I-Slice Coding Algorithm

1. Spatially predict pixels from neighboring pixel values
2. Transform 4x4 or 8x8 blocks from spatial domain to other differentiable domain
3. Perform perceptual-based quantization
4. Scan quantized coefficients
   1. Zig-zag or field-based
5. Compress with Entropy Coding
   1. CAVLC (Context-Adaptive Variable Length Coding)
   2. CABAC (Context-Adaptive Binary Arithmetic Coding

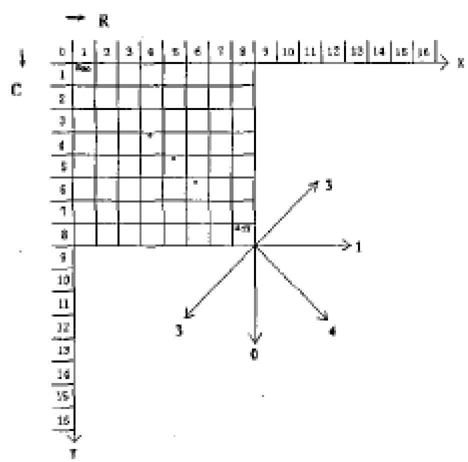
H.264 Intra-Spatial Prediction

- Due to Spatial correlation among pixels we can compress data by intra-spatial prediction
  - H.264 supports three basic types of **intra spatial predictions**





Spatial Prediction for 4x4 Block



Intra prediction directions

Predictor can be average
Value of neighbor values in
Different directions

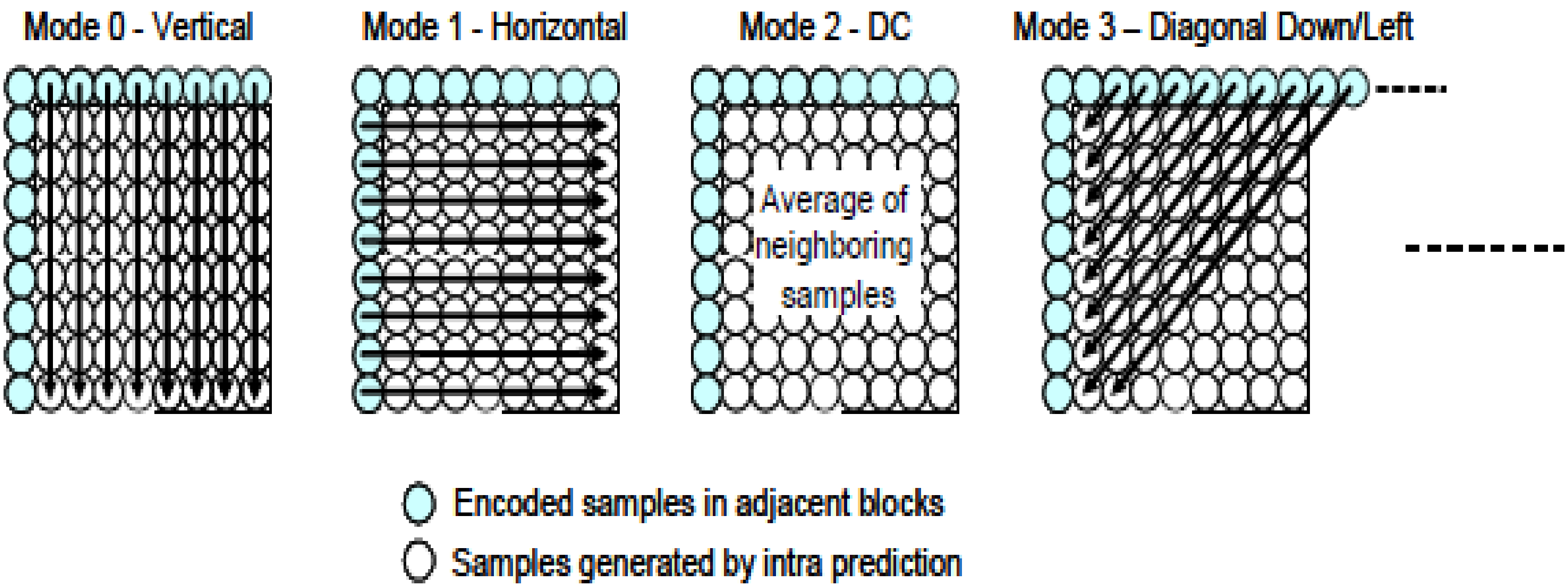
Comparison of Original and Intra Predicted Images

(a) Original input image

(b) Intra prediction image

Source: Technology Overview, “AVC-Intra Compression” Panasonic Broadcast, September 7, 2007

Examples of Spatial Intra Prediction Modes



Source: Technology Overview, “AVC-Intra Compression” Panasonic Broadcast, September 7, 2007

H.264 Transformation

- Goal of transformation – de-correlate data spatially
  - New approach
    - Use integer inverse transform design, rather than IDCT
    - Use integer transform for **forward transform**
      - Apply transformation matrices to every block of pixels

$$T_{4x4} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{bmatrix}, \quad T_{8x8} = \begin{bmatrix} 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 \\ 12 & 10 & 6 & 3 & -3 & -6 & -10 & -12 \\ 8 & 4 & -4 & -8 & -8 & -4 & 4 & 8 \\ 10 & -3 & -12 & -6 & 6 & 12 & 3 & -10 \\ 8 & -8 & -8 & 8 & 8 & -8 & -8 & 8 \\ 6 & -12 & 3 & 10 & -10 & -3 & 12 & -6 \\ 4 & -8 & 8 & -4 & -4 & 8 & -8 & 4 \\ 3 & -6 & 10 & -12 & 12 & -10 & 6 & -3 \end{bmatrix}$$

These are Transformation matrices For 4x4 and 8x8 blocks Of pixels

H.264 Transform

- When *16x16 intra-prediction mode* is used with 4x4 transform, then DC coefficients of the sixteen 4x4 luma blocks in the macro-block are further transformed by Hadamard transform

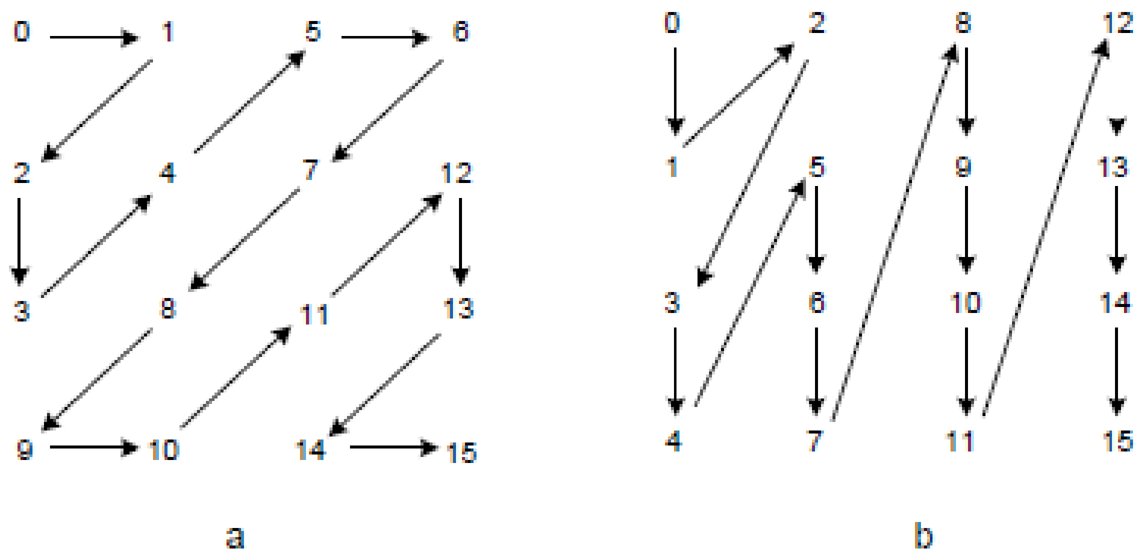
$$H_{2x2} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, \quad H_{4x4} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix}.$$

Hadamard Transform matrices for DC Coefficients in 16x16 macroblock

H.264 Perceptual-based quantization scaling

- Encoder specifies
  - For each transform block size
  - For intra and inter prediction
  - A customized scaling factor
- Decoder gets to tune quantization fidelity based on human visual system
- We don’t improve objective fidelity, but we improve subjective fidelity

Scanning



Frame-mode zig-zag scanning

Field-mode

- Frame-mode scan ordering designed to order the highest variance coefficients first and maximize number of consecutive zero-value coefficients
- Field-mode reflects decreasing correlation of the source data in the vertical dimension

Entropy Coding

- Lossless coding techniques – replace data with coded representation
- Variable Length Coding (VLC)
- Binary Arithmetic Coding (BAC)
- H.264/AVC supports context adaptation
  - ACVLC
  - ACBAC

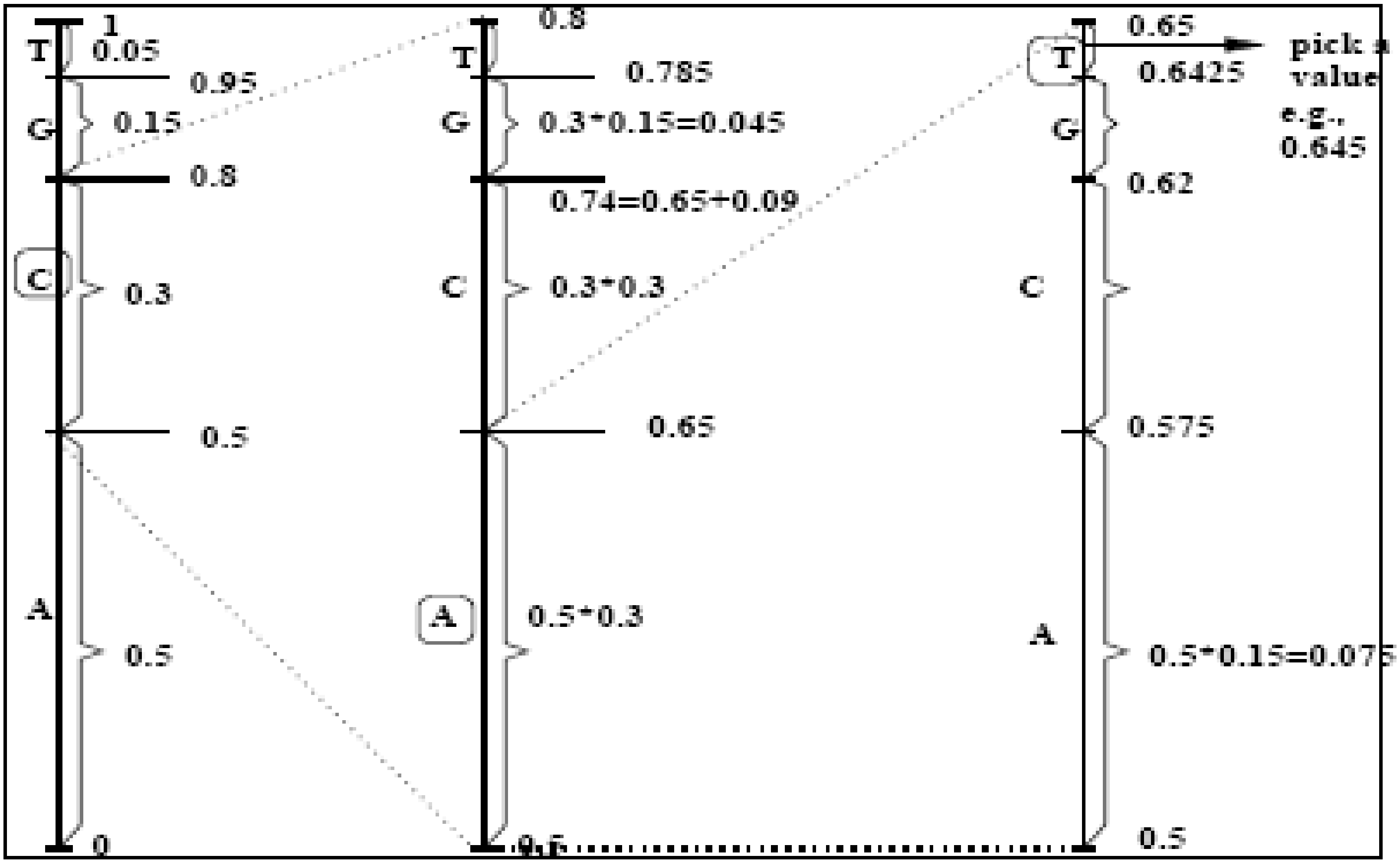
Arithmetic Coding

- Optimal algorithm as Huffman coding wrt compression ratio
- Better algorithm than Huffman wrt transmitted amount of information
  - Huffman – needs to transmit Huffman tables with compressed data
  - Arithmetic – needs to transmit length of encoded string with compressed data

Binary Arithmetic Coding

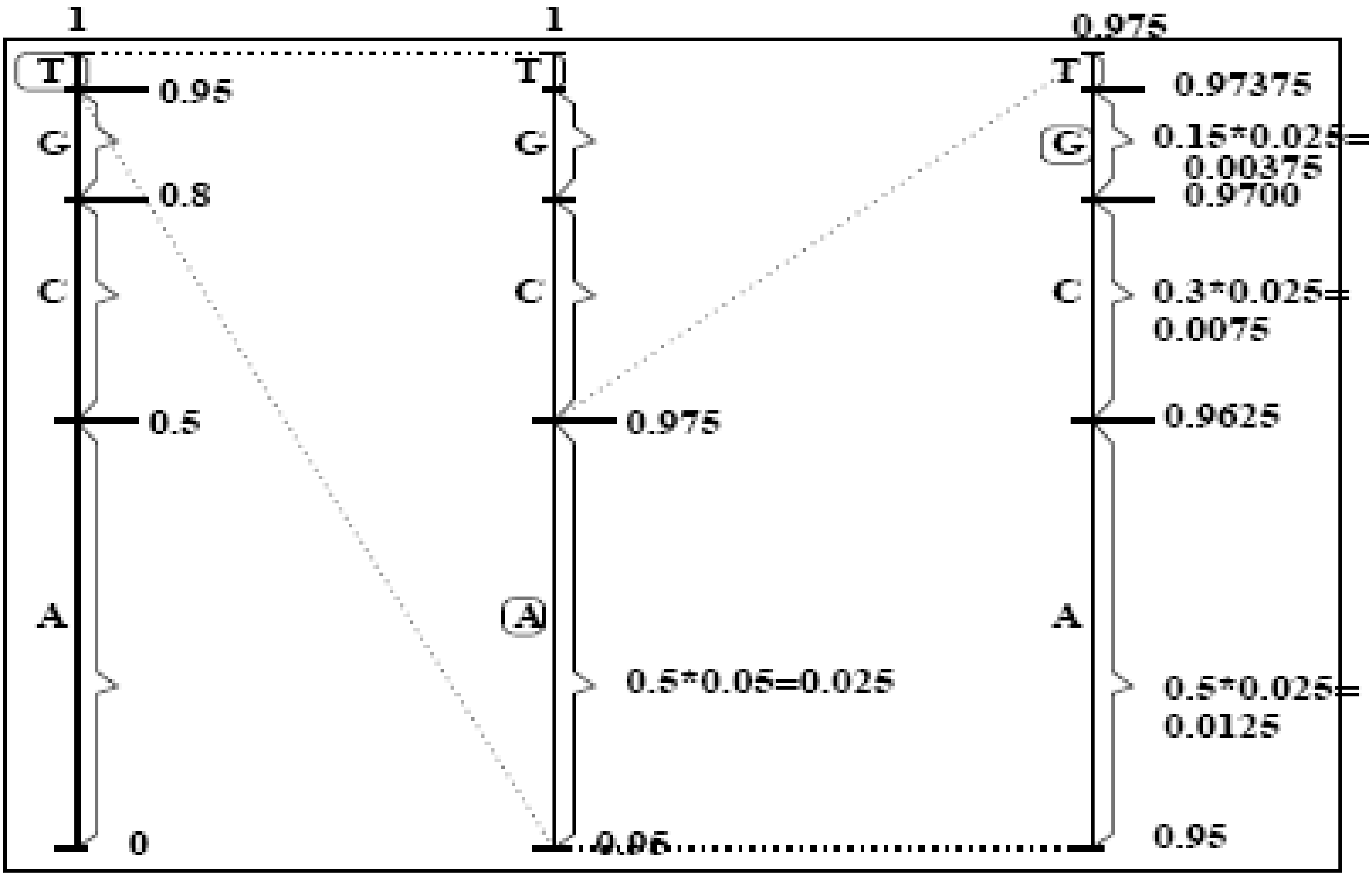
- Each symbol is coded by considering the prior data
- Encoded data must be read from the beginning, there is no random access possible
- Each real number (< 1) is represented as binary fraction
  - $0.5 = 2^{-1}$ (binary fraction = 0.1); $0.25 = 2^{-2}$ (binary fraction = 0.01), 0.625 = 0.5+0.125 (binary fraction = 0.101) ….

P(A)=0.5, P(C) = 0.3, P(G) = 0.15, P(T) = 0.05 => Encode CAT



Send Value: 0.645

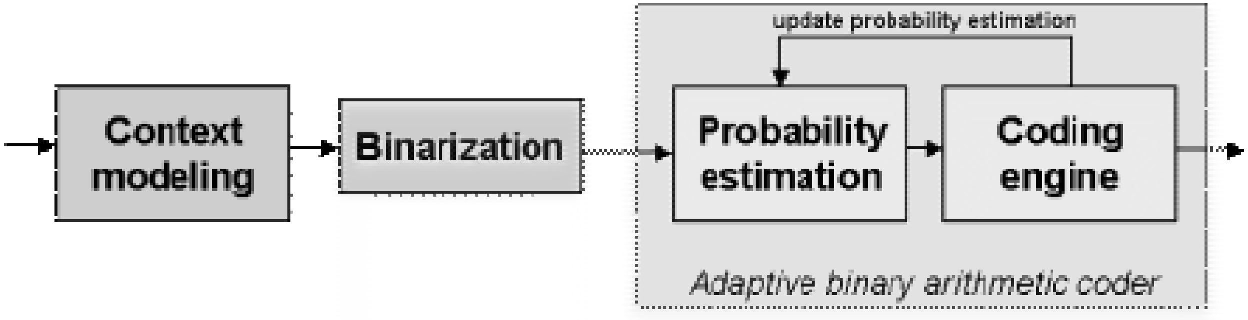
Given: P(A)=0.5, P(C) = 0.3, P(G) = 0.15, P(T) = 0.05 => Decode: 0.9715



Decoded Word: TAG

CABAC

- CABAC mode improves compression efficiency by ~10% relative to CAVLC
- CABAC much more computationally complex



H.264 P-Slices

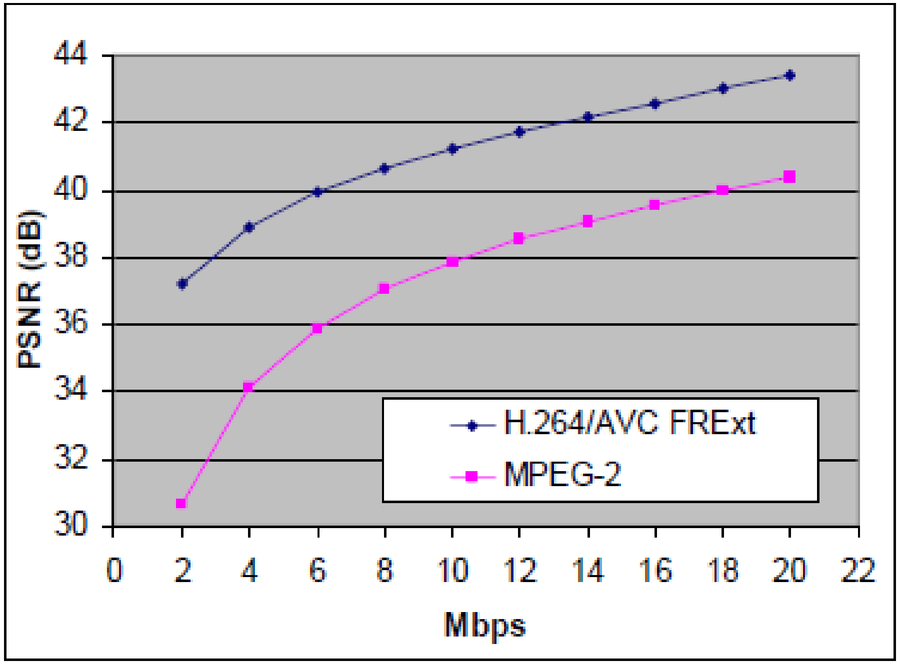
- Temporal prediction is used with estimation of motion between pictures
- Motion is estimated at 16x16 macro-block level or by partitioning macro-block into smaller regions 16x8, 8x16, 8x8, 8x4, 4x8, 4x4
- **One motion vector** can be sent for **each sub-macro-block partition**
- Motion is estimated from multiple pictures that lie **either in past or in future** in display order

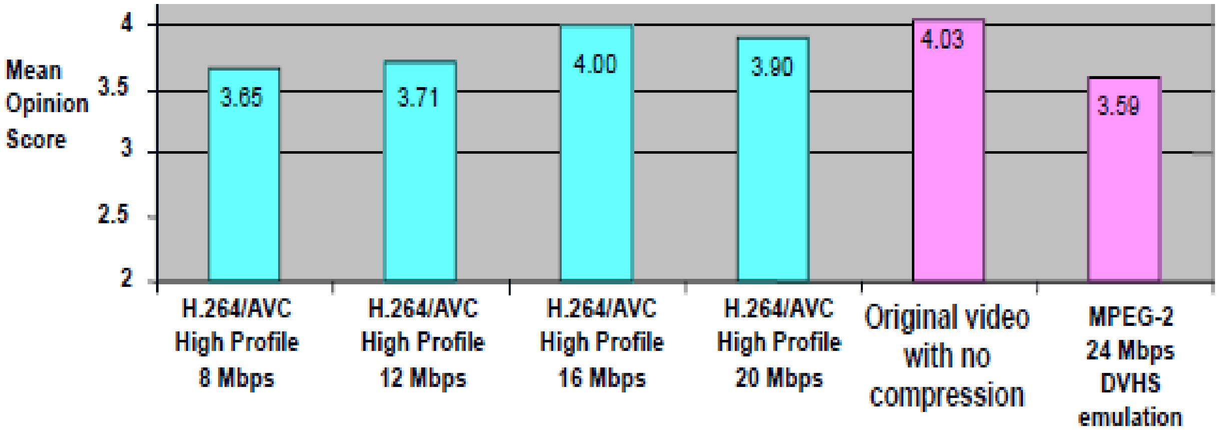
H.264 B-Slices

- Temporal prediction with two motion vectors representing two estimates of motion per macro-block or sub-macro-block
- Consider any reference picture in future or past in display order
- **Weighted prediction concept** is further extended in usage of **weighted average** between two predictions

Performance

- H.264 AVC is a standard that has the potential to improve efficiency as well as subjective perception





H.265 - HEVC

HEVC / H.265

- HEVC – High Efficiency Video Coding
- The same as H.265 and MPEG-H Part 2
- Successor of AVC (H.264 or MPEG-4 Part 10)
  - Extends concepts (usage of parallel algorithms and simplification of many H.264 concepts)
  - Doubles the data compression ratio at the same level of quality
  - Improves video quality at the same bit rate
  - Supports 8K UHD

HEVC (2)

- Proposal for HEVC issued in 2010
- HEVC version 1 was approved in 2013
- HEVC version 2 was approved in 2014
- Now we have HEVC version 4 (approved 2016)

Subjective Video Performance

Video coding standard	Average bit rate reduction compared with H.264/MPEG-4 AVC HP			
	480p	720p	1080p	2160p
HEVC	52%	56%	62%	64%

https://en.wikipedia.org/wiki/High_Efficiency_Video_Coding

Final remarks

- VP9 – Google video compression format used in YouTube
  - Competes with HEVC/H.265
  - Protected by Patent (many parts)
  - Uses refinements and augmentations of H.264/MPEG-4/Part 10 algorithms

References

- Slides based on Literature Reference
  - Gary Sullivan et al, “The H.264/AVC Advanced Video Coding Standard: Overview and Introduction to the Fidelity Range Extensions”, SPIE Conf. on Applications of Digital Image Processing, 2004.
  - Zhang Nan et al., “Spatial Prediction Based Intra Coding”, IEEE ICME 2004
  - Technology Overview, “AVC-Intra (H.264 Intra) Compression”, Panasonic Broadcast, 2007

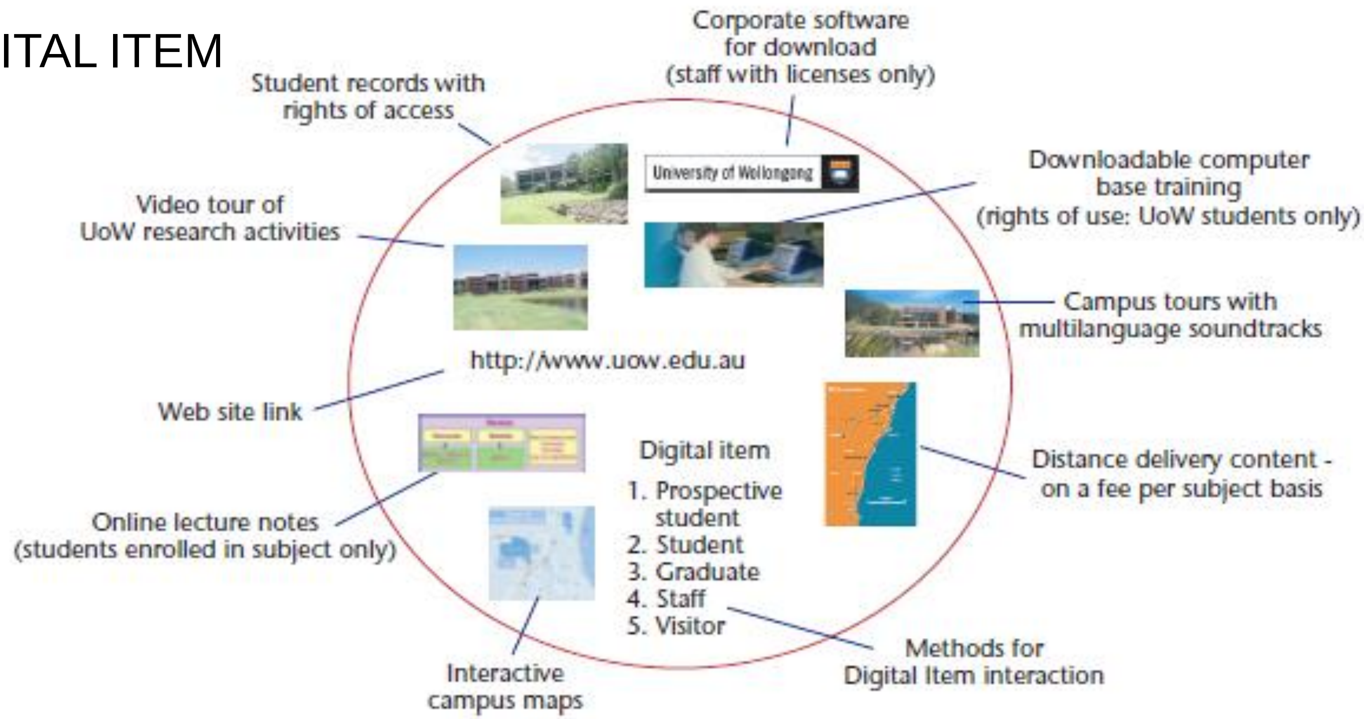
ADDITIONAL SLIDES

MPEG-21

MPEG-21 Objectives

- Extension of MPEG standard from client-server based to peer-to-peer applications
- Interoperability of
  - Content structure and set of communication processes
- Relation between ‘what’ and ‘who’ within multimedia framework

DIGITAL ITEM



- ‘what’ is “Digital Item”

  Structured digital object with standard representation, identification, metadata

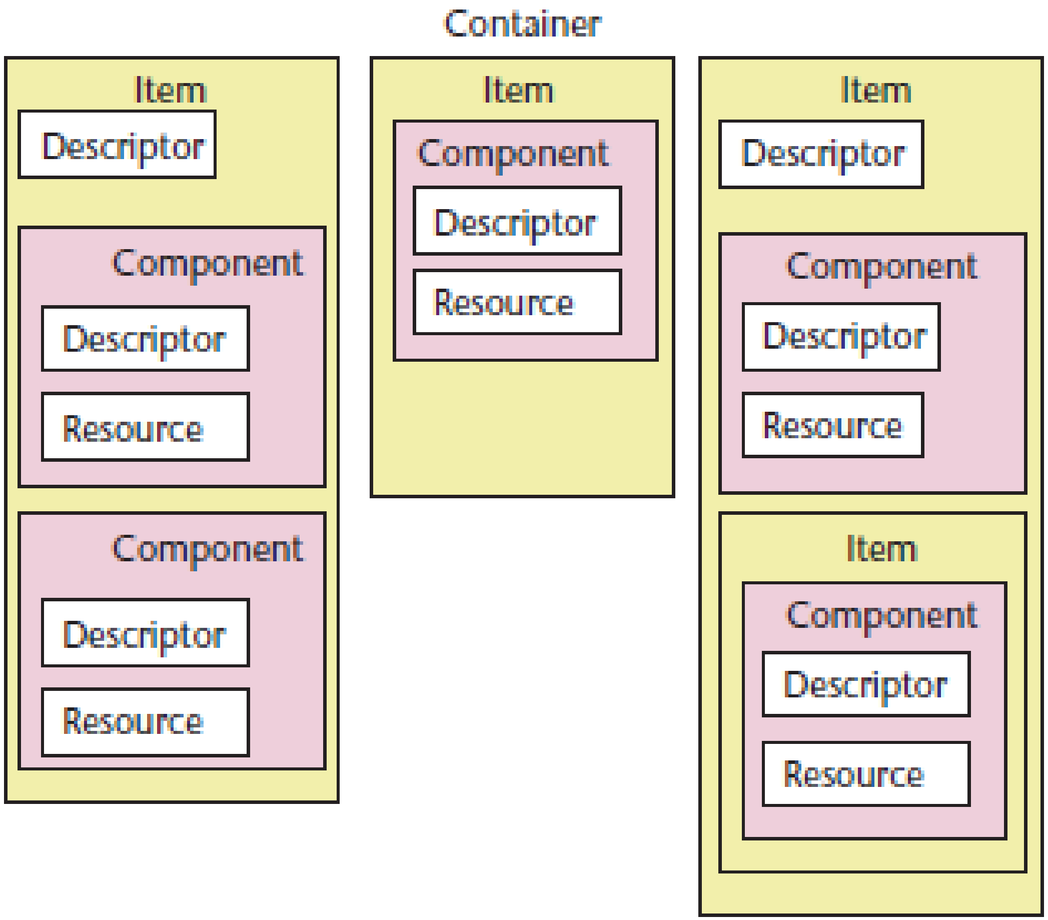
- ‘who’ is user who interacts in MPEG-21 environment

  User of Digital Item

MPEG-21 Parts

- Digital Item Declaration
  - Digital Item Declaration Language
- Digital Item Identification
- Digital Item Adaptation
- File format
- Digital Item Processing
- Rights Management

Digital Item Declaration model elements and their relationships



Digital Item Adaptation (1)

- Goal of MPEG-21 – achieve **interoperable transparent access** to distribute advanced multimedia content by shielding users from
  - network and terminal installations,
  - management, and
  - implementation issues!!!

Digital Item Adaptation (2)

- To achieve goal need
  - Resource adaptation engine
  - Description adaptation engine
  - DID adaptation engine
- Usage of description of content, content format description, usage environment