

# Multimedia Concept & Topics

- Multimedia Concept
- Multimedia Computing
- Multimedia Classification
- Multimedia Topics
- Multimedia Driving Forces
- Multimedia Applications
  
- Course Outline

# What is Multimedia

- **Multi:** more than one
- **Medium** (singular): middle, intermediary, mean
- **Media** (plural): means for conveying information
  - Media in the press, newspaper, radio and TV context - mass media
  - Media in communications: cables, satellite, network – transmission media
  - Media in computer storage: floppy, CD, DVD, HD, USB – storage media
  - Media in HCI context: text, image, audio, video, CG – interaction media
- **Multimedia:** refers to various information forms text, image, audio, video, graphics, and animation in a variety of application environments

Multimedia ... :

product, application, technology, platform, board, device, network  
computer, system, classroom, school, ...

Word “multimedia” is widely used to mean many different things

# What is Multimedia in terms of Computing

Computing: Computer-based technologies and applications

→ What computers? → Various forms of computers/devices!

In terms of computing, four fundamental multimedia attributes:

- Digitized: All media including audio/video are represented in digital format
- Distributed: The information conveyed is remote, either pre-produced and stored or produced in realtime, distributed over networks
- Interactive: It is possible to affect the information received, and send own information, in a non-trivial way beyond start, stop, fast forward
- Integrated: The media are treated in a uniform way, presented in an orchestrated way, but are possible to manipulate independently

## Definition of Multimedia:

Computer-based techniques of text, images, audio, video, graphics, animation, and any other medium where every type of information can be represented, processed, stored, transmitted, produced and presented digitally.

**This course focus → Audio and Video**

# Benefits of Multimedia

Some authors claim that humans get their information in the following way:

- more than 80 % by sight - of which 20 % is remembered
- 11 % by hearing - of which 30 % is remembered
- 3.5 % by smell
- 1.5 % by touch and taste.

... where 50 % of what is both seen and heard is remembered

... further 80 % of what is seen, heard and done, is remembered

That is, multiple, media, and interactive should be a good thing



# A Classification of Multimedia

- Text - ASCII/Unicode, HTML, Postscript, PDF
- Audio – Sound, music, speech, structured audio (e.g. MIDI)
- Still Image - Facsimile, photo, scanned image
- Video (Moving Images) – Movie, a sequence of pictures
- Graphics – Computer produced image
- Animation – A sequence of graphics images
  
- Discrete Media (DM, Static): text, image, graphics
- Continuous Media (CM, Dynamic): audio, video, animation
  
- Captured vs Synthesized media
- Standalone vs Networked media

# System Implications of Multimedia

Multimedia imposes new requirements on all parts of the system architecture:

- Representation
  - digitization and coding (compressing)
- Storage
  - database, larger volumes and new access patterns
- Processing
  - OS, scheduling, indexing, searching
- Understanding
  - speech/object recognition, content analysis
- Production
  - more complex authoring and user interface software
- Presentation
  - user perception, user friendly in HCI (Human Computer Interface)
- Protection
  - media encryption, copyright, privacy
- Distribution
  - media delivery and broadcast
- Communication
  - media transmission over network/internet, session control

# Why is Multimedia Important ?

- Digital audio/video is revolutionizing music, film, game, and video & audio industries
- Convergence of computers, telecommunication, radio, and TV
  - Caused by technology and competition
  - Dramatic changes in products and infrastructure
- New application potential
  - Huge potential markets
  - Improving our lives (learning, entertainment, and work)
- Interesting technical issues

Multimedia has become hot and been emerged in CS/IT since 1985

# Forces Driving the Multimedia Revolution

- Evolution of communication and data networks: Increasing availability of bandwidth on demand in the office, home, road.... Thanks to high-speed data modems, cable modems, hybrid fiber-coax systems, xDSL, wireless.
- Ubiquitous access to network. Via local-area networks (LAN), wireline and wireless networks, Internet, world wide web, → “anywhere, anytime”.
- Fast processor and large capacity storage devices, including 3-D hardware. Moore’s law: computation and memory capacity of chips doubles every 18 months or so.

## Forces Driving the Multimedia Revolution (Cont...)

- New algorithms and data structures. Compression techniques, graphics, computer vision, speech understanding...
- Smart terminals such as digital phones, screen phones, multimedia PC's, web-TV, personal digital assistants, etc., accessing and interacting the network with wired and wireless connections.
- And of foremost importance, the digitization of virtually any device : cameras, video capture and playback devices, handwriting terminals, sound capture, etc., together with plug-and-play standards; and the digitization of text/audio/video documents and libraries that allows better communications, storage, and fast access and browsing.

# Technological Aspects

- Techniques for compressing and coding the various media: models, algorithms, forms, standards, etc.
- Communications aspects: downloading and streaming techniques, synchronization, layering of signals, issues involved in the definition of QoS (quality of service.)
- Techniques for accessing multimedia signals by providing tools that match user to the machine: “natural” spoken language queries, media conversion tools and multimodal user interface (speech recognition, lip reading, face tracking, OCR,..), agents that monitor the multimedia sessions and provide assistance in all phases of access and utilization.
- Techniques for organizing, storing and retrieving multimedia, for searching and browsing individual multimedia documents and libraries.

# Are Multimedia Applications Hard?

- Large size of multimedia objects
  - Speech: 8000 samples/s – **8 Kbytes/s**
  - CD audio: 44,100 samples/sec, 2 bytes/sample, stereo audio – **176 Kbytes/s**
  - NTSC video: 30 frames/s, 640x480 pixels, 3 bytes/pixel – **30 Mbytes/s**  
(too big, 2-8 Mbits/s if compressed)
  - More storage required
  - More main memory
  - 10-30 GB secondary storage
  - TB's of tertiary storage
- Real-time performance requirements

# Are Multimedia Applications Hard? (Cont...)

- Higher data rates
  - Fast I/O subsystem (SCSI, fiber channel, HIPPI)
    - E.g., Ultra SCSI2 – 80 Mbytes/s
  - High speed backplane (PCI or faster)
  - Faster network (1-25Mbs per video stream)
    - 1-4 Gbits/s network
- Hardware CODEC, modified CPU (?), and modified frame buffer/graphics subsystem

*Essentially, new hardware and software*

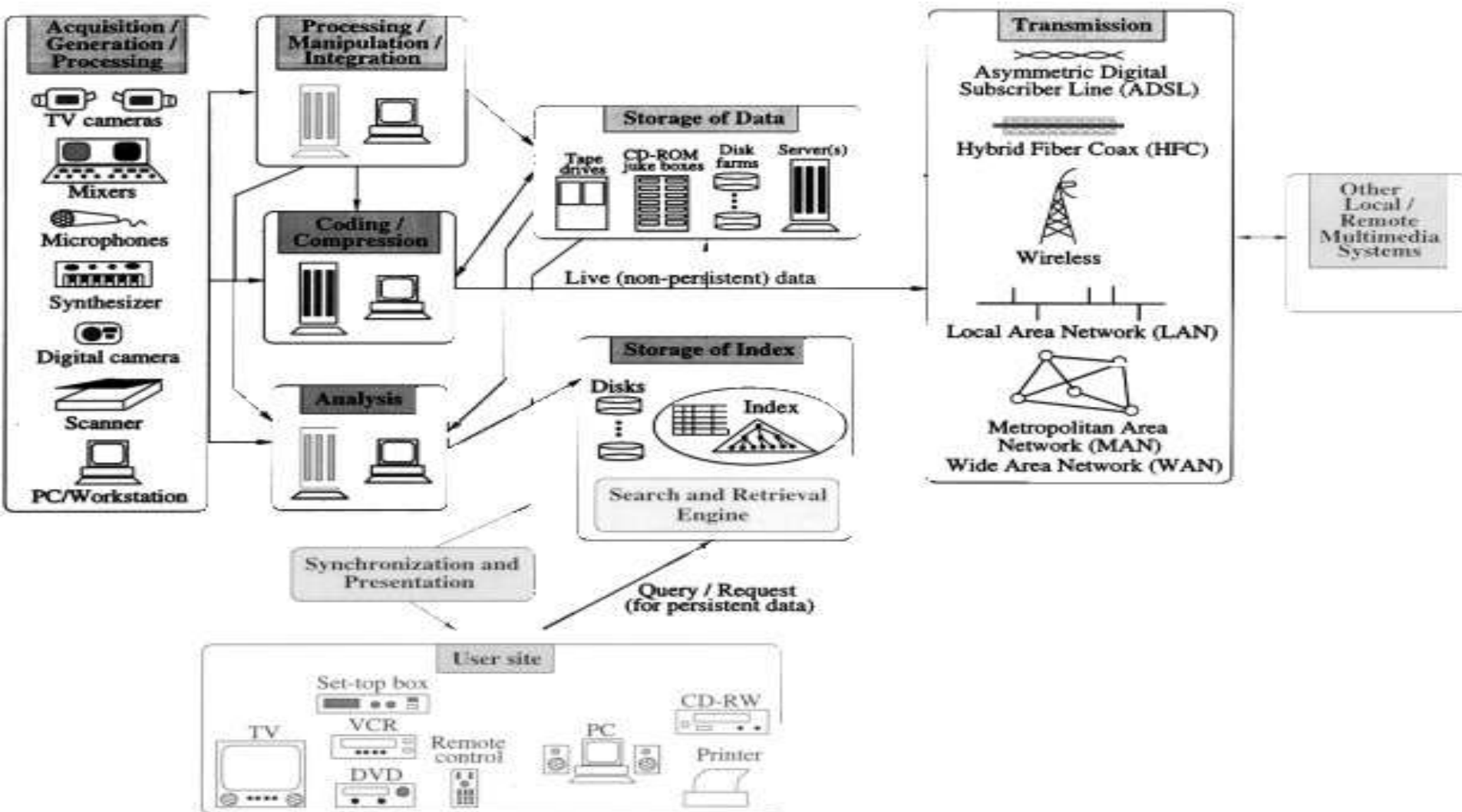
*Further, audio/image/video "**content**" processing*



# Examples of Multimedia Applications

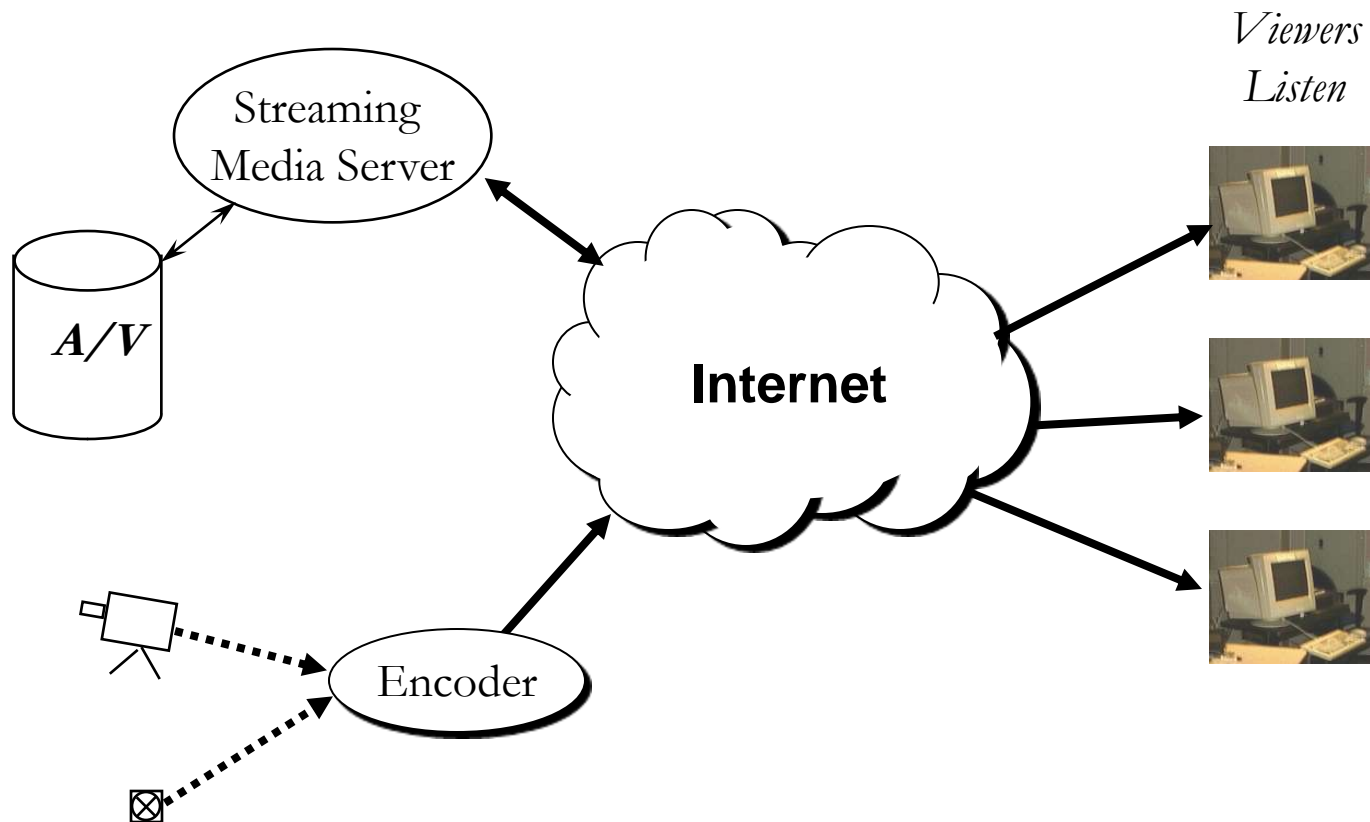
- Residential services
  - Video-On-Demand
  - Video phone, A/V conferencing
  - Home shopping
- Business services
  - Corporate education
  - E-business
- Education
  - Digital libraries
  - Distance learning
- Science and technology
  - Virtual environment
  - Scientific visualization, prototyping
- Entertainment
  - Games
  - Interactive TV
  - Post production of movie and music
- Medicine, Web applications, etc.

# General Overview of a Multimedia System

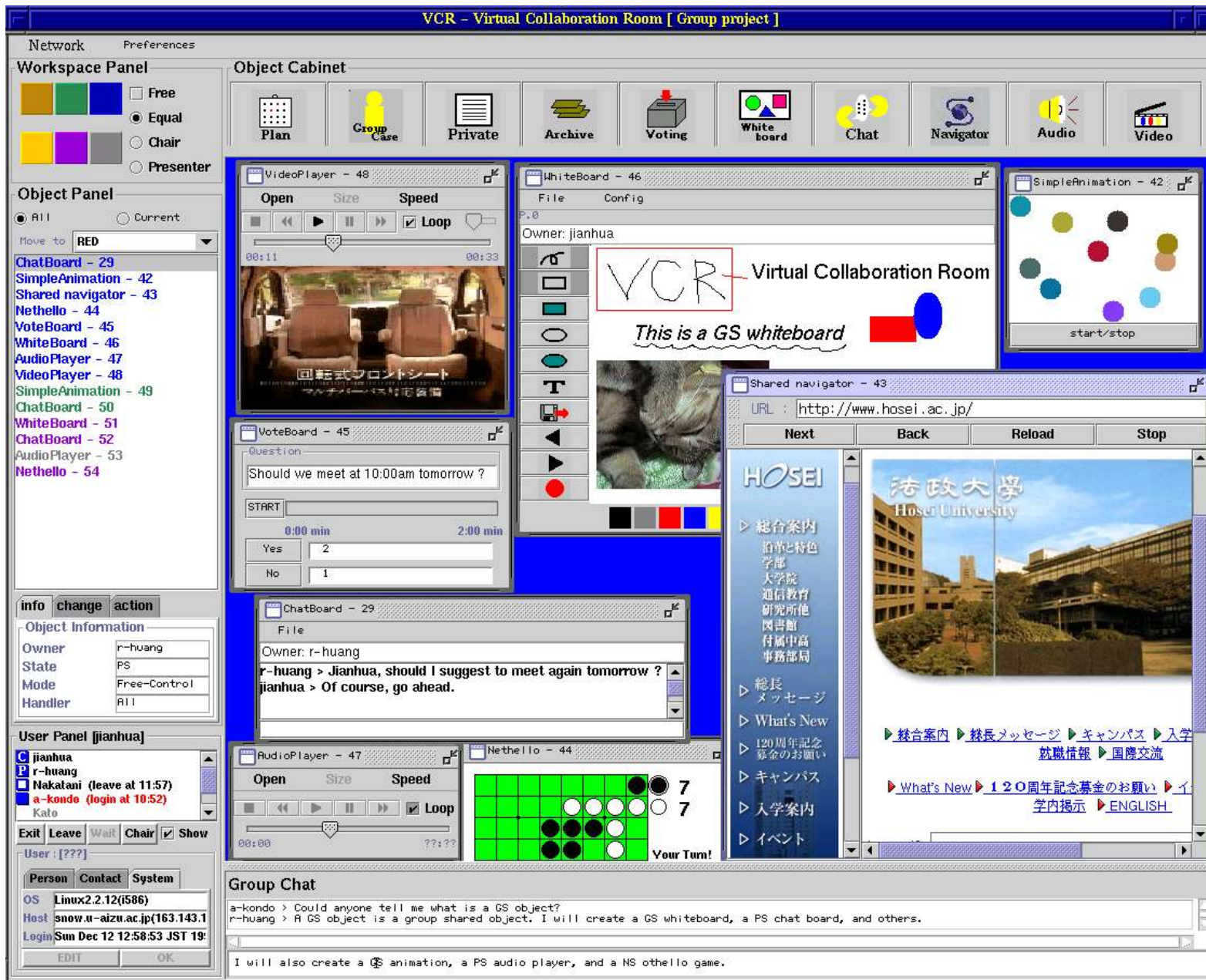




# Audio/Video Broadcast over the Internet

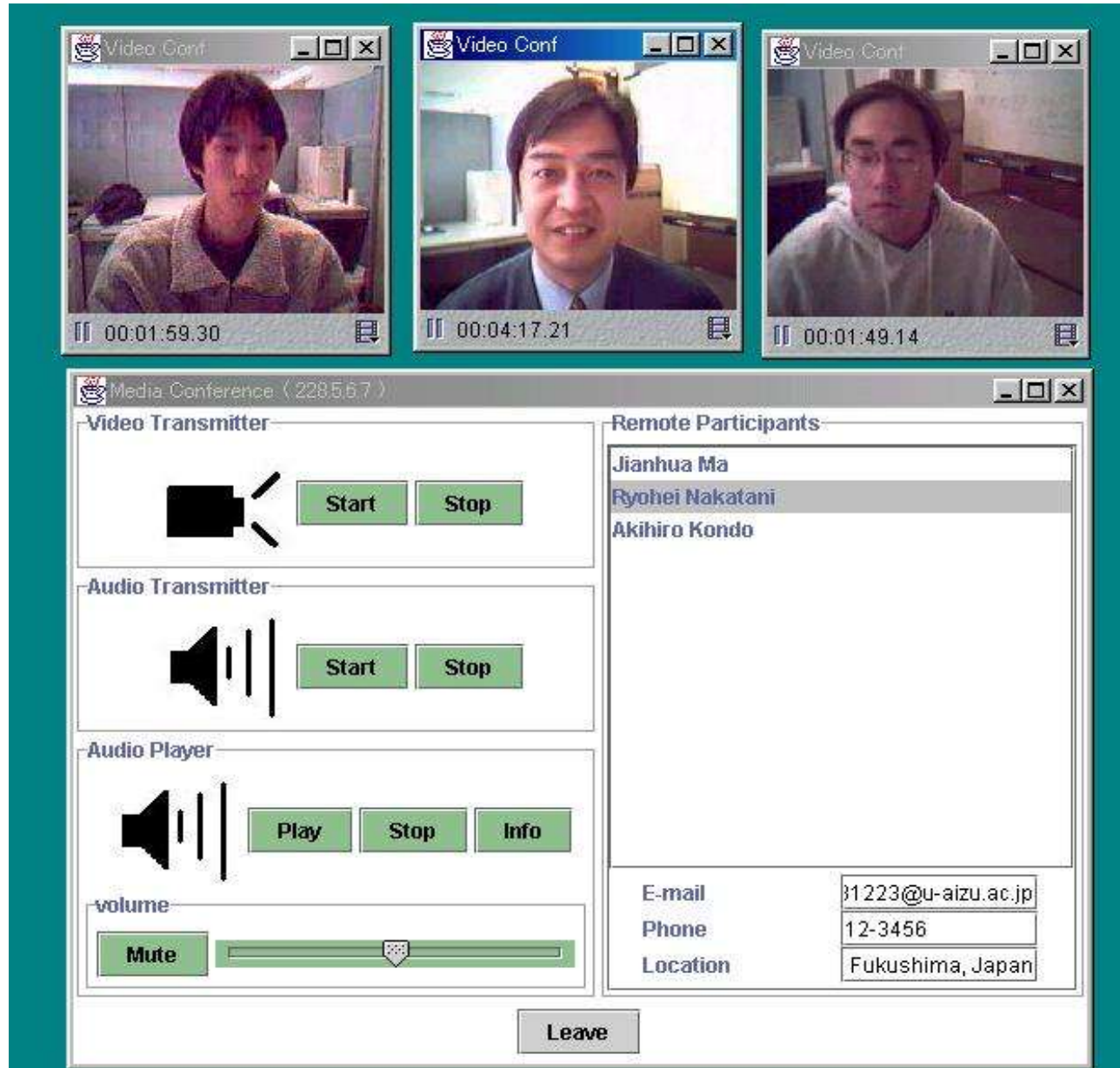


# Shared Applications and CSCW





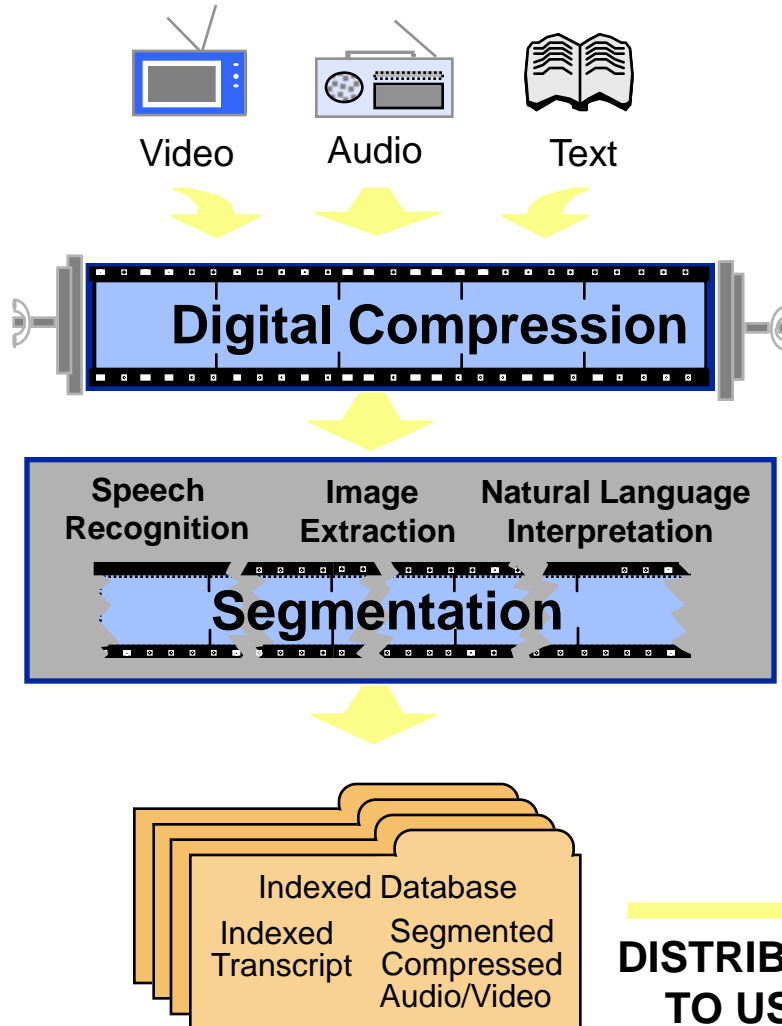
# Desktop Audiovisual Conferencing



# Digital Library

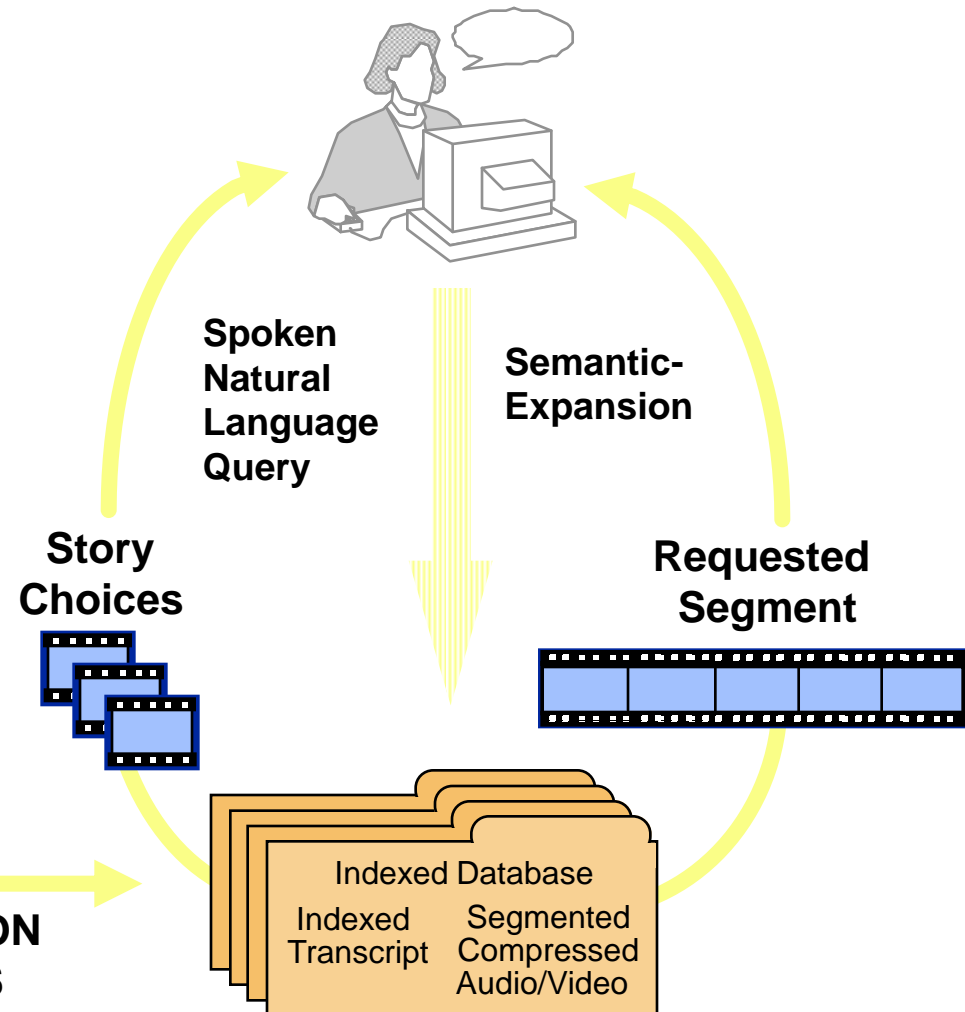
## Library Creation

### Offline



## Library Exploration

### Online



# Teaching Plan

## Part I: Multimedia Fundamentals and Coding Techniques

Lesson 1. Multimedia Concept and Topics

Lesson 2. Audio Fundamentals

Lesson 3. Audio Coding and Standard

Lesson 4. Image/Video Fundamentals

Lesson 5. Image/Video Coding: JPEG and H.26x

Lesson 6. MPEG Coding Standards

Lesson 7. Review of Advanced MM Coding

Quiz Test I. Questions related to Part I

Report I. Summary of Audio and Video Coding, or  
A Study on a Specific Coding Technique



# Teaching Plan

## Part II: Multimedia Technologies and Applications

Lesson 8. Media Object Production

Lesson 9. Media Integration and Presentation

Lesson 10. Media Protection

Lesson 11. Media Retrieval

Lesson 12. Media Distribution Across Internet

Lesson 13. Media Communication - IP Telephony & Teleconference

Lesson 14. Mobile Multimedia Service over Wireless Networks

Report II. Summary of Multimedia Technologies, or  
A Study on a Specific Multimedia Technology

Quiz Test II. Questions related to Part II

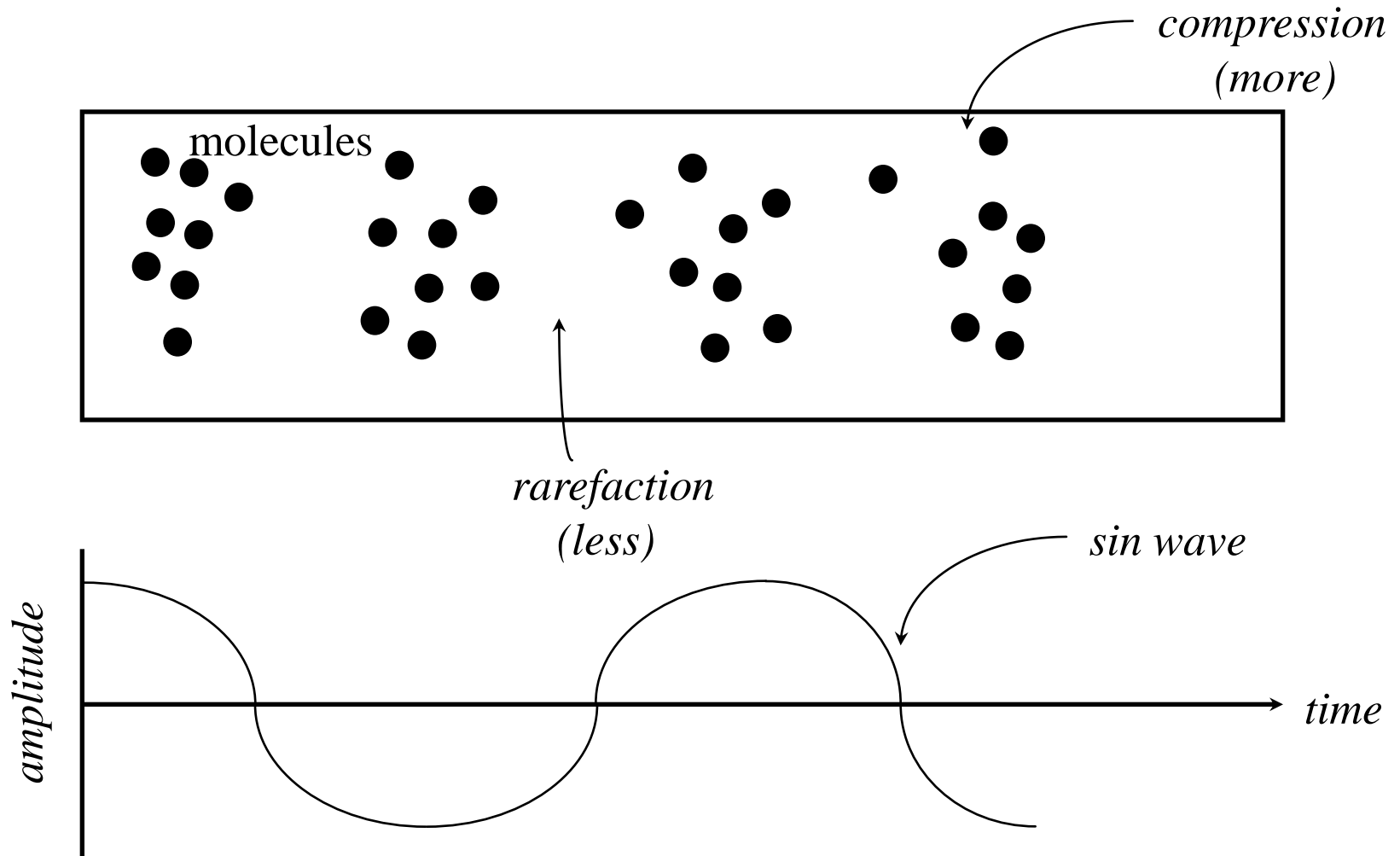
# Audio Fundamentals

- Sound, Sound Wave and Sound Perception
- Sound Signal
- Analogy/Digital Conversion
- Quantization and PCM Coding
- Fourier Transform and Filter
- Nyquist Sampling Theorem
- Sound Sampling Rate and Data Rate
- Speech Processing

# Sound

- Sound, sound wave, acoustics
  - **Sound** is a continuous wave that travels through a medium
  - **Sound wave**: energy causes disturbance in a medium, made of pressure differences (measure pressure level at a location)
  - **Acoustics** is the study of sound: *generation, transmission, and reception* of sound waves
- Example is striking a drum
  - Head of drum vibrates => disturbs air molecules close to head
  - Regions of molecules with pressure above and below equilibrium
  - Sound transmitted by molecules bumping into each other

# Sound Waves

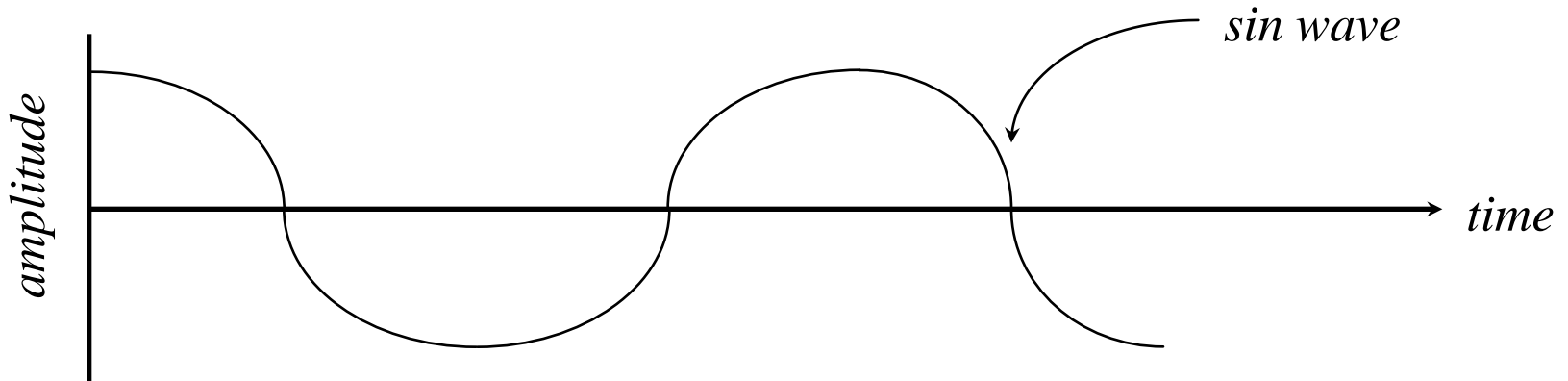


# Sound Transducer

- Transducer
  - A device transforms energy to a different form (e.g., electrical energy)
- Microphone
  - placed in sound field and responds sound wave by producing electronic energy or **signal**
- Speaker
  - transforms electrical energy to sound waves

# Signal Fundamentals

- Pressure changes can be periodic or aperiodic



- Periodic vibrations
  - cycle* - time for compression/rarefaction
  - cycles/second* - frequency measured in hertz (Hz)
  - period* - time for cycle to occur ( $1/\text{frequency}$ )
- Human perception frequency ranges of audio [20, 20kHz]

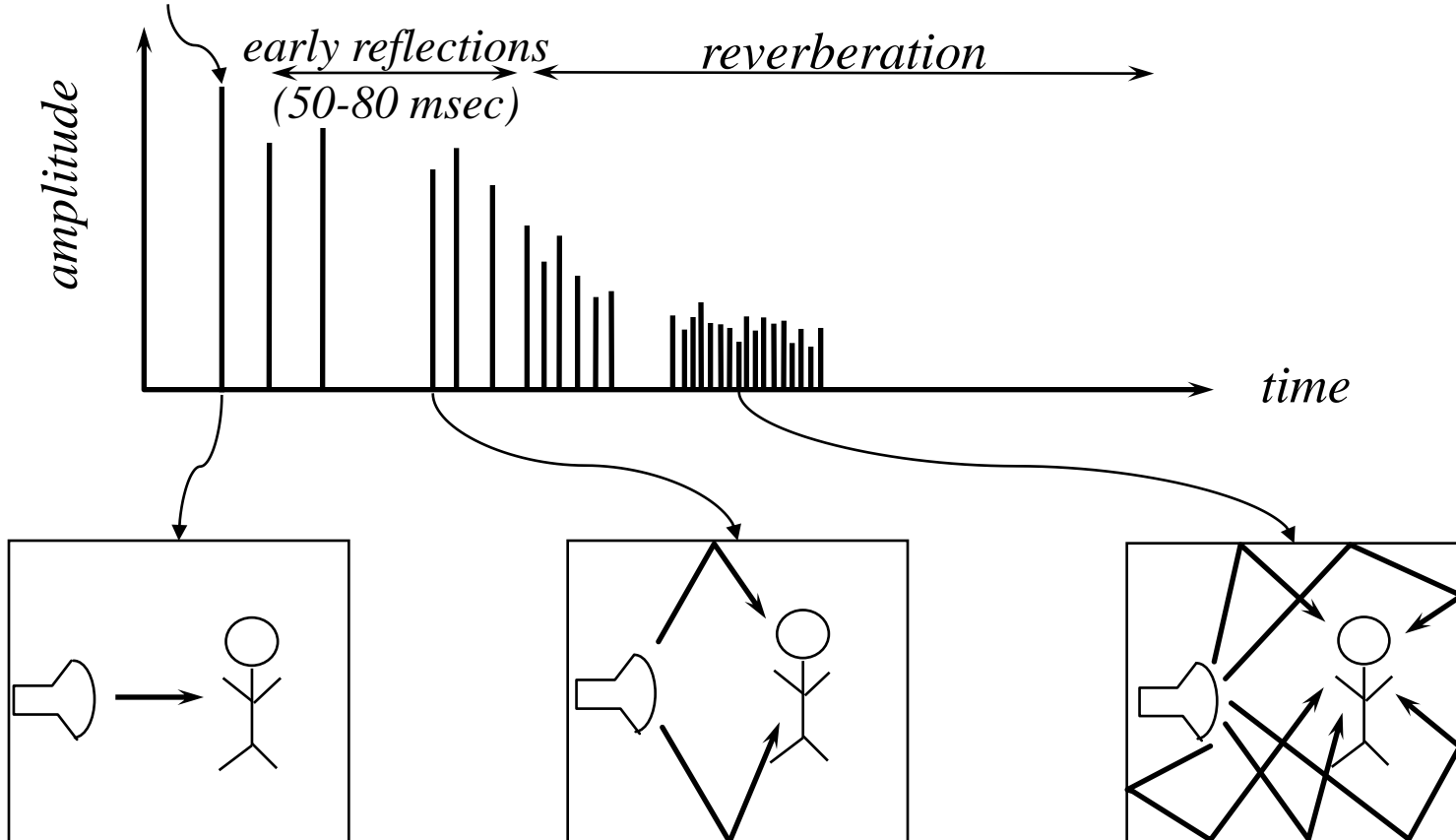
# Measurement of Sound

- A sound source is transferring energy into a medium in the form of sound waves (acoustical energy)
- Sound volume related to pressure amplitude:
  - *sound pressure level (SPL)*
- SPL is measured *in decibels* based on ratios and logarithms because of the extremely wide range of sound pressure that is audible to humans (from one trillionth= $10^{-12}$  of an acoustic watt to one acoustic watt).
  - $SPL = 10 \log (pressure/reference)$  decibels (dB)
  - where reference is  $2 \times 10^{-4}$  dyne/cm<sup>2</sup>
  - 0 dB SPL - no sound heard (hearing threshold)
  - 35 dB SPL - quiet home
  - 70 dB SPL - noisy street
  - 110 dB SPL - thunder
  - 120 dB SPL - discomfort (threshold of pain)

# Sound Phenomena

- Sound is typically a combination of waves
  - Sine wave is fundamental frequency
  - Other waves added to it to create richer sounds

*directed sound*



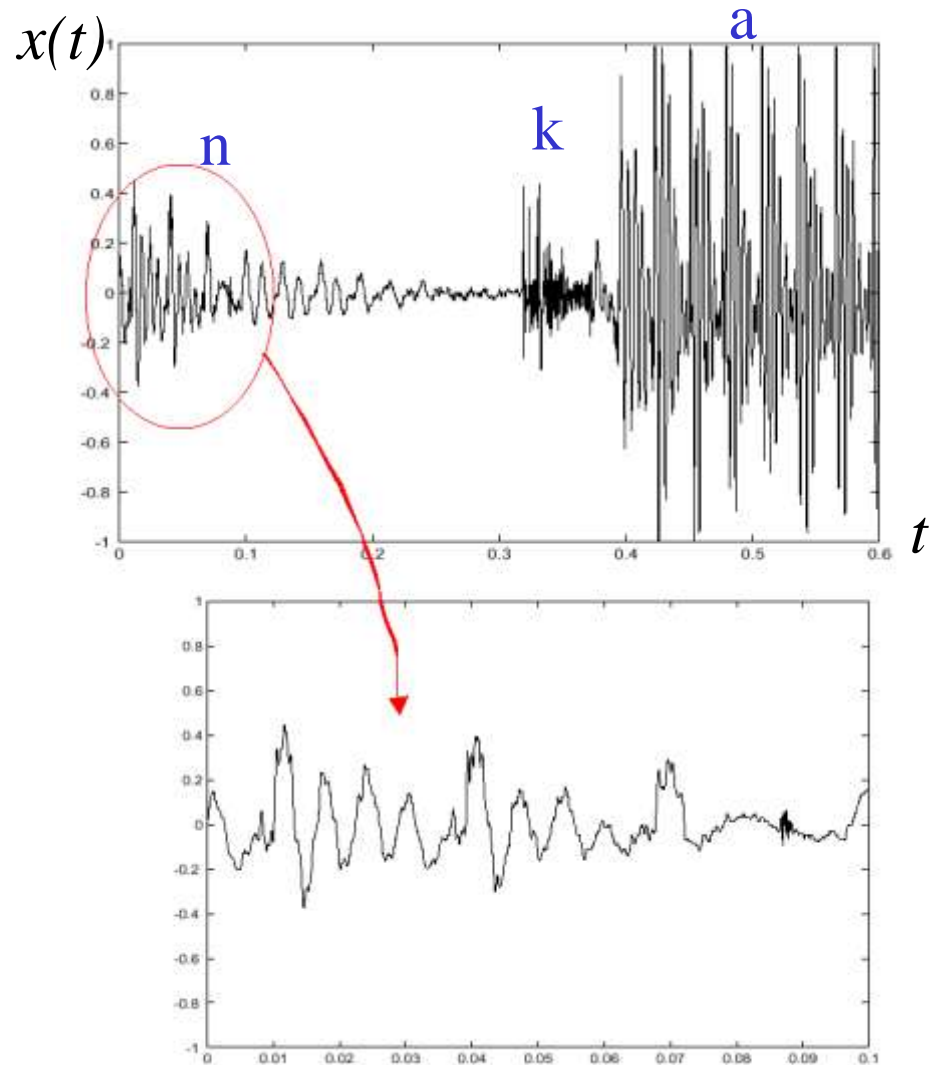


# Human Perception

- Perceptable sound intensity range 0~120dB
  - Most important 10~100dB
- Perceptable frequency range 20Hz~20KHz
- Humans most sensitive to low frequencies
  - Most important region is 2 kHz to 4 kHz
- Hearing dependent on room and environment
- Sounds masked by overlapping sounds
- Speech is a complex waveform
  - Vowels (*a,i,u,e,o*) and bass sounds are low frequencies
  - Consonants (*s,sh,k,t,...*) are high frequencies

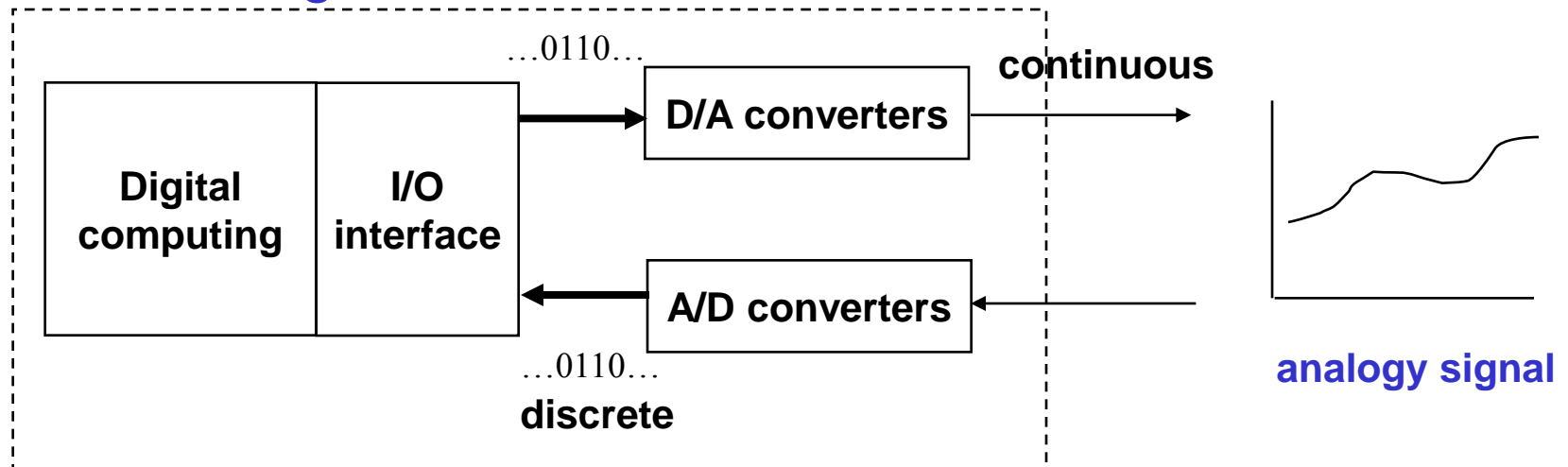
# Sound Wave and Signal

- For example, audio acquired by a microphone
  - Output voltage  $x(t)$  where  $t$  is time (continuous) and  $x(t)$  is a real number
  - One dimensional function
  - Called electronic **sound wave** or **sound signal**

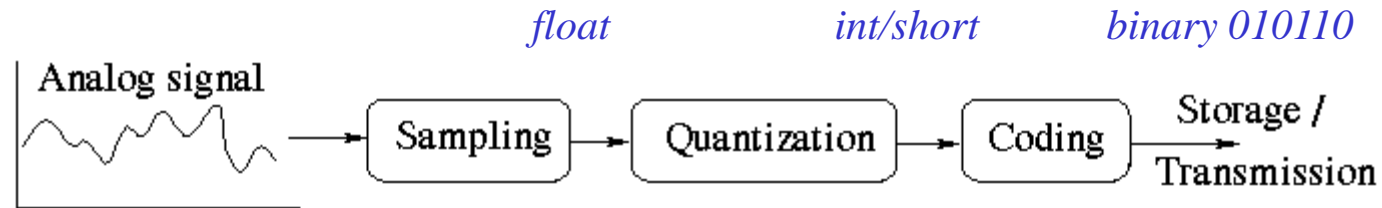


# Analog/Digital Conversion

- **Analog signal** (continuous change in both temporal and amplitude values) should be acquired in digital forms (**digital signal**) for the purpose of
  - Processing
  - Transmission
  - Storage & display
- *How to digitize ?*



# Process of AD Conversion

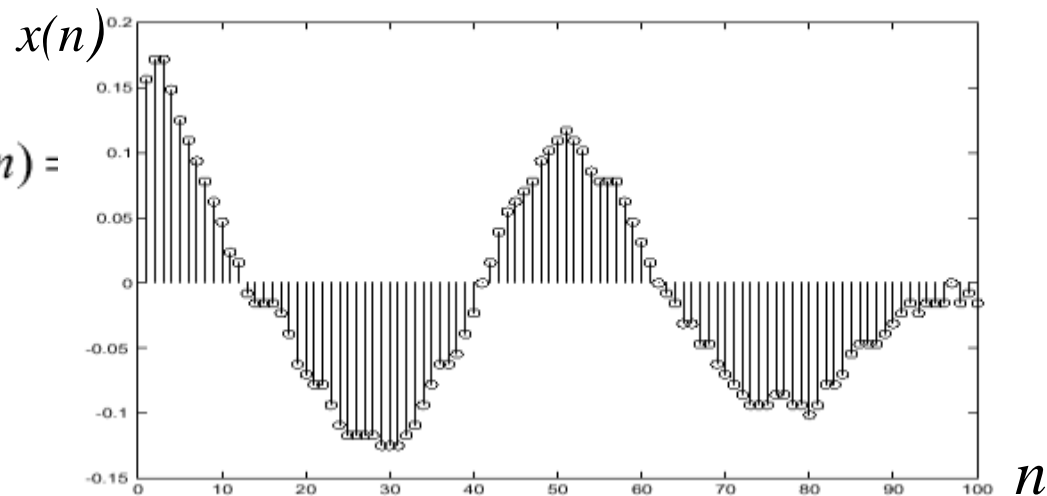
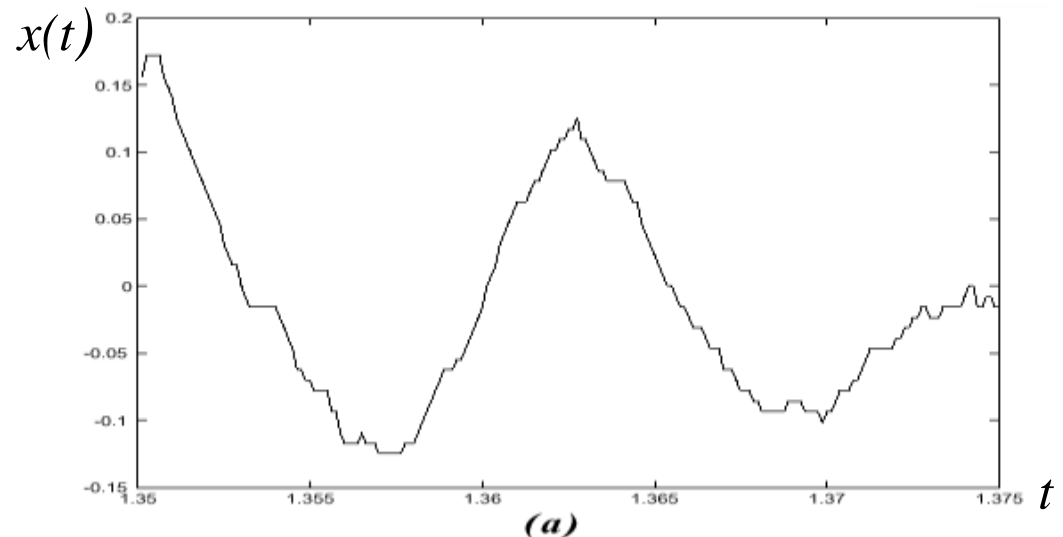


- **Sampling** (*horizontal*):  
 $x(n) = x(nT)$ ,  
 $T$  -- sampling period  
 Opposite transformation,  
 $x(n) \rightarrow x(t)$ , *interpolation*.
- **Quantization** (*vertical*):

$$\hat{x}(n) = Q(x(n))$$

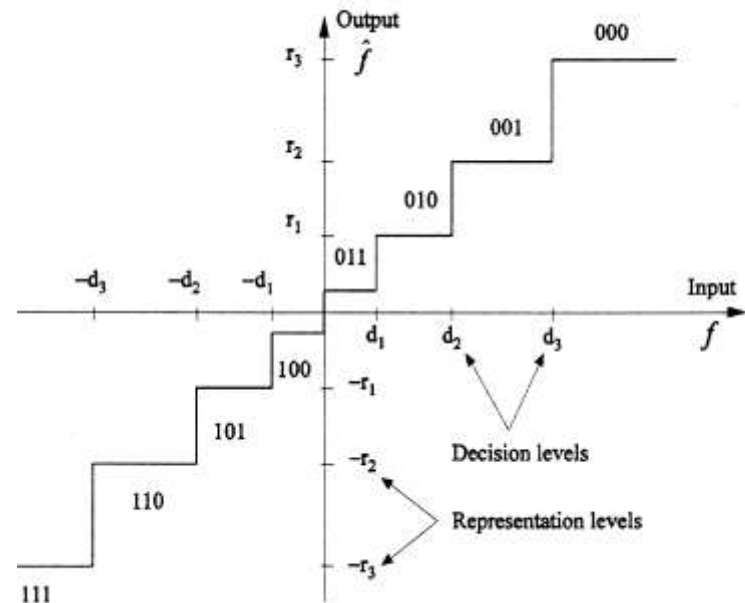
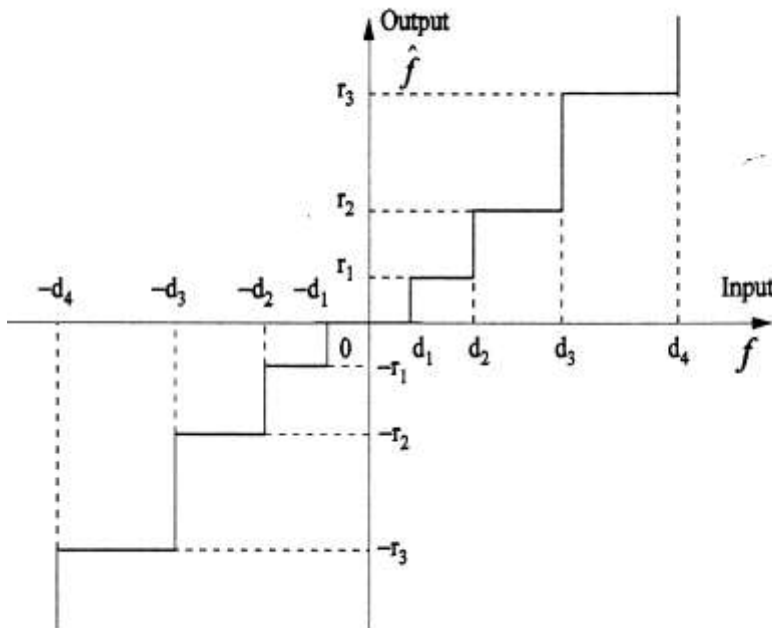
$Q()$  is a rounding function which maps the value  $x(n)$  (real number) into value in one of  $N$  levels (integer)

- **Coding**:  
 Convert discrete values to binary digits

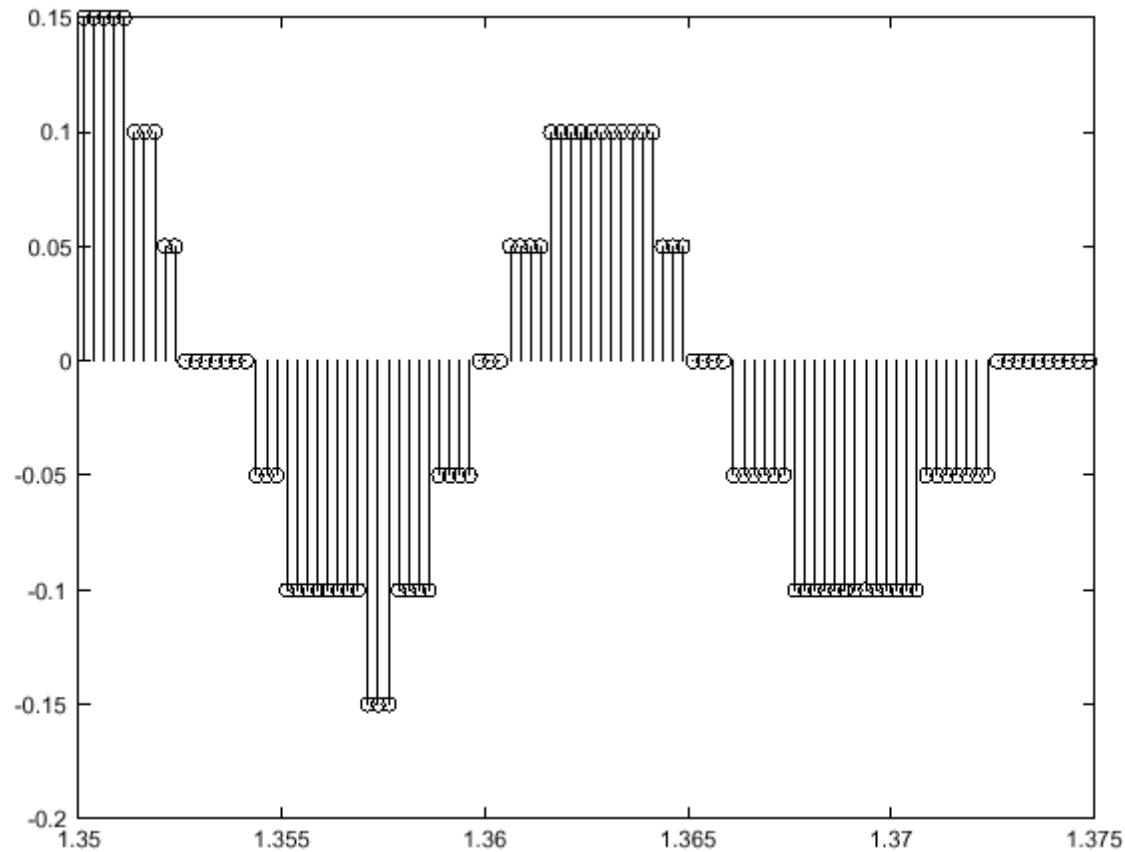


# Quantization and PCM Coding

- **Quantization**: maps each sample to the nearest value of N levels (*vertical*)
- **Quantization error** (or quantization noise) is the difference between the actual value of the analog signal at the sampling time and the nearest quantization interval value
- **PCM coding** (Pulse Code Modulation): Encoding each N-level value to a m-bit binary digit
- The precision of the digital audio sample is determined by the number of bits per sample, typically 8 or 16 bits



# Quantized Sound Signal



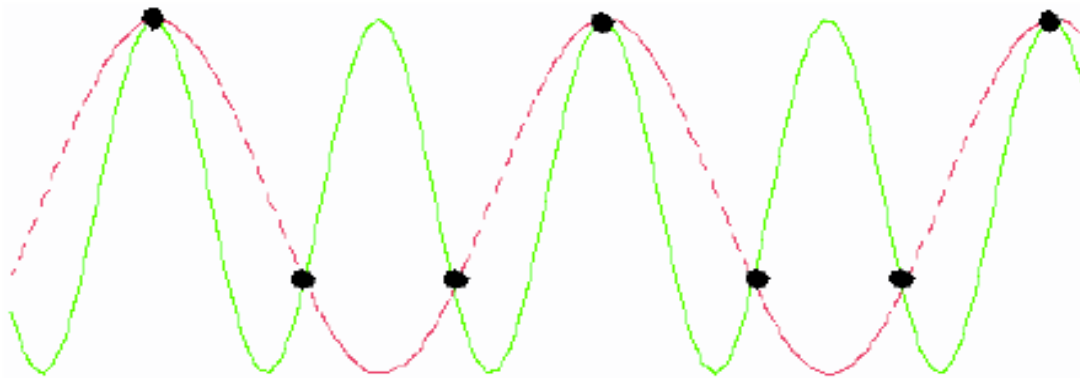
Quantized version of the signal

# Sampling Rate and Bit Rate

- Q. 1: What is the **bit-rate** (bps, bits per second) of the digitized audio using PCM coding? E.g.: CD.
- **Sampling frequency** is  $F=44.1$  KHz  
(**Sampling period**  $T=1/F=0.0227$  ms)
- Quantization with  $B=16$  bits ( $N=2^{16}=65,536$ ).
- Bit rate =  $BXF = 705.6$  Kbps = 88.2KBytes/s  
E.g.: 1 minute stereo music: more than 10 MB.
- Q.2: What is the “correct” sampling frequency  $F$ ? If  $F$  is too large, we have too high a bit rate. If  $F$  is too small, we have distortion or aliasing . Aliasing means that we loose too much information in the sampling operation, and we are not able to reconstruct ( interpolate ) the original signal  $x(t)$  from  $x(n)$  anymore.

# Nyquist Sampling Theorem

- Intuitively, the more samples per cycle, the better signal
- A sample per cycle -> constant
- 1.5 samples per cycle -> *aliasing*



- *Sampling Theorem*: a signal must be sampled at least twice as fast as it can change (2 X the cycle of change: *Nyquist rate*) in order to process that signal adequately.



# Fourier Transform

- Fourier transform tells how the energy of signal distributed along the frequencies

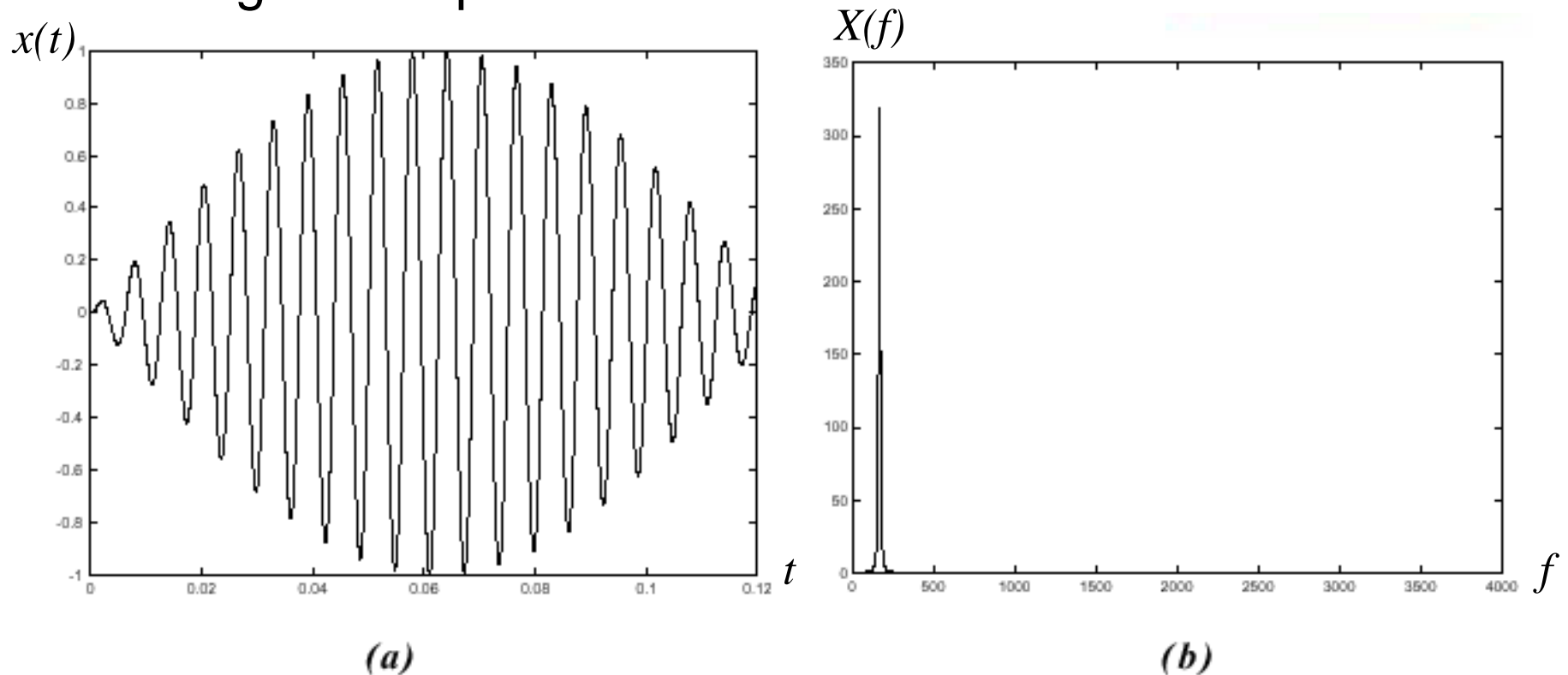
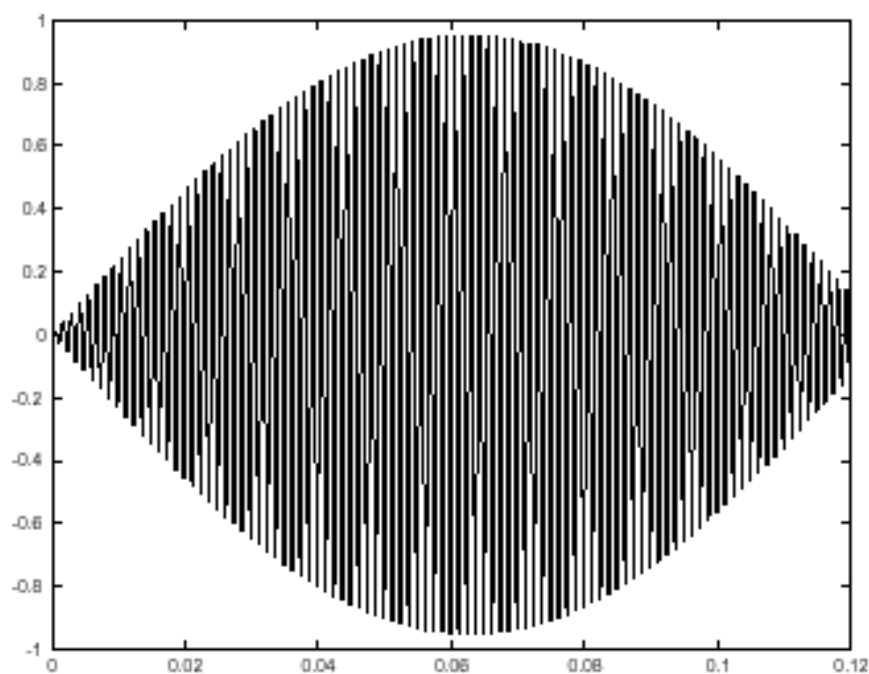
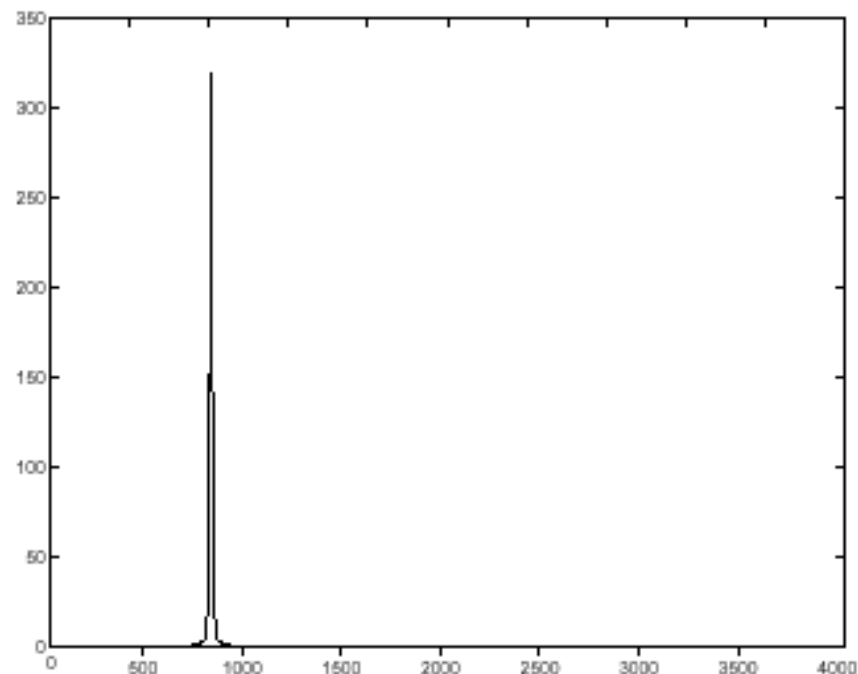


Figure 5: (a): A tone at 200 Hz. (b): Its Fourier Transform.

# Fourier Transform (Cont...)



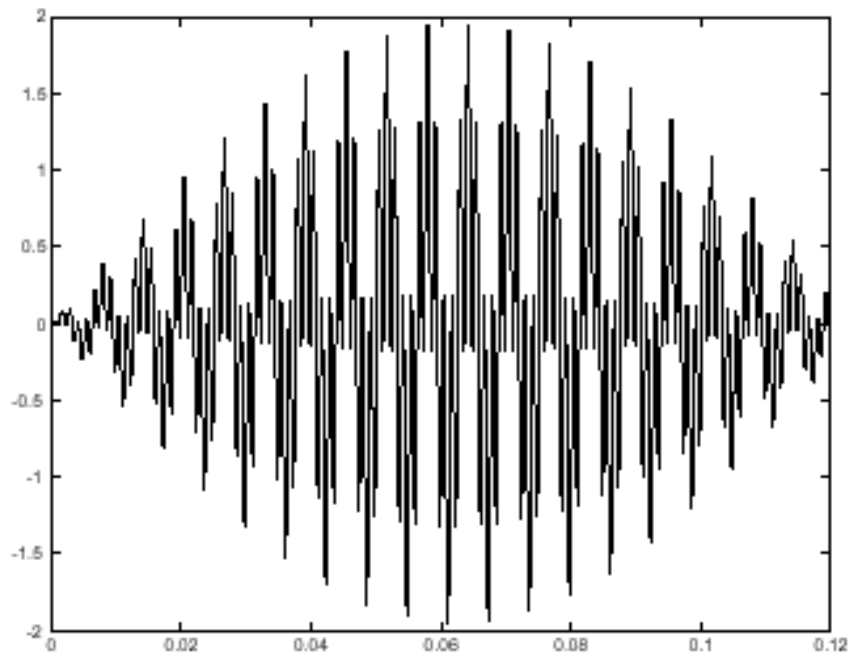
(a)



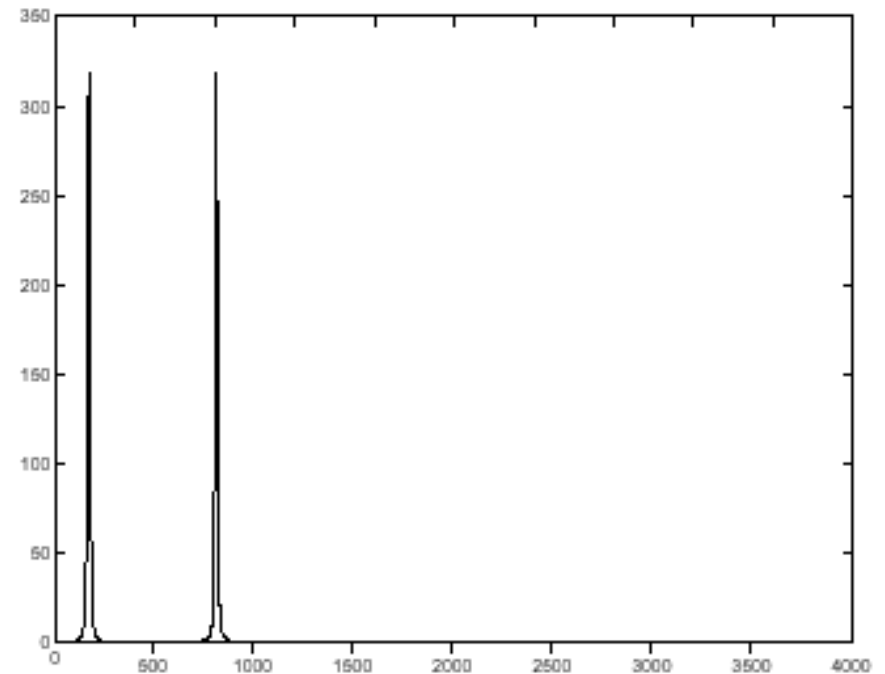
(b)

*Figure 6: (a): A tone at 800 Hz. (b): Its Fourier Transform.*

# Fourier Transform (Cont...)



(a)



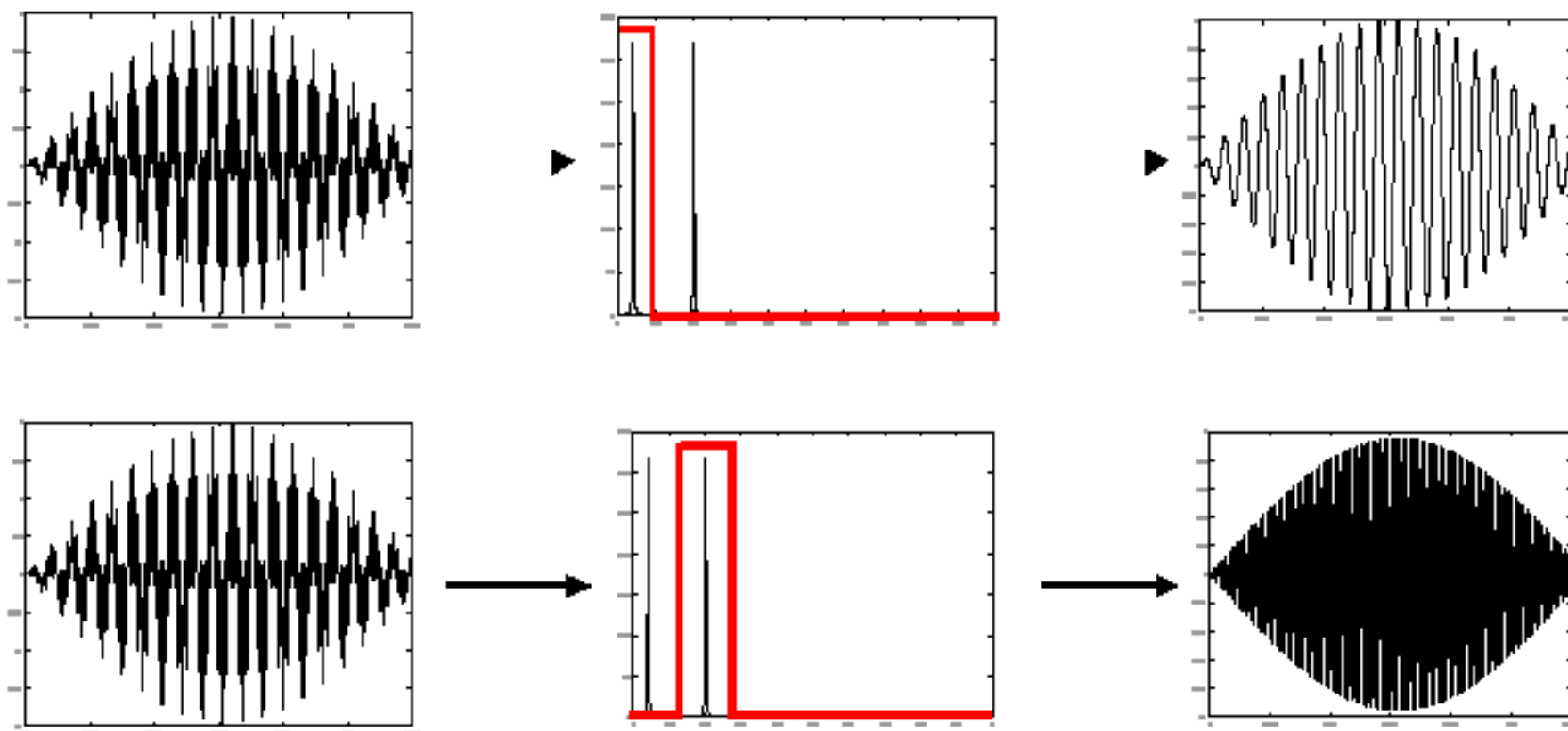
(b)

**Figure 7:** (a): The sum of a tone at 200 Hz and a tone at 800 Hz. (b): Its Fourier Transform.

# Fourier Transform (Cont...)

- Using the Fourier's theorem, *“any periodic or aperiodic waveform, no matter how complex, can be analyzed, or decomposed, into a set of simple sinusoid waves with calculated frequencies, amplitudes, phase angles”*
- Change the discussion from time domain to frequency domain
- The mathematical manipulations required for Fourier analyses are quite sophisticated. However, human brain can perform the equivalent analyses almost automatically, both blending and decomposing complex sounds.

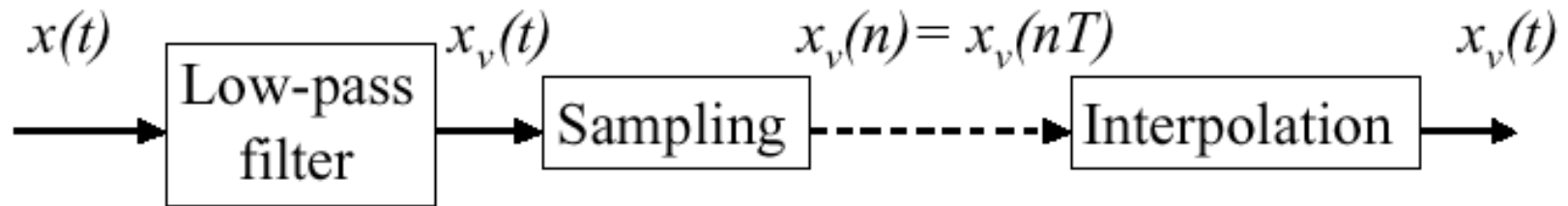
# Filters



*Figure 8: Filters with bandwidth between 0 and 400 Hz (first row) and between 400 and 1200 Hz (second row) and their action to the signal of Figure 7.*

# Sampling

- Sequence of sampling



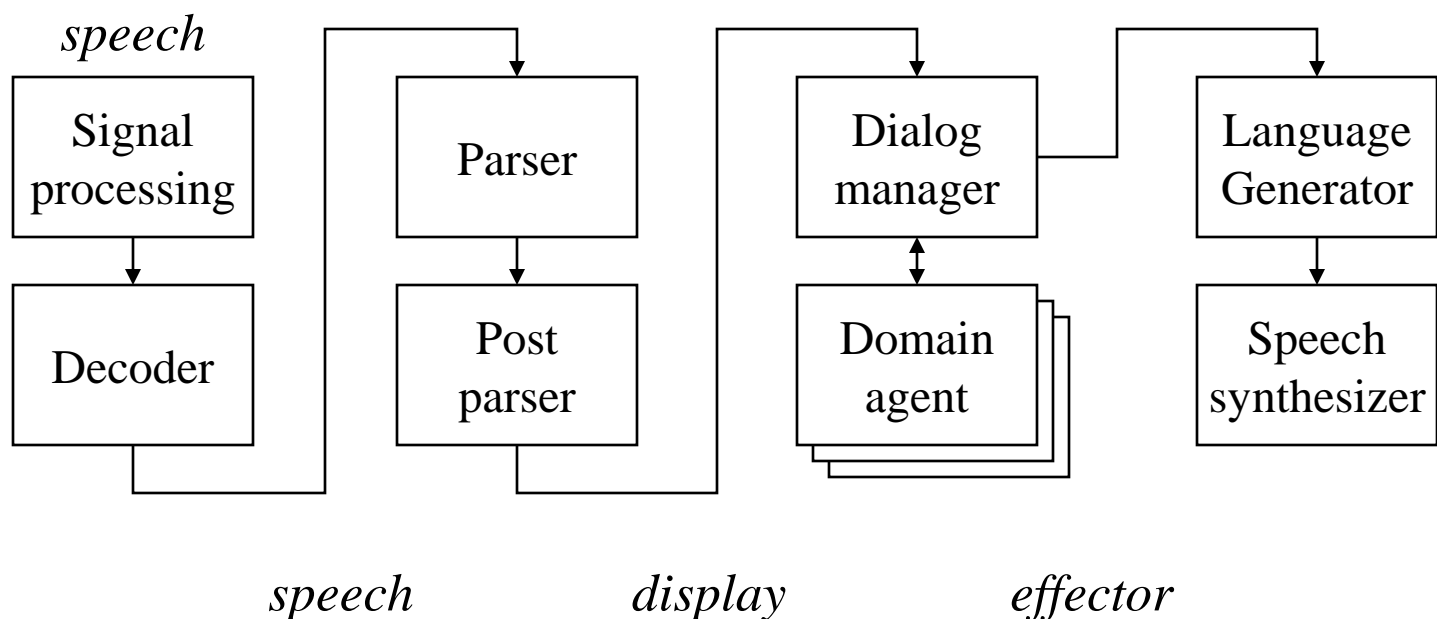
- A signal bandwidth-limited to  $B$  can be fully reconstructed from its samples, if the sampling rate is *at least twice of the highest frequency of the signal*, i.e., the sampling period is less than  $1/2B$  – **Nyquist sampling rate**
- Subsampling: a technique where the overall amount of data that will represent the digitized signal has been reduced (because this violate the sampling theorem, many types of distortion/aliasing may be noticeable)

# Sampling Rate and PCM Data Rate

Quality	Sampling Rate (KHz)	Bits per Sample	Data Rate Kbits/s Kbytes/s	Freq. Band
Telephone	8	8 (Mono)	64 8	200-3,400 Hz
AM Radio	11.025	8 (Mono)	88.2 11.0	100-5,000 Hz
FM Radio	22.050	16 (Stereo)	705.6 88.2	50-10,000 Hz
CD	44.1	16 (Stereo)	1411.2 176.4	20-20,000 Hz

# Speech Processing

- Speech enhancement
- Speech recognition
  - Transcription
    - dictation, information retrieval
  - Command and control
    - data entry, device control, navigation
  - Information access
    - airline schedules, stock quotes
- Speech synthesis





# Audio Coding and Standards

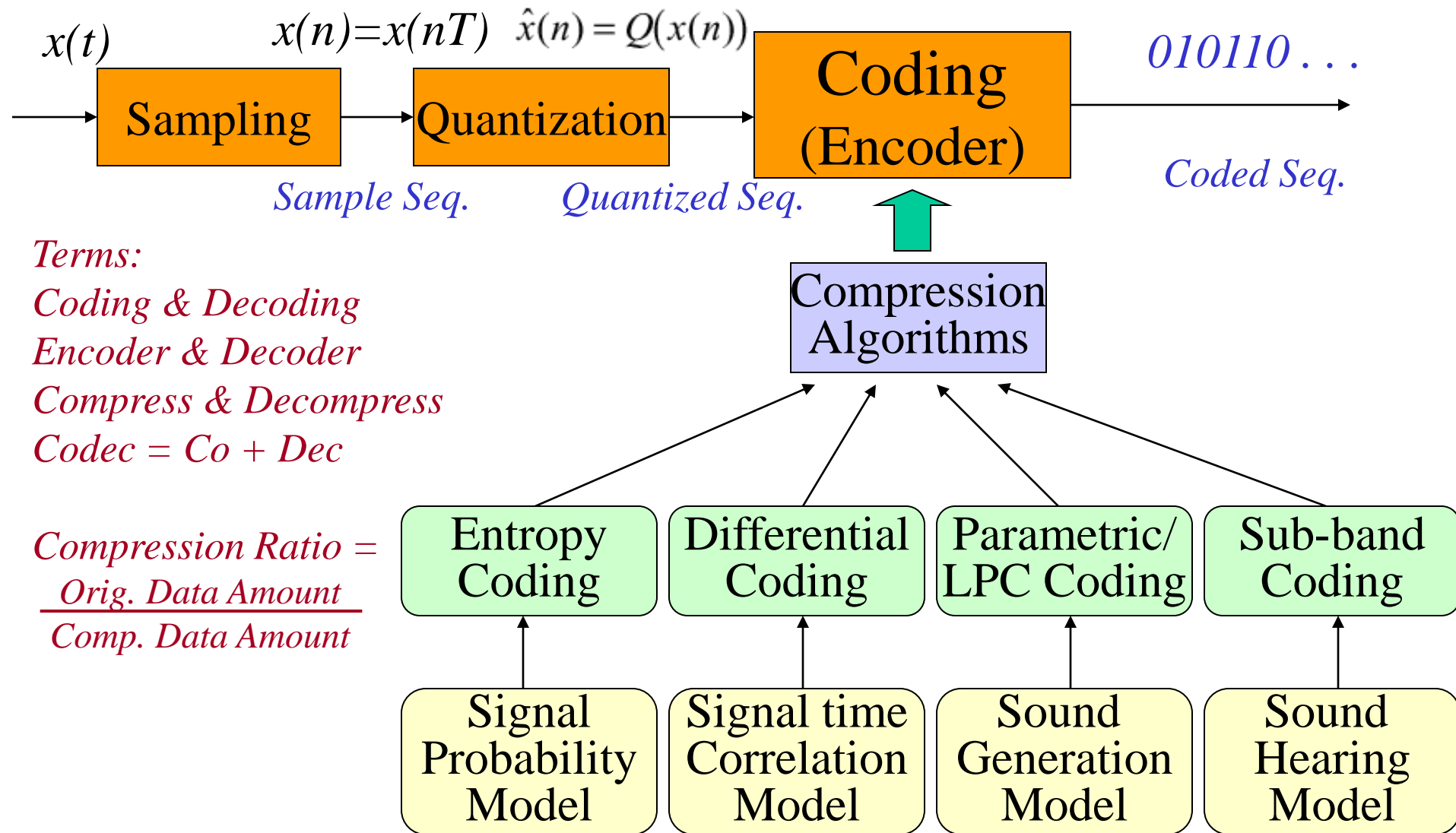
- Models, Techniques & Requirements of Sound Coding
- Entropy Coding: Run length Coding & Huffman coding
- Differential Coding – DPCM & ADPCM
- LPC and Parametric Coding
- Sound Masking Effect and Sub-band Coding
- ITU G.72x Speech/Audio Standards
- ISO MPEG-1/2/4 Audio Standards
- MIDI and Structured Audio
- Common Audio File Formats

# PCM Audio Data Rate and Data Size

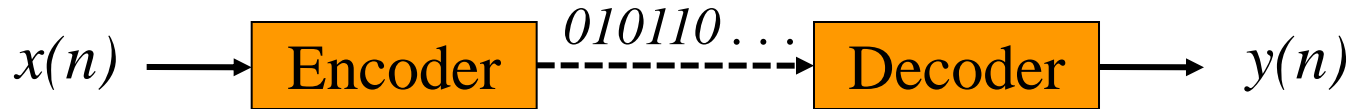
Quality	Sampling Rate (KHz)	Bits per Sample	Data Rate Kbits/s KBytes/s	Data Size in 1 minute 1 hour
Telephone	8	8 (Mono)	64Kbps 8KB/s	480KB 28.8MB
AM Radio	11.025	8 (Mono)	88.2Kbps 11.0KBps	660KB 39.6MB
FM Radio	22.050	16 (Stereo)	705.6Kbps 88.2KBps	5.3MB 317.5MB
CD	44.1	16 (Stereo)	1.41Mbps 176.4KBps	10.6MB 635MB

Conclusion → Need better coding for compressing sound data

# Models & Techniques of Sound Compression



# Requirements for Compression Algorithms



- **Lossless compression:**  $y(n) = x(n)$ 
  - Decoded audio is mathematically equivalent to the original one
  - Drawback : achieves only a small or modest level of compression
- **Lossy compression:**  $y(n) \doteq x(n)$ 
  - Decoded audio is worse than the original one  $\rightarrow$  *Distortion*
  - Advantage: achieves very high degree of compression
  - Objective: maximize the degree of compression in certain quality
- **General compression requirements:**
  - Ensure a good quality of decoded/uncompressed audio
  - Achieve high compression ratios
  - Minimize the complexity of the encoding and decoding process
  - Support multiple channels
  - Support various data rates
  - Give small delay in processing

# Entropy Coding

- **Entropy encoding (lossless):** Ignores semantics of input data and compresses media streams  $x(n)$  by regarding them as sequences of digits or symbols
  - Examples: run-length encoding, Huffman encoding , ...
- **Run-length encoding:**
  - A compression technique that replaces consecutive occurrences of a symbol with the symbol followed by the number of times it is repeated
    - a a a a a  $\Rightarrow$  ax5
    - 0000000000000000000000000000111111  $\Rightarrow$  0x20 1x7
  - Most useful where symbols appear in long runs: e.g., for images that have areas where the pixels all have the same value, fax and cartoons for examples.

# Huffman Coding

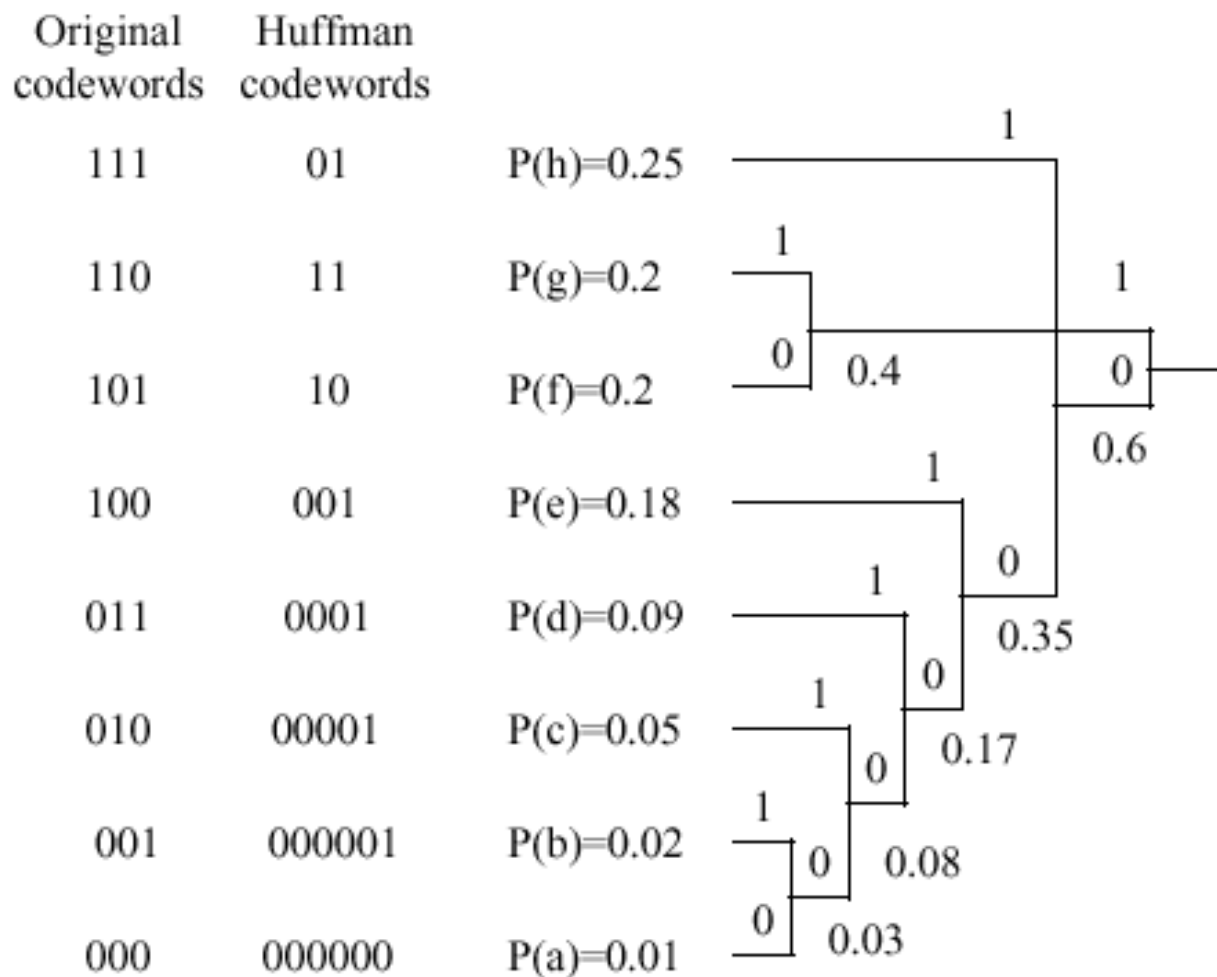
## Huffman encoding:

- A popular compression technique that  
→ assigns variable length binary codes to symbols, so that the most frequently occurring symbols have the shortest codes
- Huffman coding is particularly effective where the data are dominated by a small number of symbols, e.g.  
 $\{x(n)\} = hfeeeeegheeeegdeeehehcfbeeeeeeqghf...$
- Suppose to encode a source of  $N=8$  symbols:  $X(n) \rightarrow \{a,b,c,d,e,f,g,h\}$
- The probabilities of these symbols are:  $P(a) = 0.01$ ,  $P(b)=0.02$ ,  $P(c)=0.05$ ,  $P(d)=0.09$ ,  $P(e)=0.18$ ,  $P(f)=0.2$ ,  $P(g)=0.2$ ,  $P(h)=0.25$
- If assigning 3 bits per symbol (000~111), the average length of symbols is:  
$$\bar{L} = \sum_{i=1}^8 3P(i) = 3 \text{ bits/symbol}$$
- The theoretical lowest average length – **Entropy**  
$$H(P) = - \sum_{i=1}^N P(i) \log_2 P(i) = 2.57 \text{ bits /symbol}$$
- If we use Huffman encoding, the average length = 2.63 bits/symbol

# Huffman Coding (Cont...)

- The Huffman code assignment procedure is based on a *binary tree* structure. This tree is developed *by a sequence of pairing operations* in which the *two least probable symbols* are joined at a node to form two branches of a tree. More precisely:
  - 1. The list of probabilities of the source symbols are associated with the leaves of a binary tree.
  - 2. Take the two smallest probabilities in the list and generate an intermediate node as their parent and label the branch from parent to one of the child nodes 1 and the branch from parent to the other child 0.
  - 3. Replace the probabilities and associated nodes in the list by the single new intermediate node with the sum of the two probabilities. If the list contains only one element, quit. Otherwise, go to step 2.

# Huffman Coding (Cont...)

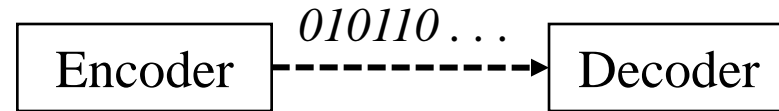




# Huffman Coding (Cont...)

Huffman Table	
h:01	d:0001
g:11	c:00001
f: 10	b:000001
e: 001	a:0000001

- The new average length of the source



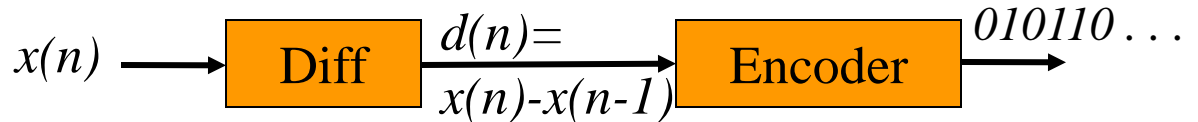
$$\bar{L}_{\text{Huf}} = 0.25 \times 2 + 0.2 \times 2 + 0.2 \times 2 + 0.18 \times 3 + 0.09 \times 4 + 0.05 \times 5 + 0.02 \times 6 + 0.01 \times 6 = 2.63 \text{ bits/symbol}$$

- The efficiency of this code is  $H / \bar{L}_{\text{Huf}} = 98\%$ .
- How do we estimate the  $P(i)$  ? Relative frequency of the symbols
- How to decode the bit stream ? Share the same Huffman table
- How to decode the variable length codes ? Prefix codes have the property that no codeword can be the prefix (i.e., an initial segment) of any other codeword. Huffman codes are prefix codes !
  - 00000100100110 => ? **beef**
- Does the best possible codes guarantee to always reduce the size of sources? No. Worst case exists. Huffman coding is better averagely.
- Huffman coding is particularly effective where the data are dominated by a small number of symbols

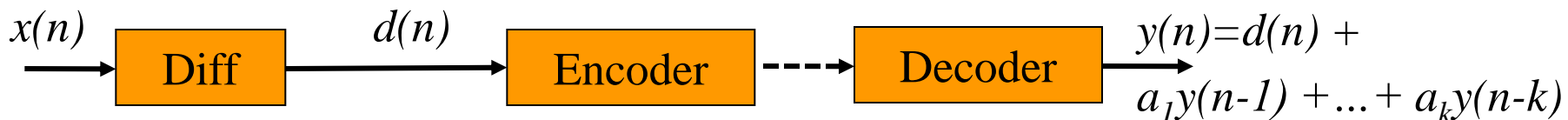
# Differential Coding – DPCM & ADPCM

- Based on the fact that neighboring samples  $\dots x(n-1), x(n), x(n+1), \dots$  in a discrete audio sequence changing slowly in many cases

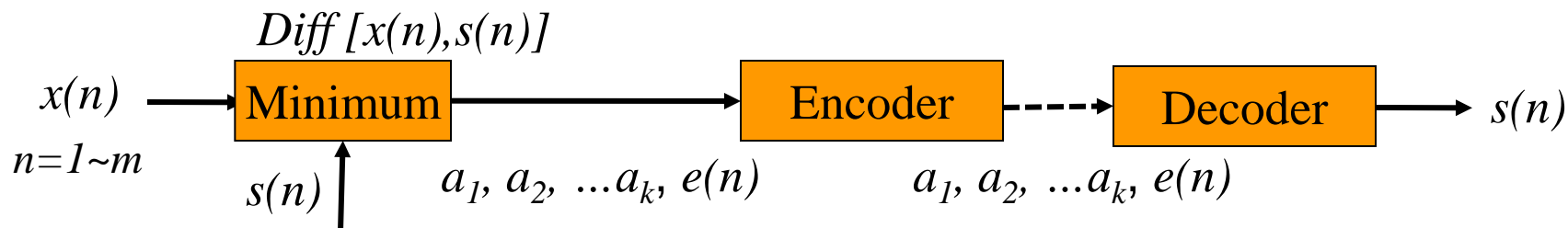
- A differential PCM coder (DPCM) quantizes and encodes the difference  $d(n) = x(n) - x(n-1)$



- Advantage of using difference  $d(n)$  instead of the actual value  $x(n)$ 
  - Reduce the number of bits to represent a sample
- General DPCM:  $d(n) = x(n) - a_1x(n-1) - a_2x(n-2) - \dots - a_kx(n-k)$   
 $a_1, a_2, \dots, a_k$  are fixed
- Adaptive DPCM:  $a_1, a_2, \dots, a_k$  are dynamically changed with signal



# LPC and Parametric Coding



## LPC (Linear Predictive Coding)

- Based on the human utterance organ model

$$s(n) = a_1 s(n-1) + a_2 s(n-2) + \dots + a_k s(n-k) + e(n)$$

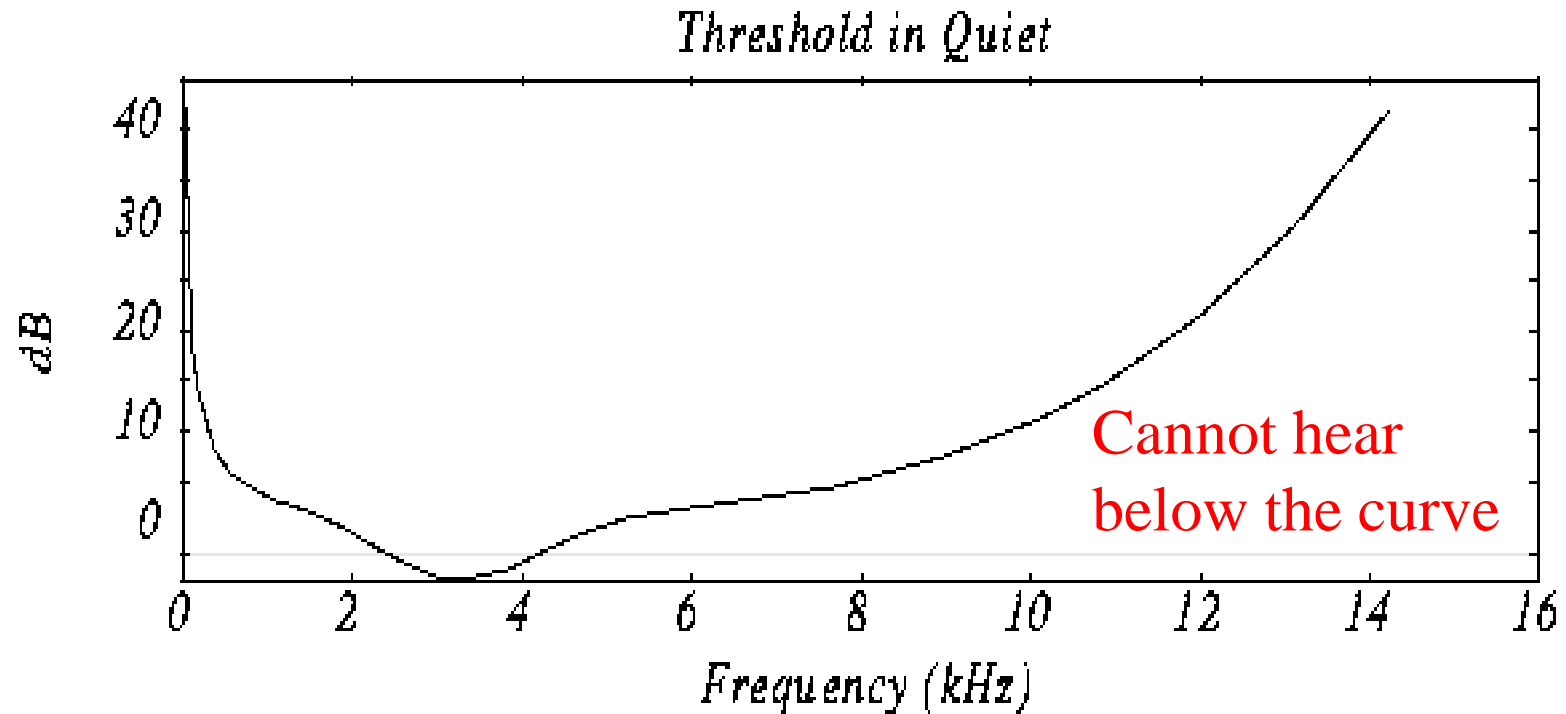
- Estimate  $a_1, a_2, \dots, a_k$  and  $e(n)$  for each piece (frame) of speech
- Encode and transmit/store  $a_1, a_2, \dots, a_k$  and type of  $e(n)$
- Decoder reproduce speech using  $a_1, a_2, \dots, a_k$  and  $e(n)$ 
  - very low bit rate but relatively low speech quality

## Parametric coding:

- Only coding parameters of sound generation model
- LPC is an example where parameters are  $a_1, a_2, \dots, a_k, e(n)$
- Music instrument parameters: pitch, loudness, timbre, ...

# Sub-band Coding

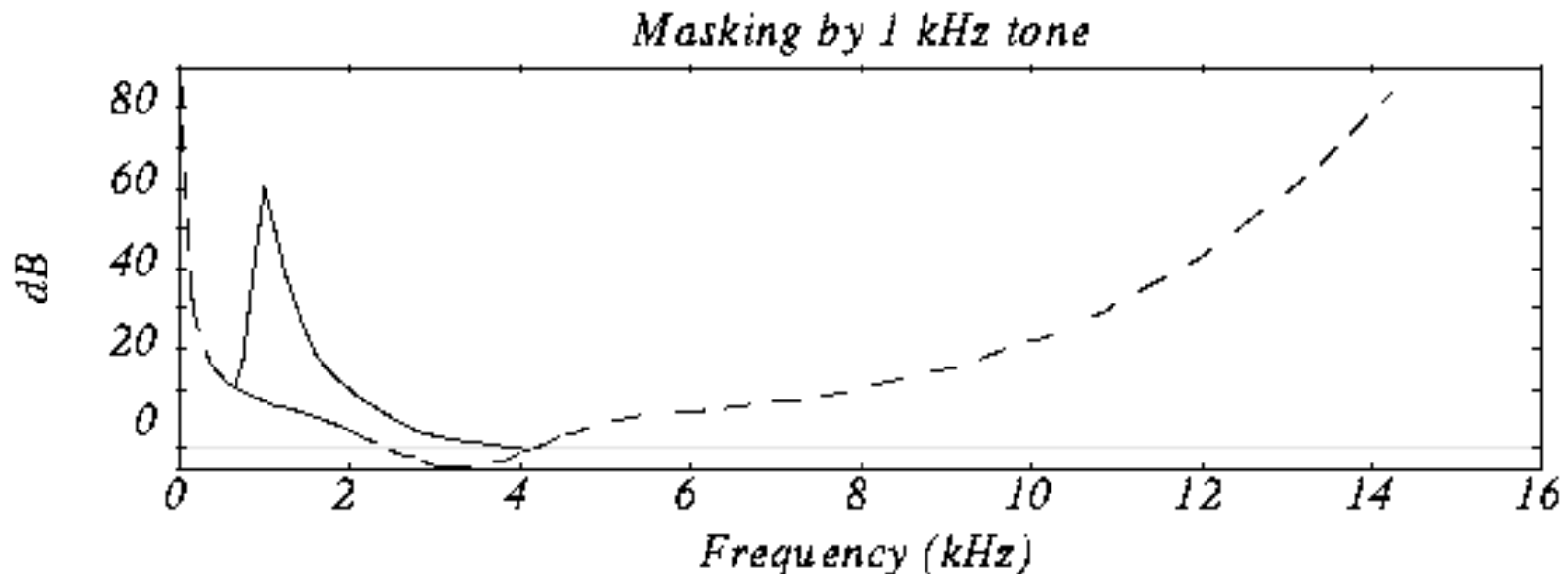
- Human auditory system has limitations
  - Frequency range: 20 Hz to 20 kHz, sensitive at 2 to 4 KHz.
  - Dynamic range (quietest to loudest) is about 96 dB



- Moreover, based on psycho-acoustic characteristics of human hearing, algorithms perform some tricks to further reduce data rate

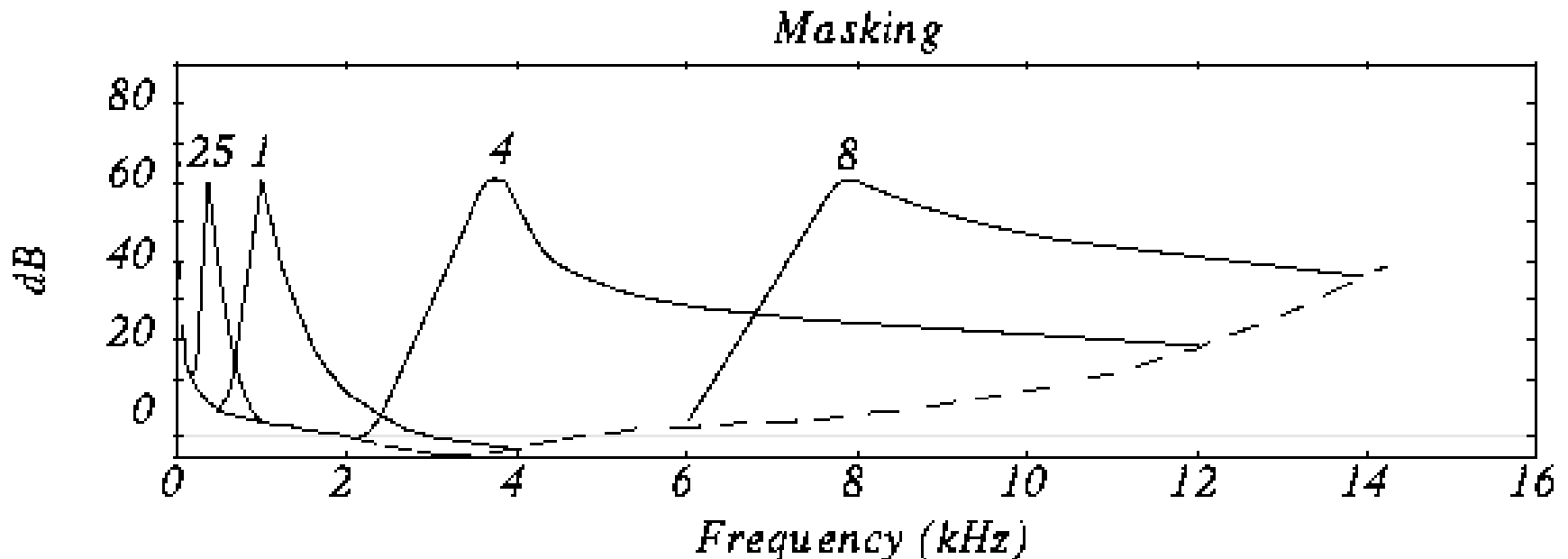
# Masking Effects

- **Frequency Masking:** If a tone of a certain frequency and amplitude is present, then other tones or noise of similar frequency cannot be heard by the human ear
- the louder tone (masker) makes the softer tone (maskee)  
=> no need to encode and transfer the softer tone



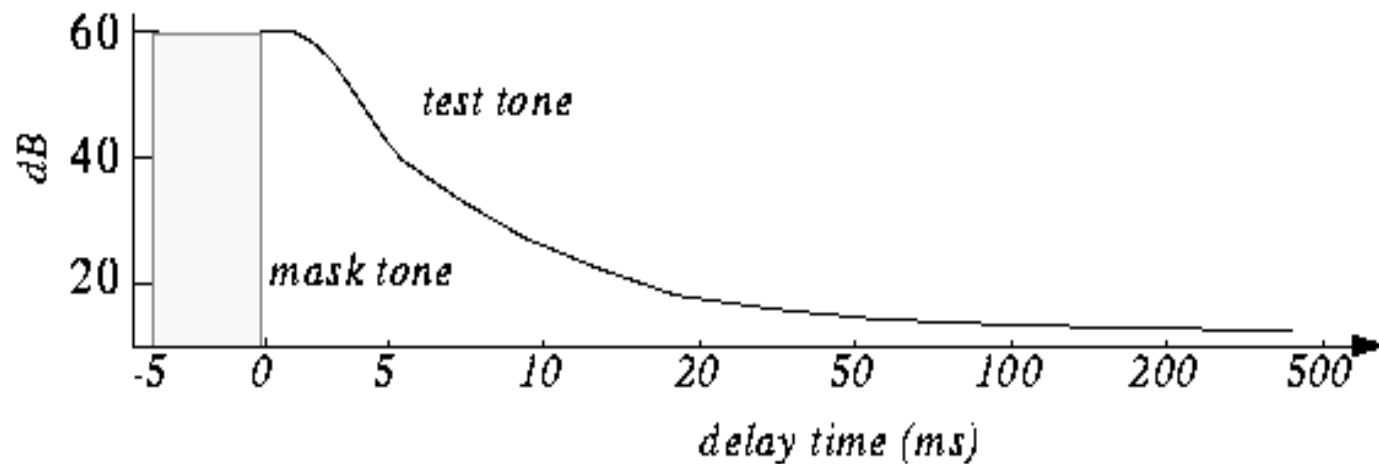
# Masking Effects (Cont...)

- Repeat for various frequencies of masking tones
- **Masking Threshold:** Given a certain masker, the maximum non-perceptible amplitude level of the softer tone



# Masking Effects (Cont...)

- **Temporal Masking:** If we hear a loud sound, then it stops, it takes a little while until we can hear a soft tone nearby.



- The Masking Threshold is used by the audio encoder to determine the maximum allowable quantization noise at each frequency to minimize noise perceptibility: remove parts of signal that we cannot perceive

# Speech Compression

- Handling speech with other media information such as text, images, video, and data is the essential part of multimedia applications
- The ideal speech coder has a low *bit-rate*, high perceived *quality*, low signal *delay*, and low *complexity*.
- **Delay**
  - Less than 150 ms one-way end-to-end delay for a conversation
  - Processing (coding) delay, network delay
  - Over Internet, ISDN, PSTN, ATM, ...
- **Complexity**
  - Computational complexity of speech coders depends on algorithms
  - Contributes to achievable bit-rate and processing delay



# G.72x Speech Coding Standards

- **Quality**
  - “intelligible” → “natural” or “subjective” quality
  - Depending on bit-rate
- **Bit-rate**

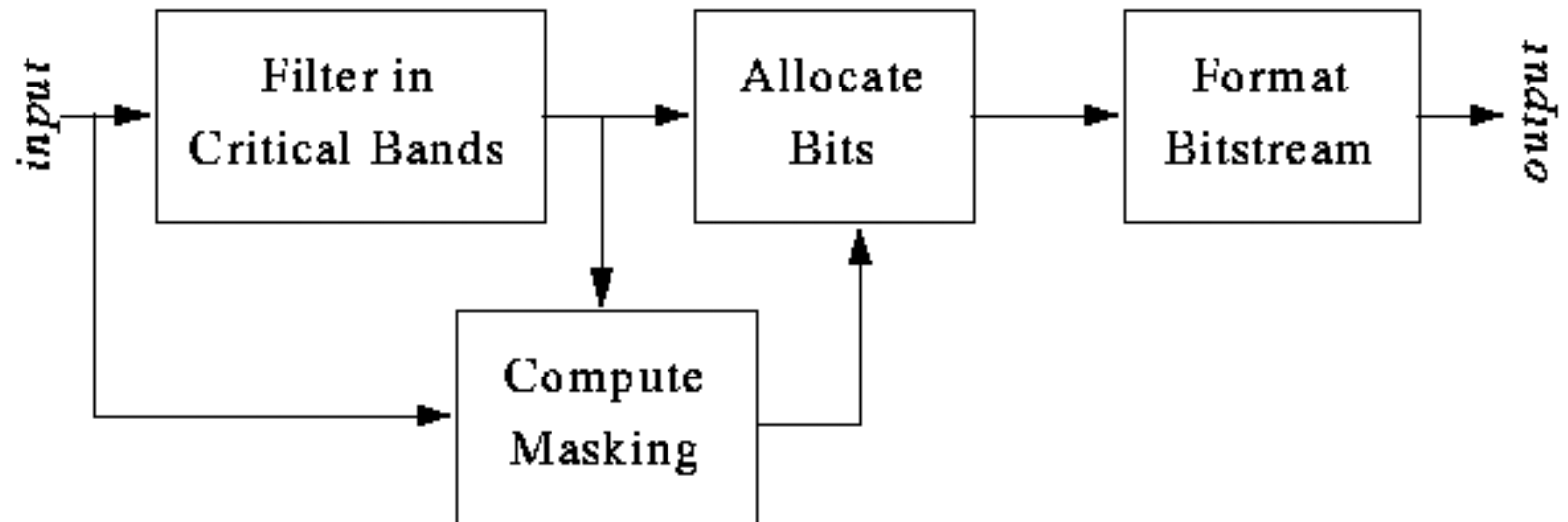
Standard	Bit rate	Frame size/ Look-ahead	Complexity
G.711 PCM	64 Kb/s	0 / 0 ms	0 MIPS
G.726, G.727	16,24,32,40 Kb/s	0.125 / 0 ms	2 MIPS
G.722	48,56,64 Kb/s	0.125 / 1.5 ms	5 MIPS
G.728	16 Kb/s	0.625 / 0 ms	30 MIPS
G.729	8 Kb/s	10 / 5 ms	20 MIPS
G.723	5.3 & 6.4 Kb/s	30/7.5 ms	16 MIPS

# G.72x Audio Coding Standards

- **Silence Compression** - detect the "silence", similar to run-length coding
- **Adaptive Differential Pulse Code Modulation (ADPCM)**  
e.g., in CCITT G.721 -- 16 or 32 Kb/s.
  - (a) Encodes the difference between two or more consecutive signals; the difference is then quantized  
→ hence the loss (*speech quality becomes worse*)
  - (b) Adapts at quantization so fewer bits are used when the value is smaller.
  - It is necessary to predict where the waveform is headed → difficult
- **Linear Predictive Coding (LPC)** fits signal to speech model and then transmits parameters of model  
→ sounds like a computer talking, 2.4 Kb/s.

# MPEG-1/2 Audio Compression

1. Use filters to divide the audio signal (e.g., 20-20kHz sound) into 32 frequency subbands --> *subband filtering*.
2. Determine amount of masking for each band caused by nearby band using the *psycho-acoustic model*.
3. If the power in a band is below masking threshold, don't encode it.
4. Otherwise, determine no. of bits needed to represent the coefficient such that noise introduced by quantization is below the masking effect (one fewer bit of quantization introduces about 6 dB of noise).
5. Format bitstream



# MPEG Audio Compression Example

- After analysis, the first levels of 16 of the 32 bands are these:

---

Band	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Level(db)	0	8	12	10	6	2	10	60	35	20	15	2	3	5	3	1

---

- If the level of the 8th band is 60dB, it gives a masking of 12 dB in the 7th band, 15dB in the 9th.
- Level in 7th band is 10 dB (  $< 12$  dB ), so ignore it.
- Level in 9th band is 35 dB (  $> 15$  dB ), so send it.  
[ Only the amount above the masking level needs to be sent, so instead of using 6 bits to encode it, we can use 4 bits -- saving 2 bits (= 12 dB). ]

# MPEG Audio Layers

- MPEG defines 3 layers for audio. Basic model is same, but codec complexity increases with each layer.
- Divides data into frames, each of them contains 384 samples, 12 samples from each of the 32 filtered subbands.
- Layer 1: DCT type filter with one frame and equal frequency spread per band. Psycho-acoustic model only uses frequency masking.
- Layer 2: Use three frames in filter (before, current, next, a total of 1152 samples). This models a little bit of the temporal masking.
- Layer 3: Better critical band filter is used (non-equal frequencies), psycho-acoustic model includes temporal masking effects, takes into account stereo redundancy, and uses Huffman coder
- MP3: Music compression format using MPEG Layer 3

# MPEG Audio Layers (Cont...)

<b>Layer</b>	<b>Target Bit-rate per channel</b>	<b>Ratio</b>	<b>Quality at 64 kb/s</b>	<b>Quality at 128 kb/s</b>	<b>Theoretical Min. Delay</b>
<b>Layer 1</b>	192 kb/s	4:1	---	---	19 ms
Layer 2	128 kb/s	6:1	2.1 to 2.6	4+	35 ms
Layer 3	64 kb/s	12:1	3.6 to 3.8	4+	59 ms

- Quality factor: 5 - perfect, 4 - just noticeable, 3 - slightly annoying  
2 - annoying, 1 - very annoying
- Real delay is about 3 times of the theoretical delay

# MPEG-1 Audio Facts

- MPEG-1: 64K~320Kbps for audio
  - Uncompressed CD audio => 1.4 Mb/s
- Compression factor ranging from 2.7 to 24.
- With Compression rate 6:1 (16 bits stereo sampled at 48 KHz is reduced to 256 kb/s) and optimal listening conditions, expert listeners could not distinguish between coded and original audio clips.
- MPEG audio supports sampling frequencies of 32, 44.1 and 48 KHz.
- Supports one or two audio channels in one of the four modes:
  1. Monophonic -- single audio channel
  2. Dual-monophonic -- two independent chs, e.g., English and French
  3. Stereo -- for stereo channels that share bits, but not using Joint-stereo coding
  4. Joint-stereo -- takes advantage of the correlations between stereo channels

# MPEG-2 Audio Coding

- MPEG-2/MC: Provide theater-style surround sound capabilities
  - Five channels: left, right, center, rear left, and rear right
  - Five different modes: mono, stereo, three ch, four ch, five ch
  - Full five channel surround stereo: 640 Kb/s
  - 320 Kb/s for 5.1 stereo (5 channels+sub-woofer ch)
- MPEG-2/LSF (Low sampling frequency: 16k, 22K, 24k)
- MPEG-2/AAC (Advanced Audio Coding)
  - 7.1 channels
  - More complex coding
- Compatibility
  - Forward: MPEG-2 decoder can decode MPEG-1 bitstream
  - Backward: MPEG-1 decoder can decode a part of MPEG-2



# MPEG-4 Audio Coding

- Consists of natural coding and synthetic coding
- Natural coding
  - General coding: AAC and TwinVQ based arbitrary audio  
twice as good as MP3
  - Speech coding:
    - \* CELP I: 16K samp., 14.4~22.5Kbps
    - \* CELP II: 8K & 16K samp., 3.85~23.8Kbps
    - \* HVXV: 8M samp., 1.4~4Kbps
- Synthetic coding: structured audio
  - Interface to Text-to-Speech synthesizers
  - High-quality audio synthesis with Structured Audio
- AudioBIFS: Mix and postproduce multi-track sound streams

# Structured Audio

- A description format that is made up of semantic information about the sounds it represents, and that makes use of high-level (algorithmic) models.
  - E.g., MIDI (Musical Instrument Digital Interface).
- Normal music digitization: perform *waveform coding* (we sample the music signal and then try to reconstruct it exactly)
- MIDI: only record *musical actions* such as the key depressed, the time when the key is depressed, the duration for which the key remains depressed, and how hard the key is struck (pressure).
- MIDI is an example of **parameter** or **event-list representation**
  - An event list is a sequence of control parameters that, taken alone
  - Do not define the quality of a sound but instead specify the ordering and characteristics of parts of a sound with regards to some external model.

# Structured Audio Synthesis

- **Sampling synthesis**

- Individual instrument sounds are digitally recorded and stored in memory
- When the instrument is played, the note recording are reproduced and mixed (added together) to produce the output sound.
- This can be a very effective and realistic but requires a lot of memory
- Good for playing music but not realistic for speech synthesis
- Good for creating special sound effects from sample libraries

# Structured Audio Synthesis (Cont...)

- Additive and subtractive synthesis
  - synthesize sound from the superposition of sinusoidal components (additive)
  - Or from the filtering of an harmonically rich source sound - typically a periodic oscillator with various form of waves (subtractive).
  - Very compact representation of the sound
  - the resulting notes often have a distinctive “analog synthesizer” character.

# Applications of Structured Audio

- Low-bandwidth transmission
  - transmit a structural description and dynamically render it into sound on the client side rather than rendering in a studio on the server side
- Sound generation from process models
  - the sound is not created from an event list but rather is dynamically generated in response to evolving, non-sound-oriented environments such as video games
- Music applications
- Content-based retrieval
- Virtual reality together with VRML/X3D

# Common Audio File Formats

- Mulaw (Sun, NeXT) .au
- RIFF Wave (MS WAV) .wav
- MPEG Audio Layer (MPEG) .mp2 .mp3
- AIFC (Apple, SGI) .aiff .aif
- HCOM (Mac) .hcom
- SND (Sun, NeXT) .snd
- VOC (Soundblaster card proprietary standard) .voc
- AND MANY OTHERS!

# Demos of Audio Coding and Formats

# Image and Video Fundamentals

- Light and Color Models
  - RGB, HSB
  - Luminance and Chrominance  
YIQ, YUV, YCrCb
- Image Data Formats
- Video Camera and Display
- Scanning Video and Interlaced Scanning
- Analogy NTSC and PAL Video
- Digital Video
- Luma Sampling and Chroma Sub-Sampling
- Video Coding Standards Organizations



# History

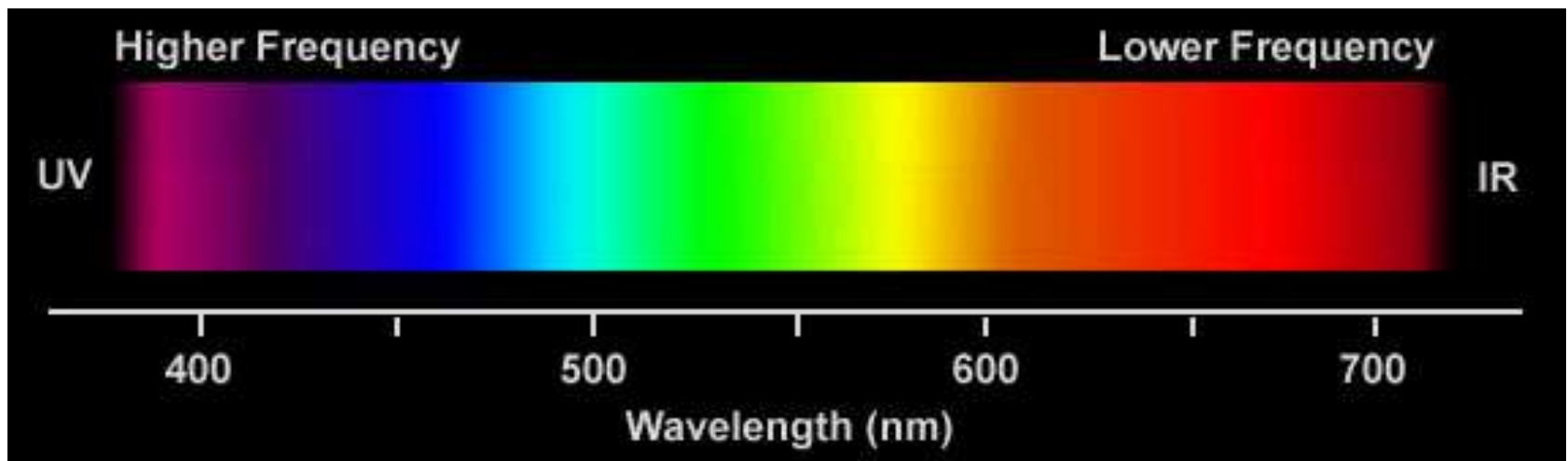
- 1839: Daguerreotype Cameras
- 1893: Telephone Audio Broadcasting (Puskas)
- 1895: Wireless Communication (Marconi, Popov)
- 1895: Film Presentation (Lumiere Brothers)
- 1919: Radio Broadcasting (Holland, Canada)
- 1934: US establishes FCC
- 1935: TV Broadcasting (Germany, Britain)
- 1941: US B&W TV

## History (Cont...)

- 1951: Videotape Recorder (Bing Crosby Enterprises)
- 1953: US Color TV (NTSC)
- 1963: Geostationary Satellites
- 1985: FCC establishes ATSC - standard by 1993?
- 1989: Analog HDTV Broadcasting (Japan)
- 1993: VCD (Video on CD Based on MPEG-1)
- 1994: Digital Video Broadcast & CD Based on MPEG-2
- 1996: ATSC Standard Adopted
- 1999: Internet/Web Video Broadcasting (MPEG-4)
- 2001: Wireless Internet Video Communications
- 2003: Digital TV Broadcast (Japan)

# Light

- Light exhibits some properties that make it appear to consist of particles; at other times, it behaves like a wave.
- Light is electromagnetic energy that radiates from a source of energy (or a source of light) in the form of waves
- Visible light is in the 400 nm – 700 nm range of electromagnetic spectrum



# Intensity of Light

- The strength of the radiation from a light source is measured using the unit called the candela, or candle power. The total energy from the light source, including heat and all electromagnetic radiation, is called radiance and is usually expressed in watts.
- Luminance is a measure of the light strength that is actually perceived by the human eye. Radiance is a measure of the total output of the source; luminance measures just the portion that is perceived.
- Brightness is a subjective, psychological measure of perceived intensity. Brightness is practically impossible to measure objectively. It is relative. For example, a burning candle in a darkened room will appear bright to the viewer; it will not appear bright in full sunshine.
- The strength of light diminishes in inverse square proportion to its distance from its source. This effect accounts for the need for high intensity projectors for showing multimedia productions on a screen to an audience.

# Basics of Color

- **Color** is the sensation registered when light of different wavelengths is perceived by the brain.
- Observed in objects that reflect or emit certain wavelengths of light.
- Can create the sensation of any color by mixing appropriate amounts of the three primary colors — *red*, *green*, and *blue*.
- Can create colors on computer monitors using the emission of three wavelengths of light in appropriate combinations.
- **Hue** distinguishes among colors such as red, green, and yellow.
- **Saturation** refers to how far color is from a gray of equal intensity.
- **Lightness** embodies the achromatic notion of perceived intensity of a reflecting object.
- **Brightness** is used instead of lightness for a self-luminous object such as CRT.

# Hue, Saturation and Brightness/Luminance



H  
*dominant  
wavelength*



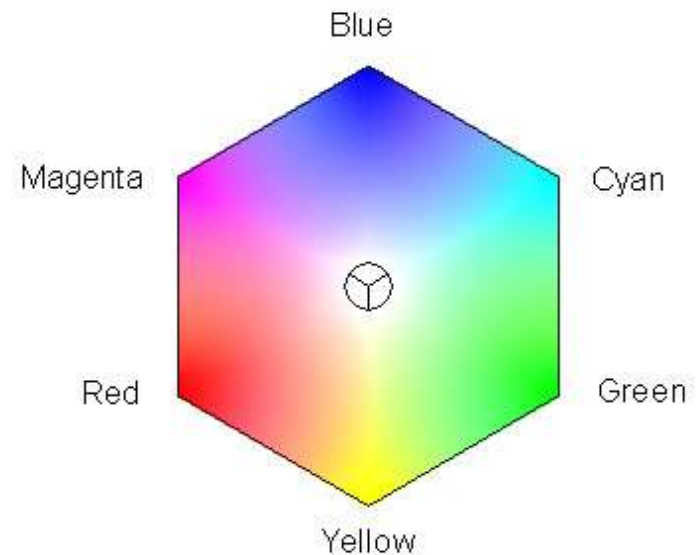
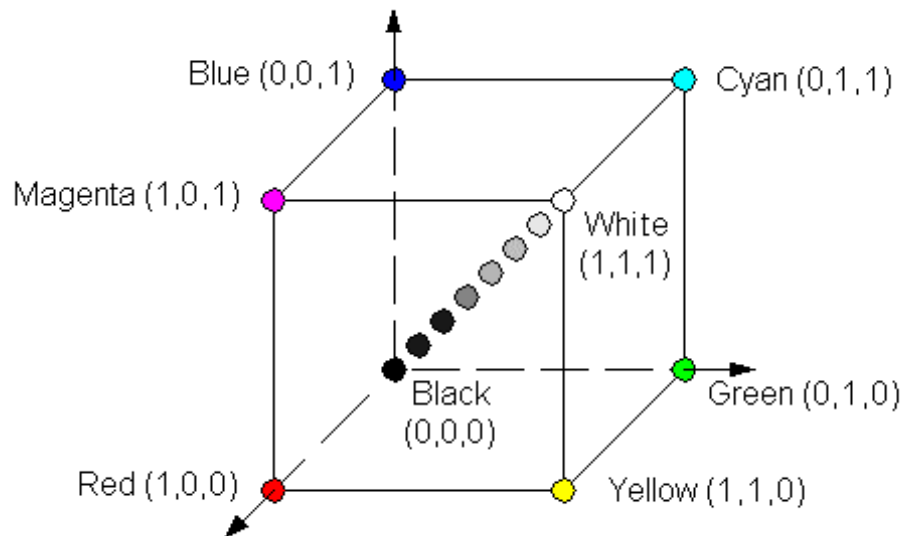
S  
*purity  
% white*



B/L  
*luminance*

# Color Models in Images

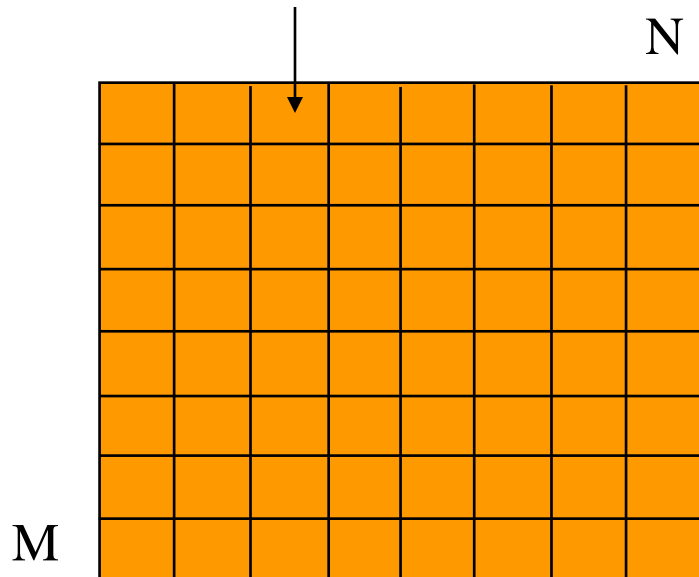
- **RGB color model:** each displayed color is described by three independent parameters- the luminance of each of the three primary colors (0 – 1) - primary used in color CRT monitors
- Employs a Cartesian coordinate system. The RGB primaries are *additive*; which means that individual contributions of each primary are added for the creation of a new color.



# Graphic/Image Data Structure

- Pixels: picture elements in digital images
- Image resolution ( $M*N$ ): number of pixels in a digital image

Pixel:  $p(x,y)=(r_{xy}, g_{xy}, b_{xy})$



**Pixel Array/Matrix**

$$\begin{bmatrix} p(1,1) & p(1,2) & \dots & p(1,N) \\ p(2,1) & p(2,2) & \dots & p(2,N) \\ & & \dots & \\ & & \dots & \\ p(M,1) & p(M,2) & \dots & p(M,N) \end{bmatrix}$$



# Monochrome & Gray-scale Images

- Monochrome image
  - Each pixel is stored as a single bit ( $p(x,y)=0$  or  $1$ )
  - A 640 X 480 monochrome image requires 37.5 Kbytes
- Gray-scale image ( $p(x,y)=0\sim 1$ )
  - Each pixel is usually stored as a byte (0 to 255 levels)
  - A 640 X 480 gray-scale image requires over 300 KBytes



# Pseudo & True-Color Images

- 8-bit (pseudo) color image
  - One byte for each pixel
  - Support 256 colors
  - A 640 X 480 8-bit color image requires 307.2 KBytes
- 24-bit (true) color image
  - Three bytes for each pixel
  - Support 256X256X256 colors
  - A 640 X 480 24-bit color image requires 921.6 KBytes



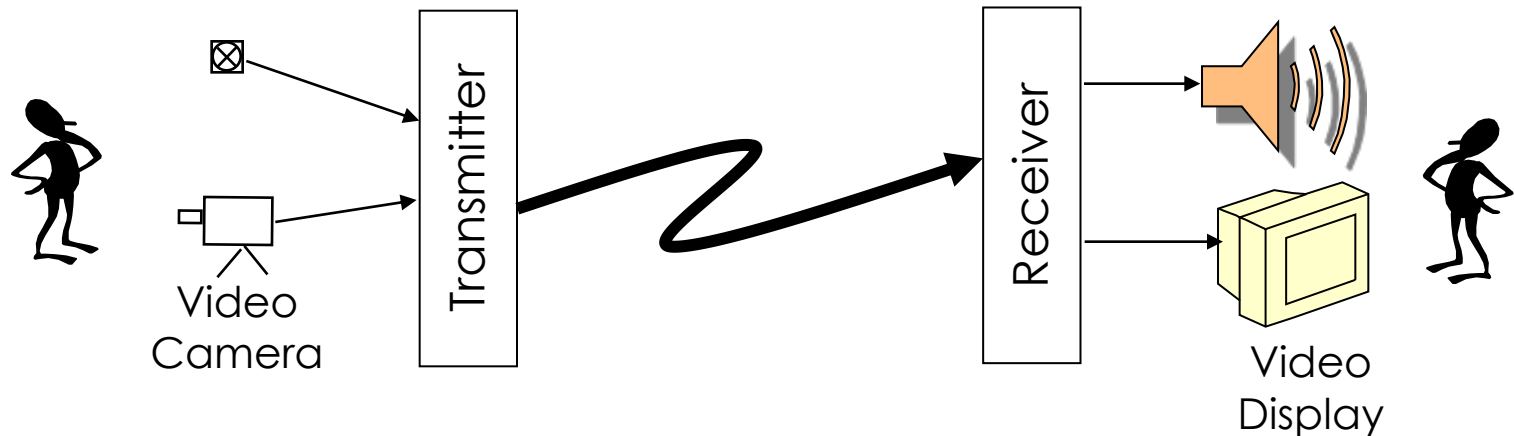
# Image Data Formats

- Standard system independent formats
  - **GIF**: Graphics Interchange Format by the UNISYS and CompuServe
    - initially designed for transmitting images over phone lines
    - Limited to 8-bit color images
  - **JPEG**: a standard for photographic (still) image compression by the Joint Photographic Experts Group
    - Take advantage of limitations of human vision system to achieve high rates of compression
    - Lossy compression which allows user to set the desired level of quality
  - **TIFF**: Tagged Image File Format by the Aldus Corp.
    - Lossless format to store many different types of images
    - No major advantages over JPEG and not user-controllable

# Image Data Formats (Cont...)

- **PS/EPS**: a typesetting language
  - including vector/structured graphics and bit-mapped images
  - Used in several popular graphics programs (Adobe)
  - no compression, files are large
- System dependent formats
  - **BMP**: support 24-bit bitmap images for Microsoft Windows
  - **XBM**: support 24-bit bitmap images for X Windows systems
  - Many, many others

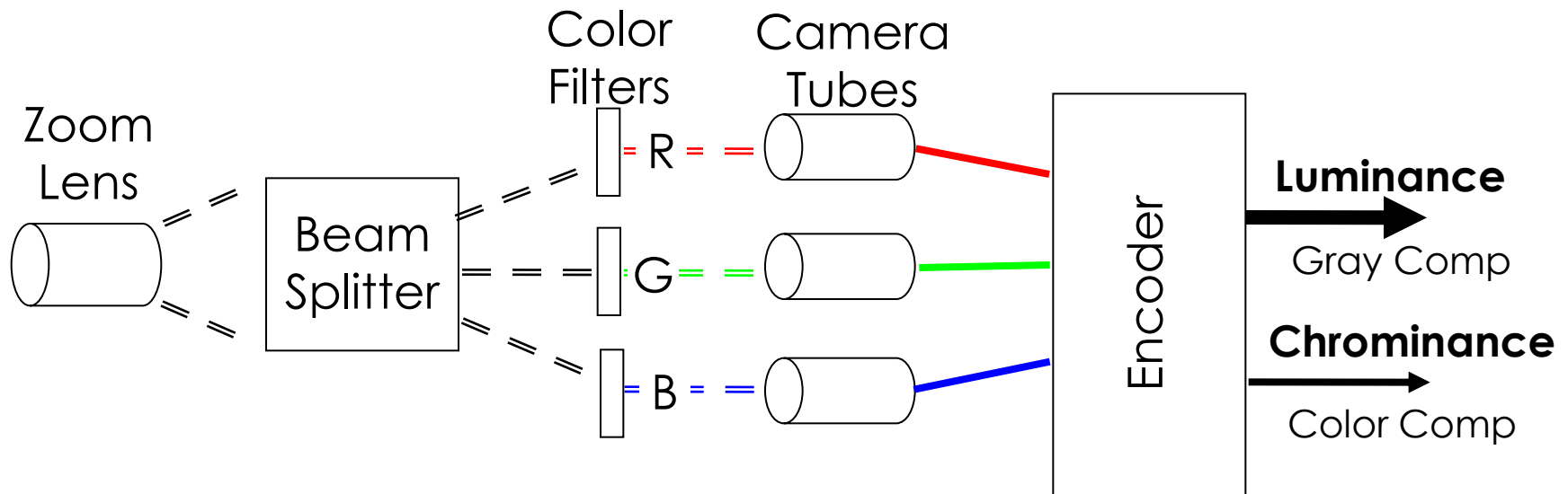
# Video Communication/Broadcast System



## Goals:

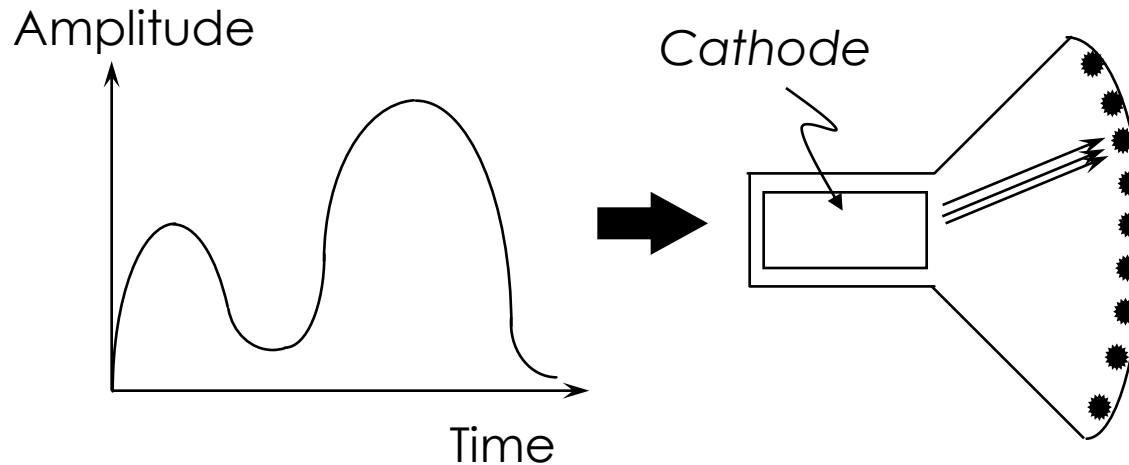
1. Efficient use of bandwidth
2. High viewer perception of quality

# Camera Operation



- Camera has 1, 2, or 3 tubes for sampling
- More tubes (CCD's) and better lens produce better pictures
- Video composed of luminance and chrominance signals
- Composite video combines luminance and chrominance
- Component video sends signals separately

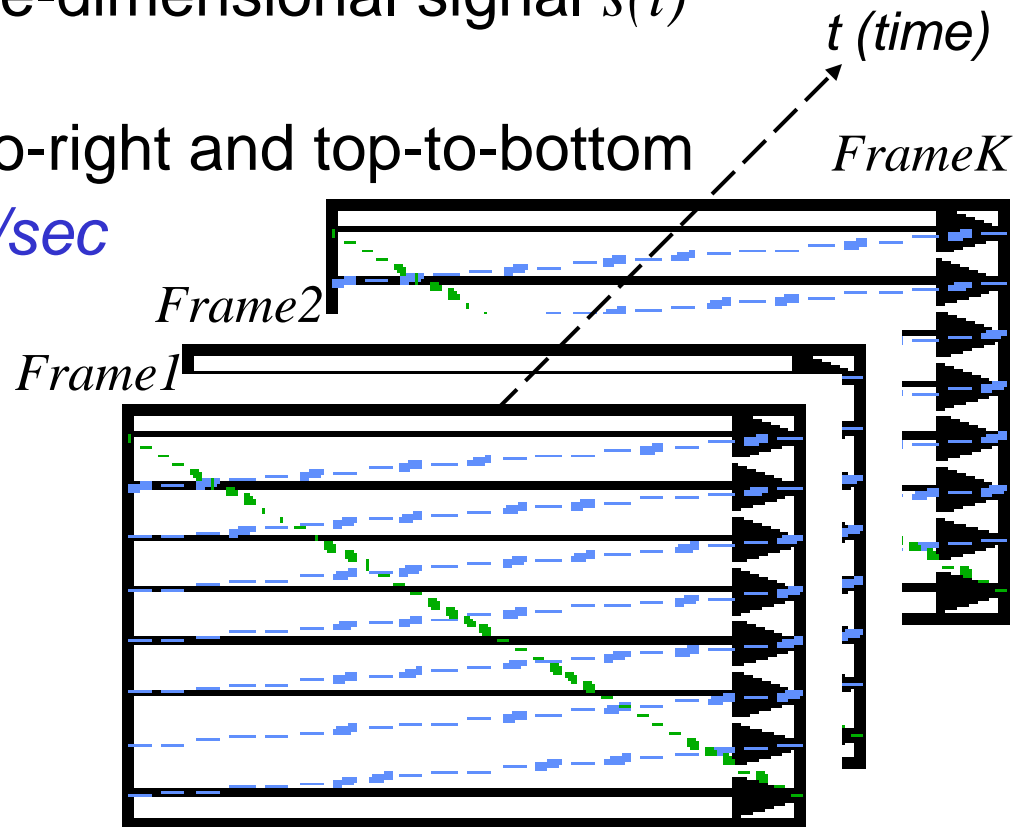
# Video Display Scanning



- Three guns (RGB) energize phosphors
  - Varying energy changes perceived intensity
  - Different energies to different phosphors produces different colors
  - Phosphors decay so you have to refresh
- Different technologies
  - Shadow mask (delta-gun dot mask)
  - PIL slot mask
  - Single-gun (3 beams) aperture-grille (Trinitron)

# Scanning Video

- Video is obtained via *raster scanning*, which transforms a 3-D signal  $p(x, y, t)$  into a one-dimensional signal  $s(t)$  which can be transmitted.
- Progressive scanning: left-to-right and top-to-bottom
  - Samples in time: *frames/sec*
  - Samples along  $y$ : *lines*
  - Samples along  $x$ : *pixels*  
(only for digital video)
- We perceive the images as continuous, not discrete:  
human visual system  
performs the interpolation !
- How many frames, lines, and pixels ?

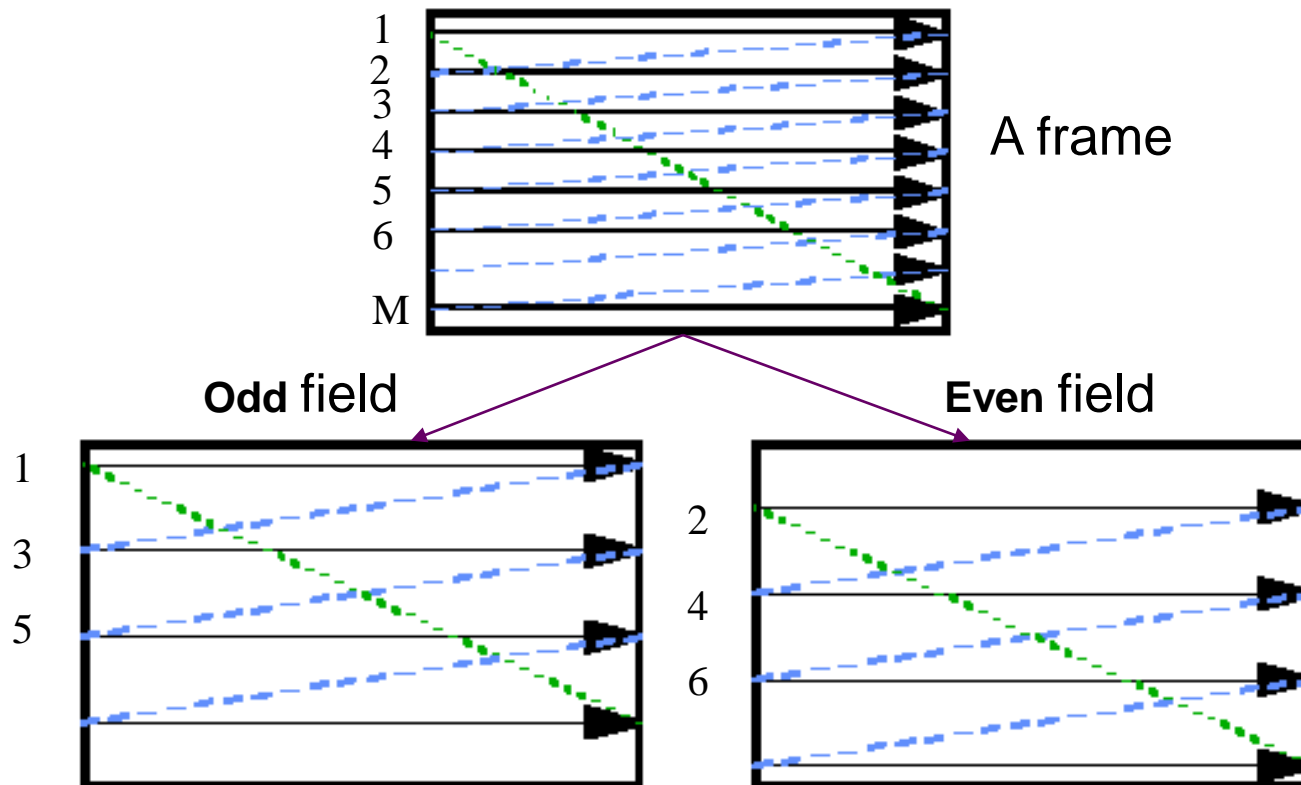


Progressive scanning



# Interlaced Scanning

- If the frame rate is too slow - > flickering and jagged movements
- Tradeoff between spatial and temporal resolution
  - Slow moving objects with high spatial resolution
  - Fast moving objects with high frame rate
- **Interlaced scanning:** scan all even lines, then scan all odd lines.
- A frame is divided into 2 fields (sampled at different time)



# RGB Color Model

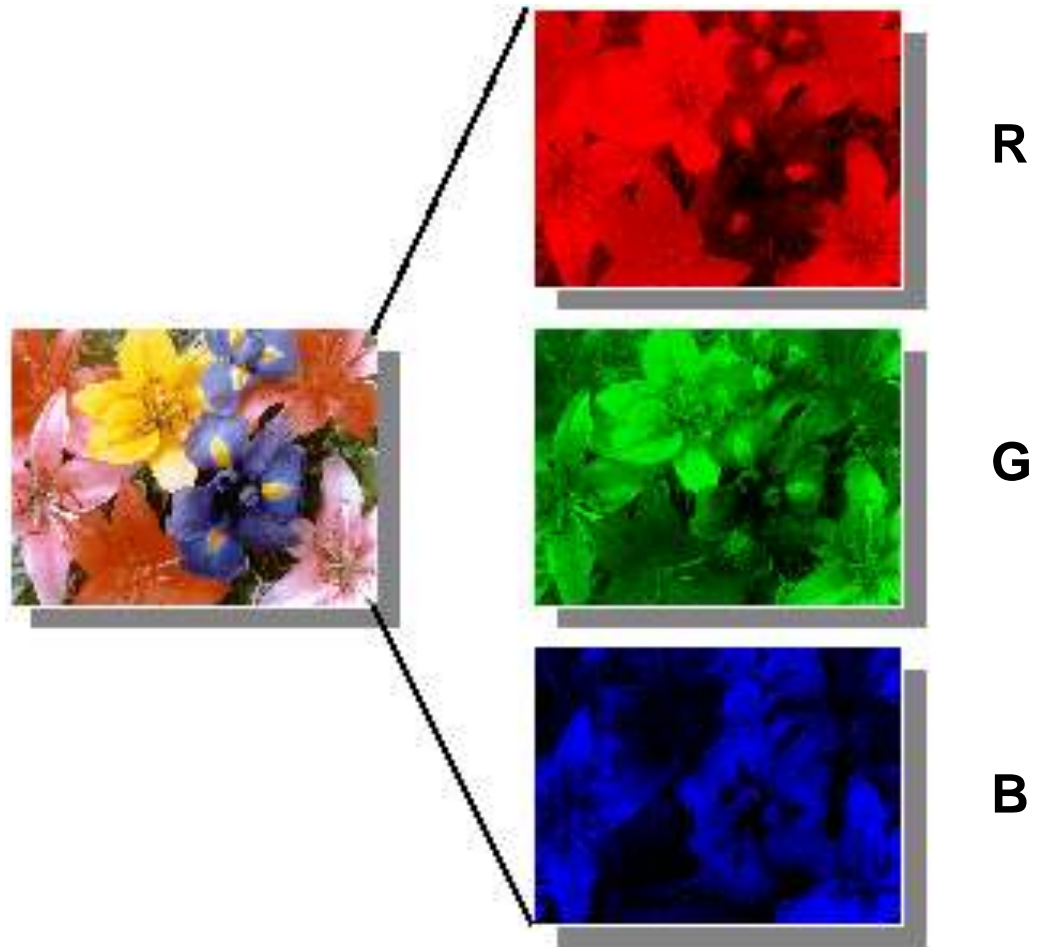
- Three basic colors

R: Red

G: Green

B: Blue

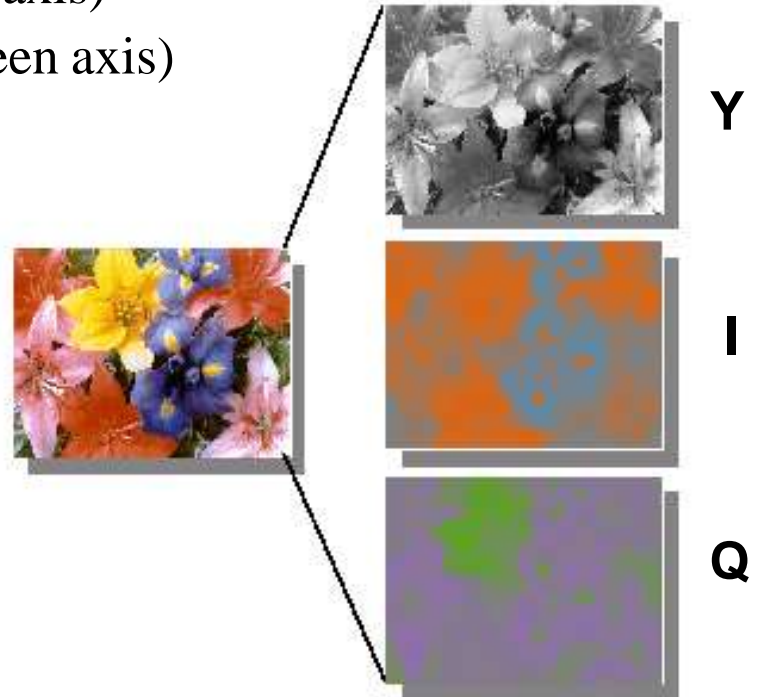
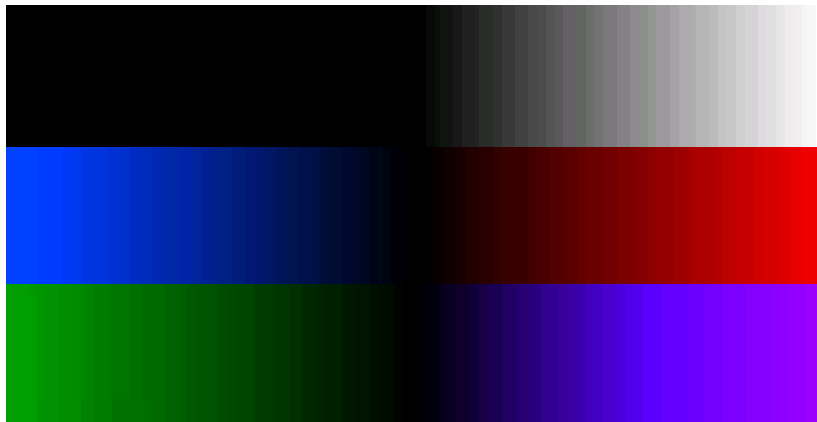
→ A picture consists of three images



# YIQ Color Model

**YIQ color model:** used in NTSC color TV

- **Y - Luminance** containing brightness and detail (monochrome TV)
- To create the Y signal, the red, green and blue inputs to the Y signal must be balanced to compensate for the color perception misbalance of the eye.
  - $Y = 0.3R + 0.59G + 0.11B$
- **Chrominance**
  - $I = 0.6R - 0.28G - 0.32B$  (cyan-orange axis)
  - $Q = 0.21R - 0.52G + 0.31B$  (purple-green axis)
- Human eyes are most sensitive to Y, next to I, next to Q.

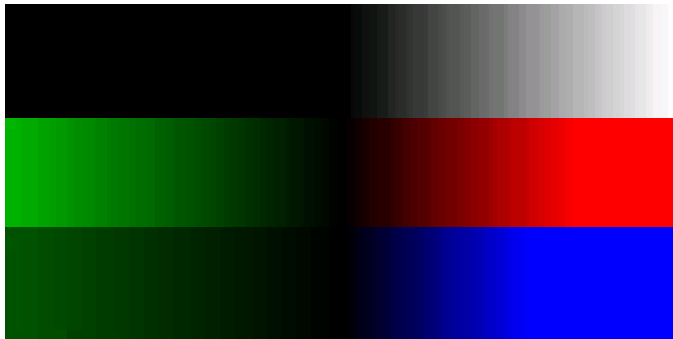


# YUV Color Model

- **YUV color model:** used for PAL TV and CCIR 601 standard
- Same definition for Y as in YIQ model
- Chrominance is defined by U and V – the color differences

$$- U = B - Y$$

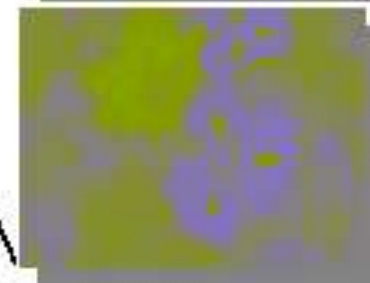
$$- V = R - Y$$



Y



U



V

# YCrCb Color Model

- **YCbCr color model**: used in JPEG and MPEG
- Closely related to YUV: scaled and shifted YUV
  - **$Cb = ((B - Y)/2) + 0.5$**
  - **$Cr = ((R - Y)/1.6) + 0.5$**
- Chrominance value in YCbCr are always in the range of **0 to 1** (normalization)  
→ *Make digital processing easy*

# Color Models in Video (Cont...)

- Color models based on linear transformation from RGB color space

$$C = M_{3 \times 3} \times C_{\text{RGB}}$$

YIQ (used in NTSC TV standard). Change of basis matrix:

$$\begin{pmatrix} 0.299 & 0.587 & 0.114 \\ 0.596 & -0.274 & -0.322 \\ 0.211 & -0.523 & 0.312 \end{pmatrix}$$

YUV (used in PAL and SECAM). Change of basis matrix:

$$\begin{pmatrix} 0.299 & 0.587 & 0.114 \\ -0.147 & -0.289 & 0.436 \\ 0.615 & -0.515 & -0.100 \end{pmatrix}$$

YCrCb (used in JPEG and MPEG). Change of basis matrix:

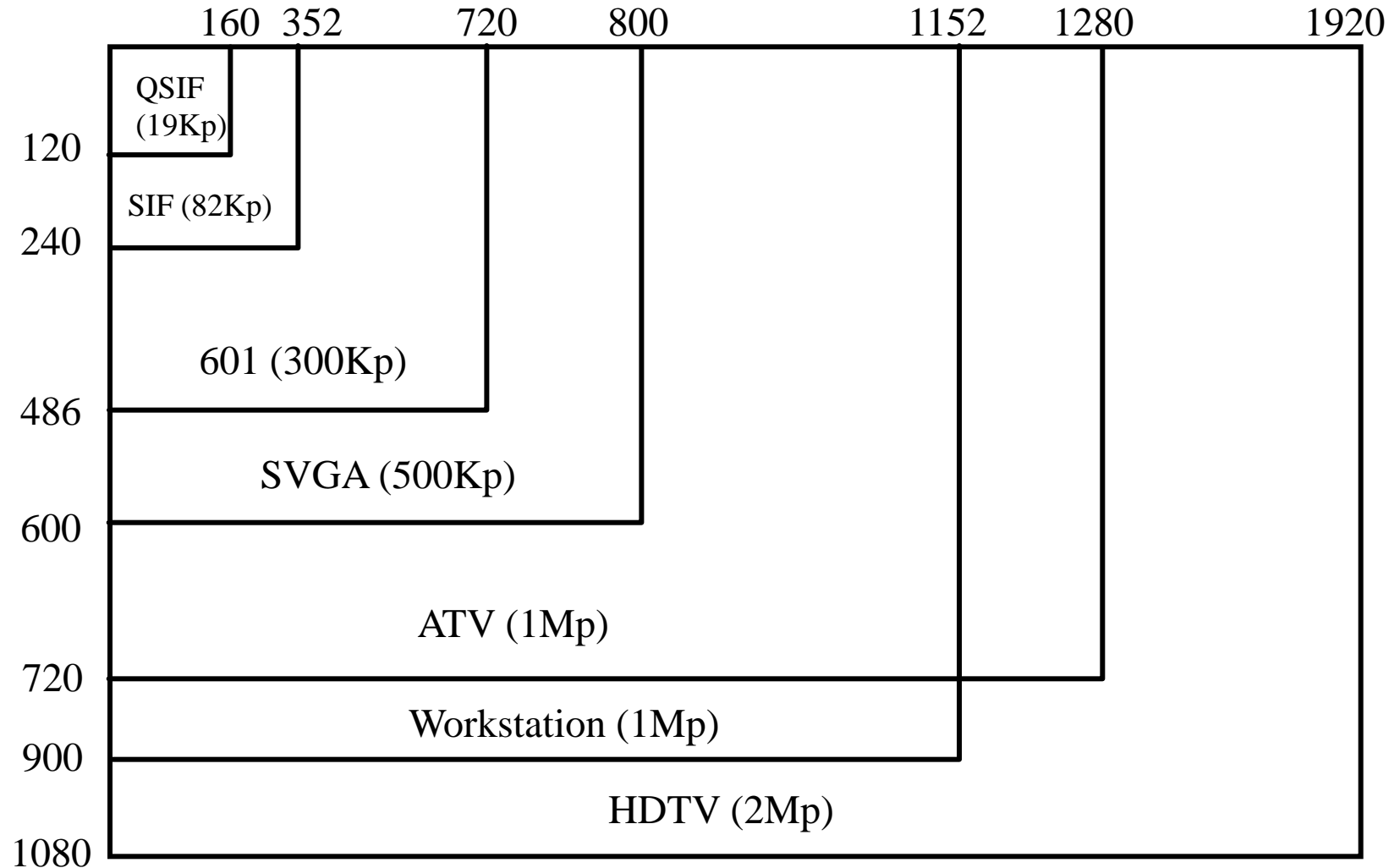
$$\begin{pmatrix} 0.2990 & 0.5870 & 0.1140 \\ 0.5000 & -0.4187 & -0.0813 \\ -0.1687 & -0.3313 & 0.5000 \end{pmatrix}$$

# Analog NTSC and PAL Video

- **NTSC Video:** Japan, US, ...
  - 525 scan lines per frame, 30 frames per second
  - Interlaced, each frame is divided into 2 fields, 262.5 lines/field
  - 20 lines reserved for control information at the beginning of each field
  - So a maximum of 485 lines of visible data
  - Color representation: YIQ color model
- **PAL Video:** China, UK, ...
  - 625 scan lines per frame, 25 frames per second (40 msec/frame)  
Interlaced, each frame is divided into 2 fields, 312.5 lines/field
  - Uses YUV color model
  - Approximately 20% more lines than NTSC
  - NTSC vs. PAL → roughly same bandwidth

# Digital Video

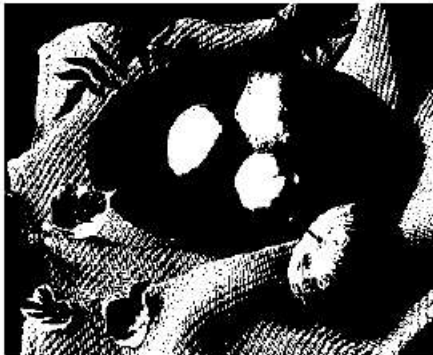
- Analog TV is a continuous signal
- Digital TV uses discrete numeric values
  - Signal is sampled, and samples are quantized
  - Sub-sampling to reduce image resolution or size
- Image represented by pixel array





# Sample Quantization – Pixel Resolution

- Pixel resolution depends quantization levels/bits
- Usually, 8 bits for each luma/chroma sample when no compression  
→ 8bits/1byte per pixel for gray image, 24bits/3bytes for true color image



(a)



(b)



(c)



(d)

Luminance (gray) picture

Num.	Level	Bit
(a)	2	1 (Monochrome)
(b)	4	2
(c)	8	3
(d)	16	4
(e)	32	5
(f)	64	6



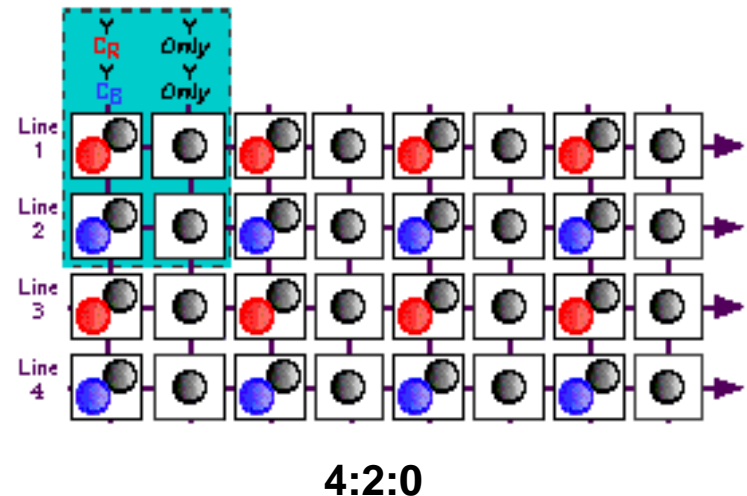
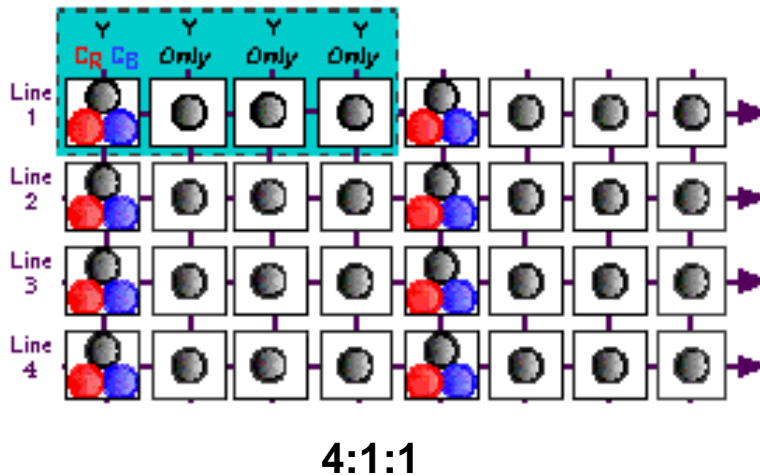
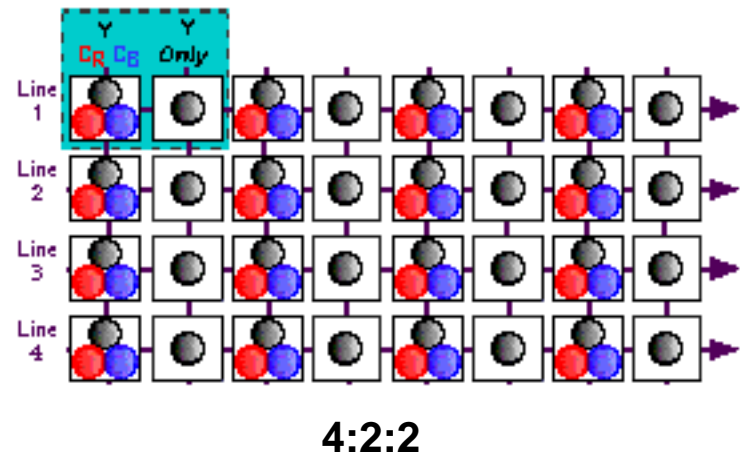
(e)



(f)

# Luma Sampling and Chroma Sub-Sampling

- *Chroma subsampling*: human visual system is **more sensitive to luminance than chrominance**  
→ We can subsample chrominance
- 4:4:4 – No subsampling
- 4:2:2, 4:1:1 – horizontally subsample
- 4:2:0 – horizontally and vertically



# Standards for Video

	HDTV	CCIR 601 NTSC	CCIR 601 PAL	CIF	QCIF
Luminance Resolution	1920 x 1080	720 x 486	720 x 576	352 x 288	176 x 144
Chrominance Resolution	960 x 540	360 x 486	360 x 576	176 x 144	88 x 72
Color Subsampling	4:2:2	4:2:2	4:2:2	4:2:0	4:2:0
Frames/sec	60	30	25	15	15
Aspect Ratio	16:9	4:3	4:3	4:3	4:3
Interlacing	Yes	Yes	Yes	No	No

CCIR – Consultative Committee for International Radio

CIF – Common Intermediate Format (approximately VHS quality)

QCIF – Quarter CIF

# Video Bit Rate Calculation

**width** ~ pixels (160, 320, 640, 720, 1280, 1920, ...)

**height** ~ pixels (120, 240, 480, 485, 720, 1080, ...)

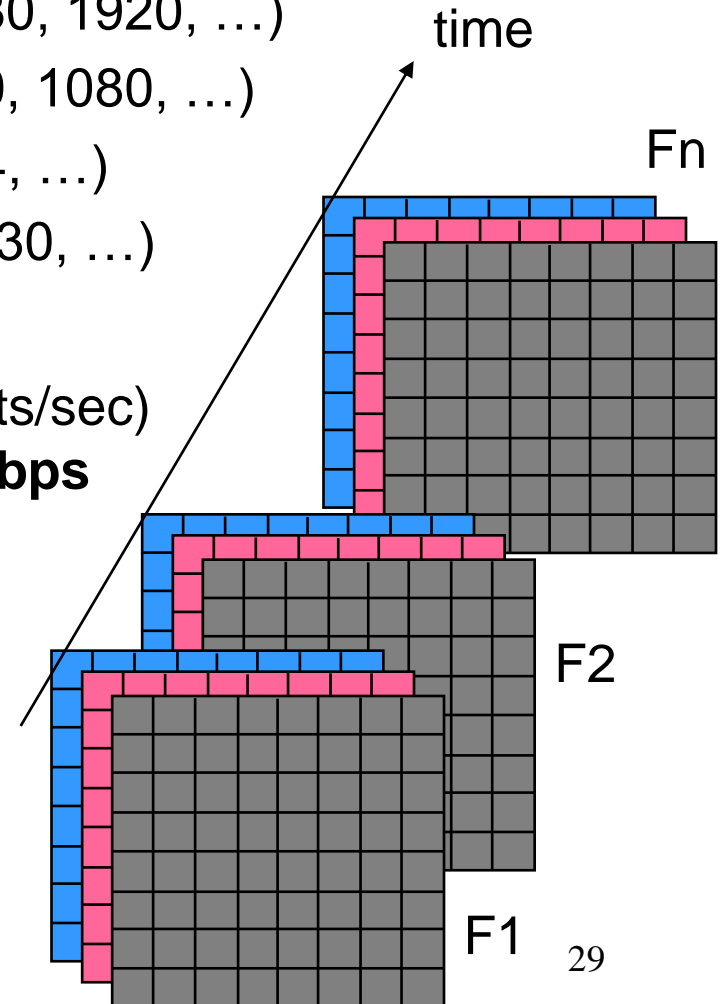
**depth** ~ bits per pixel (1, 4, 8, 15, 16, 24, ...)

**fps** ~ frames per second (5, 15, 20, 24, 30, ...)

Bit Rate = width \* height \* depth \* fps (bits/sec)

**bps**

One Frame =  
3 pictures  
(YCrCb)



# Data Rate of No-Compressed Video

- Example 1: Resolution 720x385, frame rate 30 frames per sec (fps)
  - $720 \times 485 = 349,200$  pixels/frame
  - 4:4:4 sampling gives  $720 \times 485 \times 3 = 1,047,600$  bytes/frame
  - 30fps  $\rightarrow 1.05\text{M} \times 30 = 31.5\text{MBytes/sec} \rightarrow 31.5\text{M} \times 8\text{bits} = \underline{250\text{Mbps}}$
  - 4:2:2 subsampling gives  $720 \times 485 \times 2 = 698,400$  bytes/frame
  - 30fps  $\rightarrow 0.698 \times 30 = 21\text{ MB/sec} \rightarrow 21\text{M} \times 8 = \underline{168\text{Mbps}}$
- Example 2: Resolution 1280x720, frame rate 30fps
  - $1280 \times 720 = 921,600$  pixels/frame
  - 4:2:0 subsampling gives  $921,600 \times 1.5 = 1,382,400$  bytes/frame
  - 30fps  $\rightarrow 1.38\text{M} \times 30 = 41\text{MB/sec} \rightarrow 41 \times 8 = \underline{328\text{Mbps}} \text{ (} \underline{656\text{Mbps}} \text{ 4:4:4)}$
- Example 3 Resolution 1080x1920, frame rate 60fps
  - $1080 \times 1920 = 2,073,600$  pixels per frame
  - 4:4:4 sampling =  $2,073,600 \times 3 = 6,220,800$  bytes/frame
  - 60fps  $\rightarrow 2,073,600 \times 60 = 373,248,000$  bytes per second  
 $\rightarrow 374\text{MB/s} = 374\text{M} \times 8 = \underline{3\text{Gbps}}$

-- bps (bit rate)  
bits per second

➔ **Conclusion: Compressing Digital Video !!!**

# Video Coding Standards Organizations

- **ITU-T**: International Telecommunication Union
  - Formerly CCITT
  - A United Nations Organization
  - Group: Video Coding Experts Group (VCEG)
  - Standards: **H.261**, **H.263**, **H.264**, etc
- **ISO**: International Standards Organization
  - Joint Photographic Experts Group (JPEG)
    - Standards: **JPEG/JPEG2000** (still image), MJPEG (motion picture)
  - Moving Picture Experts Group (MPEG)
    - Standards: **MPEG-1**, **MPEG-2**, **MPEG-4**, (**MPEG-7**, **MPEG-21**)
- ... and more!

# Demos of Image Color Models

# JPEG and H.26x Standards

- Video Data Size and Bit Rate
- DCT Transform and Quantization
- JPEG Standard for Still Image
- Intra-frame and Inter-frame Compression
- Block-based Motion Compensation
- H.261 Standard for Video Compression
- H.263, H.263+, H.263++, H.26L, H.264



# Video Bit Rate Calculation

**width** ~ pixels (160, 320, 640, 720, 1280, 1920, ...)

**height** ~ pixels (120, 240, 480, 485, 720, 1080, ...)

**depth** ~ bits per pixel (1, 4, 8, 15, 16, 24, ...)

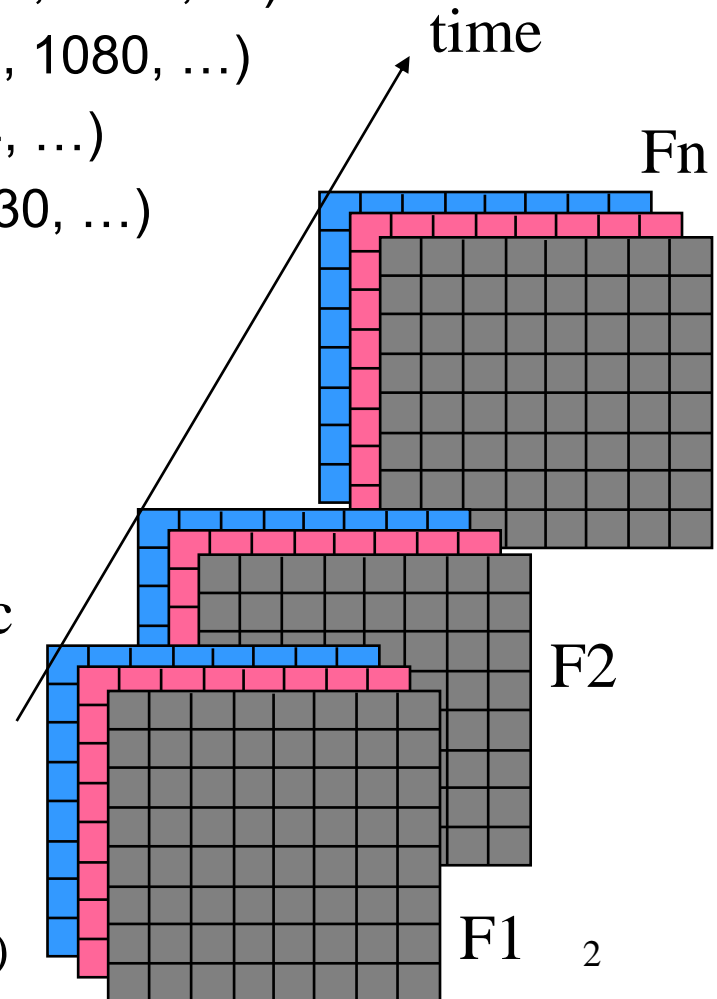
**fps** ~ frames per second (5, 15, 20, 24, 30, ...)

**compression factor** (1 ~ 100 ~ )

$$\left[ \frac{\text{width} * \text{height} * \text{depth} * \text{fps}}{\text{compression factor}} \right] = \text{bits/sec}$$

**bps**

One Frame =  
3 pictures (YCrCb)



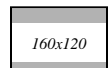
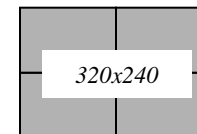
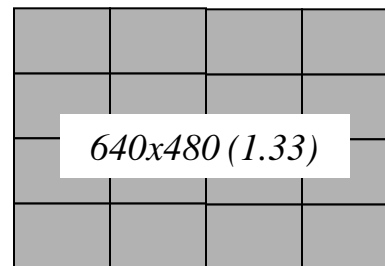
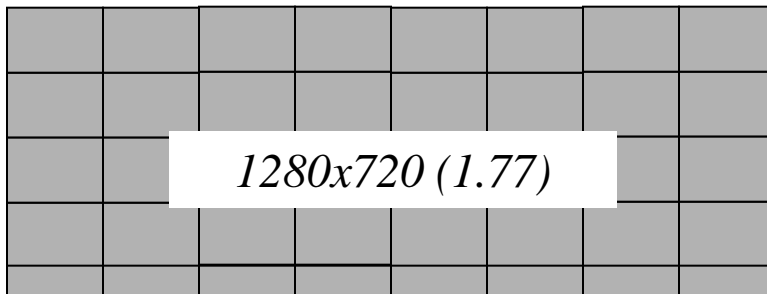
# Uncompressed Video Data Size

*compression factor = 1*

*Size of uncompressed video in gigabytes*

	1920x1080	1280x720	640x480	320x240	160x120
1 sec	0.19	0.08	0.03	0.01	0.00
1 min	11.20	4.98	1.66	0.41	0.10
1 hour	671.85	298.60	99.53	24.88	6.22
1000 hours	671,846.40	298,598.40	99,532.80	24,883.20	6,220.80

*Image size of video*



# Effects of Compression

*storage for 1 hour of compressed video in megabytes*

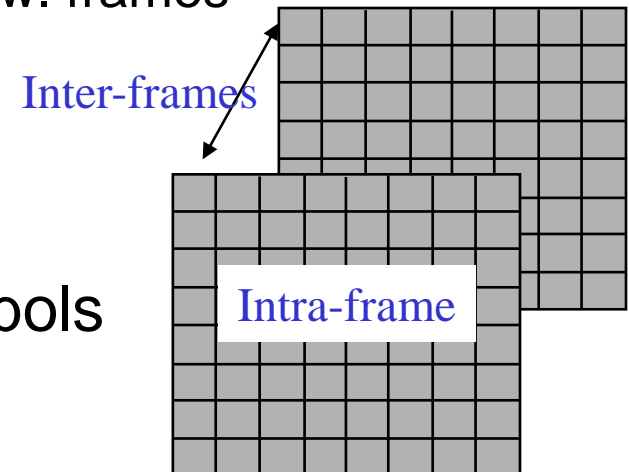
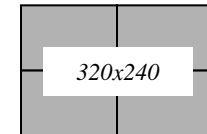
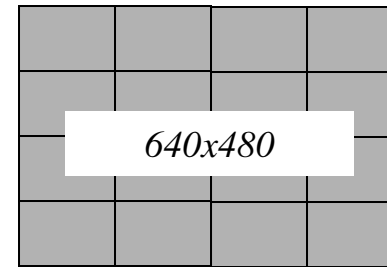
*Compression  
ration*

	1920x1080	1280x720	640x480	320x240	160x120
1:1	671,846	298,598	99,533	24,883	6,221
3:1	223,949	99,533	33,178	8,294	2,074
6:1	111,974	49,766	16,589	4,147	1,037
25:1	26,874	11,944	3,981	995	249
100:1	6,718	2,986	995	249	62

*3 bytes/pixel, 30 frames/sec*

# Coding Overview

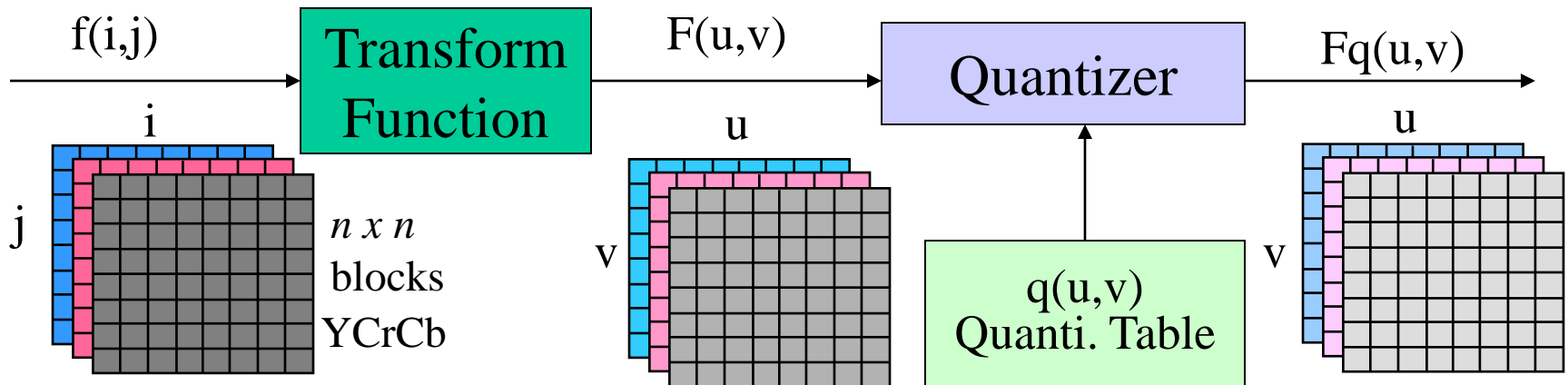
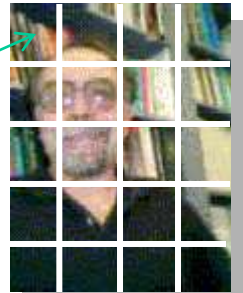
- Digitize
  - Subsample to reduce data
- Compression algorithms exploit:
  - Spatial redundancy - correlation between neighboring pixels
    - Intra-frame compression
    - remove redundancy within frame
  - Temporal redundancy - correlation betw. frames
    - Inter-frame compression
    - Remove redundancy between frames
- Symbol Coding
  - Efficient coding of sequence of symbols
    - RLC (Run Length Coding)
    - Huffman coding



# Transform Coding

- An image conversion process that transforms an image from the spatial domain to the frequency domain.
- Subdivide an individual  $N \times M$  image into small  $n \times n$  **blocks**
- Each  $n \times n$  block undergoes a **reversible transformation**
- **Basic approach:**
  - De-correlate the original block - radiant energy is redistributed amongst only a small number of transform coefficients
  - Discard many of the low energy coefficients (through quantization)

$N \times M$  image



# DCT – $n \times n$ Discrete Cosine Transform

$$F = D \times f \quad F, D, f \text{ are } n\text{-by-}n \text{ matrixes}$$

$$F[u,v] = \frac{4C(u)C(v)}{n^2} \sum_{j=0}^{n-1} \sum_{k=0}^{n-1} f(j,k) \cos \left[ \frac{(2j+1)u\pi}{2n} \right] \cos \left[ \frac{(2k+1)v\pi}{2n} \right]$$

where

$$C(w) = \begin{cases} \frac{1}{\sqrt{2}} & \text{for } w=0 \\ 1 & \text{for } w=1,2,\dots,n-1 \end{cases}$$

- IDCT is very similar
- 8x8 DCT coefficients

0.7	0.7	0.7	0.7	0.7	0.7	0.7	0.7
1	0.8	0.6	0.2	-0.2	-0.6	0.8	-1
0.9	0.4	-0.4	-0.9	-0.9	-0.4	0.4	0.9
0.8	-0.2	-1	-0.6	0.6	1	0.2	-0.8
0.7	-0.7	-0.7	0.7	0.7	-0.7	-0.7	0.7
0.6	-1	0.2	0.8	-0.8	-0.2	1	-0.6
0.4	-0.9	0.9	-0.4	-0.4	0.9	-0.9	0.4
0.2	-0.6	0.8	-1	1	-0.8	0.6	-0.2

# Quantization

- Purpose of quantization
  - Achieve high compression by representing DCT coefficients with no greater precision than necessary
  - Discard information which is not visually significant
- After output from the FDCT, each of the 64 DCT coefficients is quantized
  - Many-to-one-mapping => fundamentally lossy process
  - $F_q[u,v] = \text{Round} ( F[u,v] / q[u,v] )$
  - Example:  $F[u,v] = 101101 = 45$  (6 bits).  
If  $q[u,v] = 4$ , truncate to 4 bits,  $F_q[u,v] = 1011$

Example:  
2x2 block

$$F[u,v] = \begin{bmatrix} 45 & 12 \\ 8 & 3 \end{bmatrix} \quad Q[u,v] = \begin{bmatrix} 4 & 6 \\ 6 & 8 \end{bmatrix} \quad F_q[u,v] = \begin{bmatrix} 11 & 2 \\ 1 & 0 \end{bmatrix}$$

- Quantization is the principal source of lossiness in DCT-based encoders
- Uniform quantization: each  $F[u,v]$  is divided by the same constant  $N$
- Non-uniform quantization: use quantization tables from psychovisual experiments to exploit the limit of human visual system

# DCT and Quantization Example

DC component, others called AC

139	144	149	153	155	155	155	155	235.6	-1.0	12.1	-5.2	2.1	-1.7	-2.7	1.3	16	11	10	16	24	40	51	61
144	151	153	156	159	156	156	156	-22.6	-17.5	-6.2	-3.2	-2.9	-0.1	0.4	-1.2	12	12	14	19	26	58	60	55
150	155	160	163	158	156	156	156	-10.9	-9.3	-1.6	1.5	0.2	-0.9	-0.6	-0.1	14	13	16	24	40	57	69	56
159	161	162	160	160	159	159	159	-7.1	-1.9	0.2	1.5	0.9	-0.1	0.0	0.3	14	17	22	29	51	87	80	62
159	160	161	162	162	155	155	155	-0.6	-0.8	1.5	1.6	-0.1	-0.7	0.6	1.3	18	22	37	56	68	109	103	77
161	161	161	161	160	157	157	157	1.8	-0.2	1.6	-0.3	-0.8	1.5	1.0	-1.0	24	35	55	64	81	104	113	92
162	162	161	163	162	157	157	157	-1.3	-0.4	-0.3	-1.5	-0.5	1.7	1.1	-0.8	49	64	78	87	103	121	120	101
162	162	161	161	163	158	158	158	-2.6	1.6	-3.8	-1.8	1.9	1.2	-0.6	-0.4	72	92	95	98	112	100	103	99

(a) source image samples  $\mathbf{f}$

(b) forward DCT coefficients  $\mathbf{F}$

(c) quantization table  $\mathbf{Q}$

15	0	-1	0	0	0	0	0	240	0	-10	0	0	0	0	0	144	146	149	152	154	156	156	156
-2	-1	0	0	0	0	0	0	-24	-12	0	0	0	0	0	0	148	150	152	154	156	156	156	156
-1	-1	0	0	0	0	0	0	-14	-13	0	0	0	0	0	0	155	156	157	158	158	157	156	155
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	160	161	161	162	161	159	157	155
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	163	163	164	163	162	160	158	156
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	163	164	164	164	162	160	158	157
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	160	161	162	162	162	161	159	158
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	158	159	161	161	162	161	159	158

(d) normalized quantized  $\mathbf{Fq}$  coefficients

(e) denormalized quantized  $\mathbf{F}^{-1}$  coefficients

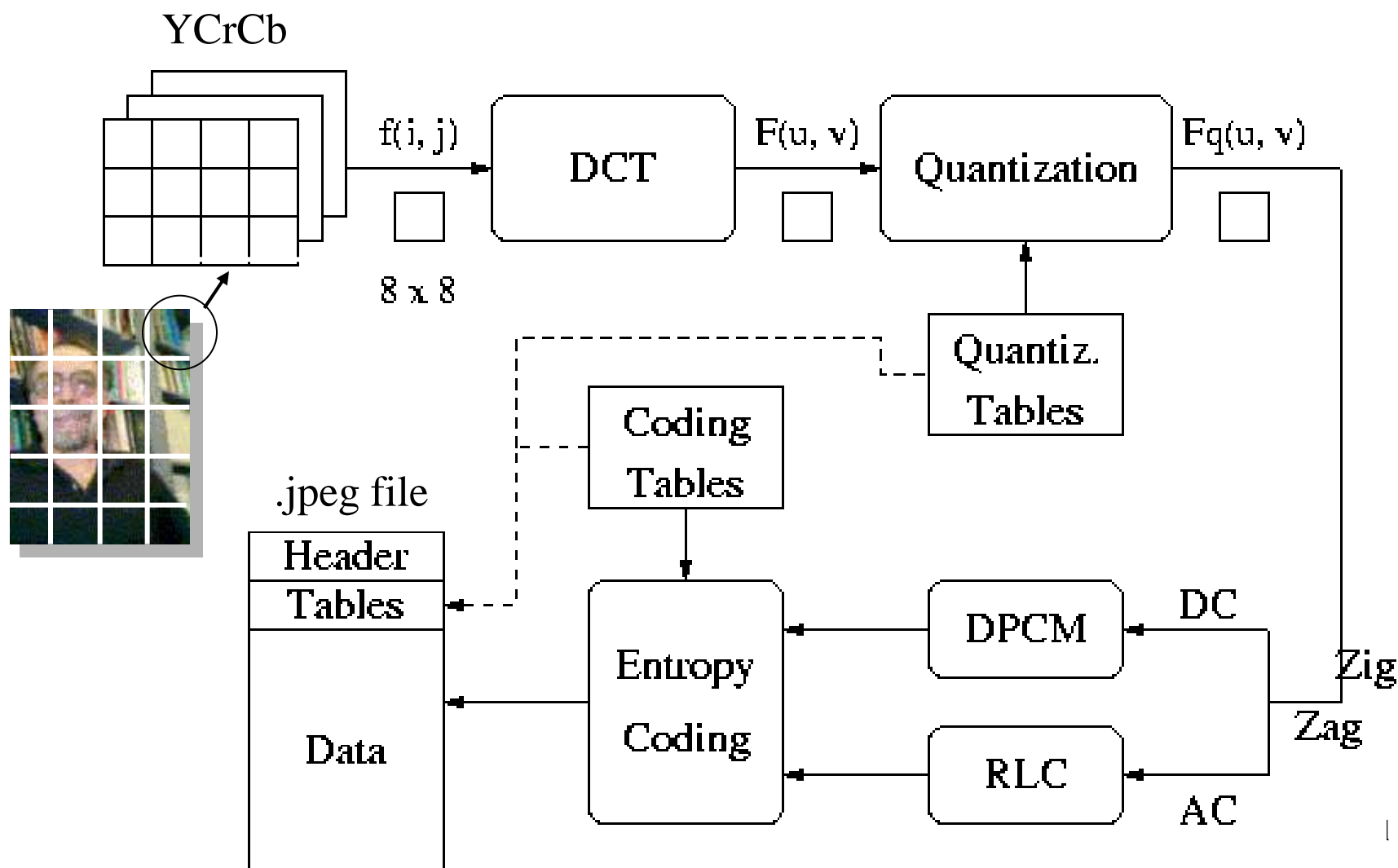
(f) reconstructed image samples  $\mathbf{f}^{-1}$



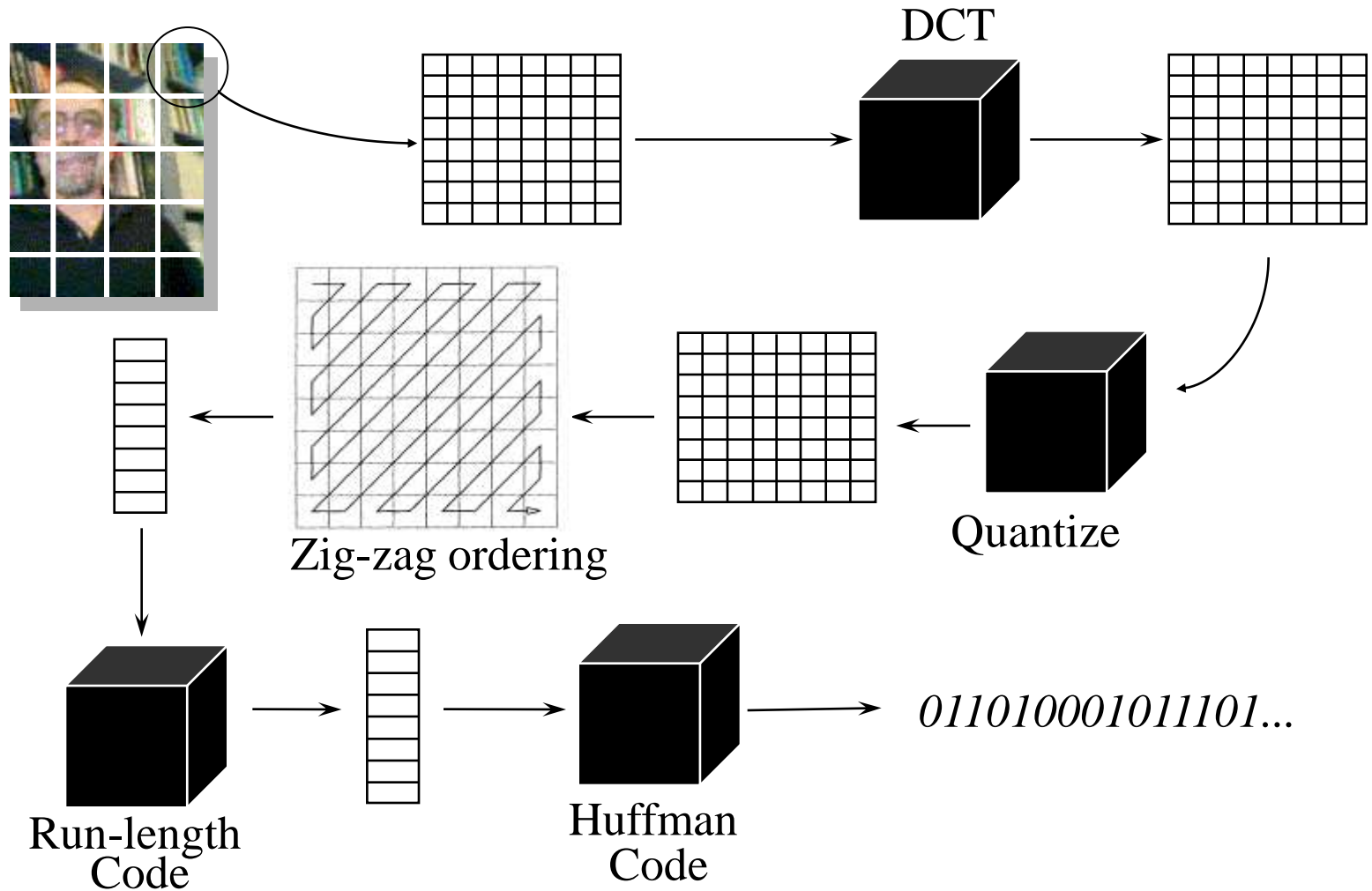
# JPEG Image Compression Standard

- Mainly for still image (gray and color)
- Four Modes:
  - Lossless JPEG
  - Sequential (Baseline) JPEG
  - Progressive JPEG
  - Hierarchical JPEG
- Hybrid Coding Techniques:
  - DCT Coding
  - Run Length Encoding(RLE)
  - Huffman Coding
  - Linear Prediction (only in lossless mode)
- New Standard: JPEG2000
- Motion JPEG for video

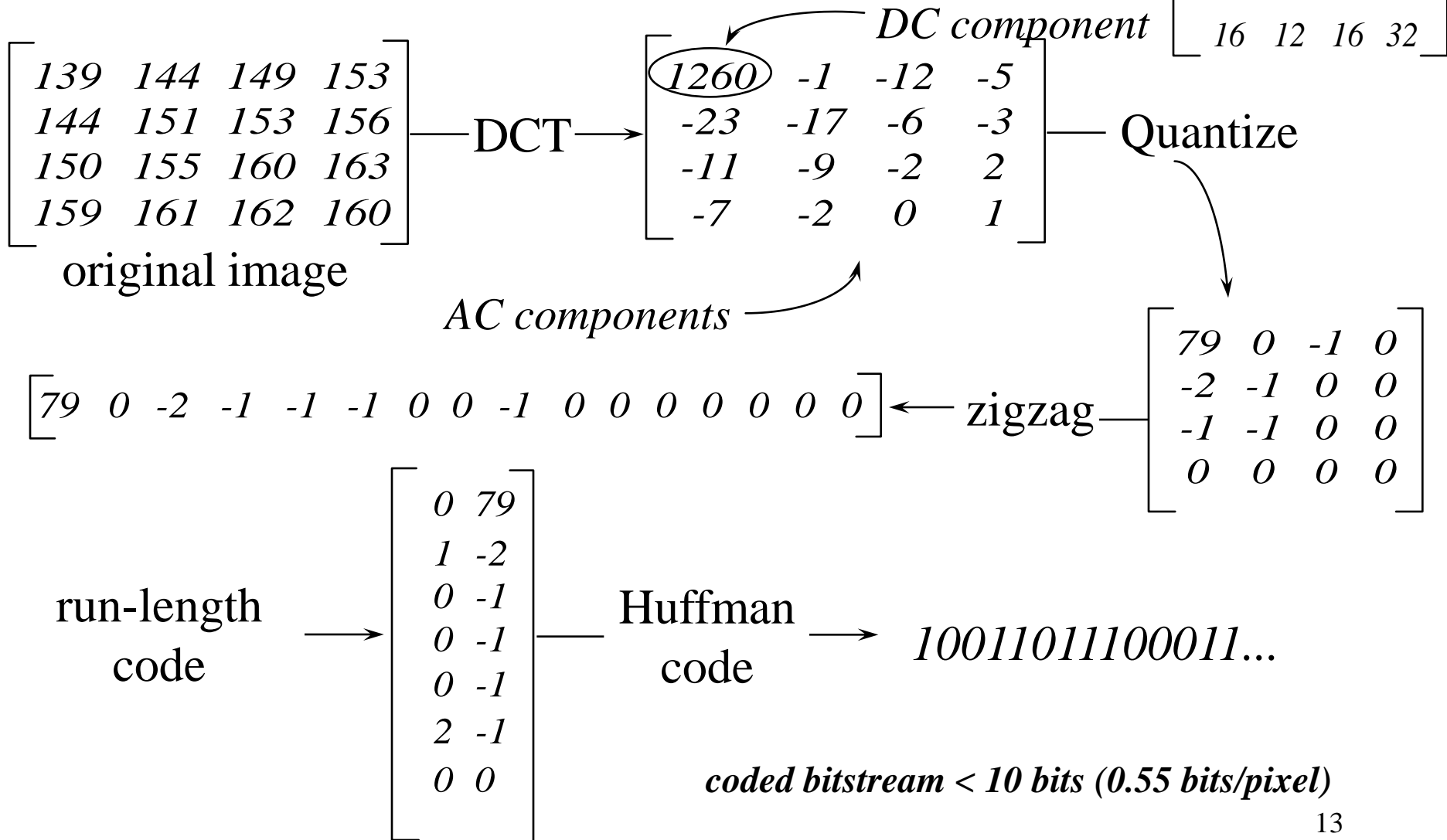
# Overview of Baseline JPEG



# Block Transform Encoding



# Example of Block Encoding



# Result of Coding/Decoding

$$\begin{bmatrix} 139 & 144 & 149 & 153 \\ 144 & 151 & 153 & 156 \\ 150 & 155 & 160 & 163 \\ 159 & 161 & 162 & 160 \end{bmatrix}$$

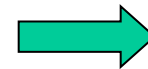
*original block*

$$\begin{bmatrix} 144 & 146 & 149 & 152 \\ 156 & 150 & 152 & 154 \\ 155 & 156 & 157 & 158 \\ 160 & 161 & 161 & 162 \end{bmatrix}$$

*reconstructed block*

$$\begin{bmatrix} -5 & -2 & 0 & 1 \\ -4 & 1 & 1 & 2 \\ -5 & -1 & 3 & 5 \\ -1 & 0 & 1 & -2 \end{bmatrix}$$

*errors*



*Small Loss  
Neglect-able*

# Examples



*Uncompressed*  
(262 KB)

*8 bits/pixel*



*Compressed (50)*  
(22 KB, 12:1)

*0.67 bit/pixel*



*Compressed (1)*  
(6 KB, 43:1)

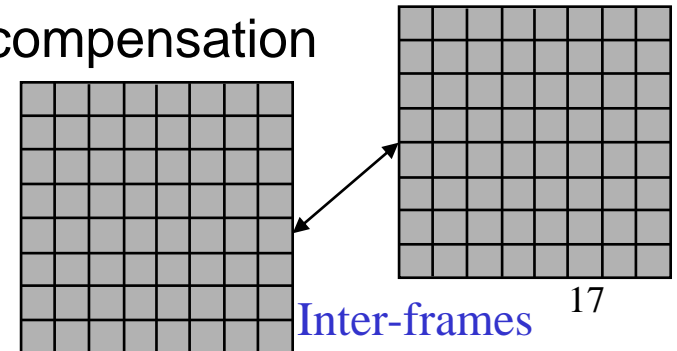
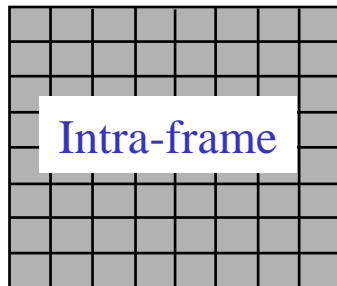
*0.17 bit/pixel*

# JPEG vs. GIF

- JPEG Advantages
  - more colors (GIF limited to 256)
  - lossless option
  - best for scanned photographs
  - progressive JPEG downloads rough image before whole image arrives
- GIF Advantages
  - transparent color setting
  - animated GIFs
  - better for flat color fields: clip art, cartoons, etc.
  - interlaced delivery downloads low resolution image before whole image arrives

# Intra- vs. Inter-frame Compression

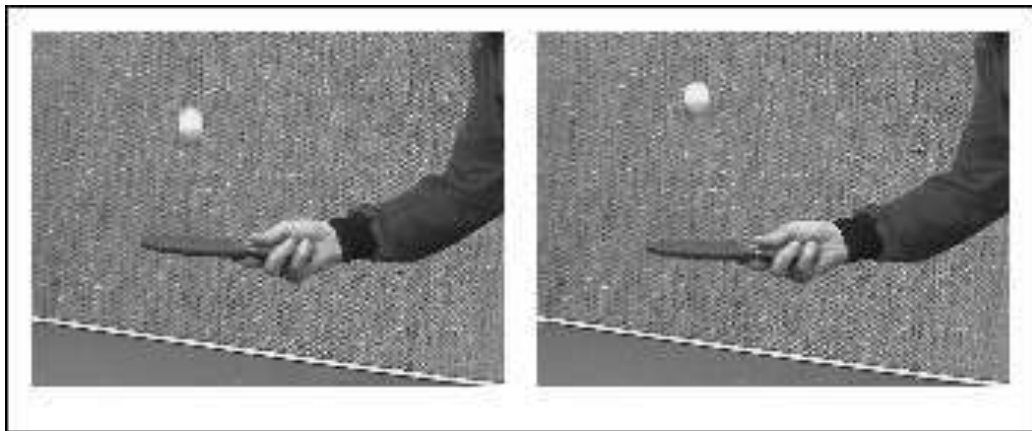
- **Intra-frame compression**
  - For still image like JPEG
  - Exploit the redundancy in image (*spatial redundancy*)
  - Can be applied to individual frames in a video sequence
- Techniques
  - Subsampling (small size)
  - Block transform coding
  - Coarse quantization
- **Intra + inter-frame compression**
  - For video like H.26x & MPEG
  - Exploit the similarities between successive frames (*temporal redundancy*)
- Techniques
  - Subsampling (small frame rate)
  - Difference coding
  - Block-based difference coding
  - Block-based motion compensation





# Difference Coding

- Compare pixels with previous frame
  - Only pixels that have been changed are updated
  - A fraction of the number of pixel values will be recorded



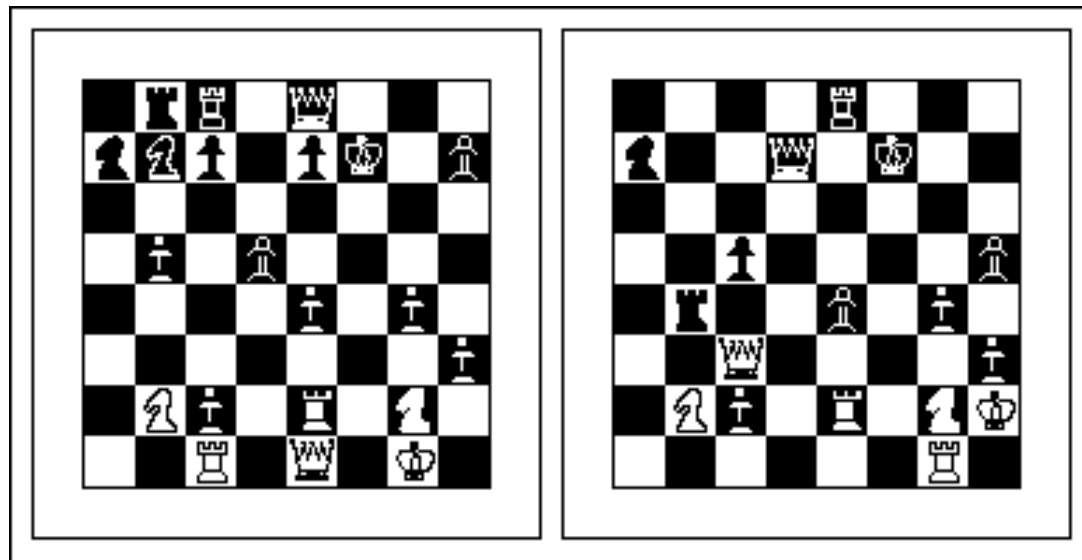
- Overhead associated with which pixels are updated: what if a large number of pixels are changed ?
- Pixels values are slightly different even with no movement of objects: ignore small changes (lossy)

# Block-based Difference Coding

- Difference coding *at the block level*
  - Send sequence of blocks rather than frames
  - If previous block similar, skip it or send difference
  - Update a whole block of pixels at once
  - 160 x 120 pixels (19200 pixels) => 8x8 blocks (300 blocks)
  - Possible artifact at the border of blocks
- Limitations of difference coding
  - Useless where there is a lot of motion (few pixels unchanged)
  - What if a camera itself is moving ?
- Need to compensate for object motion

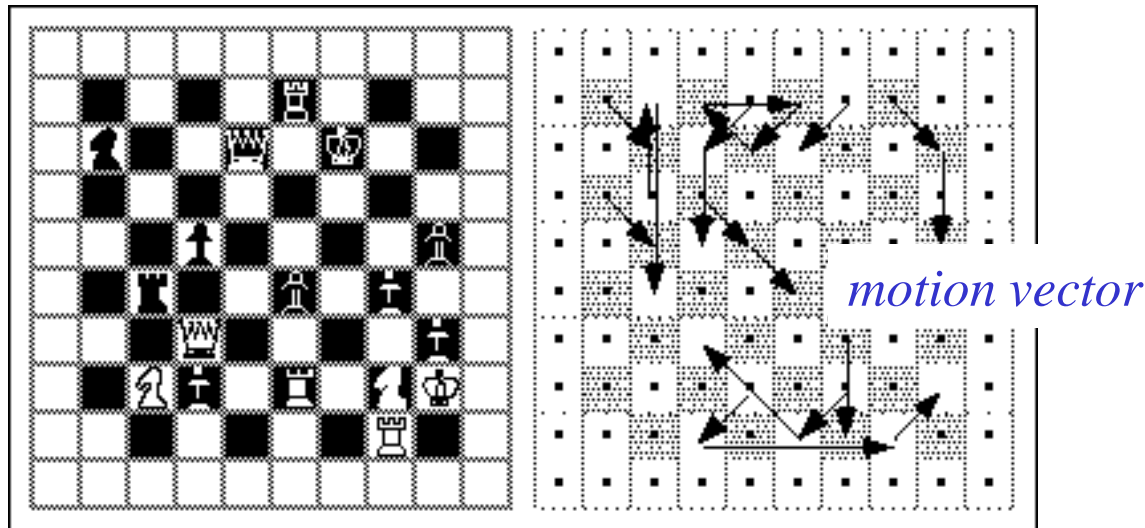
# Block-based Motion Compensation

- **Motion compensation** assumes that current frame can be modeled as a translation of a previous frame
- Search around block in previous frame for a better matching block and encode position and error difference



# Block-based Motion Compensation

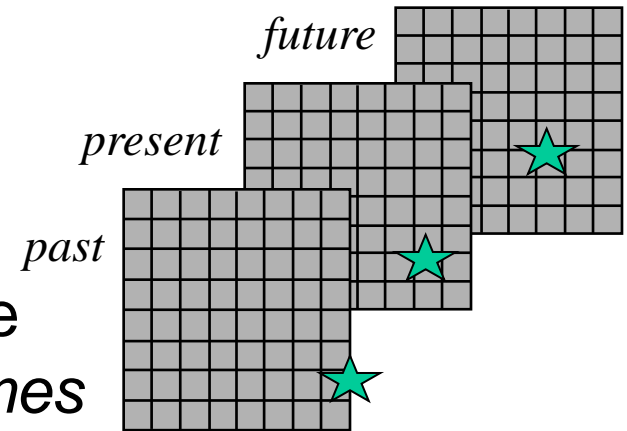
- Current frame is divided into *uniform non-overlapping blocks*
- Each block in the current frame is compared to areas of similar size from the preceding frame in order to find an area that is similar
- The relative difference in locations is known as the *motion vector*
- Because fewer bits are required to code a motion vector than to code actual blocks, compression is achieved.



# Bidirectional Motion Compensation

- **Bidirectional motion compensation**

- Areas just uncovered are not predictable from the past, but can be predicted from the future
- Search in *both past and future frames*



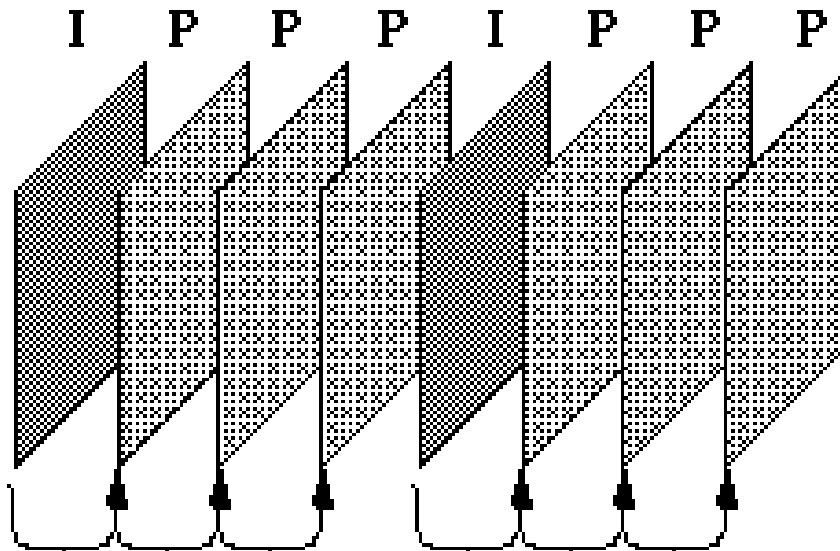
- Effect of noise and errors can be reduced by averaging between previous and future frames
- Bi-directional interpolation provides a high degree of compression
  - Requires that frames be encoded and transmitted in a different order from which they will be displayed.
- In reality, exact matching is not possible, thus lossy compression

# Overview of H.261

- Developed by CCITT (Consultative Committee for International Telephone and Telegraph) in 1988-1990
- Designed for videoconferencing, video-telephone applications over ISDN telephone lines.
  - Bit-rate is  $p \times 64$  Kbps, where  $p$  ranges from 1 to 30 (2048 kbps)
- Supports CCIR 601 CIF (352 x 288) and QCIF (176 x 144) images with 4:2:0 subsampling.
- Significant influence on H.263, MPEG 1-4, etc.

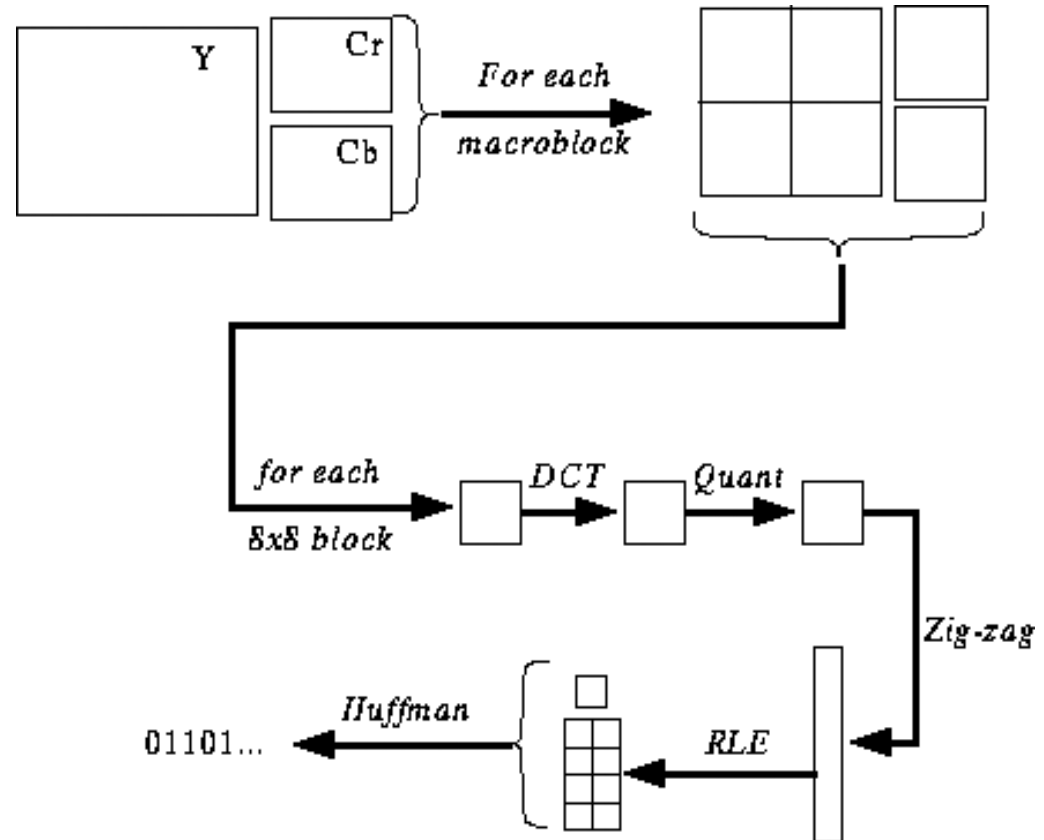
# Frame Sequence of H.261

- Two frame types: Intra-frames (*I-frames*) and Inter-frames (*P-frames*): I-frame provides an accessing point, it uses basically JPEG.
- P-frames use "**pseudo-differences**" from previous frame ("predicted"), so frames depend on each other.



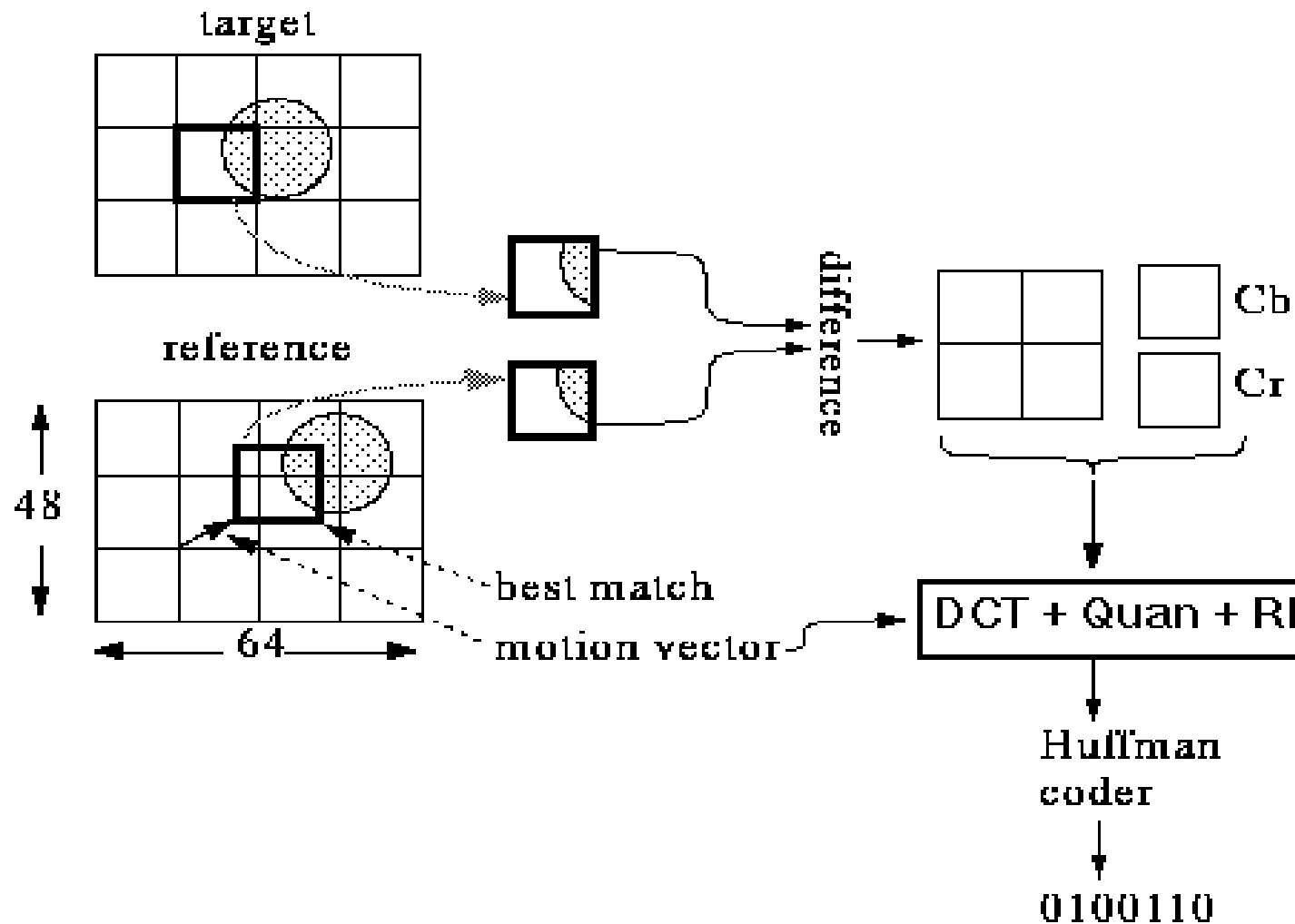
# Intra-frame Coding

- **Macroblock:**
  - 16 x 16 pixel areas on Y plane of original image.
  - Usually consists of 4 Y blocks, 1 Cr block, and 1 Cb block (4:2:0 or 4:1:1)
- Quantization is by constant value for all DCT coefficients (i.e., no quantization table as in JPEG).



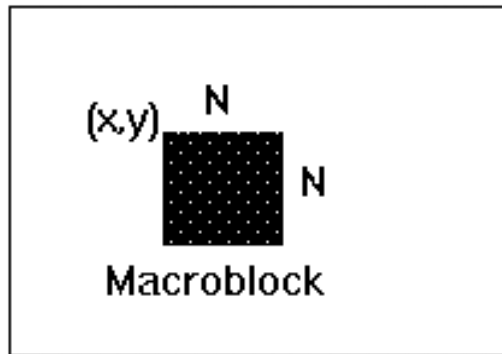


# Inter-frame Coding



# Motion Vector Searches

Target

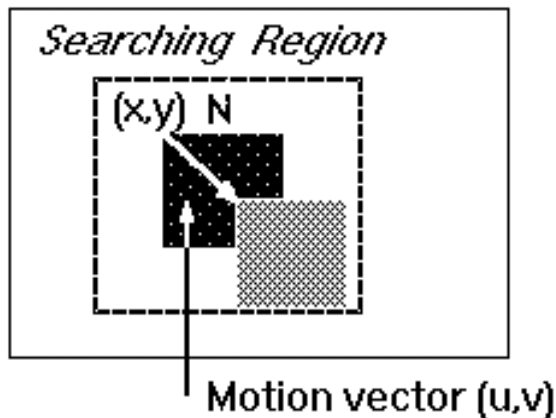


$C(x+k, y+l)$ : macro block pixels in the target  
 $R(x+i+k, y+j+l)$ : macro block pixels in the reference

$$MAE(i, j) =$$

$$\frac{1}{N^2} \sum_{k=0}^{N-1} \sum_{l=0}^{N-1} |C(x+k, y+l) - R(x+i+k, y+j+l)|$$

Reference

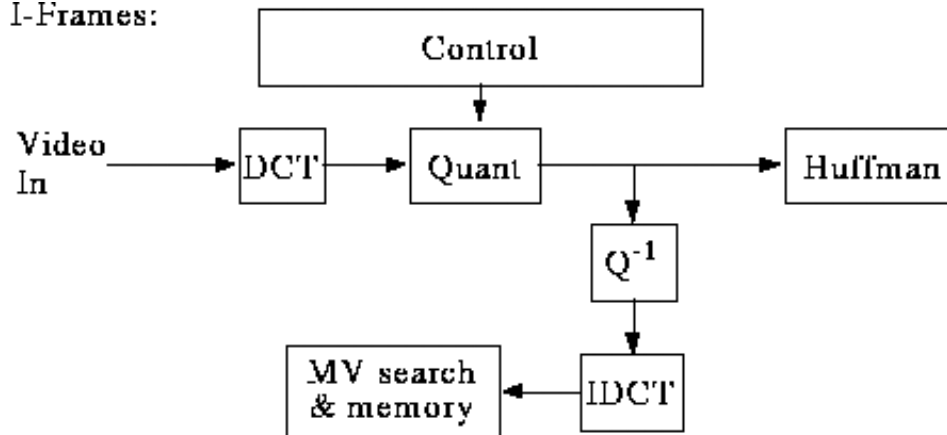


The goal is to find a vector  $(u, v)$  such that the mean Absolute Error,  $MAE(u, v)$  is minimum:

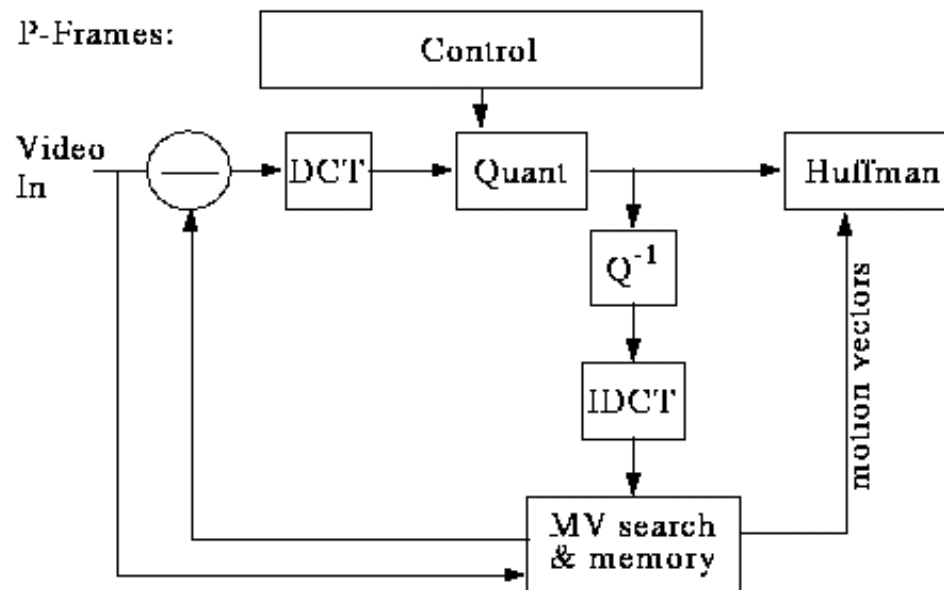
1. Full Search Method
2. Two-dimensional Logarithmic Search
3. Hierarchical Motion Estimation

# Encoder

I-Frames:



P-Frames:



# H.262, H.263 and H.264

- H.262 = MPEG-2 jointly by ITU and ISO/IEC
- ITU-T Rec. H.263 v1 (1995)
  - Current best standard for practical video telecommunication
  - Has overtaken H.261 as videoconferencing codec
  - Superior to H.261 at all bit rates (1/2)
  - Video size: Sub-QCIF (128x96), QCIF (176x144), CIF(352x288), 4CIF(704X576), 16CIF (1408x1152)
  - PB frames mode (bidirectional prediction)
  - 4 motion vector for each block,  $\frac{1}{2}$  pixel accuracy
  - Arithmetic coding efficient than Huffman coding in H.261
- H.263 v2 (H.263+, 1997)
- H.263 v3 (H.263++, 2000), H.26L (2002)
- H.264/AVC (now)

# Demos of Image GIF and JPEG Coding

# MPEG Standards

- **MPEG**
  - **M**oving **P**icture **E**xperts **G**roup
- Standards
  - MPEG-1
  - MPEG-2
  - MPEG-4
  - MPEG-7
  - MPEG-21

# What is MPEG

- MPEG: Moving Picture Experts Group
  - established in 1988
- ISO/IEC JTC 1 /SC 29 / WG 11
  - Int. Standards Org. / Int. Electro-technical Commission
  - Joint Technical Committee Number 1
  - Subcommittee 29, Working Group 11
- Develop standards for the coded representation of
  - moving picture and associated audio
- Sometimes collaborating with other standard organization
  - VCEG (ITU-T Video Coding Experts Group)
  - W3C (World Wide Web Consortium)
  - Web3D (Web3D Consortium, precious VRML)

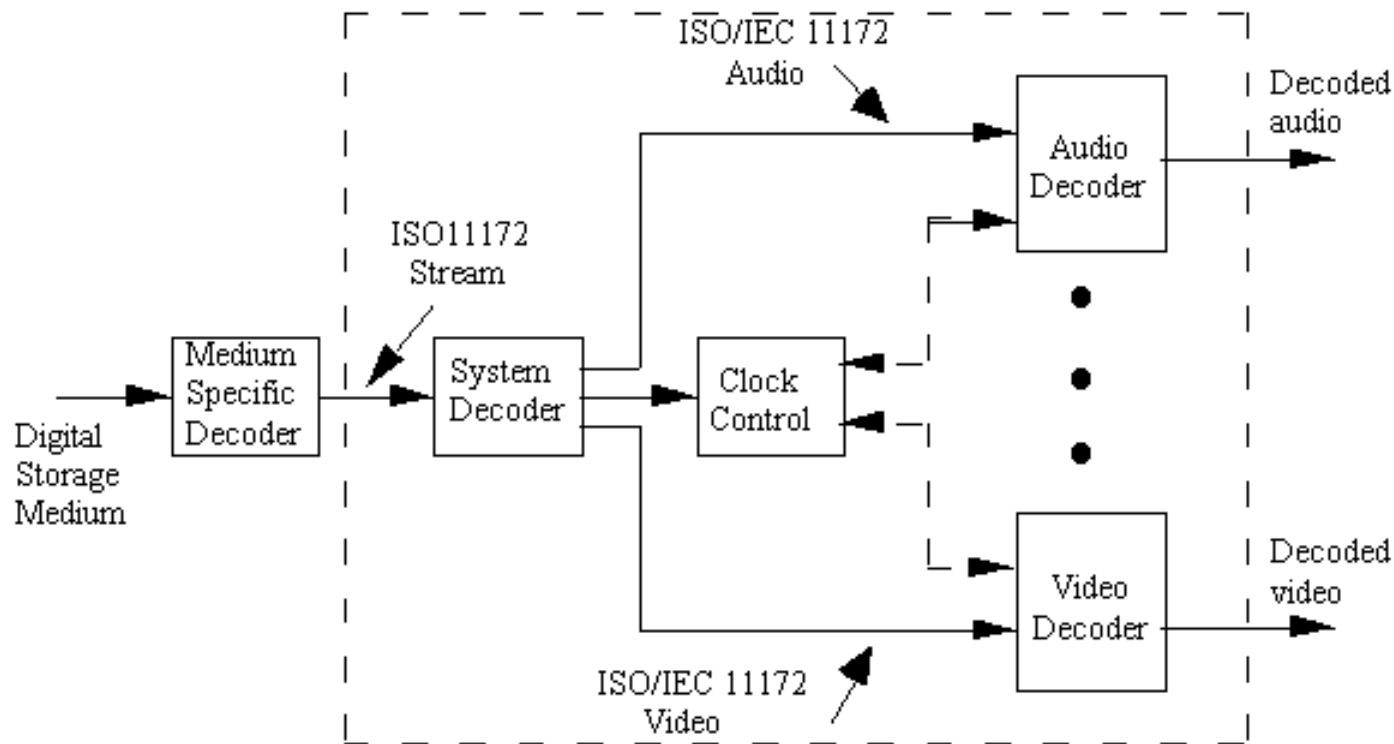
# Overview of MPEG Standards

- MPEG-1 (1992)
  - Coding of video and audio for storage media (CD-ROM, 1.5Mbps)
  - VCD, MP3
- MPEG-2 (1994)
  - Coding of video and audio for transport and storage (4~80Mbps)
  - Digital TV (HDTV) and DVD
- MPEG-4 (v1:1999, v2: 2000, v3: 2001)
  - Coding of natural and synthetic media objects
  - Web and mobile applications
- MPEG-7 (2001~)
  - Multimedia content description for AV materials
  - Media searching and filtering
- MPEG-21 (2001~)
  - Multimedia framework for integration of multimedia technologies
  - Transparent and augmented use of multimedia resources



# MPEG-1 System

- Standard had three parts/layers: Video, Audio, and System (control interleaving of streams)
- combines one or more data streams from the video and audio parts with timing information to form a single stream suited to digital storage or transmission



# MPEG-1 Video Layer

- For compressing video (NTSC 625-line and 525-lines)
- CIF/SIF (352x288/240)
- YCrCb: **4:2:0** sub-sampling
- Storage media at continuous rate of about **1.5 Mbps**
- **Intra-frame encoding:** DCT-based compression for the reduction of spatial redundancy (similar to JPEG)
- **Inter-frame encoding:** block-based *bidirectional* motion compensation for the reduction of temporal redundancy
- The difference signal, the prediction error, is further compressed using the discrete cosine transform (DCT) to remove spatial correlation and is then quantized.
- Finally, the motion vectors are combined with the DCT information, and coded using variable length codes

# Frame Sequence of MPEG-1

- **I-frames**

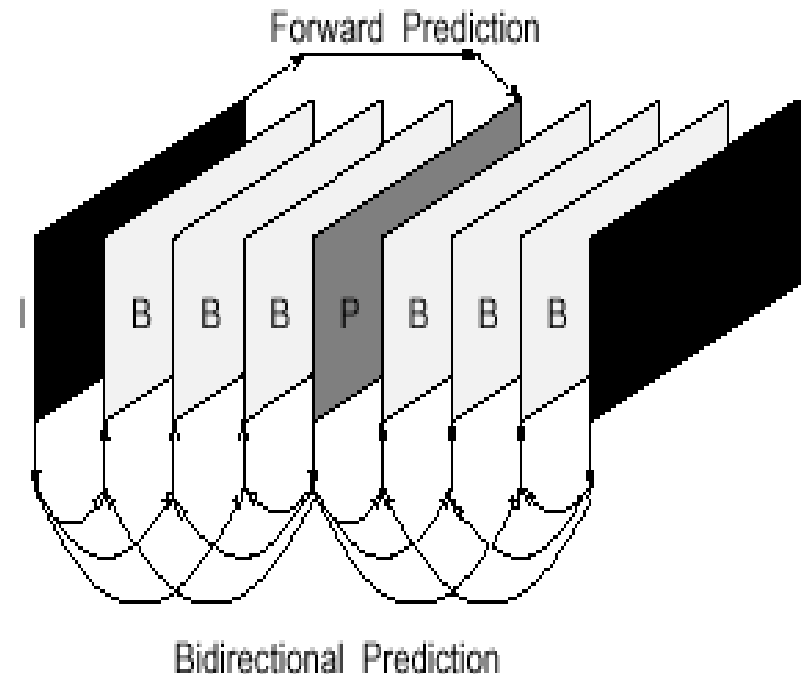
- Intra-coded frames providing access points for random access
- Moderate compression

- **P-frames**

- Predicted frames with reference to a previous **I** or **P** frame

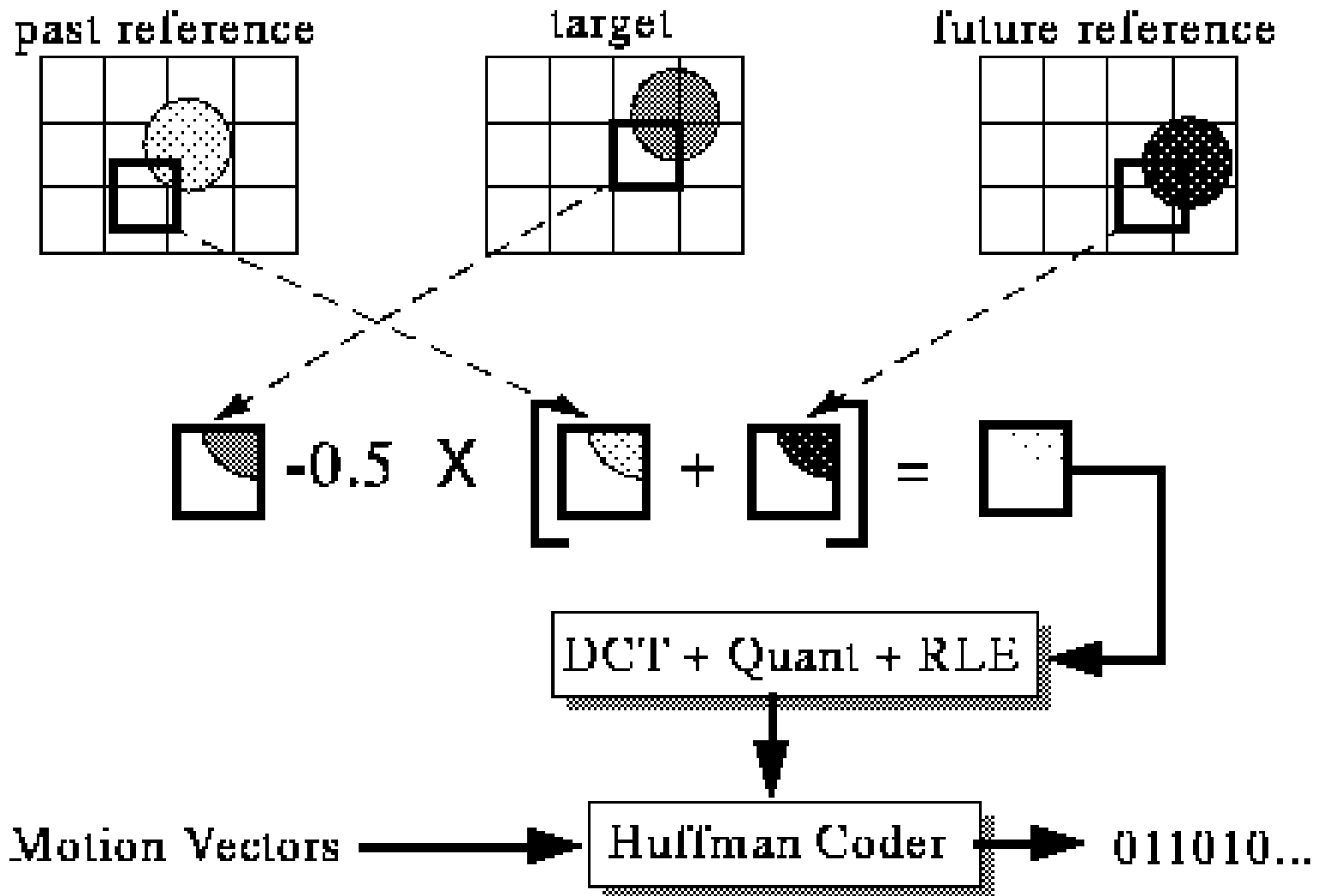
- **B-frames**

- Bidirectional frames encoded using the previous and the next **I/P** frames
- Maximum compression

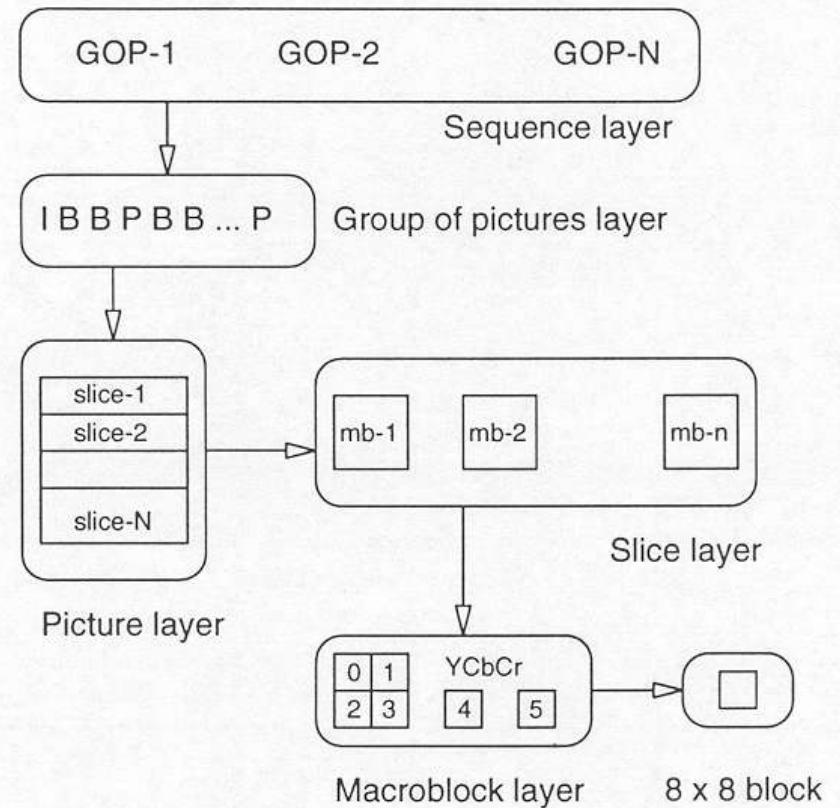
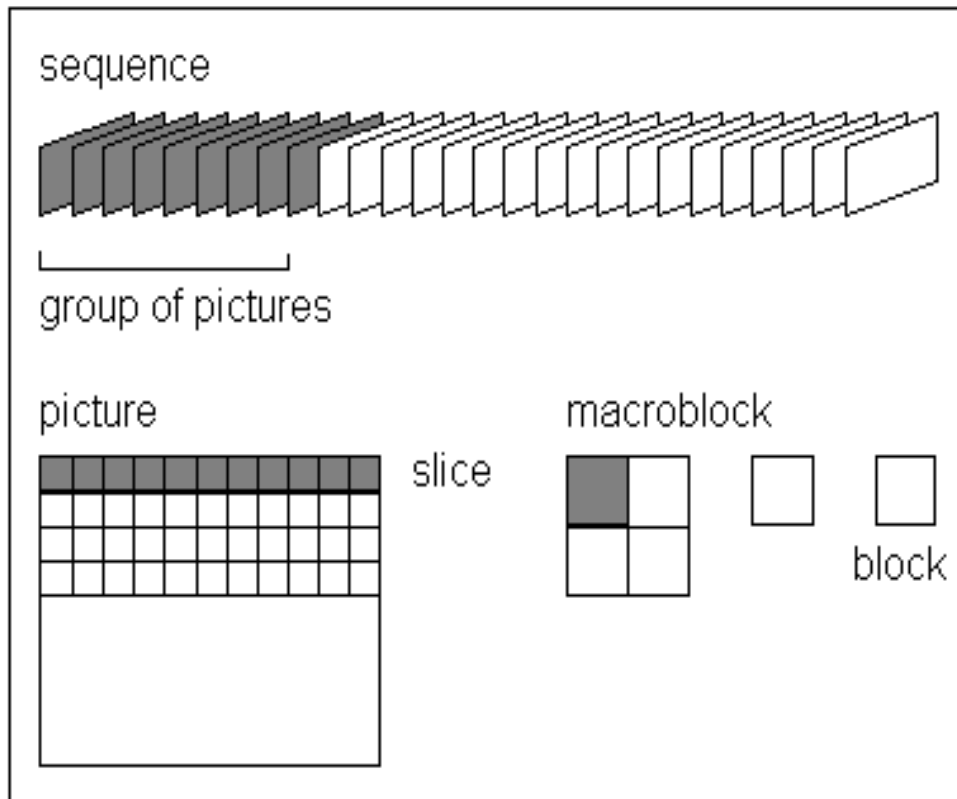


Fr. Type	Size	Compr Ratio
I	18 KB	7:1
P	6 KB	20:1
B	2.5 KB	50:1
Average	4.8 KB	27:1

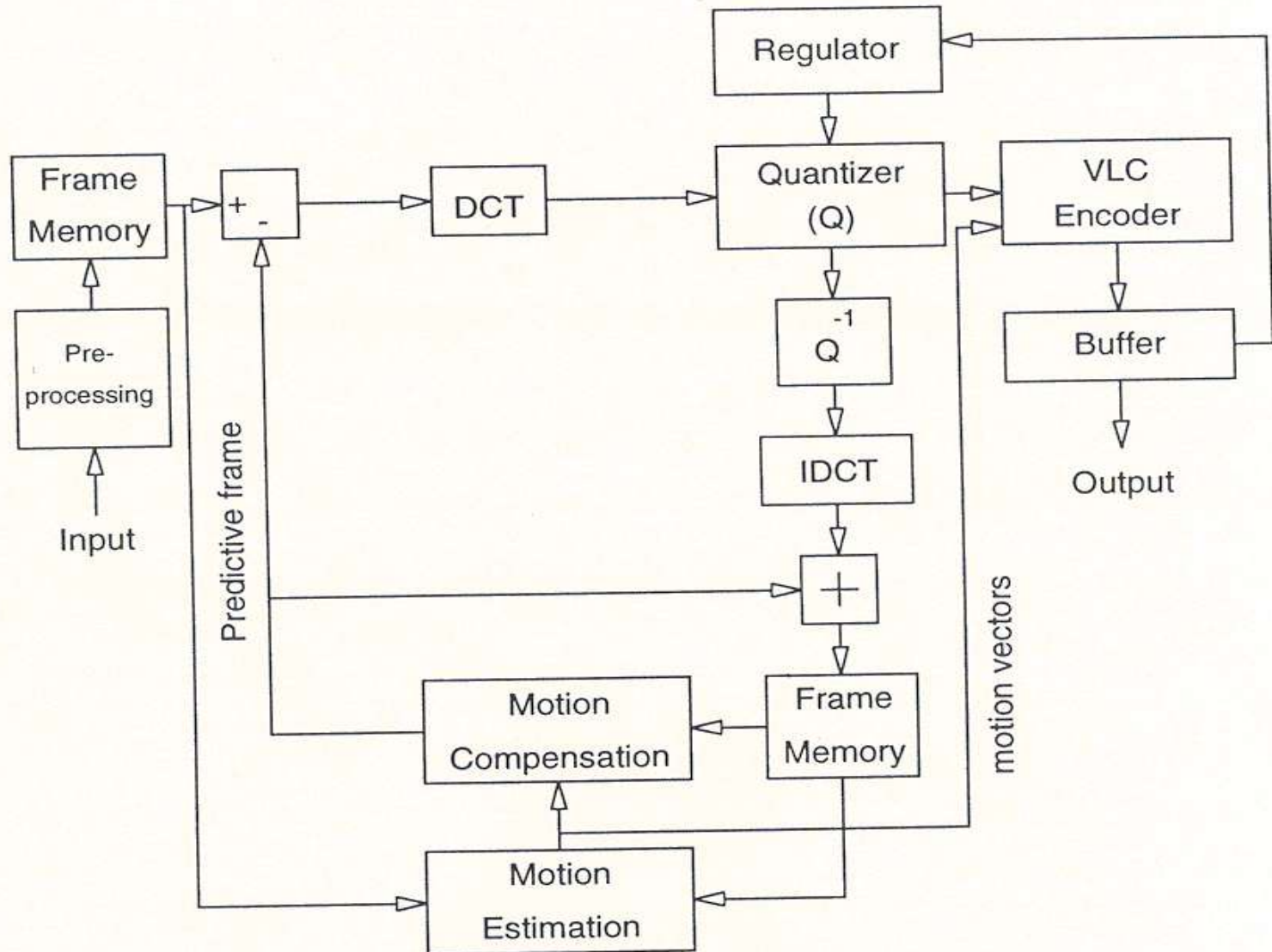
# Bidirectional Motion Compensation



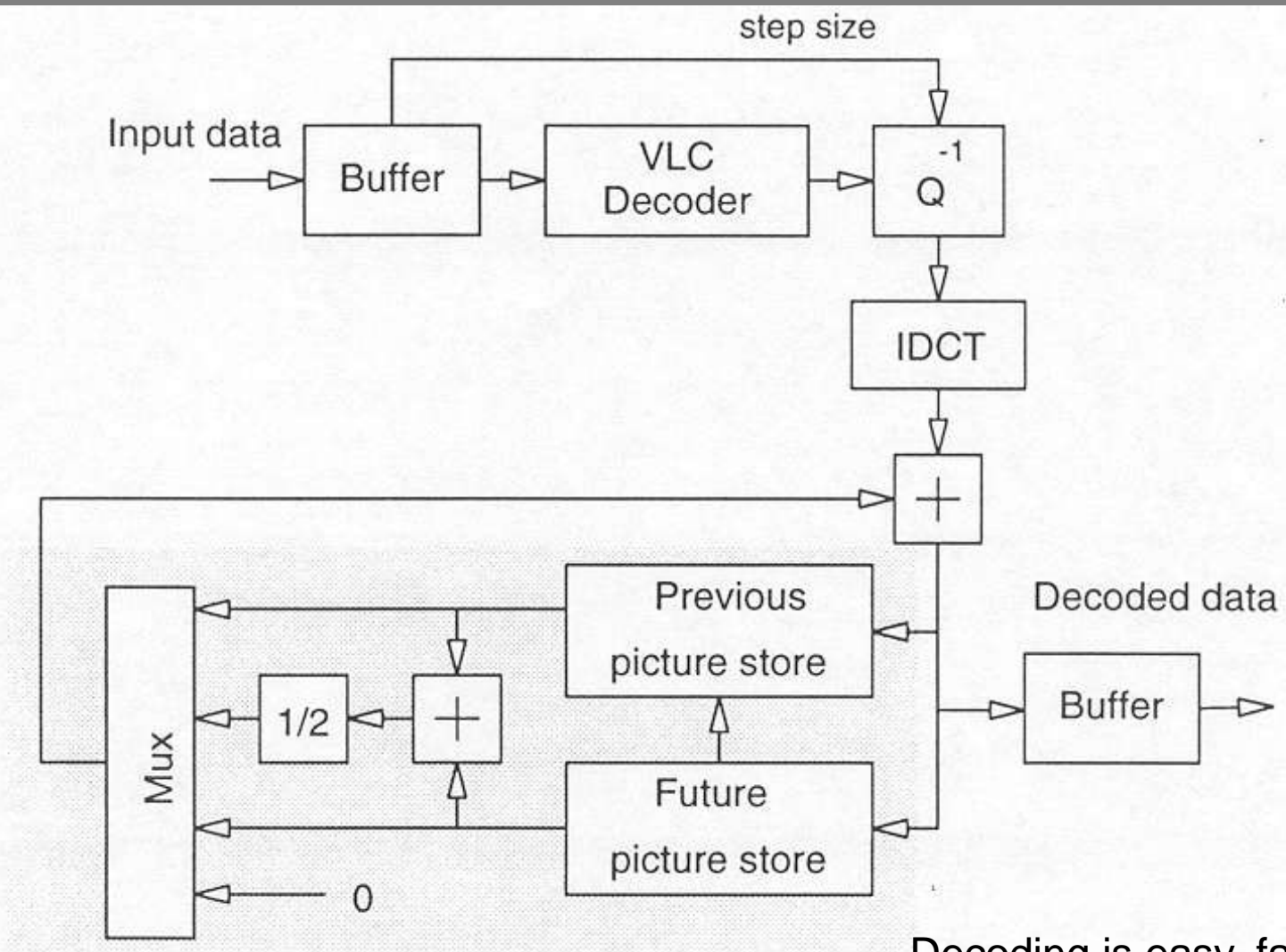
# Syntax Layers in MPEG-1



# MPEG-1 Encoder



# MPEG-1 Decoder



## Motion compensation

Decoding is easy, fast, cheap  
as compared encoding

# Differences from H.261

- Larger gaps between I and P frames, so need to expand motion vector search range.
- To get better encoding, allow motion vectors to be specified to fraction of a pixel (1/2 pixel).
- Bitstream syntax must allow random access, forward/backward play, etc.
- Added notion of *slice* for synchronization after loss/corrupt data.
- B frame macroblocks can specify *two* motion vectors (one to past and one to future), indicating result is to be averaged.

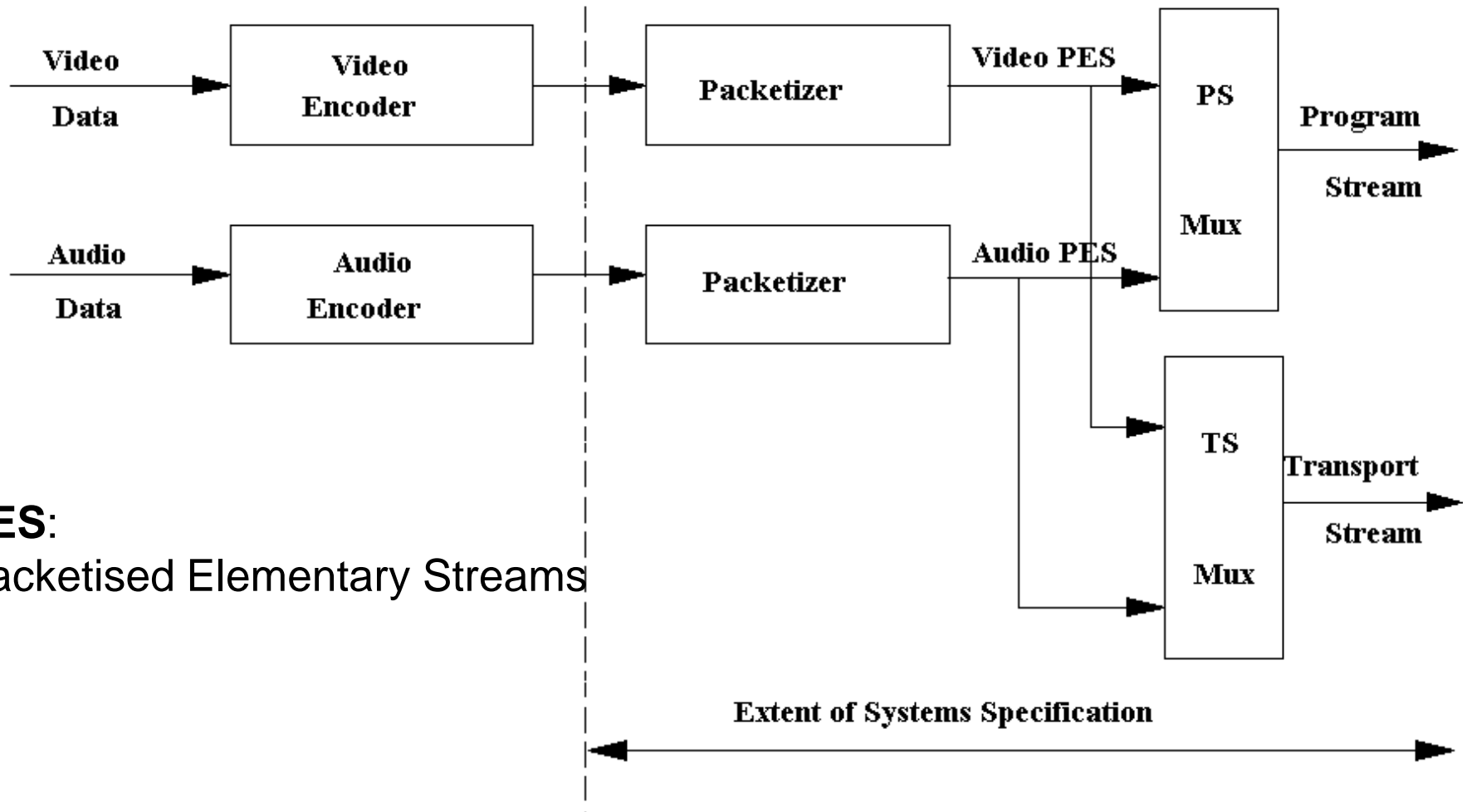


# MPEG-2

- Unlike MPEG-1 which is basically a standard for storing and playing video on a single computer
- MPEG-2 is a standard for digital TV (HDTV and DVD)

Level	Size	Pixels/sec	Bit-rate (Mb/s)	Application
Low	352 x 288 x 30	3 M	4	VHS, TV
Main	720 x 576 x 30	12 M	15	Studio TV
High 1440	1440 x 1152 x 60	96 M	60	Consumer HDTV
High	1920 x 1152 x 60	128 M	80	HDTV, Film

# MPEG-2 System



# New Features in MPEG-2

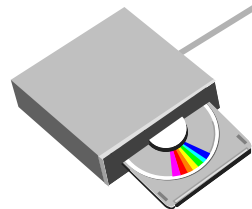
- Support both field prediction and frame prediction.
- Besides 4:2:0, also allow 4:2:2 and 4:4:4 subsampling
- Scalable Coding
  - SNR Scalability -- similar to JPEG Progressive mode, adjusting the quantization steps of the DCT coefficients (image quality)
  - Spatial Scalability -- similar to hierarchical JPEG, multiple spatial resolutions (image size: CIF, SDT to HDTV).
  - Temporal Scalability -- different frame rates (5~60f/s)
- Many minor fixes

# Application Scenarios of MPEG-4

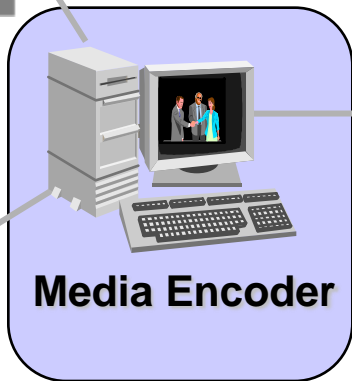
Live Content



Live Feed  
On-demand  
Content



Stored  
Content



Media Encoder



License  
Server



Media  
Services Server  
*Streaming from a  
Media Server  
(or Web Server)*

*Download & Play  
Streaming*



Wired &  
Wireless

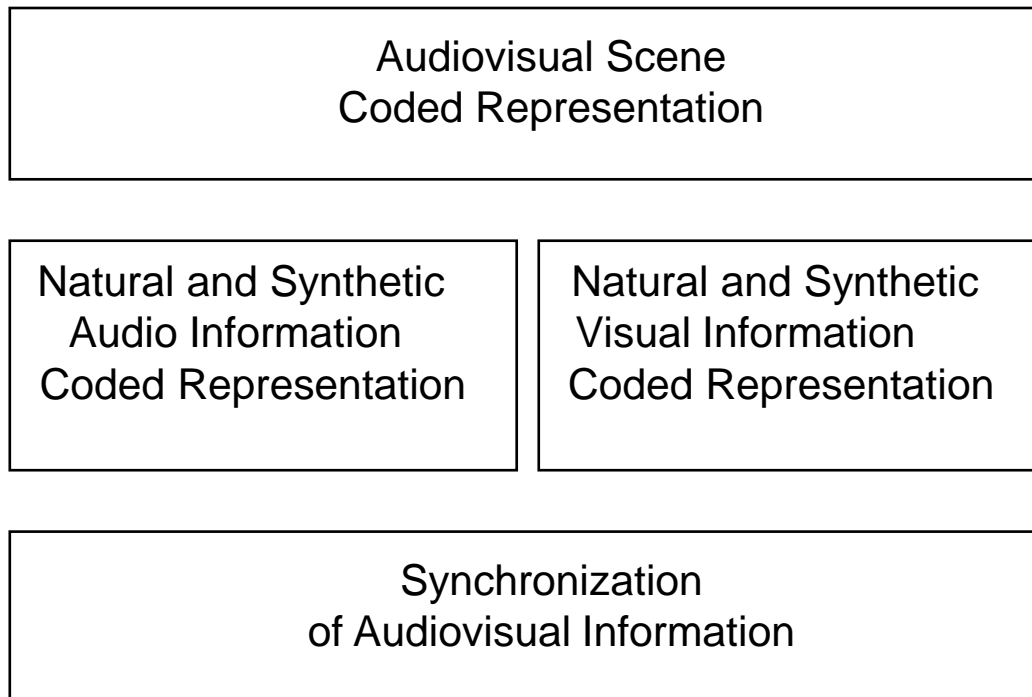


Media Player  
*PC, Hand-held, STB*

Compression → Access → Interaction

# Overview of MPEG-4

- The **coded representation** of the combination of streamed elementary audiovisual information
- 1) **Compression**, 2) **Content-based interactivity**, 3) **Universal access**
- To provide a bridge between the Web and conventional AV media
- To delivery streaming AV media on the Internet and wireless networks



# MPEG-4 Video Coding

Baseline coding	Extended coding
Compression	Object-based Coding
Error Resilience	
Scalability	Still Texture Coding
<i>Conventional coding</i>	<i>Object coding</i>

Natural visual coding for captured pictures

Synthetic visual coding for graphic/animation pictures

Synthetic/Natural Hybrid Coding (SNHC) for the mixed two

# Integration of Natural and Synthetic Contents

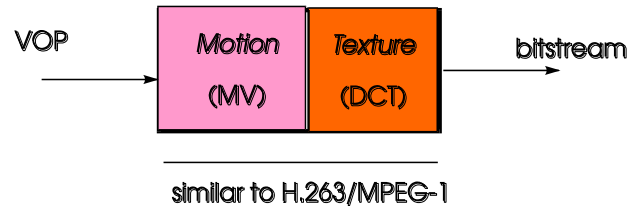


Augmented/Mixed Reality

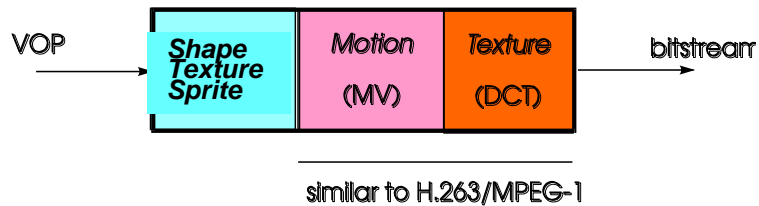
# Baseline and Extended Coding

VOP: Visual Object Plane (MPEG-4 term for a frame)

## MPEG-4 Core Coder



## Extended MPEG-4 Core Coder





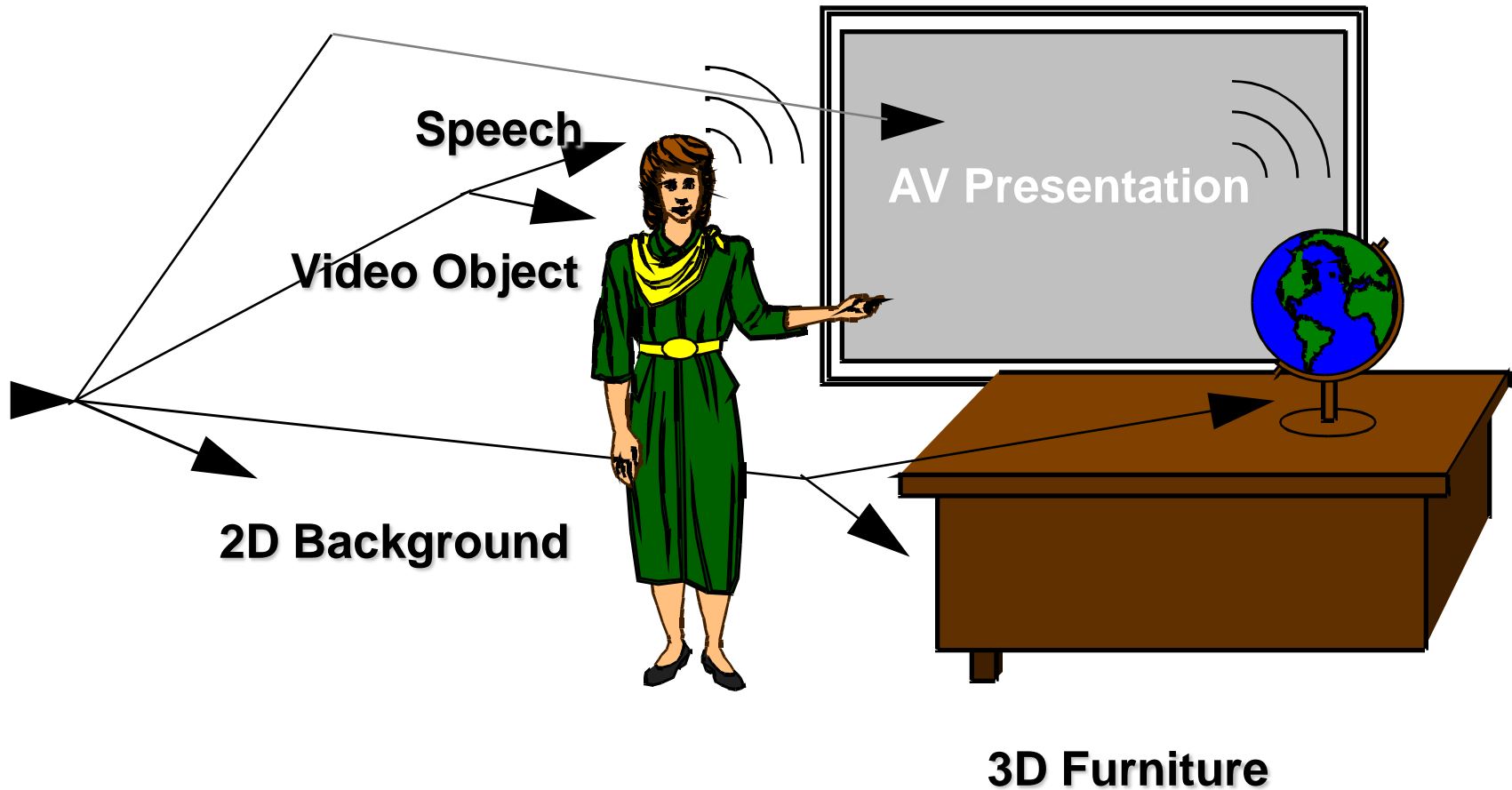
# MPEG-4 Baseline Coding

- ❑ Support both progressive and interlaced scanning
- ❑ Arbitrary size from 8x8 to 2048x2048
- ❑ YCrCb: 4:0:0, 4:2:0, 4:2:2 and 4:4:4
- ❑ Continuously various frame rate
- ❑ Bit rates: 5Kbps ~ 1Gbps from very small TV to Studio TV
  - low (<64Kbps), intermediate (64~484kbps)
  - high (384K~4Mbps) and very high (>4Mbps)
- ❑ MPEG-4 Video is Compatible to Baseline H.263
- ❑ And Almost Compatible to MPEG-1
- ❑ And almost compatible to MPEG-2
- ❑ Better coding efficiency than MPEG-1/2 and H.263

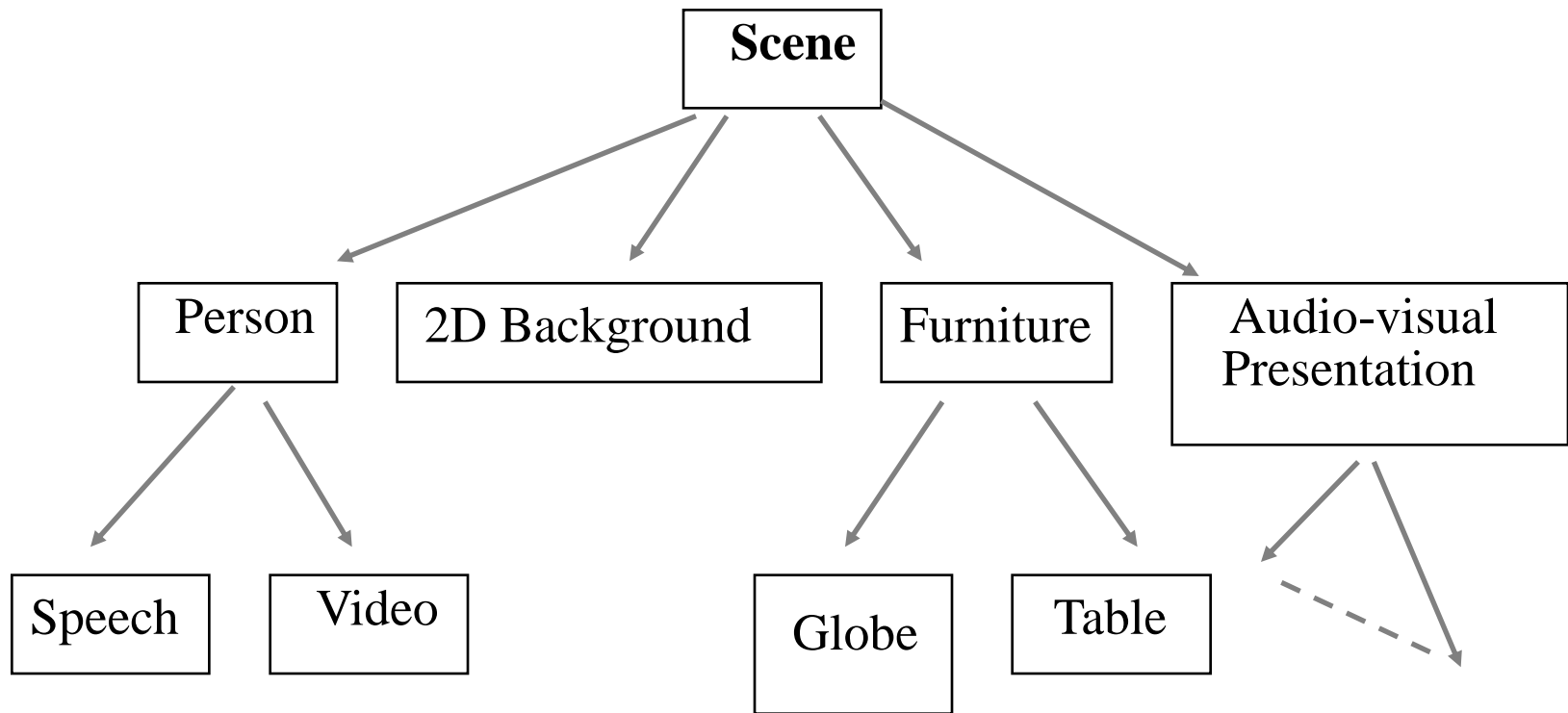
## - Extended Functionalities - *Object-Based Coding of Video*

- Object-Based Coding = Content-Based Coding
- Object-based coding increases compression efficiency
- Object-based coding allows the user to access arbitrarily-shaped objects in a coded scene
- Object-based coding enables high interaction with scene content
- Manipulation of scene content on bitstream level

# Objects in Audio-Visual Scene

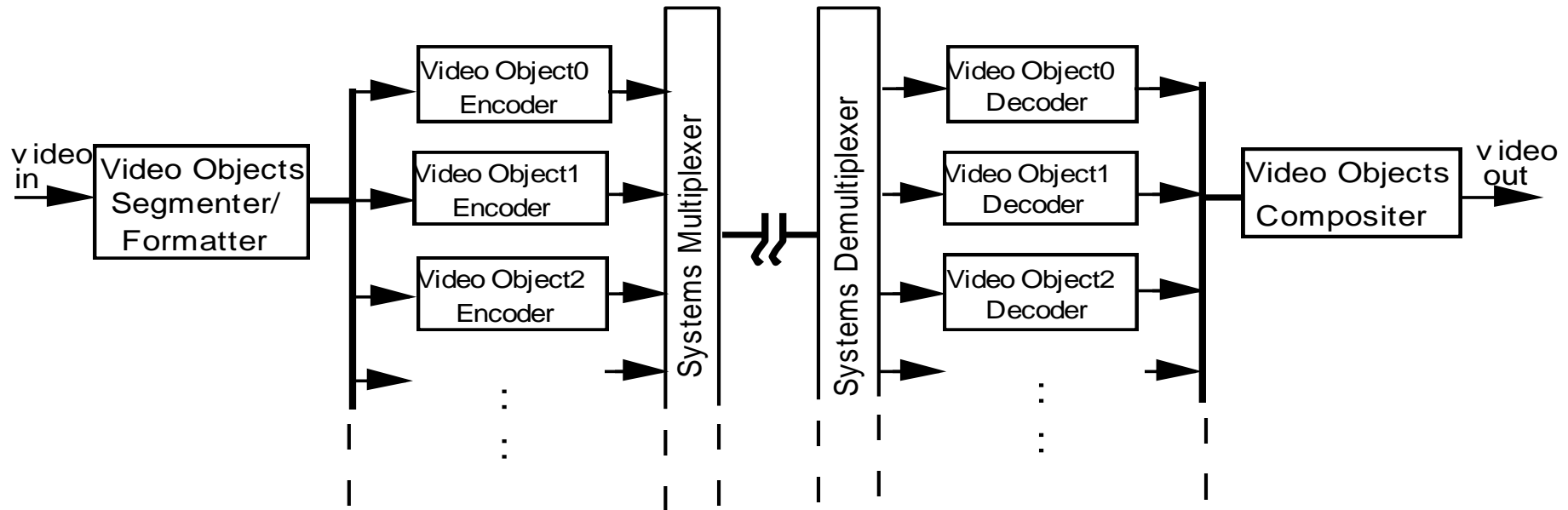


# BIFS – Binary Format for Scene

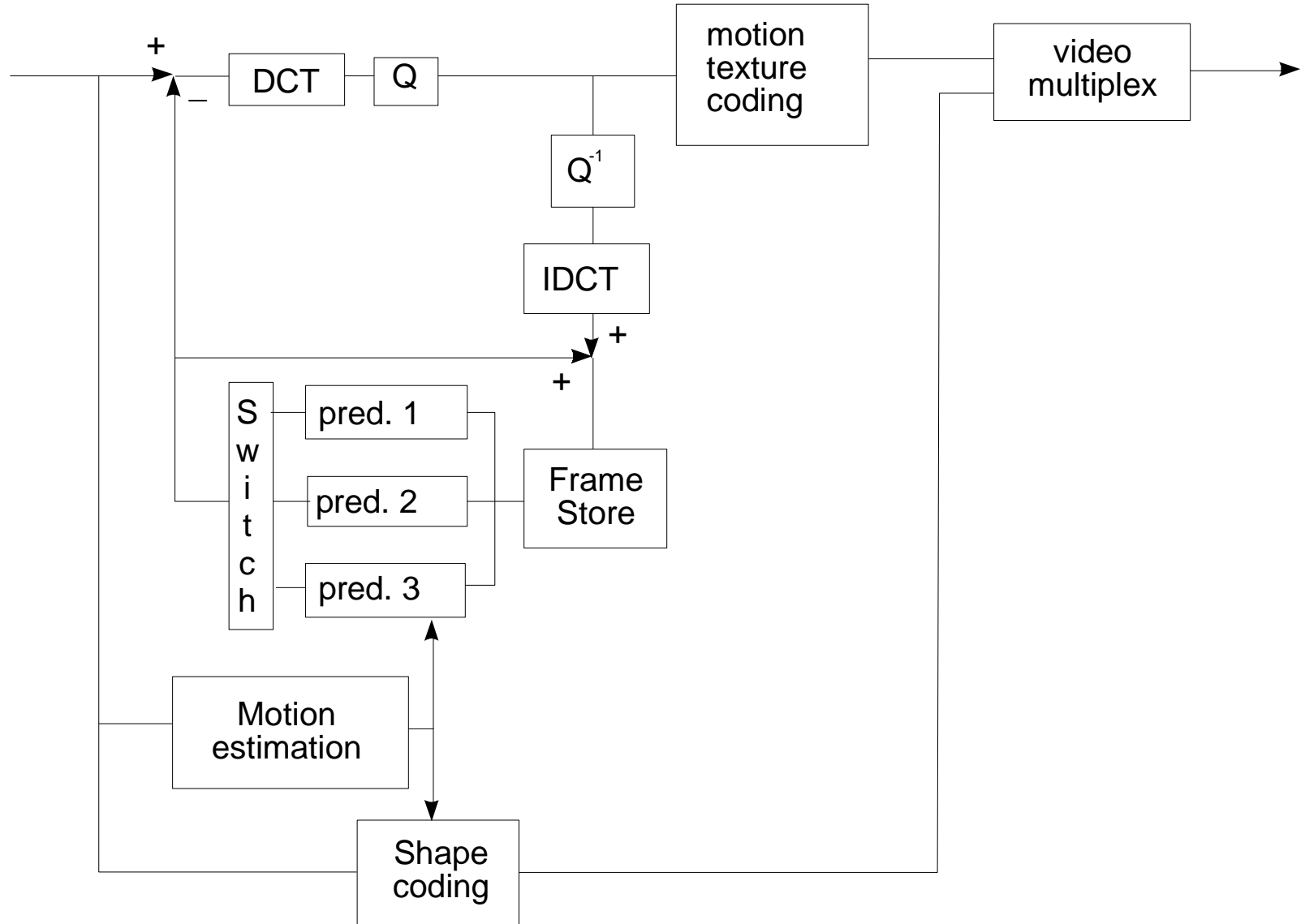


# Object-Based Coding

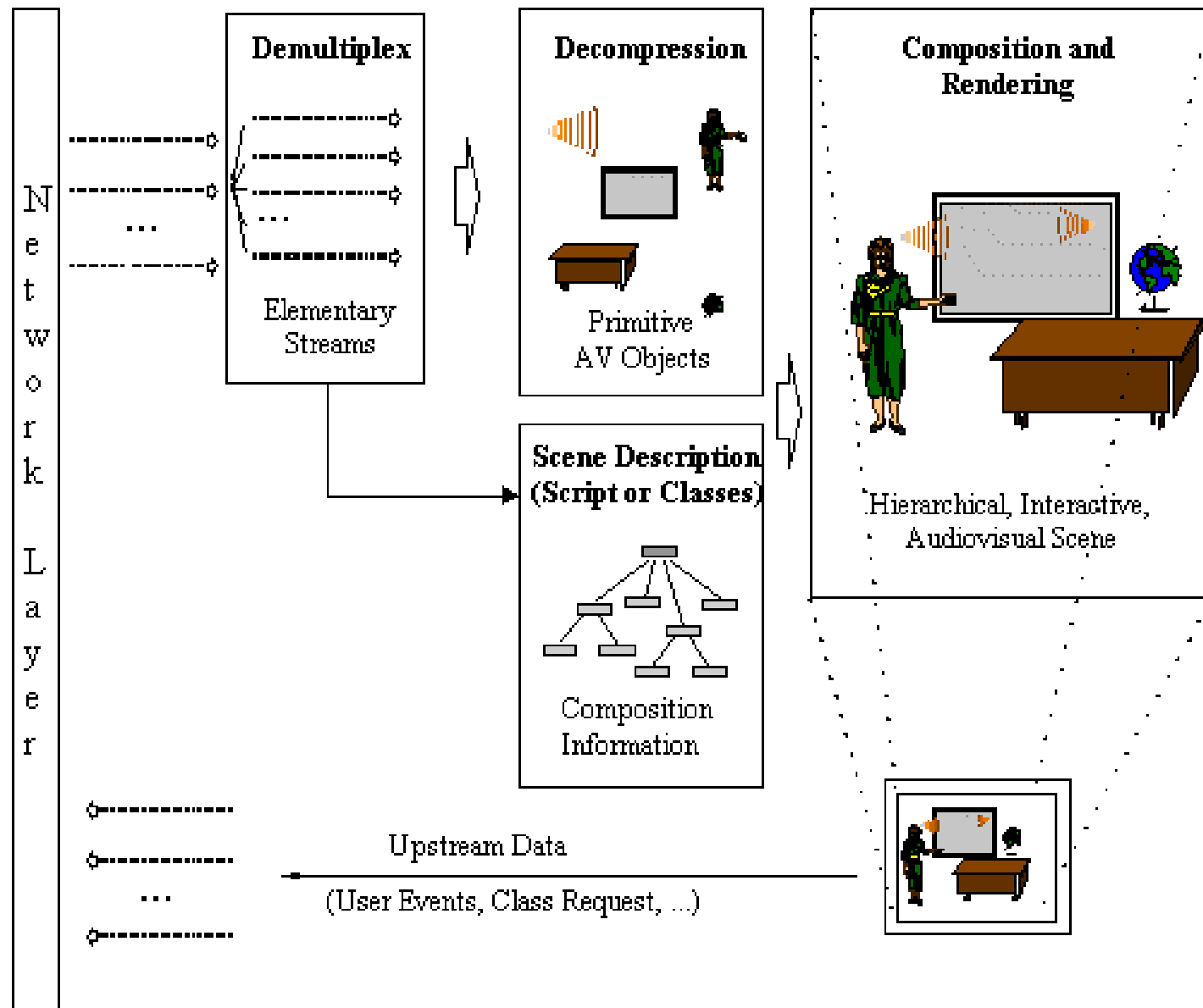
- Each video object in a scene is coded and transmitted separately



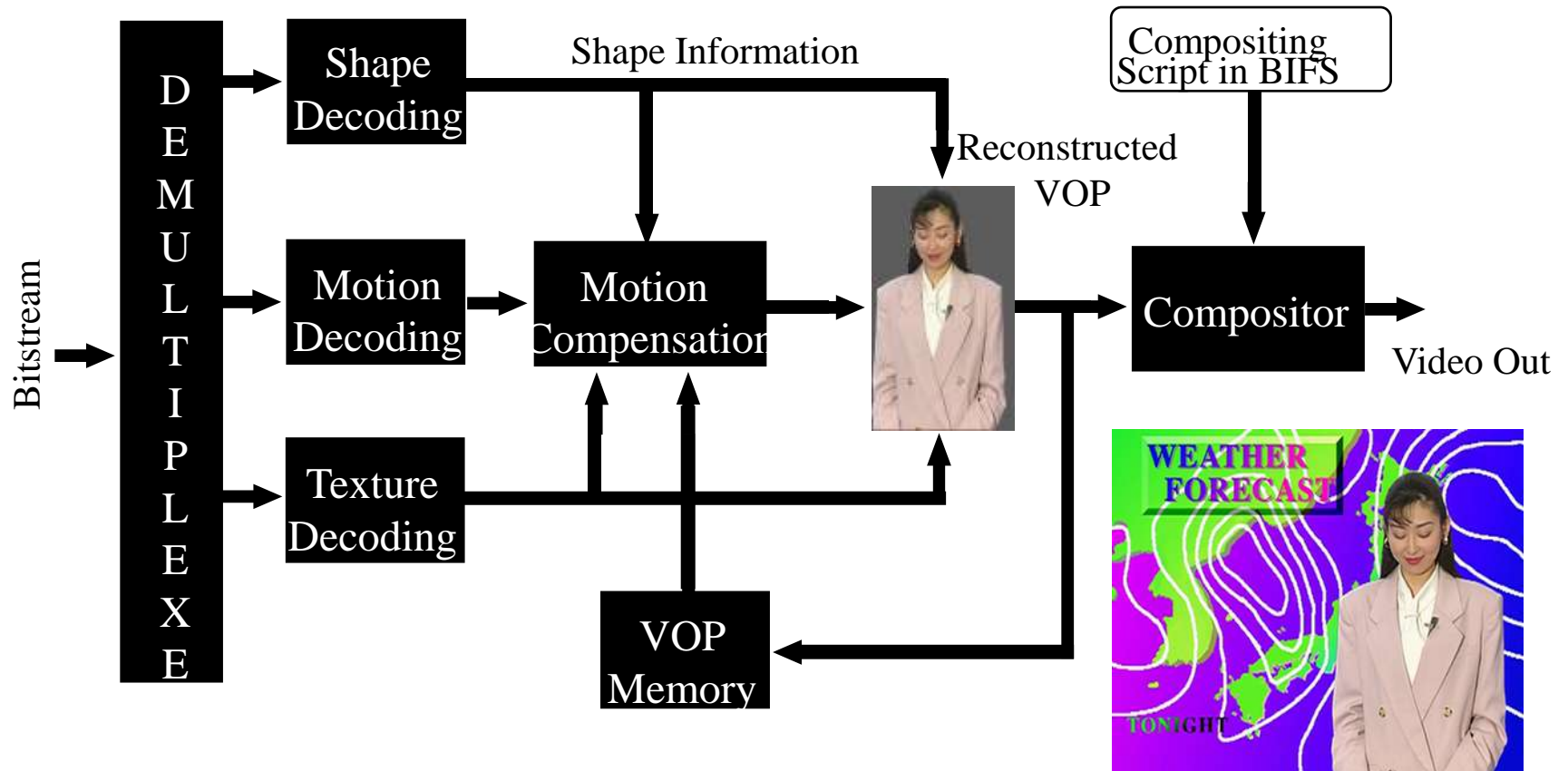
# Object-Based Encoding



# Scene Reconstruction



# Example of Video Decoding





# Sprite Coding

- Original in computer graphics
- Long term background objects
- Real time rotation, translation, zooming



sprite



player



# Various Applications of MPEG-4

- ♦ **IVS**    **Internet Video Streaming**
- ♦ **VA**    **Video Archive**
- ♦ **VCD**    **Video Content Distribution**
- ♦ **IMM**    **Internet Multimedia**
- ♦ **IVG**    **Interactive Video Games**
- ♦ **IPC**    **Interpersonal Communications (videoconferencing, videophone, etc.)**
- ♦ **ISM**    **Interactive Storage Media (optical disks, etc.)**
- ♦ **MMM**    **Multimedia Mailing**
- ♦ **NDB**    **Networked Database Services (via ATM, etc.)**
- ♦ **WMM**    **Wireless Multimedia**

# MPEG-7: What Is It ?

## THE MPEG 7 STANDARD

IS NOT a STANDARD for  
FEATURE  
EXTRACTION/MATCHING

IS NOT a COMPRESSION Standard  
similar to MPEG-1/2/4 or their  
Extension

Content Description of  
Various Audio Visual  
Information

### Types of Audio Visual Information

- Audio, speech
- Moving video, still pictures, graphics
- Information on how objects are combined in scenes

# Why do we need MPEG-7 ?

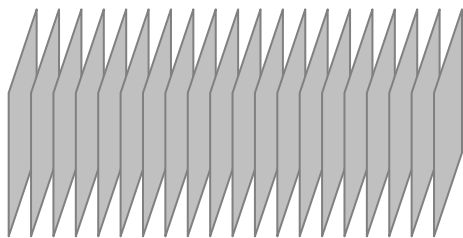


# Main Elements of MPEG-7

- Descriptors (D)
  - syntax and semantics of each feature representation
- Description Schemes (DS)
  - structure and semantics of the relationships between components
- Description Definition Language (DDL)
  - creation of new DS's
  - modification/extension of existing DS's

# Low level Audio and Visual descriptors

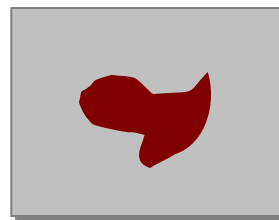
## Video segments



### Contents

- Color
- Camera motion
- Motion activity
- Mosaic

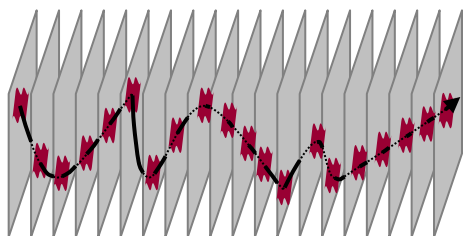
## Still regions



### Contents

- Color
- Shape
- Position
- Texture

## Moving regions



### Contents

- Color
- Motion trajectory
- Parametric motion
- Spatio-temporal shape

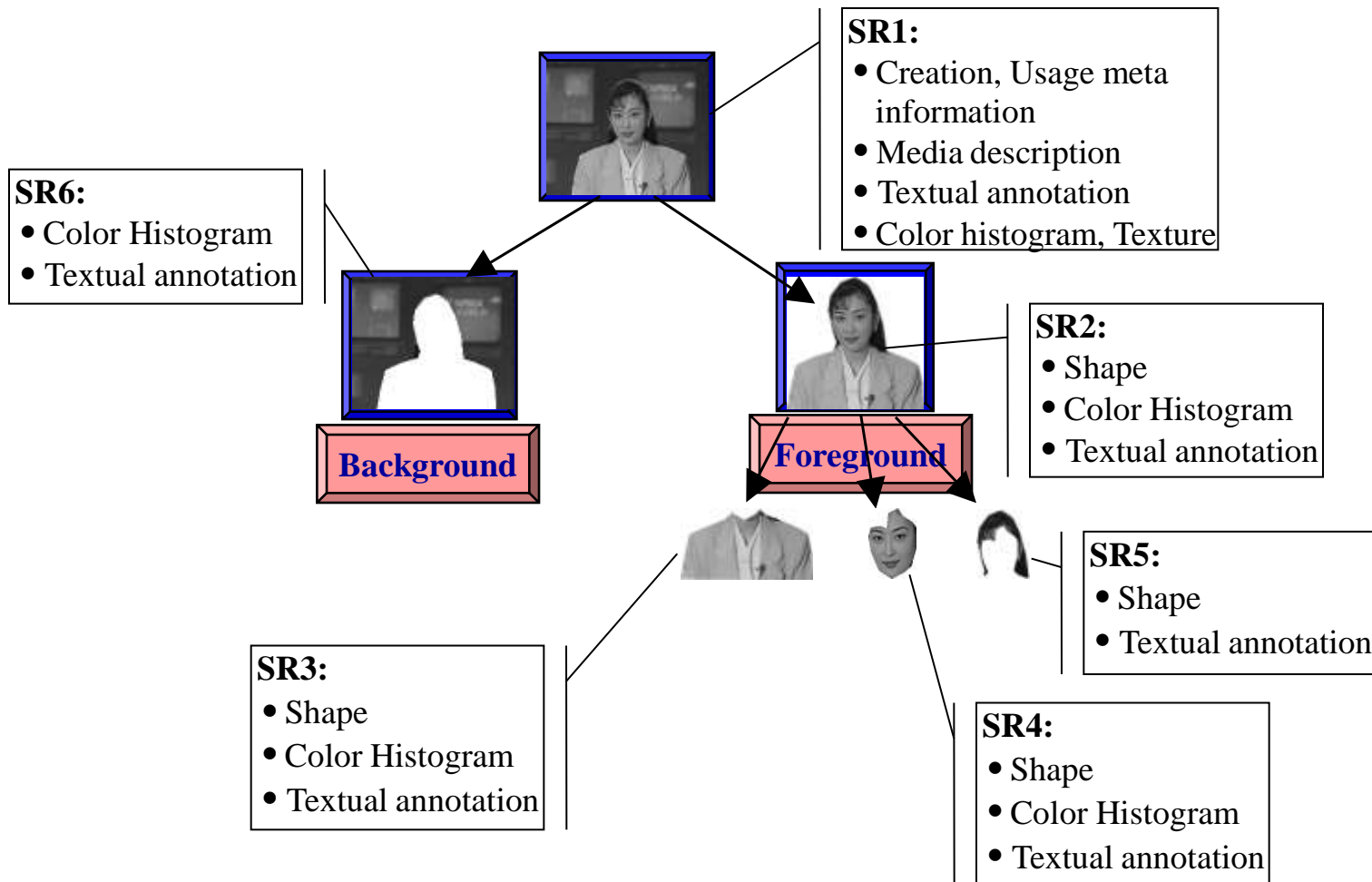
## Audio segments



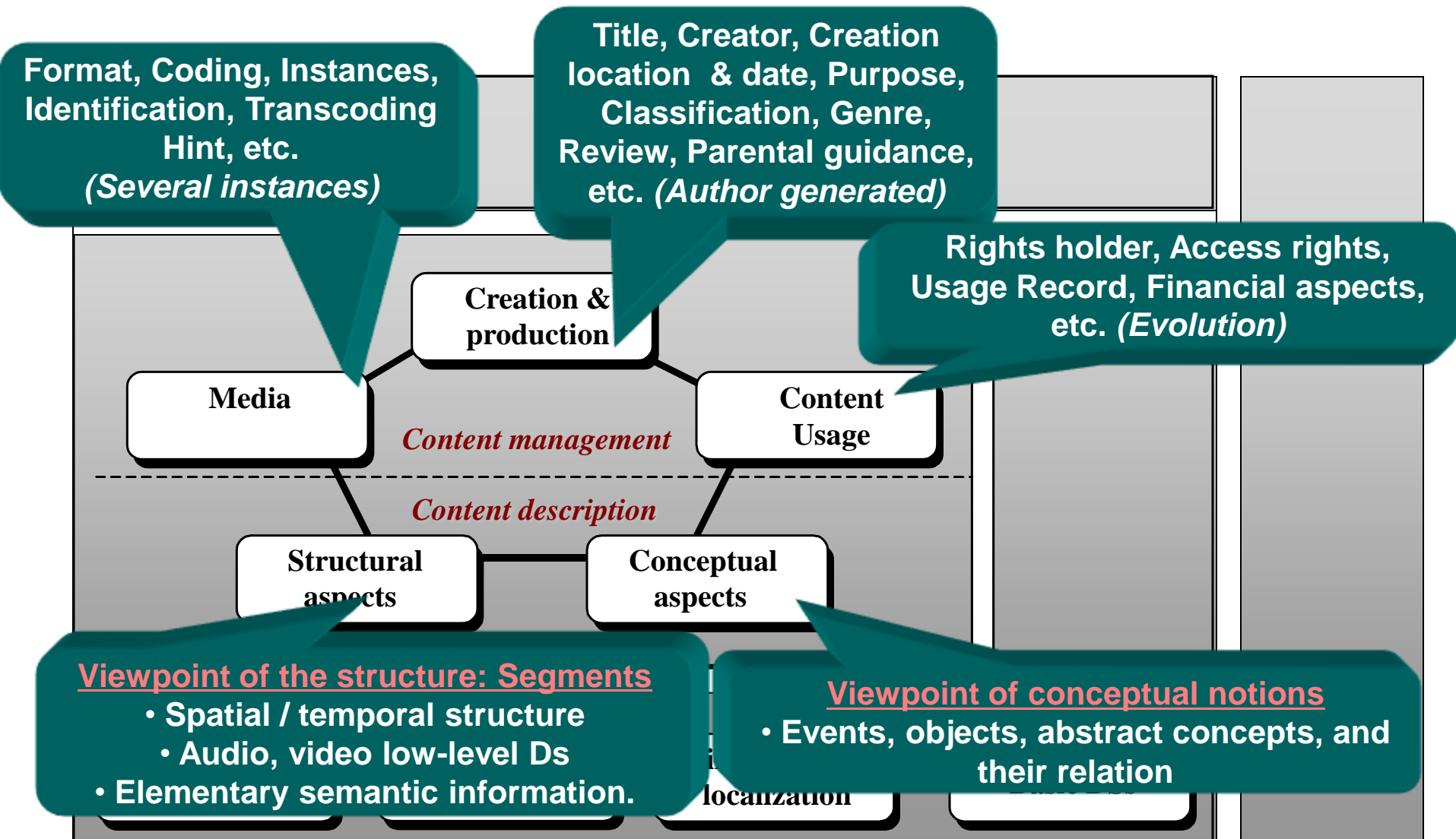
### Contents

- Spoken content
- Spectral characterization
- Music: timbre, melody

# Low Level Descriptors and Segment Trees



# Content Management and Description





Time  
Axis

### Segment Tree

Shot1 Shot2 Shot3

Segment 1

Sub-segment 1

Sub-segment 2

Sub-segment 3

Sub-segment 4

segment 2

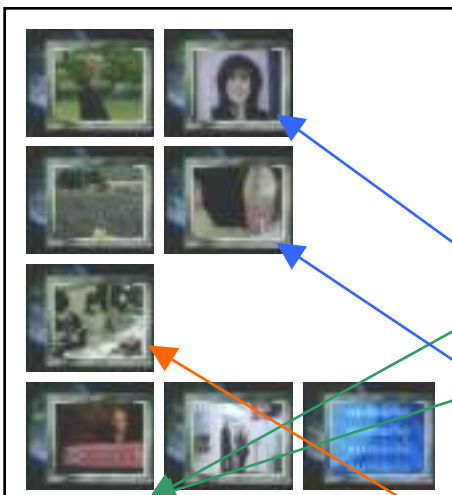
Segment 3

Segment 4

Segment 5

Segment 6

Segment 7



### Semantic DS (Events)

• Introduction

• Summary

• Program logo

• Studio

• Overview

• News Presenter

• News Items

• International

• Clinton Case

• Pope in Cuba

• National

• Twins

• Sports

• Closing

# MPEG-21

- Seeks to describe a multimedia framework and set out a vision for the future of an environment that is capable of supporting the delivery and use of all content types by different categories of users in multiple application domains
- Financial, content, consumer, technology, delivery applications
- MPEG-21 digital item – A structured digital object with a standard representation, identification and metadata with this framework. This entity is also the fundamental unit for distribution and transaction within this framework.
  - Digital Item Declaration
  - Digital Item Representation
  - Digital Item Identification and Description
  - Digital Item Management and Usage
  - Intellectual Property Management and Protection
  - Terminals and Networks
  - Event Reporting

# Demos of Video Coding

# Review of Advanced Coding

- JPEG2000
- H.264
- MPEG-21

# What is JPEG 2000?

- JPEG 2000 is a wavelet-based image-compression standard, developed by the same ISO committee that previously developed JPEG, although with a different group of participants and contributors.
- JPEG 2000 was conceived as a next generation image compression standard that would improve on the performance of JPEG while, more significantly, adding features and capabilities not available with Baseline JPEG compression.

# Why use JPEG 2000?

- Open Standard
  - Royalty free
- One master supports multiple derivatives
  - One file for both lossless and lossy data
  - Progressive display and scalable rendering
  - One algorithm for both lossless and lossy compression
- Region-of-Interest (ROI) on coding and access
- Easily handles large images
  - Multiple components and high bit-depth images
- Generous metadata support

# JPEG 2000 Standard - Parts 1-6

Part 1: Core  
Coding System

Part 3: Motion  
JPEG2000

Part 4: Conformance  
Testing

Part 6: Compound  
Image File Format

Part 2: Extensions

Part 5: Reference  
Software



# JPEG 2000 Standard - Parts 8-13

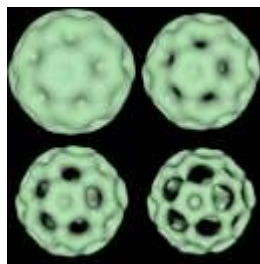
Part 8: JPSEC  
Secure JPEG2000



Part 9: JPIP  
Interactivity Tools



Part 10: JP3D  
3D & Floating Pt



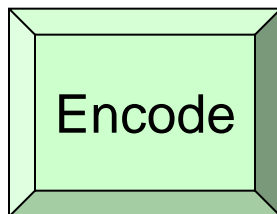
Part 11: JPWL  
Wireless



Part 12: ISO  
Media File Format



Part 13: Entry-Level  
JPEG2000 Encoder



Key

Under  
Development

Published



# One Master → Multiple Derivatives

- A single JPEG2000 master can serve multiple uses
  - Scale by resolution
    - Thumbnail image
    - Screen resolution image
    - Print quality image
  - Scale by quality
    - Lossless → Lossy
    - Preset file size
- Key enabling technologies
  - Wavelet transform
  - Embedded block coding

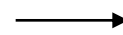
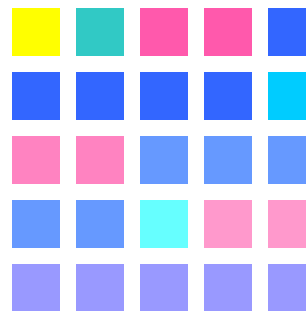
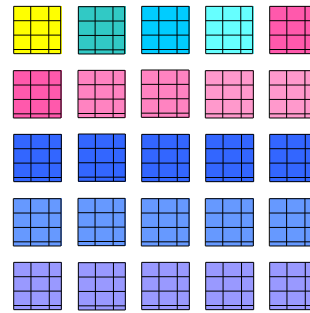
# One Master → Multiple Derivatives



Packets  
reordered  
by layer

Master Image

Derived Image



Region of  
Interest  
Selected

# JPEG 2000 File Format Family



- JP2 (JPEG 2000 Core, Part 1)
  - Single image, continuous codestream
- JPX (JPEG 2000 Extensions, Part 2)
  - Multiple codestreams, possibly fragmented

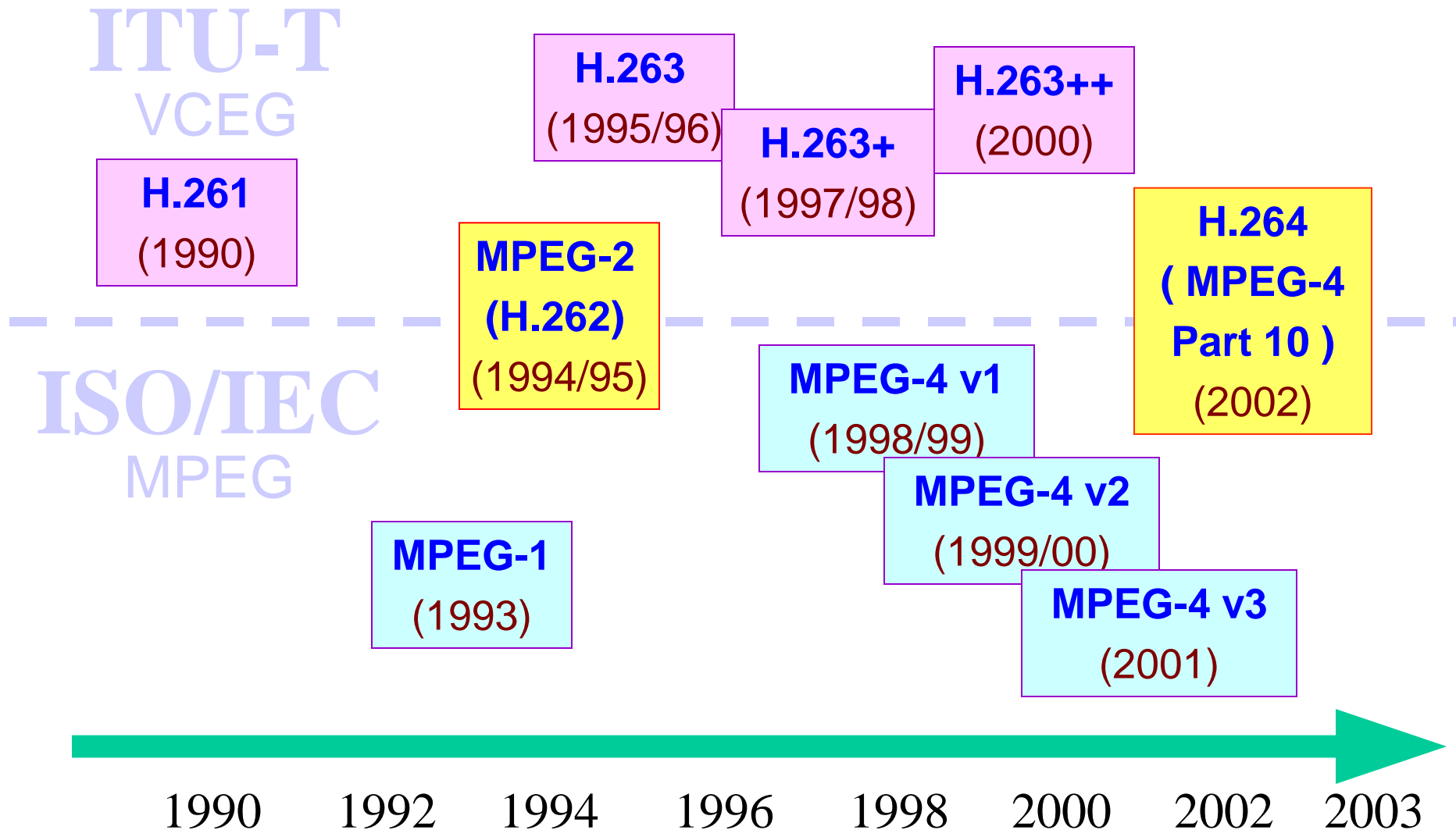


- MJ2 (Motion JPEG 2000, Part 3)
  - Timed sequence of JPEG 2000 images
  - Intra-frame coding only

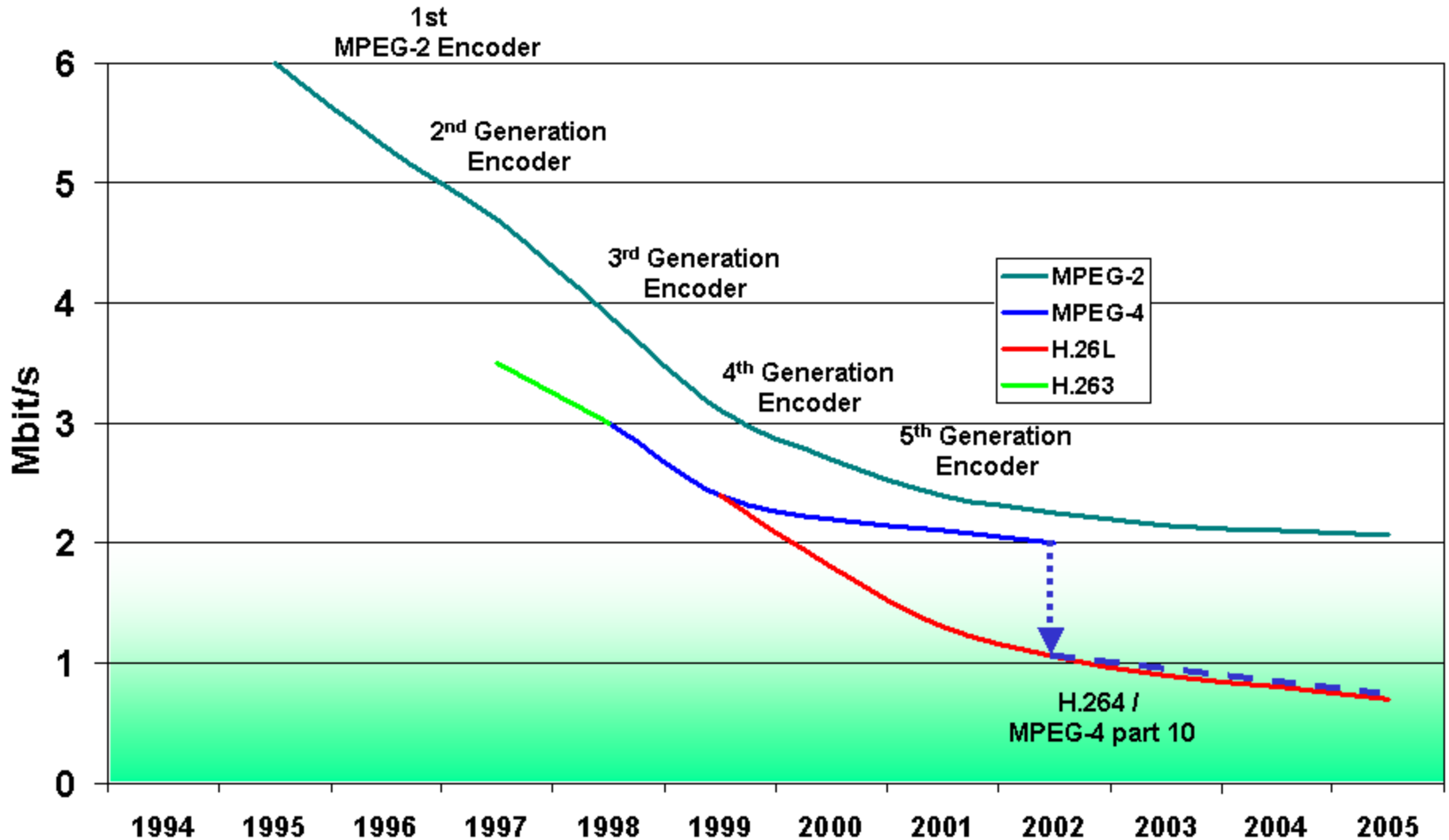


- JPM (JPEG 2000 Multi-Layer, Part 6)
  - MRC model for compound document images
  - Multiple images (binary and contone) and pages

# Chronological Table of Video Coding Standards



# Position of H.264

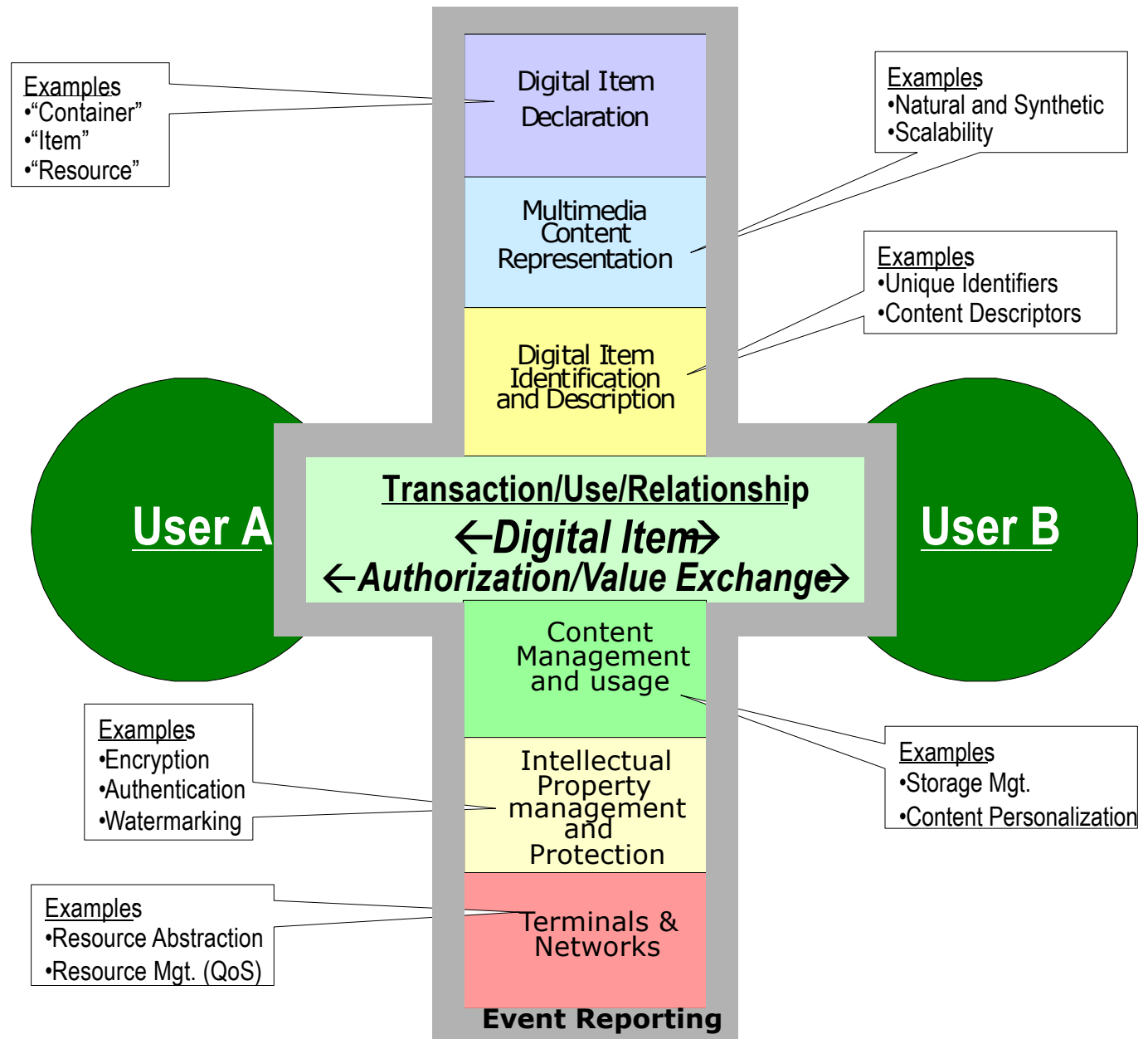


# MPEG-21: putting it all together

From the MPEG-21 Proposed Draft Technical Report:

- “ Many elements exist to build an infrastructure for the delivery and consumption of multimedia content. There is, however, no 'big picture' to describe how the specification of these elements, either in existence or under development, relate to each other. The aim of MPEG-21 is:
  - 1) to understand if and how these various components **fit together** and
  - 2) to discuss which **new standards** may be required, if gaps in the infrastructure exist and, once the above two points have been reached,
  - 3) to actually accomplish the **integration** of different standards. ”
- In MPEG-21, all Users have **Rights** and **Interests**
  - And they all need to be able to express those

# Pictorial overview of MPEG-21



# Demos of JPEG2000



# Media Object Production

## - Hardware and Software Tools

- Concept of Media Object Production
- Process of CM Media Object Production
- Audio Production
- Video Production
  - Capturing
  - Editing
  - Compressing
  - Outputting
- Demos of Live Audio/Video Capture

# Media Object

## Media Object

component in a multimedia document, presentation, etc.

- Text
- 2D graphics
- 3D graphics
- Animation
- Still image
- Audio clip
  - \* speech
  - \* music
  - \* other sound
- Video clip



# Media Production

**Media production:** process to produce a medium object

## ❑ Text

- Language, font, size, color, shadow, blink, etc
- Tools: LaTeX, Word, HTML editors, ...

## ❑ 2D/3D Graphics

- Programming languages: Java2/3D, OpenGL, SVG, ...
- Tools: TrueSpace, LightWave3D, Inspire3D, ...

## ❑ Animation

- Programming language: Java, Java script, VRML, ...
- Tools: Infini-D, Flash, TrueSpace, ...

## ❑ Still Image & Moving Images (Video)

- Captured via scanner, camera, software, ...
- Tools: xv, Display, PhotoDraw, Photoshop, ...

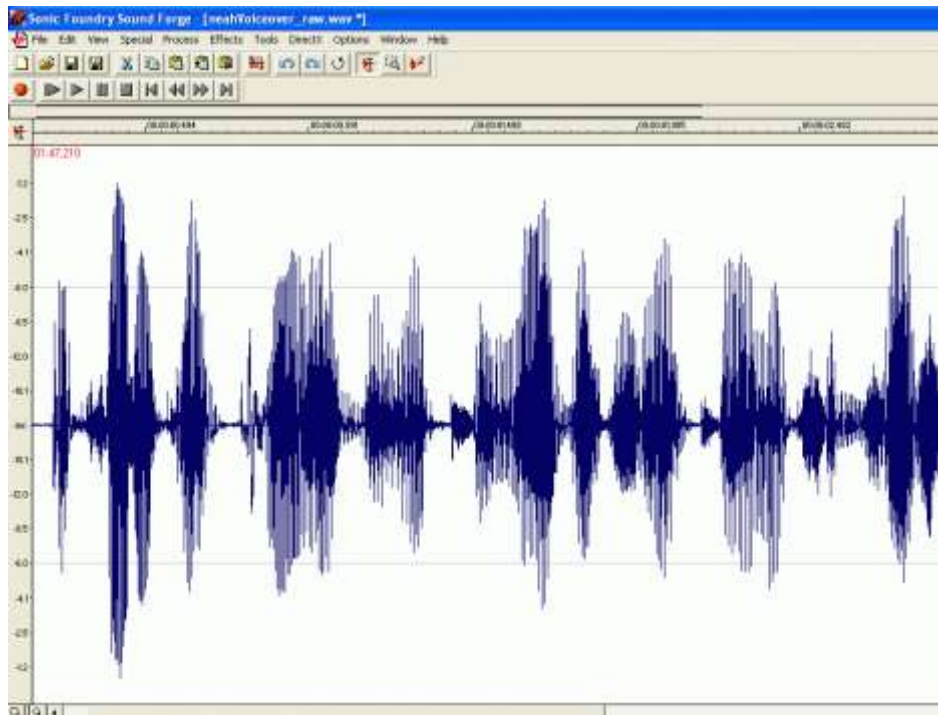
❑ **Audio** – Continuous Media (CM) → special techniques for its production

❑ **Video** – Continuous Media (CM) → special techniques for its production

# General Features of CM Production

## ❖ Features of CM

- Change with time: samples/sec (audio) and frame/sec (video)
- Large volume of data: proportional to the length
- Realtime processing power



# General Process of CM Production

## Production Process

### ☐ Pre-Production

- Clarify intended application of CM to be produced
- Prepare hardware: mic, camera, CPU power, memory/disc size, board
- Determine OS: Unix/Linux, Windows 2000/XP/VISTA/7, and Mac
- Purchase, download, install necessary software

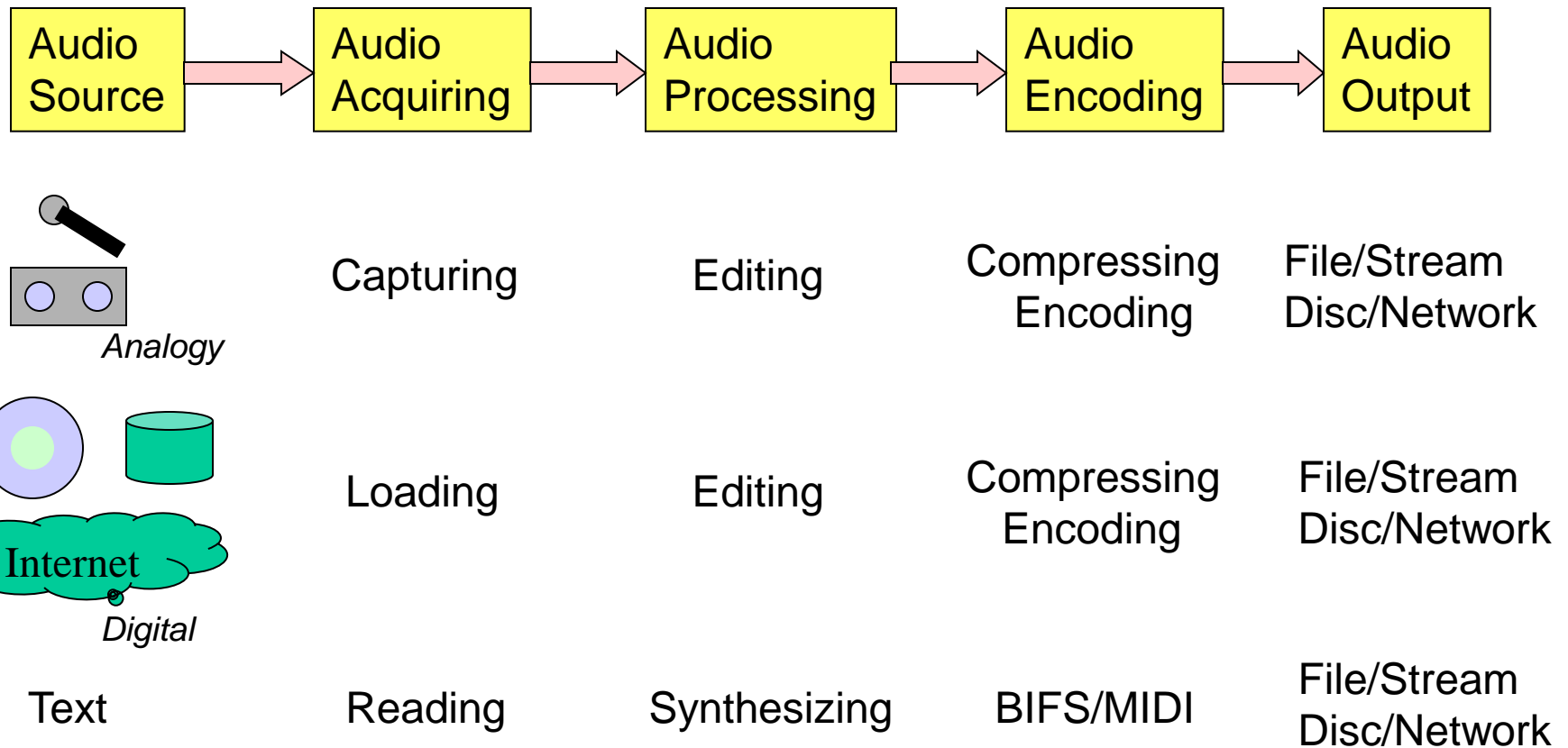
### ☐ In-Production

- CM acquiring, processing/editing, encoding and output

### ☐ Post-Production

- Testing and refining when necessary

# Audio Production Process



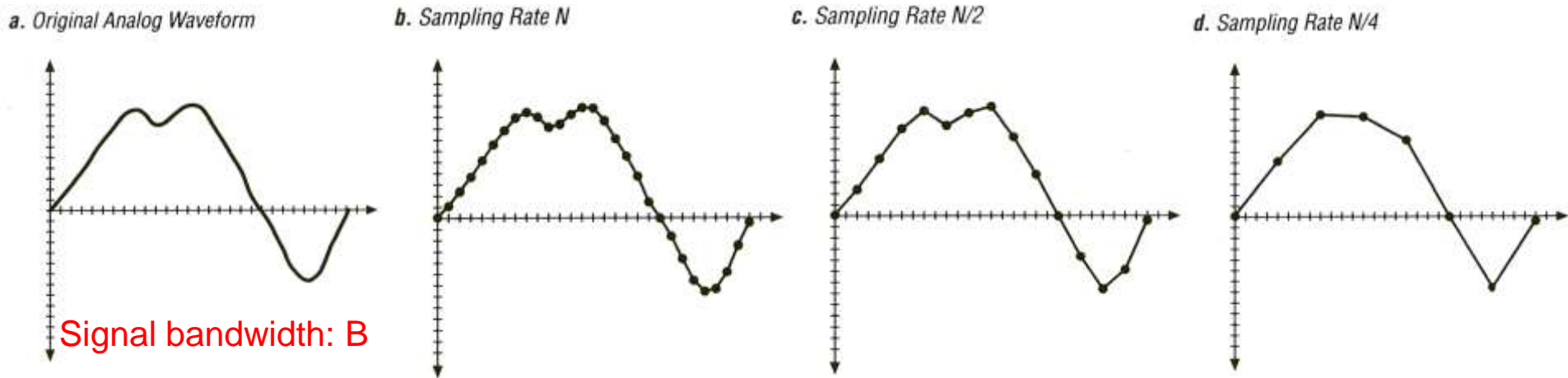
# Audio Pre-Production

- Basic Notice
  - Good source: good microphone, mixing desk
  - Signal processing: compressor, EQ unit
  - Proper recording environment



# Audio Digital Samples and Sampling Rate

Sampling Rate/Frequency: number of samples per second



More samples, better quality but larger data

Nyquist sampling rate:  $N \Rightarrow 2B \rightarrow$  No Distortion



# Audio Resolution and Quantization Levels

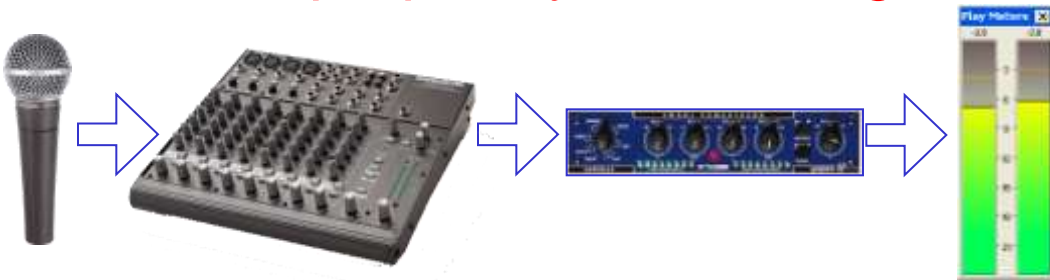
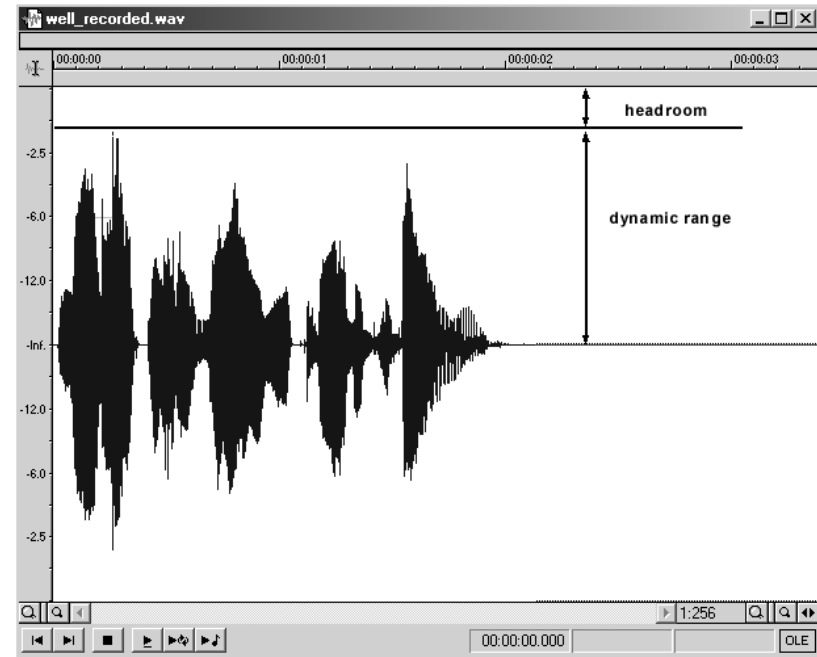
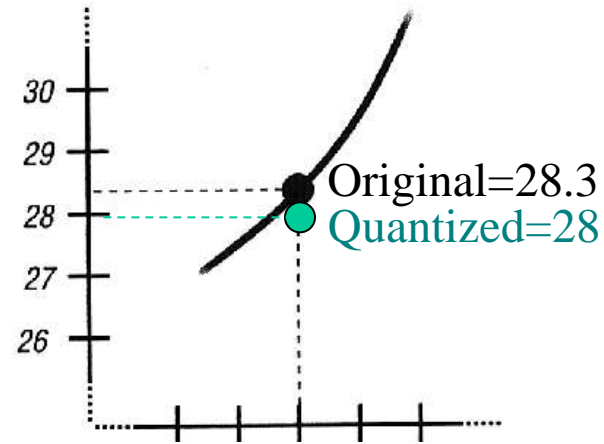
Samples are quantized into discrete values

Sample Resolution:

→ all possible values or bits per sample

- 256 values from 8 bits
- 65536 values from 16 bits

Setup “gain” carefully to  
have a proper dynamic range



# Audio Frequency Features

<b><i>FQ Range</i></b>	<b><i>Contents</i></b>
20–60 Hz	Extreme low bass. Most speakers cannot reproduce this.
60–250 Hz	The audible low-end. Files with the right amount of low end sound <i>warm</i> , files without enough sound <i>thin</i> .
250 Hz–2 kHz	The low-midrange. Files with too much in the low-mids are hard to listen to and sound telephone-like.
2 kHz–4 kHz	The high-midrange. Where most speech information resides. In fact, cutting here in the music and boosting around 3 kHz in your narration makes it more intelligible.
4 kHz–6 kHz	The presence range. Provides clarity in both voice and musical instruments. Boosting 5 kHz can make your music or voiceover (not both!) seem closer to the listener.
6 kHz–20 kHz	The very high frequencies. Boosting here adds “air” but can also cause sibilance problems

# General Rules in Audio Capturing

## ❑ Audio quality

- Target application: Disc, network no-live or live broadcast
- Set input level correctly
- Save as sound file format with no or small quality loss: au, wav, aif, ...
- Choose certain sampling rate and resolution (8/12/16 bits)

The higher rate and bits, the high quality but the more data

Data amount = Channels x SamplingRate x Bits / 8 (Bytes)

## ❑ Capture interface and sound card

- Microphone input and line input
- Many different sound cards available

## ❑ OS and software

- Audio capture methods are similar for different OSs
- Capture software embed in OS: Windows Media
- Capture software bounded with sound card
- Capture API in programming languages like Java

# Audio Editing

- Cropping: Select a piece/clip of audio from an audio file
- Cutting, copying and pasting
- Equalization
- Normalization
- Noise reduction
- Transition
  - Trimming silence
  - Fade
- Popular sound editing software
  - Sound Forge from Sonic Foundry
  - Cool Edit from Syntrillium
  - SoundEdit from Macromedia

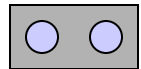
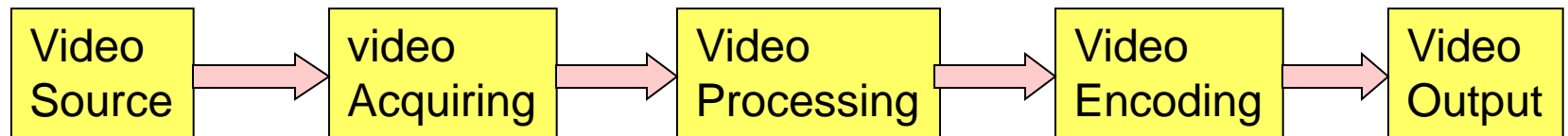
# Audio Compressing/Encoding

- Audio editing is usually based on uncompressed audio: au, wav, aif
- Compression: reduce data size  
based on sound types and targeted applications  
Make a balance between quality and size
  - channels (channel converting)
  - sampling rate (re-sampling)
  - bit resolution
- Encoded audio file formats
  - wave (compressed), QuickTime (mov), MP3, GSM, DCR, ...
- Encoded audio stream formats
  - RealAudio (ra, ram), Windows (asf), QuickTime (mov), MP3
- Music vs. speech codecs
  - Music and speech are fundamentally different
  - Codecs have been optimized for one or the other
  - When in doubt, use a music codec

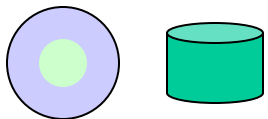
# Audio Output

- Audio output destinations
  - speaker
  - analogy storage device
  - digital storage device
  - network/Internet
- Network down loaded applications
  - stored in disc in compressed file format
- Network real time applications
  - outputted to a streaming server in stream format

# Video Production



*Analogy/digital*



*Digital*

Model &  
parameters

Capturing  
compressing

Loading

Reading

Editing

Editing

Rendering

Compressing  
Encoding

Compressing  
Encoding

models

File/Stream  
Disc/Network

File/Stream  
Disc/Network

File/Stream  
Disc/Network

# Video Pre-Production

- Basic Notice
  - Good camera: DV is most cost-efficient
  - Buy the best tripod you can afford!
  - Lighting: 3-point lighting kit

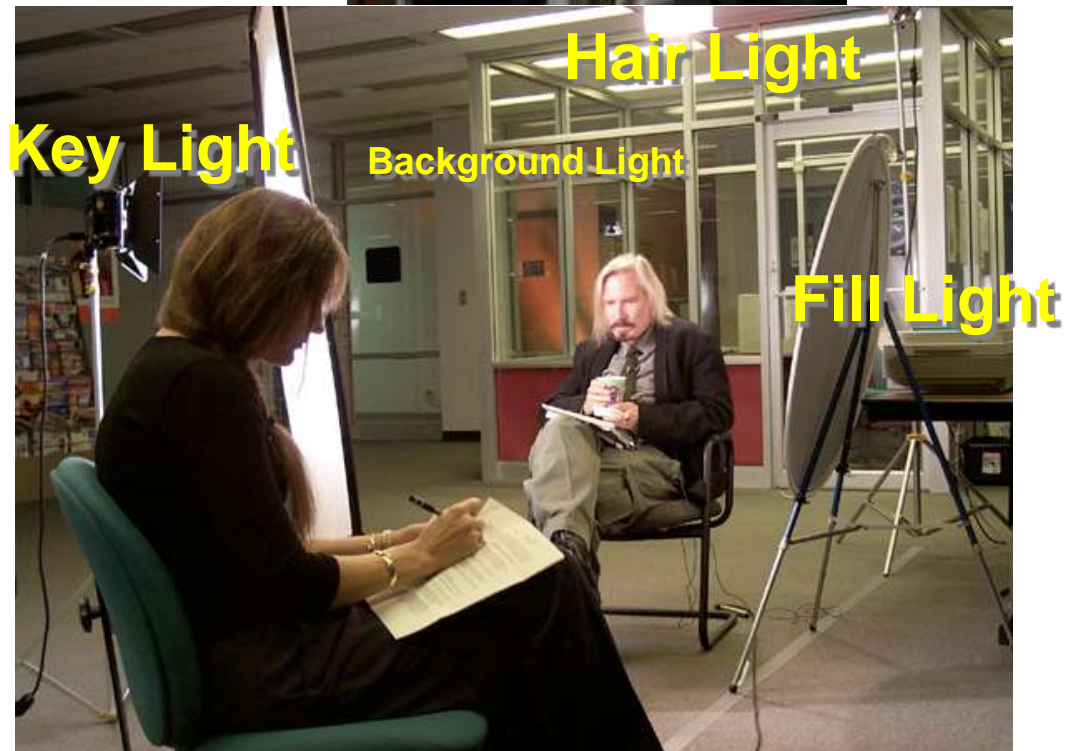
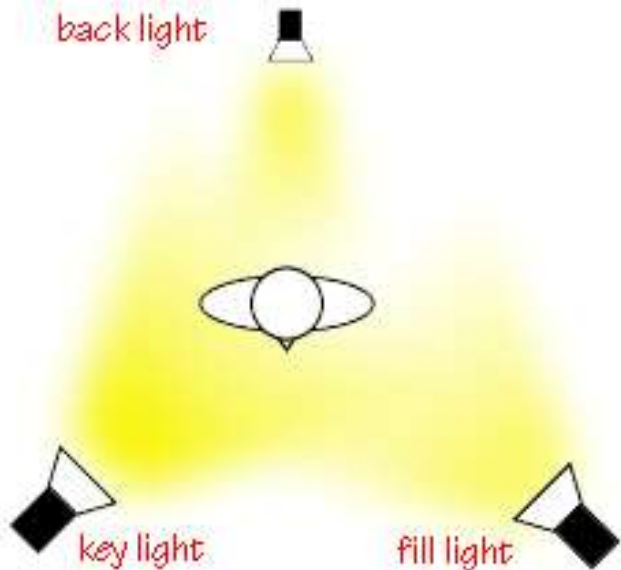




# Video Pre-Production - Lights

## Three Point Lighting

- Key Light
- Fill Light
- Back Light or Hair Light



# Video Capturing

## ❑ Video quality

- Target application: Disc, network no-live or live broadcast
- Select relative large image size: 640x480, 320x240
- Choose frame rate (fps) and true color (24 bits)
- Save as video file format with no or small quality loss: avi, mjpeg, ...
- Leave enough memory and HD space

Data amount

= width x height x fps x bits x time / 8 (Bytes, no-compressing)

= The\_above / compress\_ratio (with compressing)

## ❑ Capture interface and video capture card

- Analogy interface, digital IEEE1394 (i.Link), USB
- Video capture card: interface type and with/without hardware encoder
  - # GGV-VCP2M/PCI (software encoding)
  - # MEG-VC2
  - # IFC-IL3/DV
  - # EZDV II
  - # DVStorm-RT Light

# Video Editing - Traditional “A/B Roll” Analog Editing System





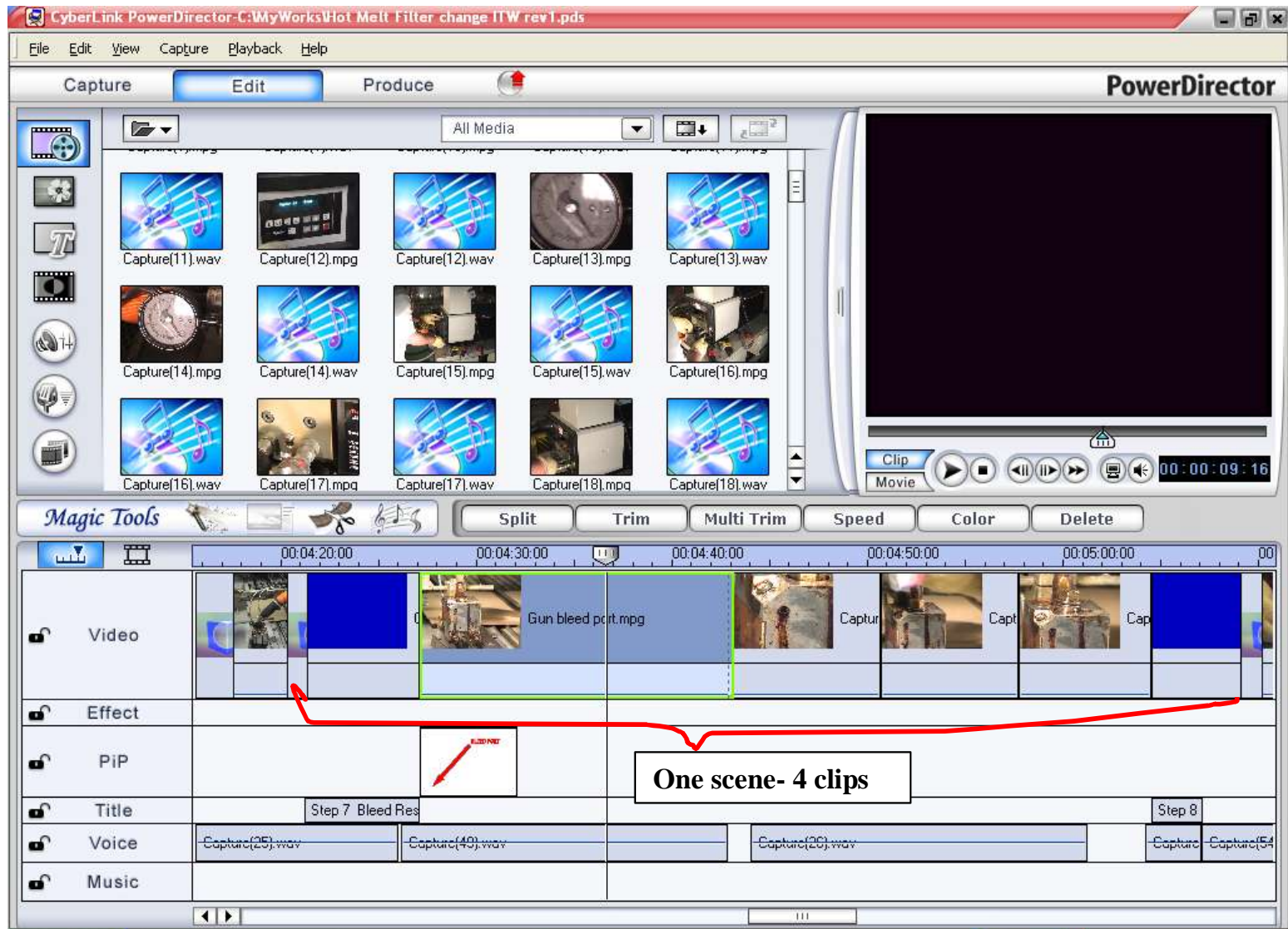
# Video Editing – Modern Digital Edit using Computer Software



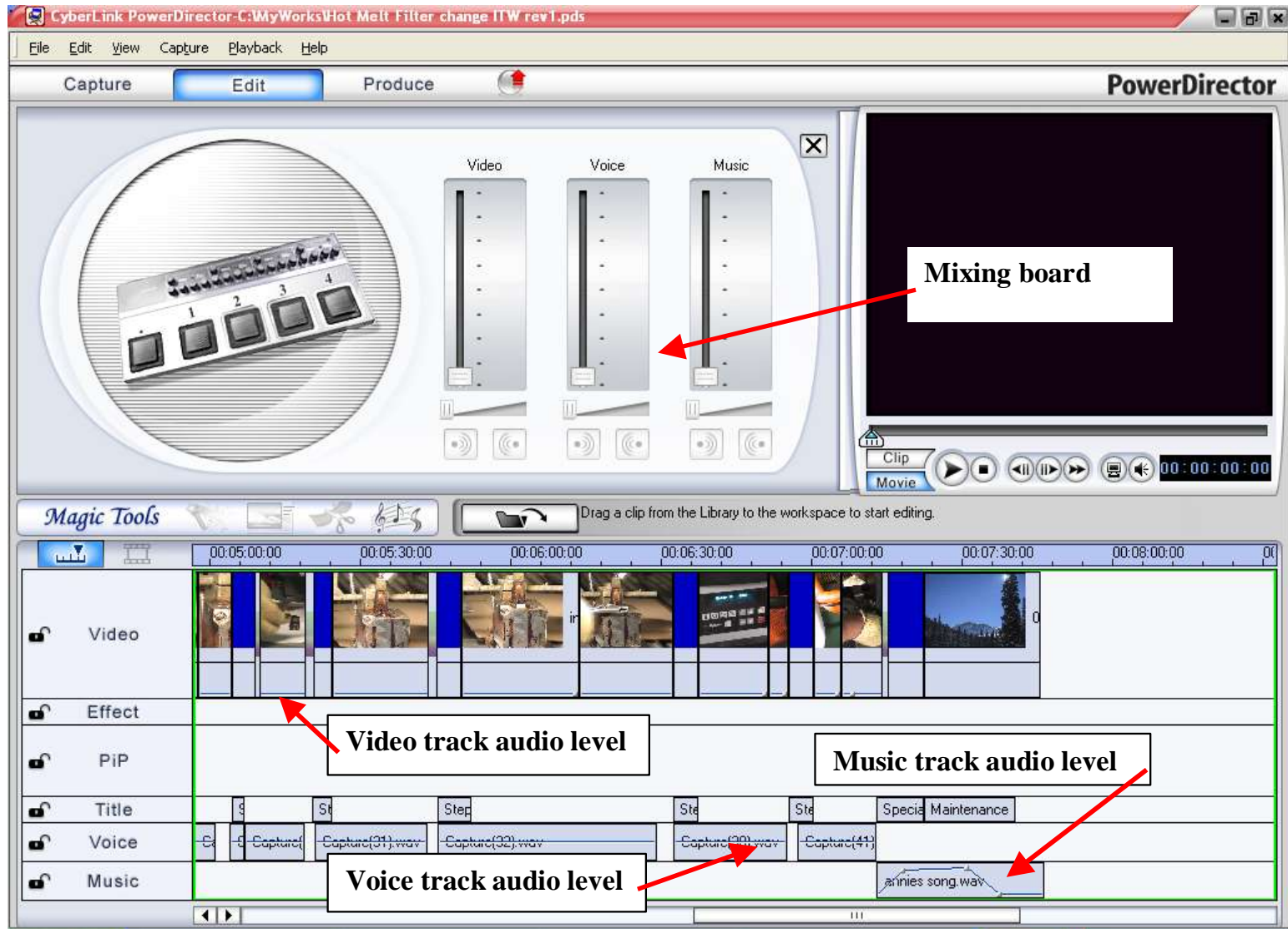
# Video Editing

- Specify the location of a video frame, called time code
- **SMPTE** (Society for Motion Picture Television Engineers, pron. "simptee")  
hours:minutes:seconds:frames  
00:04:26:05 → an image at the 4<sup>th</sup> minute, 26<sup>th</sup> second and 5<sup>th</sup> frame
- Nonlinear editing: arrange a video sequence in arbitrary order
- Special effects
  - Transition between two clips: fading, wiping, scrolling, ...
  - Superimposing: superimpose one clip over another
  - Filtering: lens flare, zoom, twist, pan, ...
  - Morphing: one image cross-fades into another
- Popular video editing software
  - Adobe's Premiere
  - Ulead's Video Studio
  - MediaStudio

# An Example of Video Editing



# An Example of Video and Audio Mixing





# Video Compressing/Encoding

- Compression: further reduce data size for particular applications  
Make a balance between quality and size
  - Reducing image size (640x480, 320x240, 240x180, 176x132, ...)
  - Reducing frame rates (15, 10, 5, ...)
  - Coding with high compression ratio
- Encoded video file formats
  - H.263/264, MPEG-2, MPEG-4, QuickTime (mov), WAM, ...
- Encoded video stream formats
  - RealVideo (ram), Windows (asf), QuickTime (mov)



# Video Output

- Video output destinations
  - TV
  - analogy storage device
  - digital storage device
  - network/Internet
- Network down loaded applications
  - stored in disc in compressed file format
- Network real time applications
  - outputted to a streaming server in stream format

# Demos of Live Audio/Video Capture

- ☐ Audio capture using Sound Recorder in Windows
- ☐ Video capture using Creative WebCam Plus
- ☐ Windows Media Player
- ☐ Windows Media Encoder

# Media Integration & Presentation

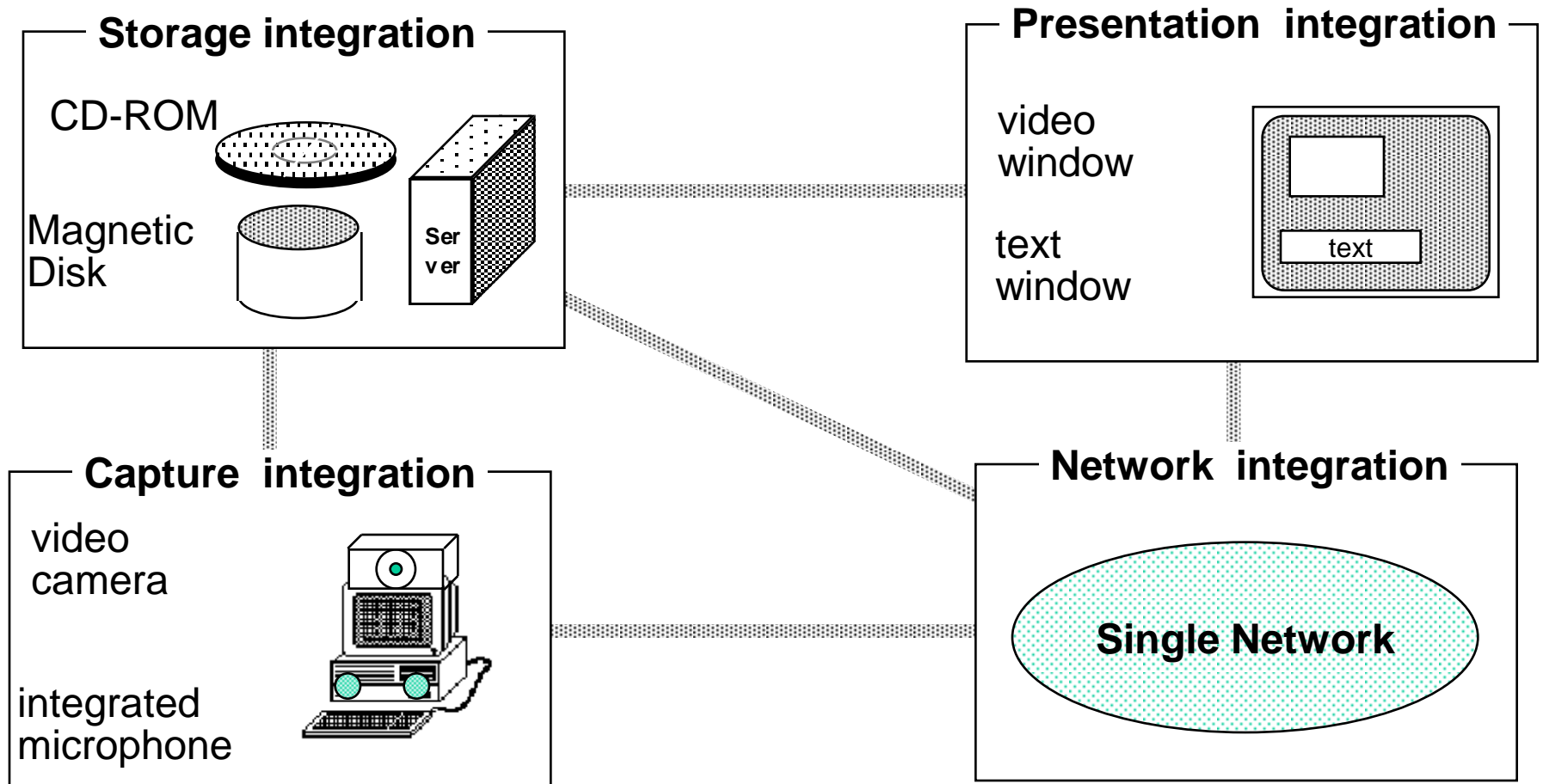
## - Languages and Tools

- Media Integration Concept
- Media Synchronization and QoS
- Media Integration in Multimedia Presentation
- Media Integration Languages
- Media Integration Authoring Tools
- SMIL (Synchronous Multimedia Integration Language)
- HTML+TIME (Timed Interactive Multimedia Extension)
- VRML (Virtual Reality Modeling Language)

# Media Integration Concept & Catalog

## ❑ Media integration

- Integrate different media into a system/application/file



# Media Integration Concept & Catalog

## ❑ Media integration

- Core issues due to shared resources: CPU, memory, network, etc.

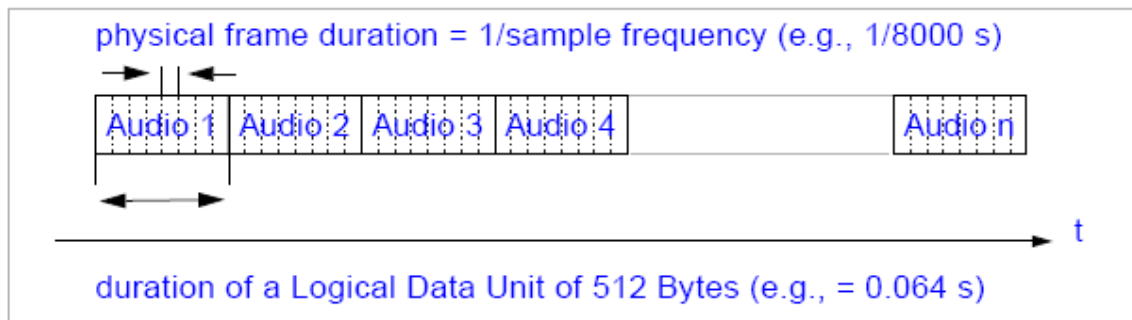
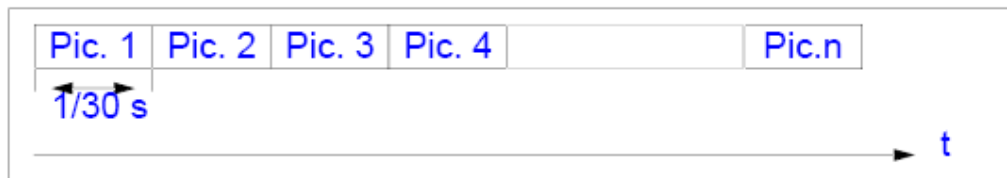
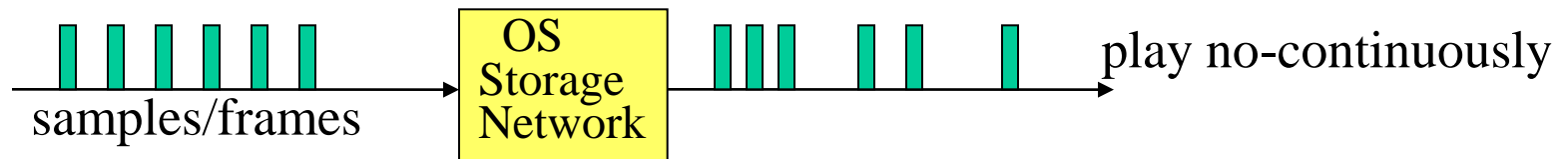
## ❑ Media integration catalog

- Media integration in operate system
- Media integration in storage system
- Media integration in database system
- Media integration in network system
- Media integration in human computer interface
- Media integration in message exchange
- Media integration in document representation
- Media integration in content presentation
- . . . . .

➔ A special & important issue: media synchronization

# Temporal Relations in Video and Audio

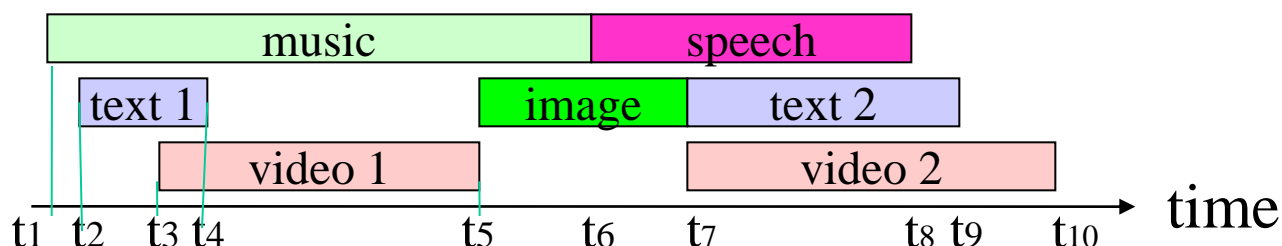
- ❑ Media are classified into
  - Discrete media (DM): text, still image, graphics image
  - Continuous media (CM): audio, video, animation
- ❑ CM are extremely time-sensitive !!!



# Media Synchronization and QoS

## Media synchronization: keep temporal relationships

- Intra-medium synchronization
- Inter-media synchronization



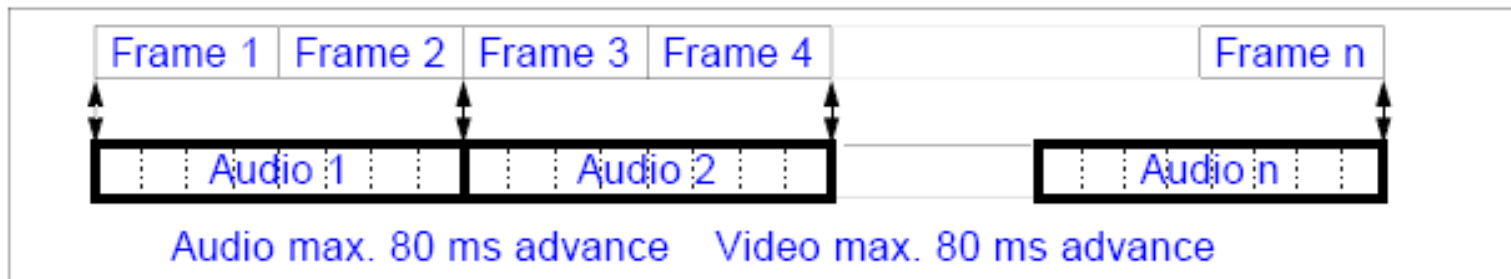
## QoS (Quality of Service):

- Specify media quality
  - The set of parameters that define the properties of media objects/streams
  - Performance, error rate, delay, jitter, time skew, ...
  - How to guarantee QoS
- key technology in mm OS, storage, network, ...

Media		Mode, Application	QoS
Video	Animation	correlated	+/- 120 ms
	Audio	lip synchronization	+/- 80 ms
	Image	overlay	+/- 240 ms
		non-overlay	+/-500 ms
	Text	overlay	+/- 240 ms
		non-overlay	+/-500 ms
Audio	Animation	event correlation (e.g., dancing)	+/- 80 ms
	Audio	tightly coupled (stereo)	+/- 11 $\mu$ s
		loosely coupled (dialogue mode with various participants)	+/- 120 ms
		loosely coupled (e.g., background music)	+/- 500 ms
	Image	tightly coupled (e.g., music with notes)	+/- 5 ms
		loosely coupled (e.g., slide show)	+/- 500 ms
	Text	Anmerkungen zu Text	+/- 240 ms
	Pointer	Audio Related to the Item	-500ms +750 ms

# Lip Synchronization

- Lip synchronization: Coupling between audio and video
- Acceptable Skew between video and audio:  $\sim 100\text{ms}$

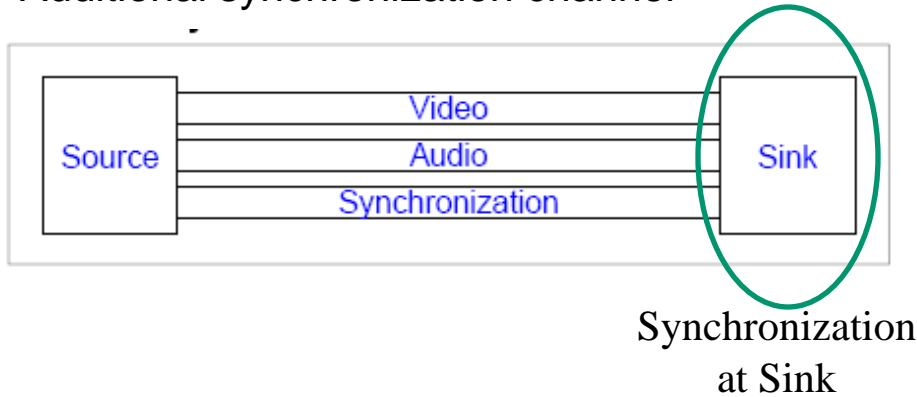




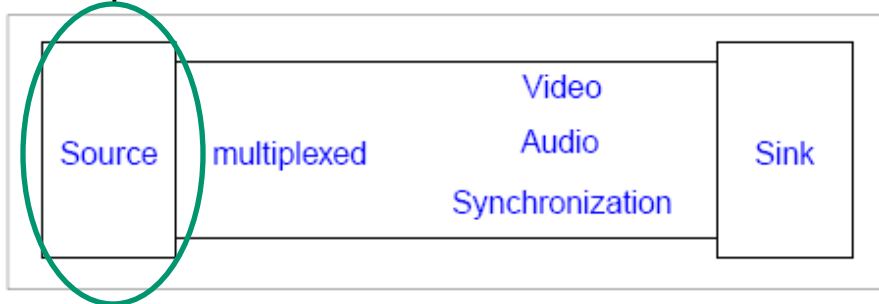
# Synchronization Specification and Location

Issues: where to put synchronization data?  
where to do synchronization task?

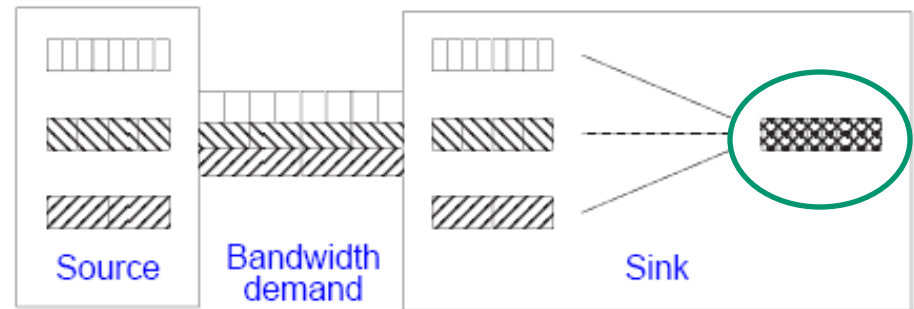
Additional synchronization channel



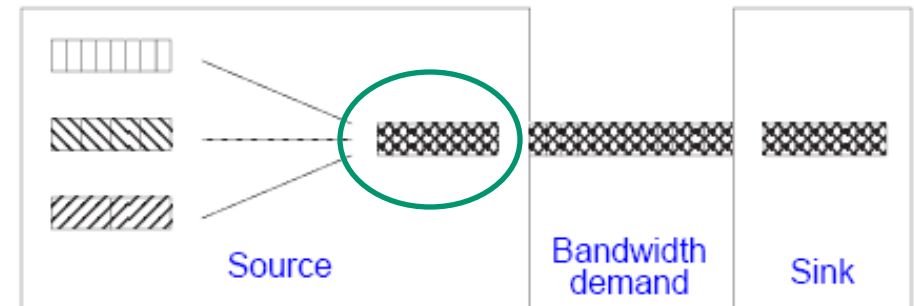
Multiplexed channel



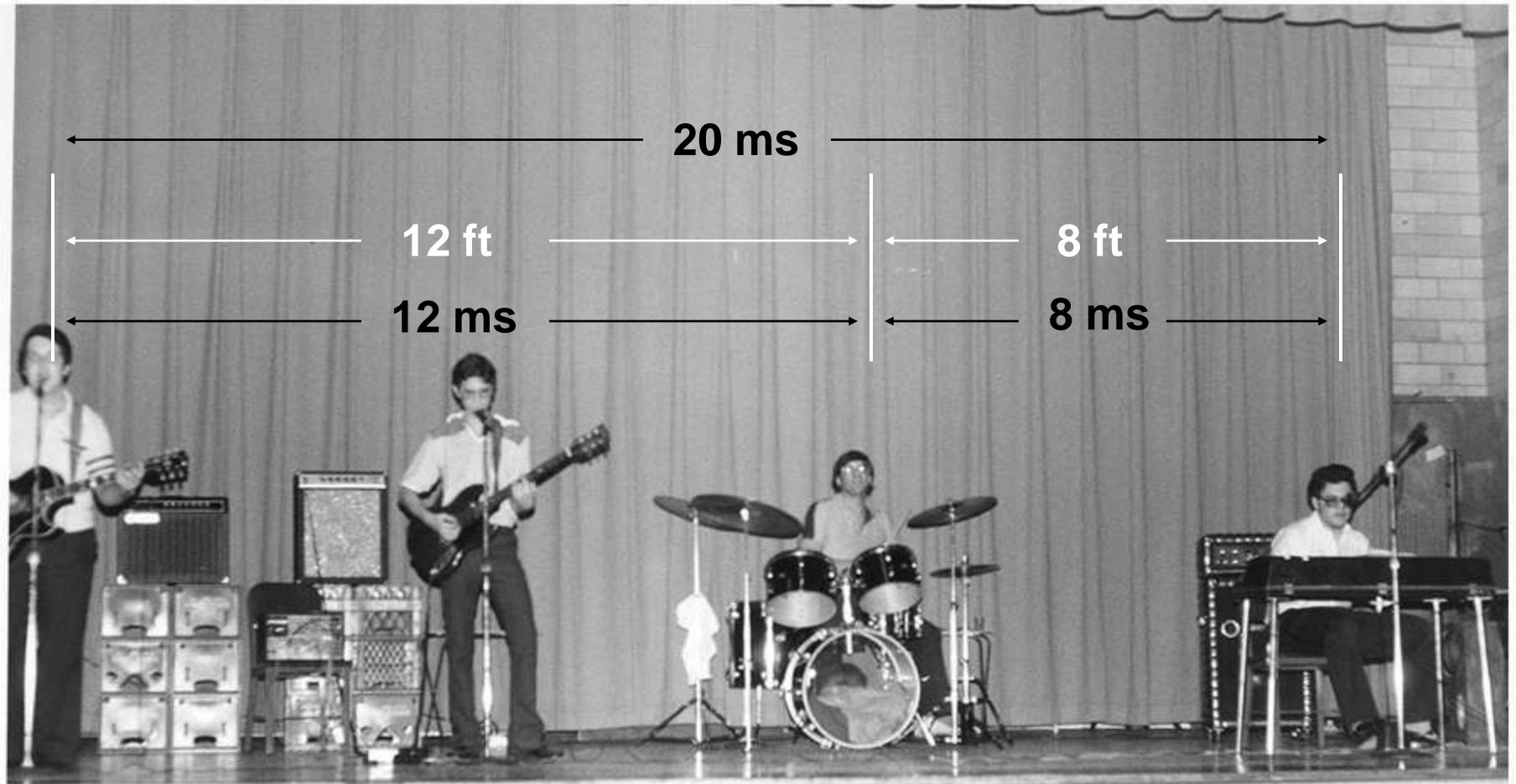
Synchronization at the sink node



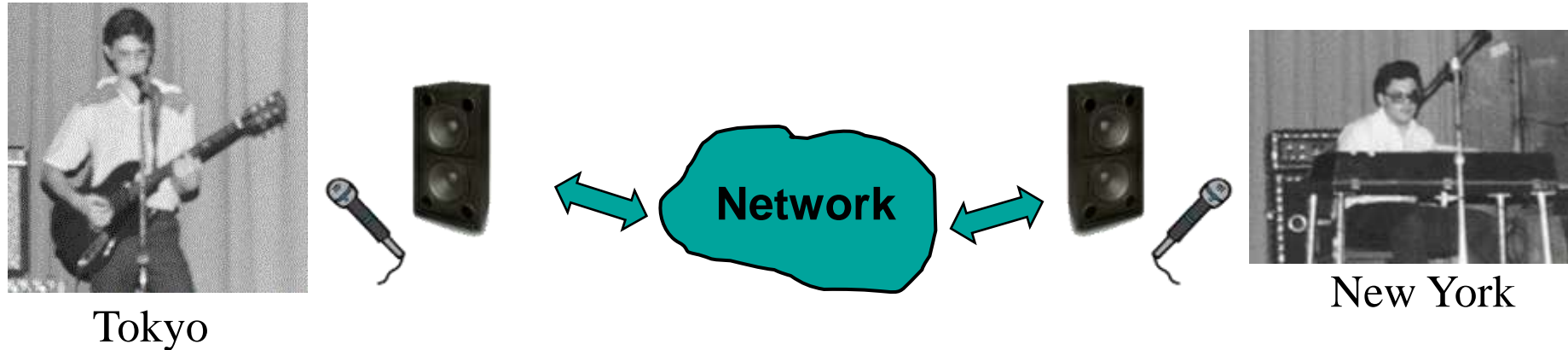
Synchronization at the source node



# Music Performance



# Distributed Music Over Network



- Adapt VOIP architecture for low latency:
  - Hosts use Real Time Protocol (RTP) to exchange audio streams
  - Effective if
    - host audio chain customized for low latency
    - low latency, over-provisioned network
    - Quality of Service (QoS) mechanisms (perhaps)
- Non-ideal network (BW limits, congestion, etc.)
  - Occasional packet delays and losses inevitable
  - Garbled sound (clicks and gurgles) due to small buffers

# Media Integration in Multimedia Presentation

## ❖ Multimedia presentation

- A process to assembly and synchronize all media objects/components that you have prepared to create a final multimedia product, such as a electronic file, a slide show, a web document, an e-book, etc.

## ❖ Presentation control elements

- Where? Spatial attribute (location, size, ...)
- When ? Time attribute (start and end time, synchronization, ...)
- How ? Effective attribute (volume, transition, relationships, ...)

## ❖ Presentation creation methods

- via computer languages
- via authoring tools

# Languages for Multimedia Presentation

General speaking, most of the computer languages are capable to make multimedia presentation products. But the following are often used:

## Programming Language

- C/C++, Visual Basic
- Perl
- Java

## Script Language

- JavaScript, ECMAScript (*European Computer Manufacturer's Association*)
- PHP
- Flash/Lingo (*Macromedia*)

## Markup Language

- HTML, DHTML, XML, SVG (*Scalable Vector Graphics*)
- SMIL, HTML+TIME
- WML

## Scene Description Language

- VRML
- BIFS (*Binary Format for Scene in MPEG-4*)
- DDL (*Description Definition Language in MPEG-7*)

# Multimedia Authoring Tools

- ❑ **Using computer language** to make multimedia presentations
  - *Need programming skill, hard for beginner, small size and flexible function*
- ❑ **Using Authoring Tools** to make multimedia presentations
  - A visualized authoring window using drag-and-drop via mouse
  - Less or no need for programming skill, large size and limited functions
- ❑ **Multimedia authoring tools**
  - Slide show based: from slide to slide in sequence of forward or backward
    - PowerPoint, Kai's Power Show, QuickTime Pro
  - Digital movie based: from begin to end
    - Macromedia Director
  - Branch based: providing users with a choice over where to go
    - Macromedia Authorware, Clickteam's Multimedia Fusion, Asymetrix's Toolbox
  - Web file based: creating a web document
    - Macromedia Dreamweaver, Adobe Golive, Frontpage, Netscape Composer
  - VR/3D file based: creating a animation or wml file
    - Macromedia Flush, MS Liquid Motion

# W3C Consortium

- ❑ **W3C**, founded in October 1994: <http://www.w3.org/>
- ❑ Purpose: develop common protocols that promote WWW's evolution and ensure its interoperability
- ❑ User Interface Domain
  - Hypertext Markup Language (HTML), Cascading Style Sheets (CSS), Document Object Model (DOM), **SMIL**, **SVG**
- ❑ Technology and Society Domain
  - Platform for Internet Content Selection (PICS), Resource Description Framework (RDF), Platform for Privacy Preferences (P3P)
- ❑ Architecture Domain
  - Hypertext Transfer Protocol (HTTP), Extensible Markup Language (XML)

...Follow links at <http://www.w3.org/> for more details...

# Embedding Audio/video into HTML

- ❑ Embed AV into a web page via programming/script language
  - Java applet
  - JavaScript or other scripts
- ❑ Embed AV into a web page via `<embed>` tag

## ➤ Embed audio

```
<embed src="path/MyAudio.wav" autostart="true" loop="true"></embed>
```

Note 1: The sound file begins to play as soon as it is loaded

Note 2: loop = "true" → play forever

Note 2: Plug-in is needed for playing audio file in .mov, .ra, .mp3, .aiff, etc.

## ➤ Embed video

```
<embed src="path/MyVideo.avi" width="320" height="240"
```

```
  autostart="true" loop="true">
```

```
</embed>
```

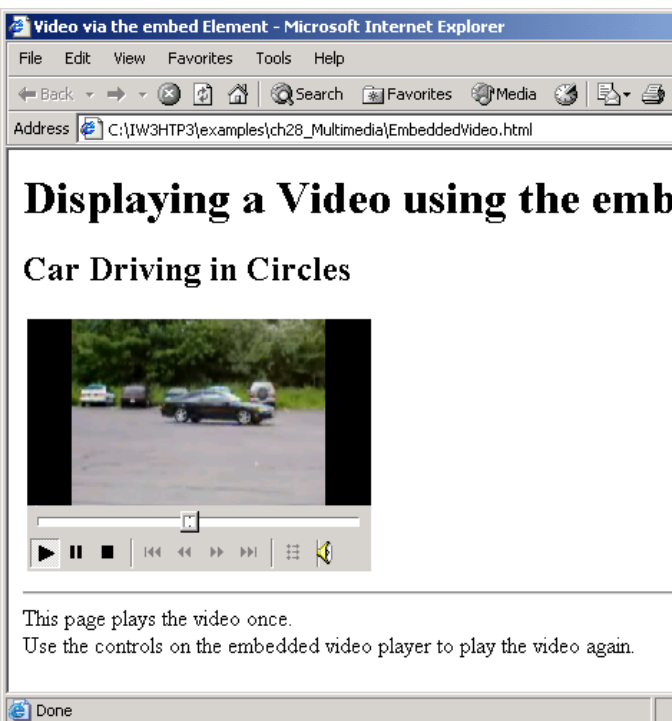
Note 1: The sound file begins to play as soon as it is loaded

Note 2: Plug-in is needed for playing video file in .mov, .ra, .mpg, etc.

## ➤ Embed tag is not enough to play multiple synchronized media object

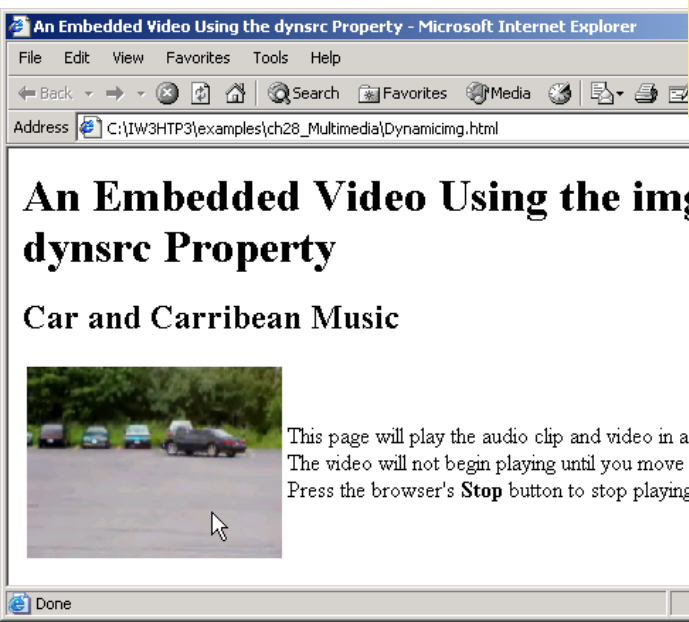


# Add Audio/video onto Webpage Using embed tag



```
1 <?xml version = "1.0"?>
2 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
3   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
4
5 <!-- Fig. 28.4: EmbeddedVideo.html -->
6 <!-- video via the embed Element -->
7
8 <html xmlns = "http://www.w3.org/1999/xhtml">
9   <head>
10     <title>Video via the embed Element</title>
11   </head>
12
13   <body>
14     <h1>Displaying a video using the embed Element</h1>
15     <h2>Car Driving in Circles</h2>
16
17     <table>
18       <tr><td><embed src = "car_hi.wmv" loop = "false"
19         width = "240" height = "176">
20       </embed></td>
21     </tr></table>
22
23     <hr />
24     This page plays the video once.<br />
25     Use the controls on the embedded video player to play the
26     video again.
27   </body>
28 </html>
```

# Add Audio/video onto Webpage Using img & dynsrc



```
1 <?xml version = "1.0"?>
2 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
3   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
4
5 <!-- Fig. 28.2: Dynamicimg.html -->
6 <!-- Demonstrating the img element's dynsrc property -->
7
8 <html xmlns = "http://www.w3.org/1999/xhtml">
9   <head>
10     <title>An Embedded Video Using the dynsrc Property</title>
11     <bgsound src =
12       "http://msdn.microsoft.com/downloads/sounds/carib.MID"
13       loop = "-1"></bgsound>
14   </head>
15
16   <body>
17     <h1>An Embedded Video Using the img element's
18       dynsrc Property</h1>
19     <h2>Car and Carribean Music</h2>
20     <table>
21       <tr><td><img dynsrc = "car_hi.wmv"
22         start = "mouseover" width = "180"
23         height = "135" loop = "-1"
24         alt = "Car driving in circles" /></td>
25       <td>This page will play the audio clip and video
26         in a loop.<br />The video will not begin
27         playing until you move the mouse over the
28         video.<br />Press the browser's<strong>Stop</strong>
29         button to stop playing the sound and the video.</td>
30     </tr>
31   </table>
32 </body>
33 </html>
```

# SMIL- Synchronized Multimedia Integration Language

- ❑ Define an XML-based language that allows authors to write interactive multimedia presentations → describe the temporal behaviour of a multimedia presentation, associate hyperlinks with media objects and describe the layout of the presentation on a screen.
- ❑ XML application enabling author to specify what should be presented **when**
- ❑ SMIL 1.0 specification, June 1998
- ❑ SMIL 2.0, August 2001, SMIL 2.1, December 2005
- ❑ SMIL 3.0, December 2008
  - Define a set of reusable markup modules that define the semantics
    - # Animation module
    - # Content control module
    - # Transition effect module
    - # .....
  - Module reuse in other XML based languages: WML, SVG, MPEG-4, etc
  - Others

# SMIL : Design Principles

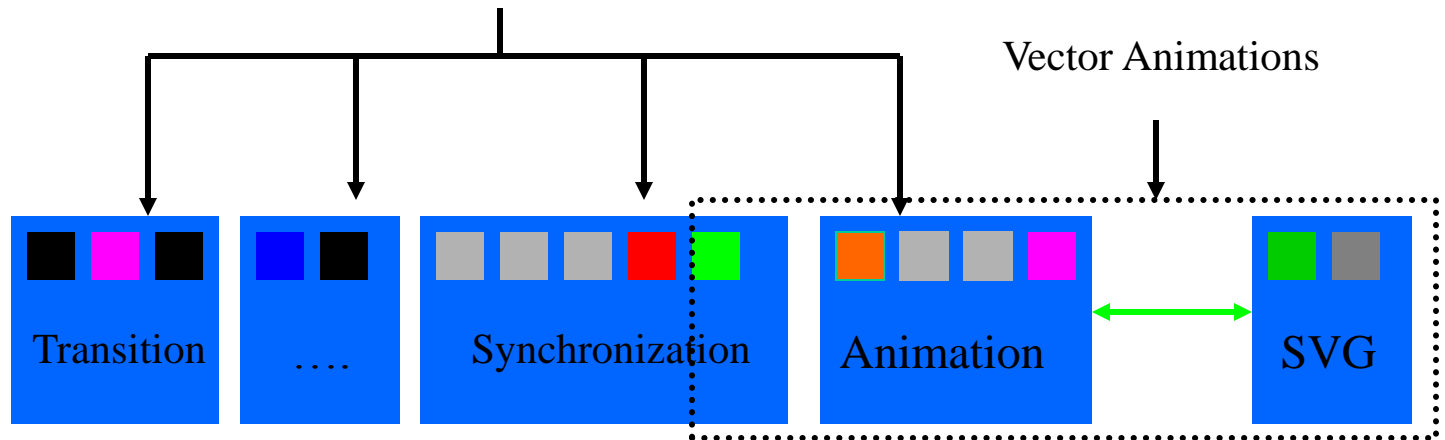
Meta-language which allows the description of multimedia documents ranging from the simplest to the very complex.

Languages space

1 application *profile*

Vector Animations

Functional space



Syntactic and  
compositional  
space,  
programming  
APIs, ...

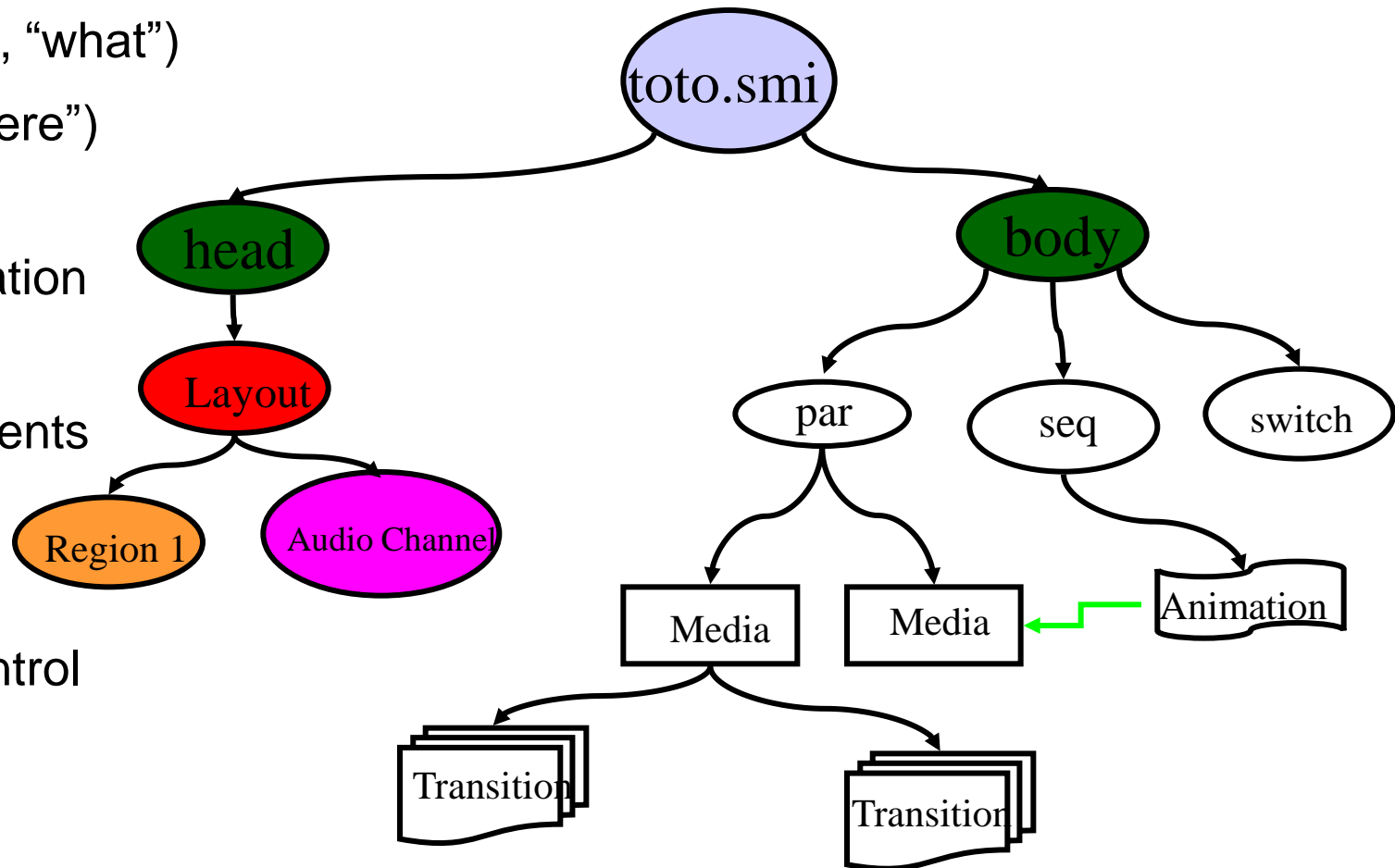
DOM 1-2  
SMIL DOM

XML

Namespaces

# SMIL Structure and Modules

- Structure
- Meta (“who”, “what”)
- Layout (“where”)
- Timing and Synchronization (“when”)
- Media Elements (“what”)
- Linking
- Content Control



# SMIL High Level Document Structure

```
<smil>  
  <head>  
    <meta>  
      <!-- ... information about the document ... -->  
    </meta>  
    <layout>  
      <!-- ... definitions used for the spatial layout ... -->  
    </layout>  
  </head>  
  <body>  
    <!-- ... objects, temporal relations, links ... -->  
  </body>  
</smil>
```

# SMIL Meta

```
<smil>  
  <head>  
    <meta ... />  
  </head>  
</smil>
```

The meta elements contain information describing the document, either to inform the human user or to assist some automation, e.g.,

```
<meta name="title" content="My Italy Trip"/>  
<meta name="copyright" content="©1998 WGBH" />  
<meta name="base" content="http://billswin.edu/Italy/" />
```

# SMIL Layout

```
<smil>
<head>
  <layout>
    <root-layout ... />
    <region id="R1" ... />
    <region id="R2" ... />
  </layout>
</head>
</smil>
```

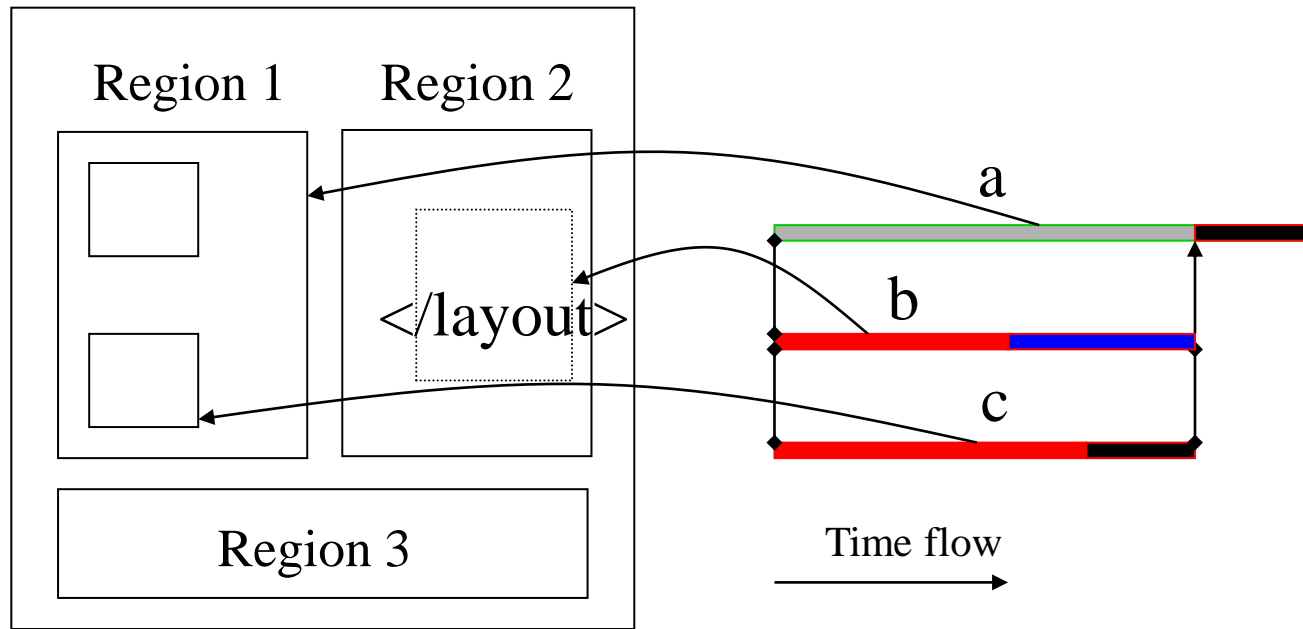
Includes the <layout>, <root-layout>, <region> elements, and related attributes.



## Example

```
<root-layout height="300" width="450"
  background-color="#FFFFFF"
  title="Venezia!"/>
<root-layout height="450" width="625"
  background-color="black"/>

<region id="title" left="5" top="150"
  width="400" height="200"
  z-index="1"/>
<region id="videoregion" top="0"
  left="0" height="240"
  width="352"/>
```





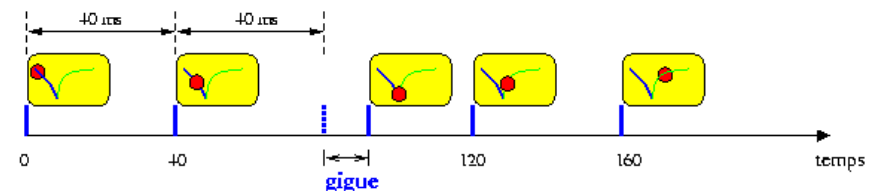
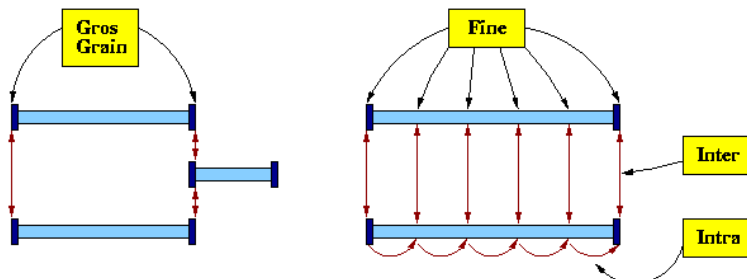
# SMIL Timing and Synchronization

```
<smil>
  <body>
    <!-- ... timing included here ... -->
  </body>
</smil>
```

- Sequence and parallel timelines, via **<seq>** and **<par>**
- Timing control properties, via **“begin”**, **“end”**, **“repeat”**, **“dur”**
- “The accuracy of synchronization between the children in a parallel group is implementation-dependent”
  - “soft synchronization” vs. “hard synchronization”
  - how to realize hard synchronization

# Hard vs. Soft Synchronization

- Hard synchronization: player synchronizes the children in the "par" (parallel play) element to a common clock
- Soft synchronization: each child of the "par" element has its own independent clock
- syncBehavior
  - **canSlip** : the synchro is loose, child elements can slip from the parent clock
  - **locked** : the Synchronization is hard (lipsync), amount of tolerated slipping (syncTolerance).
  - **Independent** : synchro completely independent
- syncTolerance = "amount of jitter"
- syncMaster = "true" clock ticker of the par element



# SMIL Media Elements

<smil>

  <body>

    <!-- ... media elements included here ... -->

  </body>

</smil>

- Includes the media declaration elements <text>, <img>, <audio>, <video>, <textstream>, <animation>, and <ref>
- all elements (animation, img, ref, text, textstream and video) are contained within a single containing block defined by the root-layout element

# SMIL Linking

<smil>

<body>

<!-- ... linking included here ... -->

</body>

</smil>

- Includes the <a> and <anchor> elements, e.g.,

<a href="http://www.w3c.org" >

<text src="media/w3c.txt" region="w3c"  
begin="14.05s" dur="15.95s" />

</a>

- Timing applied to HTML <a> and <area> tags could provide much or all of SMIL functionality; hence, linking modules under review

# An Example of SMIL File

```
<smil xmlns="http://www.w3.org/2001/SMIL20/Language">
  <head>
    <layout>
      <topLayout width="640px" height="480px">
        <region id="whole" top="0px" left="0px"
          width="640px" height="480px" />
      </topLayout>
    </layout>
  </head>
  <body>
    <seq>
      
      
        dur="3s"/>
      
      
    </seq>
  </body>
</smil>
```

# SMIL Browsers and Authoring Tools

## SMIL browser

- [RealOne Platform](#) by RealNetworks with full support for the SMIL 2.0
- [GRiNS for SMIL-2.0](#) by Oratrix provides a SMIL 2.0 player
- [Internet Explorer 6.0](#) by Microsoft including [XHTML+SMIL Profile](#)
- [X-Smiles](#), version 0.4 a new java-based XML browser

## Authoring Tools

- [GRiNS Editor](#) by Oratrix based on SMIL2 Editor family and streamlined
- [SMILGen](#) by RealNetworks, a SMIL (and XML) authoring tool
- [Ezer](#) by SMIL Media
- [Fluition](#) by Confluent Technologies
- [Grins](#) by Oratrix

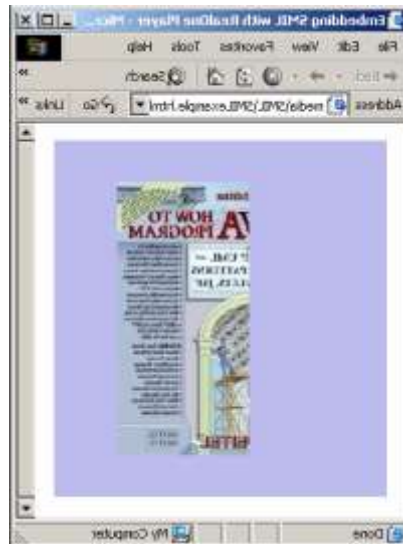
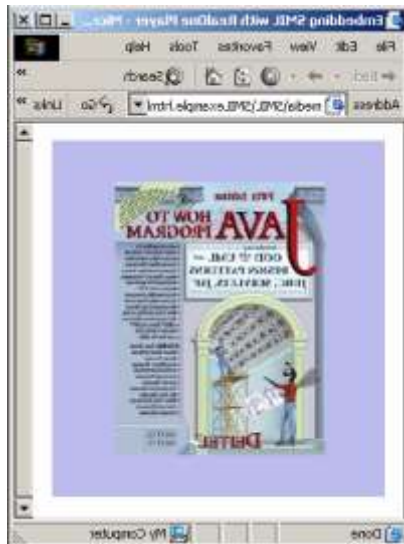
# Another Example of SMIL File

```
1 <smil xmlns="http://www.w3.org/2000/SMIL20/CR/Language">
2
3   <!-- Fig. 20.15 : exampleSMIL.smil -->
4   <!-- Example SMIL Document      -->
5
6   <head>
7     <layout>
8       <root-layout height = "300" width = "280"
9         background-color = "#bbbbee" title = "Example" />
10
11       <region id = "image1" width = "177" height = "230"
12         top = "35" left = "50" background-color = "#bbbbee" />
13     </layout>
14
15     <transition id = "wipeForward" dur = "2s" type = "barwipe" />
16     <transition id = "wipeBackward" dur = "2s" type = "barwipe"
17       subtype = "topToBottom" />
18
19     <transition id = "fadeIn" dur = "2s" type = "fade"
20       subtype = "fadeFromColor" fadeColor = "#bbbbee" />
21
22     <transition id = "fadeOut" dur = "2s" type = "fade"
23       subtype = "fadeToColor" fadeColor = "#bbbbee" />
24
```

```
25   <transition id = "crossFade" type = "fade" subtype = "crossfade"
26     dur = "2s" />
27
28 </head>
29 <body>
30   <seq>
31     <par>
32       <img src = "book1.jpg" region = "image1"
33         transIn = "wipeForward" transOut = "wipeForward"
34         alt = "book1" dur = "6s" fill = "transition"
35         fit = "fill" />
36       <audio src = "bounce.au" dur = ".5s" />
37     </par>
38     <par>
39       <img src = "book2.jpg" region = "image1" transIn = "fadeIn"
40         transOut = "fadeOut" alt = "book2" dur = "6s"
41         fit = "fill" fill = "transition" />
42       <audio src = "bounce.au" dur = ".5s" />
43     </par>
44     <par>
45       <img src = "book3.jpg" region = "image1"
46         transIn = "wipeBackward" transOut = "fadeOut"
47         alt = "book3" dur = "6s" fit = "fill"
48         fill = "transition" />
49       <audio src = "bounce.au" dur = ".5s" />
50     </par>
51     <par>
52       <img src = "book4.jpg" region = "image1" transIn = "crossFade"
53         transOut = "fadeOut" alt = "book4" dur = "6s"
54         fit = "fill" fill = "transition" />
55       <audio src = "bounce.au" dur = ".5s" />
56     </par>
57     <par>
58       <img src = "book5.jpg" region = "image1"
59         transIn = "wipeForward" transOut = "wipeBackward"
60         alt = "book5" dur = "6s" fit = "fill"
61         fill = "transition" />
62       <audio src = "bounce.au" dur = ".5s" />
63     </par>
64     <par>
65       <img src = "book6.jpg" region = "image1"
66         transIn = "crossFade" alt = "book6" dur = "6s"
67         fit = "fill" fill = "transition" />
68       <audio src = "bounce.au" dur = ".5s" />
69     </par>
70   </seq>
71
72 </body>
73 </smil>
```

# Another Example of SMIL File

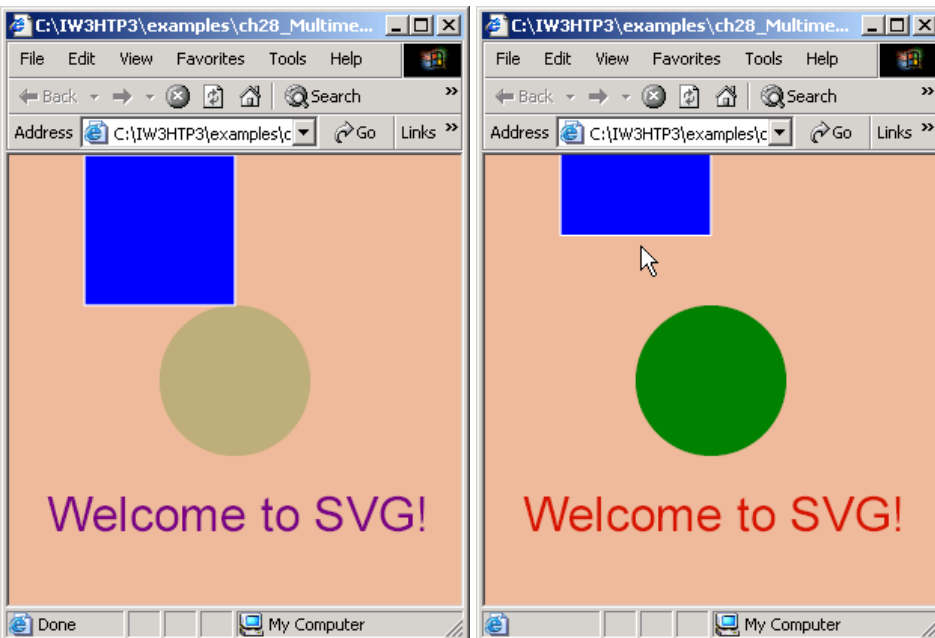
```
1 <?xml version = "1.0"?>
2 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
3   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
4
5 <!-- Fig. 28.16: SMILexample.html -->
6 <!-- embedding SMIL with RealOne Player -->
7
8 <html xmlns = "http://www.w3.org/1999/xhtml">
9   <head>
10     <title>Embedding SMIL with RealOne Player</title>
11   </head>
12   <body>
13     <div style = "text-align: center">
14       <embed src = "exampleSMIL.smil"
15         controls = "Imagewindow"
16         type = "audio/x-pn-realaudio-plugin"
17         width = "280" height = "300" autostart = "true">
18     </embed>
19   </div>
20 </body>
21 </html>
```





# SVG Scalable Vector Graphics

Produced by mathematical  
equations via XML vocabulary



```
1 <?xml version = "1.0"?>
2
3 <!-- Fig. 28.17 : shapes.svg -->
4 <!-- Simple example of SVG -->
5
6 <svg viewBox = "0 0 300 300" width = "300" height = "300">
7
8   <!-- Generate a background -->
9   <g>
10     <path style = "fill: #eebb99" d = "M0,0 h300 v300 h-300 z"/>
11   </g>
12
13   <!-- Circle shape and attributes -->
14   <g>
15
16     <circle style = "fill:green;" cx = "150" cy = "150" r = "50">
17       <animate attributeName = "opacity" attributeType = "CSS"
18         from = "0" to = "1" dur = "6s" />
19     </circle>
20
21   <!-- Rectangle shape and attributes -->
22   <rect style = "fill: blue; stroke: white"
23     x = "50" y = "0" width = "100" height = "100">
24     <animate attributeName = "y" begin = "mouseover" dur = "2s"
25       values = "0; -50; 0; 20; 0; -10; 0; 5; 0; -3; 0; 1; 0" />
26   </rect>
27
28   <!-- Text value and attributes -->
29
30   <text style = "fill: red; font-size: 24pt"
31     x = "25" y = "250"> Welcome to SVG!
32     <animateColor attributeName = "fill"
33       attributeType = "CSS" values = "red;blue;yellow;green;red"
34       dur = "10s" repeatCount = "indefinite"/>
35   </text>
36 </g>
37 </svg>
```

# HTML+TIME (Timed Interactive Multimedia Extensions)

- ❑ Proposed by Microsoft, presently not been endorsed by W3C
- ❑ HTML+TIME 1.0 is based on SMIL 1.0 and supported in IE5+
- ❑ HTML+TIME 2.0 is based on SMIL 2.0 and supported in IE 5.5+
- ❑ Add timing and media synchronization support to HTML pages
  - media elements: `t:ANIMATION`, `t:AUDIO`, `t:VIDEO`, `t:IMG`
  - control elements: `t:EXCL`, `t:SEQ`, `t:PAR`
- ❑ Use both timeline model and event-driven model
  - `BEGIN`, `DUR`, `BeginWith`
- ❑ HTML+TIME structure

```
<HTML XMLNS:t="urn:schemas-microsoft-com:time">
<HEAD>
<STYLE> .time {behavior: url(#default#time2);} </STYLE>
<?IMPORT namespace="t" implementation="#default#time2">
</HEAD>
<BODY>

    . . . . .

</BODY>
</HTML>
```

# VRML (Virtual Reality Modeling Language)

- ❑ Pronounced either V-R-M-L or “Vermal”
- ❑ A language that describes geometry and behavior of a 3D scene or “world”
- ❑ Based on SGI’s Moving World languages
- ❑ SMIL 1.0 (1995), VRML 2.0/VRML97, ISO standard (ISO/IEC-14772-1:1997)
- ❑ “World” can be single or a group of files, ranged from simple to complex scene
- ❑ A VRML file is a plain UTF-8 or ASCII text file ended with `.wrl`
- ❑ Use a plain text editor (e.g. Notepad) to input, modify and save a VRML file

```
#VRML V2.0 utf8
DEF APP Appearance {material Material{ diffuseColor 1 0 0 } }
Shape{ appearance USE APP
      geometry Cylinder{ radius 1 height 5 } }
```

- ❑ VRML file can be viewed by a specific VRML browser or a Web browser with plug-in  
- 3D ObjectViewer, Cosmo Player, Community Place, GL View, WebDimension, WorldView, etc.
- ❑ VRML file can be embedded in a HTML file via `<Frame>`, `<EMBED>`, `<OBJECT>`  
`<FRAME SRC="my.wrl" width="300" height="280">`  
`<EMBED SRC="my.wrl" width="300" height="280">`

# Demos

☐ SMIL

☐ SVG

☐ VRML

# Media Protection

- Media Protection
- Media Encryption
- Media Watermark

# What is Media Protection?

- New technologies bring with them new issues:
  - Advances in compression techniques make it possible to create high-quality digital content (audio, video, still pictures, etc.)
  - Advances in the network protocols and infrastructure makes it possible to store, stream and distribute this content in a very large scale.
- Media protection or Digital Rights Management (DRM) is the set of techniques used to:
  - Control access to content:
    - Viewing rights
    - Reproduction (copying) rights
- Essentially, media protection is the management of the author's and publisher's intellectual property (IP) in the digital world.

# Media Protection Principles

- Encryption of the content to disallow uncontrolled access.
- Decryption key management.
- Access control according to flexible usage rules
  - Number of times content can be accessed; times it can be accessed; trading of access rights.
- Copy control or copy prevention
  - Management of the number of copies that can be made of the content.
- Identification and tracing of multimedia data.
  - May be a requirement even if the copy is made from the analog version of the content, e.g., recording the analog outputs of a digital playback.

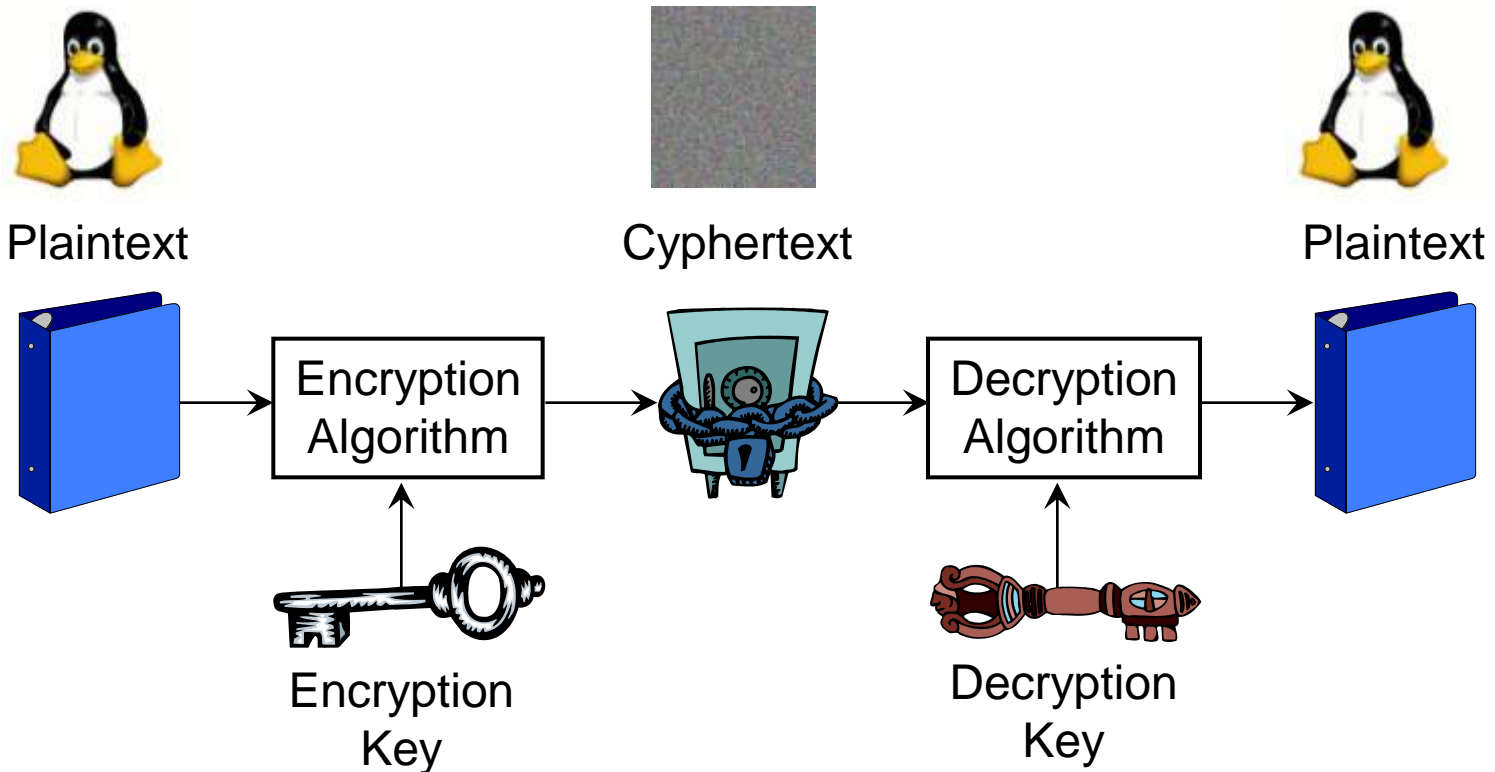
# Underlying Technologies

- DRM is based on two fundamental underlying technologies:
  - Encryption
  - Watermarking
- **Encryption** is used to “lock” the content and deny access to it to those parties that do not possess the appropriate keys
  - Encryption enforces the restrictions placed on the content by the author/publisher
- **Watermarking** is used to “mark” the content so that a particular copy can be traced back to the original user
  - Digital Watermarking is used as a deterrent to large-scale unauthorized copying of copyrighted material.



# Encryption

- Encryption is the process of “obscuring” a message (content, media, file, etc.) so that it is undecipherable without the key.



# Types of Encryption

- ***Symmetric (Secure Key) Encryption:***



encryption and decryption keys are the same.

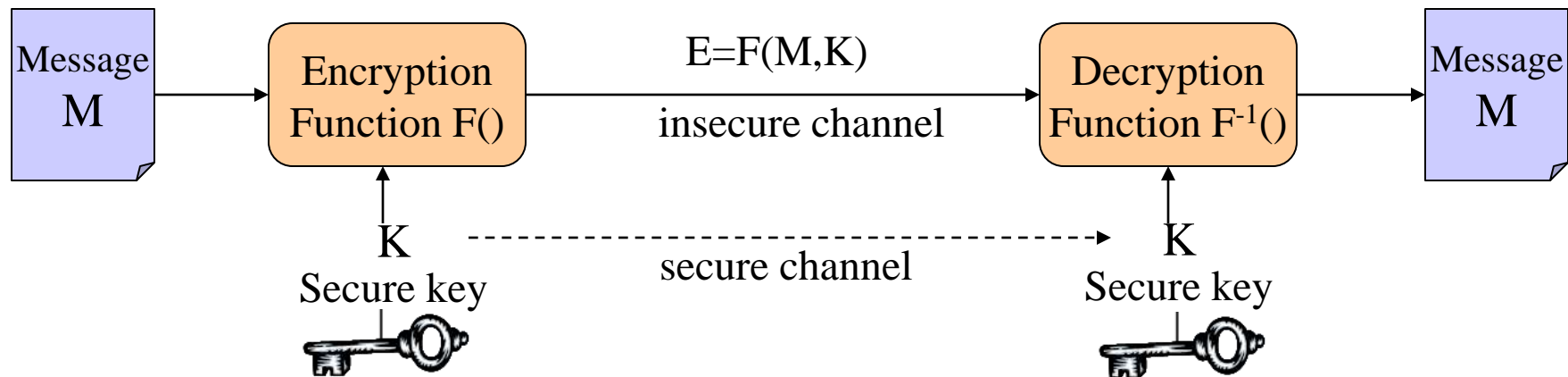
- ***Asymmetric (Public Key ) Encryption:***



keys come in pairs, one to encrypt, another to decrypt.

- Used in Public-Key cryptography, where one key in the pair is kept secret, and another is published.
- Whatever is encrypted with one key can only be decrypted with the other and vice-versa.
- Symmetric keys are very efficient, but need to remain a secret and must be securely communicated between the participants.
- Asymmetric Encryption is much slower than Symmetric Encryption and requires much larger key lengths to achieve the same level of protection.
- Asymmetric keys (public/private) are slow and inappropriate for actual content exchange.
- Idea: use asymmetric keys to encrypt the symmetric keys, in order to securely communicate them.

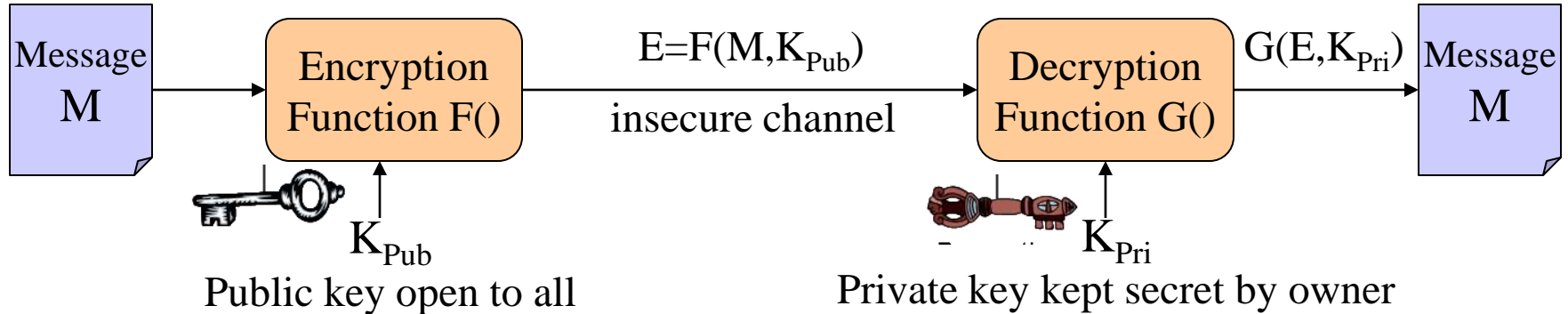
# Secure Key Encryption



## Encryption Standards

- **DES** (Data Encryption Standard)
  - designed originally by IBM, and adopted by the US government in 1977 and by ANSI in 1981
  - 64-bit block (encryption unit) and 56-bit key
  - not recommended use after 1998 because it can be broken
- **Triple-DES**
  - three keys and three executions of DES
- **IDEA** (International Data Encryption Algorithm) - 128-bit block/key
- **AES** (Advanced Encryption Standard) - 128-bit block/key

# Public Key Encryption



## RSA (Rivest, Shamir, Adleman, 1978)

### • Key Generation

- Select  $p, q$  which are primes
- Calculate  $n=p \times q$ , and  $t(n)=(p-1) \times (q-1)$
- Select integer  $e$  satisfied  $\gcd(t(n), e)=1$  and  $e < t(n)$
- Calculate  $d$  satisfied  $exd=1 \pmod{t(n)}$
- Public key:  $KU=\{e, n\}$
- Private key:  $KR=\{d, n\}$

### • Encryption

- Plaintext:  $M < n$
- Ciphertext:  $C = M^e \pmod{n}$

### • Decryption

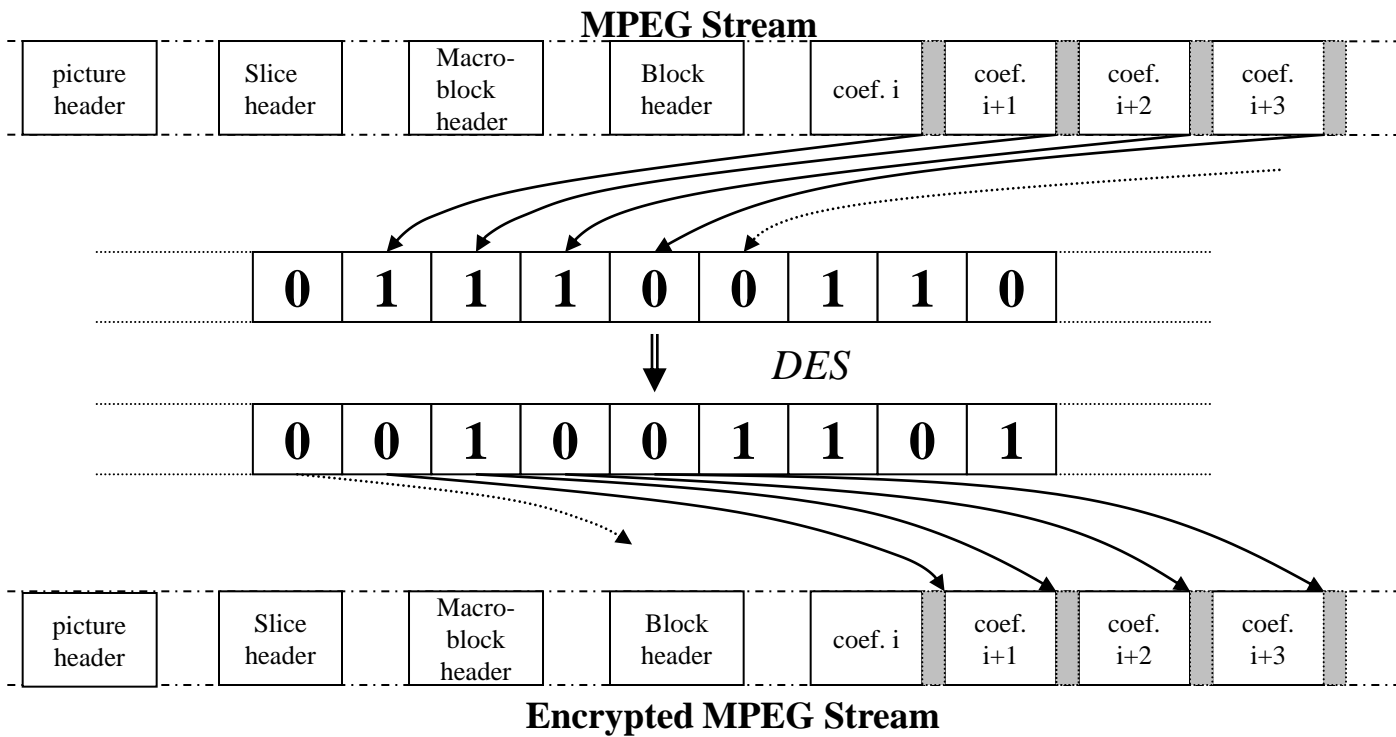
- $M = C^d \pmod{n}$

- Hard to factor  $n$  into 2 primes  $p$  and  $q$
- RSA key size: 128 to 300 decimal digitals  
i.e., 425 to 1024 bits
- RSA needs more computations than DES  
much slower than DES

### • Example

- Given  $M=19$
- Select two prime numbers  $p=7$  and  $q=17$
- Calculate  $n=7 \times 17=119$ , and  $t(n)=6 \times 16=96$
- Select  $e=5$
- Determine  $d=77$  since  $5 \times 77=385=4 \times 96+1$
- Ciphertext  $C=19^5 \pmod{119}=66$
- Decryption  $66^{77} \pmod{119}=19$

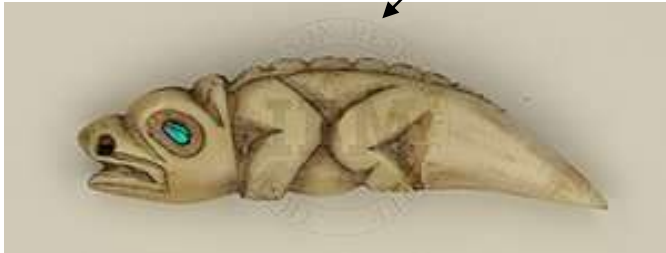
# MPEG Video Encryption Example



# Watermarking

- Watermarking is the addition of unremovable data to multimedia content, for the purposes of copy identification and tracking.
- **Visible** watermark and **invisible** watermark

Visible even very light

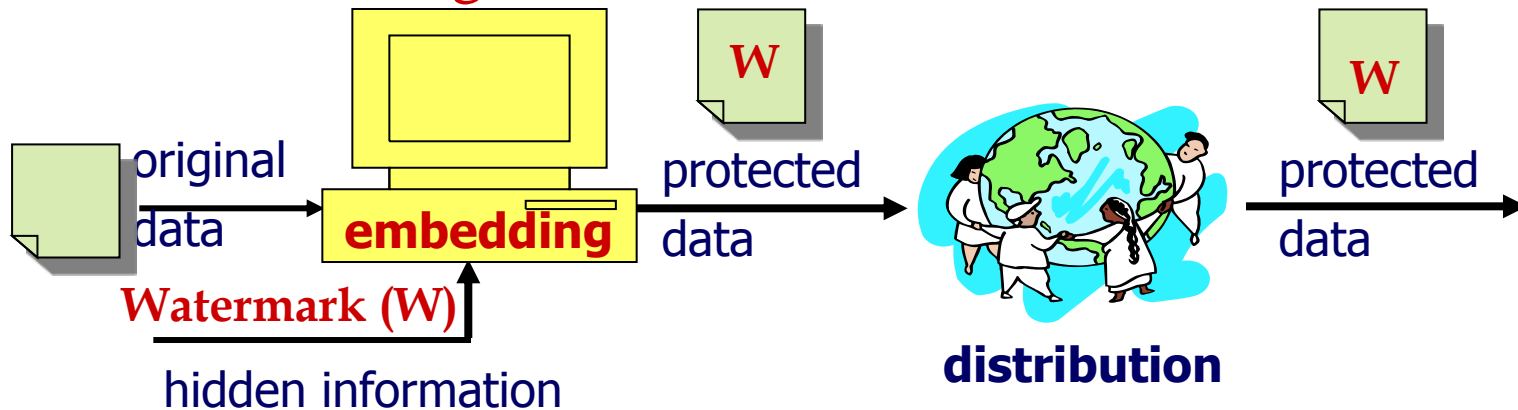


Special Mark:, e.g.,  
Owner's name/sign  
Added to the image  
But imperceptible  
Hidden mark!!



# Watermarking Principle and Requirements

- ✓ Principle: insert information that travels with the protected data, **wherever it goes**



- The requirements for such a system are:
  - **Imperceptibility:** the addition of the watermark must not degrade the content in a perceptible way.
  - **Security:** the watermark must only be accessible by authorized parties.
  - **Robustness:** the watermark must survive data manipulation, including malicious manipulation with the intent of removing the watermark.

# Watermarking of Text

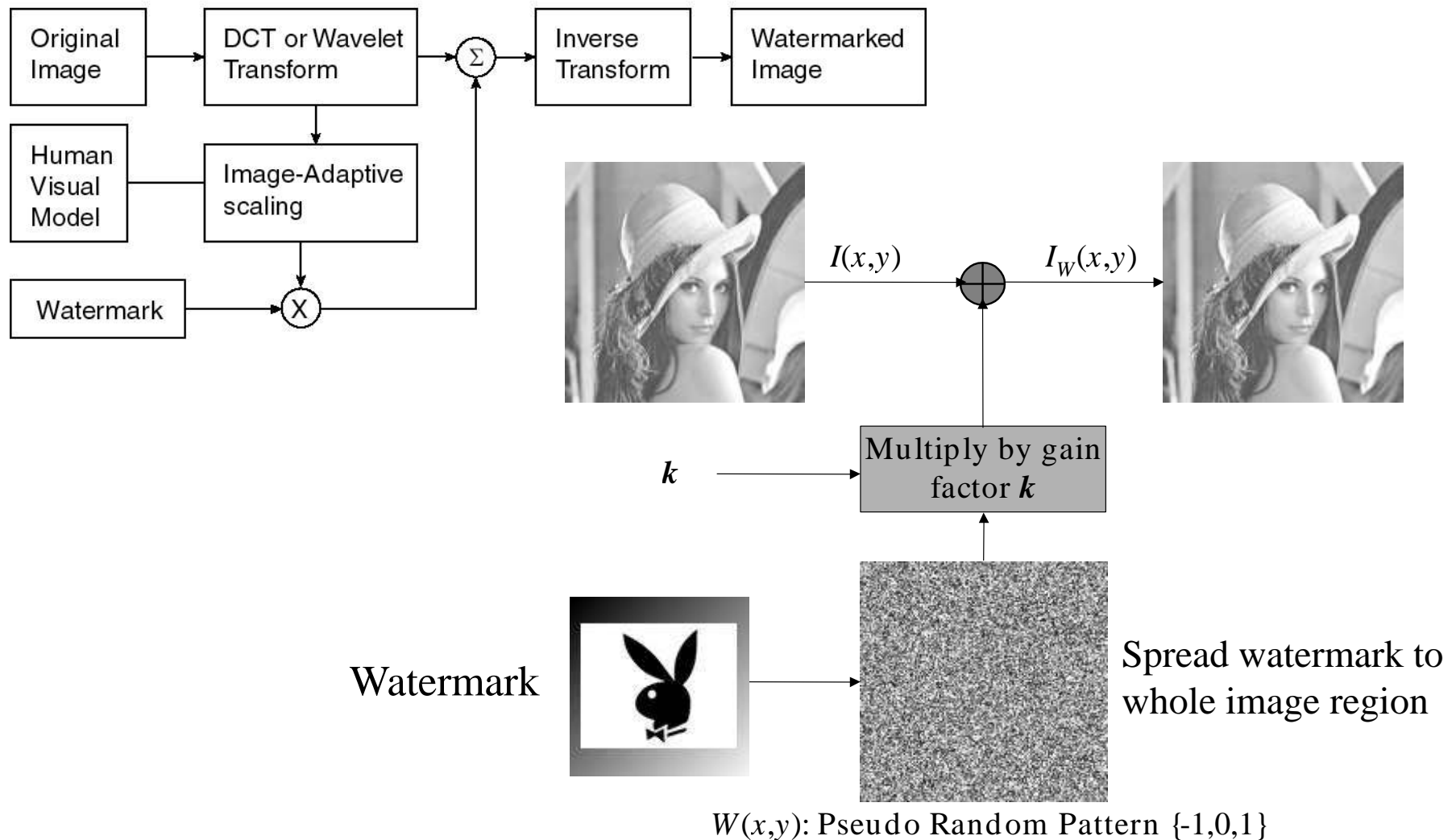
- Watermarking of formatted text is done using one of the following techniques:
  - Line shift coding: moving the lines of text up or down slightly; information is encoded in the way lines are shifted.
  - Word shift coding: same idea, but using spaces between words; much harder to extract.
  - Feature coding: slightly modify features such as the end line lengths of characters such as *b*, *d* and *h*.
- These techniques survive printing, consecutive photocopying up to 10 generations, and scanning.
- Easy to defeat, however: retype the text (OCR or manual).



# Watermarking of Still Images

- There is a large body of techniques and literature on watermarking of still images.
- In general terms, the watermark is applied to the original, uncompressed image.
  - Some watermarks are designed in the space domain, while others are applied in the frequency domain.
- Some watermarks are designed to survive still image compression (JPEG), while others cannot.
- Simplest technique: replace the LSB of each pixel by a bit from the watermark.
  - Watermarks will be encoded in sequences of bits.
  - Image may be compressed to less bits prior to the injection of the watermark.

# Additive Watermark with Spread Spectrum



# Sample Images



Original



Watermarked

Source: <http://dynamo.ecn.purdue.edu/~ace/water2/digwmk.html>

Prof. Edward J. Delp's research group

# Video Watermarking

- In general, the same techniques used for still images can be applied to video.
- Considerations:
  - The signal space for video is much larger than for still images; there is no need to use very complex schemes to minimize distortion while maximizing capacity.
  - Video watermarking schemes need to be less complex because in most cases they need to run in real time and need to address compressed video.
  - Video watermarks must be able to survive frame averaging, dropping and swapping - spread information over multiple frames
  - Depending on application, it is desirable to retrieve the watermark from short sequences from the material.

# Video Watermarking Techniques

- DCT-based method:
  - Use the watermark to modulate a pseudo-noise signal of the same dimensions as the video.
  - Compute the DCT of the watermark and add it to the DCT of the original video.
  - Do not use the coefficient if this increases data rate too much.
  - Add drift compensation to avoid artifacts.
  - Typically capable of achieving around 50 bits/sec watermark.
- Motion-Vector method:
  - Find motion vectors that point to flat areas.
  - Slightly modify them to add the watermark information (randomized).
  - Watermark can be derived directly from motion vectors.

# Audio Watermarking

- When compared with video, audio introduces the following issues:
  - Much less samples resulting in lower watermark capacity.
  - Humans are much less tolerant to audio changes than to video changes; harder to achieve imperceptibility.
- Basic spread-spectrum technique is also used for audio, but needs to be refined.
  - Example: making the power of the watermark signal vary with the overall power of the audio.
- Lot of activity in this area
  - See the “Secure Digital Music Initiative” (SDMI), at <http://www.sdmi.org>.

# Demos of Image Watermark

# Media Retrieval

- Information Retrieval
- Image Retrieval
- Video Retrieval
- Audio Retrieval

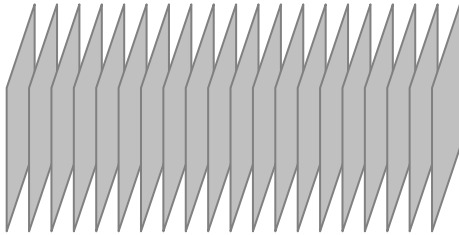


# Information Retrieval

- ❑ Retrieval = Query + Search
- ❑ Informational Retrieval: Get required information from database/web
- ❑ Text data retrieval
  - via keyword searching in a text document or through web
  - via expression such as in relational database
- ❑ Multimedia retrieval
  - Get similar images from an image database
  - Find interesting video shots/clips from a video/database
  - Select news from video/radio Internet broadcasting
  - Listen specific sound from audio database
  - Search a music
- ❑ Challenges in multimedia retrieval
  - Can't directly text-based query and search?
  - How to analysis/describe content and semantics of image/video/audio?
  - How to index image/video/audio contents?
  - Fast retrieval processing and accurate retrieval results

# Audio Visual Content/Feature

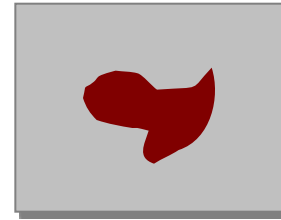
## Video segments



### Content/ Features

- Color
- Camera motion
- Motion activity
- Mosaic

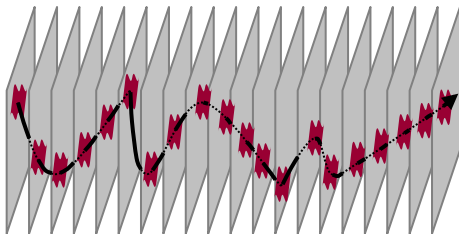
## Still regions



### Content/ Features

- Color
- Shape
- Position
- Texture

## Moving regions



### Content/ Features

- Color
- Motion trajectory
- Parametric motion
- Spatio-temporal shape

## Audio segments



### Content/ Features

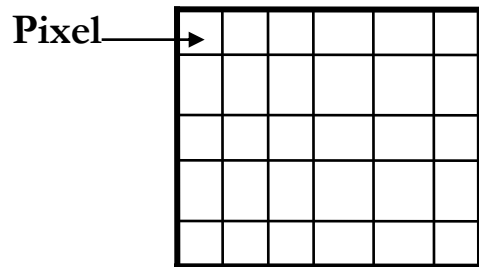
- Spoken content
- Spectral characterization
- Music: timbre, melody, pitch

# Image Content – Image Features

- What are image features?
- Primitive features
  - Mean color (RGB)
  - Color Histogram
- Semantic features
  - Color distribution, texture, shape, relation, etc...
- Domain specific features
  - Face recognition, fingerprint matching, etc...

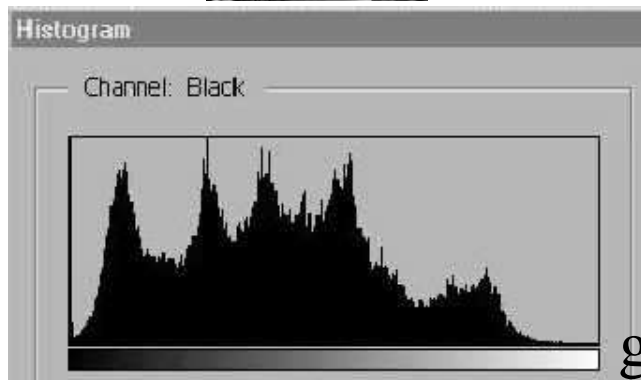
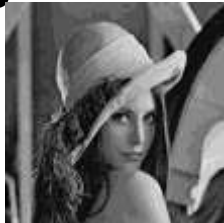
# Mean Color and Color Histogram

- Pixel Color Information: R, G, B
- **Mean Color** (R,G or B) =  $\frac{\text{Sum of that component for all pixels}}{\text{Number of pixels}}$



Number of pixels

- **Histogram:** Frequency count of each individual color

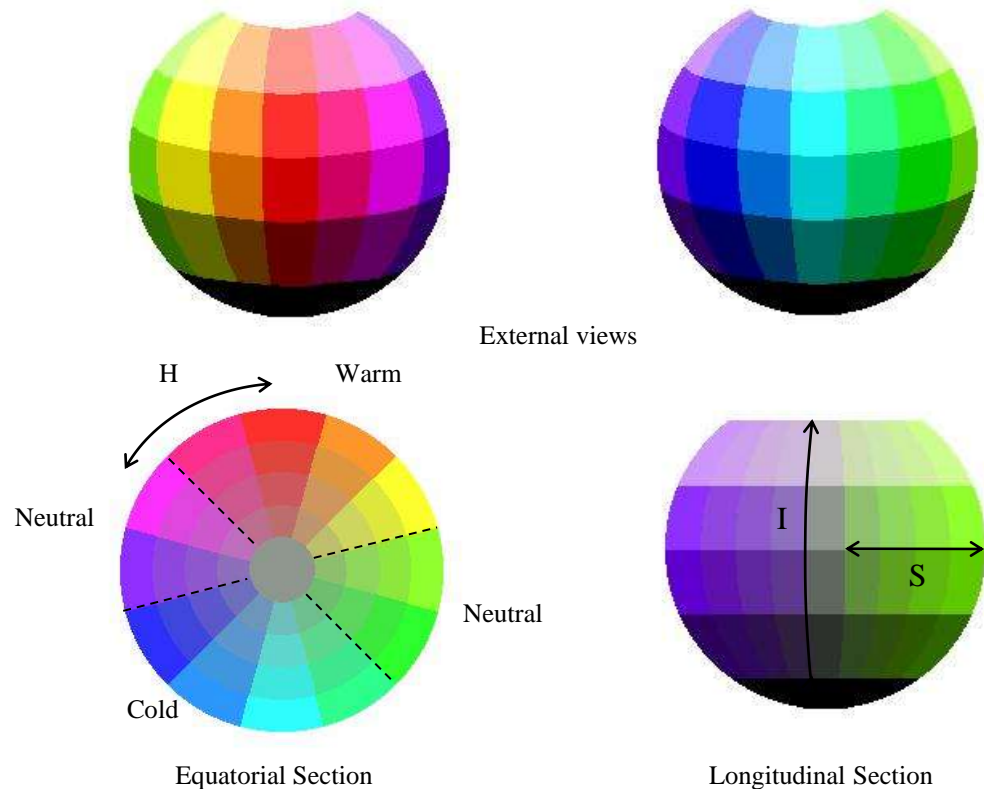
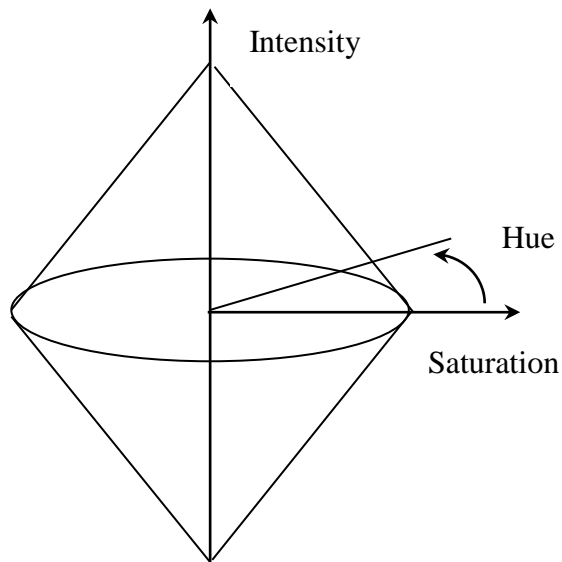


gray



# Color Models and HSI

- Many color models: RGB, CMY, YIQ, YUV, YCrCb, HSV, HSI, ...
- **HSI (Hue, Saturation, Intensity): often used**



# Similarity between Two Colors

The similarity between two colors, i and j, is given by:

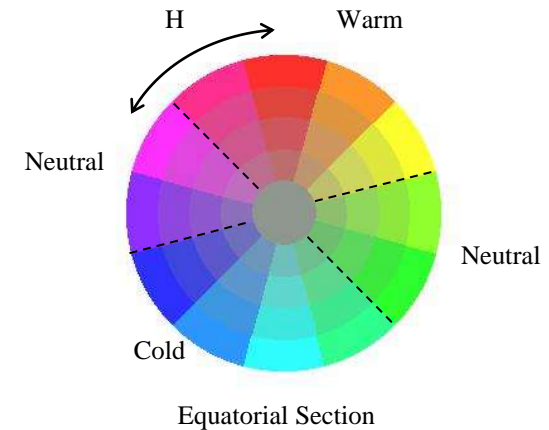
$$C(i, j) = W_h H(i, j) + W_s S(i, j) + W_i I(i, j)$$

where

$$H(i, j) = \min(|H_i - H_j|, 12 - |H_i - H_j|)$$

$$S(i, j) = |S_i - S_j|$$

$$I(i, j) = |I_i - I_j|$$



The degree of similarity between two colors, i and j, is given by:

$$CS(i, j) = \begin{cases} 0 & \text{if } H(i, j) > H_{\max} \\ 1 - \frac{C(i, j)}{C_{\max}} & \text{otherwise} \end{cases}$$



# Content Based Image Retrieval (CBIR)

- ❑ CBIR: based on similarity of image color, texture, object shape/position
- ❑ Images with similar color → *dominated by blue and green*



botanic1



CnScenery9



botanic3



raffles1



exar



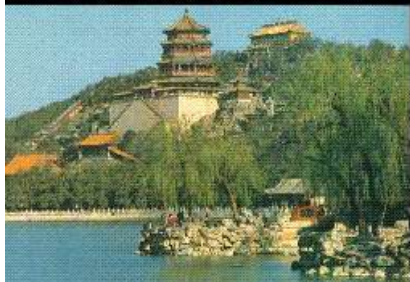
CnScenery7



shangrila



foothills



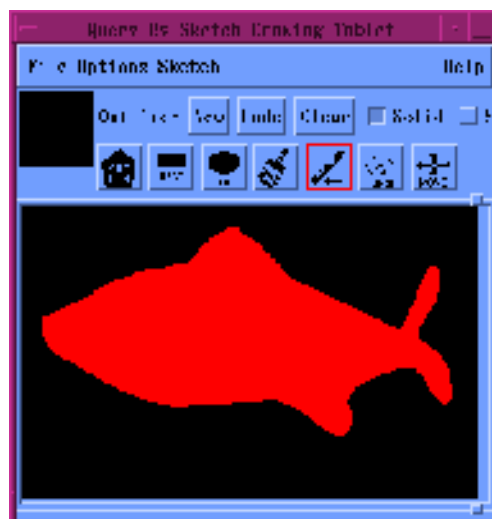
# Color Based Image Retrieval

Images with similar colors and distribution/histogram





# Shape Based Image Retrieval

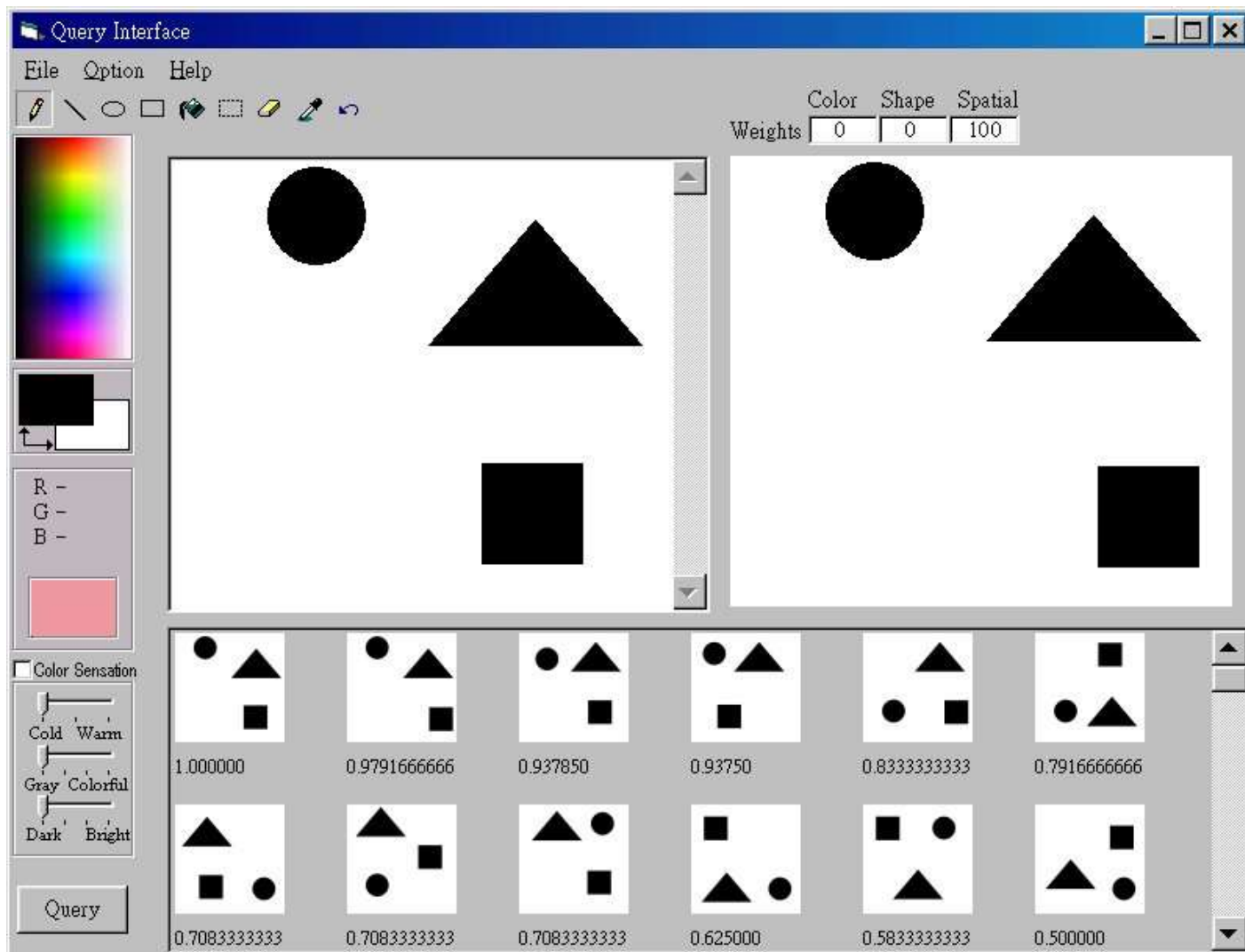


Images with similar shapes

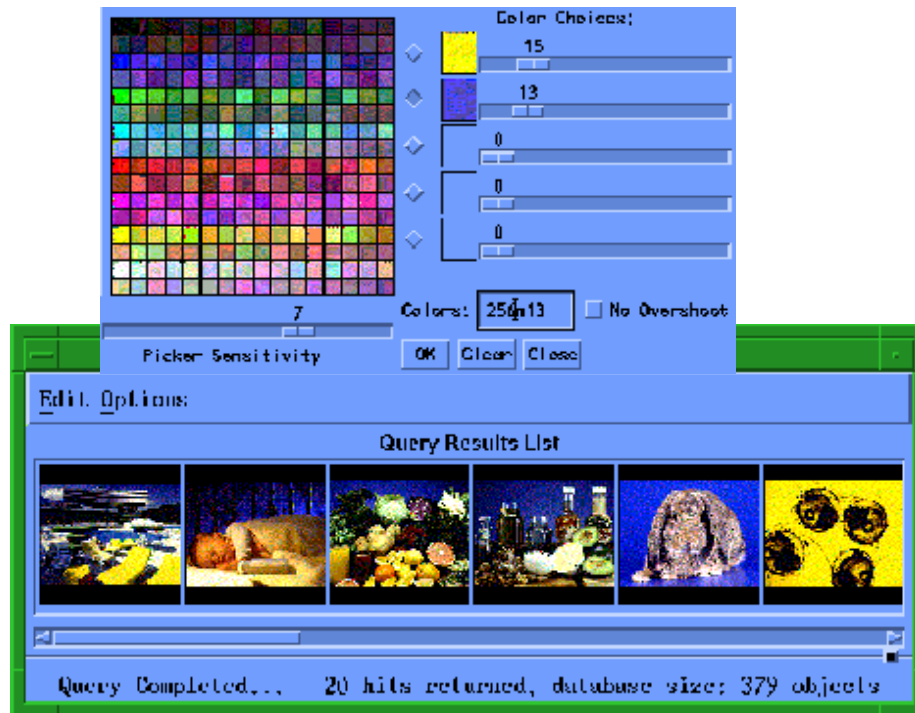


# Spatial Relation Based Image Retrieval

Images with similar shapes and their relation



# Correctness and Accuracy in CBIR



- ❑ CBIR accuracy is counted by a percentage of targeted/corrected image(s) in top-n candidate images, for example

$$\boxed{C_1, C_2, C_3, \dots, C_{n-1}, C_n} \quad C_{n+1}, \dots, C_M$$

90%

- ❑ Hybrid retrieval using color and texture plus shape can improve accuracy

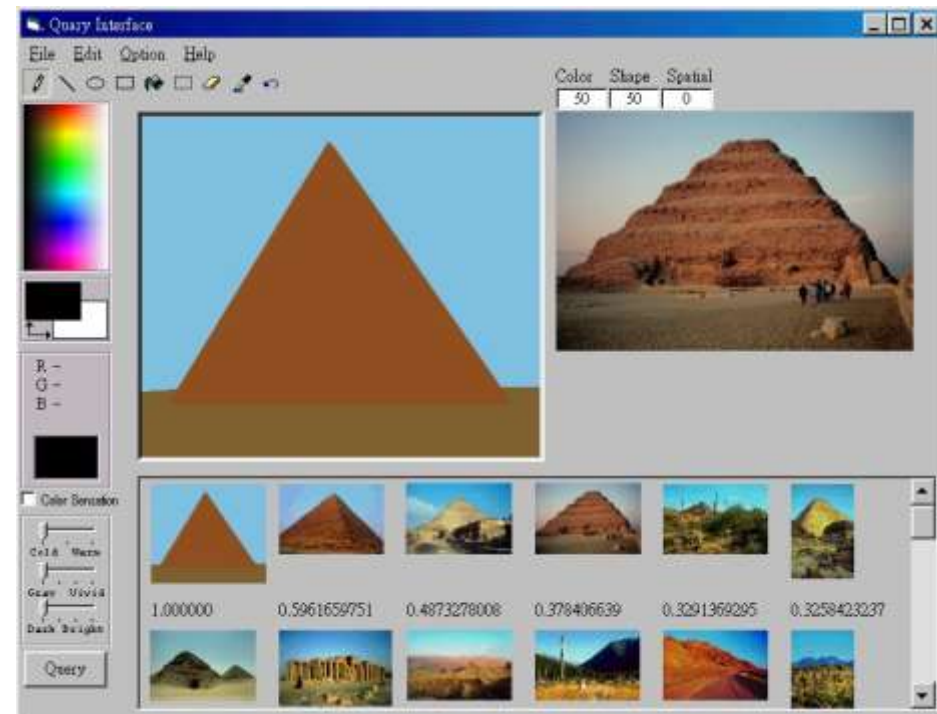
# Hybrid Retrieval – Combined Similarity

- ◆ The Similarity Measure of Color:  $CS$
  - ◆ The Similarity Measure of Shape:  $SS$
  - ◆ The Similarity Measure of Spatial Relation:  $SRS$
- **Combined Similarity Score:**

$$S = W_C * CS + W_S * SS + W_{SR} * SRS$$

Where  $CS$ ,  $SS$ ,  $SRS$  are the similarity scores of Color, Shape and Spatial Relations, and  $W_C$ ,  $W_S$ ,  $W_{SR}$  are the weights of Color, Shape and Spatial Relations

# Query by Scratch in CBIR



Please try such image search in the [Hermitage Web site](#). It uses the QBIC engine for searching archives of world-famous art.



# Query by Example in CBIR

Content-Based Image Reterival Sytem(CBIR)

Select Options | Search Options


Select the Image to be Searched on

☐ Random Browsing ☐ Upload Image

Number of Random Images: 24

Enter The full path of the Image:  Enter

Search On



Max no. of results: 40








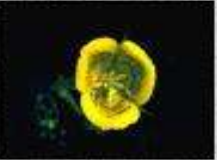
Search Image

Time Required: 0.487629 secs

Select Image to Search

First  
Second  
Third  
Fourth  
Fifth  
Sixth  
Seventh  
Eighth

Retrieved -> 10

1  2  3  4   
5  6  7  8 

Navigation: Left Arrow Right Arrow

# Query by Example in CBIR (cont.)



# Video Retrieval

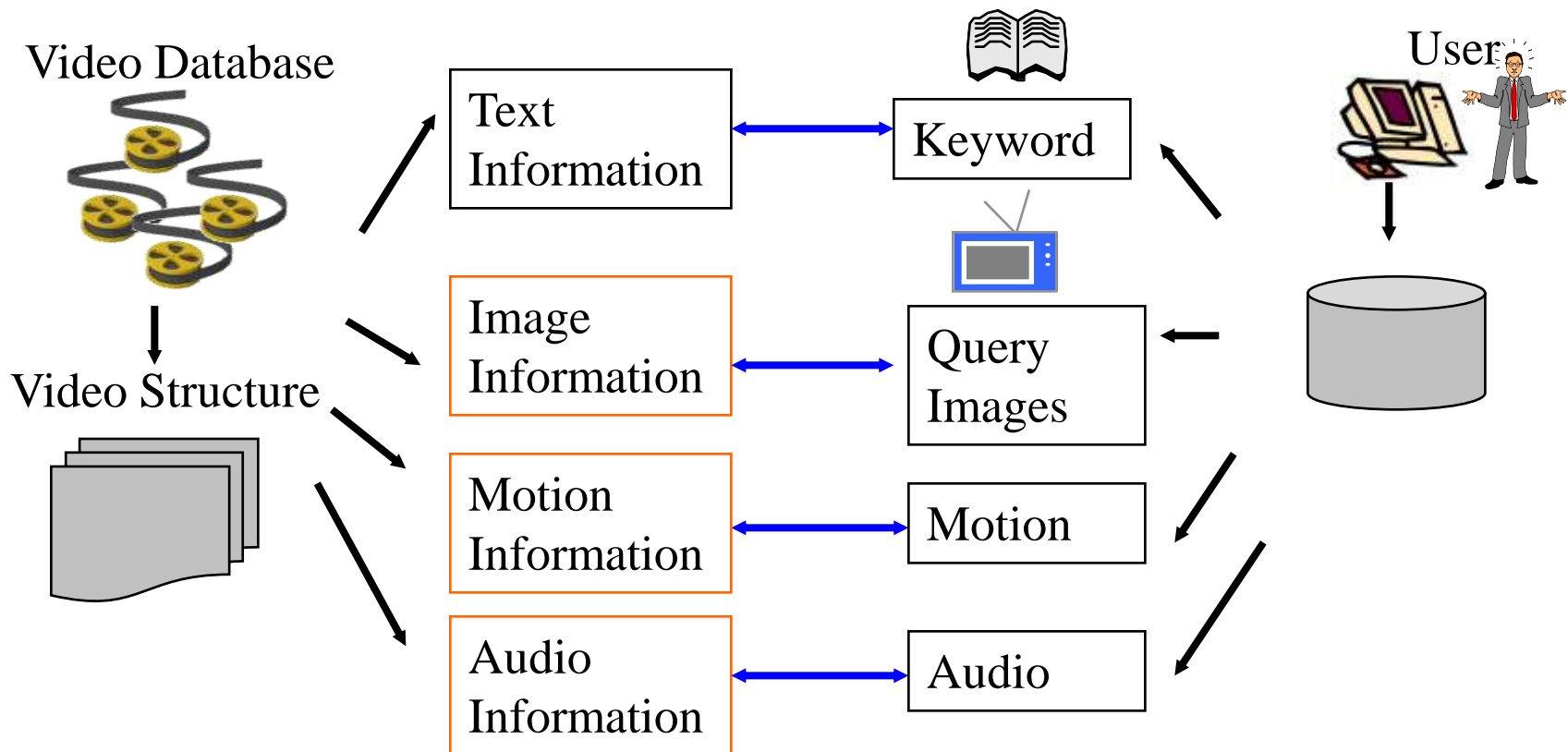
## ❑ Video retrieval:

- Find interesting video shots/segments from a movie, TV, video database
- It is hard because of many images ( $>10\text{fps}$ ) and temporal changes

## ❑ Methods of video retrieval

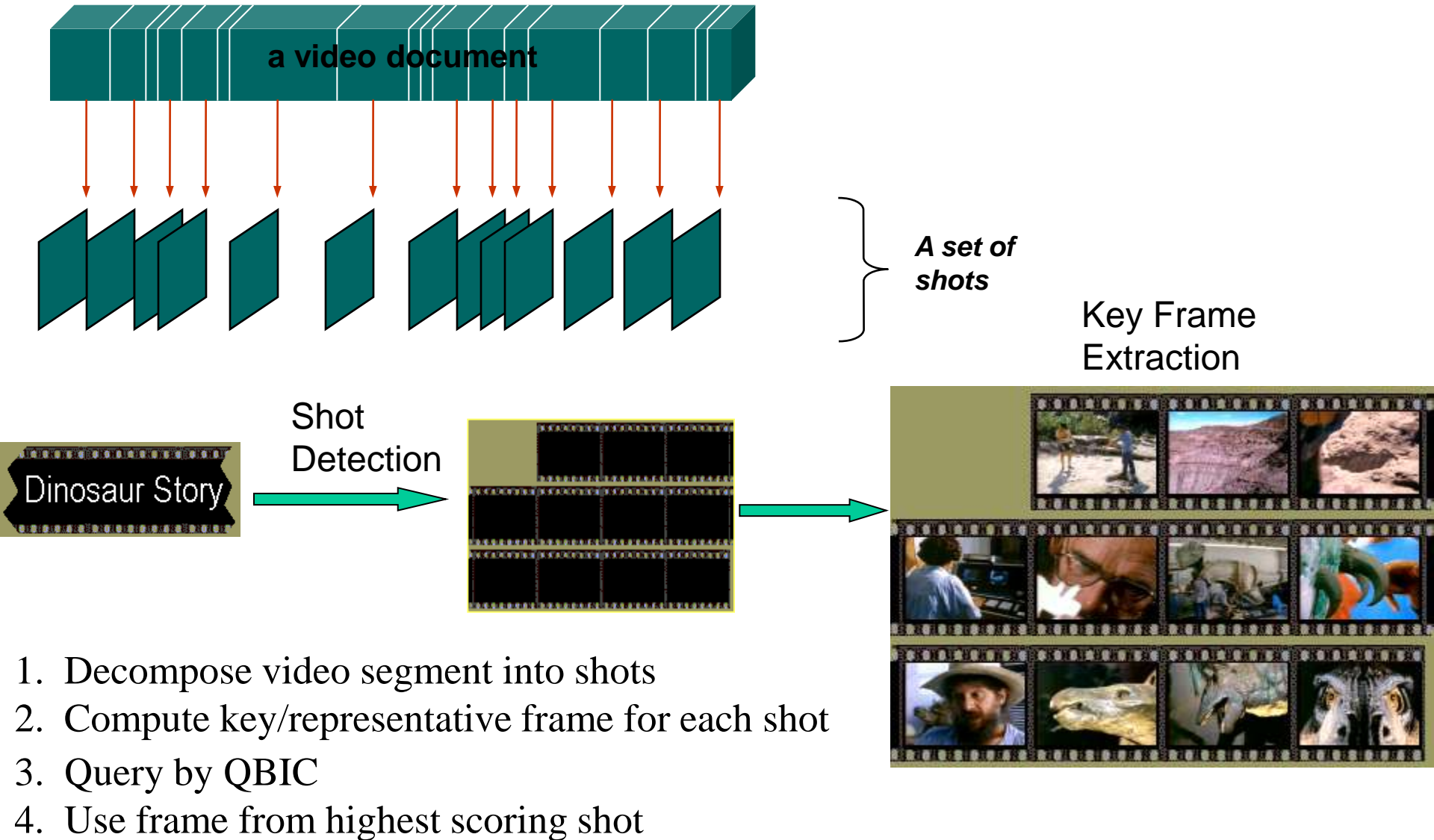
**Non-text-based:** Key frames via CBIR, color, object, background sound, etc.

**Text-based:** Extract caption, i.e., overlaid text, speech recognition, etc.

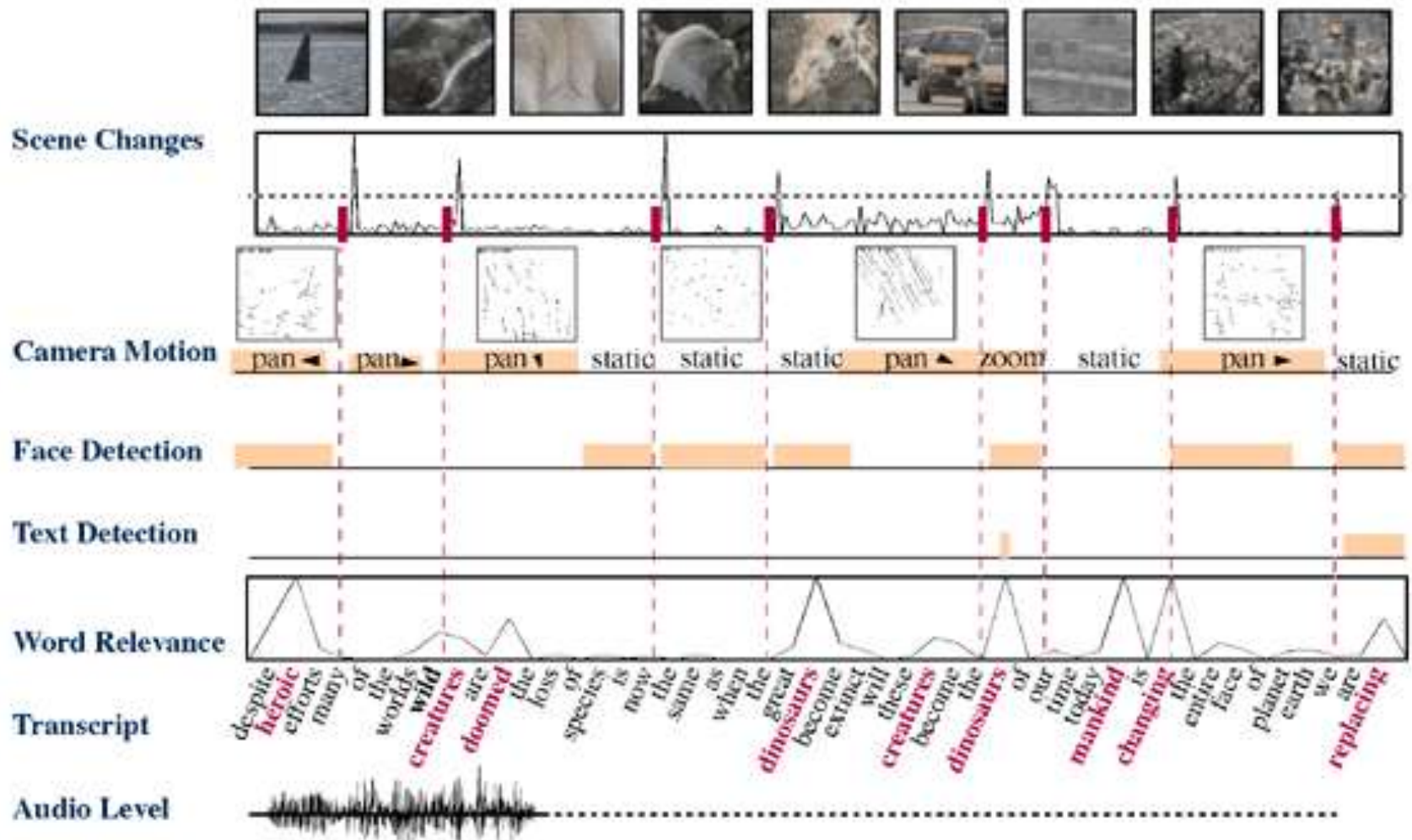




# Key Frame Extraction and Video Retrieval



# Various Clues/Contents in Video Retrieval



# Video Caption Extraction in Video Retrieval

Source Video:



Time-Based Minimum Image:



Final VOCR Results:

**FREEMAN  
BLOCK  
LOS  
ANGELES  
COUNT  
SHERIFF**

Text  
Region

SHERMAN BLOCK

Filtered  
Text

SHERMAN BLOCK

Binarized  
Segmented

SHERMAN BLOCK

OCR:

S H E R M A N B L O C K

Text  
Region

LOS ANGELES COUNTY SHERIFF

Filtered  
Text

LOS ANGELES COUNTY SHERIFF

Binarized  
Segmented

LOS ANGELES COUNT SHERIFF

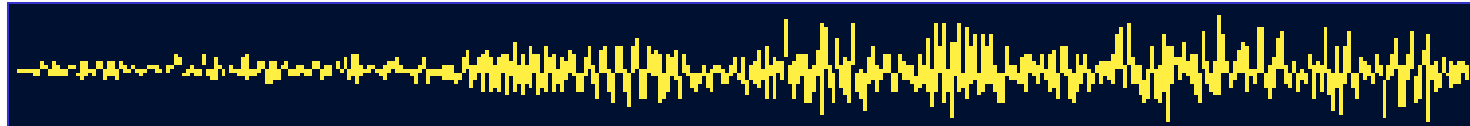
OCR:

L O S A N G E L E S C O U N T S H E R I F F

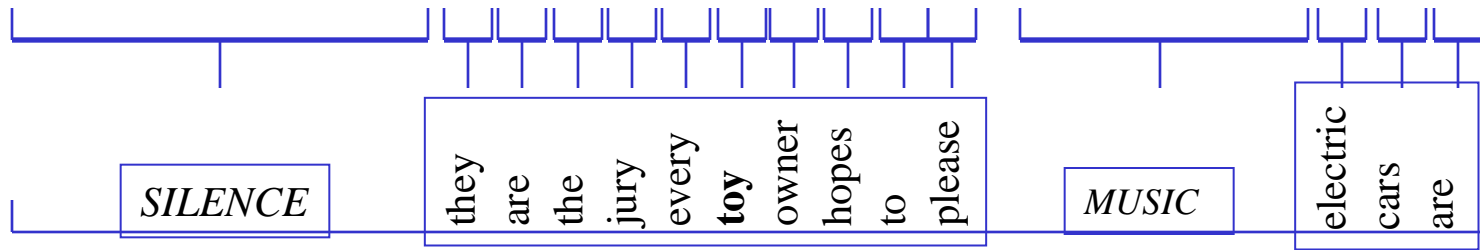
# Transcript via Speech Recognition for Video Retrieval

- Generates transcript to enable text-based retrieval from spoken language documents
- Improves text synchronization to audio/video in presence of scripts

Raw Audio



Text  
Extraction

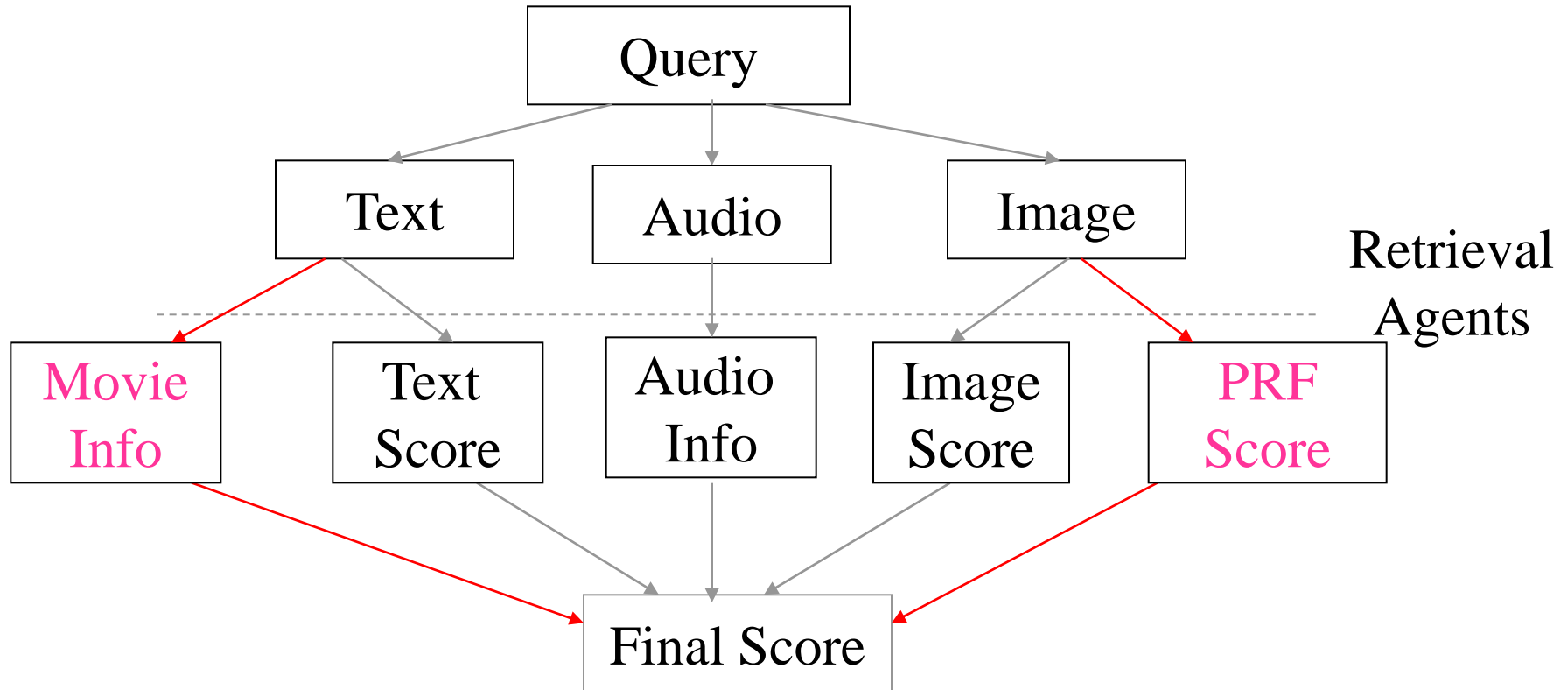


Raw Video

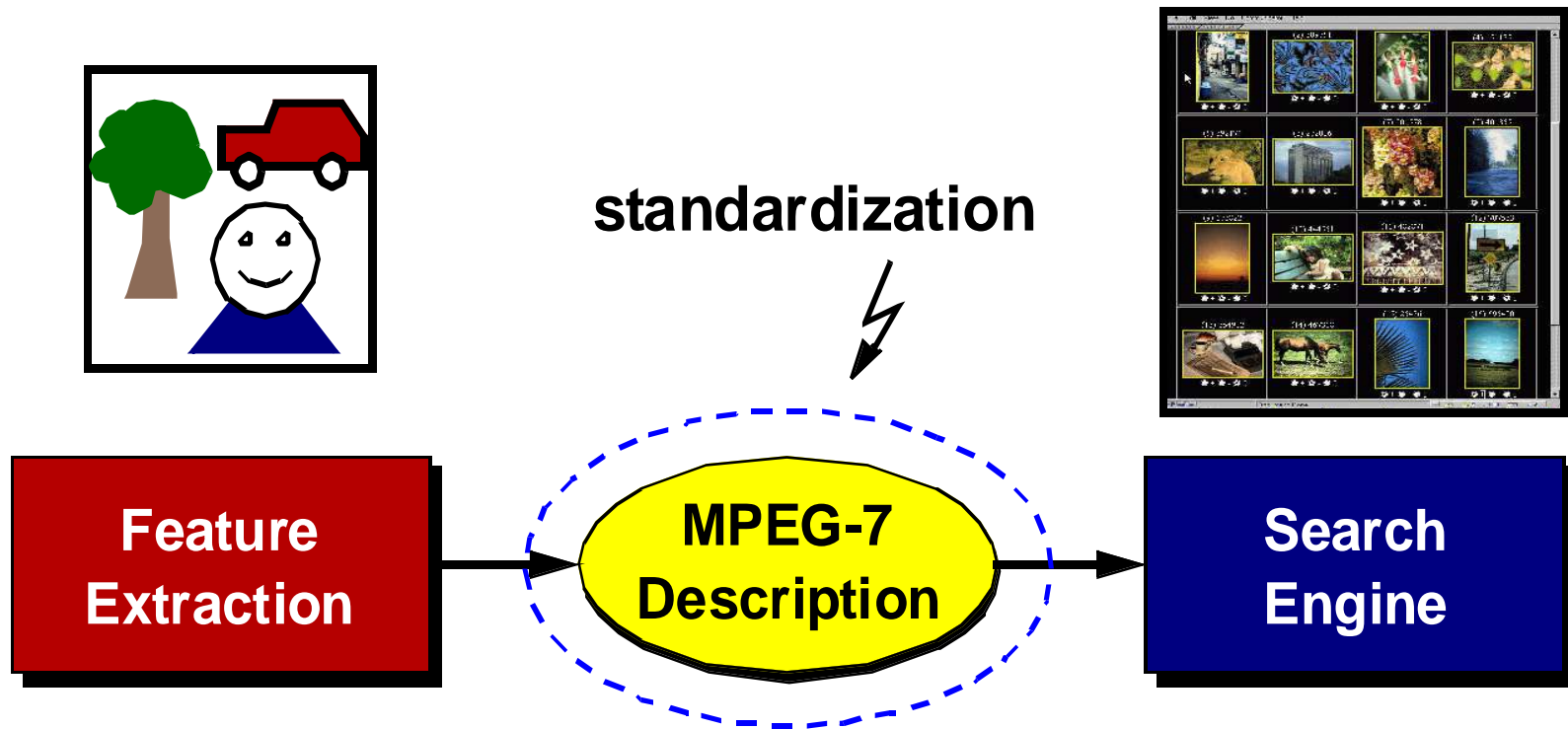




# Video Retrieval by Combining Different Features



# MPEG-7: Audiovisual Content Description



## Feature Extraction:

Content analysis (D, DS)  
Feature extraction (D, DS)  
Annotation tools (DS)  
Authoring (DS)

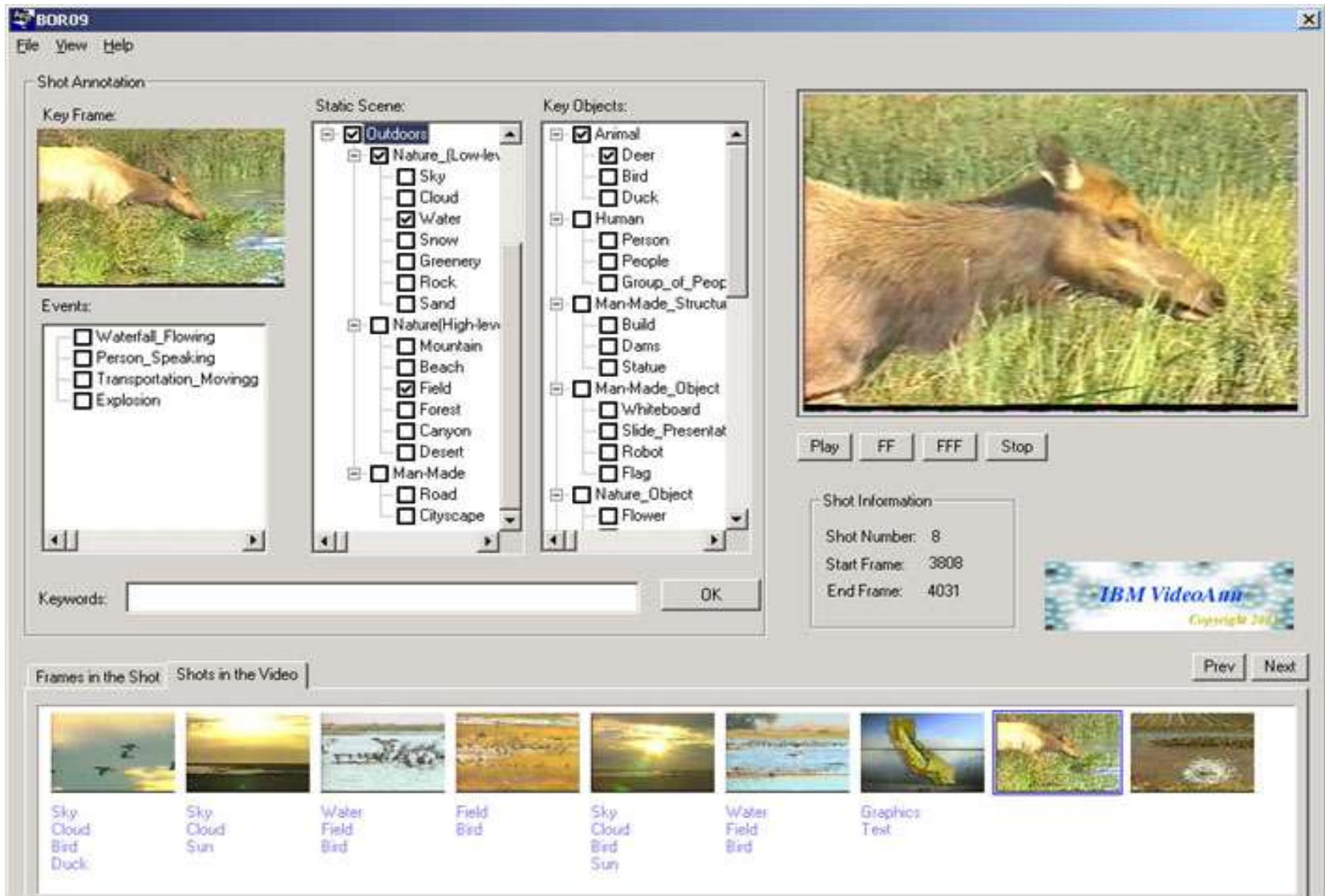
## MPEG-7 Scope:

Description Schemes (DSs)  
Descriptors (Ds)  
Language (DDL)  
Ref: MPEG-7 Concepts

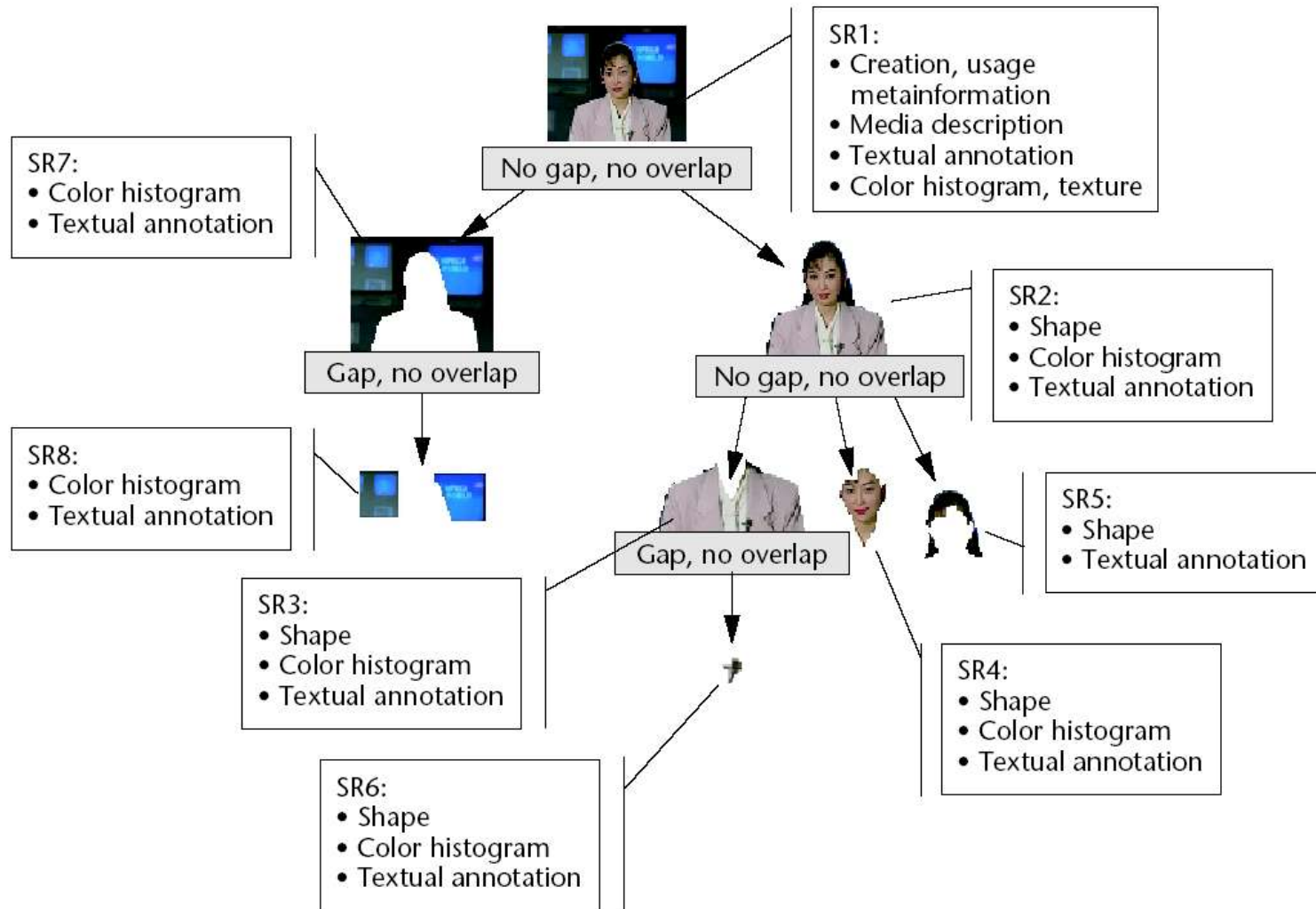
## Search Engine:

Searching & filtering  
Classification  
Manipulation  
Summarization Indexing

# Example of MPEG-7 Annotation Tool



# MPEG-7: Image Description Example





# Automatic Video Analysis and Index

Scene Cuts



Camera

Static

Static

Zoom

Objects

Adult Female

Animal

Two adults

Action

Head Motion

Left Motion

None

Captions

[None]

Yellowstone

[None]

Scenery

Indoor

Outdoor

Indoor

Time  
Axis

### Segment Tree

Shot1 Shot2 Shot3

Segment 1

Sub-segment 1

Sub-segment 2

Sub-segment 3

Sub-segment 4

segment 2

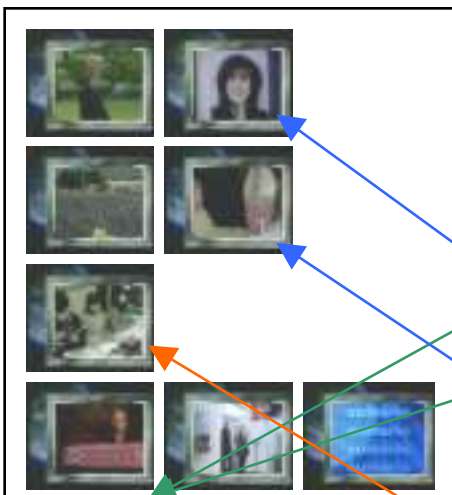
Segment 3

Segment 4

Segment 5

Segment 6

Segment 7



### Semantic DS (Events)

- Introduction
- Summary
- Program logo
- Studio
- Overview
- News Presenter
- News Items
- International
- Clinton Case
- Pope in Cuba
- National
- Twins
- Sports
- Closing

# Audio Retrieval

## ❑ Audio retrieval:

- Find required sound segment from audio database or broadcasting
- Find interesting music from song/music database or web

## ❑ Methods of audio retrieval

### Physical features of audio signal:

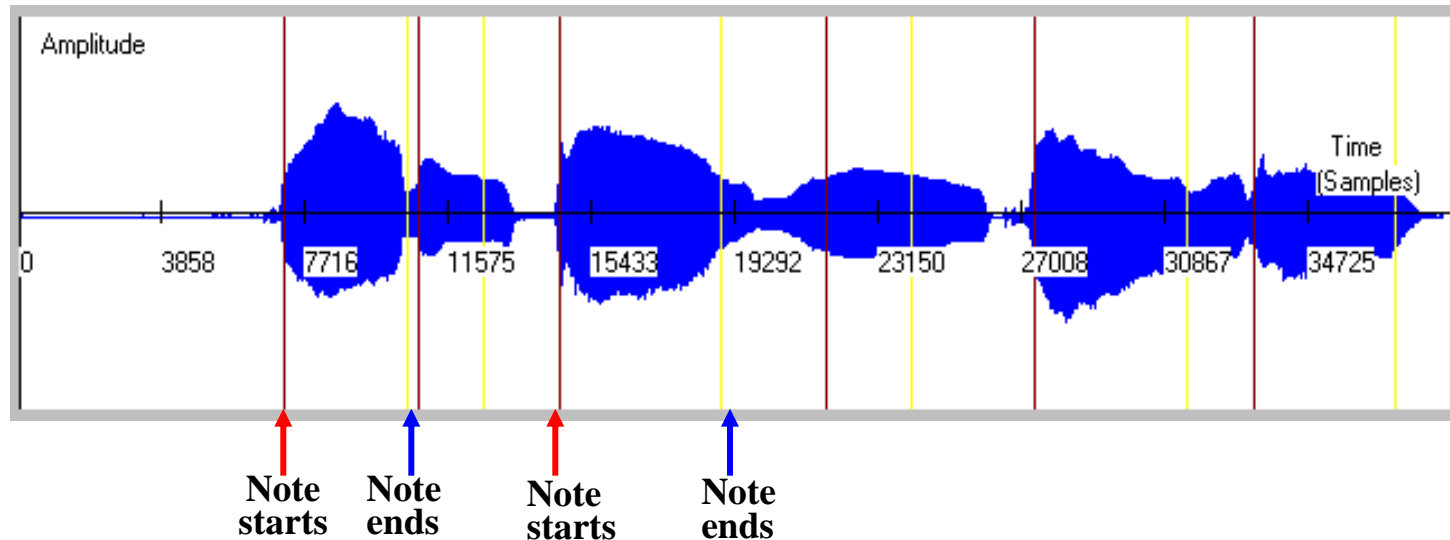
- Loudness, i.e., sound intensity (0~120dB)
- Frequency range: low, middle or high (20Hz~20KHz)
- Change of acoustic feature
- Speech, background sound, and noise
- Pitch

### Semantic features of audio:

- word or sentence via speech recognition
- Male/female, young/old
- Rhythm and melody
- Audio description/index
- Content Based Music Retrieval (CBMR)

# Music Retrieval by Singing/humming

## Happy Birthday

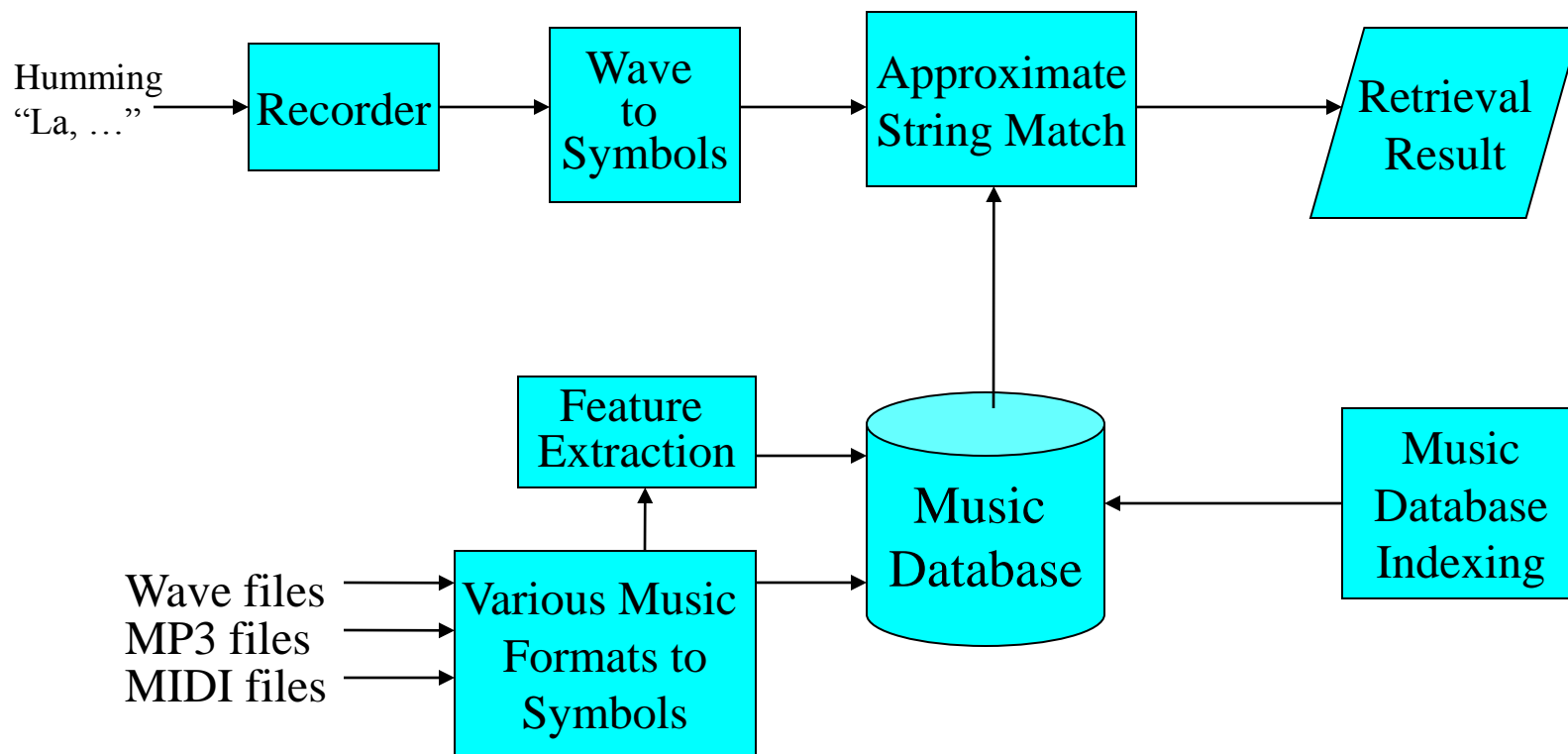


- A note has two important attributes
  - Pitch: It tells people which tone to play
  - Duration: It tells people how long a note needs to be played
  - Notes are represented by symbols

Staff {

Note name	C	D	E	F	G	A	B	C
Note pitch	Do	Re	Mi	Fa	So	La	Si	Do

# Music Retrieval by Singing/humming (Cont.)



# Demos of Content-Based Image Retrieval

# Media Distribution Across Internet

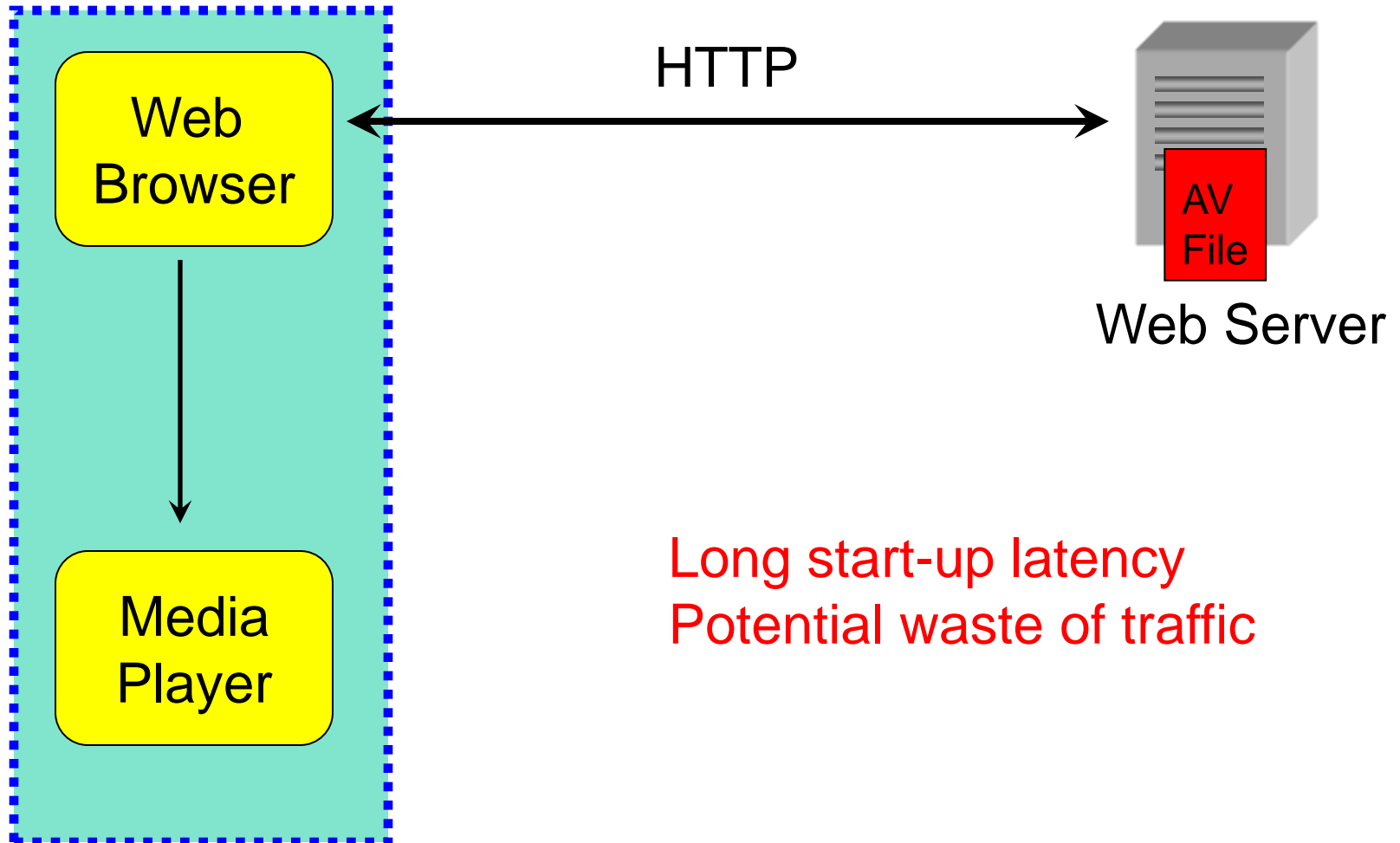
- Media Distribution Category
- Media Streaming
- Streamed Media On Demand Delivery
- Streamed Media Internet Broadcast
- Streamed Media Server and Client/Player
- Streaming Service System
- RTSP (Real Time Stream Protocol)
- RTP (Real-time Transport Protocol)

# Media Distribution Catalog

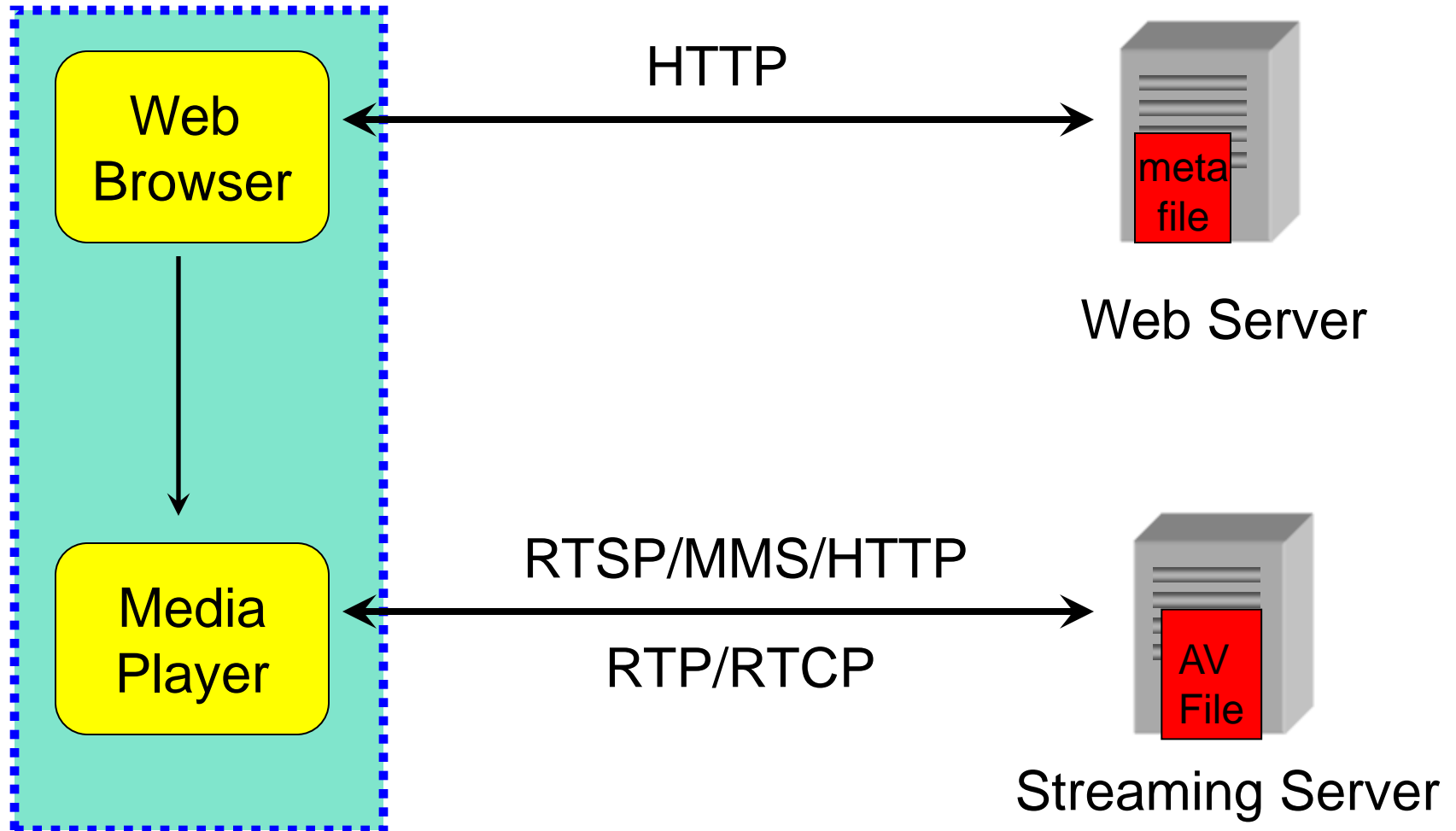
- Media distribution - Deliver media contents to users
- ✧ Delivery via disc:
  - Merits: Large storage, high audiovisual quality
  - Demerits: long delivery time, inflexible
- ✧ Delivery via Internet:
  - Non realtime delivery:
    - Called download service:
      - >download **all** data, save to disc, and play
    - Using data file transfer protocols like ftp and http via ftp or web server
  - Realtime delivery:
    - Called streaming service:
      - >download & play simultaneously, partial data in buffer, no data in disc
    - May use http and web server to provide limited streaming service
    - Often use RTSP/RTP and media server for rich streaming service



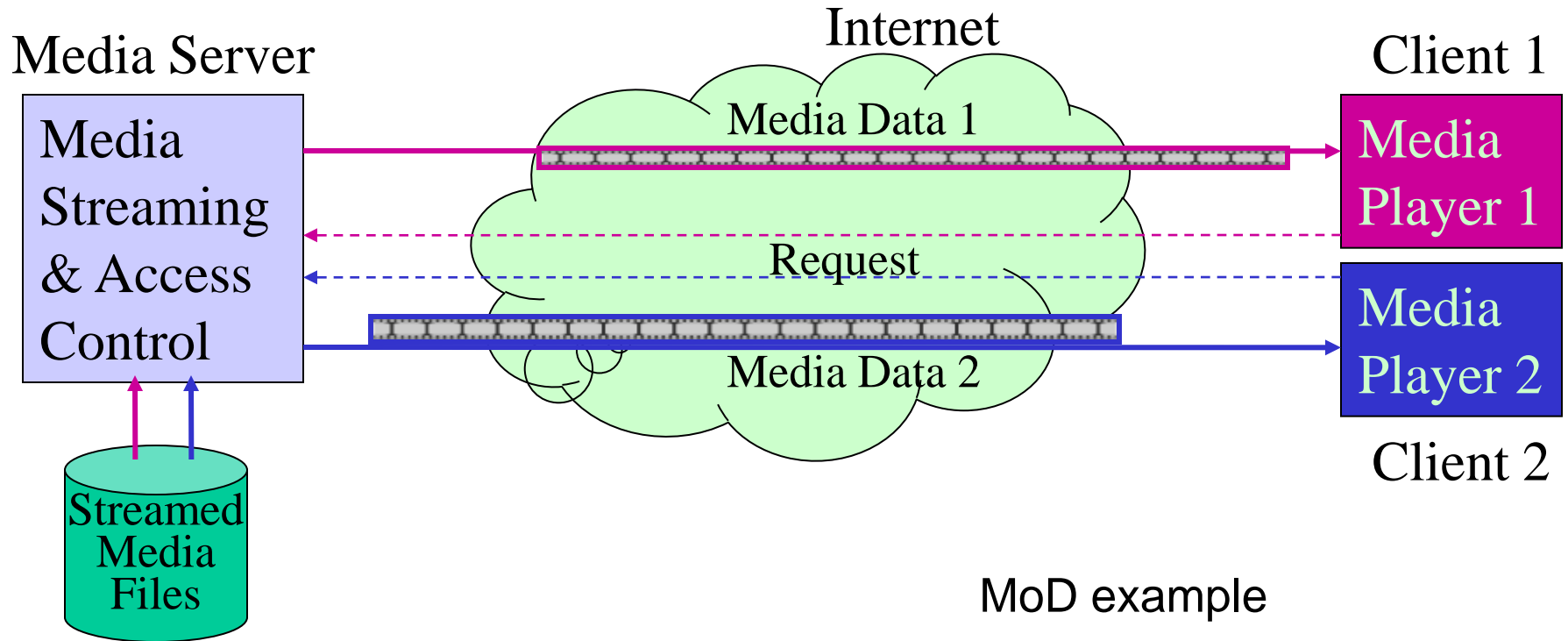
# Non Realtime Delivery: Download service



# Realtime Delivery: Stream Service

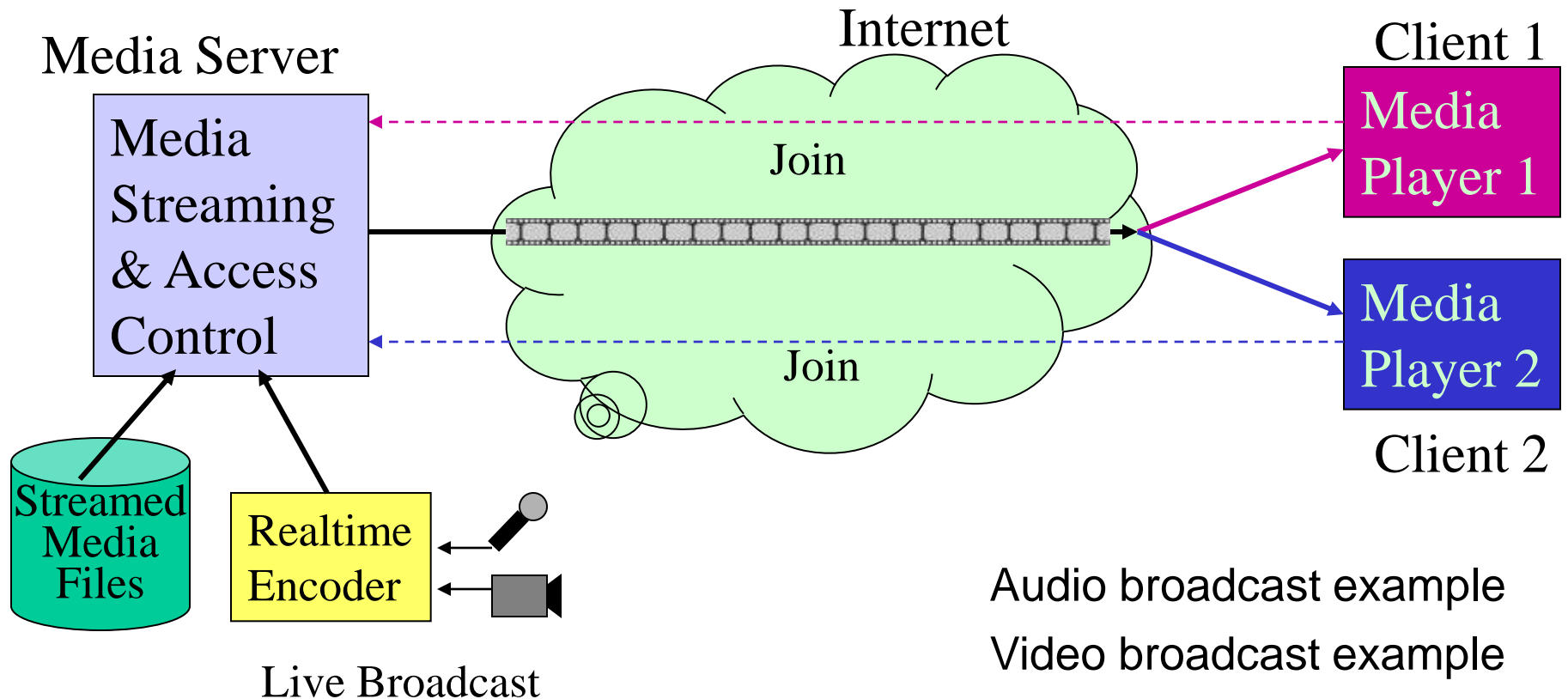


# Streamed Media On Demand Delivery



- Media on demand (MoD)
  - Streamed media are saved in media server as streamed file format
  - Clients, i.e., media player, access media contents independently
  - Media content is played from the file beginning for each client's request
  - User can control playing, such fast forward, pause, ...
  - Like rent a video tape or DVD and replay it in your cassette/DVD palyer

# Streamed Media Broadcast



- **Media Internet Broadcast (MIB) or Webcast**
  - Media may be stored in server or captured lively and encoded in realtime
  - Clients can join a broadcast and same media content goes to all clients
  - Users watch/listen the broadcast from the current state not from beginning
  - Users can't control its playing such fast forward, stop, etc.
  - Like conventional radio and TV broadcast

# Streaming Media Service History

## 1992

- MBone
- RTP version 1
- Audiocast of 23<sup>rd</sup> IETF mtg

## 1994

- Rolling Stones concert on MBone

## 1995

- ITU-T Recommendation H.263
- RealAudio launched

## 1996

- Vivo launches VivoActive
- Microsoft announces NetShow
- RTSP draft submitted to IETF

## 1997

- RealVideo launched
- Microsoft buys Vxtreme
- Netshow 2.0 released
- RealSystem 5.0 released
- RealNetworks IPO

## 1998

- RealNetworks buys Vivo
- Apple announces QuickTime Streaming
- RealSystem G2 introduced

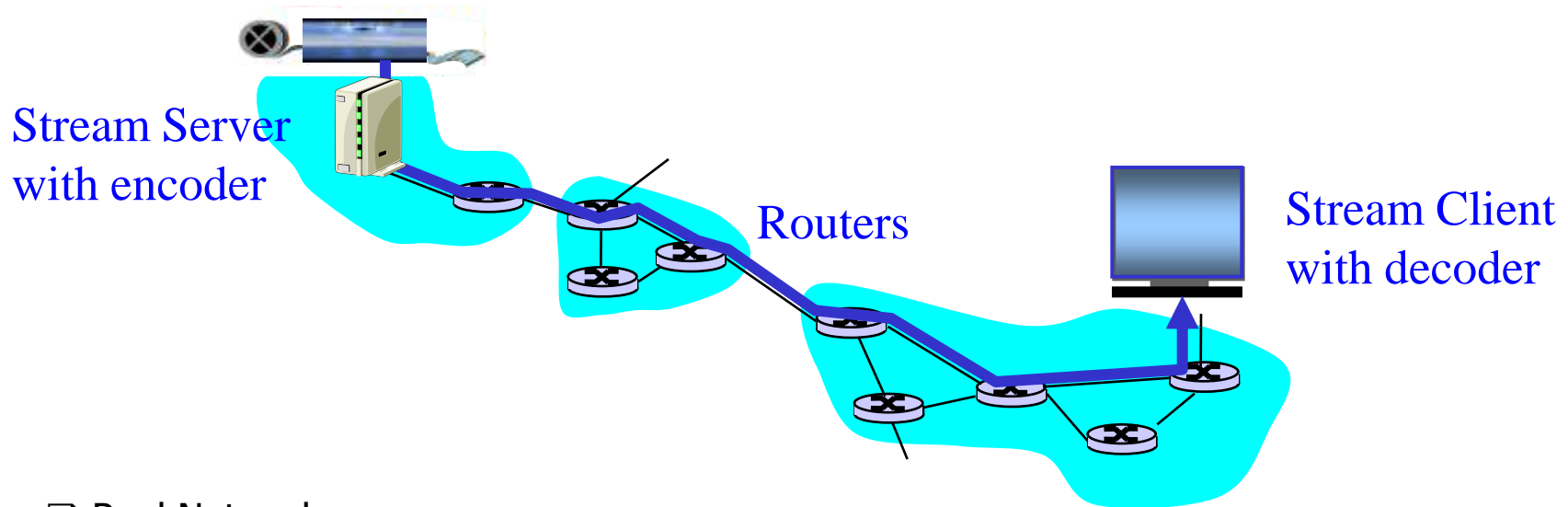
## 1999

- RealNetworks buys Xing
- Yahoo buys Broadcast.com for \$ 5.7B
- Netshow becomes WindowsMedia

## 2000

- RealPlayer reaches 100 million users
- Akamai buys InterVu for \$2.8B
- *Internet stock market bubble bursts*
- WindowsMedia 7.0
- RealSystem 8.0

# Popular Stream Media Server and Player



## ☐ Real Networks

- Real Producer: create streamed media file, end with "filename.rm"
- Real Server: streaming media to delivery across network
- Real Player: streamed media player in RM format

## ☐ Windows Multimedia Technologies

- Media Encoder: create streamed media file, end with "filename.asf/.wmv"
- Media Server: streaming media to delivery across network
- Media Player: streamed media player in ASF/WMV format

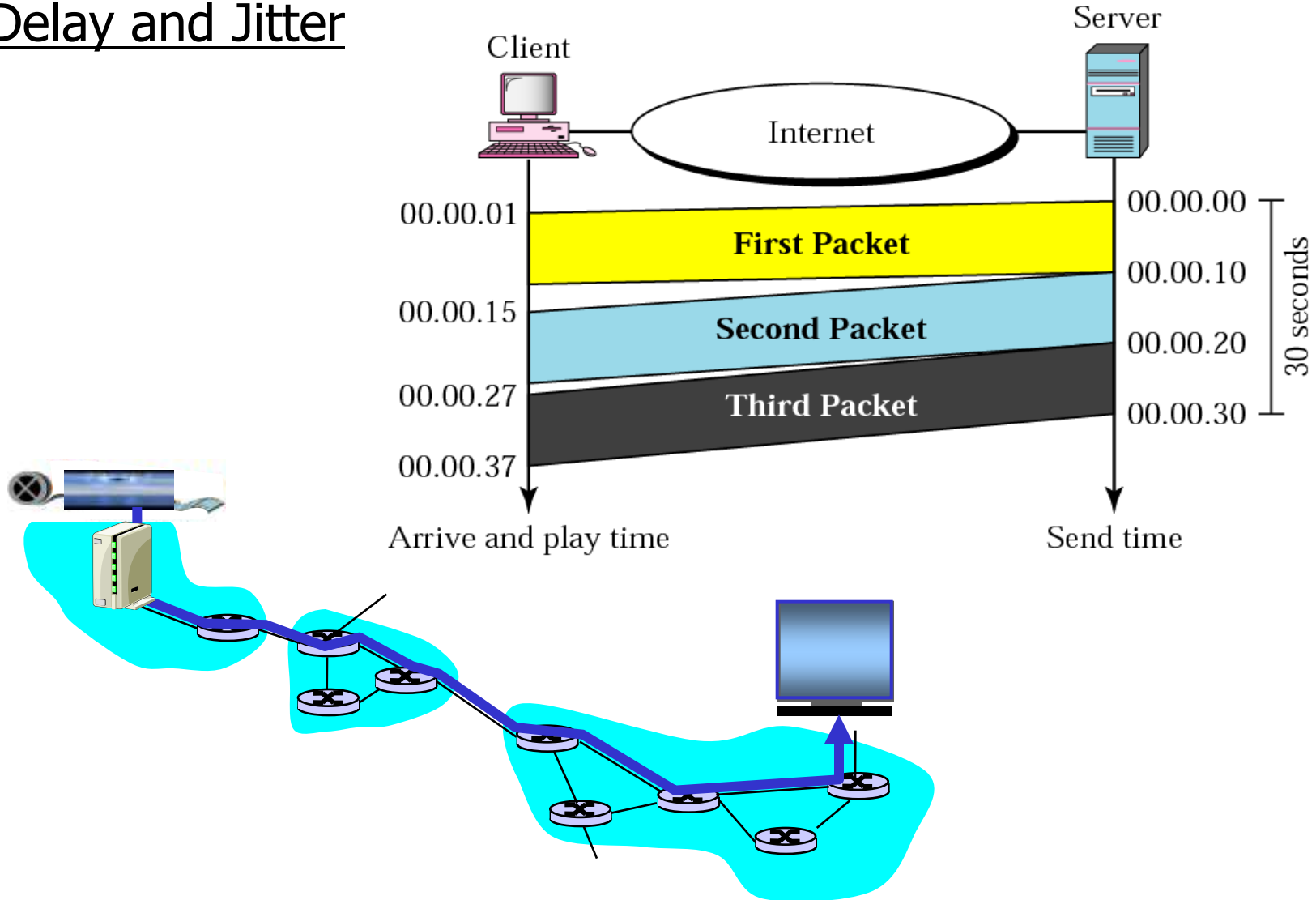
## ☐ QuickTime

- QuickTime Pro: create streamed media file, end with "filename.qt"
- QuickTime Streaming Server (Mac) and Darwin Streaming Server
- QuickTime Player: streamed media player in QT format

## ☐ Audio/MP3: Liquid Audio, SHOUTcast, icecast

# Key Points in Streaming Media Service

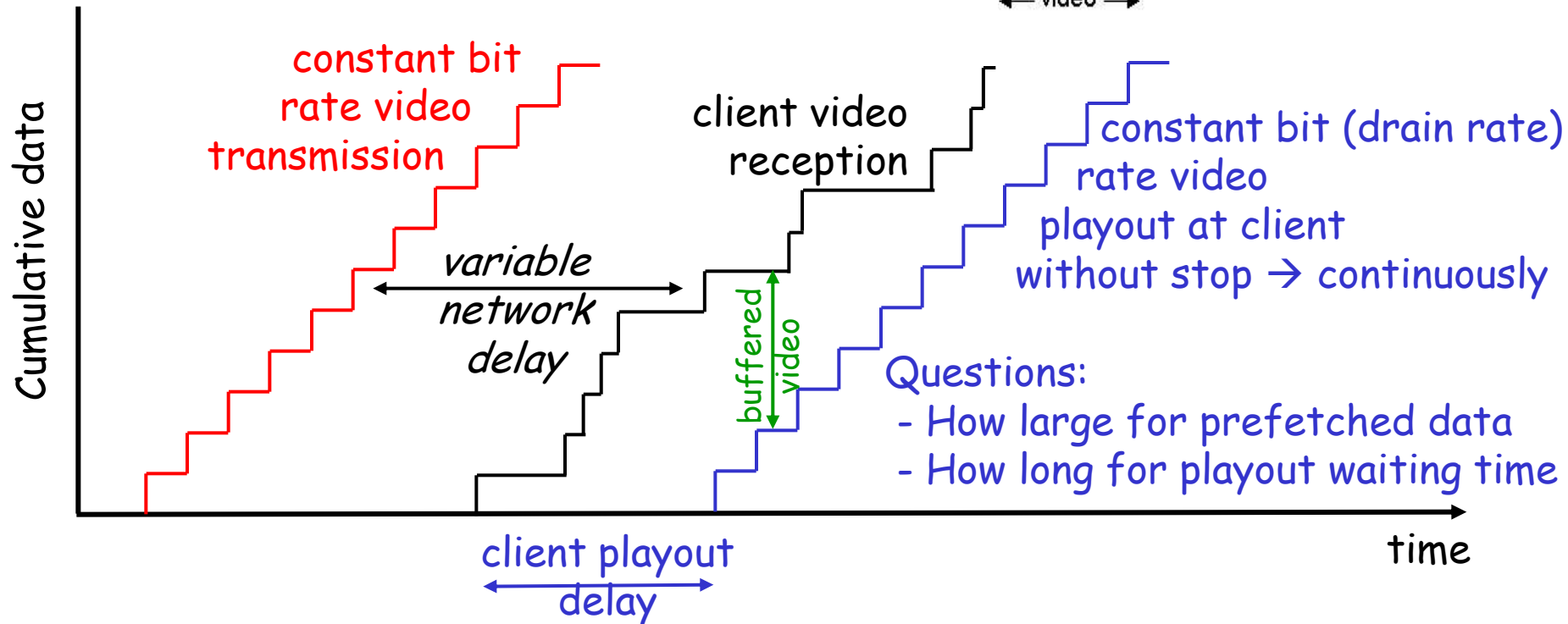
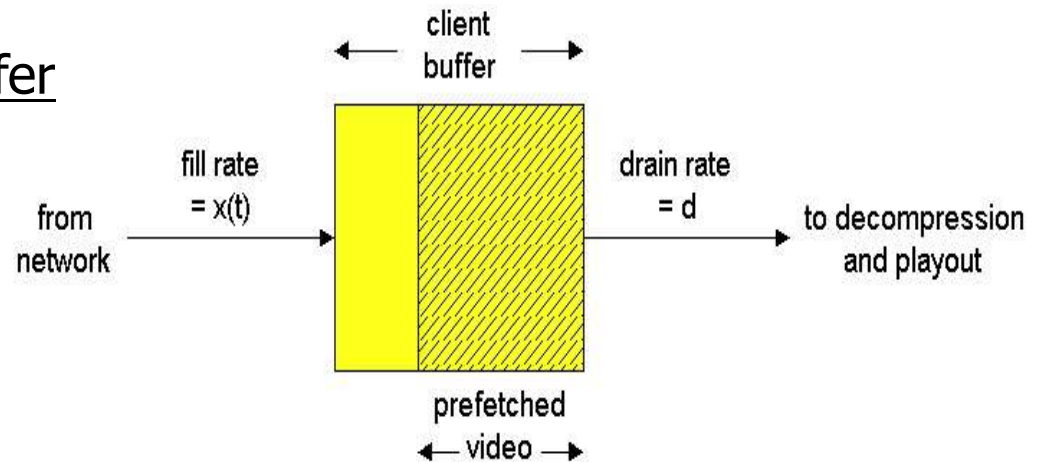
## ☐ Delay and Jitter



# Key Points in Streaming Media Service (Cont)

## □ Smooth Dealy & Jitter via buffer

- \* Client-side buffering,
- \* Playout delay,
- \* Compensate for network delay & jitter

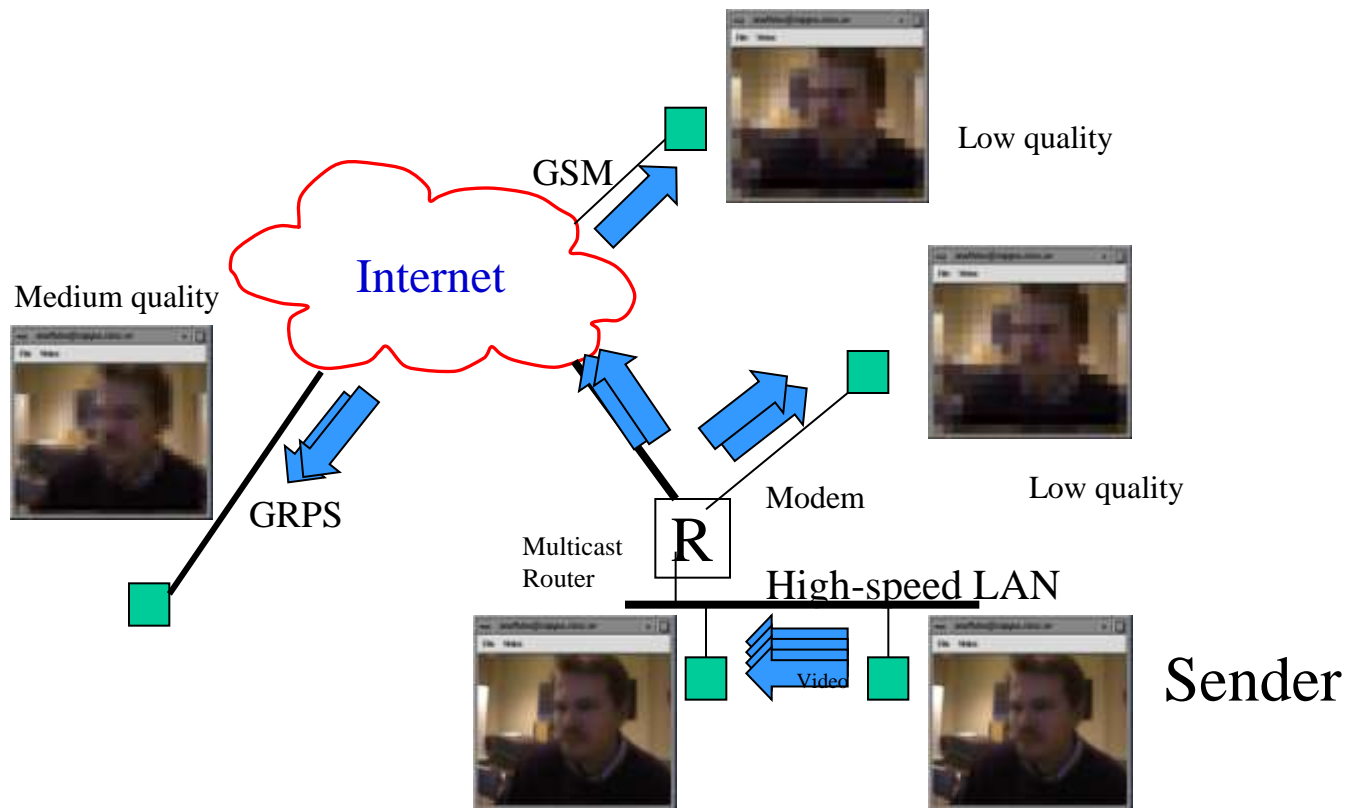




## Key Points in Streaming Media Service (Cont)

- ❑ Trade-off between media quality and network bandwidth

- Data amount of continuous media, especially video, is extremely large
- Current Internet bandwidth is relative small, 28K/56K modem, ADSL, Cable, LAN, etc.
- Before delivery, clarify targeted users and their available bandwidth

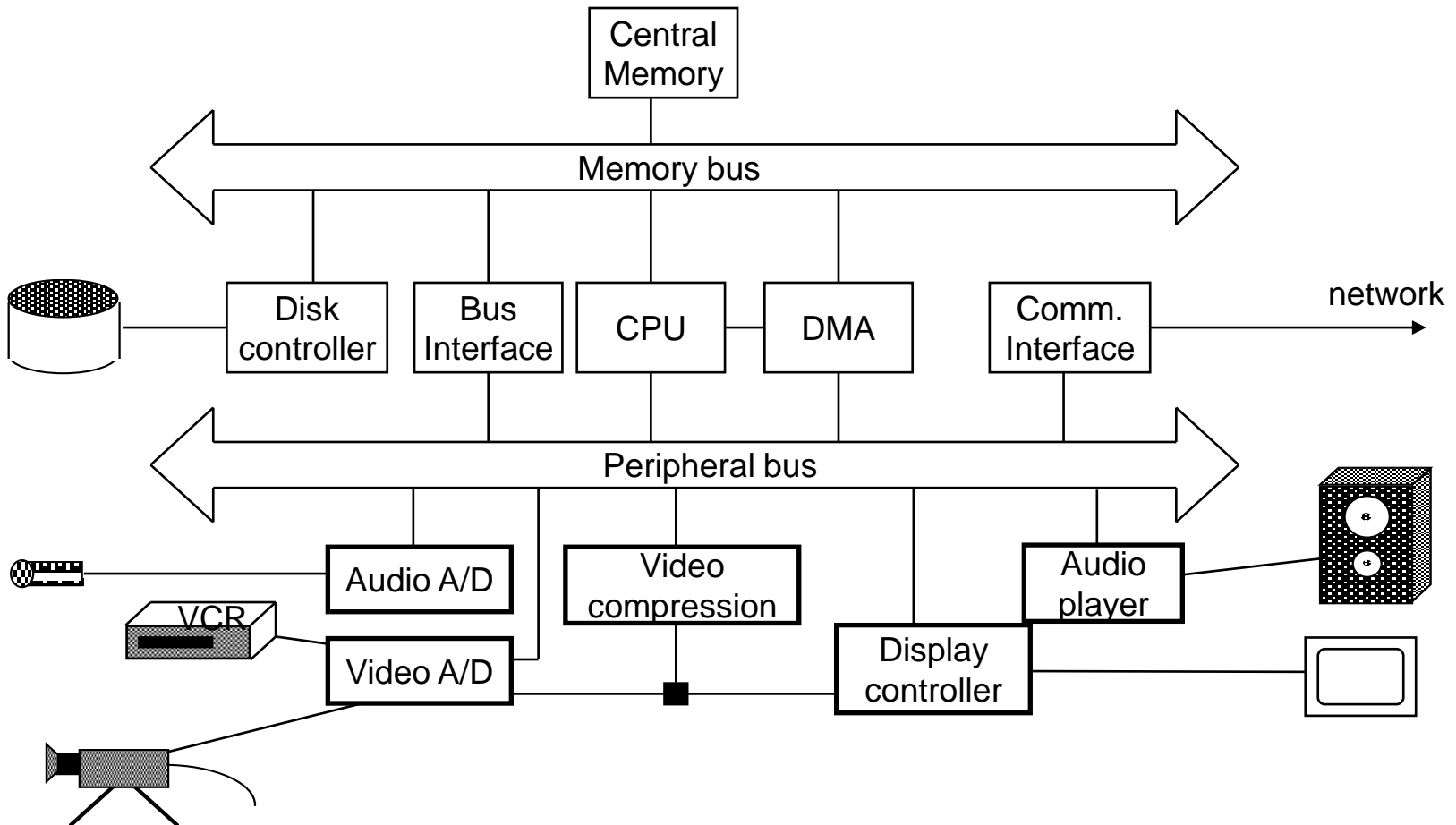


# Key Points in Streaming Media Service (cont)

## ❑ Limited server resource

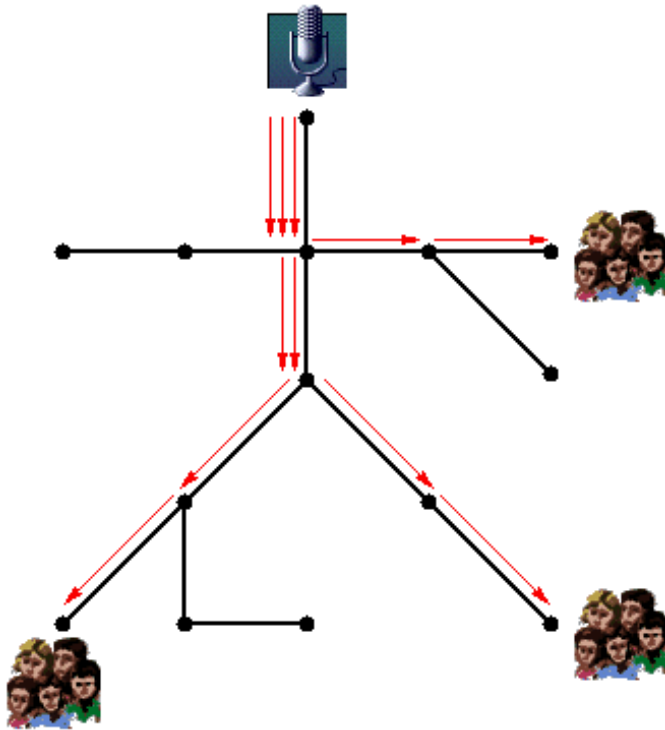
- Limited computational power in processing many media streams
- Limited storage space in saving many media data in server
- Limited IO performance in outputting many streams to networks

→ How to serve many users simultaneously ?

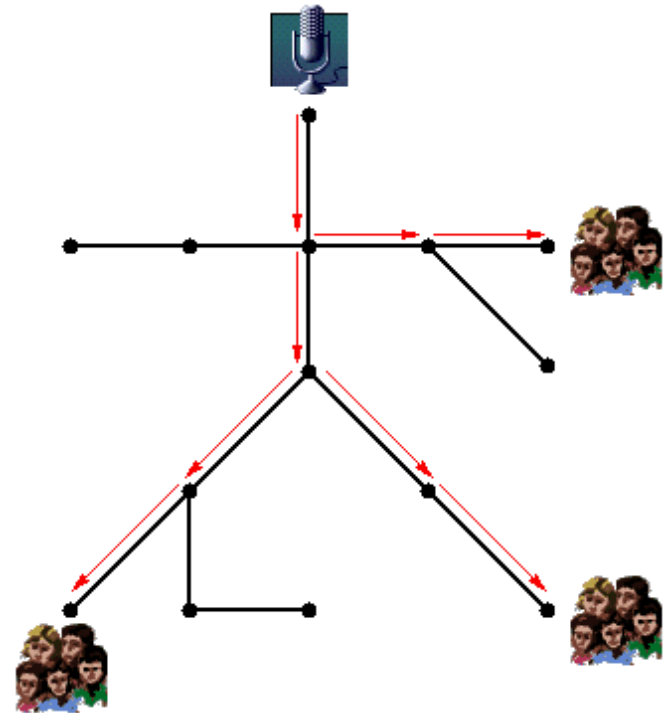


# Key Points in Streaming Media Service (Cont)

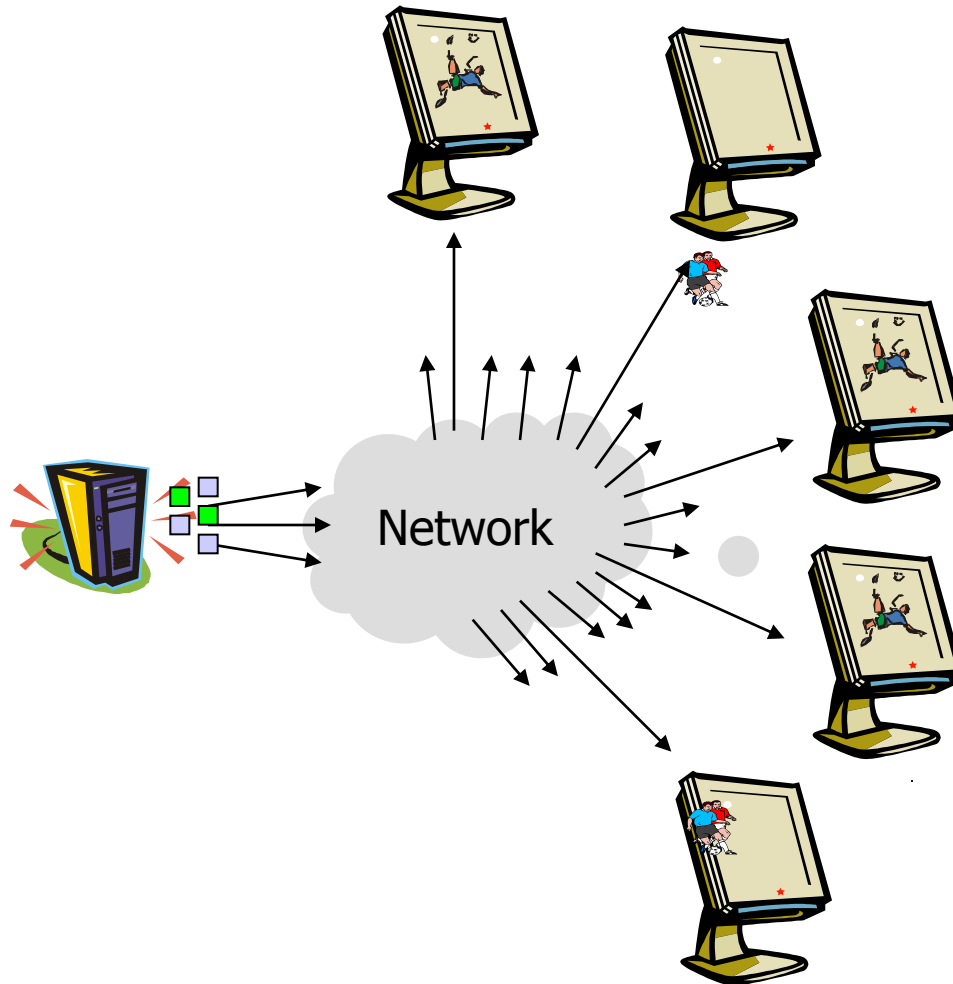
## □ Unicast



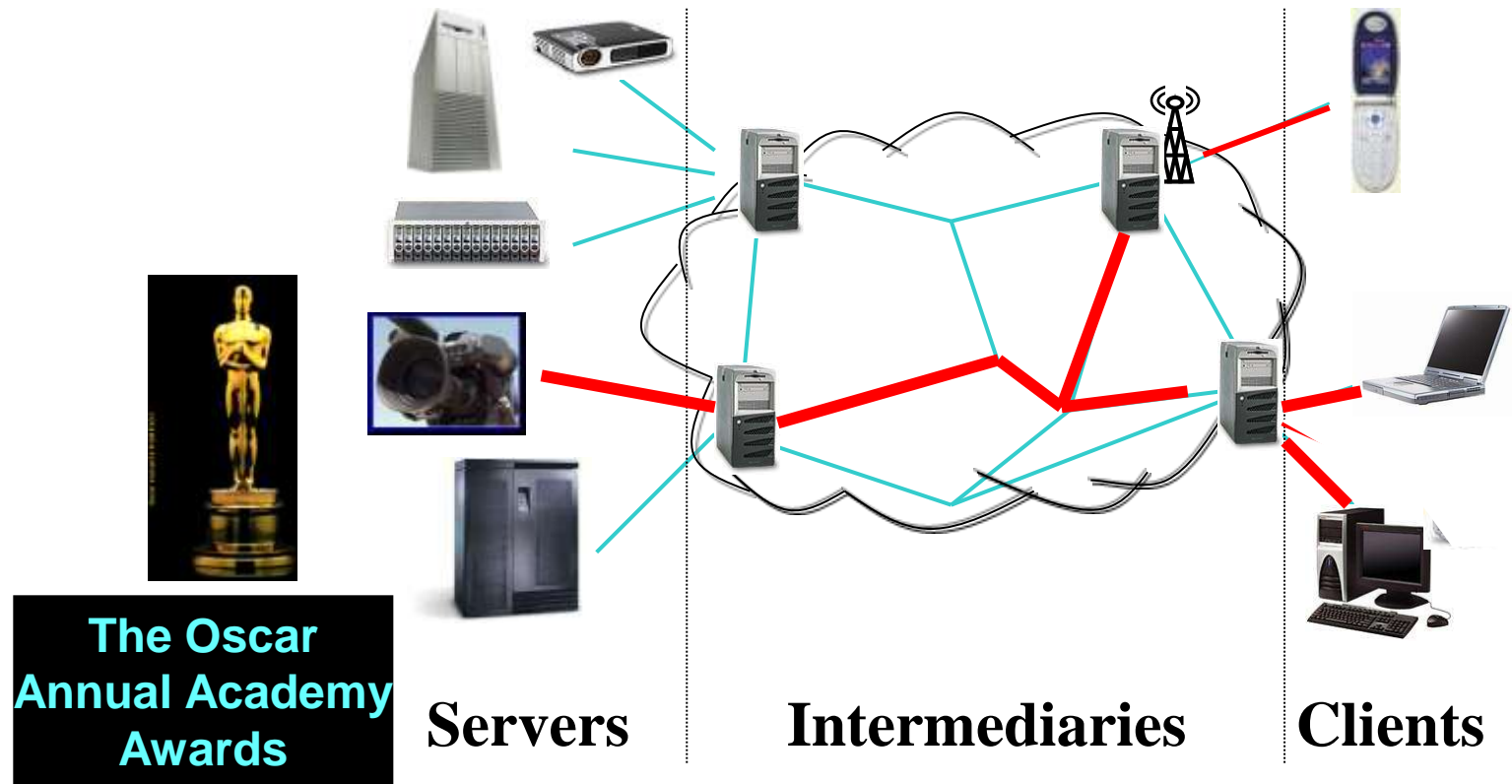
## □ Multicast



# Unicast Example: Multiple Independent Streams



# Multicast Example: Single Stream and Copy



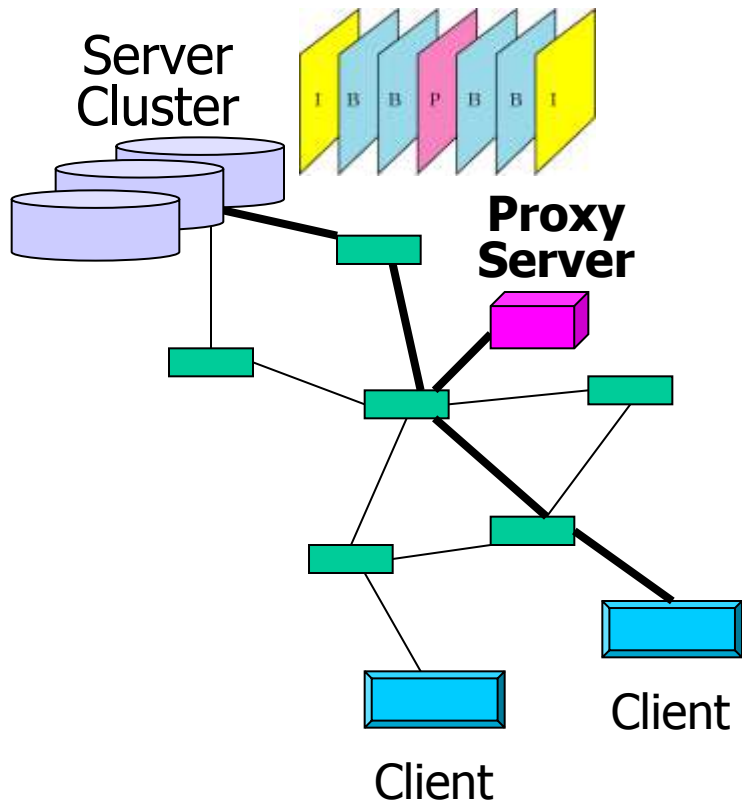
# Key Points in Streaming Media Service (Cont)

## ❑ Cache technology

- Increase IO via putting media data in memory
- The larger memory, the better

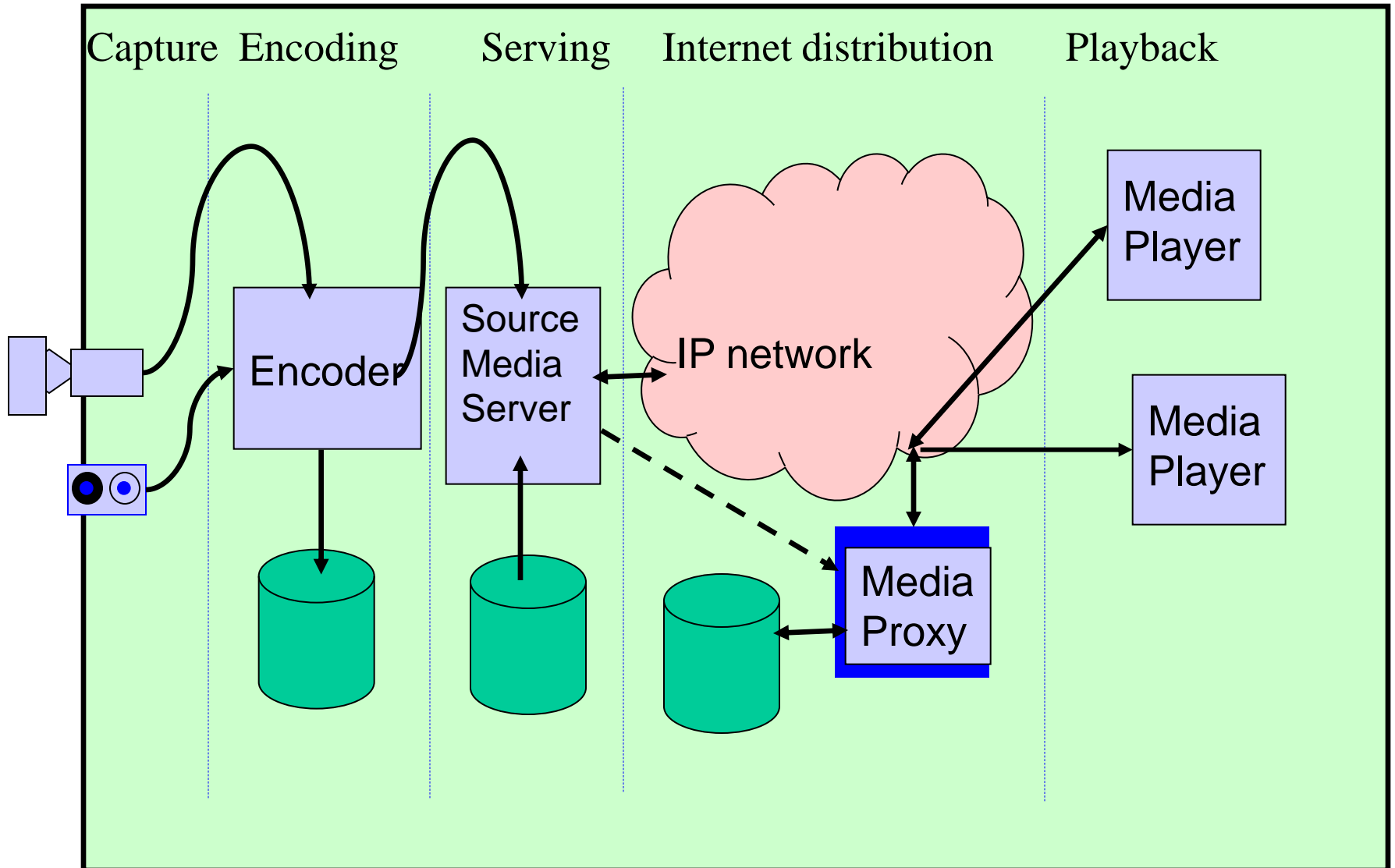
## ❑ Distributed server cluster and proxy media server

- Use a group of servers to improve processing performance
- Use proxy server to reduce number of users' direct accesses to server

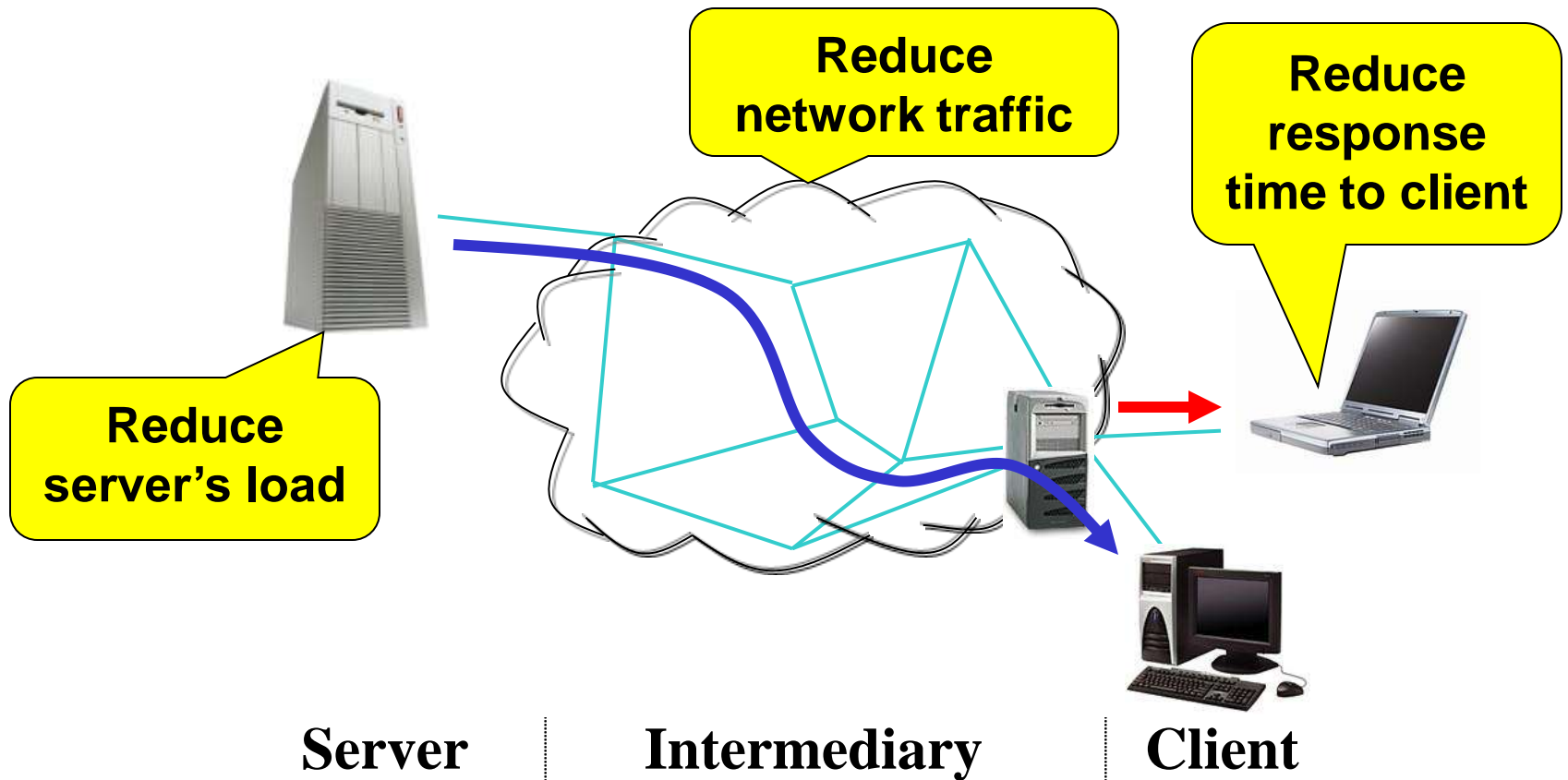


- Drop frames
  - Drop B,P frames if not enough bandwidth
- Quality Adaptation
  - Transcoding
    - Change quantization value
    - Change coding rate
- Video staging, caching, patching
  - **Staging**: store partial frames in proxy
  - **Prefix caching**: store first few minutes of movie
  - **Patching**: multiple users use same video

# Proxy Media Server



# Proxy Server: Reduce Traffic, Time, Load



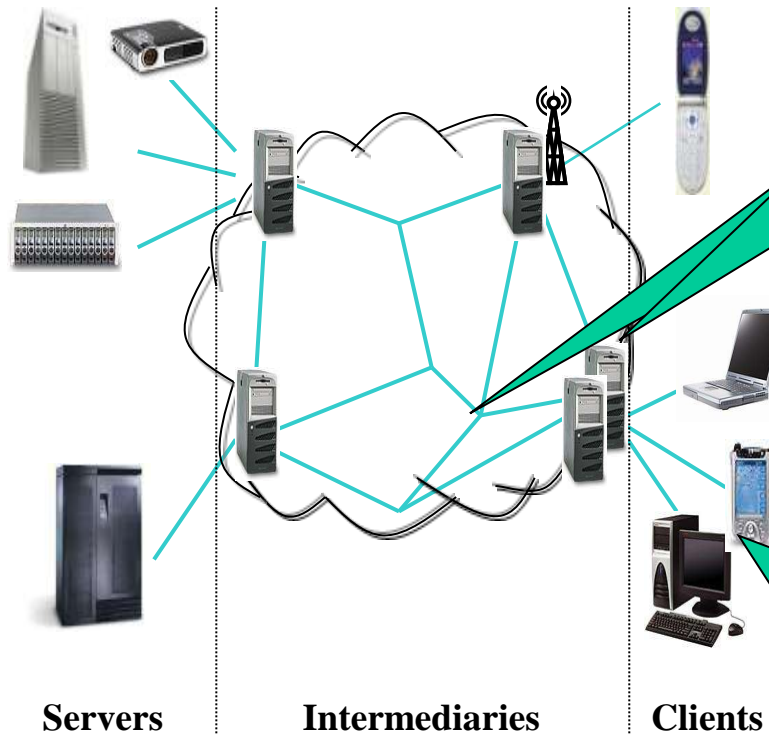


# Distributed Proxy Servers

Very large sizes

**Media Objects**

Very rigorous  
real-time delivery  
constrains:  
**small startup  
latency,  
continuous  
delivery**



Servers

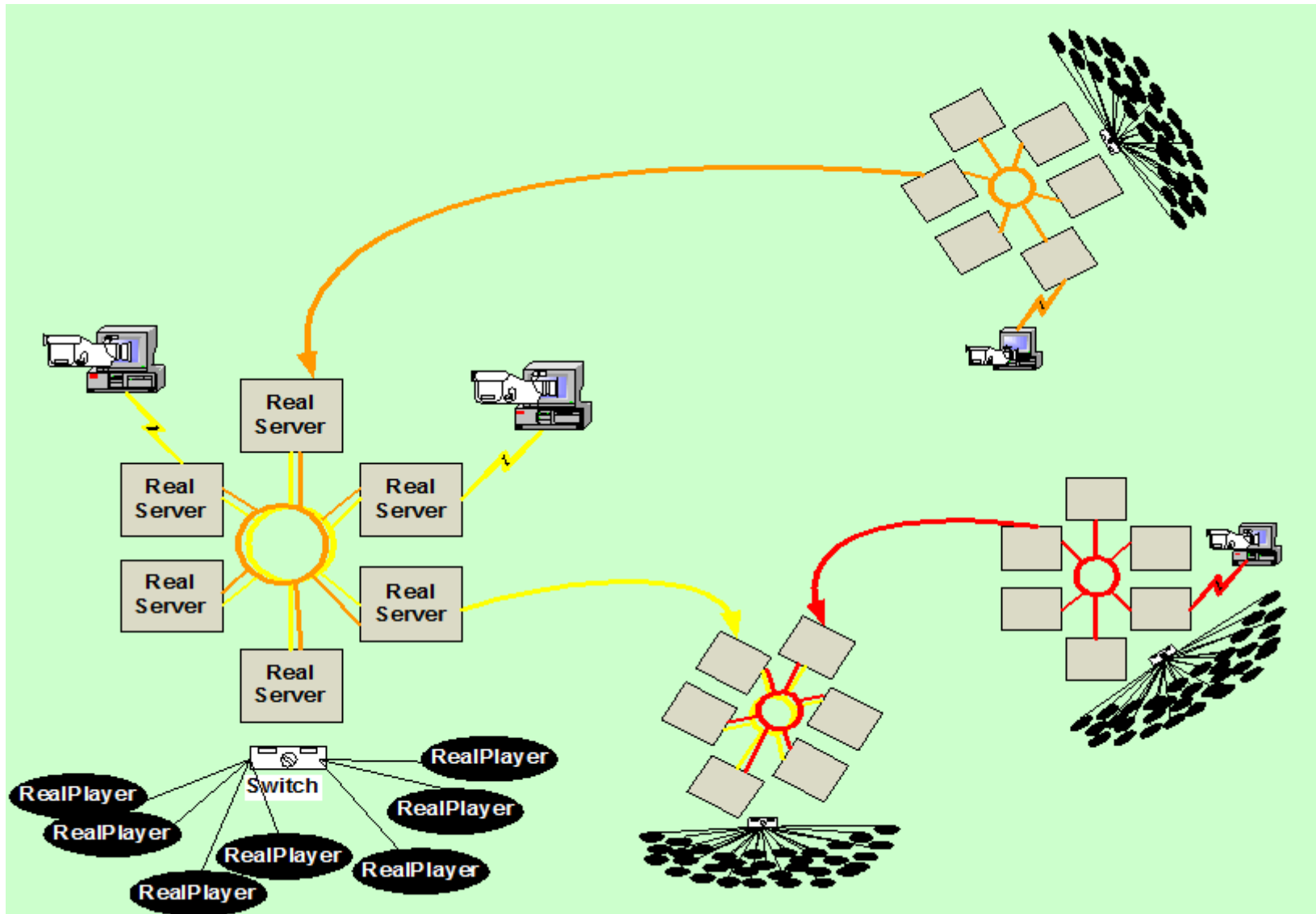
Intermediaries

Clients

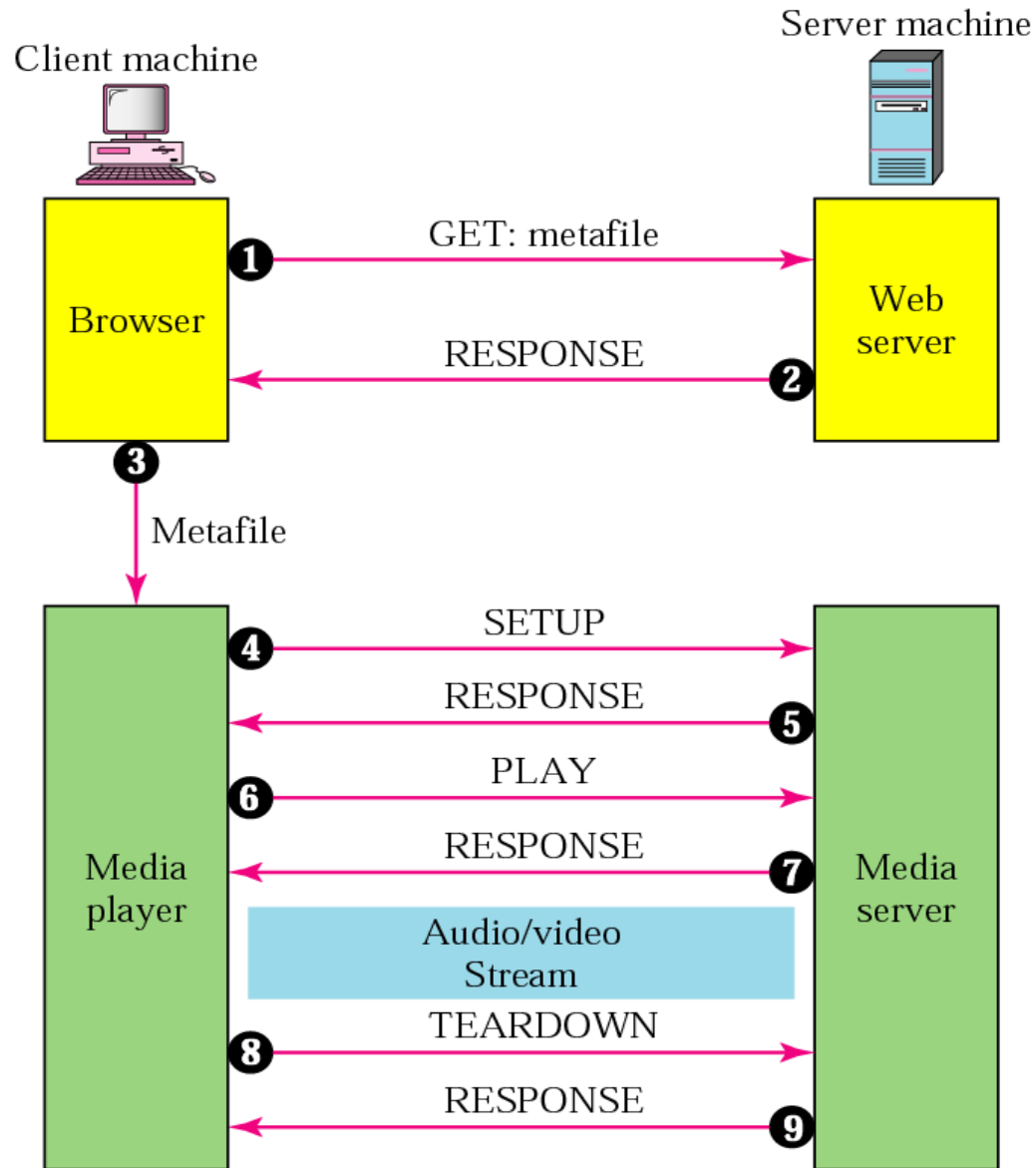
A large number  
of proxies with:  
disk, memory,  
and CPU cycles

Diverse client  
access devices:  
computers,  
PDAs,  
cell-phones

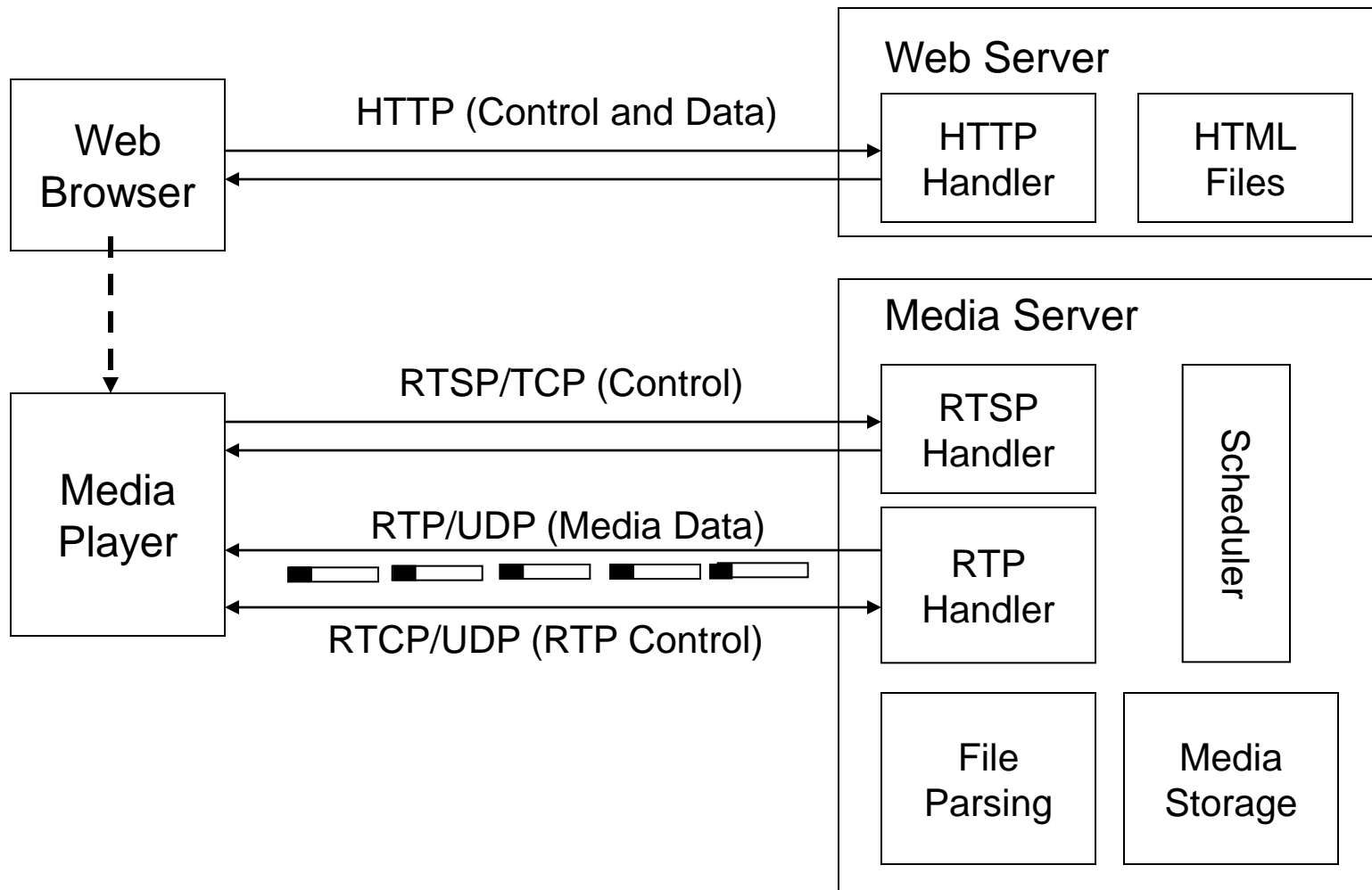
# Distributed Server Clustering



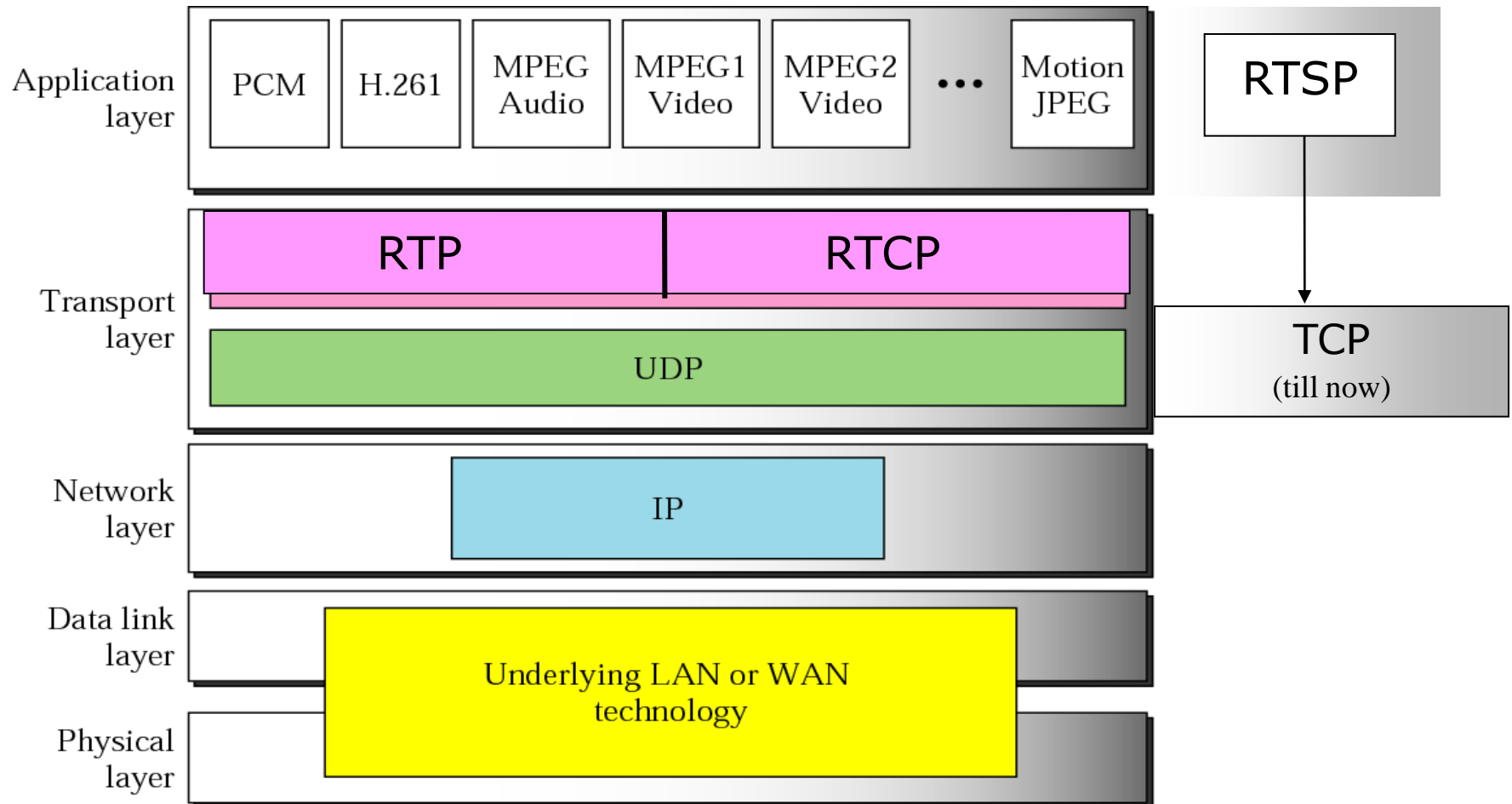
# Media Streaming Service Access Process



# Media Streaming Service Modules



# Protocol Stack for Multimedia Services



# What is RTSP?

- Real-Time Streaming Protocol (RTSP) is a standard defined in RFC 2326 by IETF in 1998
- RTSP is a control protocol intended for:
  - retrieval of media from a media server
  - establishment of one or more synchronized, continuous-media streams
  - control of such streams
- RTSP can be seen as a “network remote control”
- RTSP is not used to deliver the streams
  - use RTP or similar for that

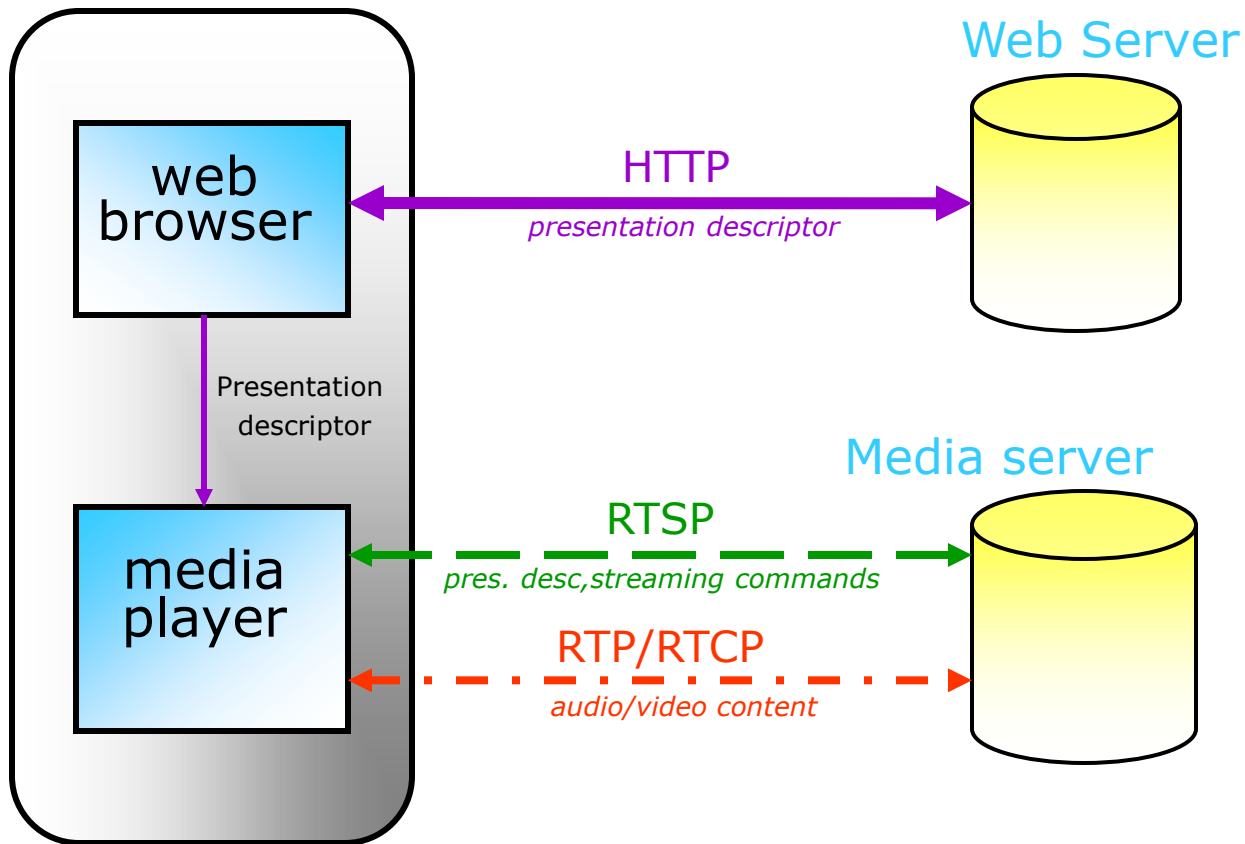
# Differences between RTSP and HTTP

- ❑ The RTSP design is based on HTTP, with the following differences:
  - new methods; different protocol identifier:  
`rtsp://audio.example.com/twister/audio.en`  
`rtsp://video.example.com/twister/video`
  - RTSP servers need to keep state while HTTP servers do not
  - Both RTSP servers and clients can issue requests
  - Data is carried by an external protocol (typically but not necessarily RTP)
  - RTSP uses UTF-8 instead of ISO 8859-1 character set
  - RTSP uses absolute request URIs
  - RTSP defines an extension mechanism

***Transport independent.*** RTSP implements application-layer reliability and can run on top of TCP, UDP, or any other protocol. Standardized ports for RTSP:

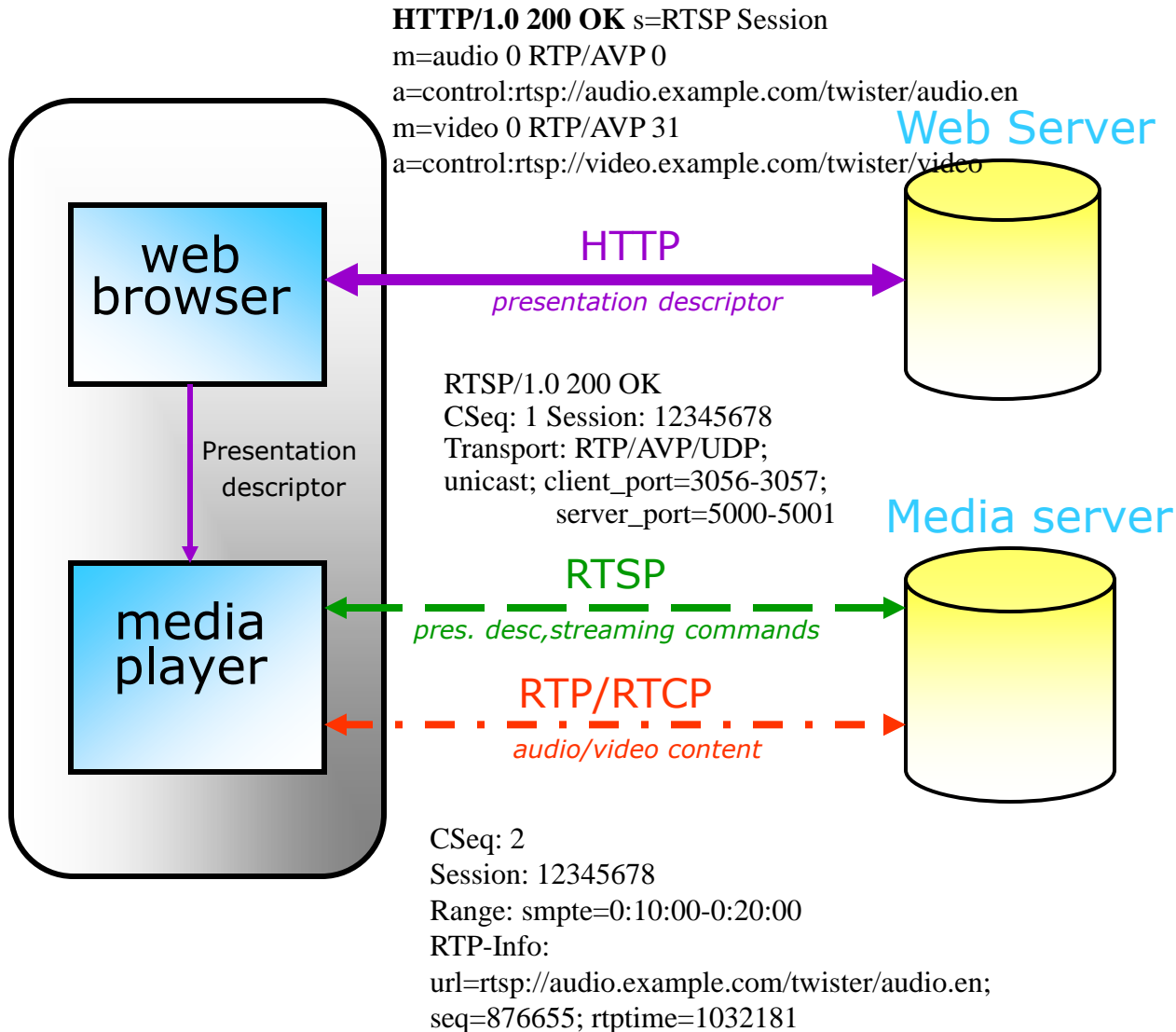
<code>rtsp</code>	<code>554/tcp</code>	Real Time Streaming Control
<code>rtsp</code>	<code>554/udp</code>	Real Time Streaming Control
<code>rtsp-alt</code>	<code>8554/tcp</code>	RTSP Alternate
<code>rtsp-alt</code>	<code>8554/udp</code>	RTSP Alternate

# HTTP and RTSP





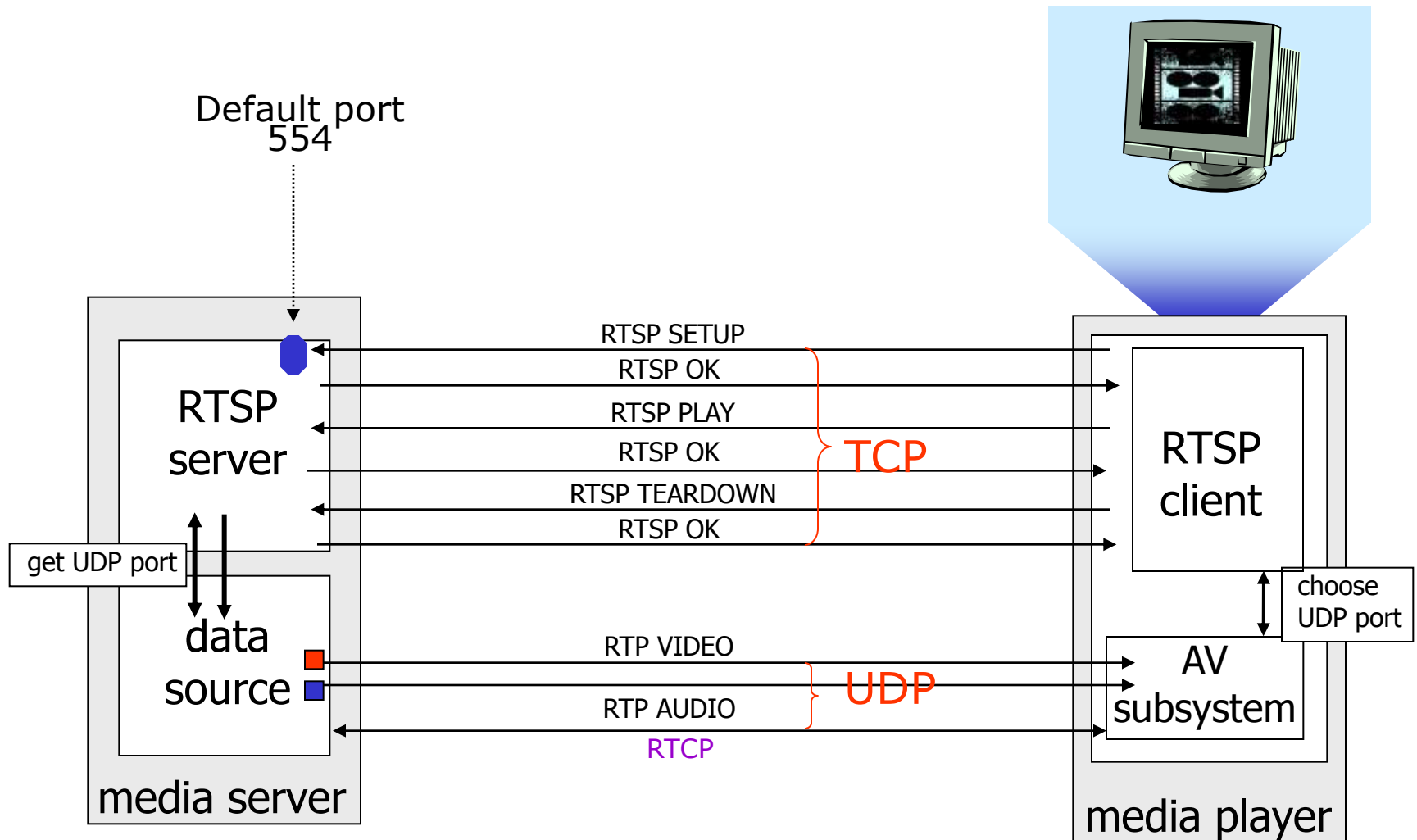
# HTTP and RTSP



# RTSP Methods

OPTIONS	$C \rightarrow S$	determine capabilities of server/client
	$C \leftarrow S$	
DESCRIBE	$C \rightarrow S$	get description of media stream
ANNOUNCE	$C \leftrightarrow S$	announce new session description
SETUP	$C \rightarrow S$	create media session
RECORD	$C \rightarrow S$	start media recording
PLAY	$C \rightarrow S$	start media delivery
PAUSE	$C \rightarrow S$	pause media delivery
REDIRECT	$C \leftarrow S$	redirection to another server
TEARDOWN	$C \rightarrow S$	immediate teardown
SET_PARAMETER	$C \leftrightarrow S$	change server/client parameter
GET_PARAMETER	$C \leftrightarrow S$	read server/client parameter

# RTSP Session

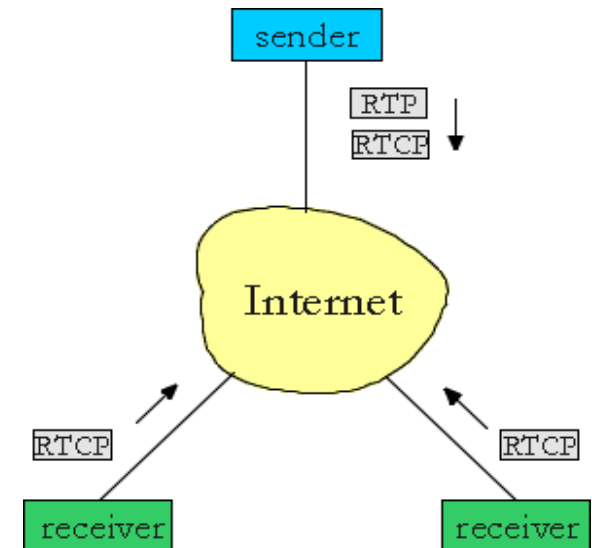
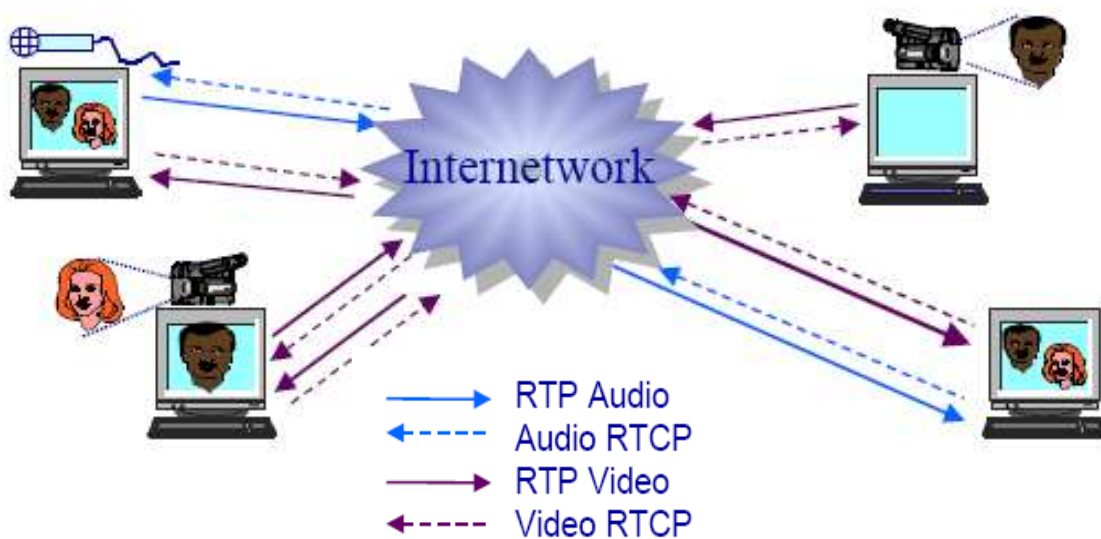


# What is RTP?

- Realtime Transport Protocol (RTP) is an IETF standard
- Primary objective: stream continuous media over a best-effort packet-switched network in an interoperable way.
- Protocol requirements:
  - Payload Type Identification: what kind of media are we streaming?
  - Sequence Numbering: to deal with lost and out-of-order packets.
  - Timestamping: to compensate for network jitter in packet delivery.
  - Delivery Monitoring: how well is the stream being received by the destinations?
- RTP does not guarantee QoS (Quality of Service), i.e., reliable, on-time delivery of the packets (the underlying network is expected to do that).
- RTP typically runs on top of UDP, but the use of other protocols is not precluded

# RTT, RTCP and Session

- RTP is composed of two closely-linked parts:
  - The Real-Time Transport Protocol (RTP), used to carry real-time data
  - The RTP Control Protocol (RTCP), used to:
    - Monitor and report Quality of Service
    - Convey information about the participants of a session



- Two connective ports are needed for media data transmissions
  - Even number  $2n$  for RTP and odd number  $2n+1$  for RTCP
- RTP defines the concept of a **profile**, which completes the specification for a particular application:
  - Media encoding specifications, Payload format specifications

# RTP Header

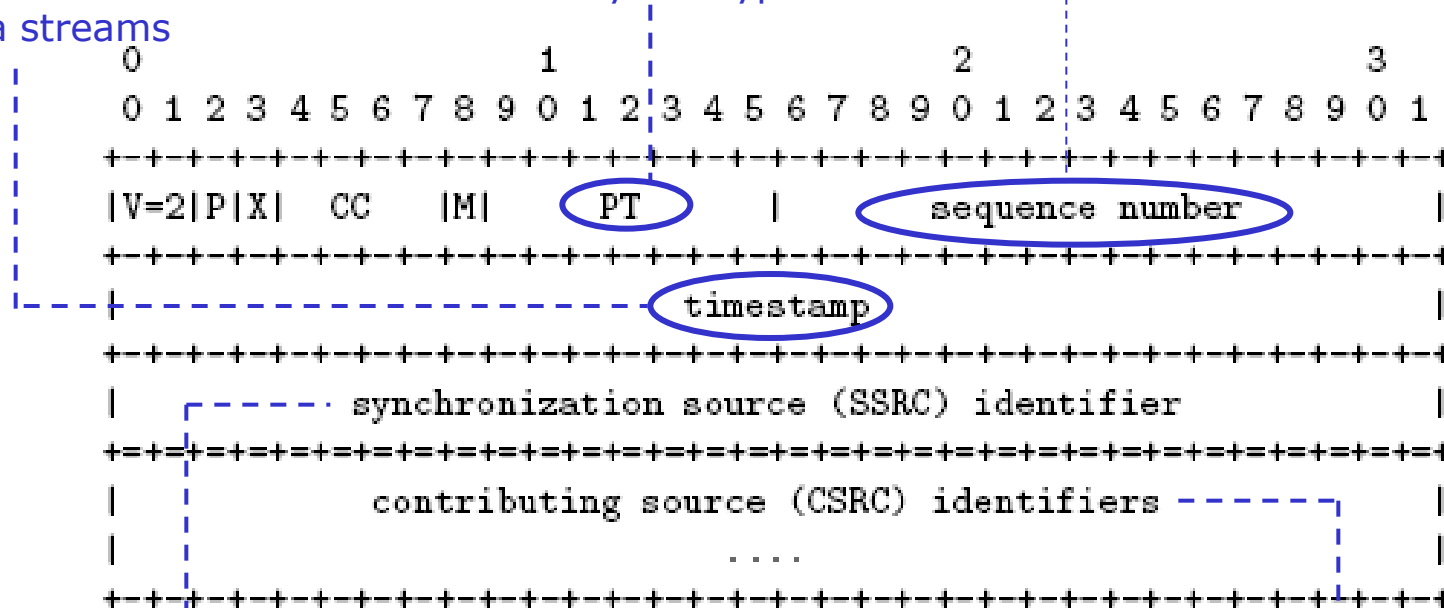
Sampling instant of first data octet

- multiple PDUs can have same timestamp
- not necessarily monotonic
- used to synchronize different media streams

Incremented by one for each RTP PDU:

- PDU loss detection
- Restore PDU sequence

Payload type

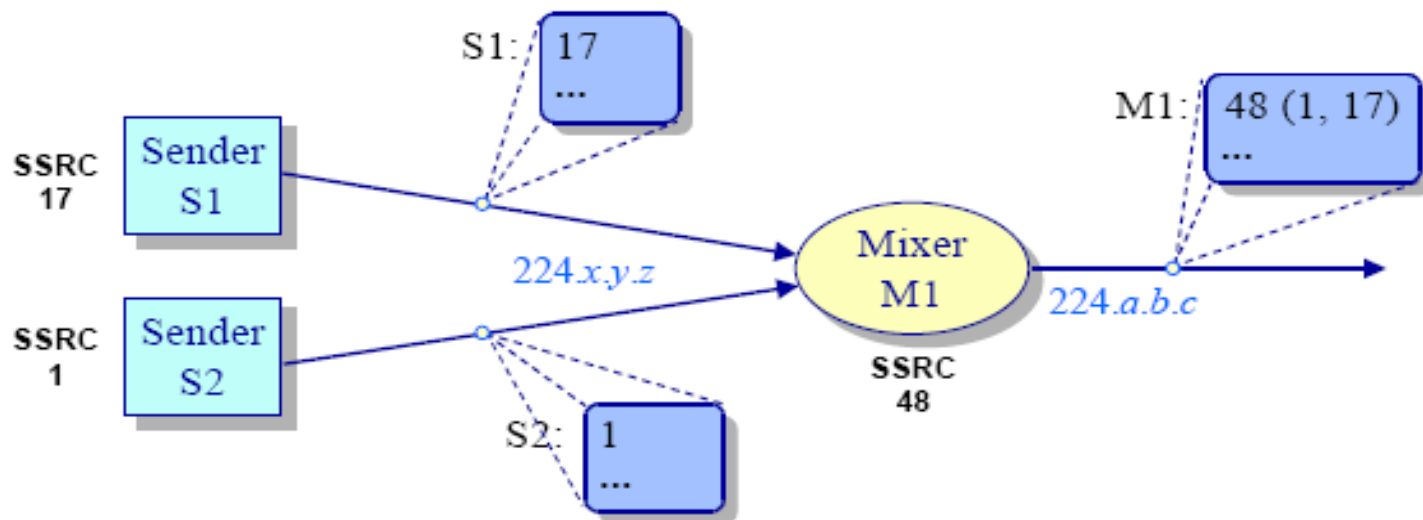


Identifies synchronization source

Identifies contributing sources  
(used by mixers)

# RTP Mixer

RTP *mixer* - an intermediate system that receives & combines RTP PDUs of one or more RTP sessions into a new RTP PDU



- Stream may be transcoded, special effects may be performed.
- A mixer will typically have to define synchronization relationships between streams. Thus...
  - Sources that are mixed together become **contributing sources (CSRC)**
  - Mixer itself appears as a new source having a new **SSRC**

# RTCP Reports

- Cumulative counts allow both long- and short-term analysis
  - any two reports can be subtracted to get activity over an interval
  - NTP timestamps in reports allow you to compute rates
  - monitoring tools needn't know anything about particular media encoding
- Sender reports give utilization information
  - average packet rate and average data rate over any interval
  - monitoring tools can compute this without reading any of the data
- Receiver reports give loss and round-trip information
  - extended sequence number can be used to compute packets expected
  - packets lost and packets expected give long term loss rate
  - fraction lost field gives short-term loss rate, with only a single report
  - LSR and DLSR give sender's ability to compute round-trip time



# Analyzing RTCP Reports

final\_rtp - Ethereal

File Edit View Go Capture Analyze Statistics Help

Filter: Expression... Clear Apply

No.	Time	Source	Destination	Protocol	Info
859	44.201738	192.168.0.101	192.168.0.103	G.723	Payload type=110, G.723, SSRC=3860006015, Seq=8704, Time=302004
860	44.227289	192.168.0.103	192.168.0.101	RTCP	Sender Report

Real-time Transport Control Protocol

- [Stream setup by H245 (frame 49)]
  - 10... = Version: RFC 1889 version (2)
  - ..0... = Padding: False
  - ...0 0001 = Reception report count: 1
  - Packet type: Sender Report (200)
  - Length: 12
  - Sender SSRC: 3879416967
  - Timestamp, MSW: 482
  - Timestamp, LSW: 1212153856
  - RTP timestamp: 302928
  - Sender's packet count: 283
  - Sender's octet count: 6792
- Source 1
  - Identifier: 3860006015
  - SSRC contents
    - Fraction lost: 1 / 256
    - Cumulative number of packets lost: 3
    - Extended highest sequence number received: 8704
    - Sequence number cycles count: 0
    - Highest sequence number received: 8704
    - Interarrival jitter: 7
    - Last SR timestamp: 3842553664
    - Delay since last SR timestamp: 122368

Real-time Transport Control Protocol

- [Stream setup by H245 (frame 49)]
  - 10... = Version: RFC 1889 version (2)
  - ..0... = Padding: False
  - ...0 0001 = Source count: 1
  - Packet type: Source description (202)
  - Length: 4
  - Chunk 1, SSRC/CSRC 3879416967
    - Identifier: 3879416967
    - SDES items
      - Type: CNAME (user and domain) (1)
      - Length: 6
      - Text: SADHAK
      - Type: END (0)

P: 1335 D: 1335 M: 0

header of SR report

sender info

receiver report block

SDES items

# Demos of Streamed Audio and Video

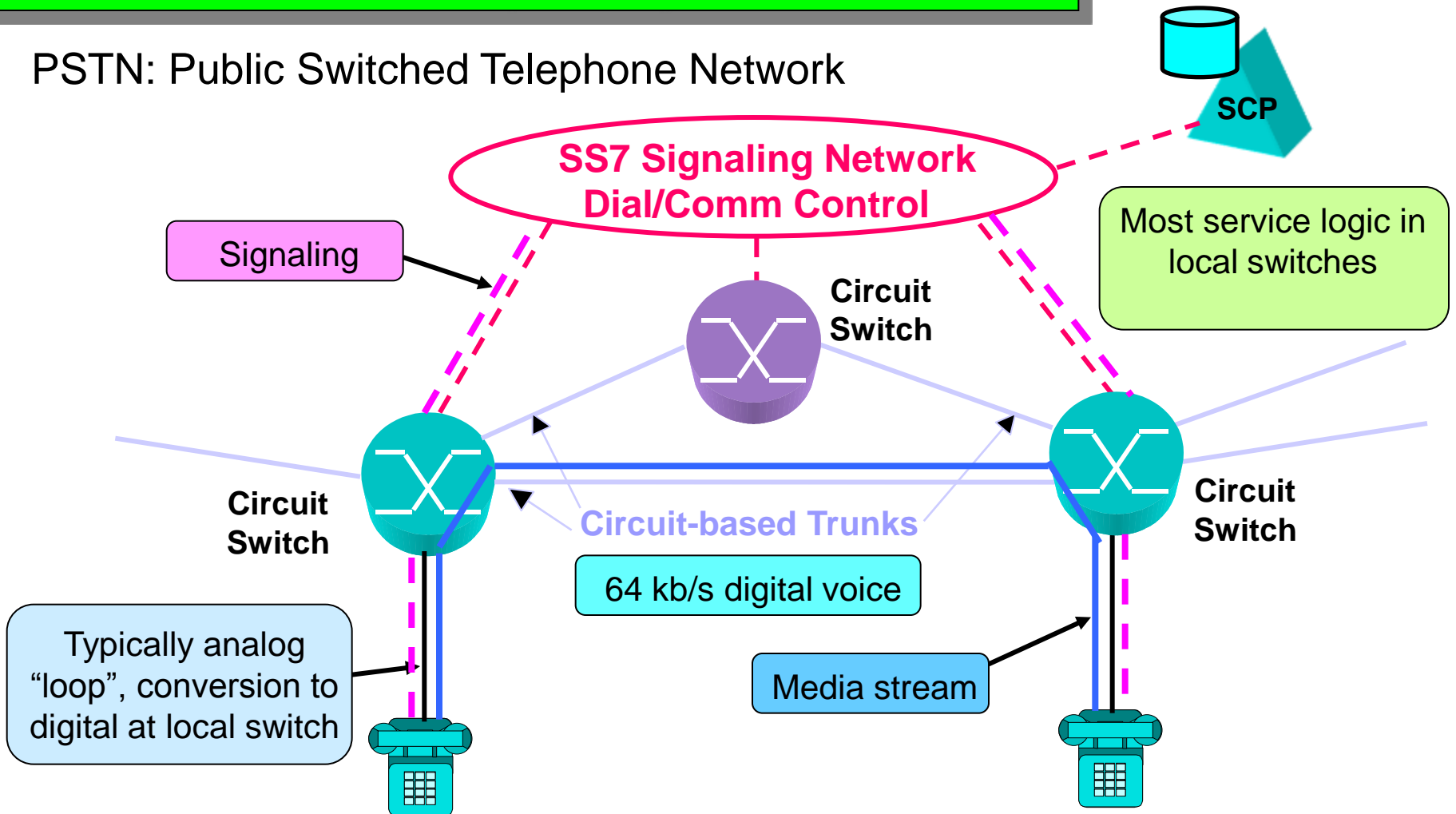
# Media Communications

## Internet Telephony and Teleconference

- Scenario and Issue of IP Telephony
- Scenario and Issue of IP Teleconference
- ITU and IETF Standards for IP Telephony/conf.
- H.323 Standard Series for IP Multimedia Comm.
- T.120 Standard Series for Data Conferencing
- SIP/SDP (Session Initiation/Description Protocol)

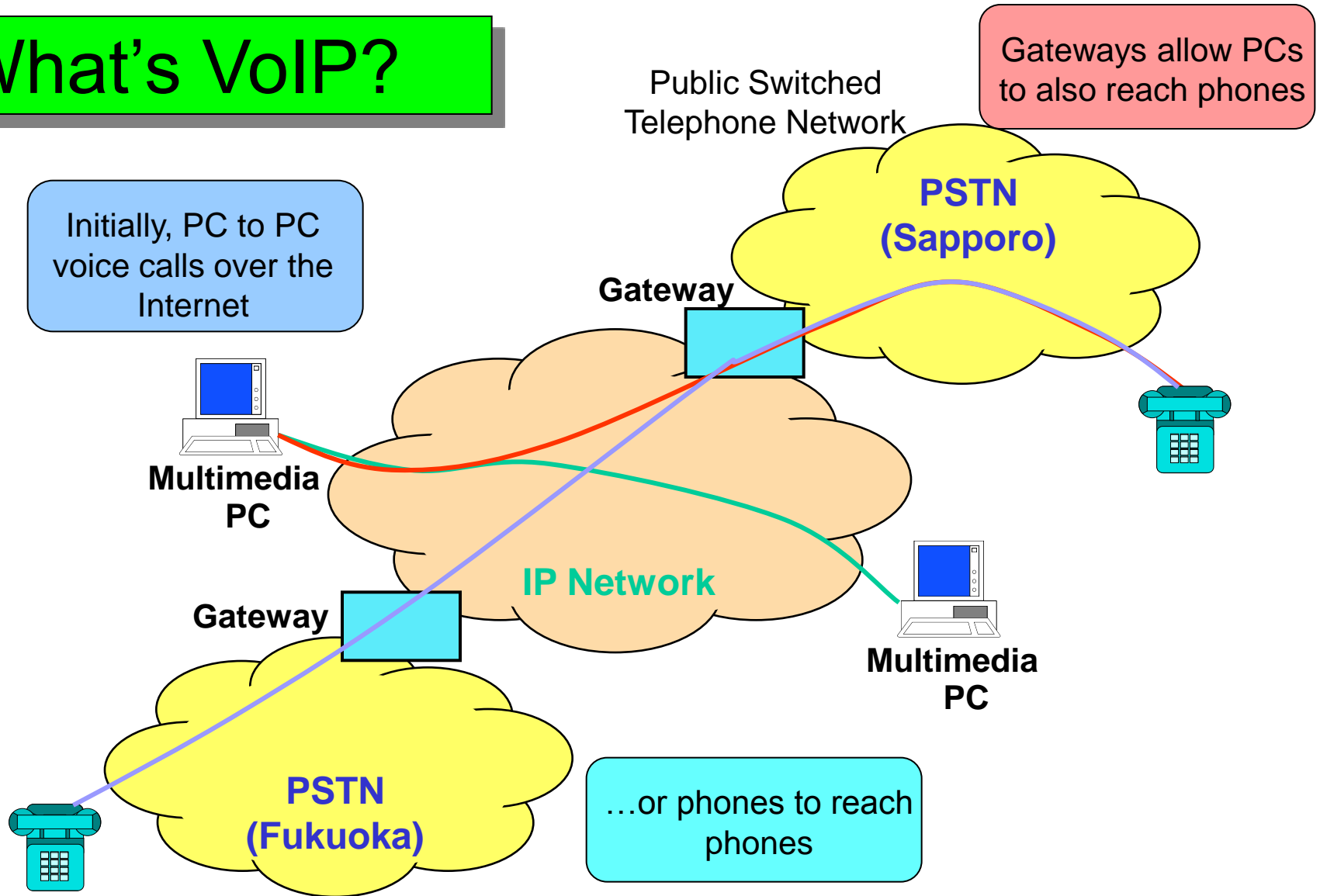
# Traditional Telephony over PSTN

PSTN: Public Switched Telephone Network



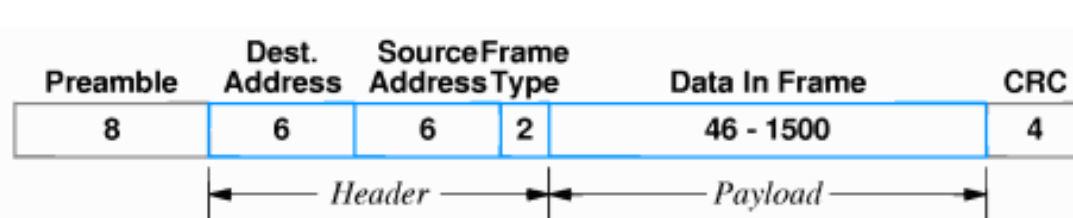
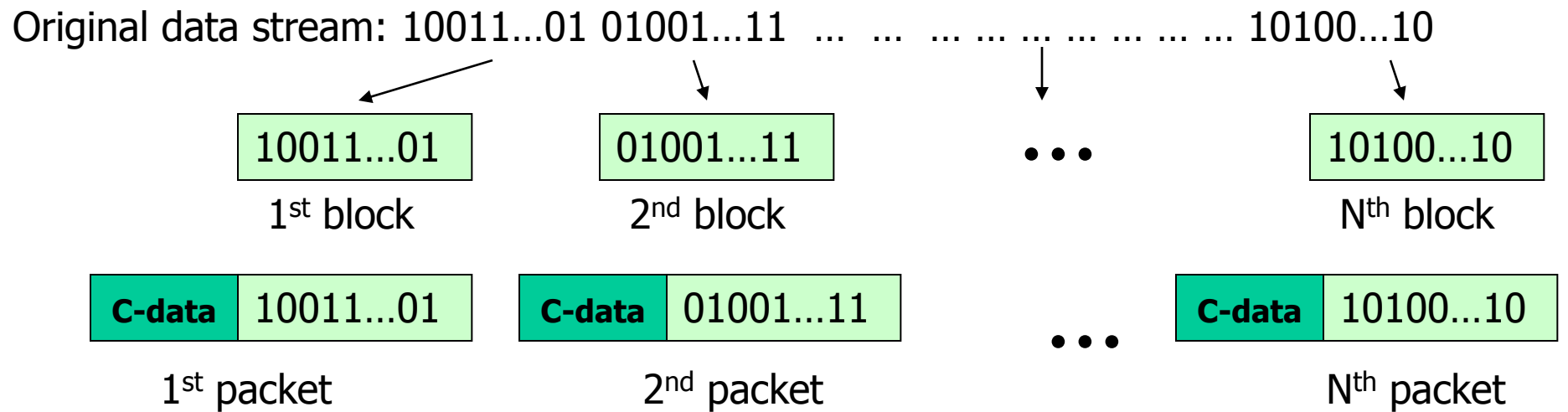
- Different pair of telephones travels over a parallel/separate links
- Features: High voice quality, low bandwidth efficiency, inflexible

# What's VoIP?

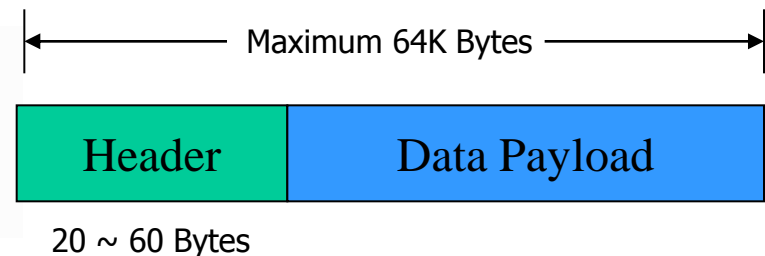


# Packet-based Network (IP Network)

The data transmission method in computer communication is conceptually similar as the postal system. A large data stream will be divided into relatively small blocks, called packet, before transmission. Each packet is transmitted individually and independently over networks → **Packet-based Communication/Network**



Ethernet Packet



Internet Packet

# Temporal Relations in Video and Audio

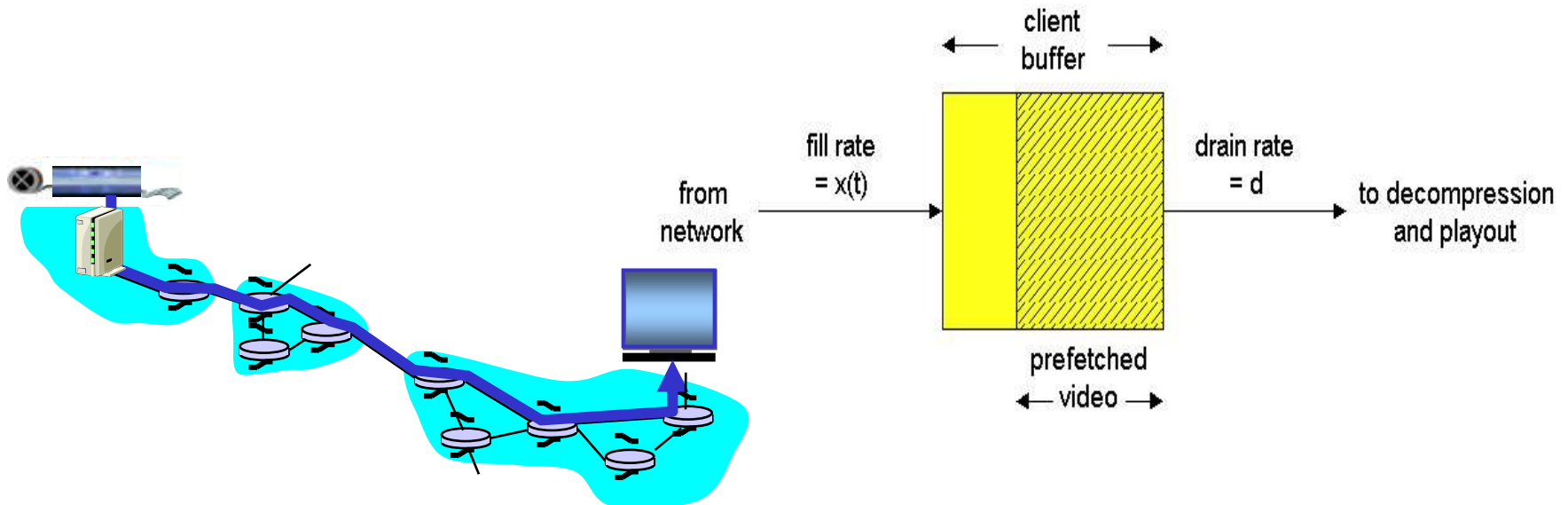
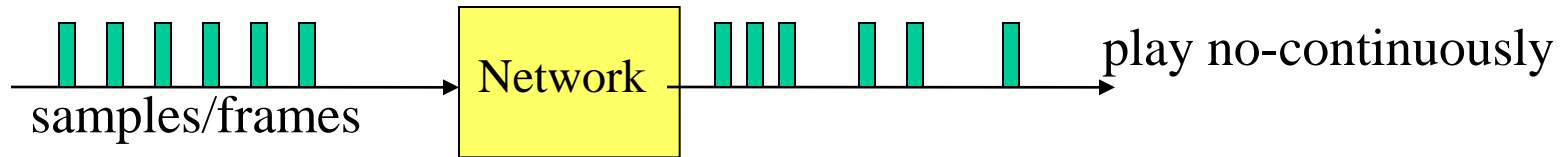
Pic. 1 Pic. 2 Pic. 3 Pic. 4 Pic.n

$1/30$  s

physical frame duration =  $1/\text{sample frequency}$  (e.g.,  $1/8000$  s)

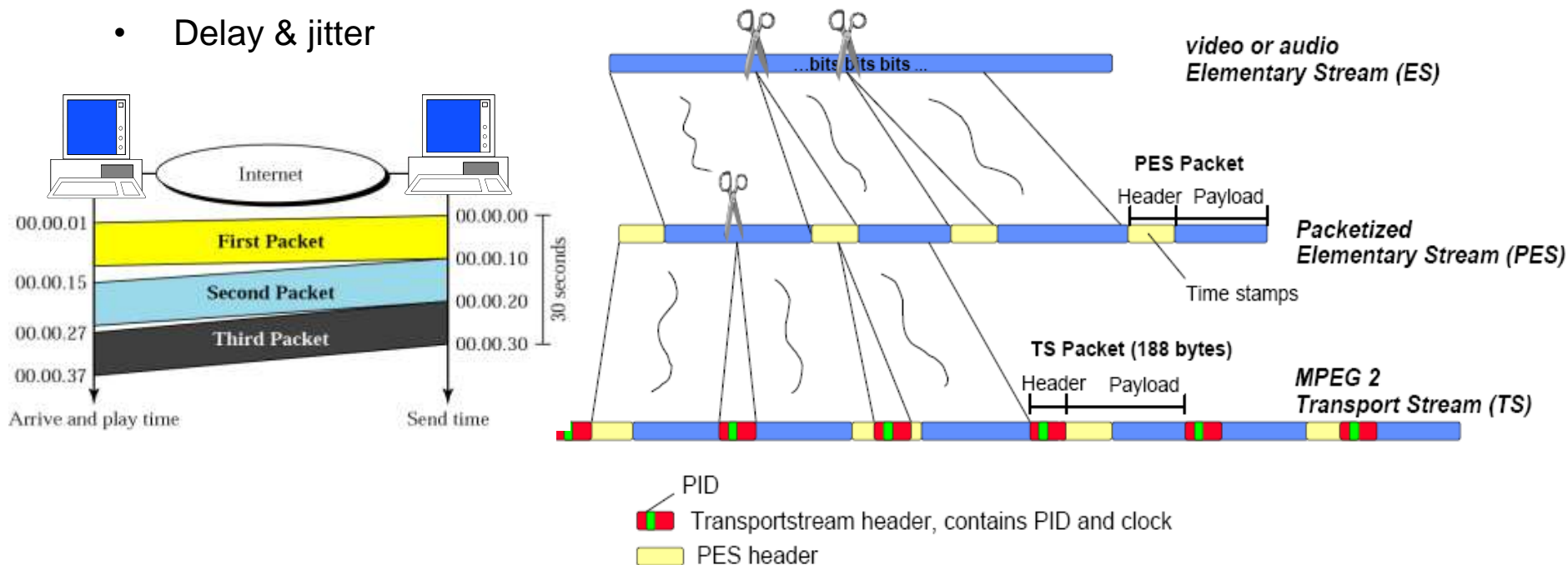


duration of a Logical Data Unit of 512 Bytes (e.g., = 0.064 s)



# VoIP Basic Features and History

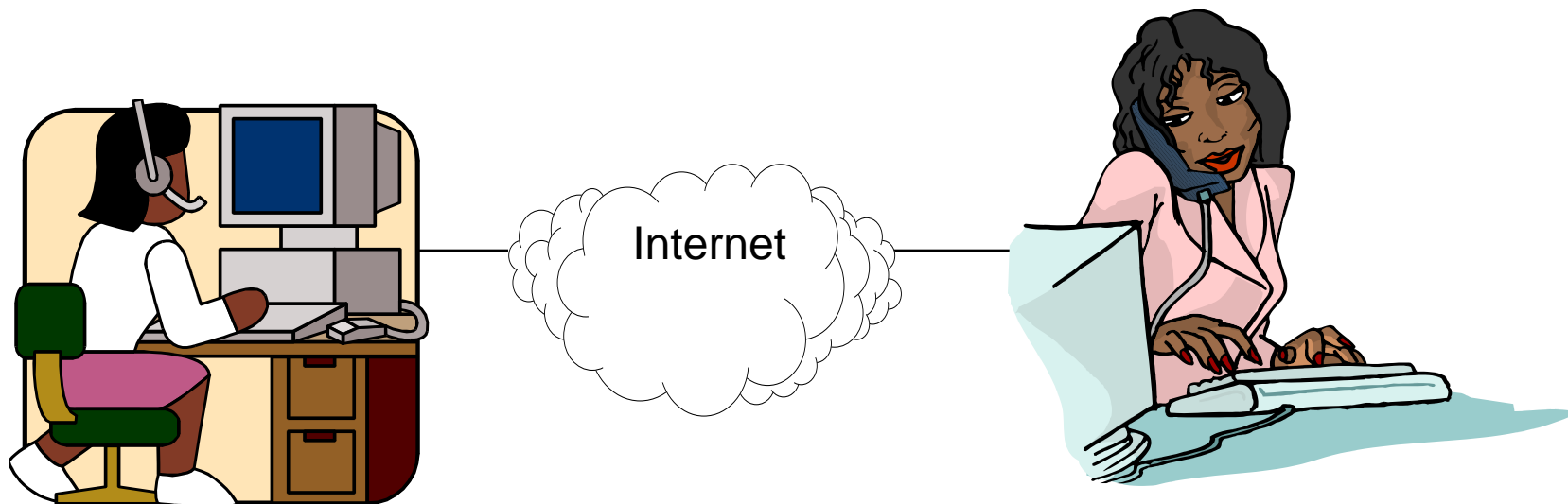
- Internet telephony, also called Voice over IP (VoIP), refers to using the IP network infrastructure (LAN, WLAN, WAN, Internet) for voice communication.  
IP (Internet Protocol) transmission unit: packet
- First product appeared in February of 1995:
  - Internet Phone Software by Vocaltec, Inc., “free” long distance call via PC
  - Software compressed the voice and sent it as IP packets.
- Other software/products soon followed → **NetMeeting, Skype, Gphone, ...**
- Delay & jitter



**Rule: Every elementary stream gets its own (Packet ID) PID**



# Scenario 1: PC to PC



- Issues:

- Addressing, i.e., VoIP phone number
- Call admission, setup, control, release, etc
- IP network related: delay, jitter, packet loss, out-of-order
- Transmission overhead: Headers
- Small delay
  - Small packet size

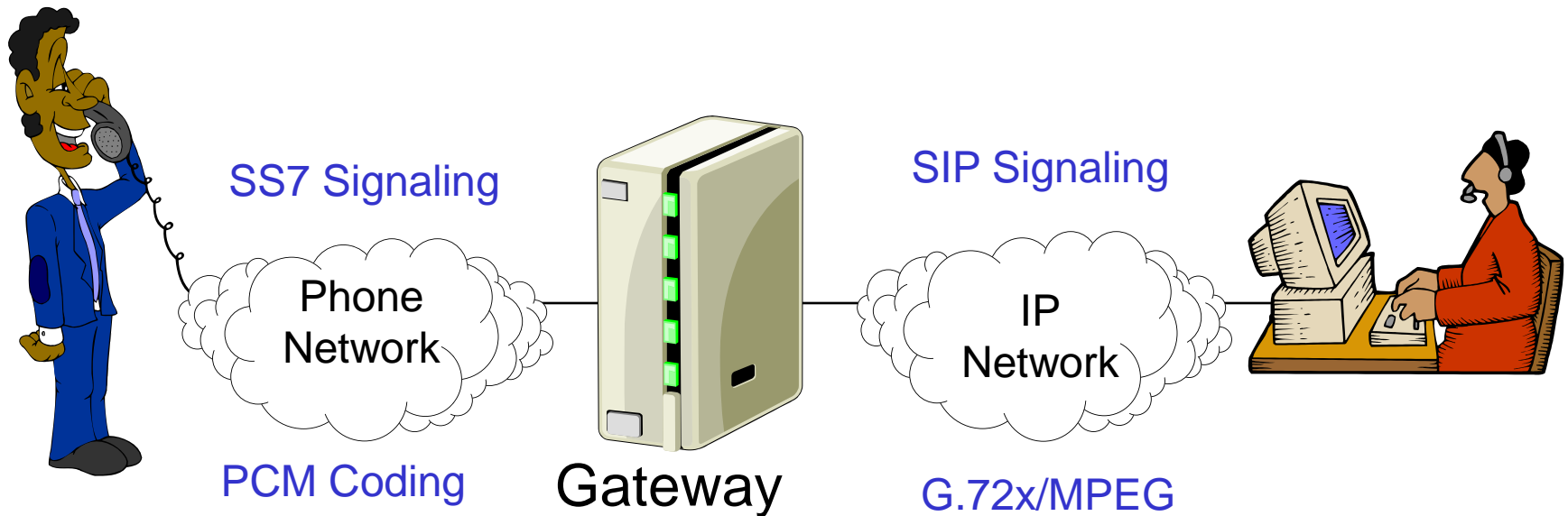


Total > 100 bytes Can't be large for voice delay

Voice data rate: 1~8KBytes/Second

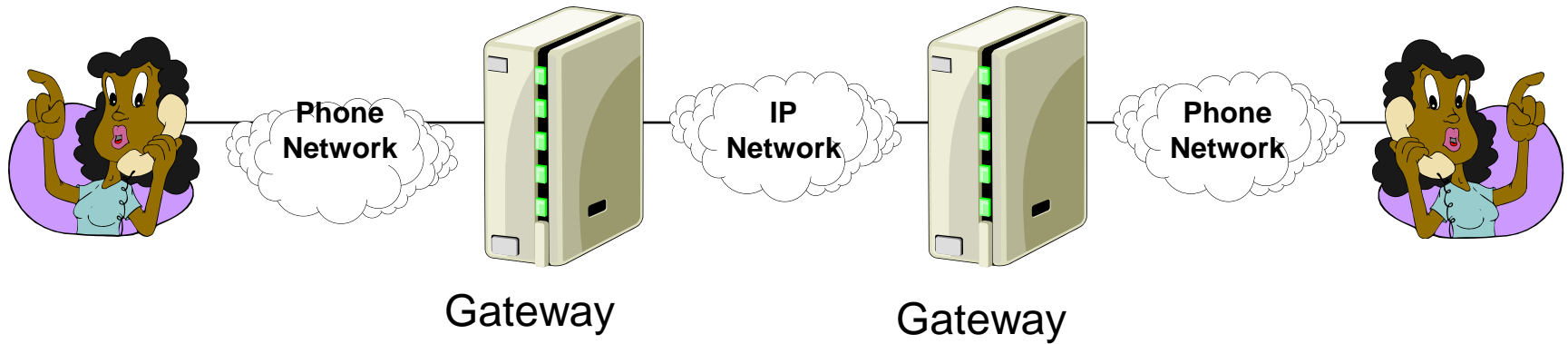
or 8~64Kbps (bits-per-second)

# Scenario 2: PC to Phone



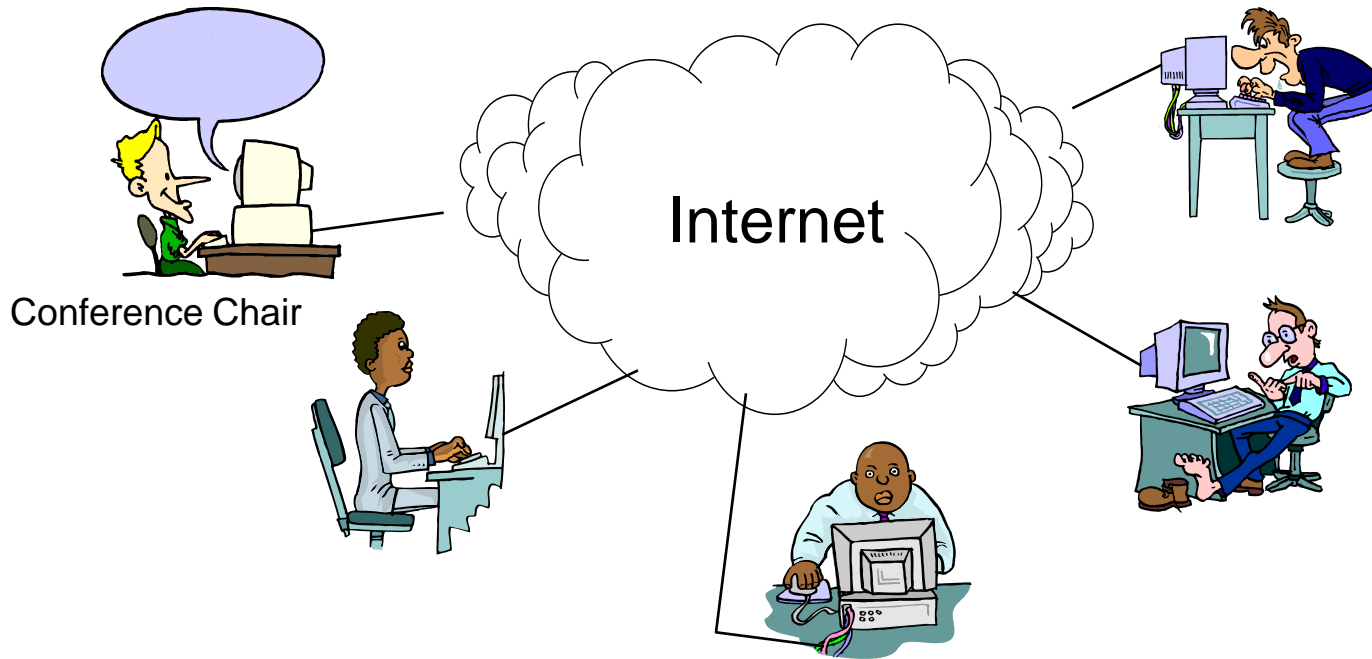
- A Gateway is needed to connect the PSTN to the IP network:
  - Signaling conversion
  - Format conversion

# Scenario 3: Phone to Phone



- Gateways will connect the phone network to the IP network.
- The IP Network can be a dedicated backbone or intranet (to provide guaranteed QoS) or can be the Internet (no guarantees ...)
- The phone network can be a company PBX (Private Branch Exchange) or carrier switches

# What is Internet Teleconference



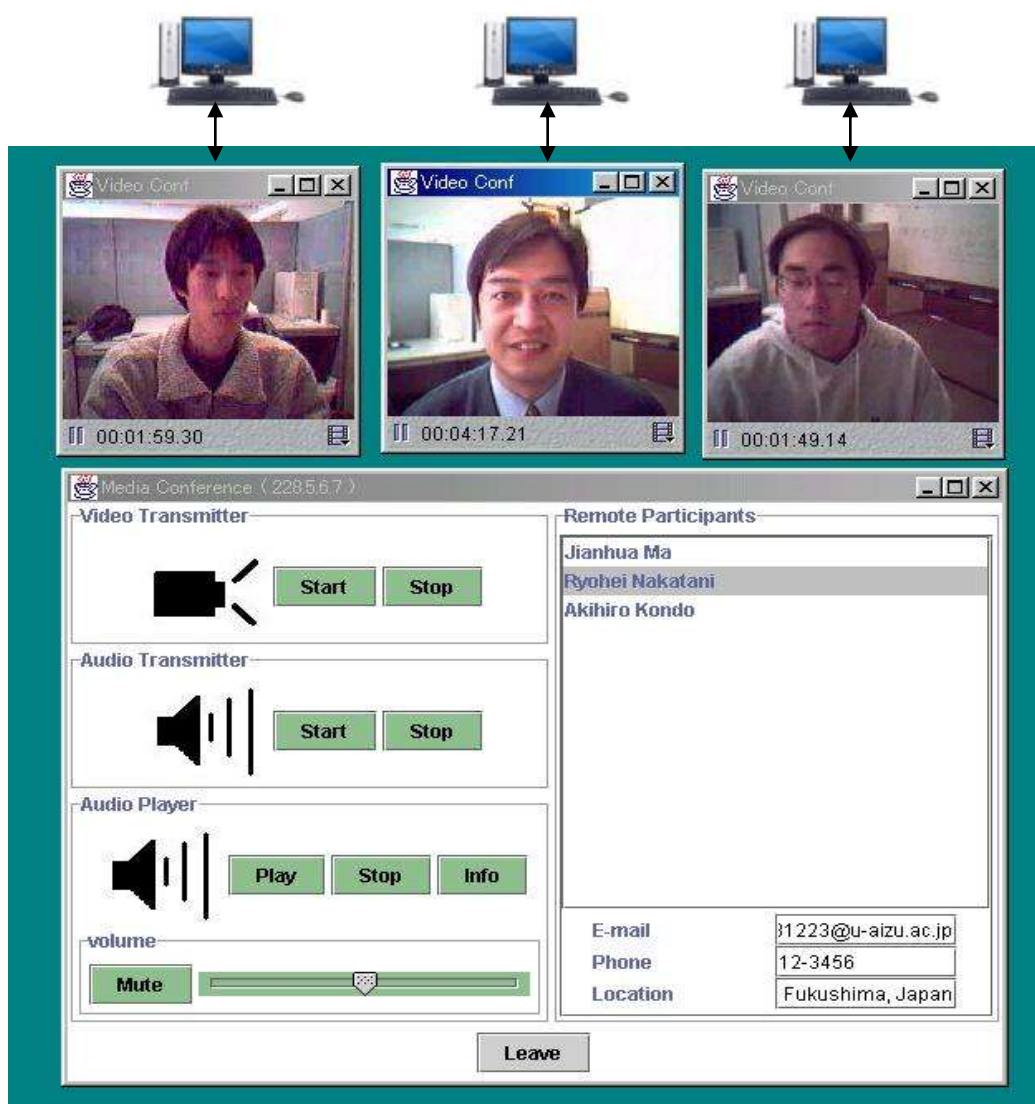
Internet teleconference: A group of people communicate each other via voice, video and/or other data over the Internet

- Conference initiation, start, join, leave, end, control, etc.
- Sending audio/video data from one-to-many (multicast)
- Sharing other conference data (data conferencing) among all participants
- Synchronization and network delay, jitter, packet loss, ...

# Example of Audiovisual Conference



NetMeeting



# What is Data Conferencing?

Data conferencing is a virtual connection between two or more computers where:

- All computers in the conference display a common graphical image of text, graphics or a combination of both.
- Each computer in the conference displays any changes to the common image in near real time.
- Participants have ability to interact with the displayed document
- WYSIWIS: What You See Is What I See

## Presentation (group broadcast)

- Broadcast event where a single presenter's electronic presentation is distributed to multiple remote computers.

## Collaboration (group meeting)

- Everyone can talk, operate, ...
- Usually involves a small conference of 3-10 participants
- Two types of Collaboration: Whiteboarding & Application Sharing



# Example of Data Conferencing: VCR

VCR - Virtual Collaboration Room [ Group project ]

Network Preferences

Workspace Panel

Object Cabinet

Plan Group Case Private Archive Voting White board Chat Navigator Audio Video

Object Panel

All Current

Move to: RED

ChatBoard - 29  
SimpleAnimation - 42  
Shared navigator - 43  
Nethello - 44  
VoteBoard - 45  
WhiteBoard - 46  
AudioPlayer - 47  
VideoPlayer - 48  
SimpleAnimation - 49  
ChatBoard - 50  
WhiteBoard - 51  
ChatBoard - 52  
AudioPlayer - 53  
Nethello - 54

info change action

Object Information

Owner: r-huang  
State: PS  
Mode: Free-Control  
Handler: All

User Panel [jianhua]

C jianhua  
P r-huang  
Nakatani (leave at 11:57)  
a-kondo (login at 10:52)  
Kato

Exit Leave Wait Chair Show

User: [???]

Person Contact System

OS: Linux2.2.12(i586)  
Host: snow.u-aizu.ac.jp(163.143.1  
Login: Sun Dec 12 12:58:53 JST 19

VideoPlayer - 48

Open Size Speed

00:11 00:33

Loop

WhiteBoard - 46

File Config

Owner: jianhua

VCR - Virtual Collaboration Room

This is a GS whiteboard

SimpleAnimation - 42

start/stop

Shared navigator - 43

URL: http://www.hosei.ac.jp/

Next Back Reload Stop

HOSEI  
Hosei University

総合案内  
沿革と特色  
学部  
大学院  
通信教育  
研究所他  
図書館  
付属中高  
事務部局

総長  
メッセージ

What's New

120周年記念  
募金のお願い

キャンパス

入学案内

イベント

総合案内 総長メッセージ キャンパス 入学  
就職情報 国際交流

What's New 120周年記念基金のお願い イ  
学内掲示 ENGLISH

VoteBoard - 45

Question

Should we meet at 10:00am tomorrow?

START

0:00 min 2:00 min

Yes 2  
No 1

ChatBoard - 29

File

Owner: r-huang

r-huang > Jianhua, should I suggest to meet again tomorrow?  
jianhua > Of course, go ahead.

AudioPlayer - 47

Open Size Speed

00:00 ???:??

Loop

Nethello - 44

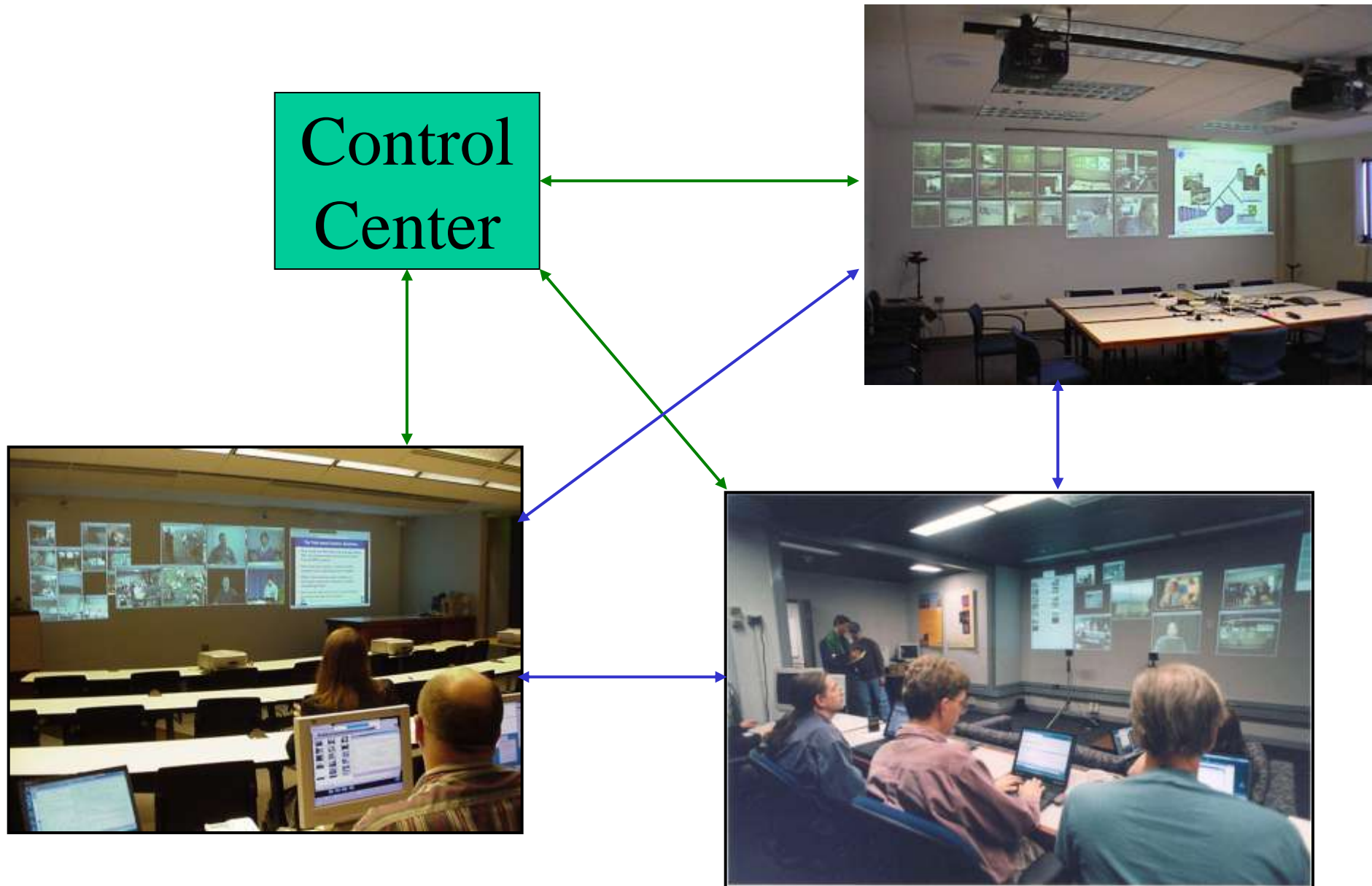
7 7

Your Turn!

Group Chat

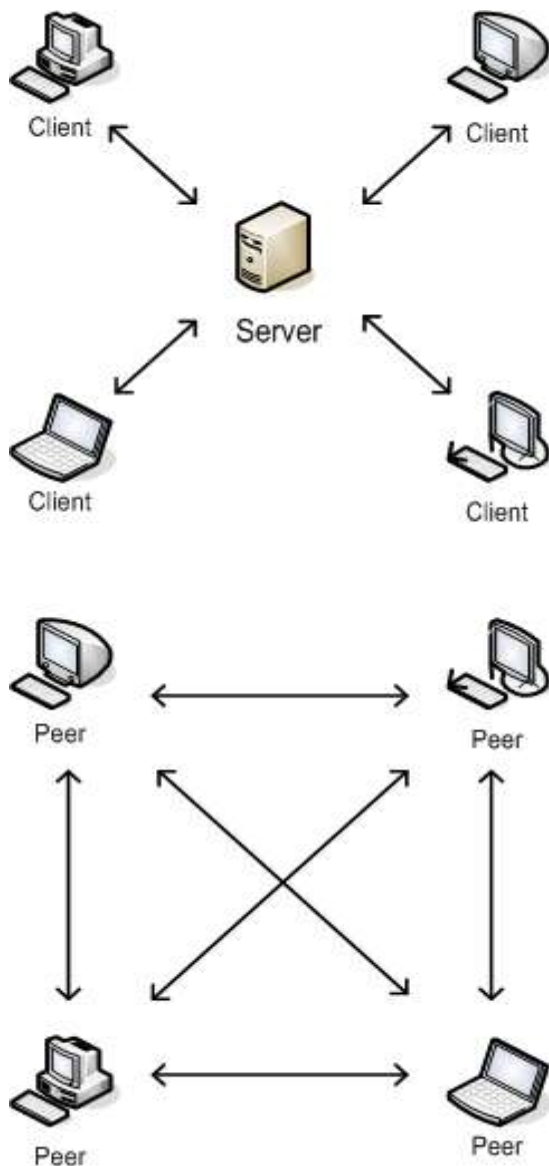
a-kondo > Could anyone tell me what is a GS object?  
r-huang > A GS object is a group shared object. I will create a GS whiteboard, a PS chat board, and others.  
I will also create a GS animation, a PS audio player, and a NS othello game.

# Example of Tele-Conference Rooms

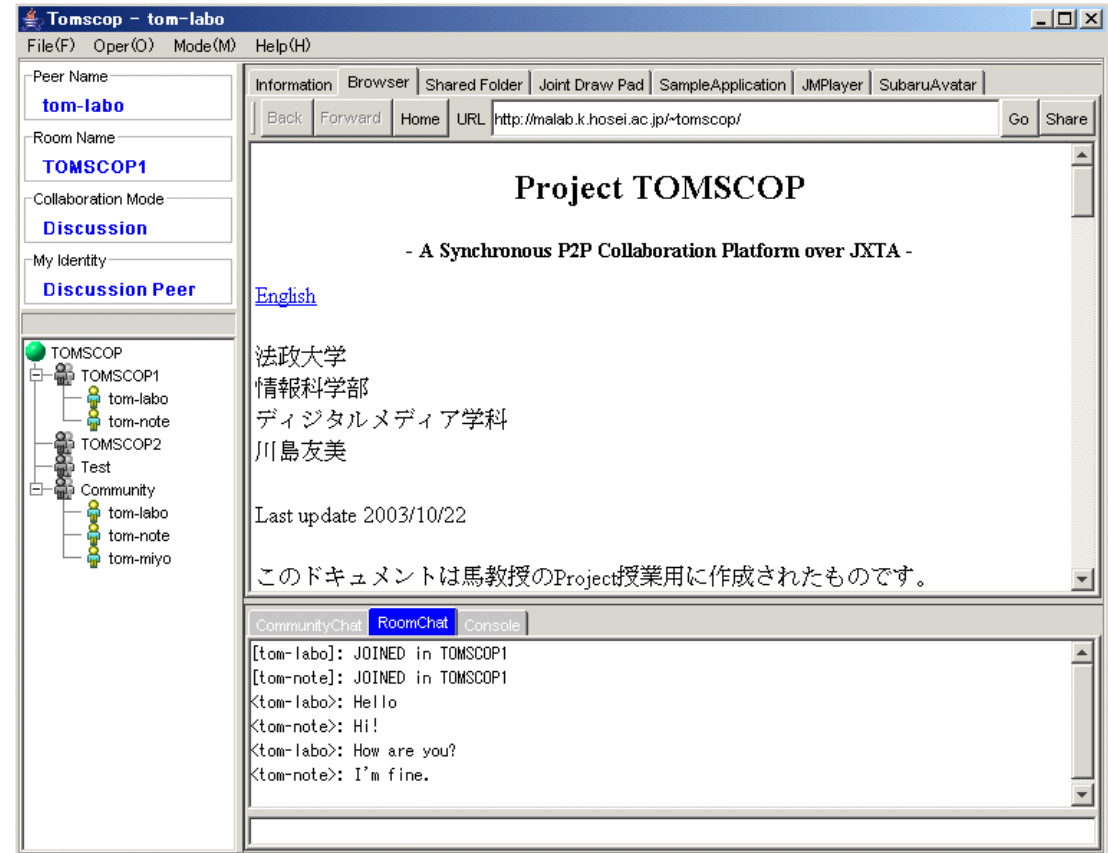




# Server-Client & P2P Communication Models

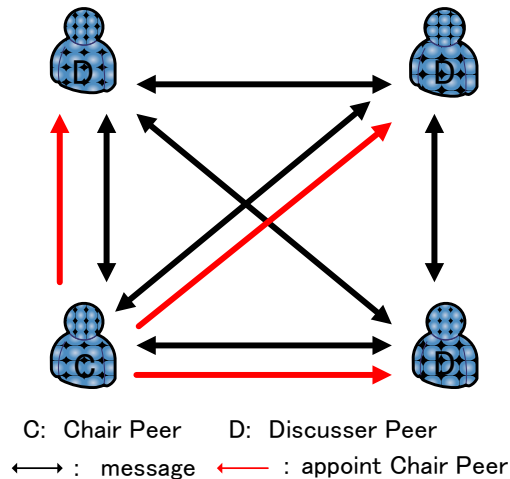


- Client/Server model: TANGO, Habanero, **VCR**  
Problem: load, cost, system down
- Peer-to-Peer model: DSC, Groove, **TOMSCOP**  
Problem: difficulty of peer/group management

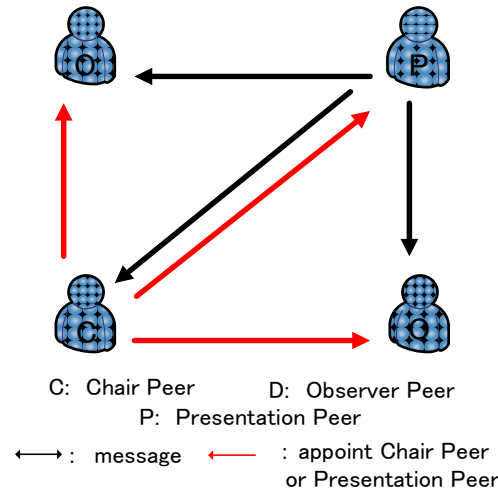


# Peer Identity and Collaboration Modes

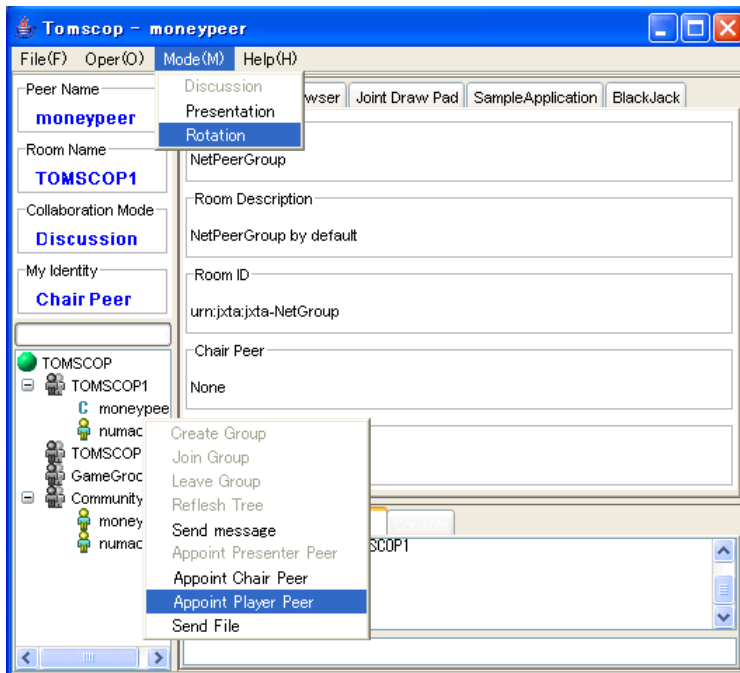
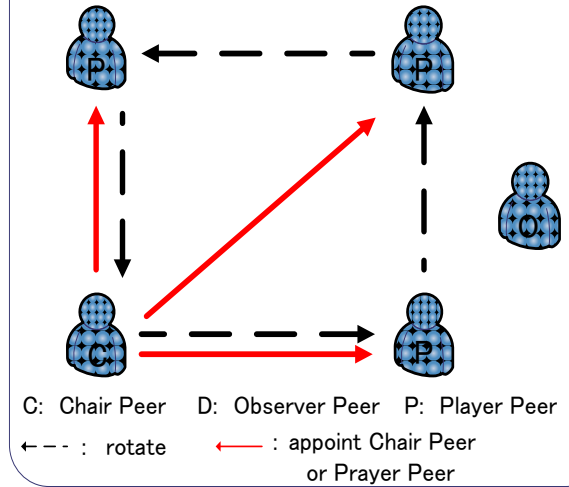
Discussion Mode



Presentation Mode



Rotation Mode

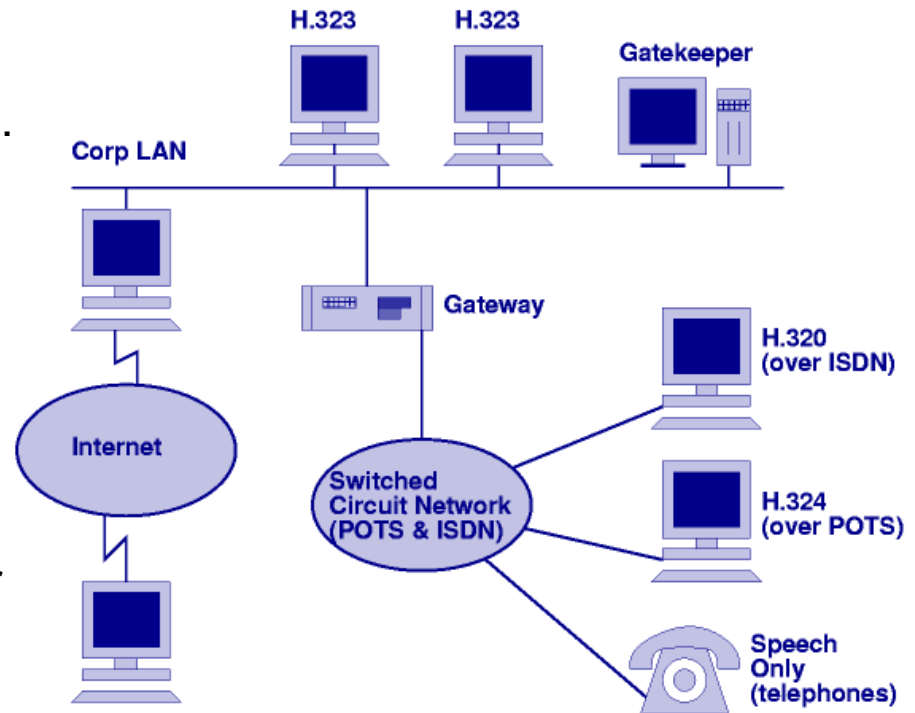


# Typical Standards: H.323 & SIP

- Self-developed communication software/middleware
- Implementations of Internet telephony and conference can use two types of popular standards
  - **H.323** standards from ITU (1996, 1<sup>st</sup> Version)
    - \* Adopt some protocols (RTP/RTCP) from IETF
    - \* More implementations
    - \* Very complex
    - \* Poor interoperability between vendors
  - **SIP** standards from IETF (1998, 1<sup>st</sup> Version)
    - \* Similar functions as H.323
    - \* Relatively easy because of textual natural instead of binary
    - \* Better interoperability
    - \* Under going and improvement, e.g., security

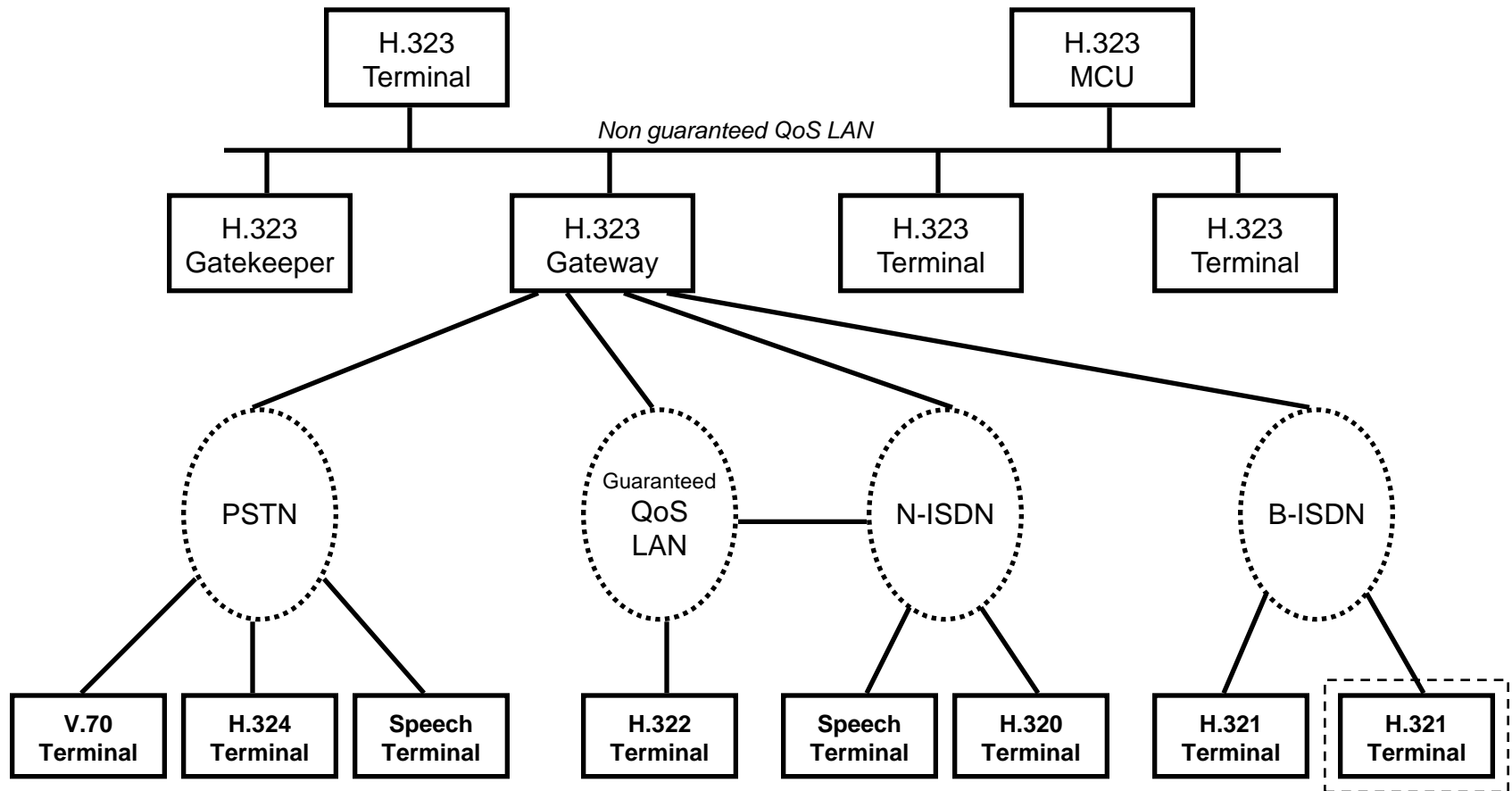
# H.323 History

- H.323 is a product of ITU-T Study Group 16.
- Version 1: “*visual telephone systems and equipment for LANs that provide a nonguaranteed quality of service (QoS)*” was accepted in October 1996.
  - Focus on multimedia communication in a LAN
  - No support for guaranteed QoS
- Version 2: “*packet-based multimedia communications systems*” was driven by the Voice-over-IP requirements and was accepted in January 1998.
- Version 3 was accepted in September 1999 and has minor incremental features (caller ID, ...) over version 2.
- Version 4 was accepted in November 2000 and has significant improvements over version 3.



# H.323 System

H.323 Entities: Terminal, Gatekeeper, Gateway, MCU (Multipoint Control Unit)



- H.310 (B-ISDN)
- H.320 (N-ISDN)
- H.321 (ATM)

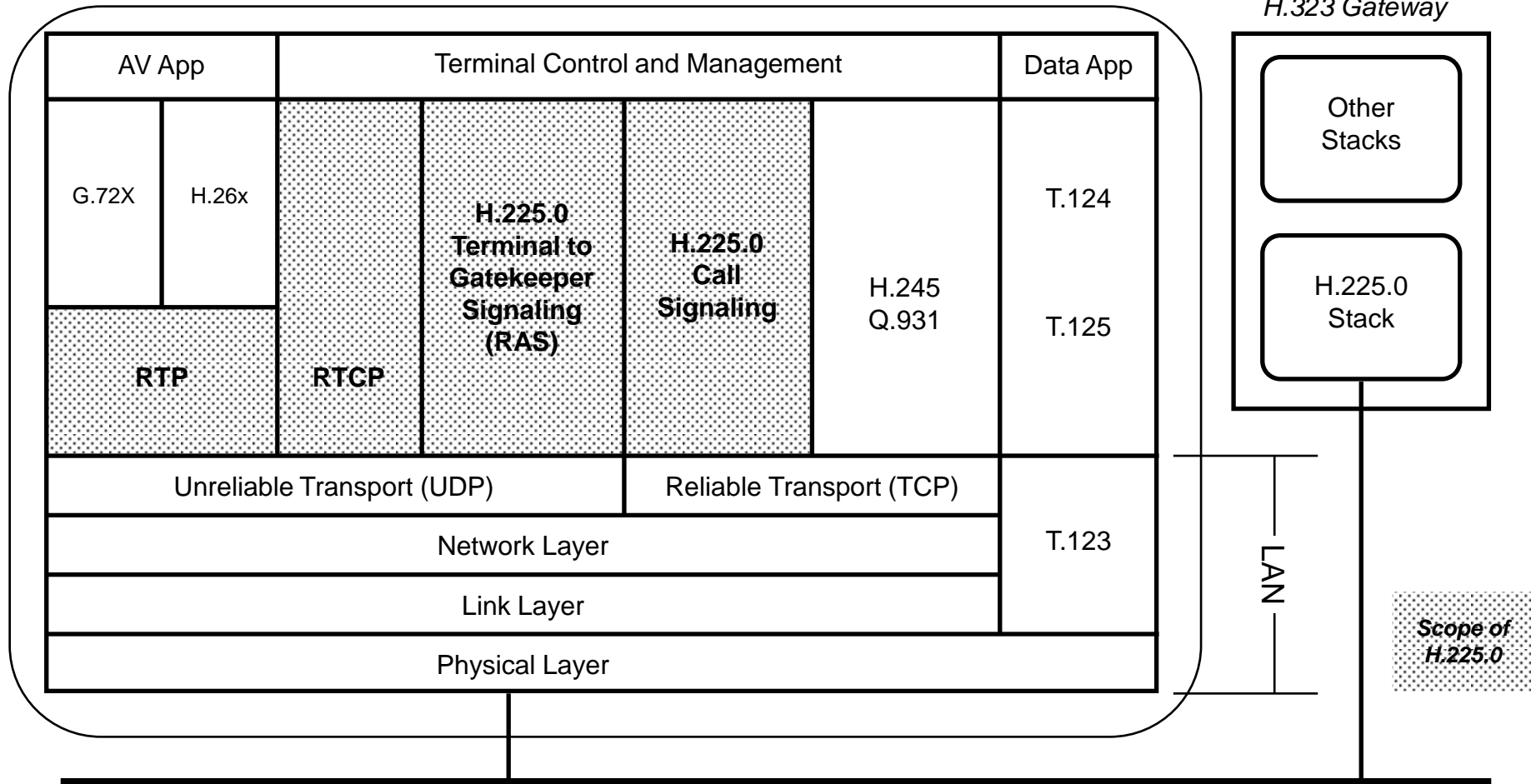
- H.322 (GQOS-LAN)
- H.324 (GSTN), H.324/M (mobile phone, 1998)
- V.70 (DSVD - Digital Simultaneous Voice & Data)

# H.323 Entities

- **Terminal**
  - An endpoint on the LAN which provides for real-time, two-way communications with another H.323 terminal, Gateway, or MCU
  - May provide audio, video, and/or data
- **Gatekeeper**
  - Provides address translation and controls access to the LAN
  - Performs bandwidth management
- **Multipoint Control Unit (MCU)**
  - Provides the capability for 3 or more terminals and Gateways to participate in a multipoint conference
- **Gateway**
  - Provides for real-time, two-way communication between H.323 terminals on a LAN and other ITU terminals on a wide-area network or another H.323 Gateway

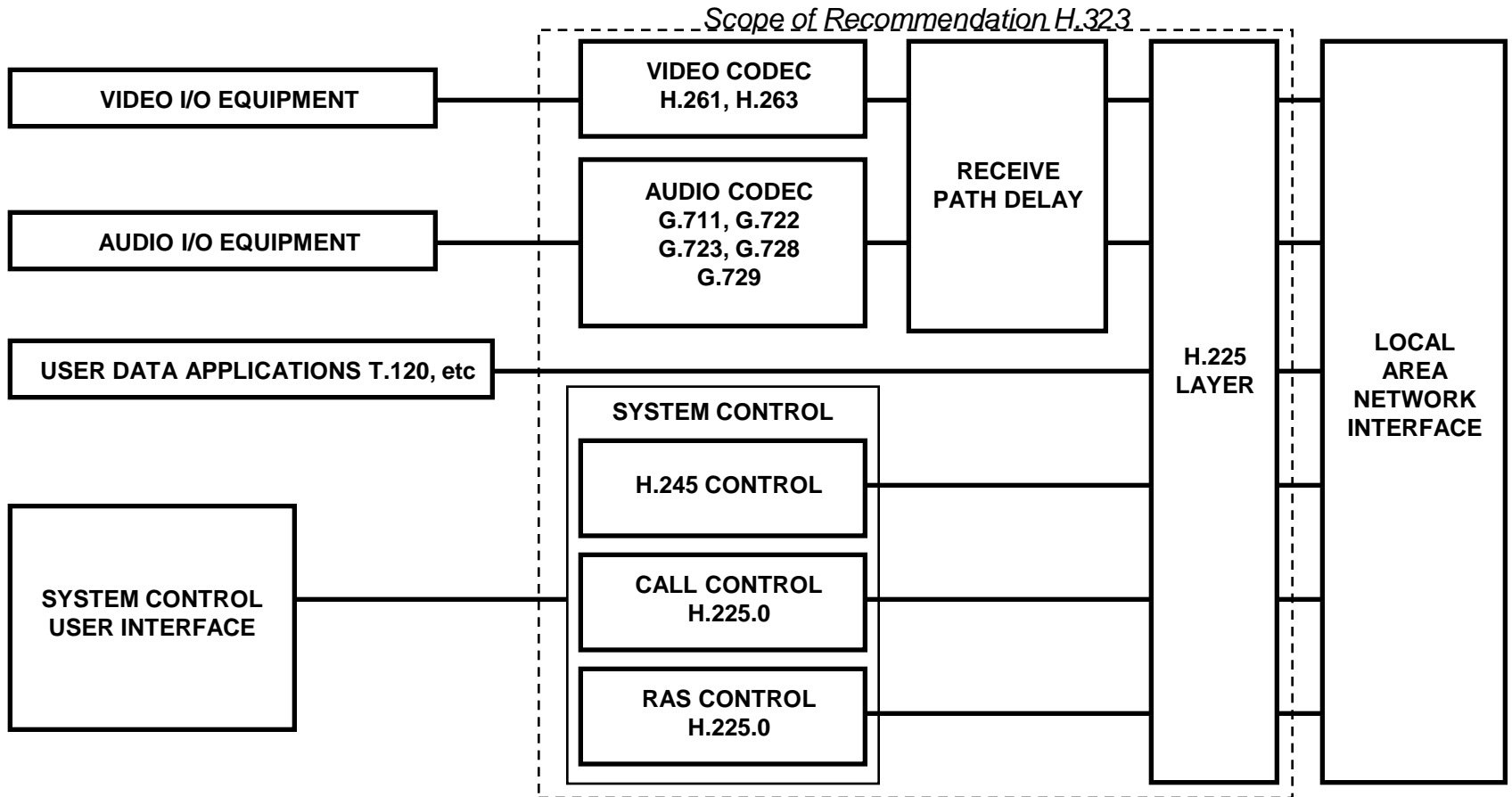
# H.323 Protocol Stack

*H.323 Protocol Stack*



**RAS:** Registration, Admission, Status

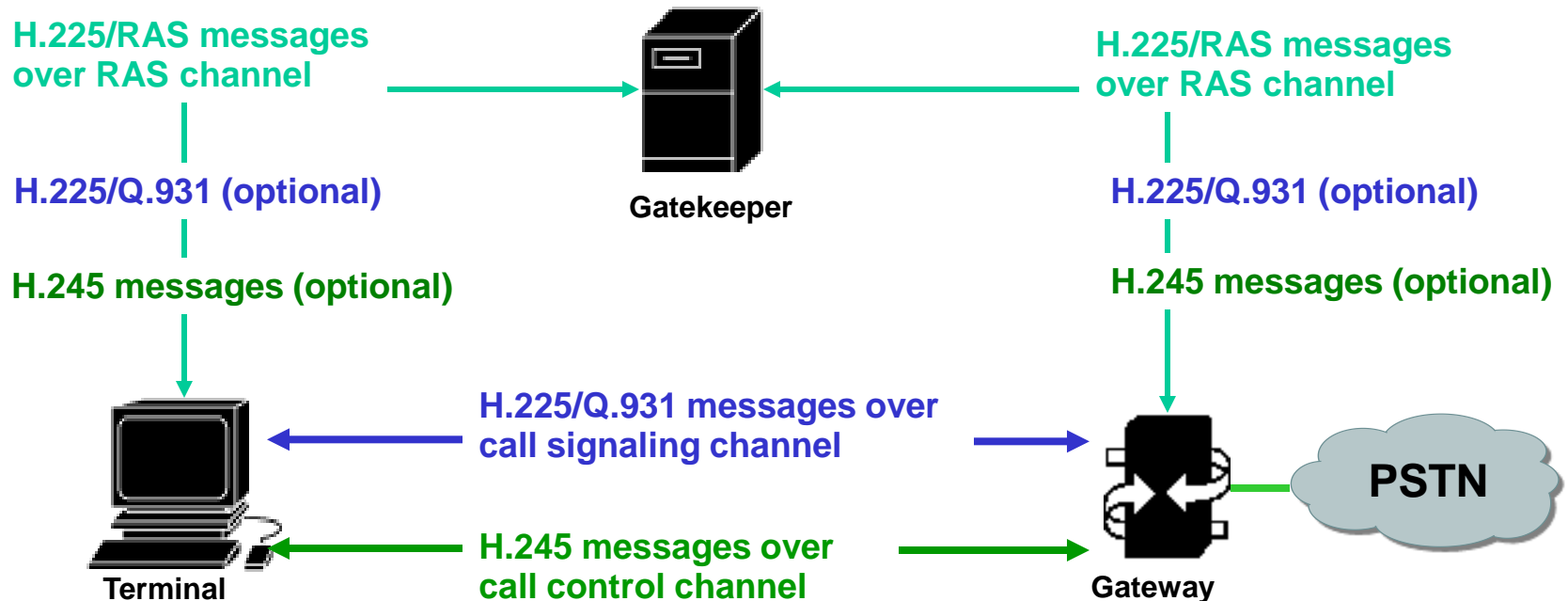
# H.323 Terminal



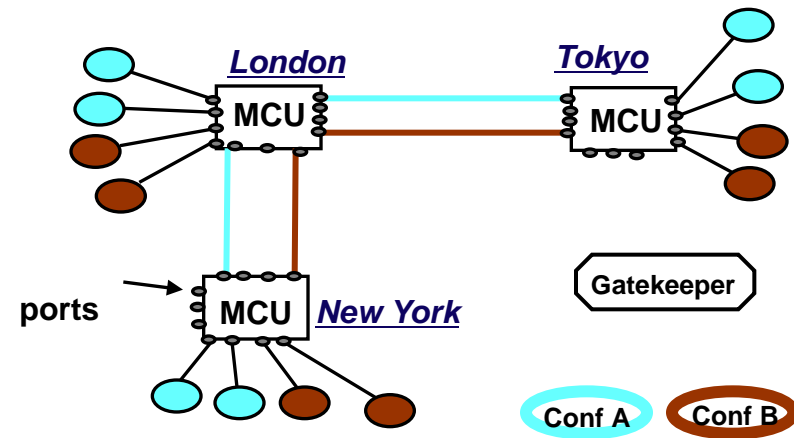


# Gatekeeper

- Provides the following services:
  - Address translation between Transport Addresses and Alias Addresses
    - # Transport Addresses: LAN IP Address + TSAP Identifier (port number)
    - # Alias Addresses: phone number, user name, email address, etc.
  - Admission control based on authorization, bandwidth, or other criteria
  - Dynamic bandwidth control during a conference
- Transport address for the H.245 Control Channel is exchanged on the Call Signaling Channel



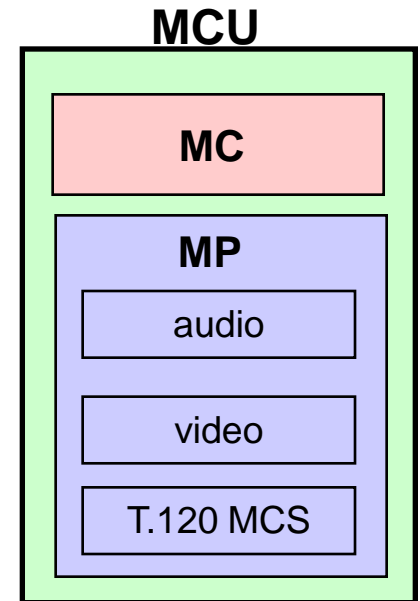
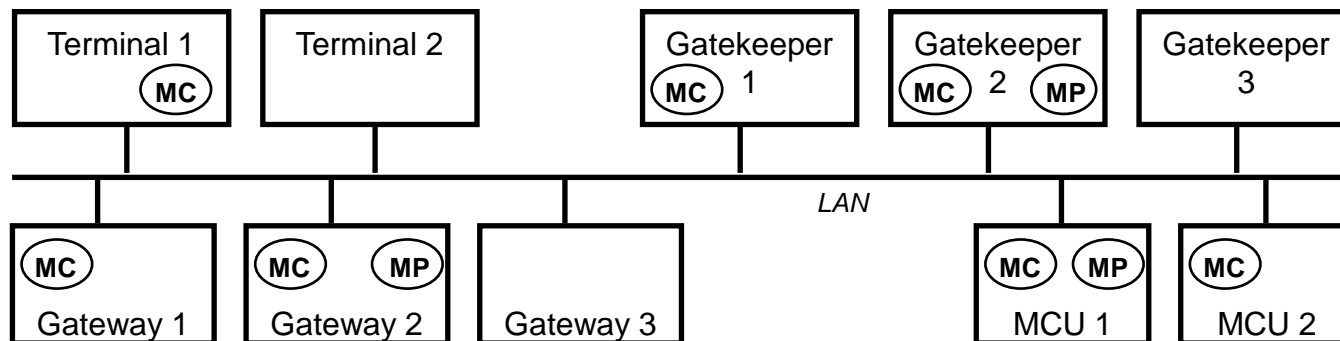
# Multipoint Entities & MCU



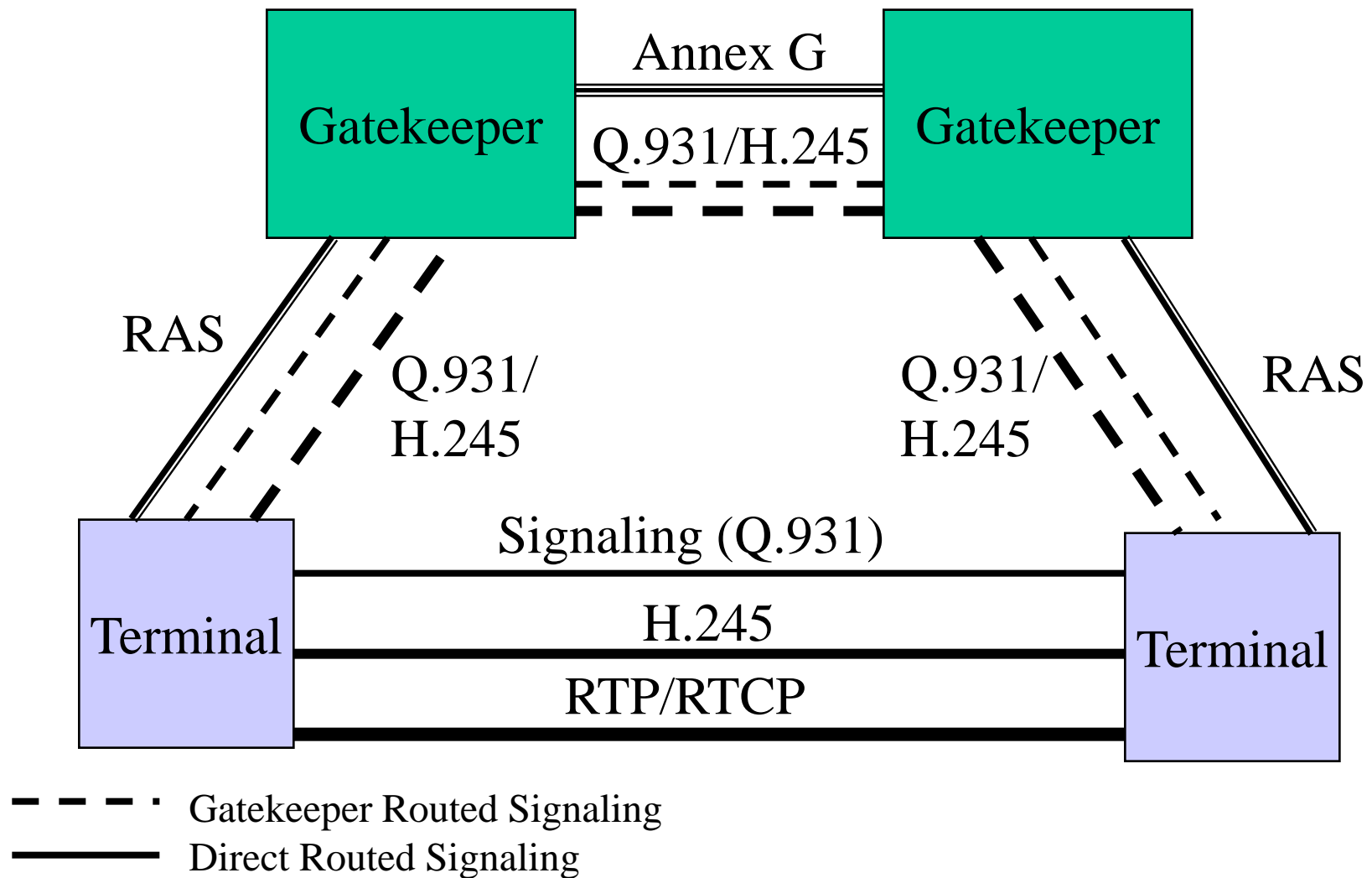
**MC:** Multipoint Controller, **MP:** Multipoint Processor

- **MC** performs capability exchanges with each endpoint and determines the media format used in a conference
  - Assigns terminal numbers to each endpoint in the conference
  - Maintains a list of all conference participants
- **MP** is used for processing of audio/video/data streams in a centralized or hybrid multipoint conference

**Note:** - *MC/MP may be co-located with a Gateway or Gatekeeper*  
- *Gateway, Gatekeeper and MCU may be a single device*

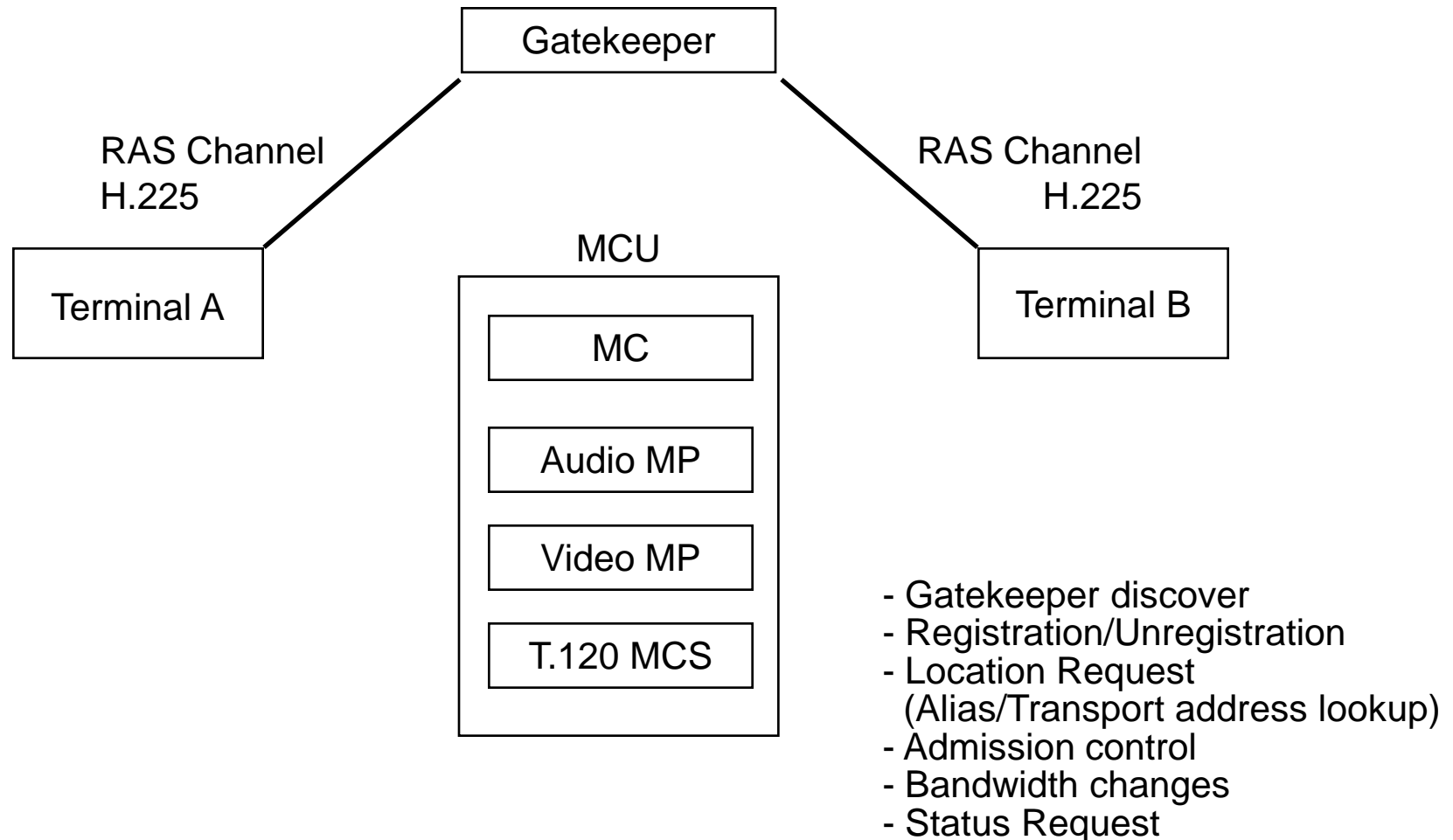


# H.323 Basic Protocols for VoIP



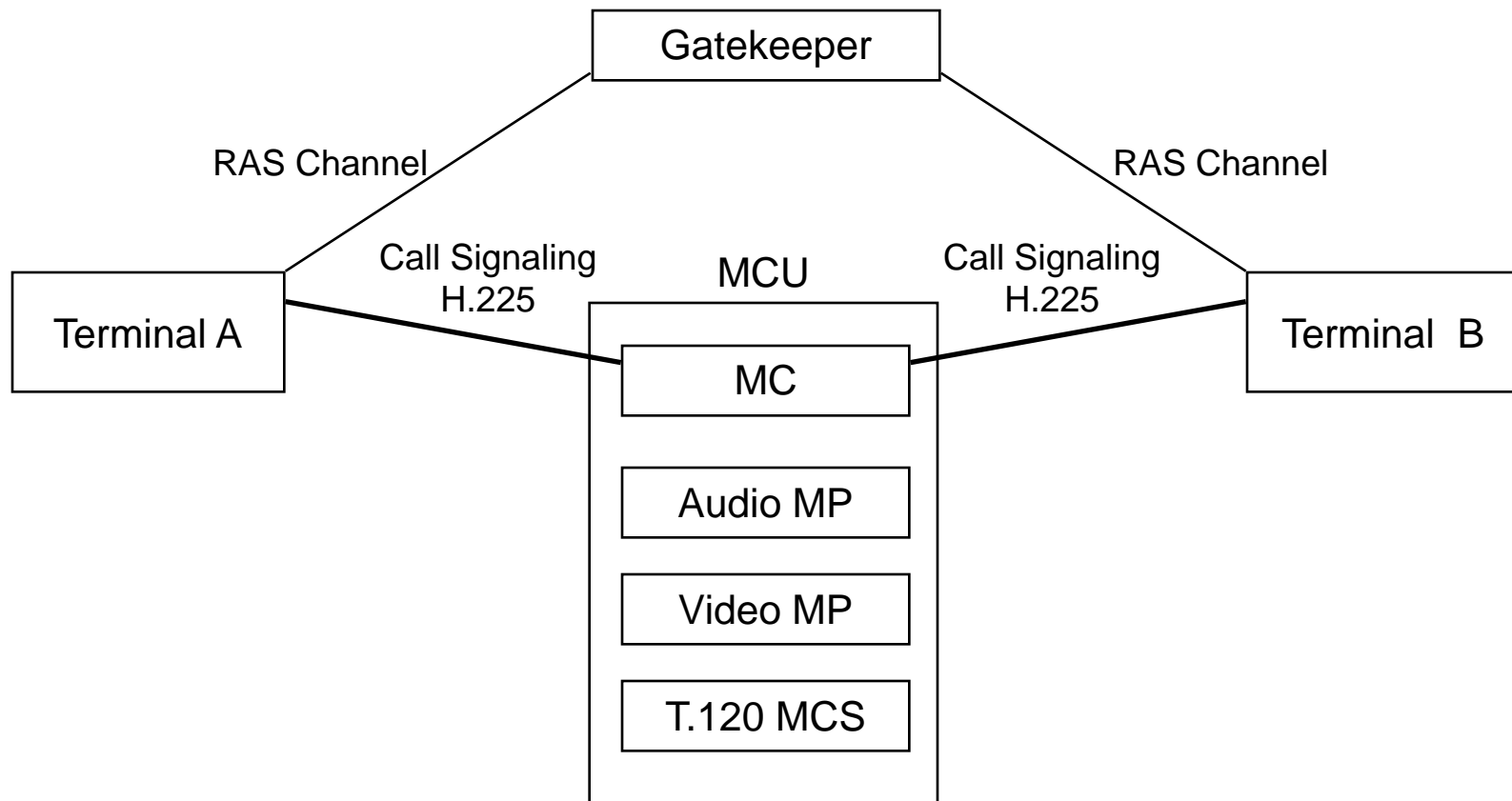
# H.323 VoIP Call Setup Procedures (1)

- Step 1: Endpoint - Gatekeeper communication



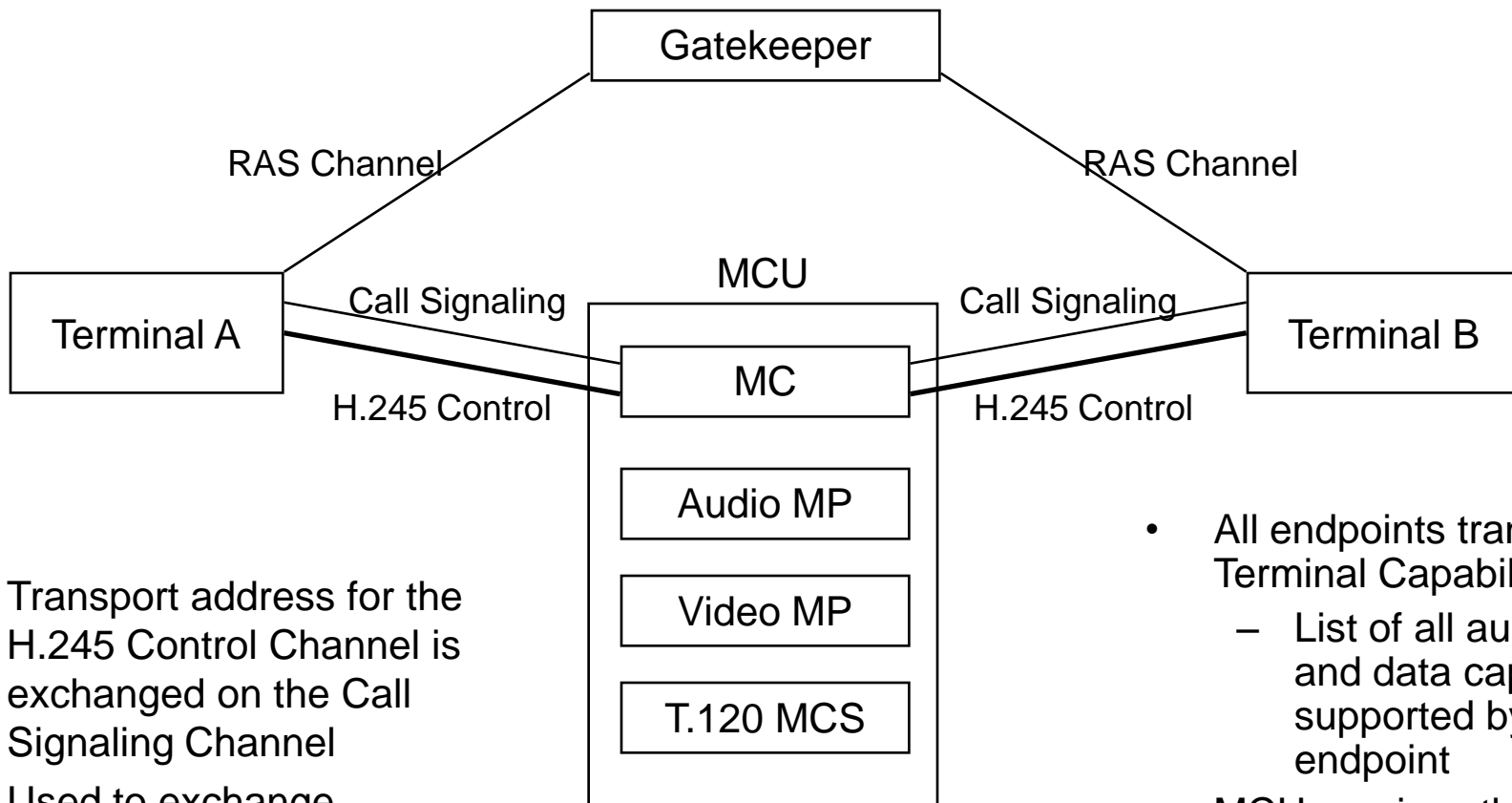
# H.323 VoIP Call Setup Procedures (2)

- Step 2: Setup initial connection with the MCU using the Call Signaling Channel via gatekeeper



# H.323 VoIP Call Setup Procedures (3)

- Step 3: Setup H.245 Control Channel with the MCU

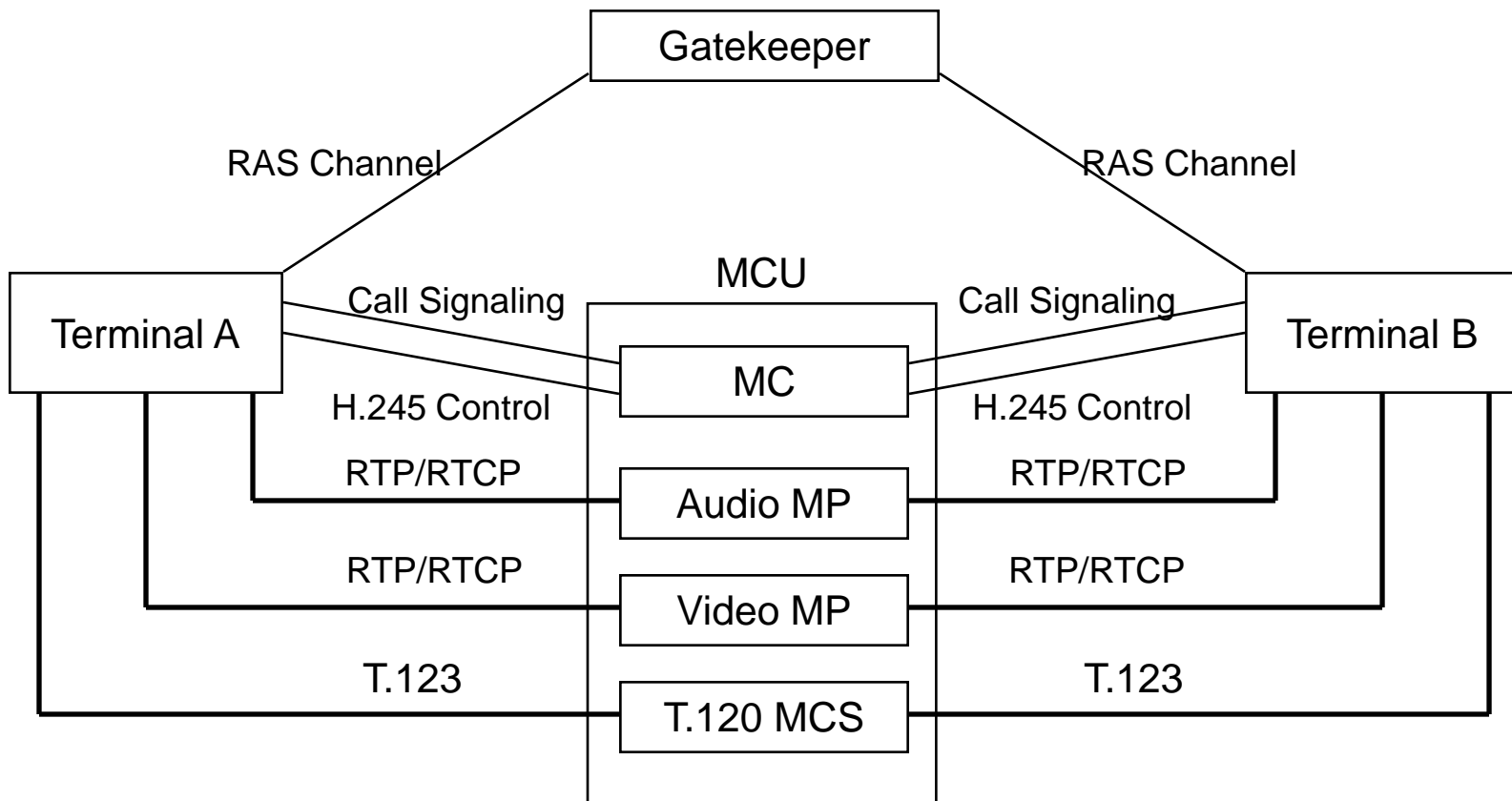


- All endpoints transmit a Terminal Capability Set
  - List of all audio, video, and data capabilities supported by the endpoint
- MCU receives the capabilities and determines the Selected Communication Mode (SCM)

- Transport address for the H.245 Control Channel is exchanged on the Call Signaling Channel
- Used to exchange capabilities, create logical channels, and exchange multipoint commands

# H.323 VoIP Call Setup Procedures (4)

Step 4: Setup additional logical channels for audio/video/data



# T.120 Multipoint Data Conferencing

- T.120 defines multipoint data communications standards in a multimedia conferencing environment
- Provides mechanism to identify the participating nodes and exchange information
- Enables multiple simultaneous conference handling and participation
- Consists of a set of protocols:

## Core Protocols:

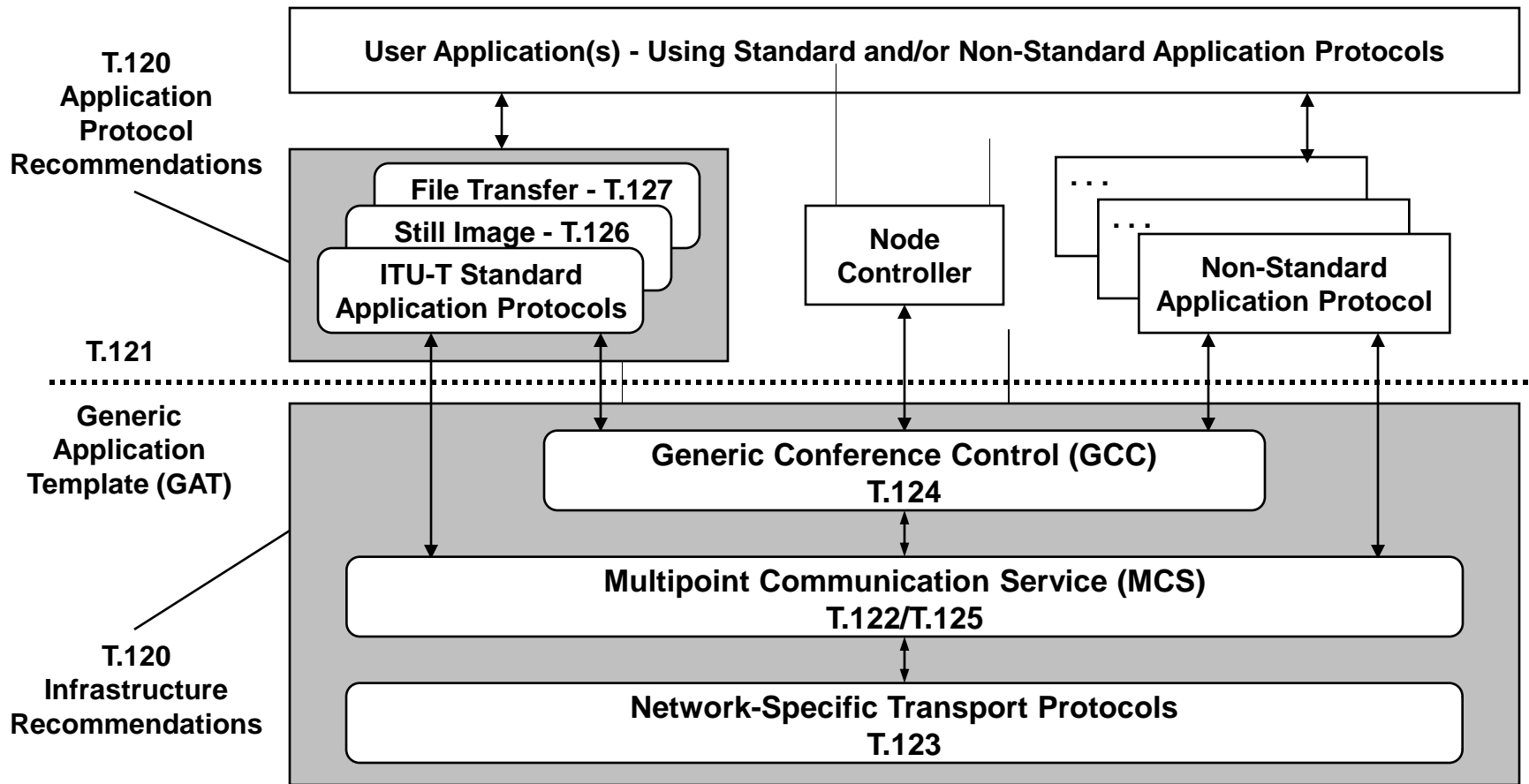
- T.123: Transport Protocol
- T.124: Generic Conference Control (GCC)
- T.125/T.122 Multipoint Communication Service (MCS)

## Optional Protocols

- T.121: Generic Application Template (GAT)
- T.126: MultiPoint Still Image and Annotation Protocol (NSIA)
- T.127: Multipoint Binary File Transfer Protocol (MBFT)
- T.128: Application Sharing (AS)

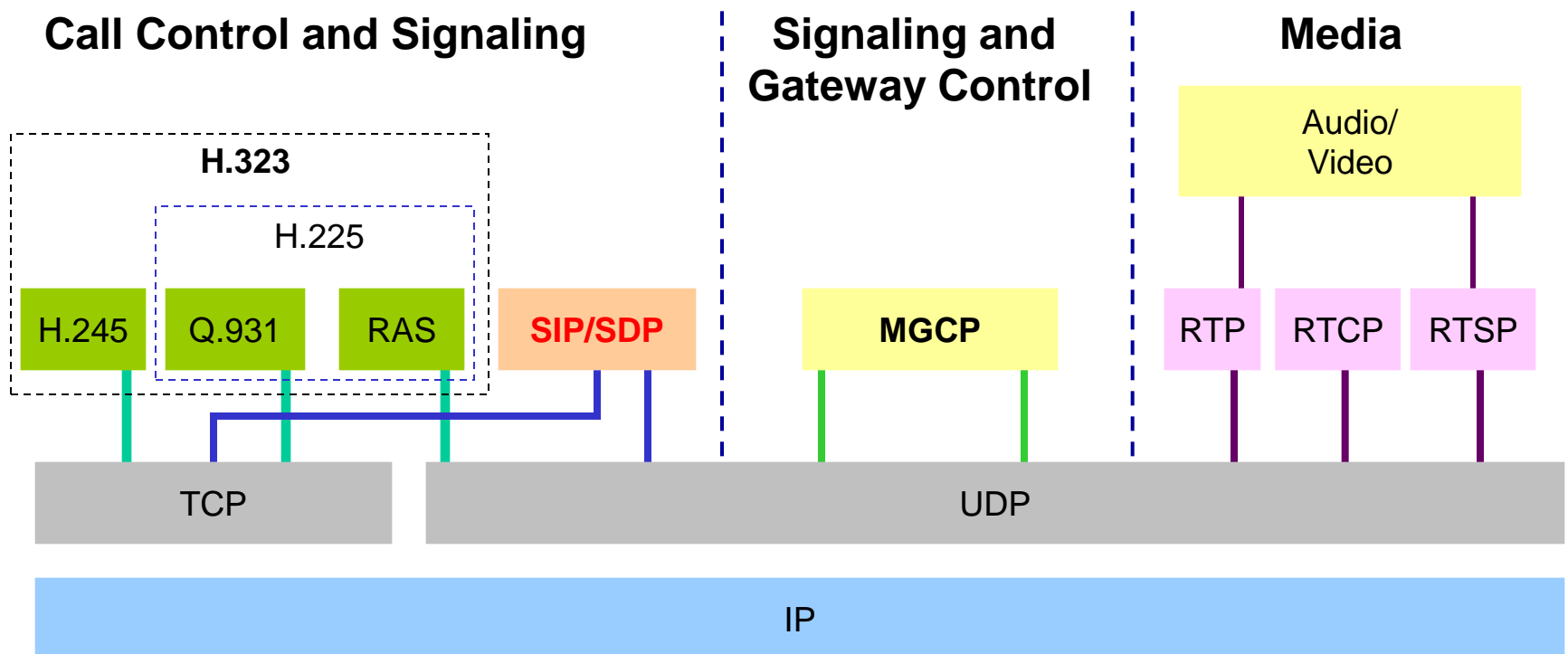


# T.120 System Model



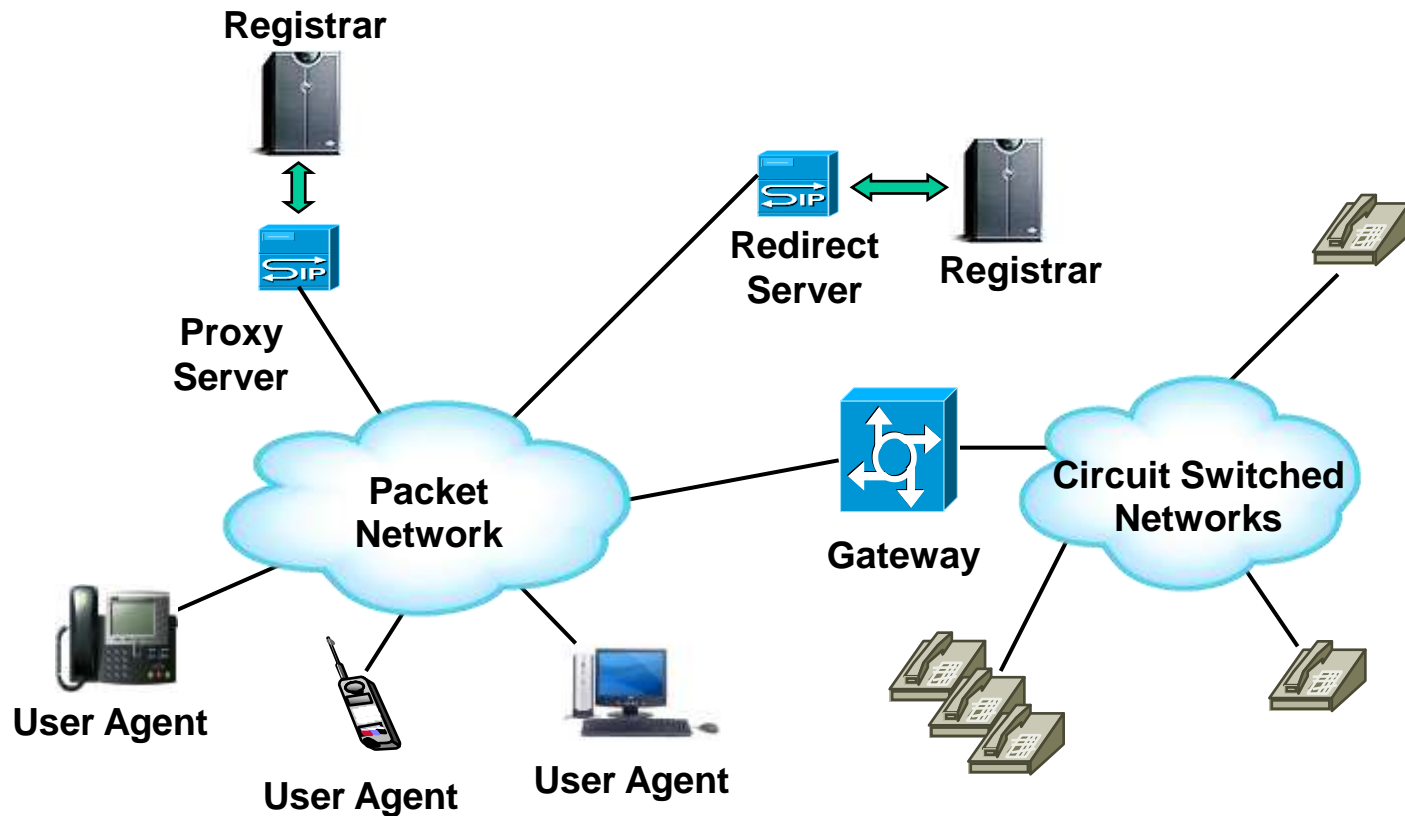
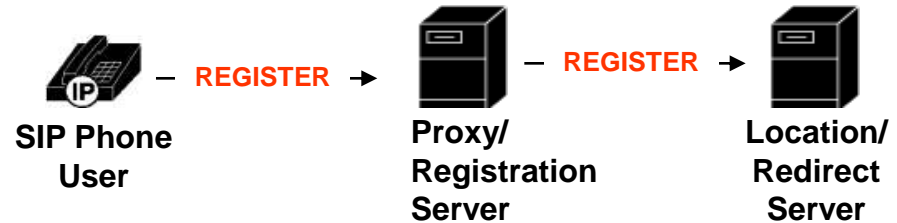
# Alternative: SIP/SDP

- The **Session Initiation Protocol** (**SIP**, RFC 2543) has been proposed as an alternative to H.323
- SIP is capable of negotiating a call
- SDP is used to describe capabilities: media, coding, protocol, address/port, crypto key
- Media still runs over RTP
- Each has merits and demerits, but quite similar

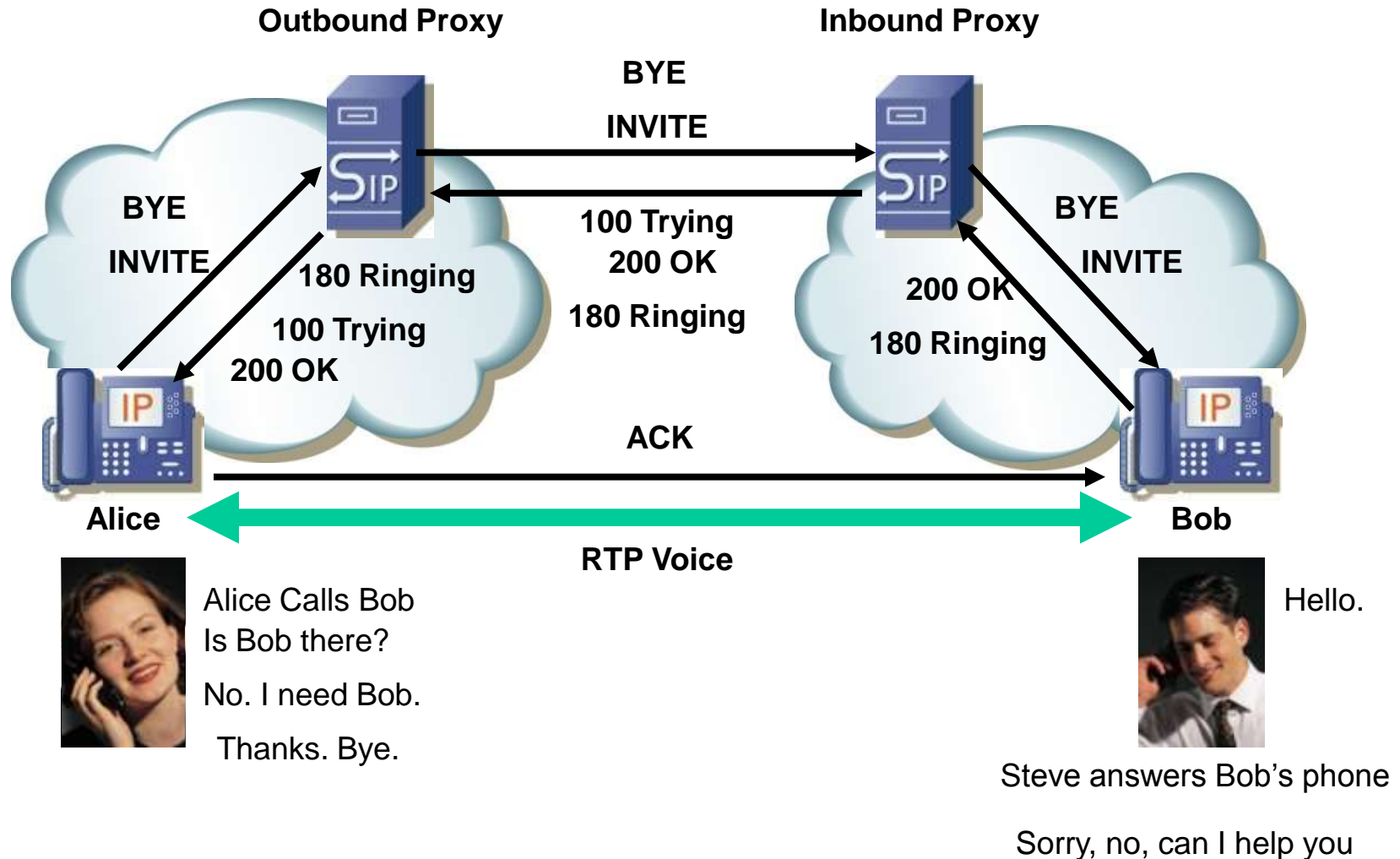


# SIP Entities and Architecture

- H.323 terminal → SIP user agent
- H.232 gatekeeper  
→ SIP server: proxy, registrar, redirect
- H.232 gateway → SIP gateway

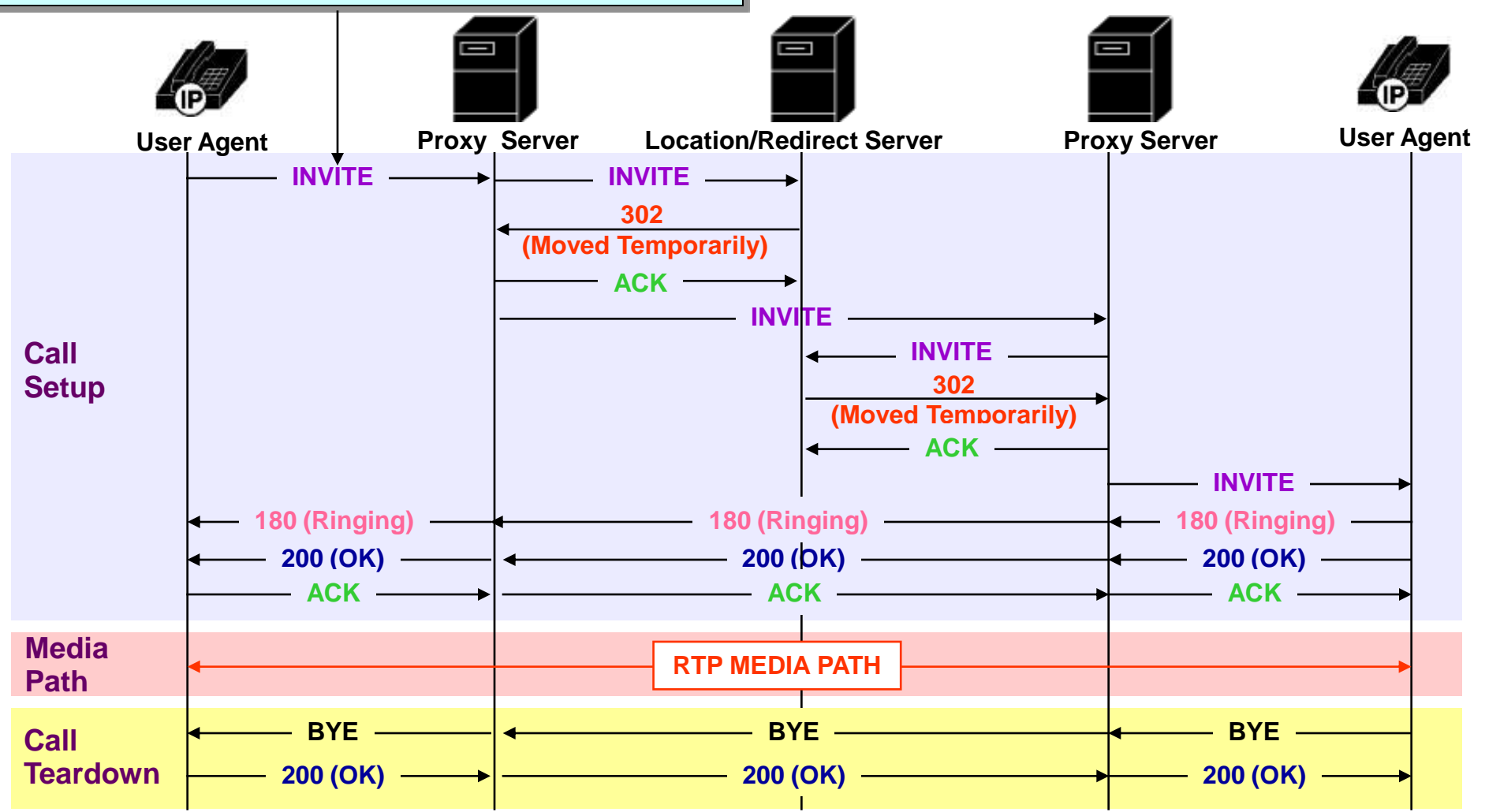


# SIP Call Flow

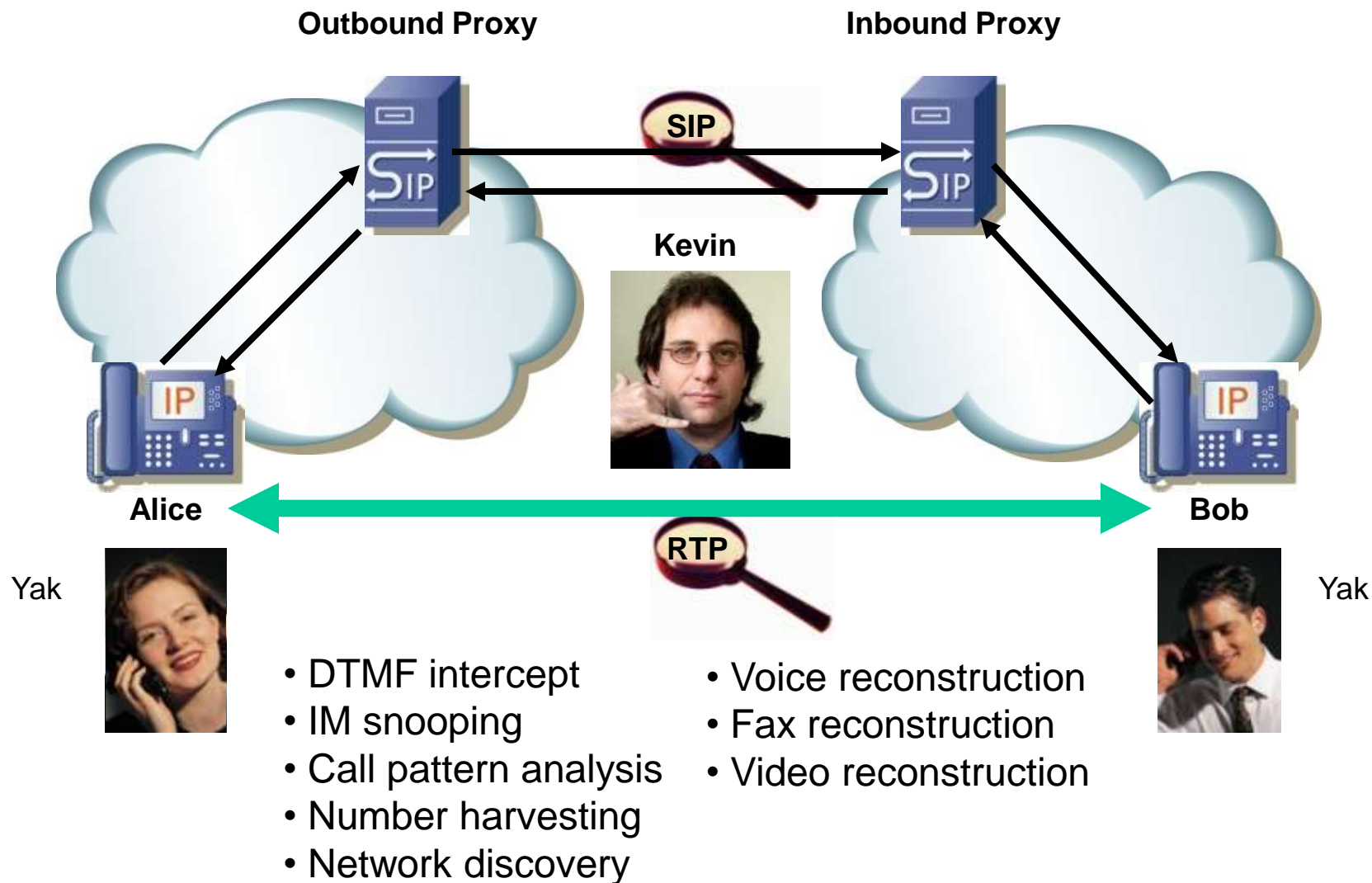


```
INVITE sip:bob@biloxi.com SIP/2.0
Via: SIP/2.0/UDP pc33.atlanta.com;branch=z9hG4bK776asdhds
Max-Forwards: 70
To: Bob <sip:bob@biloxi.com>
From: Alice <sip:alice@atlanta.com>;tag=1928301774
Call-ID: a84b4c76e66710@pc33.atlanta.com
CSeq: 314159 INVITE
Contact: <sip:alice@pc33.atlanta.com>
Content-Type: application/sdp
Content-Length: 142
```

# SIP Detailed Call Setup and Teardown



# VoIP Communication Security



# Demos of Skype for Phone Call and Tele-Meeting

# Media Retrieval

- Information Retrieval
- Image Retrieval
- Video Retrieval
- Audio Retrieval

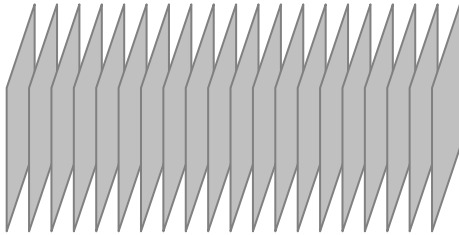


# Information Retrieval

- ❑ Retrieval = Query + Search
- ❑ Informational Retrieval: Get required information from database/web
- ❑ Text data retrieval
  - via keyword searching in a text document or through web
  - via expression such as in relational database
- ❑ Multimedia retrieval
  - Get similar images from an image database
  - Find interesting video shots/clips from a video/database
  - Select news from video/radio Internet broadcasting
  - Listen specific sound from audio database
  - Search a music
- ❑ Challenges in multimedia retrieval
  - Can't directly text-based query and search?
  - How to analysis/describe content and semantics of image/video/audio?
  - How to index image/video/audio contents?
  - Fast retrieval processing and accurate retrieval results

# Audio Visual Content/Feature

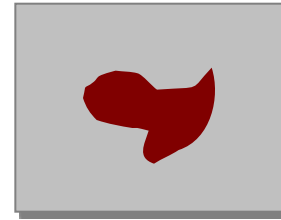
## Video segments



### Content/ Features

- Color
- Camera motion
- Motion activity
- Mosaic

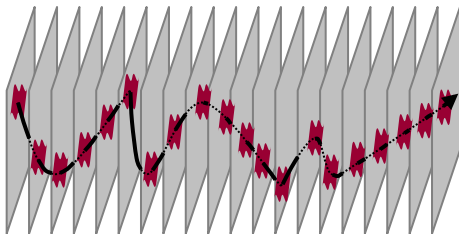
## Still regions



### Content/ Features

- Color
- Shape
- Position
- Texture

## Moving regions



### Content/ Features

- Color
- Motion trajectory
- Parametric motion
- Spatio-temporal shape

## Audio segments



### Content/ Features

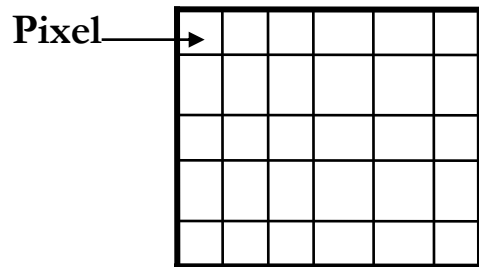
- Spoken content
- Spectral characterization
- Music: timbre, melody, pitch

# Image Content – Image Features

- What are image features?
- Primitive features
  - Mean color (RGB)
  - Color Histogram
- Semantic features
  - Color distribution, texture, shape, relation, etc...
- Domain specific features
  - Face recognition, fingerprint matching, etc...

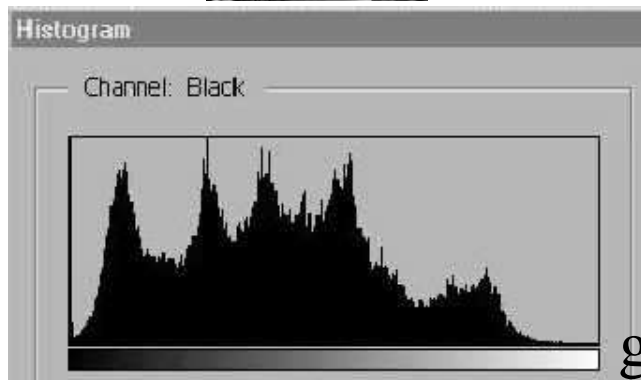
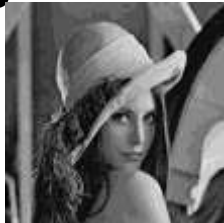
# Mean Color and Color Histogram

- Pixel Color Information: R, G, B
- **Mean Color** (R,G or B) =  $\frac{\text{Sum of that component for all pixels}}{\text{Number of pixels}}$



Number of pixels

- **Histogram:** Frequency count of each individual color

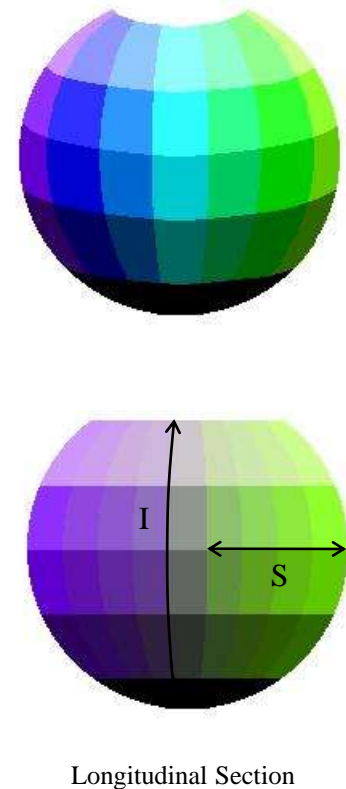
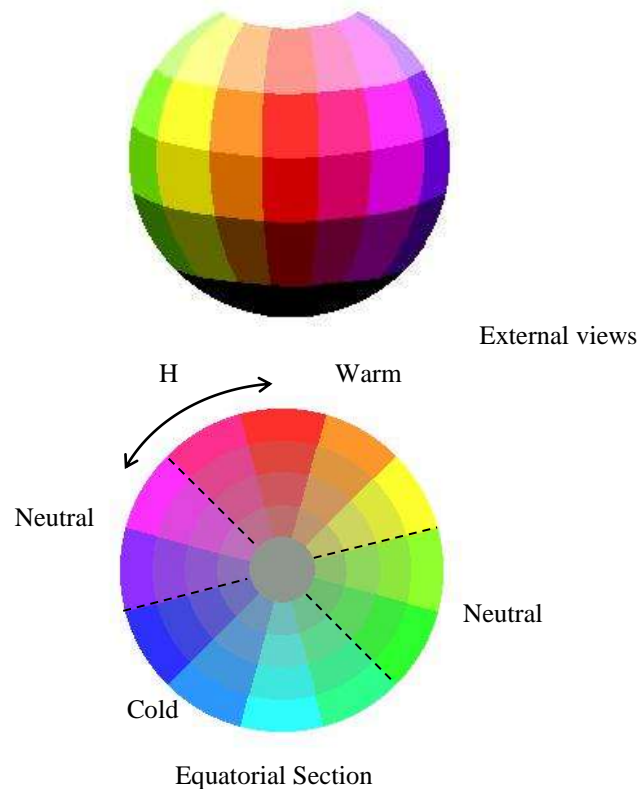
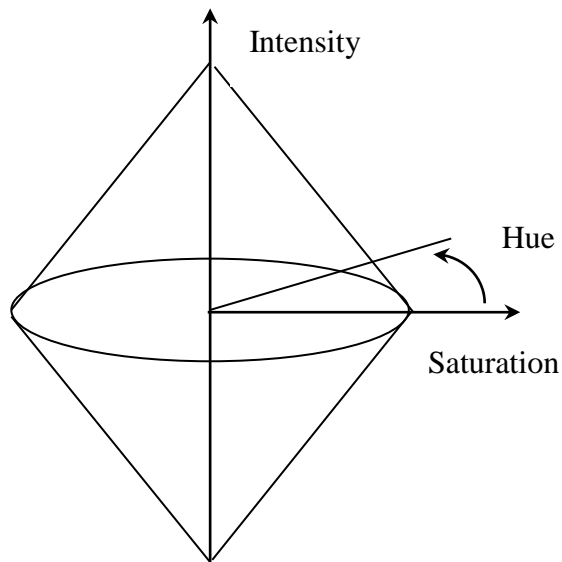


gray



# Color Models and HSI

- Many color models: RGB, CMY, YIQ, YUV, YCrCb, HSV, HSI, ...
- **HSI (Hue, Saturation, Intensity): often used**



# Similarity between Two Colors

The similarity between two colors, i and j, is given by:

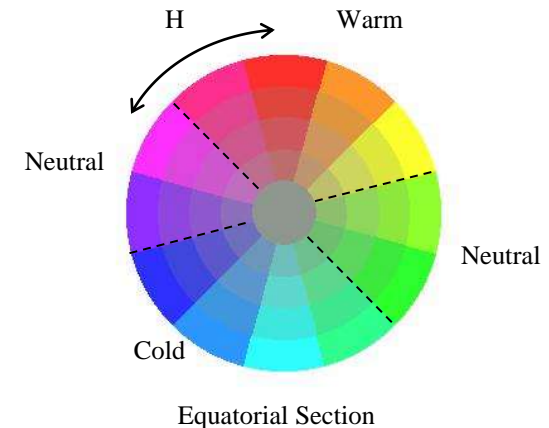
$$C(i, j) = W_h H(i, j) + W_s S(i, j) + W_i I(i, j)$$

where

$$H(i, j) = \min(|H_i - H_j|, 12 - |H_i - H_j|)$$

$$S(i, j) = |S_i - S_j|$$

$$I(i, j) = |I_i - I_j|$$



The degree of similarity between two colors, i and j, is given by:

$$CS(i, j) = \begin{cases} 0 & \text{if } H(i, j) > H_{\max} \\ 1 - \frac{C(i, j)}{C_{\max}} & \text{otherwise} \end{cases}$$



# Content Based Image Retrieval (CBIR)

- ❑ CBIR: based on similarity of image color, texture, object shape/position
- ❑ Images with similar color → *dominated by blue and green*



botanic1



CnScenery9



botanic3



raffles1



exar



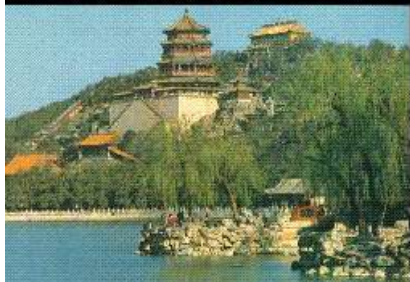
CnScenery7



shangrila



foothills



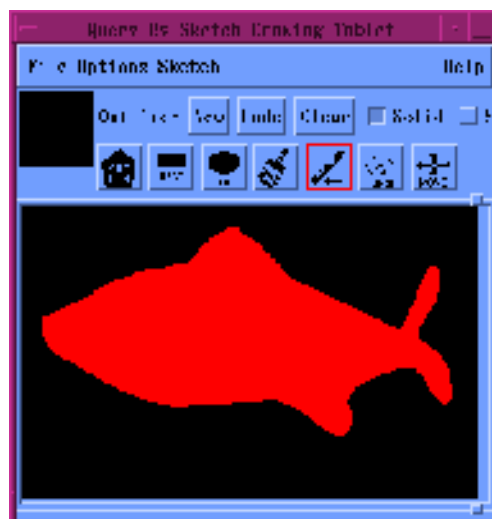
# Color Based Image Retrieval

Images with similar colors and distribution/histogram





# Shape Based Image Retrieval

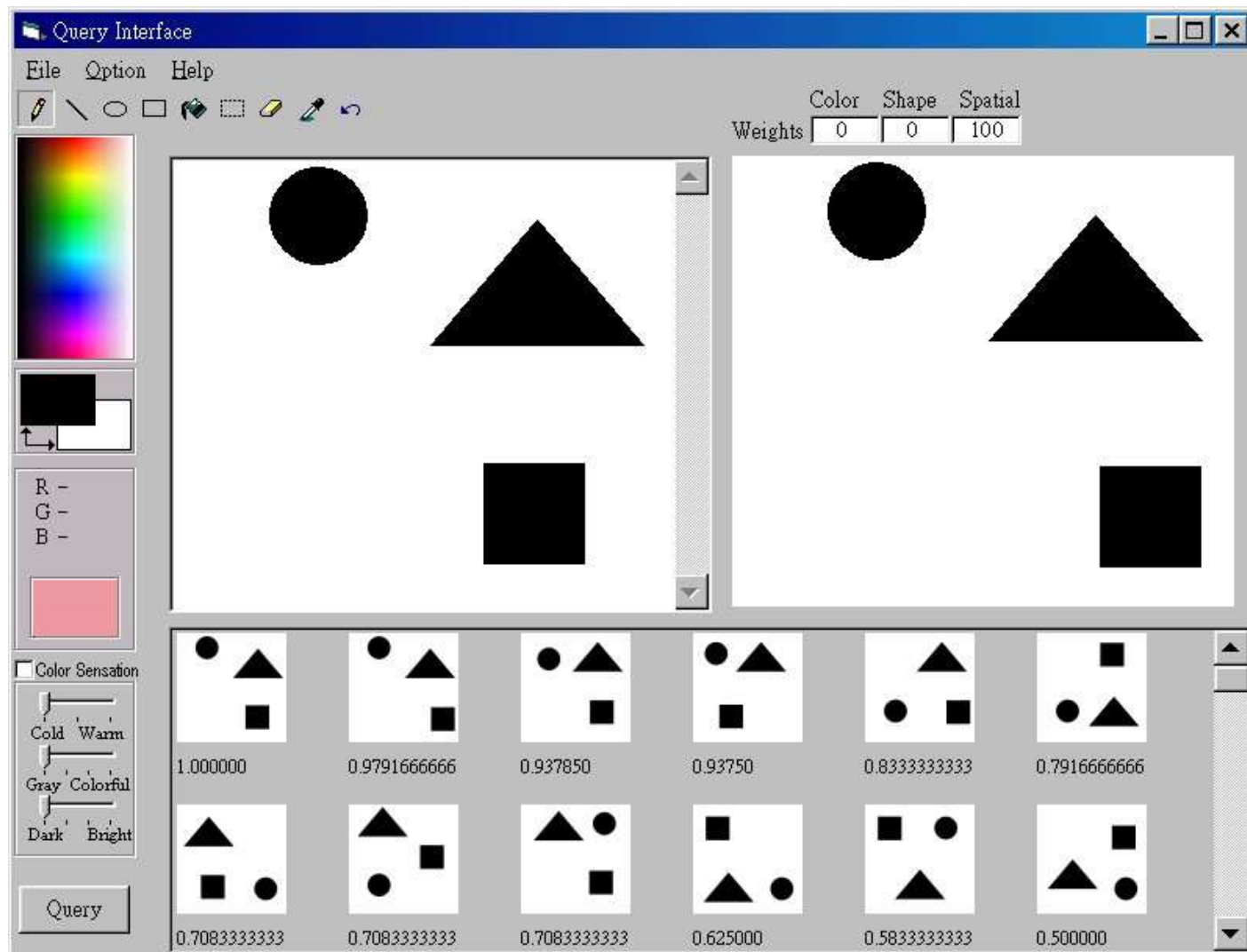


Images with similar shapes

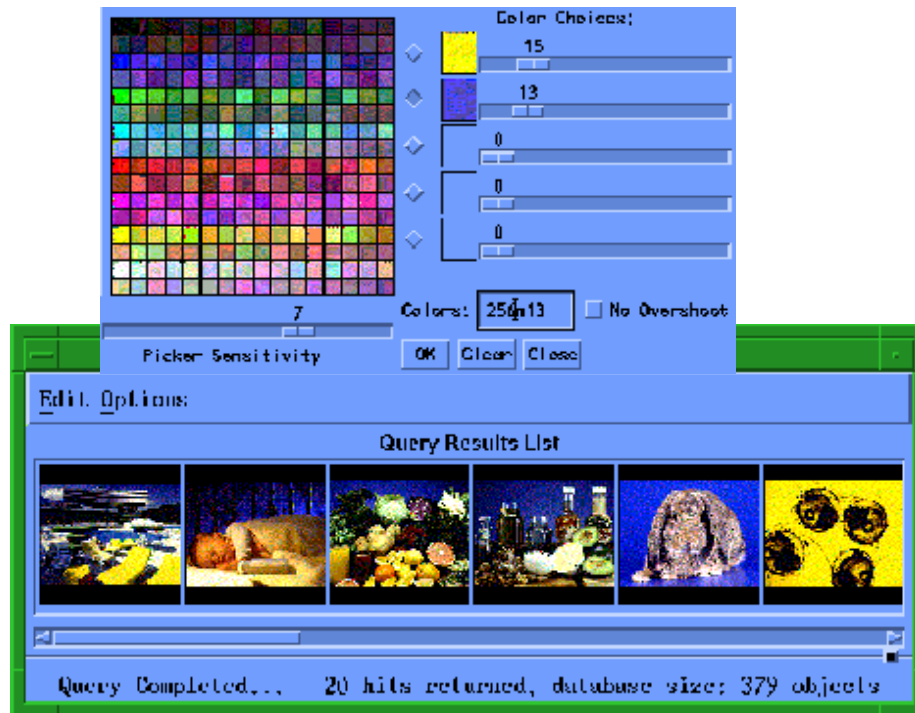


# Spatial Relation Based Image Retrieval

Images with similar shapes and their relation



# Correctness and Accuracy in CBIR



- ❑ CBIR accuracy is counted by a percentage of targeted/corrected image(s) in top-n candidate images, for example

$$\boxed{C_1, C_2, C_3, \dots, C_{n-1}, C_n} \quad C_{n+1}, \dots, C_M$$

90%

- ❑ Hybrid retrieval using color and texture plus shape can improve accuracy

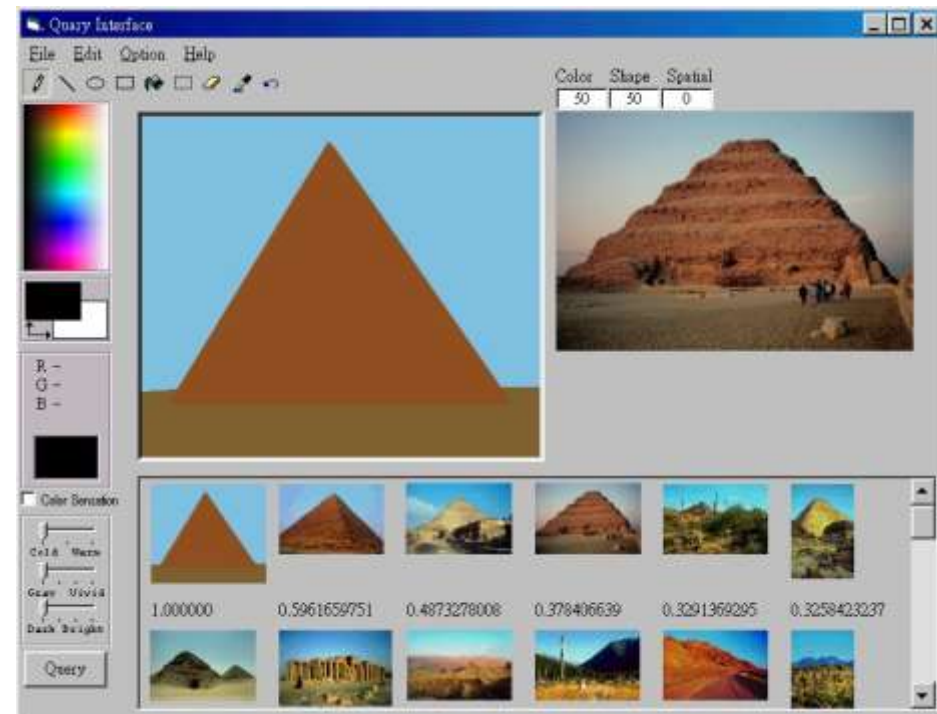
# Hybrid Retrieval – Combined Similarity

- ◆ The Similarity Measure of Color:  $CS$
  - ◆ The Similarity Measure of Shape:  $SS$
  - ◆ The Similarity Measure of Spatial Relation:  $SRS$
- **Combined Similarity Score:**

$$S = W_C * CS + W_S * SS + W_{SR} * SRS$$

Where  $CS$ ,  $SS$ ,  $SRS$  are the similarity scores of Color, Shape and Spatial Relations, and  $W_C$ ,  $W_S$ ,  $W_{SR}$  are the weights of Color, Shape and Spatial Relations

# Query by Scratch in CBIR



Please try such image search in the [Hermitage Web site](#). It uses the QBIC engine for searching archives of world-famous art.



# Query by Example in CBIR

Content-Based Image Reterival Sytem(CBIR)

Select Options | Search Options


Select the Image to be Searched on

☐ Random Browsing ☐ Upload Image

Number of Random Images: 24

Enter The full path of the Image:  Enter

Search On



Max no. of results: 40





Search Image




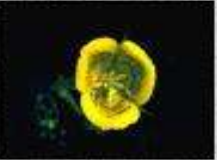
Time Required: 0.487629 secs

Select Image to Search

First  
Second  
Third  
Fourth  
Fifth  
Sixth  
Seventh  
Eighth

Retrieved -> 10

1  2  3  4 

5  6  7  8 

Navigation: Left Arrow, Right Arrow

# Query by Example in CBIR (cont.)



# Video Retrieval

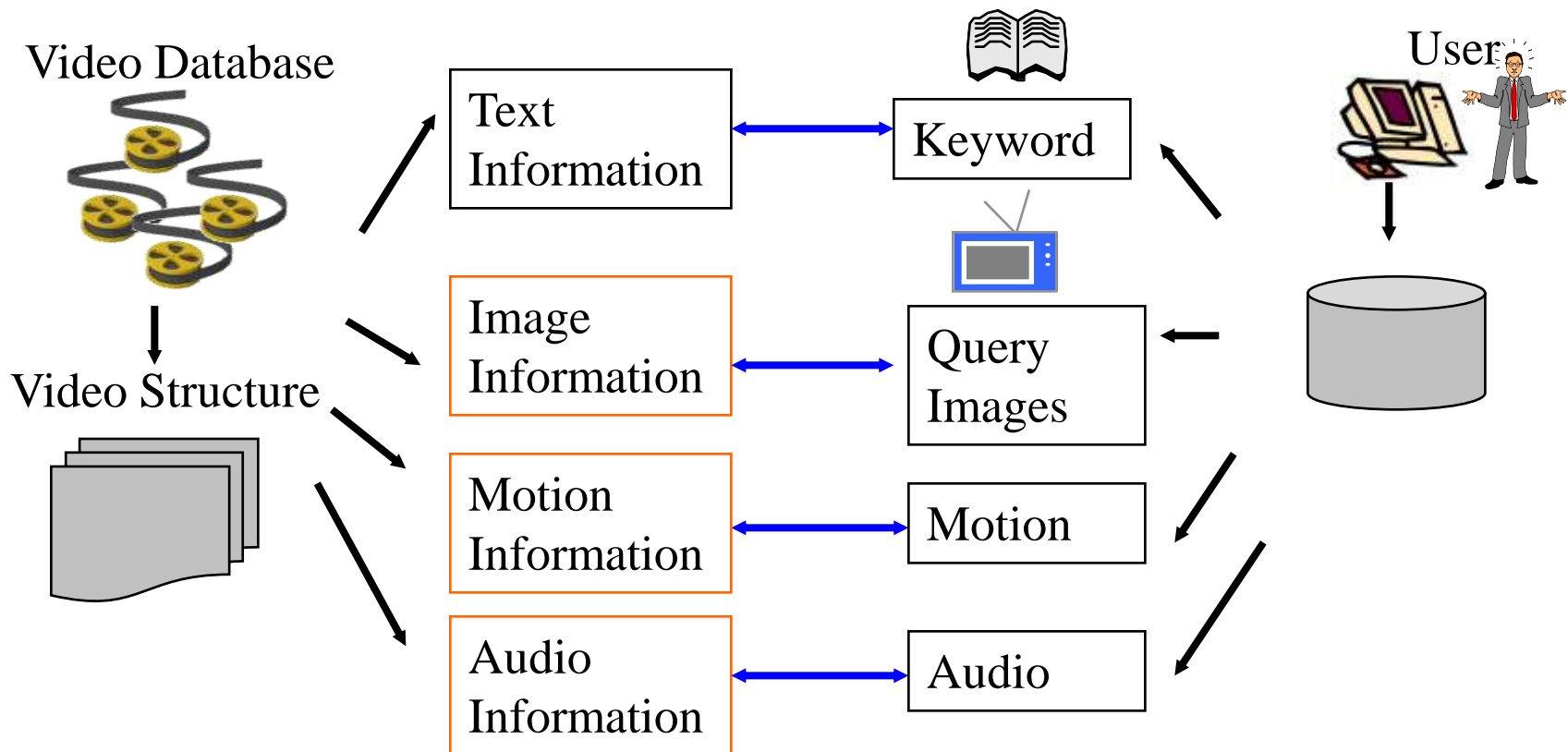
## ❑ Video retrieval:

- Find interesting video shots/segments from a movie, TV, video database
- It is hard because of many images ( $>10\text{fps}$ ) and temporal changes

## ❑ Methods of video retrieval

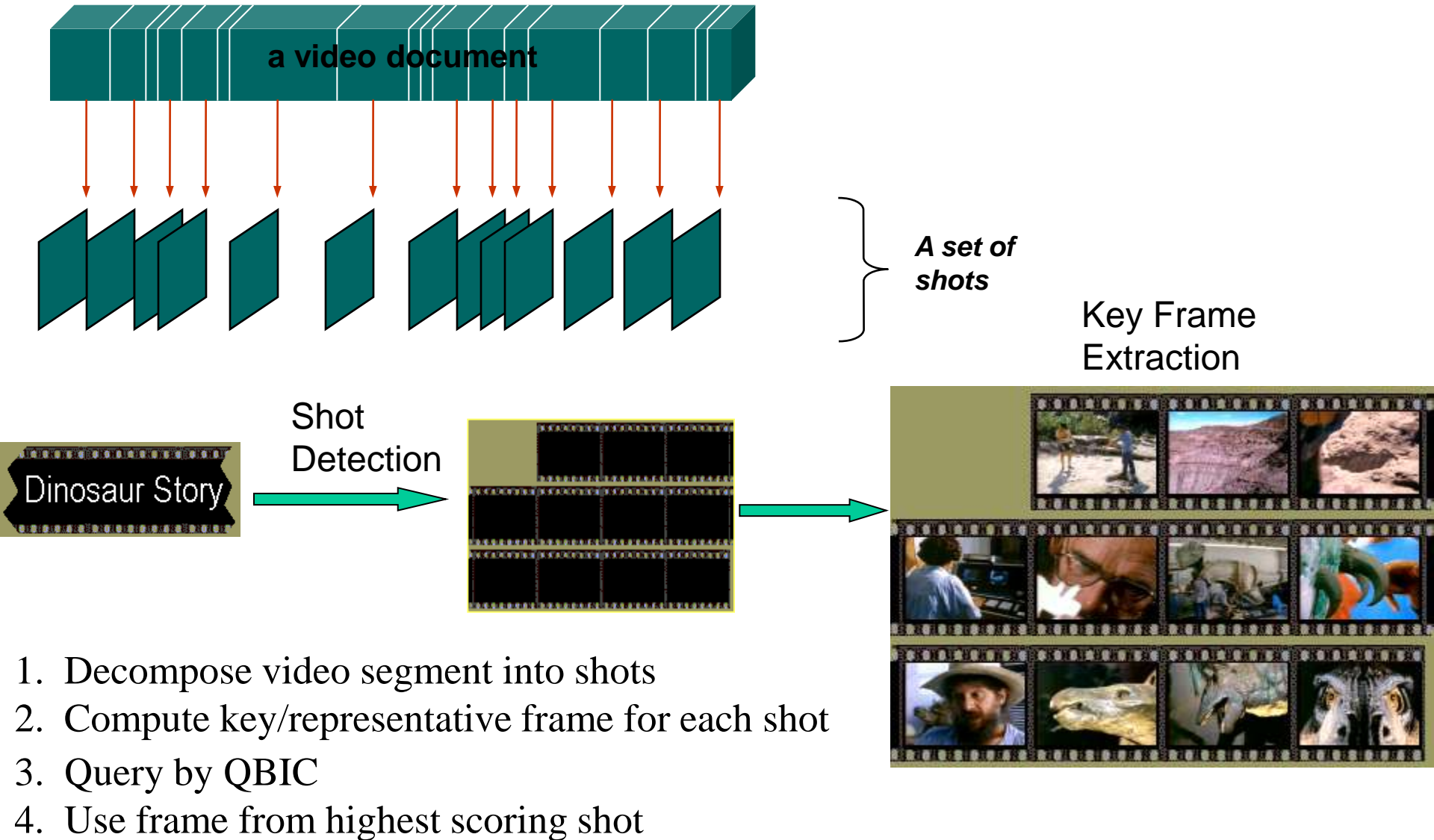
**Non-text-based:** Key frames via CBIR, color, object, background sound, etc.

**Text-based:** Extract caption, i.e., overlaid text, speech recognition, etc.

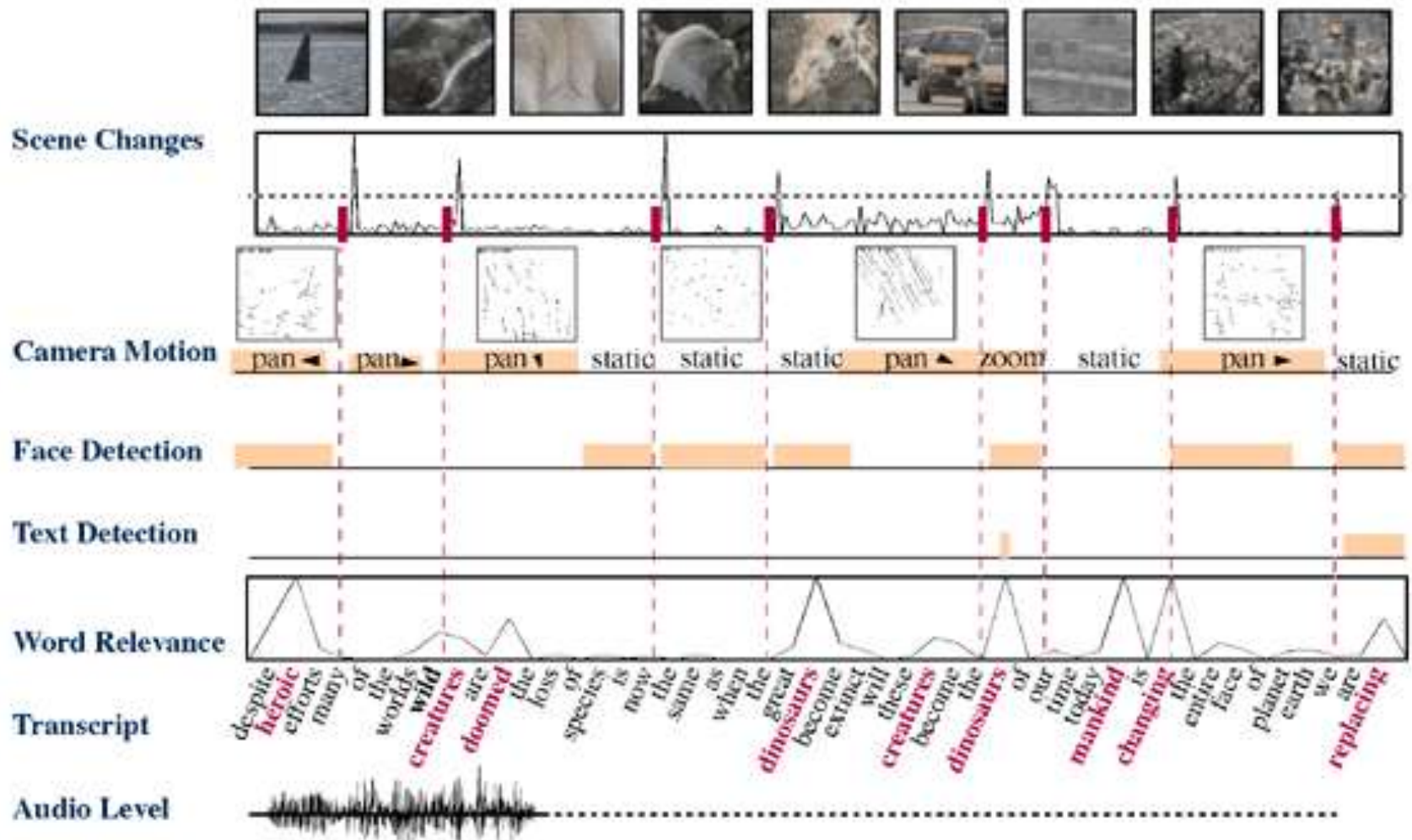




# Key Frame Extraction and Video Retrieval



# Various Clues/Contents in Video Retrieval



# Video Caption Extraction in Video Retrieval

Source Video:



Time-Based Minimum Image:



Final VOCR Results:

**FREEMAN  
BLOCK  
LOS  
ANGELES  
COUNT  
SHERIFF**

Text  
Region

SHERMAN BLOCK

Filtered  
Text

SHERMAN BLOCK

Binarized  
Segmented

SHERMAN BLOCK

OCR:

S H E R M A N B L O C K

Text  
Region

LOS ANGELES COUNTY SHERIFF

Filtered  
Text

LOS ANGELES COUNTY SHERIFF

Binarized  
Segmented

LOS ANGELES COUNT SHERIFF

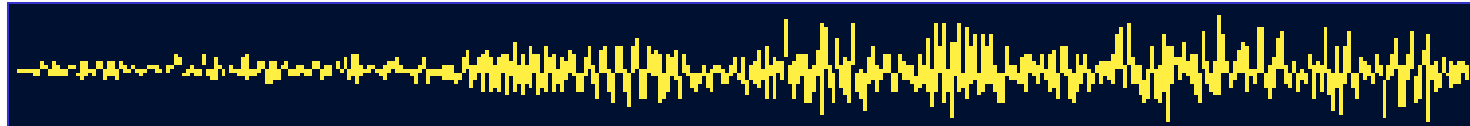
OCR:

L O S A N G E L E S C O U N A S H E R I F F

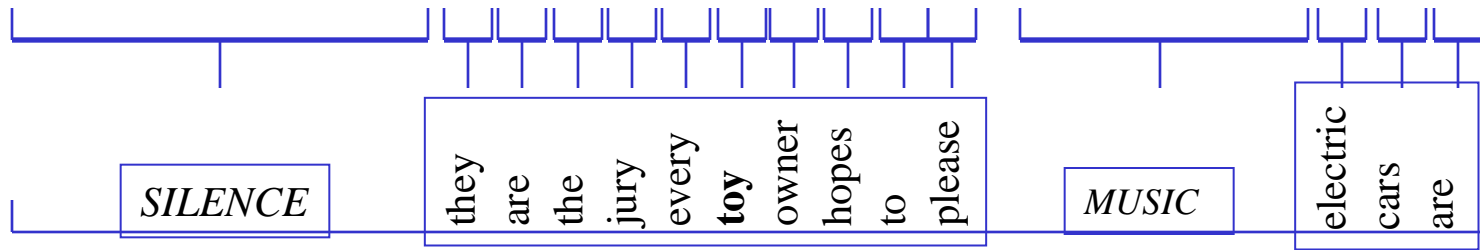
# Transcript via Speech Recognition for Video Retrieval

- Generates transcript to enable text-based retrieval from spoken language documents
- Improves text synchronization to audio/video in presence of scripts

Raw Audio



Text  
Extraction

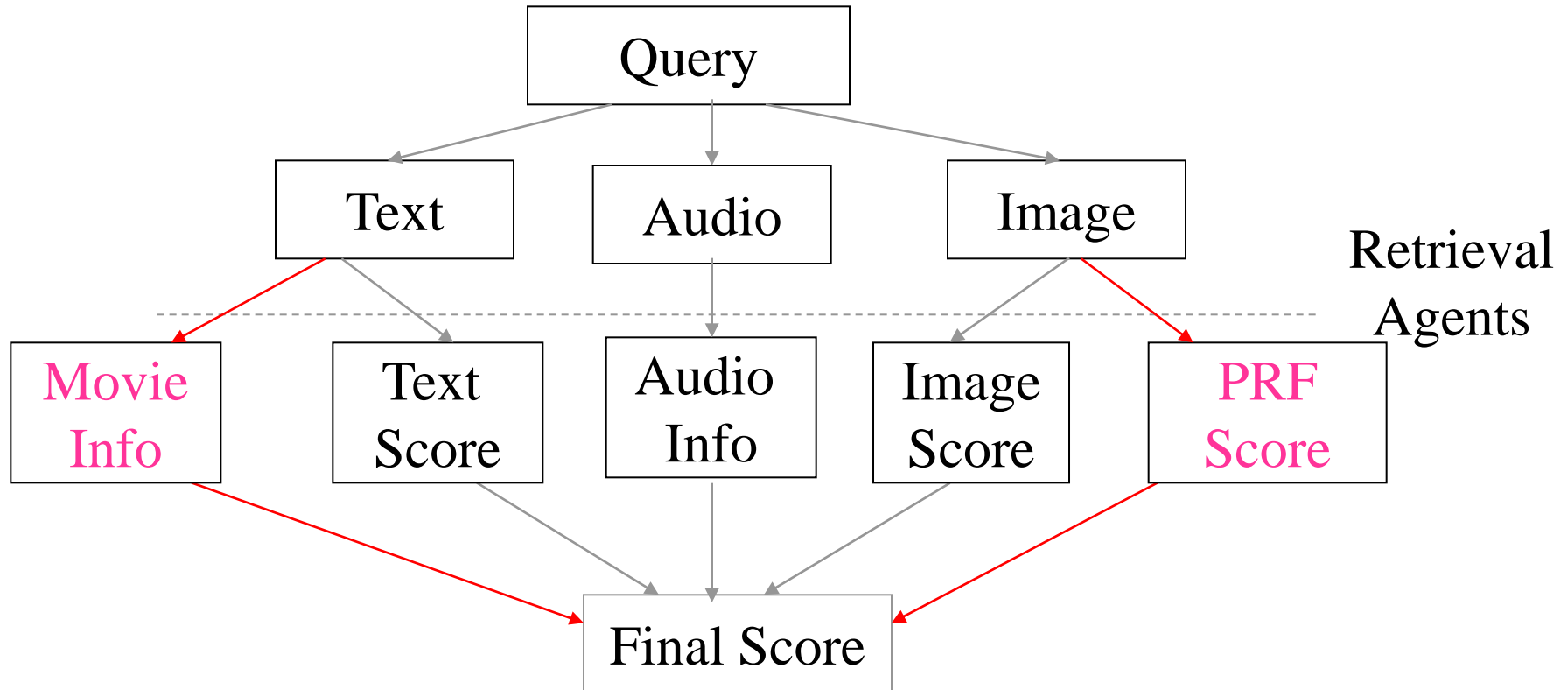


Raw Video

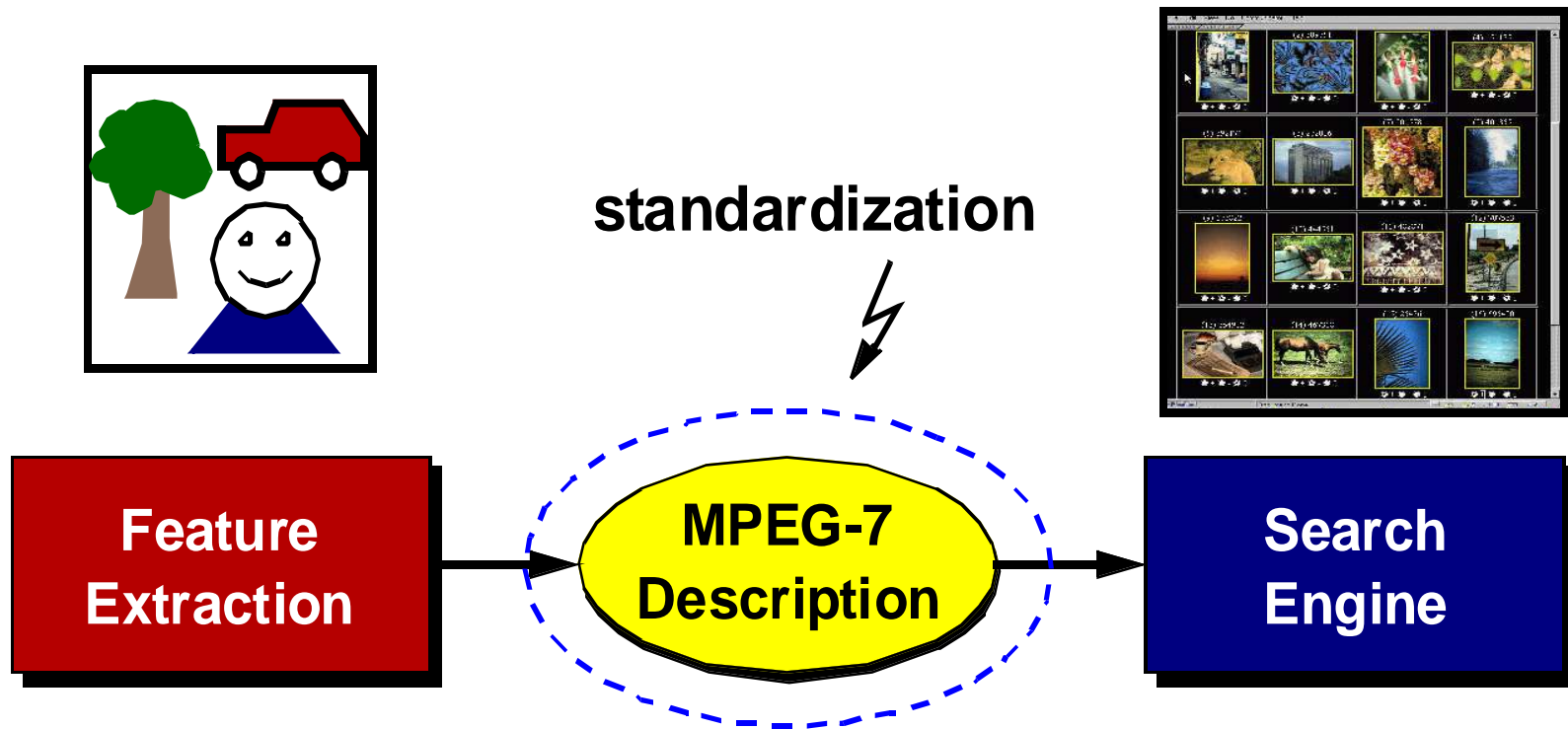




# Video Retrieval by Combining Different Features



# MPEG-7: Audiovisual Content Description



## Feature Extraction:

- Content analysis (D, DS)
- Feature extraction (D, DS)
- Annotation tools (DS)
- Authoring (DS)

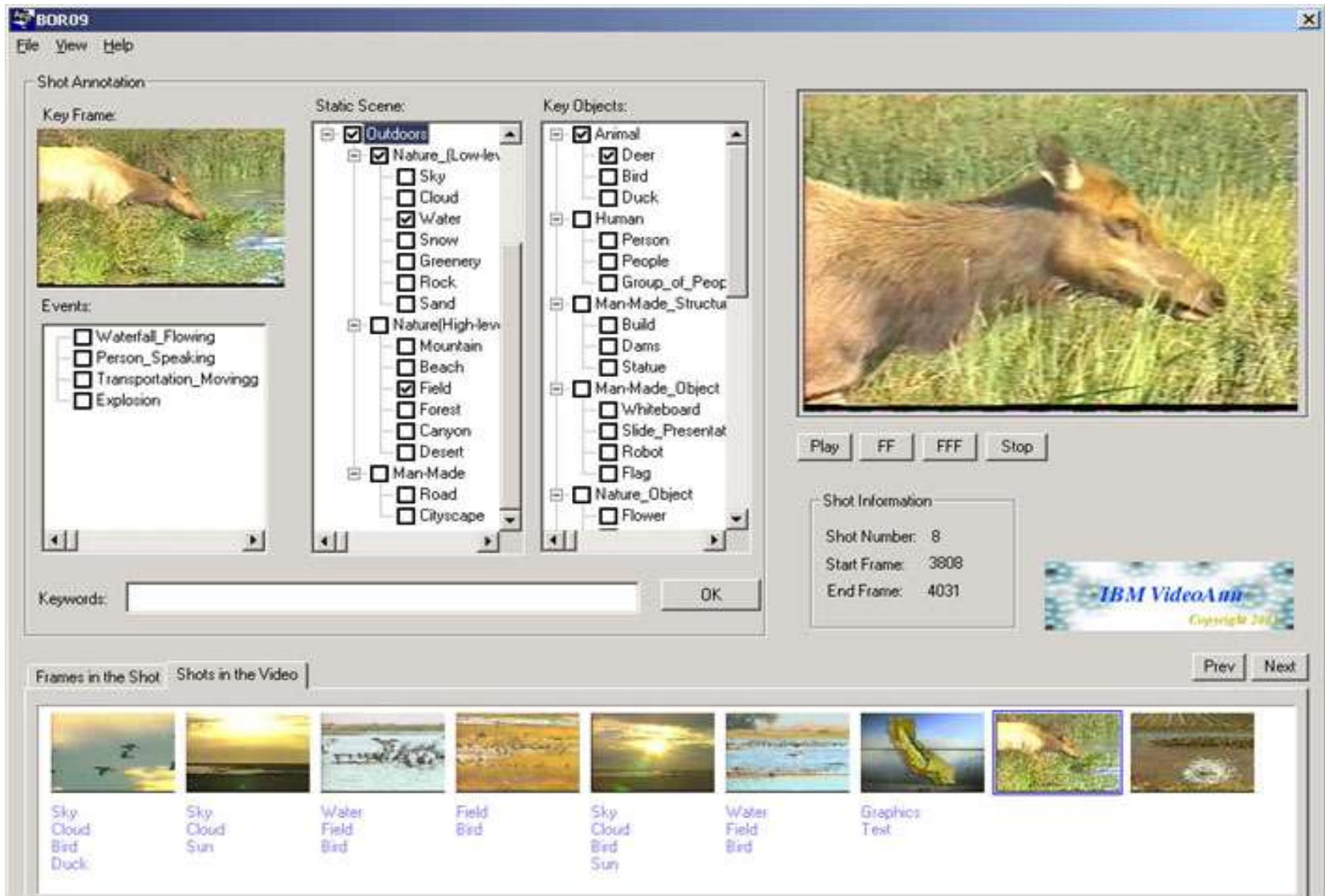
## MPEG-7 Scope:

- Description Schemes (DSs)
- Descriptors (Ds)
- Language (DDL)
- Ref: MPEG-7 Concepts

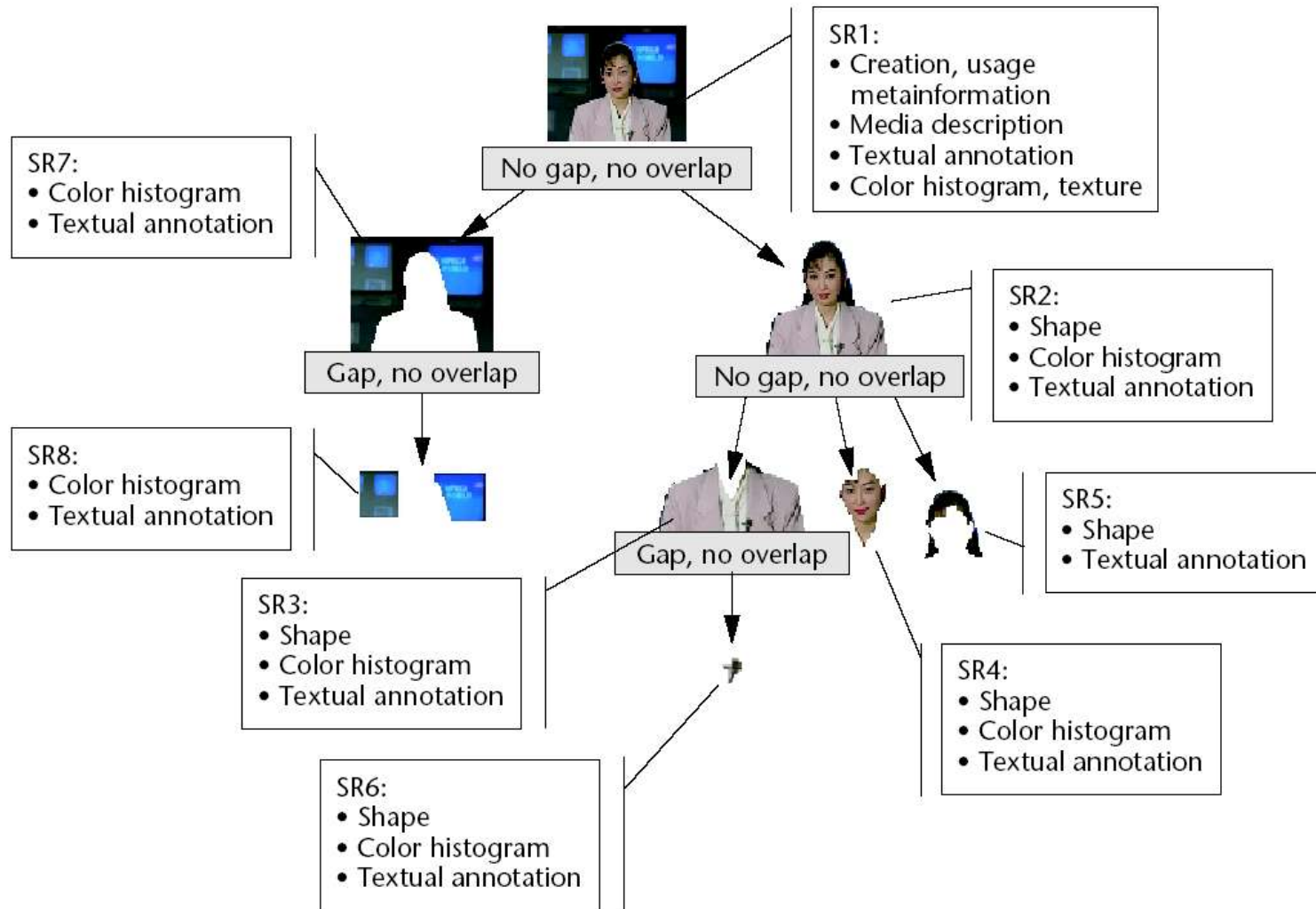
## Search Engine:

- Searching & filtering
- Classification
- Manipulation
- Summarization Indexing

# Example of MPEG-7 Annotation Tool



# MPEG-7: Image Description Example





# Automatic Video Analysis and Index

Scene Cuts



Camera

Static

Static

Zoom

Objects

Adult Female

Animal

Two adults

Action

Head Motion

Left Motion

None

Captions

[None]

Yellowstone

[None]

Scenery

Indoor

Outdoor

Indoor

Time  
Axis

### Segment Tree

Shot1 Shot2 Shot3

Segment 1

Sub-segment 1

Sub-segment 2

Sub-segment 3

Sub-segment 4

segment 2

Segment 3

Segment 4

Segment 5

Segment 6

Segment 7

### Semantic DS (Events)

• Introduction

• Summary

• Program logo

• Studio

• Overview

• News Presenter

• News Items

• International

• Clinton Case

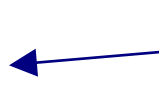
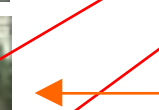
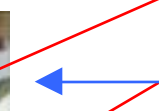
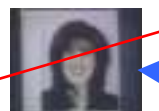
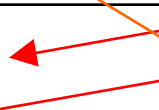
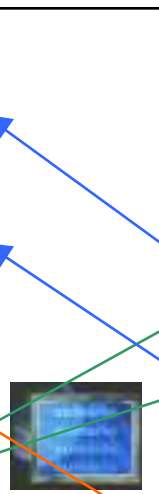
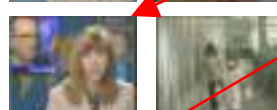
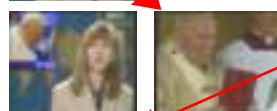
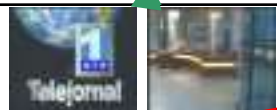
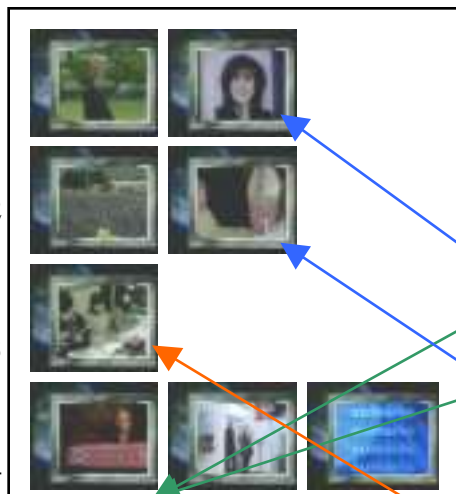
• Pope in Cuba

• National

• Twins

• Sports

• Closing



# Audio Retrieval

## ❑ Audio retrieval:

- Find required sound segment from audio database or broadcasting
- Find interesting music from song/music database or web

## ❑ Methods of audio retrieval

### Physical features of audio signal:

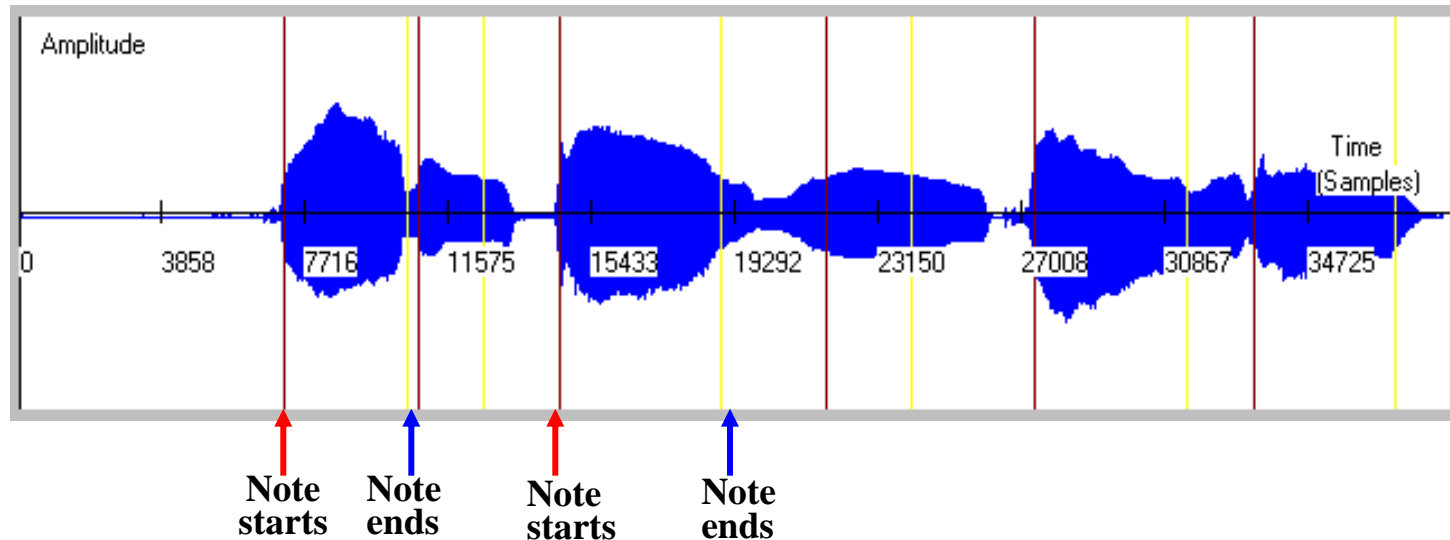
- Loudness, i.e., sound intensity (0~120dB)
- Frequency range: low, middle or high (20Hz~20KHz)
- Change of acoustic feature
- Speech, background sound, and noise
- Pitch

### Semantic features of audio:

- word or sentence via speech recognition
- Male/female, young/old
- Rhythm and melody
- Audio description/index
- Content Based Music Retrieval (CBMR)

# Music Retrieval by Singing/humming

## Happy Birthday

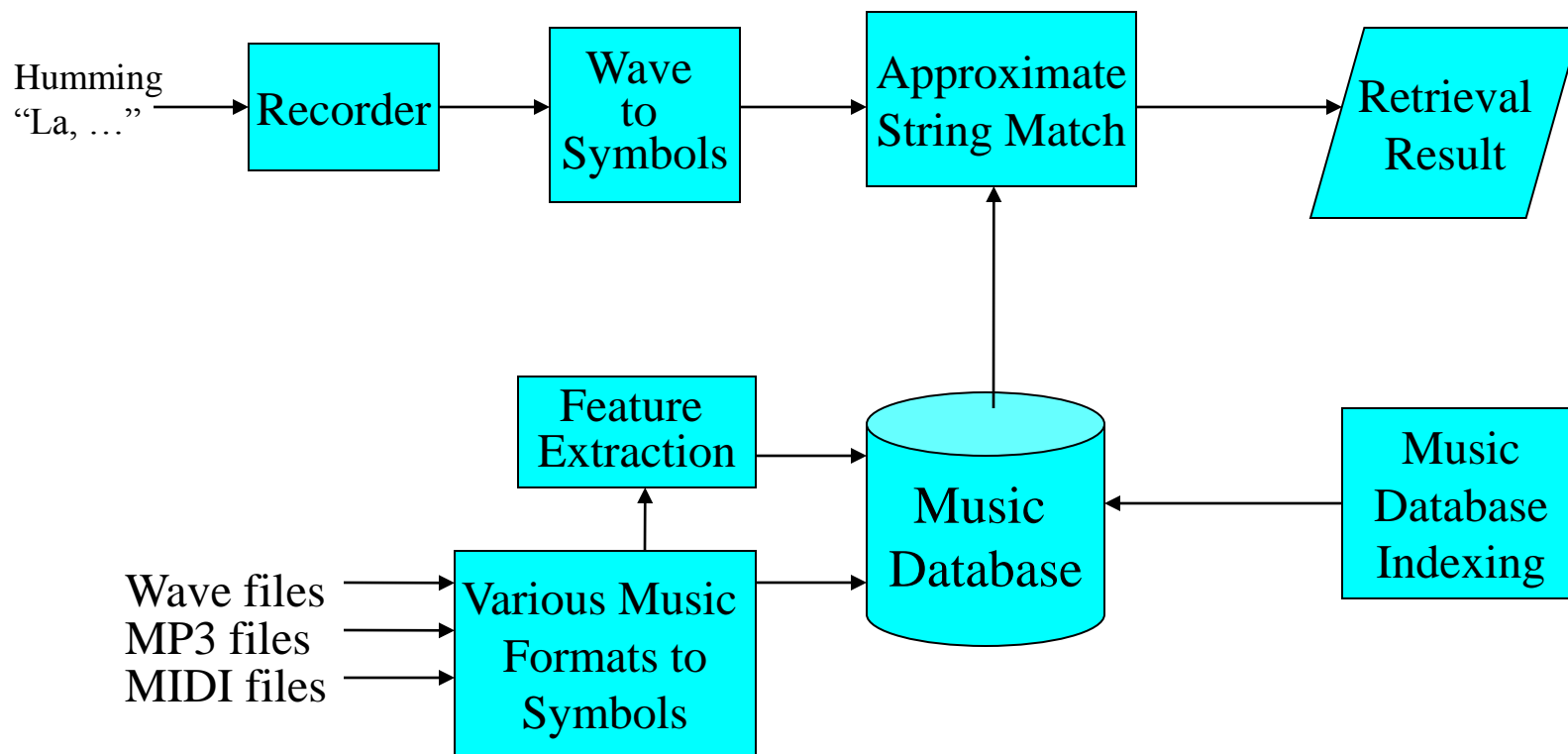


- A note has two important attributes
  - Pitch: It tells people which tone to play
  - Duration: It tells people how long a note needs to be played
  - Notes are represented by symbols

Staff {

Note name	C	D	E	F	G	A	B	C
Note pitch	Do	Re	Mi	Fa	So	La	Si	Do

# Music Retrieval by Singing/humming (Cont.)



# Demos of Content-Based Image Retrieval

# Mobile Multimedia Service Over Wireless Networks

- Mobility and Universal Services
- Wireless LAN (Local Area Network)
- Wireless WAN (Wide Area Network)
- 3G Wireless Networks and IMT-2000
- FOMA and DoCoMo Mobile Services
- WAP (Wireless Application Protocol)
- Techniques and Challenges in Mobile Multimedia

# What is Mobility?

- ❑ Terminal mobility: A terminal that moves
  - Between different geographical locations
  - Between different networks
  - Laptop, PDA, cellular phone, etc
- ❑ User mobility: A person who moves
  - Between different geographical locations
  - Between different networks
  - Between different communication devices
  - Between different applications
- ❑ Service mobility
  - A communication & information system can serve mobile device/user
  - Mobile service vs fixed service
  - Fixed networks, i.e., wired Internet, provides such service for PC/WS
  - Mobile networks, i.e., wireless Internet, supports mobile device/user



# Universal Service

Universal Service = Fixed Service + Mobile Service

-- Enable **anybody** to communicate with **anyone** and get required information from **any terminal** at **anywhere** in **anytime**

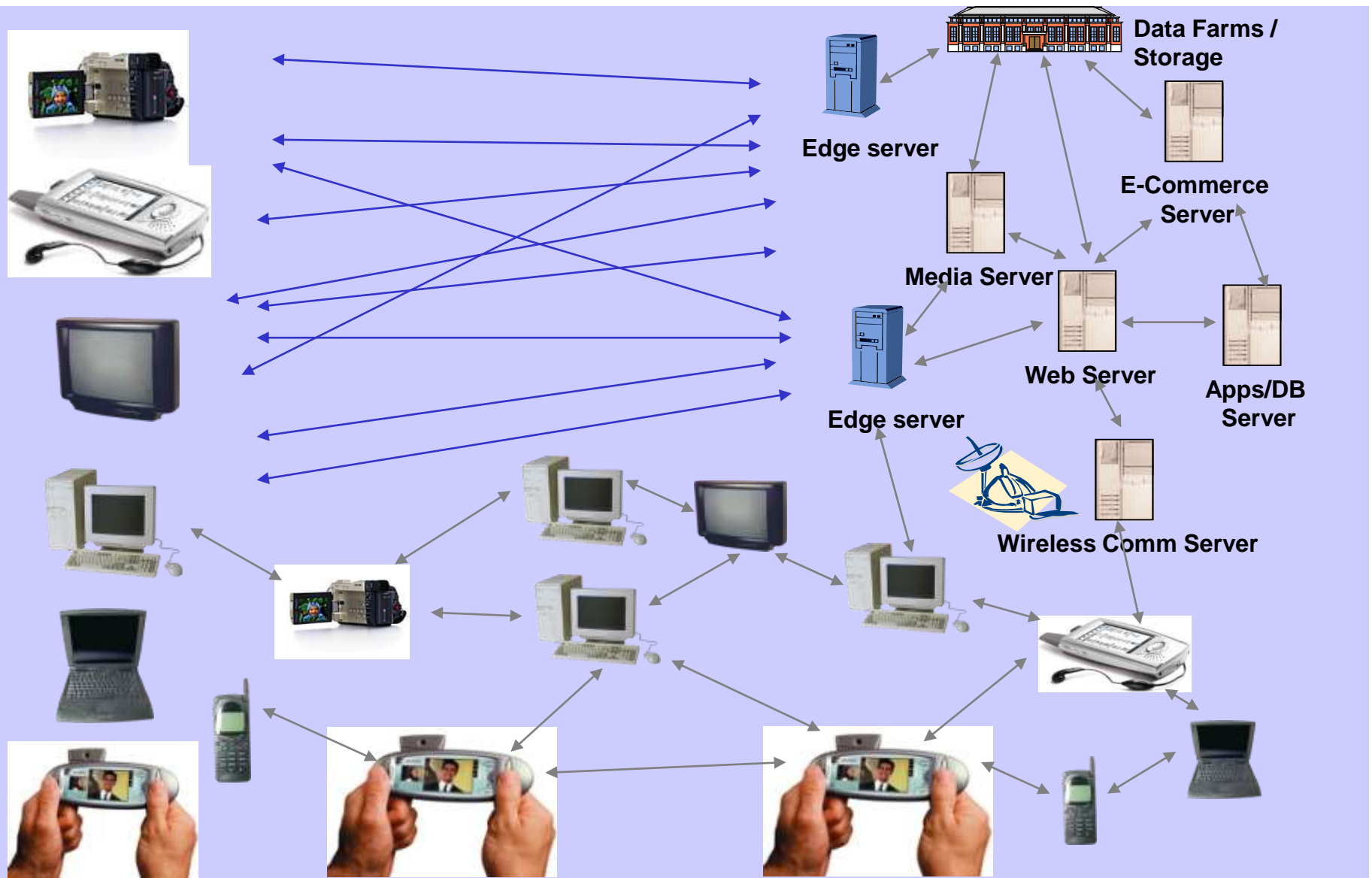


Wired & Wireless  
Internet



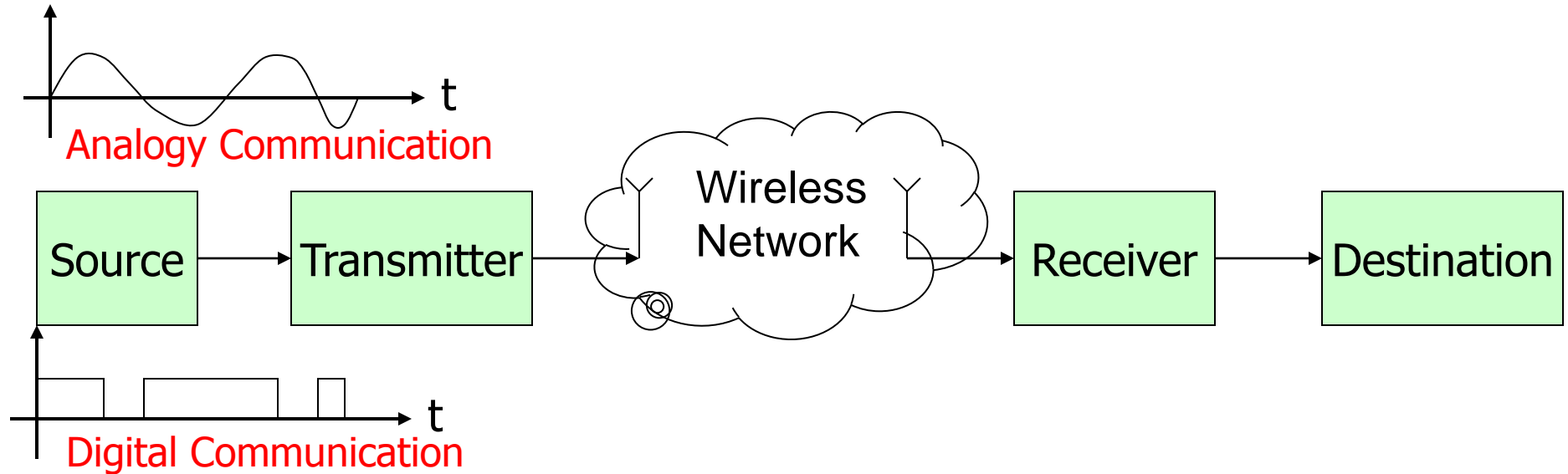
Anybody  
Anyone  
Any terminal  
Anywhere  
Anytime

# Multimedia: from Desktop, to Internet, to Hand-helds, and to Wireless Terminals



# Wireless Communications

## General Wireless Communication Model



Frequency (Hz)

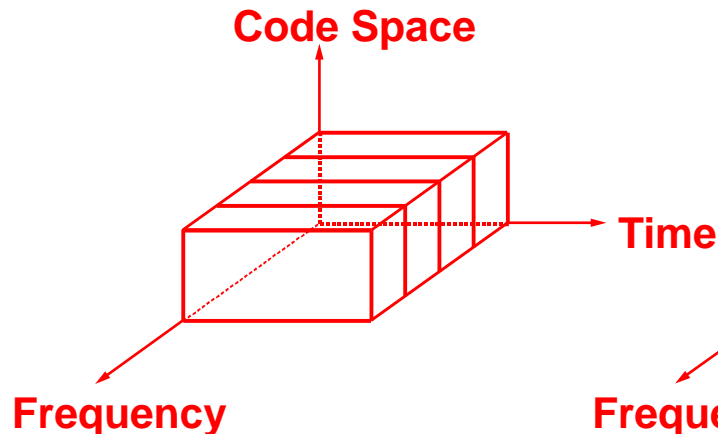
1GHz 10GHz

$10^2$	$10^3$	$10^4$	$10^5$	$10^6$	$10^7$	$10^8$	$10^9$	$10^{10}$	$10^{11}$	$10^{12}$	$10^{13}$	$10^{14}$	$10^{15}$
ELF	VF	VLF	LF	MF	HF	VHF	UHF	SHF	EHF				
Power Telephone Music Microphone			<b>Radio</b> Radio Broadcast Television HF Communication Sea Communication					<b>Microwave</b> Terrestrial Relay Satellite Comm. <b>Mobile Comm.</b> <b>Wireless LAN</b>		<b>Infrared</b> Laser Comm. Missile Comm.		<b>Visible light</b>  Optical Comm.	

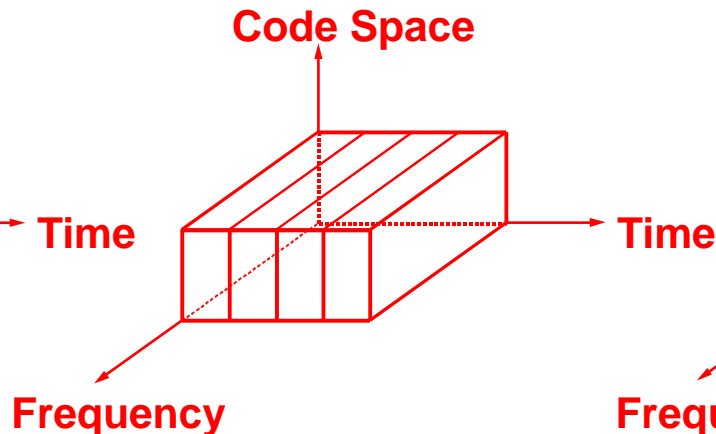
# Multiple Access Control (MAC)

- ❑ Multiple access: to effectively utilize limited frequency resources by enabling multiple users to share radio communications channels to simultaneously conduct communications. Three types of systems
  - FDMA - Frequency Division Multiple Access
  - TDMA - Time Division Multiple Access
  - CDMA - Code Division Multiple Access

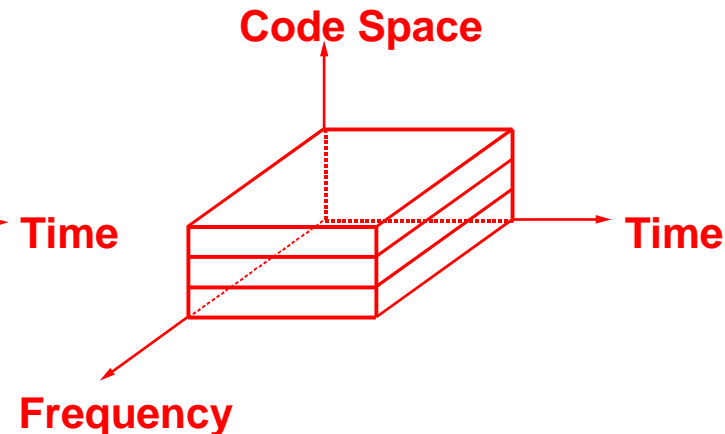
FDMA



TDMA



CDMA

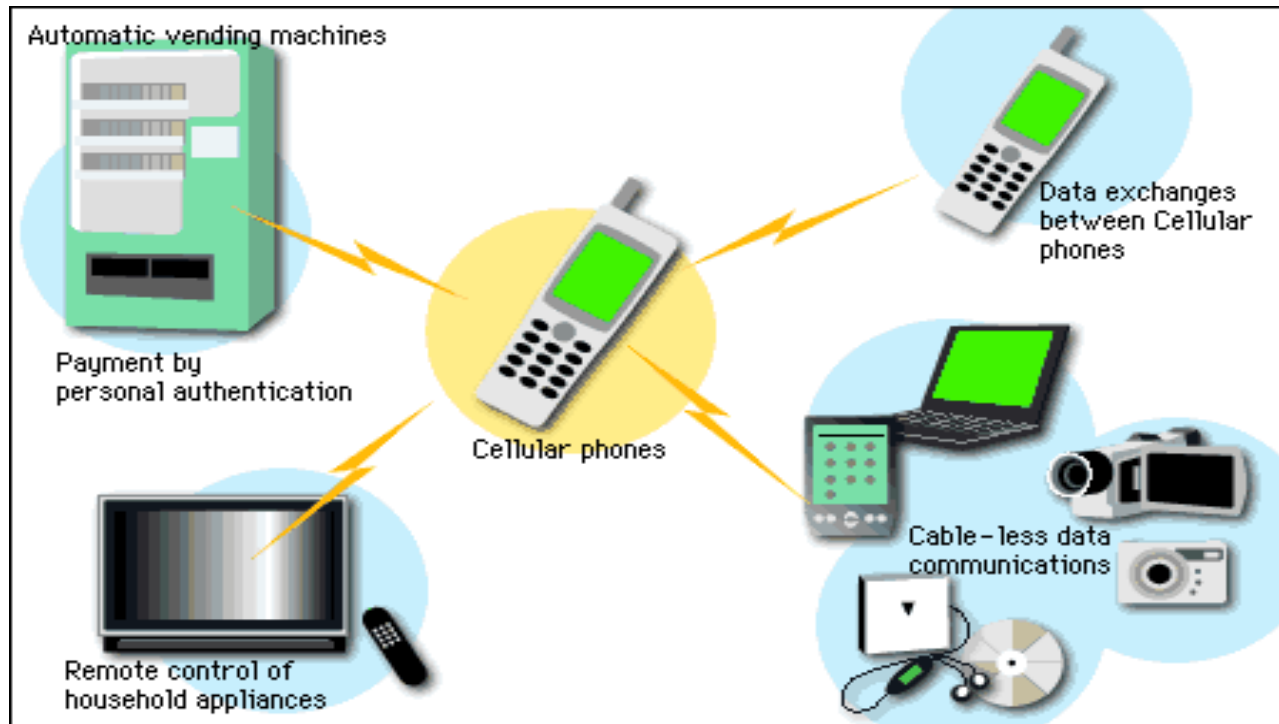
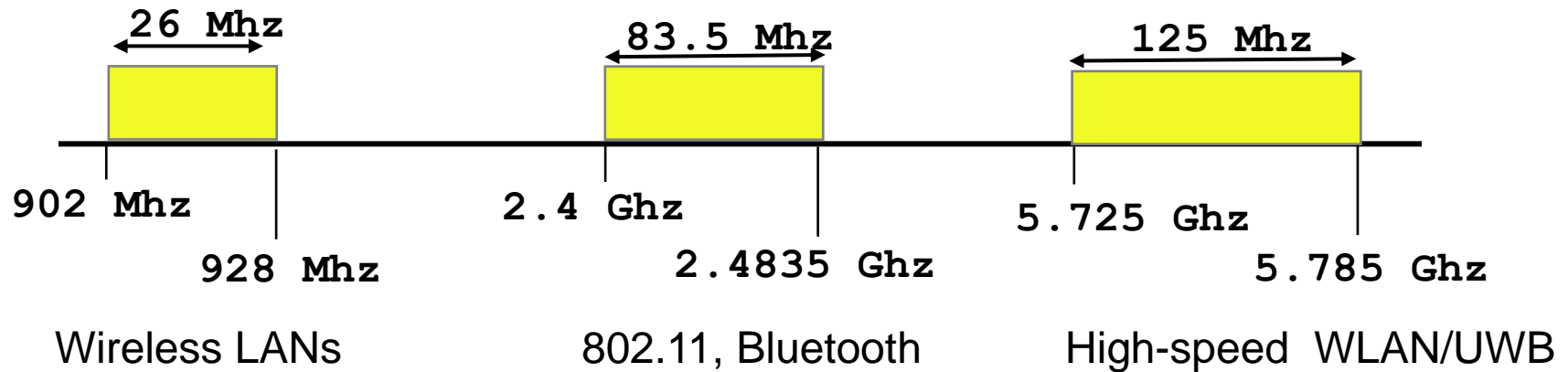


# Wireless LAN



- ❑ Wireless LAN: small range (< 100m)
- ❑ IEEE 802.11 (similar to Ethernet)
  - Defined by IEEE (Institute for Electrical and Electronic Engineers)
  - Access control: CSMA/CD (only one can send each time similar to TDMA, listen and transmit if no other transmission, otherwise wait)
  - Speed: 2Mbps (infrared), >10Mbps (Microwave, 2.4/5.2GHz)
- ❑ HIPERLAN
  - Defined by ETSI (European Telecommunication Standard Institute)
  - Access control: dynamic TDMA
  - Speed: 25Mbps (5GHz) and 155Mbps (17GHz)
- ❑ HomeRF
  - Defined by Home Radio Frequency Working Group (Industry, 1998)
  - Access control: similar IEEE 802.11 with priority and reservation control
  - Speed: 10Mbps (2.4GHz), support both data, voice and streaming
- ❑ Bluetooth
  - Defined by Bluetooth Special Interest Group (SIG, industry)
  - Access control: TDD (Time Division Duplex) with circuit and packet switch
  - Speed: >1Mbps

# WLAN Frequency & Bluetooth Applications



# Wireless Access System Frequency in JAPAN

Frequency band	2.4GHz	5GHz		22/26/38 GHz	25/27 GHz	60GHz
		outdoor	indoor			
usage	Wireless LAN Wireless access	Wireless access	Wireless LAN	FWA	Wireless LAN Wireless access	Wireless LAN Wireless access
Bandwidth (MHz)	100	160	100	2880	940	7000
Radio Station License	free	required (base station)	free	Required	free	LAN:free
Transmission speed (Mbps)	10 -> 20	5-50	20-50	156	100	Some of 100s
Note	Ordinance for enhancement not settled	Ordinance will be settled this summer	Products emerges since last autumn	MPHPT issues license to 15 operators	Ordinance Settled Expects Products this autumn	Products emerges since last year



# Ultra-Wideband (UWB)

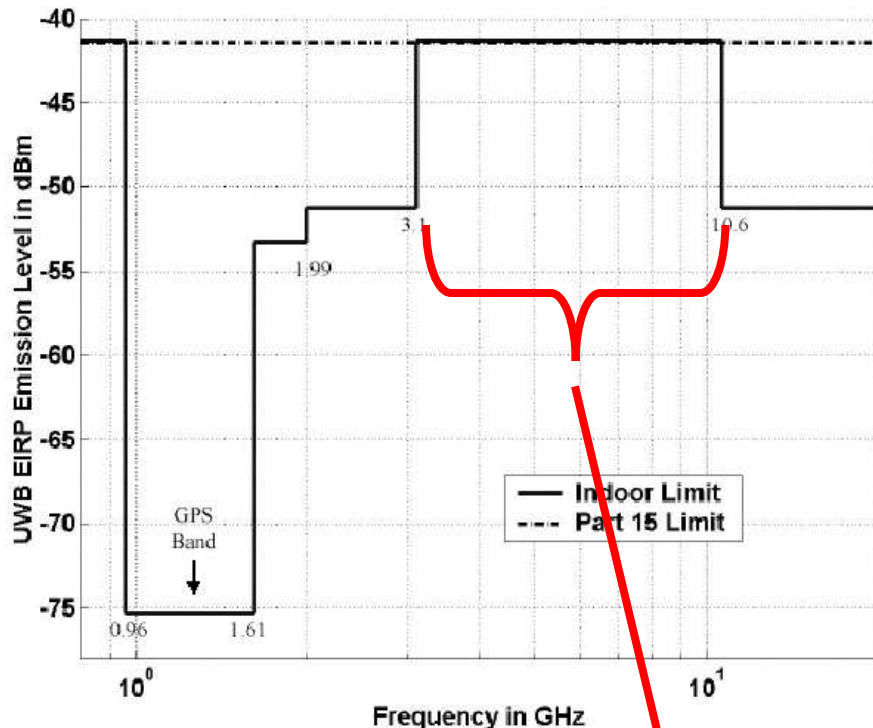


Figure 1. UWB spectral mask for indoor communication systems.

Unlicensed bands	Frequency of operation	Bandwidth
ISM at 2.4GHz	2.4000-2.4835	83.5MHz
U-NII at 5GHz	5.15-5.35GHz 5.75-5.85GHz	300MHz
UWB	3.1-10.6GHz	7,500MHz

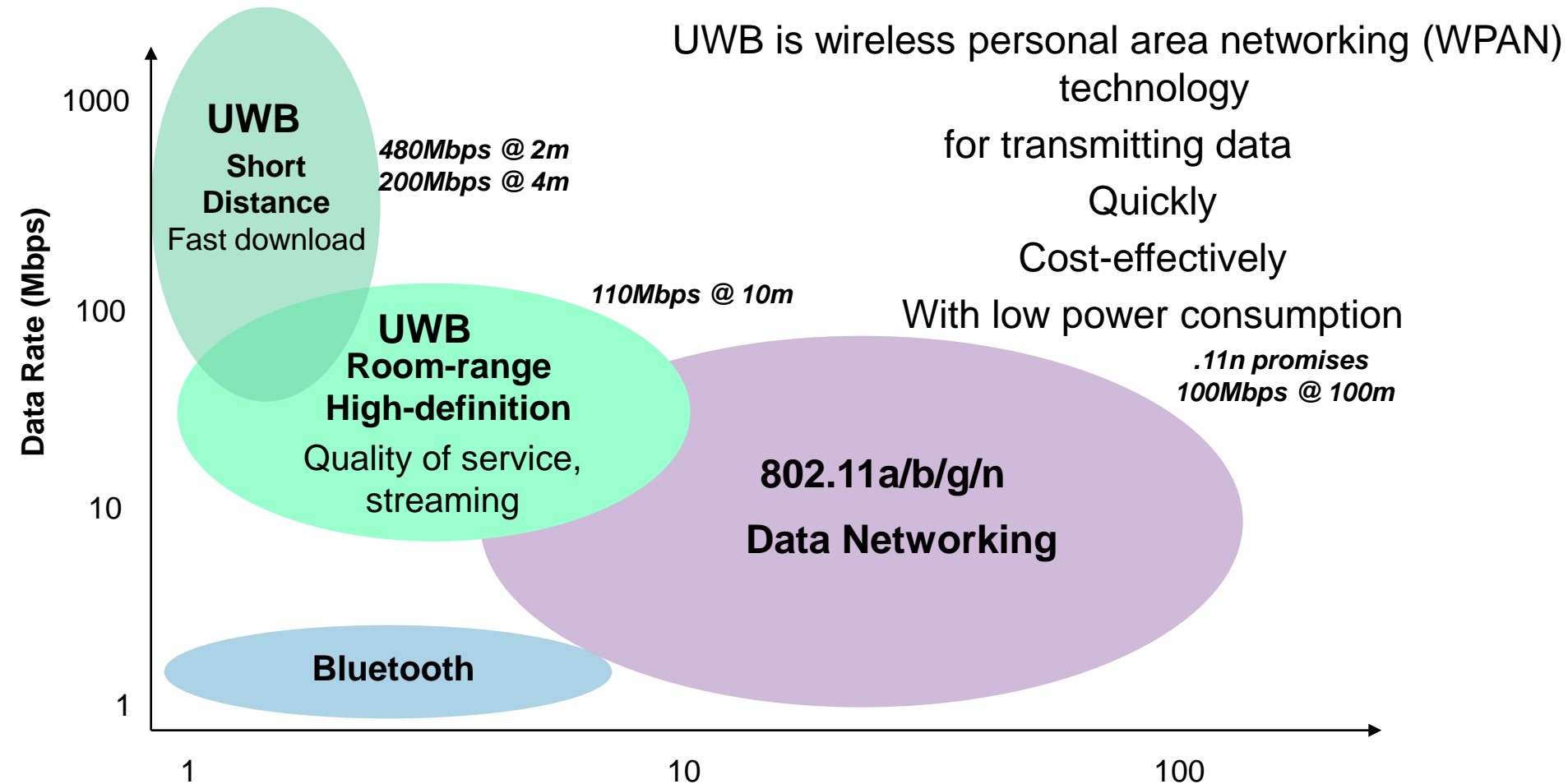
Table 1. US spectrum allocation for unlicensed use.

## Regulated in the US since February 2002

- UWB is available spectrum, not a specific technology
- 7,500MHz of unlicensed spectrum
- First regulation ever that allows spectrum sharing: low emission limit (-41.3dBm/MHz EIRP) doesn't cause harmful interference
- Transmitters need to occupy at least 500MHz all the time
- UWB devices are NOT defined as impulse radios or by any specific modulation
- Enough spectrum to reach much higher data rates than in the ISM band (83.5MHz at 2.4GHz) or the U-NII bands (300MHz at 5GHz)
- Optimized for short-distances applications



# UWB Communication/Network



# UWB Application Vision



**Personal  
Wireless Storage/Wallet**



**Share video clips  
Music & Photos**



**SHARE and EXCHANGE  
Create New User Models  
not possible in the  
Cabled World  
Connecting PC, CE and  
Mobile Segments**



**Photo Printer**



**Media Center**



**Multi Channel  
Speakers**



**In Car Media center  
& video**

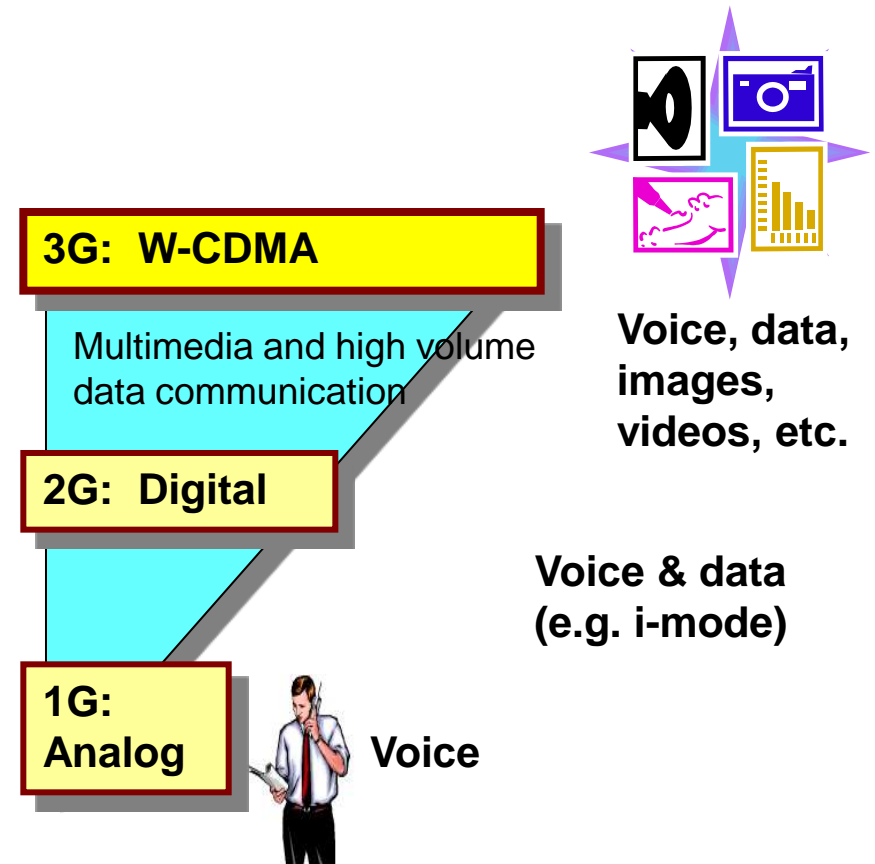


**Photo & Video Clip  
Display**

# Wireless WAN

**W-WAN** (Wireless Wide Area Network): city, country, continent, the globe

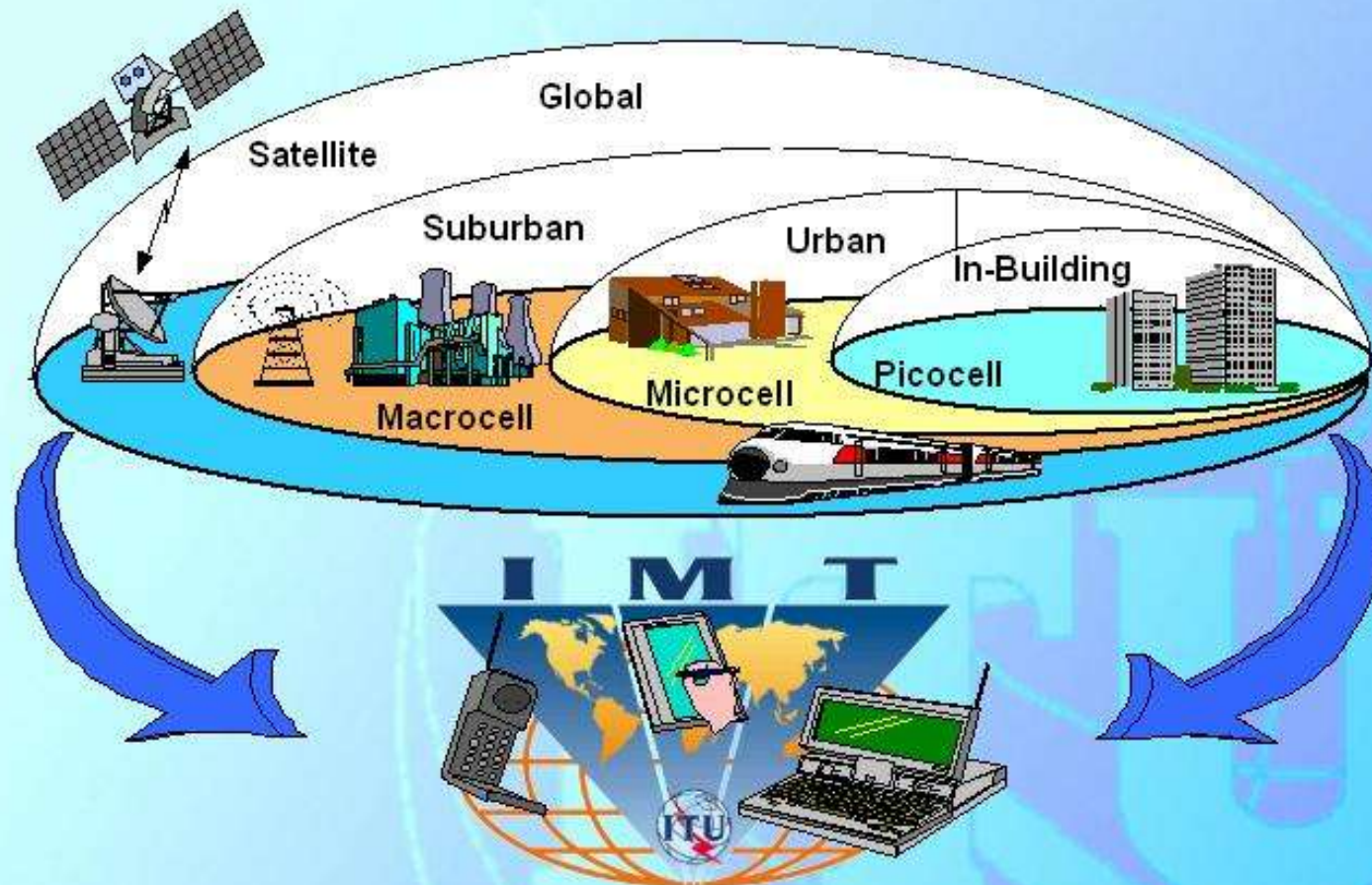
- ❑ **1G** (1<sup>st</sup> generation) wireless networks (1980's)
  - Analog and FDMA, Data rate: < 2.4Kbps
- ❑ **2G** wireless networks (1990's)
  - Digital and TDMA/CDMA
  - **2G**: GSM, PDC, IS-136, IS-96/CDMAOne, Data rate: 10Kbps
  - **2.5G**: GPRS, EDGE, IS-95B, Data rate: 64Kbps
- ❑ **3G** wireless networks (2000's)
  - Digital and CDMA: WCDMA, UWC-136, CDMA2000
  - Data rate: 144Kbps (Vehicular), 384Kbps (Pedestrian), 2Mbps (Indoor)
- ❑ **4G** wireless networks
  - Research/service is under going



# International Mobile Telecommunication

## IMT-2000

THE emerging network of the 21<sup>st</sup> century



# Systems beyond IMT-2000

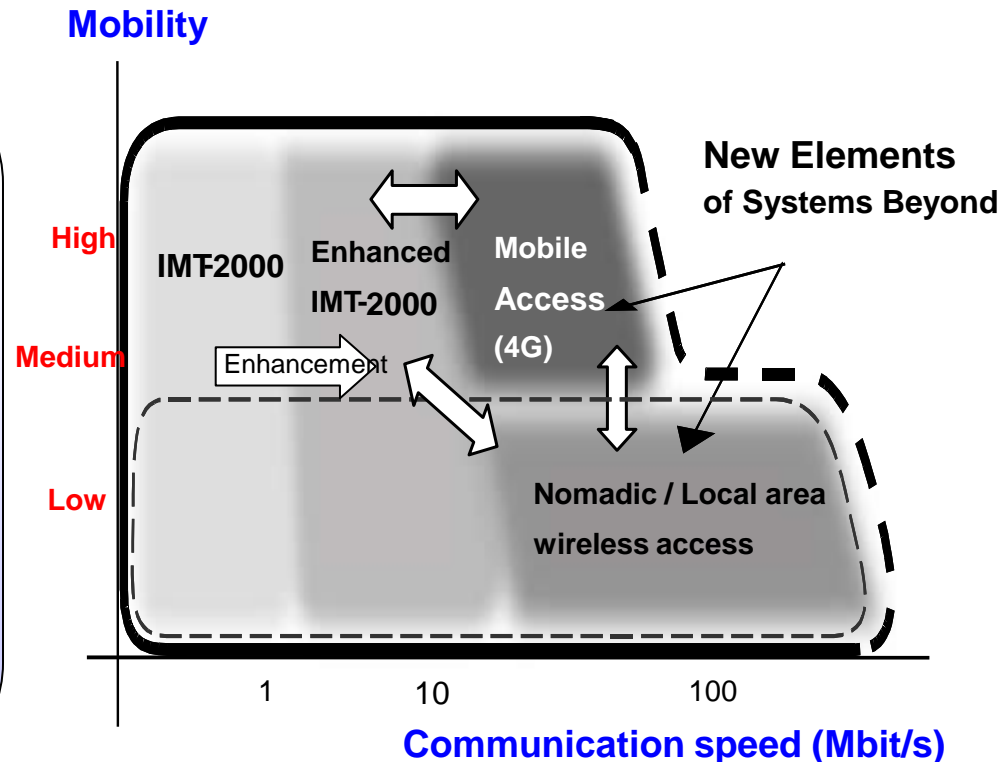
## Systems beyond IMT-2000

A long-term plan required for R&D and allocating frequency

## Key Elements

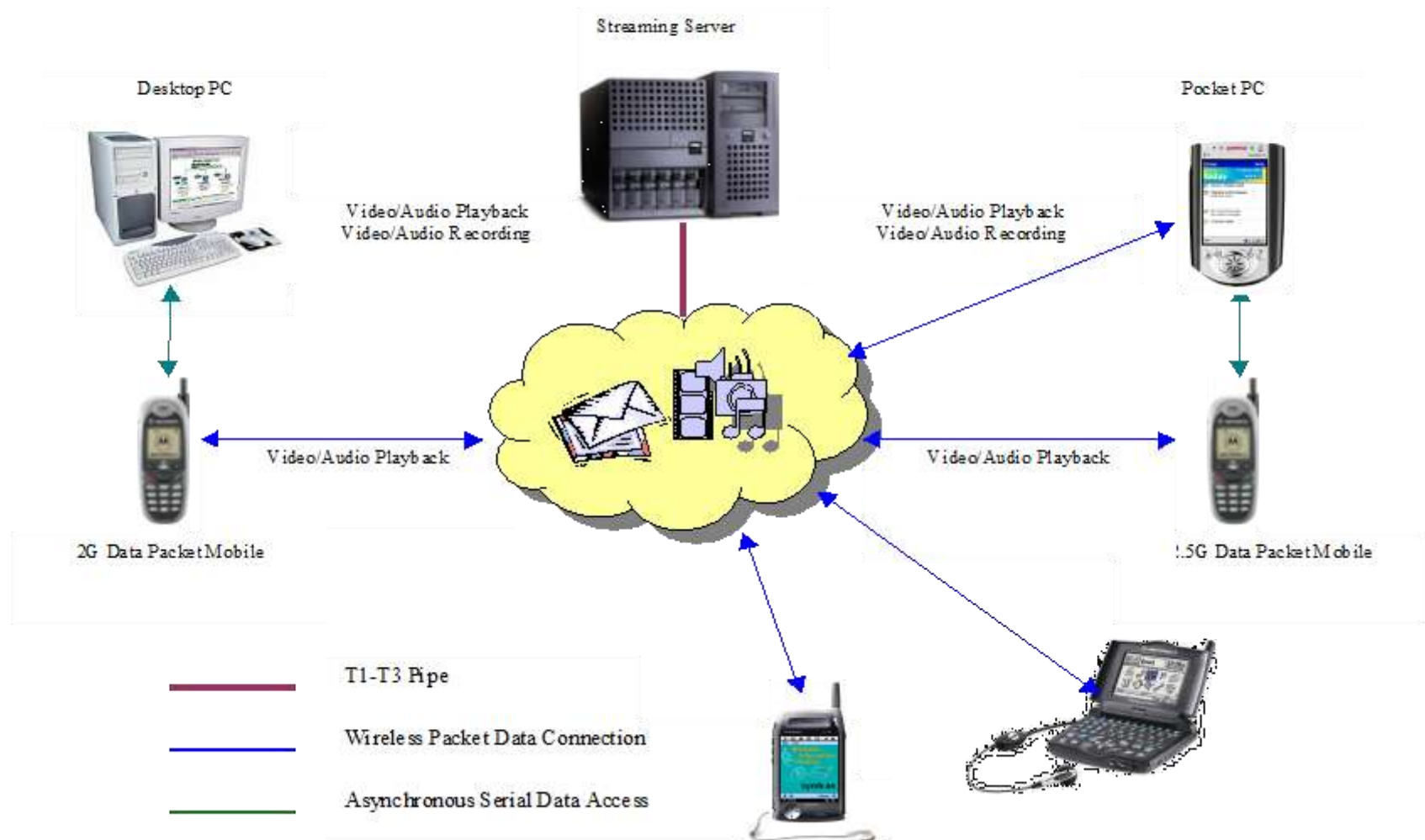
1. Very high-speed communication (50-100Mbps) equivalent to OPT fiber
2. All IP network (IPV6)
3. Integration of cellular type and wireless LAN type system.
4. Use of Software Defined Radio technology

(Note: 3G network will not be replaced by new elements, rather co-exists with them)

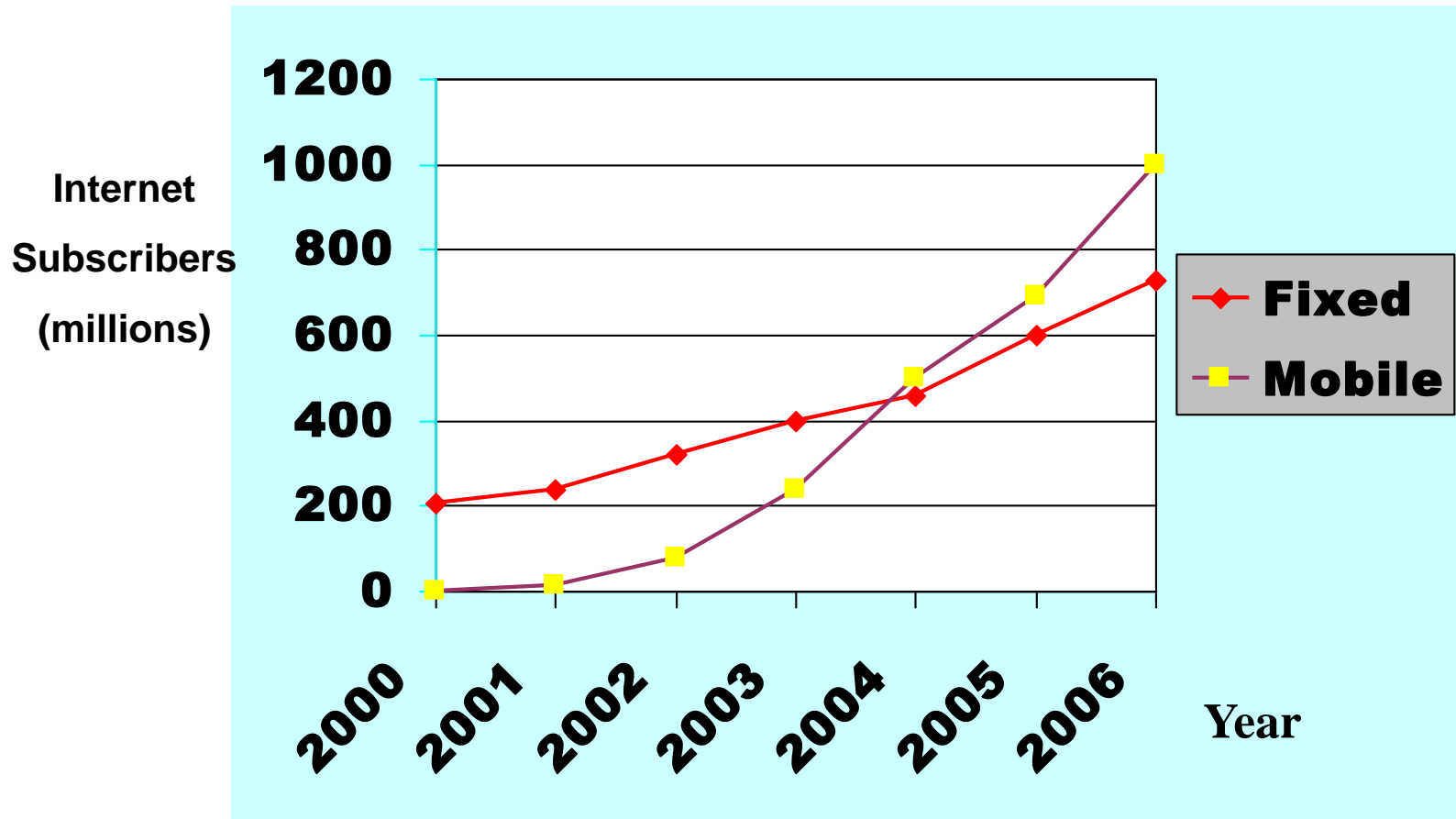




# Mobile Multimedia Access

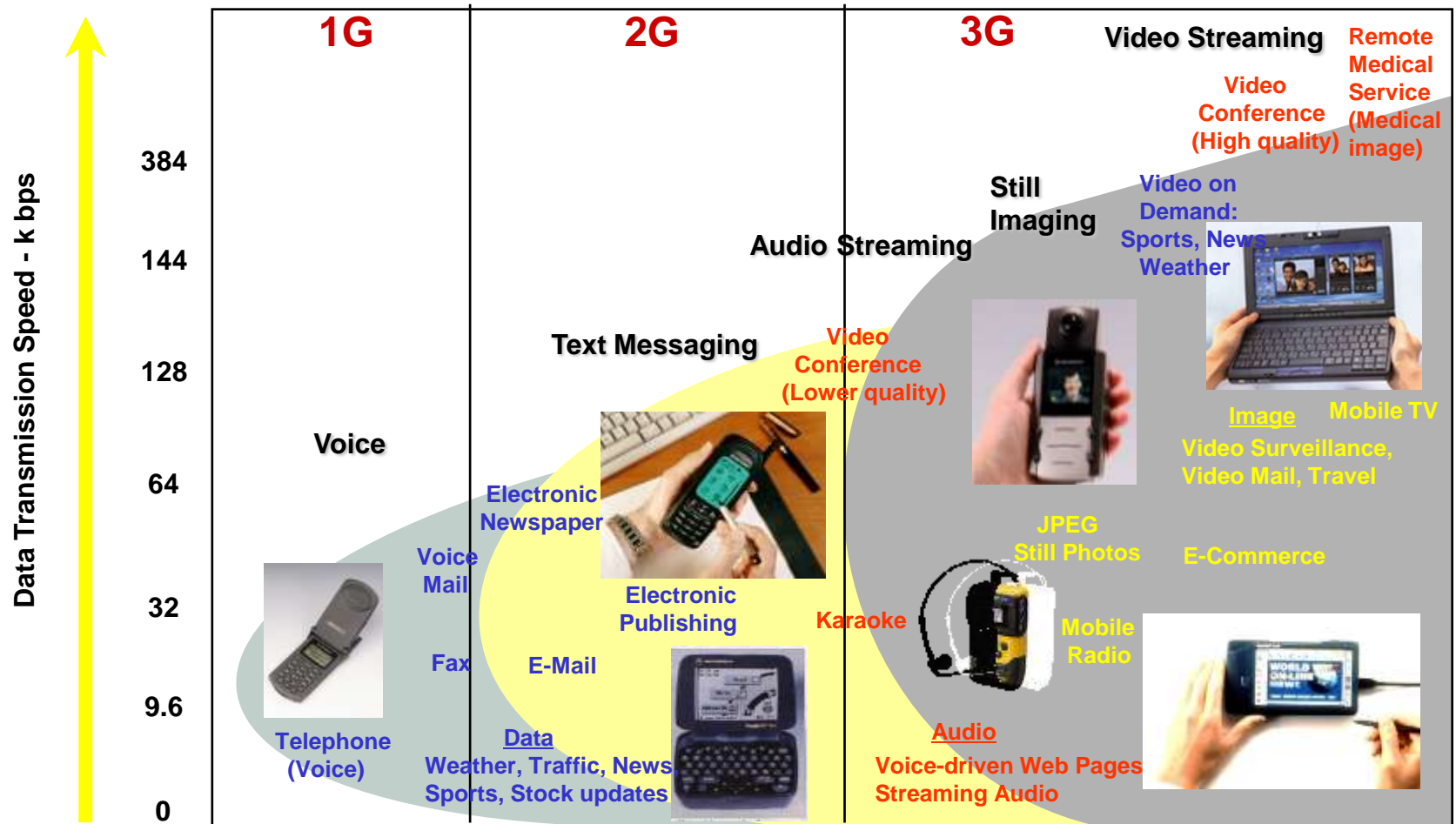


# Mobile Internet Access



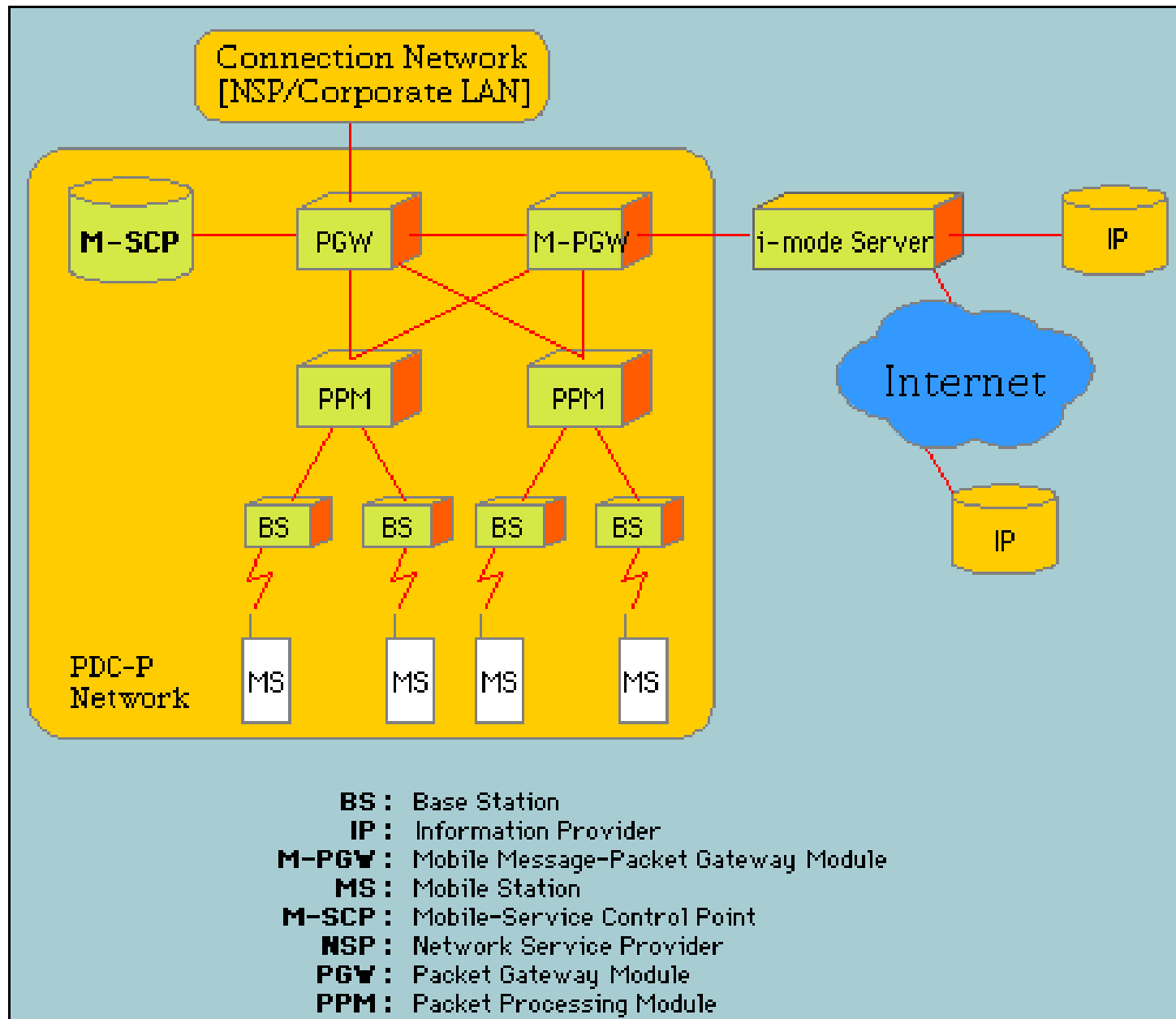
Source: Ericsson

# Mobile Multimedia Data Service



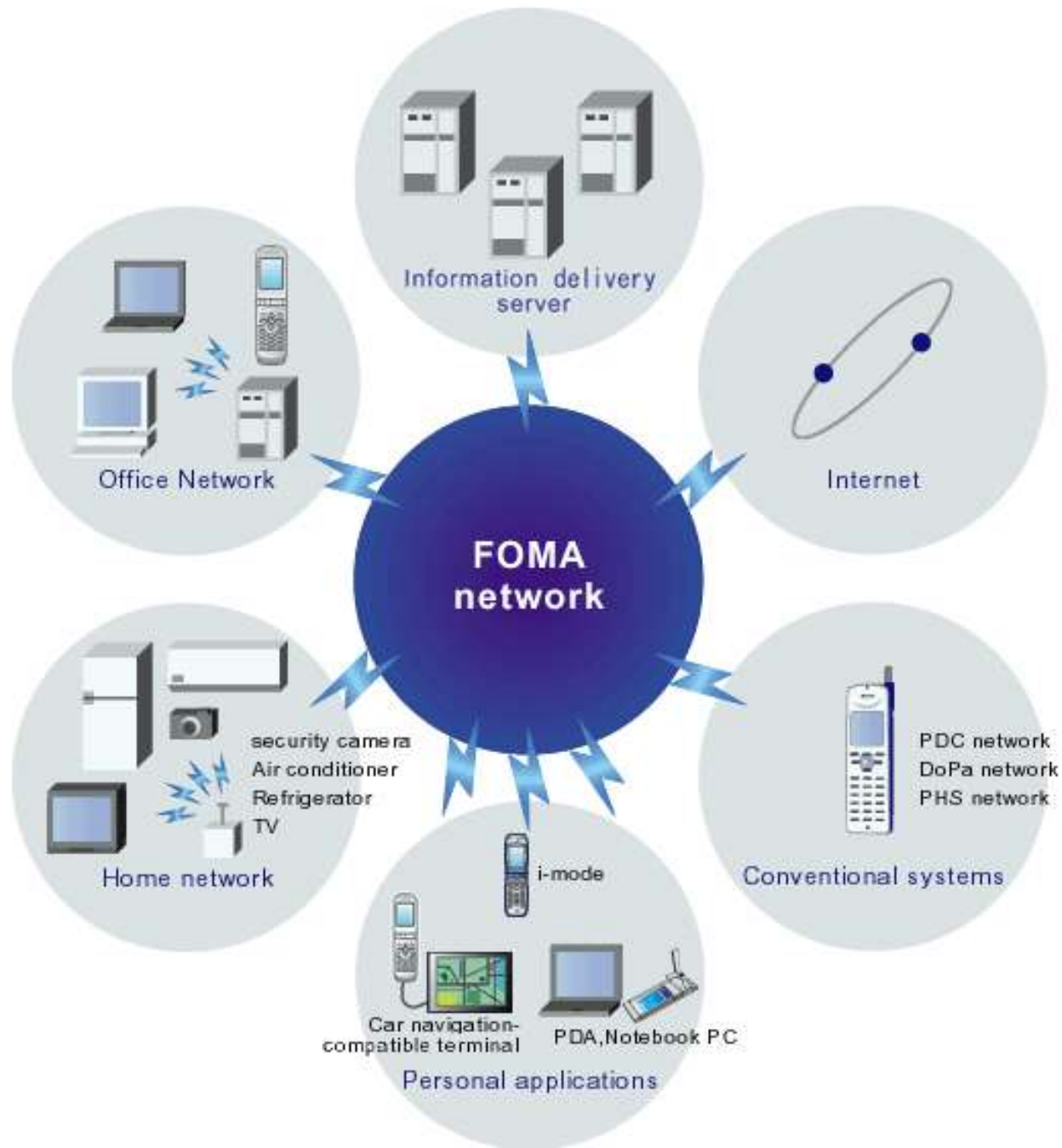


# PDC-P Network and i-mod Server



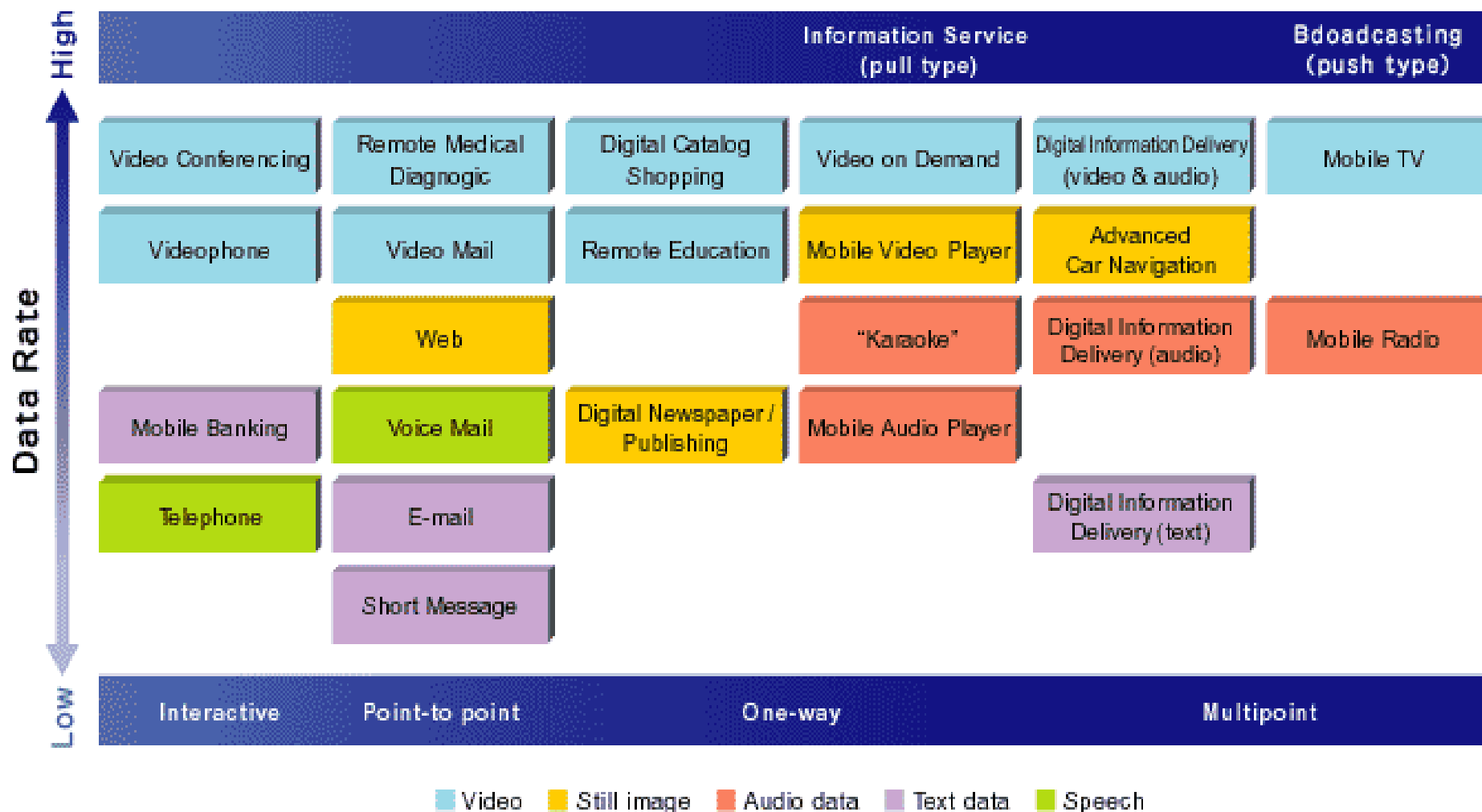
# FOMA

- ❑ Freedom of Mobile Multimedia Access
- ❑ Used in Japan for NTT DoCoMo's 3G service
- ❑ W-CDMA
- ❑ Service starts from October 1, 2001



# Current and Future FOMA Services

## Service and applications to come



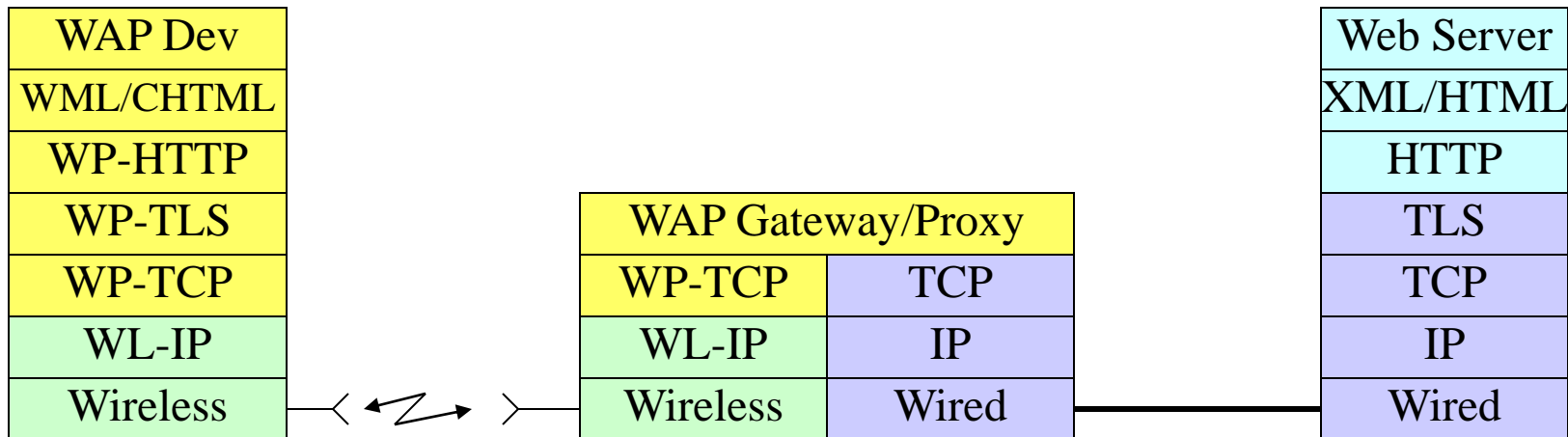
# WAP (Wireless Application Protocol)

- ❑ Performance of mobile terminal

	Desktop		Laptop		PDA		Cell Phone
CPU/NetB/Power	High	→	Middle	→	Low	→	Very Low
M/Storage/Screen	Large	→	Middle	→	Small	→	Very Small

- ❑ Need special OS: PalmOS, EPOC, Windows CE, OS/9, JavaOS
- ❑ Need special data representation, delivery, browser
- ❑ WAP Forum is the industry association to develop world standard for wireless information and telephony services on digital mobile phones, pagers, PDA and other wireless terminals
- ❑ WAP is an open, global specification for mobile users with wireless devices to easily access and interact with information and services
- ❑ WAP is an analogy of Internet protocol for wireless networks  
WAP → Integrated OMA (Open Mobile Alliance)

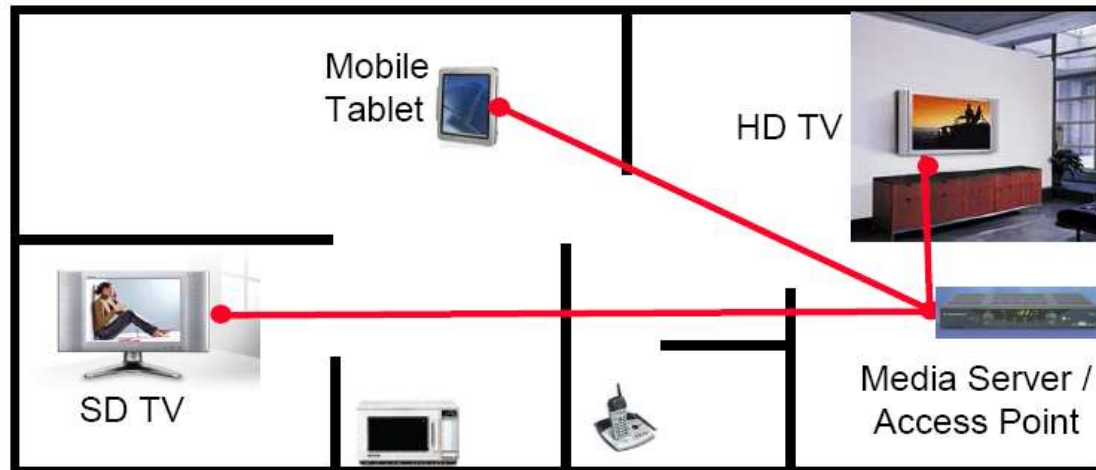
# WAP Protocol Stack



- ❑ WML (Wireless Markup Language) and simplified XHTML
- ❑ CHTML (simplified HTML) from DoCoMo is included for i-mod
- ❑ WP-HTTP is a wireless profiled HTTP, 1.1 compatible
- ❑ WP-TLS is a wireless profiled TLS (Transport Layer Security)
- ❑ WP-TCP is a wireless profiled TCP, optimized for wireless environment

# Data Transform for Optimal Wireless Delivery

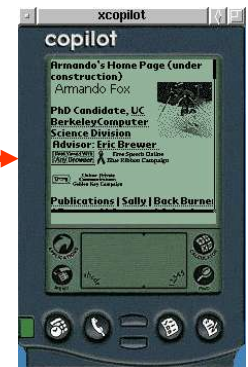
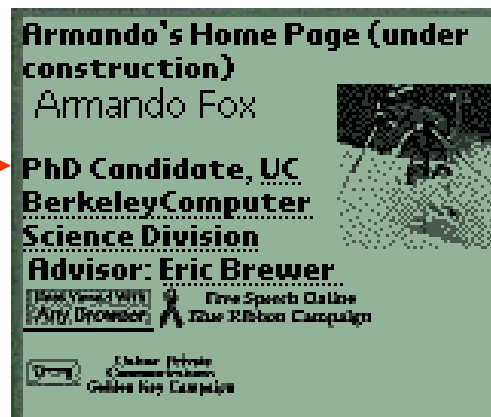
- Send out content from origin Internet servers to different devices/demands
  - Reduce media data via reduce quality: color, size, resolution, sample/frame rate
  - Devise/Content-based transformation, transmission, presentation, etc.
- ➔ called **transcoding, scalable coding, adaptive coding, ...**



**Armando Fox**

PhD Candidate, [UC Berkeley Computer Science Division](#)

Advisor: [Eric Brewer](#)



# Example of a Scalable Coding

Original



0.384  
Mbps



2 Mbps



0.144  
Mbps



# Mobile Multimedia Challenges

**Adaptive Decoding** - Optimizing rich digital media for mobile information devices with limited processing power, limited battery life and varying display sizes

**Error Resilience** - Delivering rich digital media over wireless networks that have high error rates and low and varying transmission speeds

**Low Power/Energy** – Multimedia computing and communication with less energy consumption, one of core issues in mobile multimedia terminals



# Audio & Video Tech. and Applications

**This Course** → Audio and Video technologies

→ Combine AV and CG/Web Technology in Mobile Devices

**セカイカメラ:** 頓智ドット株式会社 が開発したiPhone向けARアプリケーションのことで、リアルタイムに撮影している映像と重ねて、「エアタグ」と呼ばれる半透明のアイコンをタッチすれば、そのタグに関する詳細な情報が現れる。日本時間2009年9月24日iPhone App Storeで公開された

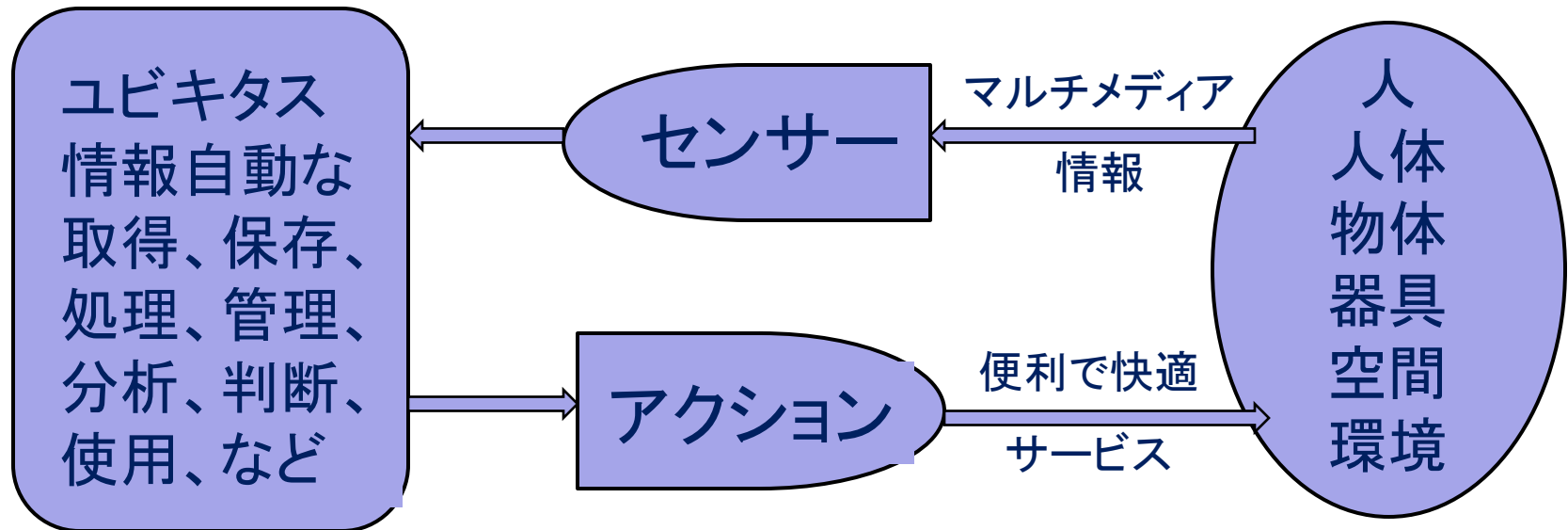


# Beyond Audio & Video

**MM** → Human's Five Senses

→ Various Sensors & Devices

→ Ubiquitous Media and Services



# Demos of MM Applications on Smart Phones