Part II Compression

- Requirements
- Classification
- Examples
- Entropy and Source Coding
- JPEG
- MPEG

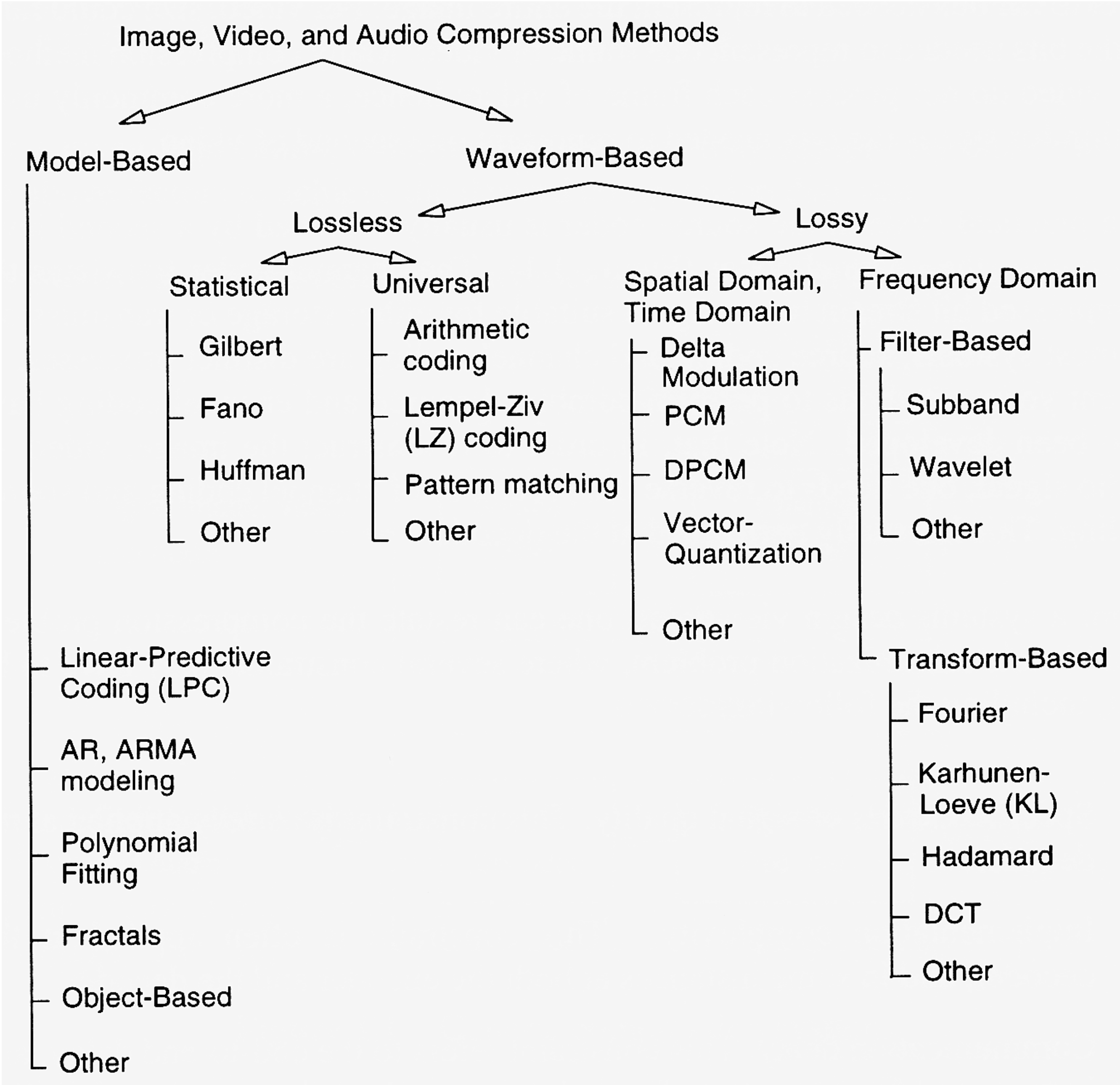
Compression Requirements

- for dialog mode applications
  - end-to-end delay (EED) should not exceed 150 ms
  - face-to-face application needs EED of 50 ms (incl. comp.)
- retrieval mode applications
  - fast forward and backward data retrieval with simultaneous display
  - random access to single images and audio samples, access time < 0.5 s
- decompression without a link to other data units—allows random access and editing

Compression Requirements

- for dialog and retrieval mode applications
  - support scalable video in different systems
  - support of various audio and data rates
  - synchronization of audio and video
  - economical solutions

Compression Taxonomy



Compression Techniques-Classification

entropy coding	run-length coding	
	Huffman coding	
	arithmetic coding	
source coding	prediction	DPCM
		DM
	transformation	FFT
		DCT
	vector quantization	
hybrid coding	JPEG	
	MPEG	
	px64	
	DVI (RTV, PLV)	

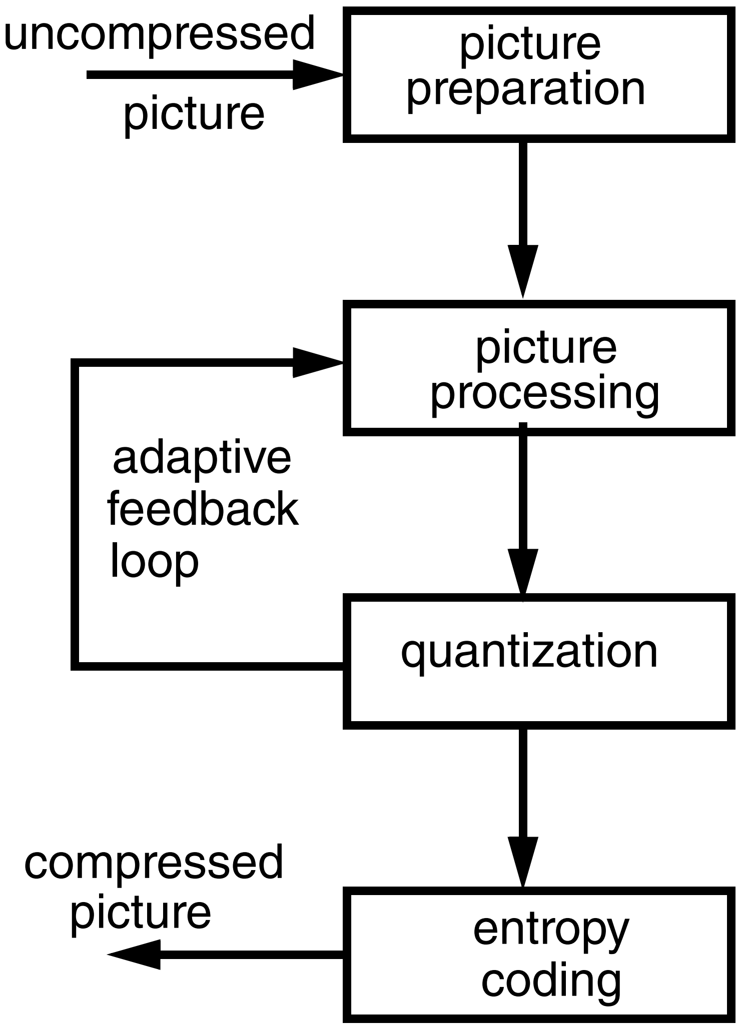
Classification (continued)

- entropy coding
  - lossless encoding
  - used regardless of media's specific characteristics
  - data taken as simple digital sequence
- source coding
  - lossy encoding
  - takes into account the data semantics
  - degree of compression depends on data content

Decompression Requirements

- dialog mode applications need symmetric compression
- retrieval mode applications need asymmetric compressions
  - compression is performed once and sample time available
  - decompression is performed frequently and needs to be done fast

Compression Steps



- analog-to-digital conversion
- generation of appropriate digital representation
- image division into 8 x 8 blocks
- fix number of bits/pixel

Compression Algorithm

- transformation from time to frequency domain e.g., using DCT
- motion vector application for motion video

- mapping of real numbers to integers (reduction in precision)
  example: μ-law audio—from 12 bits to 8 bit quantization

- compress a sequential digital stream without loss

Run-length Encoding

- entropy coding algorithm—content dependent coding
- replaces the sequence of the same consecutive bytes with the number of occurrences
- the number of occurrences is indicated by a *special flag (!)*
- procedure:
  - if the same byte occurs at least 4 times count number of occurrences
  - write compressed data in the format
    the counted byte ’!’ number of its occurrences

Run-length Encoding

- example:
  - uncompressed sequence (20 Bytes):
    ABCCCCCCCCCDEFFFFGGG
  - compressed sequence (13 Bytes):
    ABC!9DEF!4GGG
- variations
  - zero suppression technique
  - text compression technique (also for images, video. audio)
  - diatomic encoding

Statistical Encoding

- frequency dependent encoding—belongs to entropy encoding
- given is a sequence of symbols: S_1, S_2, S_3. .... and the probability of occurrence of each symbol $p(S_i)=p_i$
- Ex.: p(A)=0.16, p(B)=0.51, p(C)=0.09, p(D)=0.13, p(E)=0.11
  first choice: encode A,B,C,D,E as 000,001,010,011,100
- Q: What is the minimum number of bits per symbol?
- A: The theoretical minimum average number of bits per code word is known as *entropy* (H), according to Shannon

  $H = - \Sigma\, p_i \log_2 p_i$ bits per code word (Ex: 1.36 bits/symbol)

Huffman Encoding

- statistical encoding, depends on occurrence frequency of single characters or sequences of data bytes
- characters stored with their probabilities
- length (number of bits) of the coded characters differs
- shortest code assigned to the most frequently occurring character
- to determine Code, it is useful to construct a binary tree
  - leaves are characters to be encoded
  - nodes carry occurrence probabilities of the characters belonging to the subtree

Huffman Encoding—Example

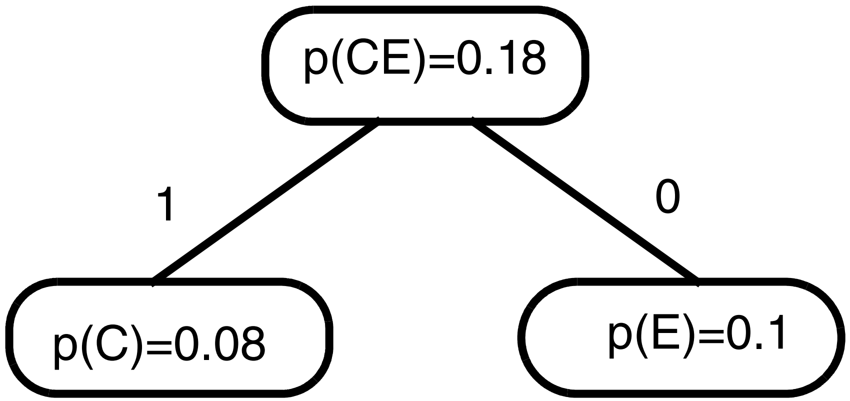
build a Huffman Tree:

p(A)=0.17, p(B)=0.51, p(C)=0.08, p(D)=0.14, p(E)=0.1

```
build ordered list of symbols (increasing probability)
do while list contains at least 2 elements
    construct tree using the first two elements in list
    add parent node for the union of these elements and
        compute probability
    mark edges by '0' and '1'
    delete the first 2 elements in list; insert parent into list
end
```

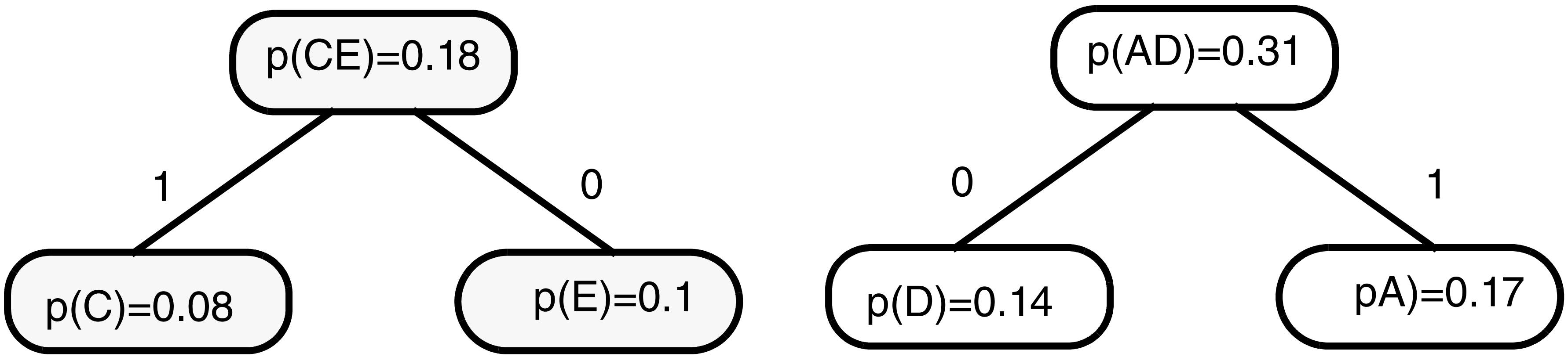
Huffman Encoding—Example

step 1: p(C)=0.08, p(E)=0.1, p(D)=0.14, p(A)=0.17, p(B)=0.51



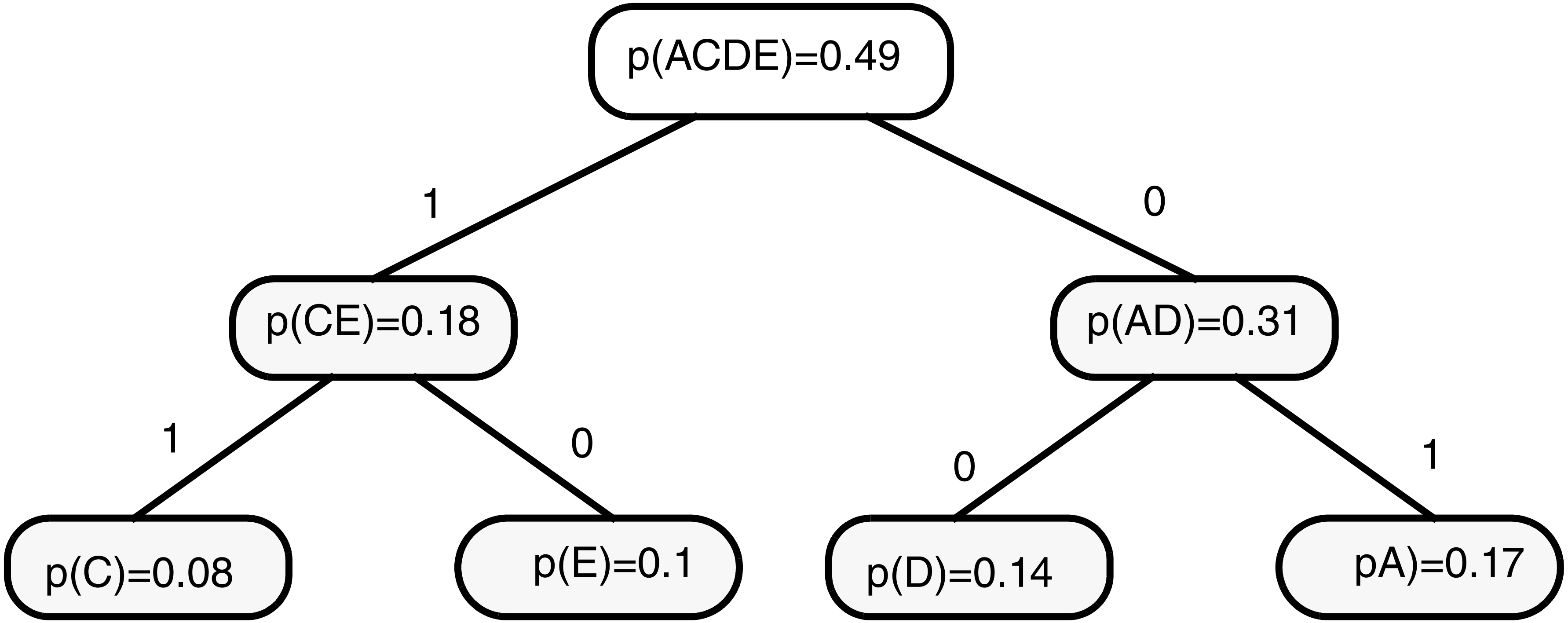
Huffman Encoding—Example

step 2: p(D)=0.14, p(A)=0.17, p(CE)=0.18, p(B)=0.51



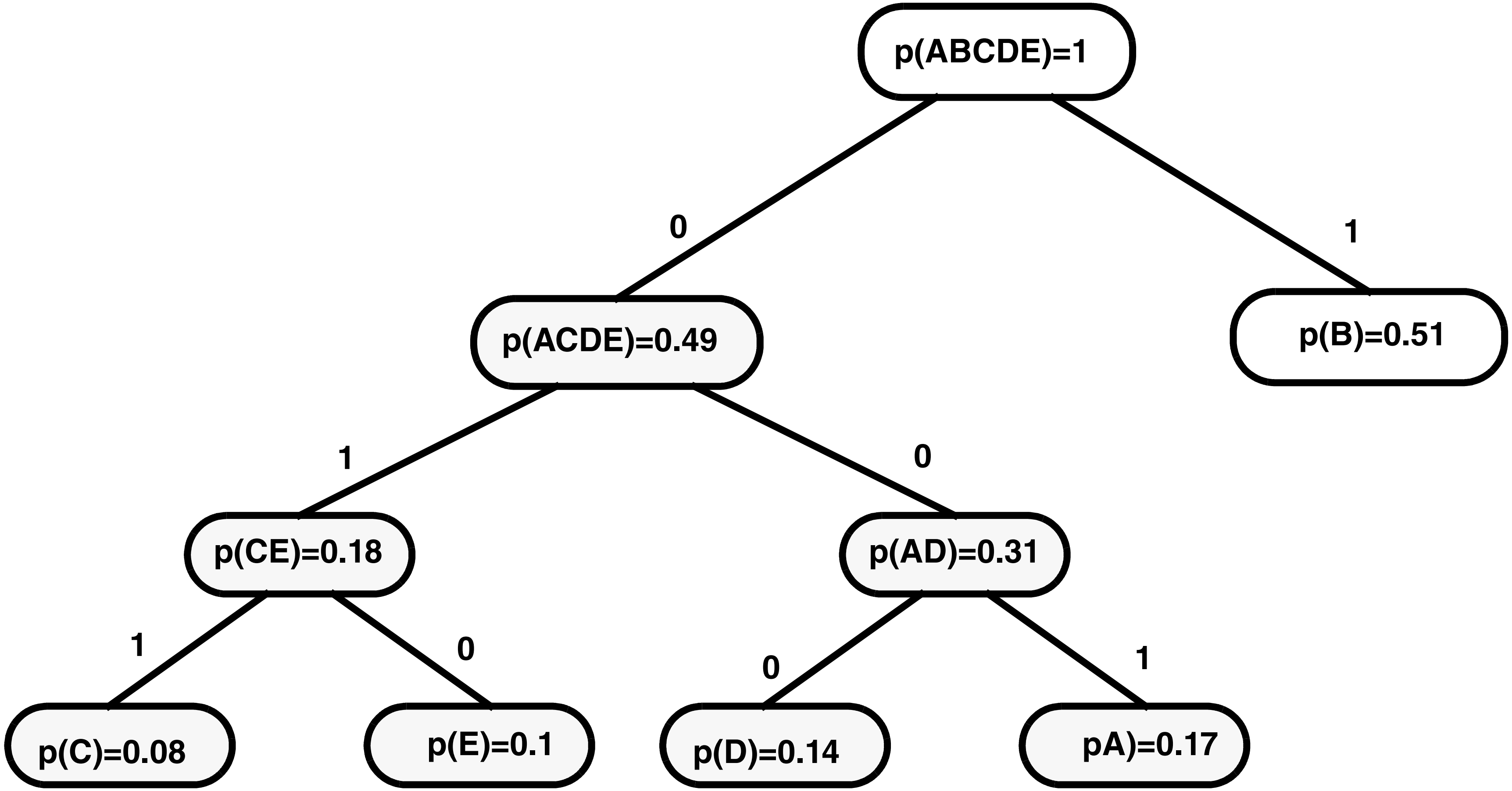
Huffman Encoding—Example

step 3: p(CE)=0.18, p(AD)=0.31, p(B)=0.51



Huffman Encoding—Example

last step: p(ACDE)=0.49, p(B)=0.51



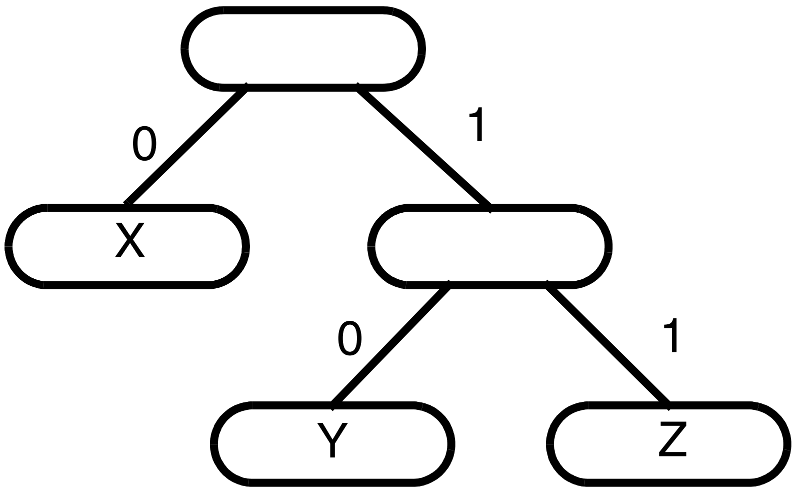
Huffman Encoding—Example

encode symbols according to tree

A	001
B	1
C	011
D	000
E	010

Huffman Decoding—Example

given: 000101011 and Huffman Tree



reconstruct message from bit stream
using finite state machine

Application of Huffman Coding for Video

- if the image information can be transformed into a bit stream, Huffman table can be used to compress data without loss
- simple way to generate a bit stream is to code pixels individually and read them line by line
- for video: Huffman table can be used for a single sequence of images
    - for a set of scenes
    - for an entire film clip
- need Huffman table for encoding and decoding

Differential Encoding

- belongs to source coding
- consider sequence of symbols S_1, S_2, S_3, ... where values are not zeros, but do not vary much; calculate difference from previous value
- Ex.: still images—calculation of differences between nearby pixels or pixel groups

0	0	0	0	0
0	255	94	87	100
0	0	0	0	255
0	0	0	0	0

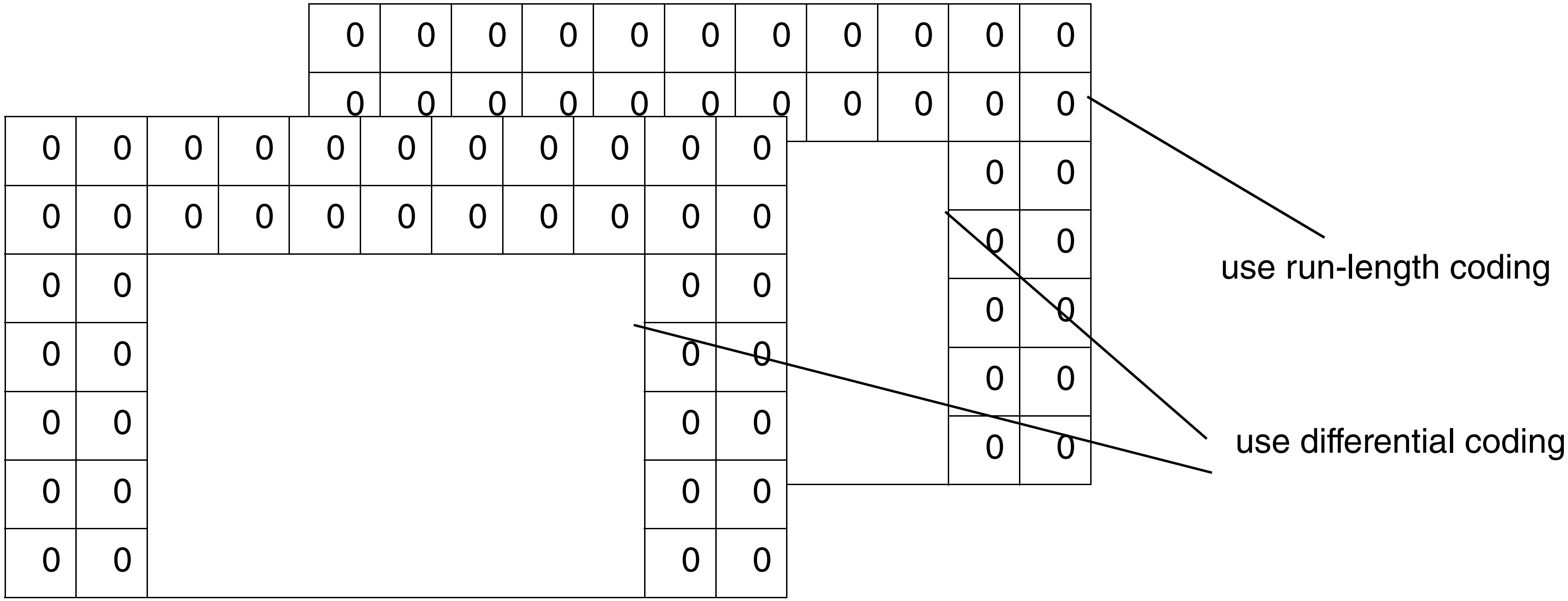
edges result in large values

similar chrominance or luminance values result in small values

- zeros can be suppressed by run-length coding

Differential Coding in Video (Examples)

- static background (videoconferencing, newscast, etc.)



- motion compensation—8x8 blocks are compared; areas similar, only shifted, e.g., to the right (*motion vector*)

Joint Photographic Experts Group (JPEG)

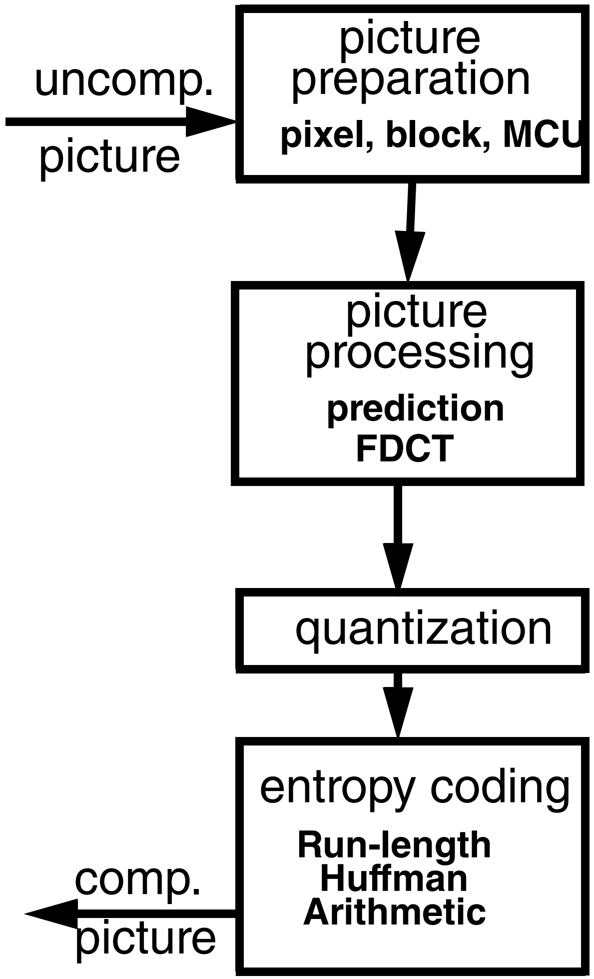
requirements:

- to be near the state-of-the-art for degree of compression versus image quality,
- to be parameterizable (user can select parameters),
- to be applicable to any kind of source image, without regard to dimensions, content, image aspect ratio, pixel aspect ratio, etc.,
- to have computational requirements that are reasonable for both, hw and sw implementations,
- to run on as many standard platforms as possible

Joint Photographic Experts Group (JPEG)

- to support four different modes of operation:
  - sequential encoding—encoded in same order as scanned
  - progressive encoding—multiple pass encoding
  - lossless encoding
  - hierarchical encoding—encoded at multiple resolutions

JPEG Processing Steps



Sequential Encoding	**Progressive Encoding**	**Lossless Encoding**	**Hierarchical Encoding**
block, MCU 8-bit/pixel	12 bit/pixel	2-12 bit/pixel	
Transformation Source Coding lossy DCT	Layered Coding	Predictive Entropy Coding	switch between lossy DCT and loss-less technique
Run-length Huffman			

JPEG General Image Model

- not based on
    - 9-bit YUV coding
    - fixed number of lines, columns
    - mapping of encoded chrominance
- independence from image parameters
- source image consists of 1 to 255 components (planes)
- all pixels of all components within the same image are coded with the same number of bits

JPEG Image Preparation

- images divided in data units (blocks), DCT operates on blocks
- lossy mode operates on 8x8 pixel blocks
  lossless mode operates on data units of 1 pixel
- in most cases data units are processed component by component and passed to image processing
- processing order of data units
  - left-to-right, top-to-bottom
  - interleaved data ordering
- interleaved data units of different components are combined to *minimum coded units* (MCU)

JPEG Image Preparation

after image preparation step:

- uncompressed image samples are grouped into data units of 8x8 pixels and passed to JPEG encoder
- order of data units is defined by MCUs
- values in the range [0, 255]

JPEG Image Processing

first step:

- pixel values shifted into the range [-128, 127]
- values in the 8x8 pixel blocks are defined by S_{yx}, y,x ε [0,7]
- Forward Discrete Cosine Transformation (Forward DCT) maps values from time to frequency domain

$$svu = \frac{1}{4} cu cv \sum_{x=0}^{7} \sum_{y=0}^{7} syx \cos\frac{(2x+1)u\pi}{16} \cos\frac{(2y+1)v\pi}{16}$$

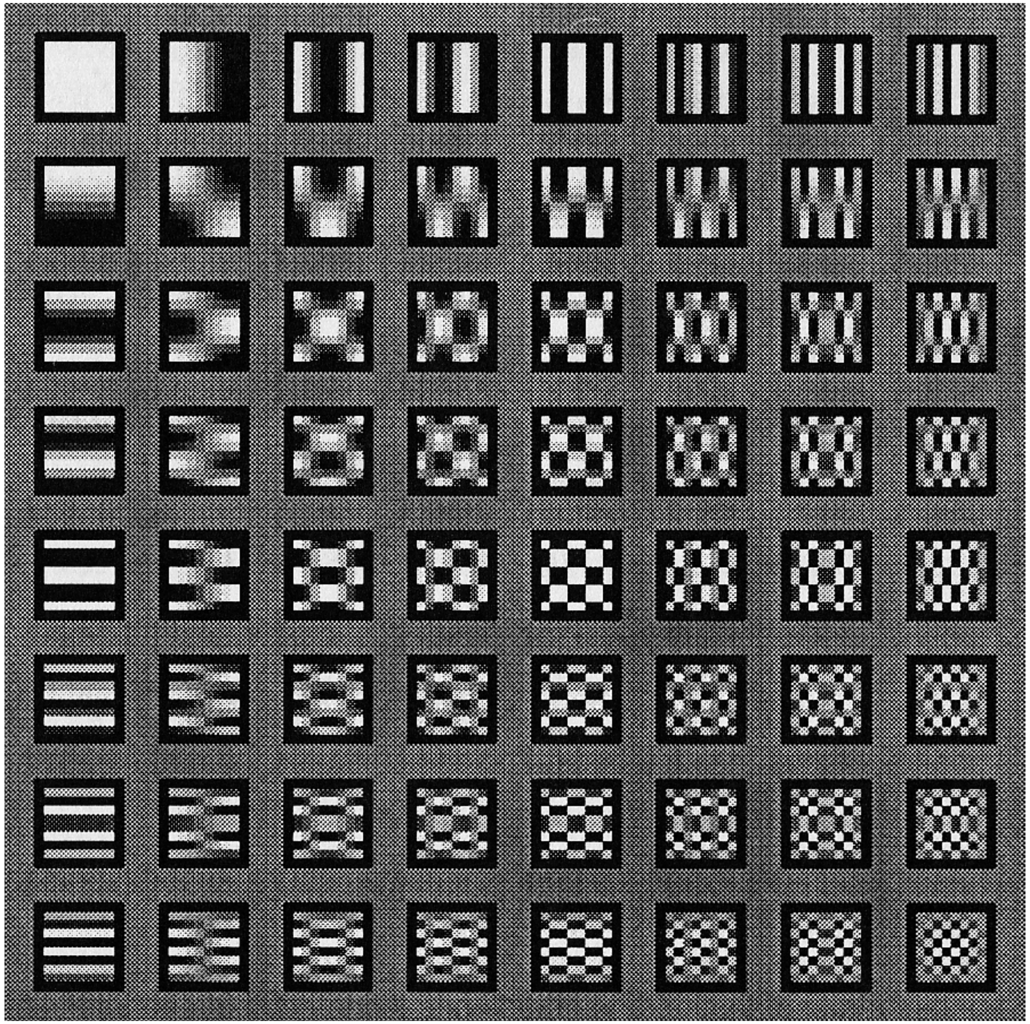
with c_u, c_v = $1/\sqrt{2}$ for u,v=0; otherwise c_u, c_v = 1

JPEG Image Processing (DCT)

S(u,v) coefficients

- S(0,0)—lowest frequency in both directions is called DC coefficient
  - determines the fundamental color of the block
  - frequency = 0 in both directions
- S(0,1), ..., S(7,7) are called AC coefficients
  - frequency in one or both directions non-zero
- computing of DCT—use factoring

DCT basis functions



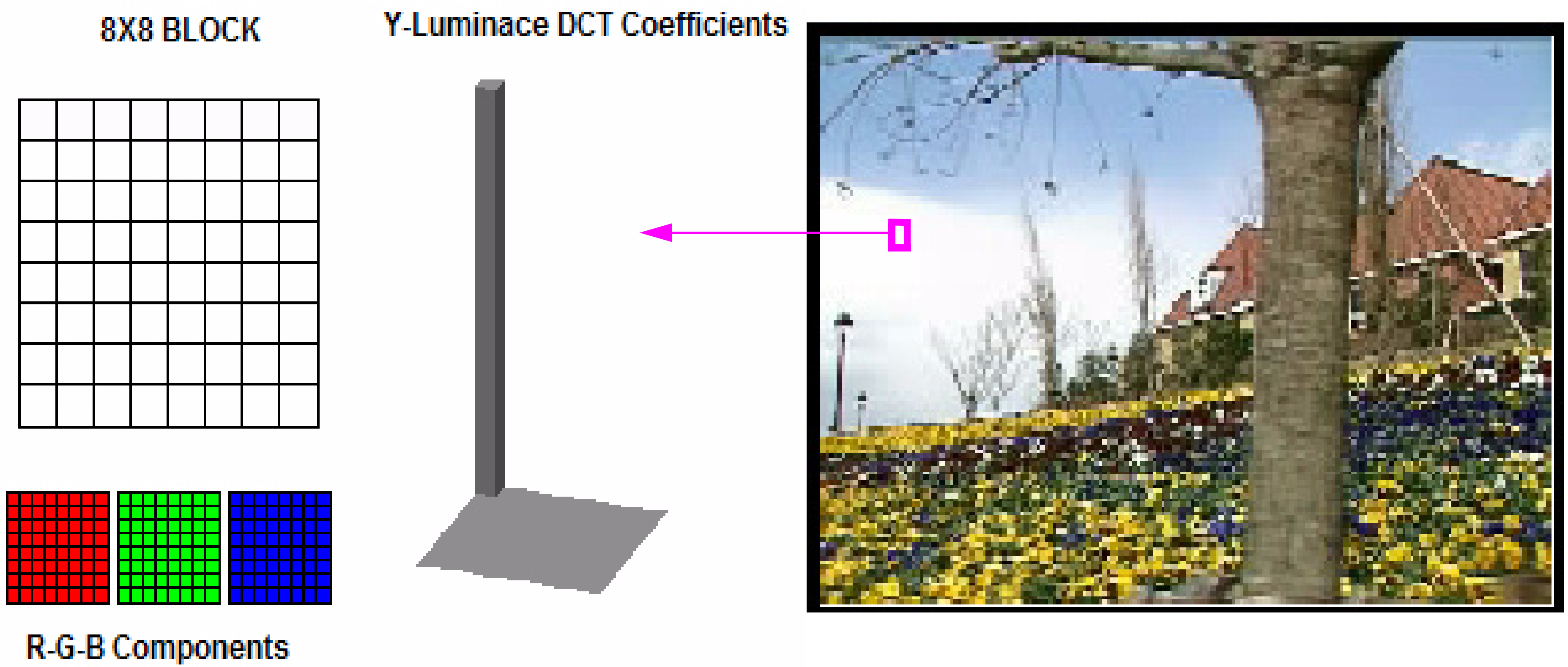
Discrete Cosine Transform

human beings have only a restriced capability to perceive high frequences. Images reduced from their high frequent elements may have a high degree of quality while being compressed quite considerably

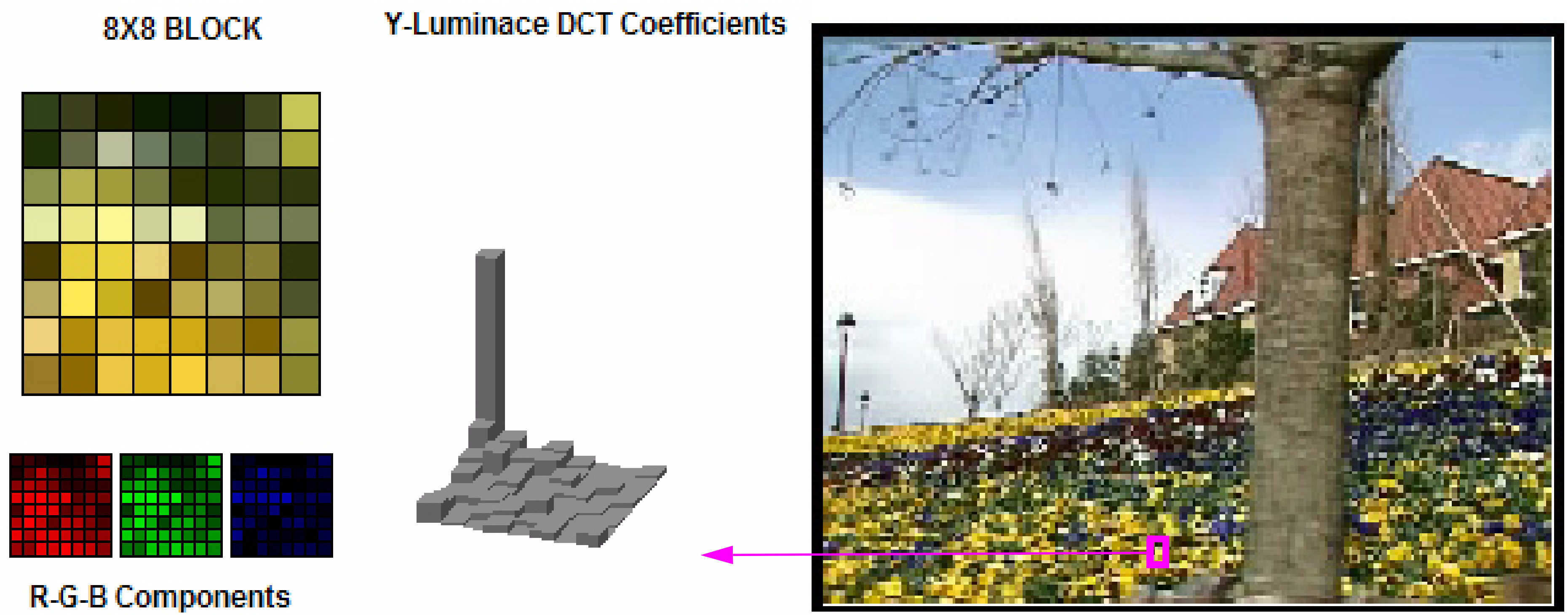
- Low frequency components of an image correspond to brief outlines of image objects
- High frequency components of an image correspond to fine structures

To decrease the high frequencies the data must be sorted by frequencies $\rightarrow$ DCT

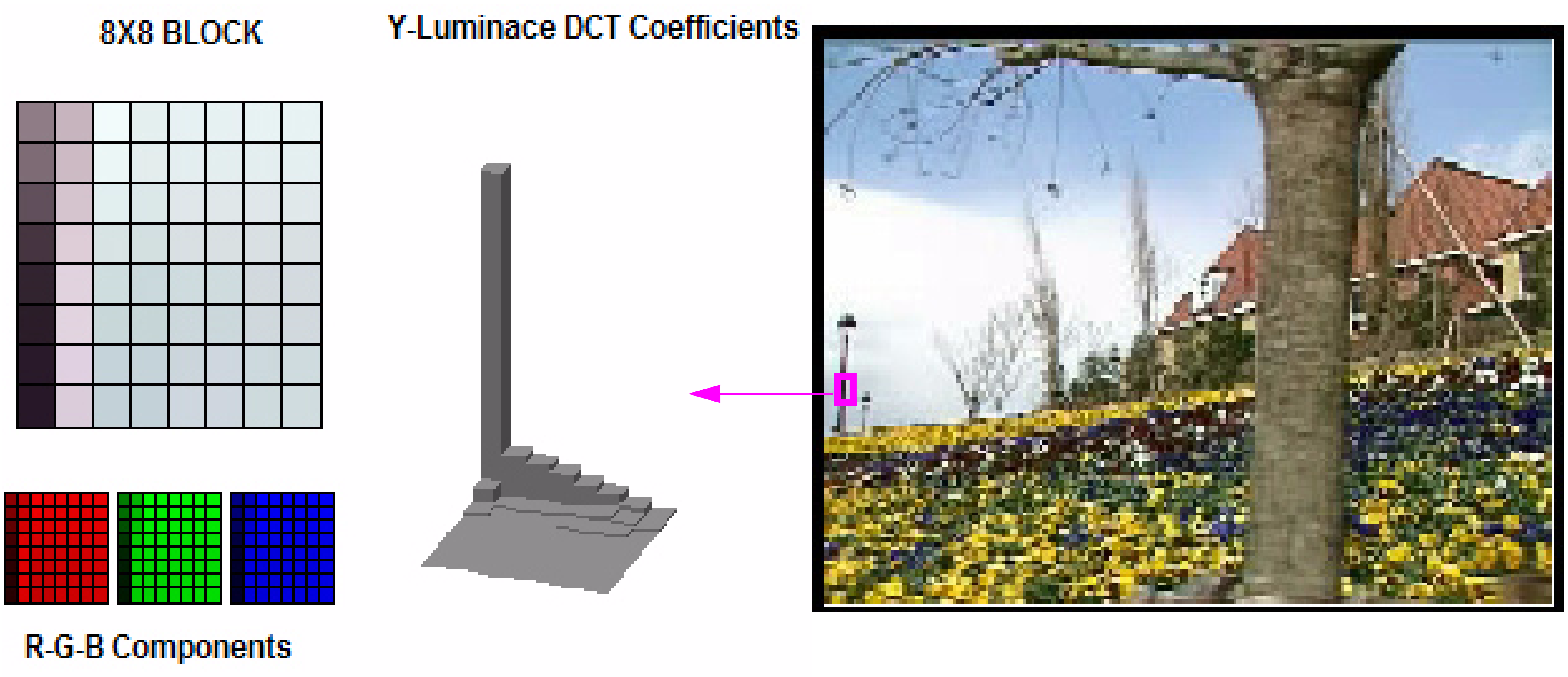
DCT-Coefficients



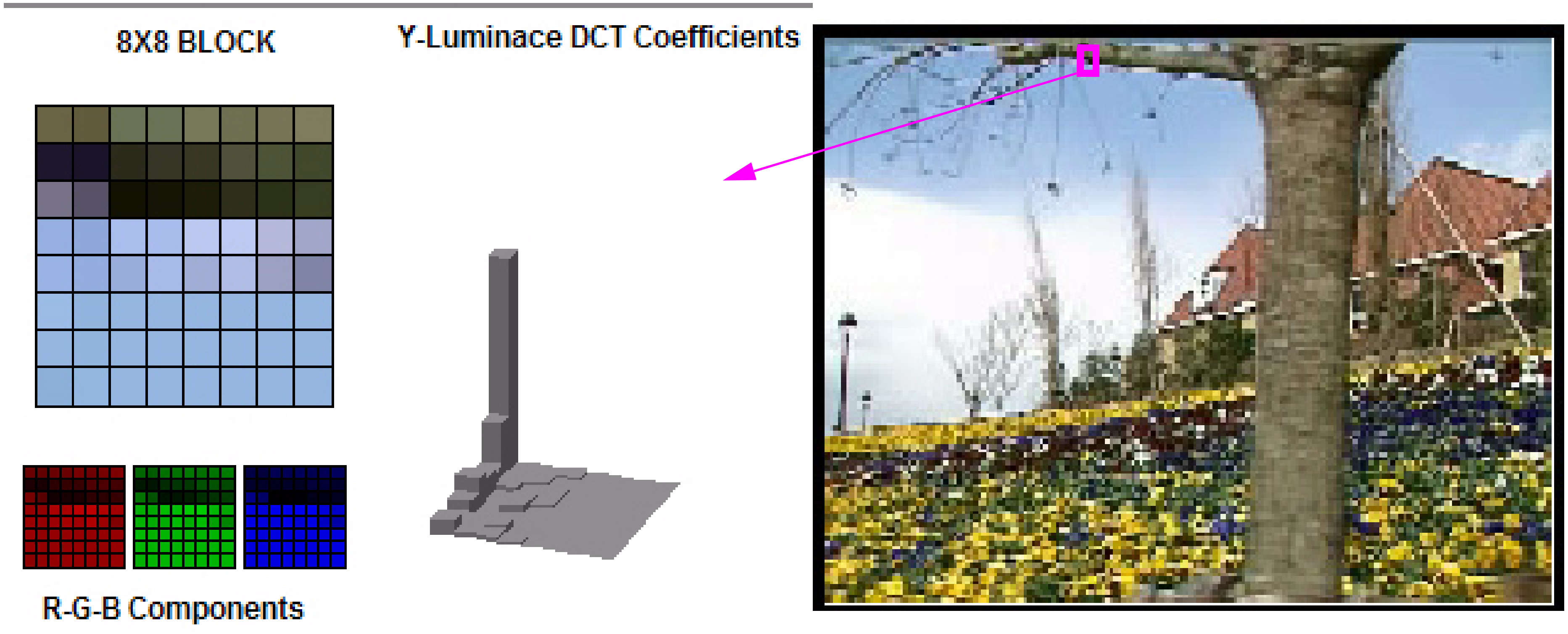
DCT-Coefficients



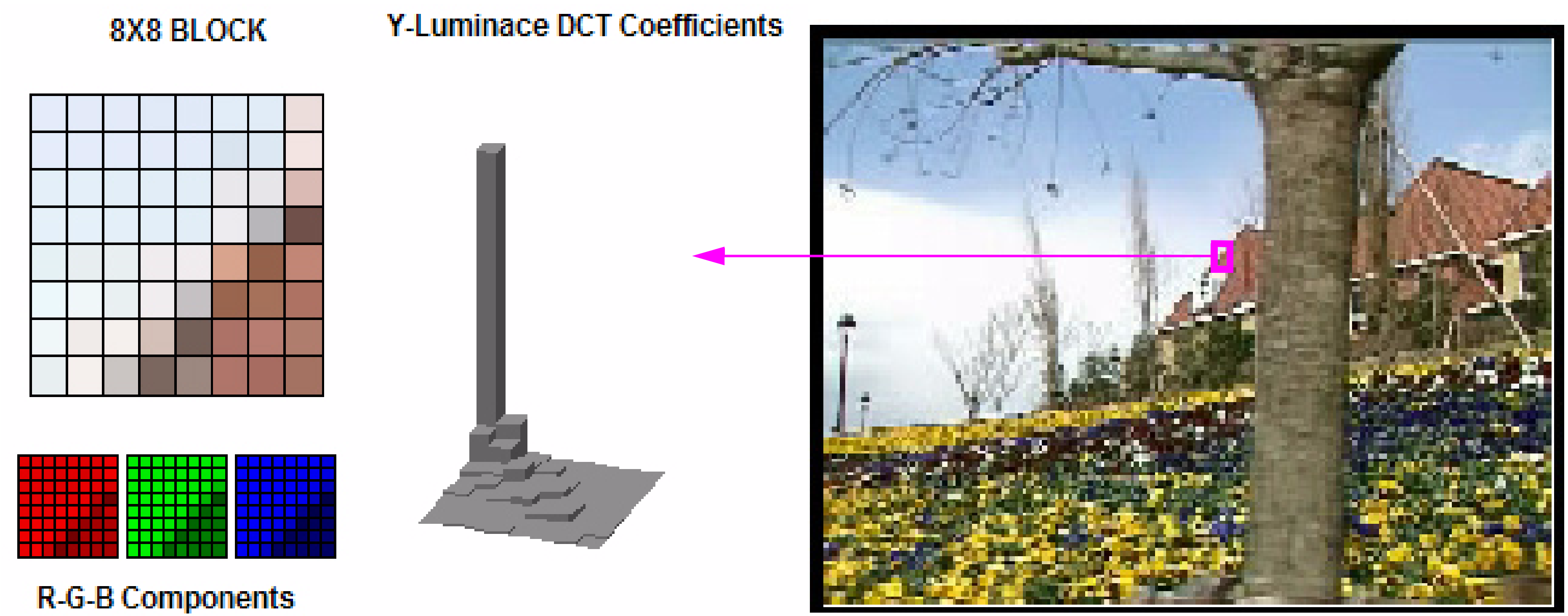
DCT-Coefficients



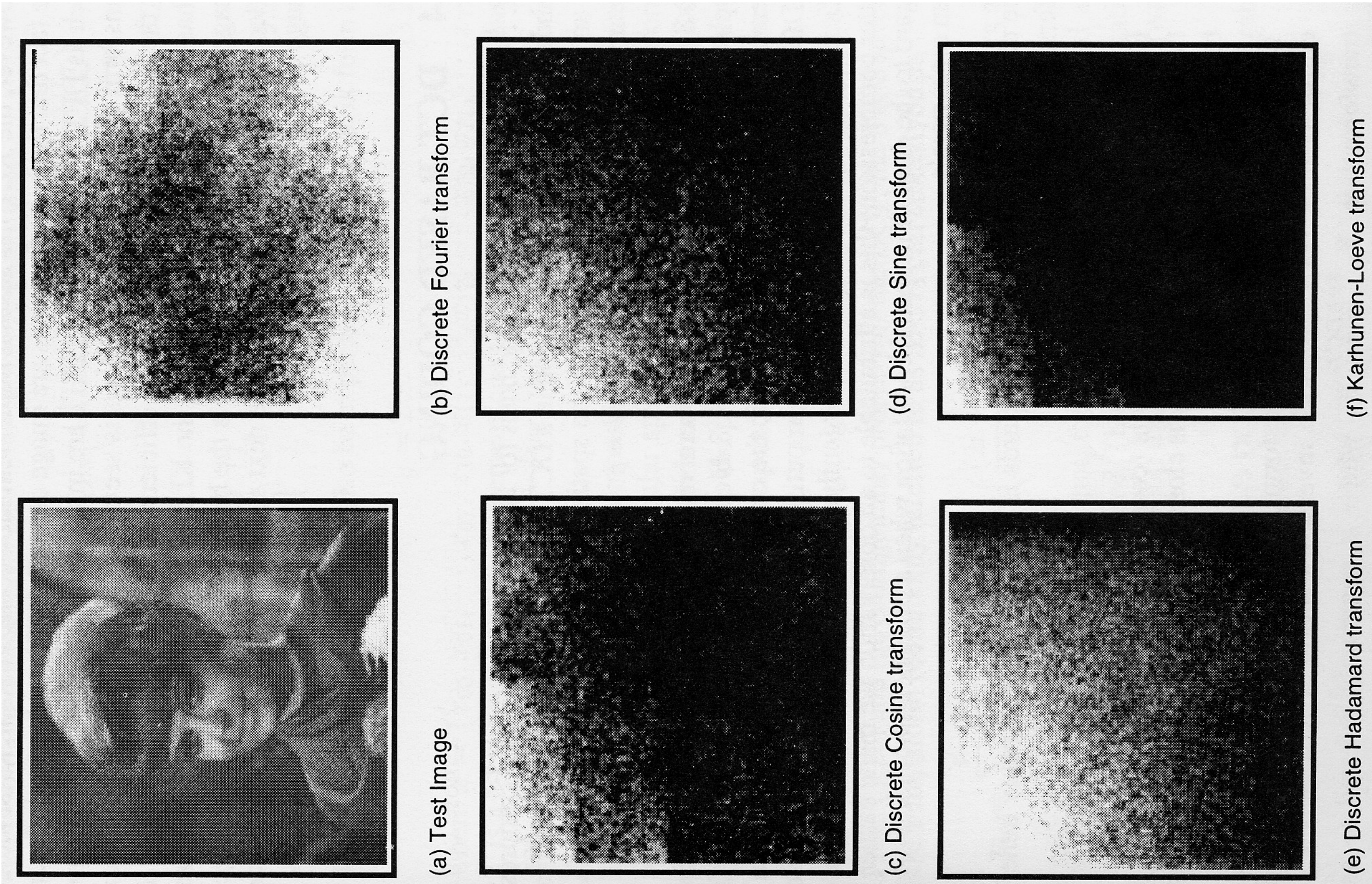
DCT-Coefficients



DCT-Coefficients



Why DCT?


(a) Test Image
(b) Discrete Fourier transform
(c) Discrete Cosine transform
(d) Discrete Sine transform
(e) Discrete Hadamard transform
(f) Karhunen-Loeve transform

JPEG Image Processing (DCT)

- Inverse Discrete Cosine Transformation (I-DCT) maps DCT coefficients to sampled values

$$s_{xy} = \frac{1}{4} \sum_{u=0}^{7} \sum_{v=0}^{7} c_u c_v s_{vu} \cos\frac{(2x+1)u\pi}{16} \cos\frac{(2y+1)v\pi}{16}$$

with c_u, $c_v = 1/\sqrt{2}$ for u,v=0; otherwise c_u, $c_v = 1$

- DCT and I-DCT cannot be calculated in full precision ==> lossy compression
- JPEG does not define precision parameters, therefore various implementations exist
- many AC coefficients with small values (around zero)

JPEG Quantization

- goal: to “throw out” bits (truncation)
- uniform quantization: divide coefficient values S(u,v) by N and round result

Q: In S(u,v) how many bits should be truncated?

A: use quantization tables

- quantization tables consist of 64 elements, each value uses 8-bit: Q_{uv}
- new compressed values by using tables
  $Sq_{uv} = S_{uv} / Q_{uv}$
- standard defines two default tables (luminance, chroma)

JPEG Quantization

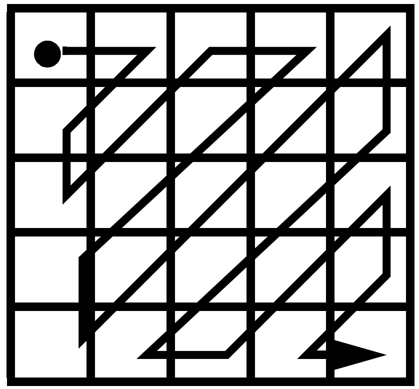
luminance quantization table

16	11	12	14	12	10	16	14
13	14	18	17	16	19	24	40
26	24	22	22	24	49	35	37
29	40	58	51	61	60	57	51
56	55	64	72	92	78	64	68
87	69	55	56	80	109	81	87
95	98	103	104	103	62	77	113
121	112	100	120	92	101	103	99

human eye most sensitive to low frequencies (upper left corner)

JPEG Entropy Encoding

- first step: map 8x8 pixel blocks into 64 element vector using zig-zag-scan



- DC coefficients processing:
  - DC coefficients determine basic color of data unit
  - DC coefficient is large, but often close to previous value ==> encode difference

JPEG Entropy Encoding

DC_0	DC_1	DC_2
DC_3	DC_4	DC_5
DC_6	DC_7	DC_8

Predictor

DC_0	$Diff_1$	$Diff_2$
$Diff_3$	$Diff_4$	$Diff_5$
$Diff_6$	$Diff_7$	$Diff_8$

$$Diff_i = DC_i - DC_{i-1} \text{ for } i>0$$

- AC coefficients processing:
  - processing order of the AC coefficients using zig-zag scan (coefficients with lower frequencies are encoded first) ==>
  - sequence of similar data bytes ==> efficient entropy coding
- JPEG standard specifies Huffman or Arithmetic coding, but sequential encoding mode uses only Huffman coding.

JPEG — Entropy Encoding

Algorithm:

1. apply run-length coding of AC coefficient of zero values
2. apply Huffman coding on DC and AC coefficients

(only DC coefficients explained in detail)

JPEG — Entropy Encoding

DC coefficients—Huffman coding

- categorize DC values into DC code tables
- difference magnitude categories for DC coefficients (12 categories)

Diff Values	SSSS (number of bits needed to encode Diff
0	0 bit
-1, 1	1 bit
-3, ..., 3	2 bits
-7, ..., 7	3 bits
.	
.	
.	
-2047, ..., 2047	11 bits

JPEG — Entropy Encoding

DC coefficients —Huffman encoding

- handle SSSS as Huffman symbol, get p(0), p(1), ... p(11) and create Huffman tree ==> Huffman code for SSSS
- for each category, an additional bits field is appended to the code word to uniquely identify which difference in that category actually occurred.
- send: (Huffman codeword, actual value)

EX: if SSSS=2 has the Huffman code 001 and Diff=-3 ==>

$$\text{send } \underbrace{001}_{\text{Huffman code for 2}}\underbrace{00}_{\text{-3 in two's complement is 00 because 3=>11}}$$

Comments

- applications do not have to include both an encoder and decoder if the compression process agrees on a common table
- the encoded data stream has a fixed *Interchange Format*
  - encoded image data
  - chosen parameters
  - tables of the coding process
- in regular mode, the interchange format includes all of the information necessary for decoding without any previous knowledge of the coding process

JPEG Block Encoding (Example)

139	144	149	153
144	151	153	156
150	155	160	163
159	161	162	160

DCT

DC component

AC components

1260	-1	-12	-5
-23	-17	-6	-3
-11	-9	-2	2
-7	-2	0	1

quantize

79	0	-1	0
-2	-1	0	0
-1	-1	0	0
0	0	0	0

zig-zag-scan

79	0	-2	-1	-1	-1	0	0	-1	0	0	0	0	0	0	0

run-length code

0	79
1	-2
0	-1
0	-1
0	-1
2	-1
0	0
79	0

Huffman code

10011011100011 .....

coded bitstream 0.55 bits/pixel

JPEG Coding/Decoding (Example)

original block

139	144	149	153
144	151	153	156
150	155	160	163
159	161	162	160

reconstructed block

144	146	149	152
148	150	152	154
155	156	157	158
160	161	161	162

errors

-5	-2	0	1
-4	1	1	2
-5	-1	3	5
-1	0	1	-2

Motion Picture Experts Group (MPEG)

- MPEG Objectives and Standards
- General Information about MPEG
- MPEG-1
- MPEG-2
- MPEG-4
- (MPEG-7 und MPEG-21)

MPEG Objectives

- to deliver acceptable video quality at compressed data rates between 1.0 and 1.5 Mbps (MPEG-1)
- to support either symmetric or asymmetric compress/ decompress applications
- when compression takes it into account, random-access playback is possible
- when compression takes it into account, fast-forward, fast-reverse or normal reverse playback modes are available
- audio/video synchronization will be maintained

MPEG Objectives

- catastrophic behavior in the presence of data errors should be avoidable
- when required, compression/decompression delay can be controlled
- editability should be available when required by applications
- sufficient format flexibility to support playing of video in windows
- the processing requirements should not preclude the development of low-cost chipsets which are capable of encoding in real-time

MPEG Standards

Standard specifies audio, video and system layers

- several standards defined
  - MPEG-1 targeted at low data rates (VHS quality at 1.5 Mbits/sec) [1992]
  - MPEG-2 targeted at high quality, hence high data rates (studio quality up to 15Mbits/sec) [1994]
  - MPEG-4 targeted at very low bit rates (<64 kbs) with small images [1998]

MPEG General Information

- MPEG standard defines audio, video coding and system data streams with synchronization
- MPEG considers explicitly functionalities of other standards, e.g. it uses JPEG
- MPEG stream provides information on
  - aspect ratio (e.g., 1:1, 4:3, 16:9)
  - refresh frequencies (8 frequencies encoded: 23.976 Hz. 24 Hz, 25 Hz, 29.97 Hz, 30 Hz, 50 Hz, 59.94 Hz and 60 Hz)

MPEG-1 General Information

- audio and video compression designed to operate at CD-ROM speeds (video—1.150 Mbits/sec, audio—0.256 Mbits/sec, system 0.094 Mbits/sec)
- audio compressor uses subband coder with psychoacoustic model to reduce bit rate
- video compressor uses block transform coder with motion-compensated inter-frame coding
- must distinguish syntax and semantics in the standard—syntax is much more flexible than current standard (wide variation in usage: e.g., variable/constant bit rates, higher bit rates, larger images)

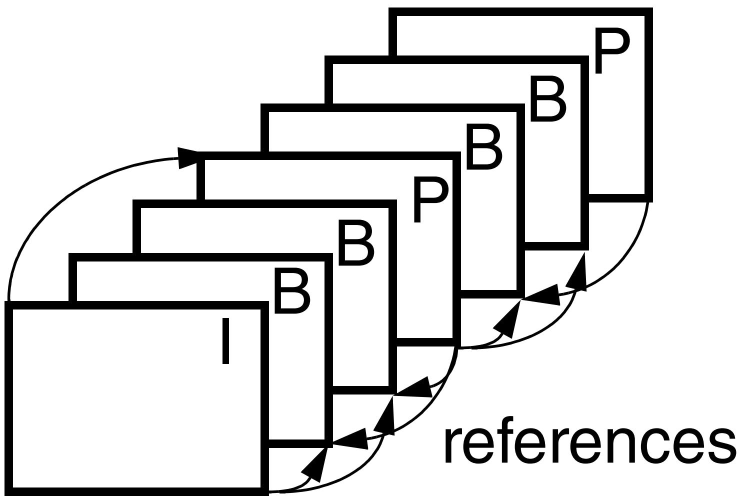
MPEG Video Standard

- each image consists of three components (1 luminance, 2 chrominance with half resolution)
- pixel precision—8 bits for each component
- example of a video format: 352x240 pixels, 30 frames/sec, chrominance components: 176x120 pixels
- each image is divided into areas called macroblocks (useful for compression based on motion estimation)
- each macroblock is partitioned into 16x16 pixels for luminance, 8x8 pixels for chrominance components

MPEG Video Standard

Video/Image Processing

- 4 types of image coding for video processing



why?

- demand for an efficient coding scheme and fast random access
- achieve high compression rates; exploit temporal redundancies of subsequent frames (interframe)

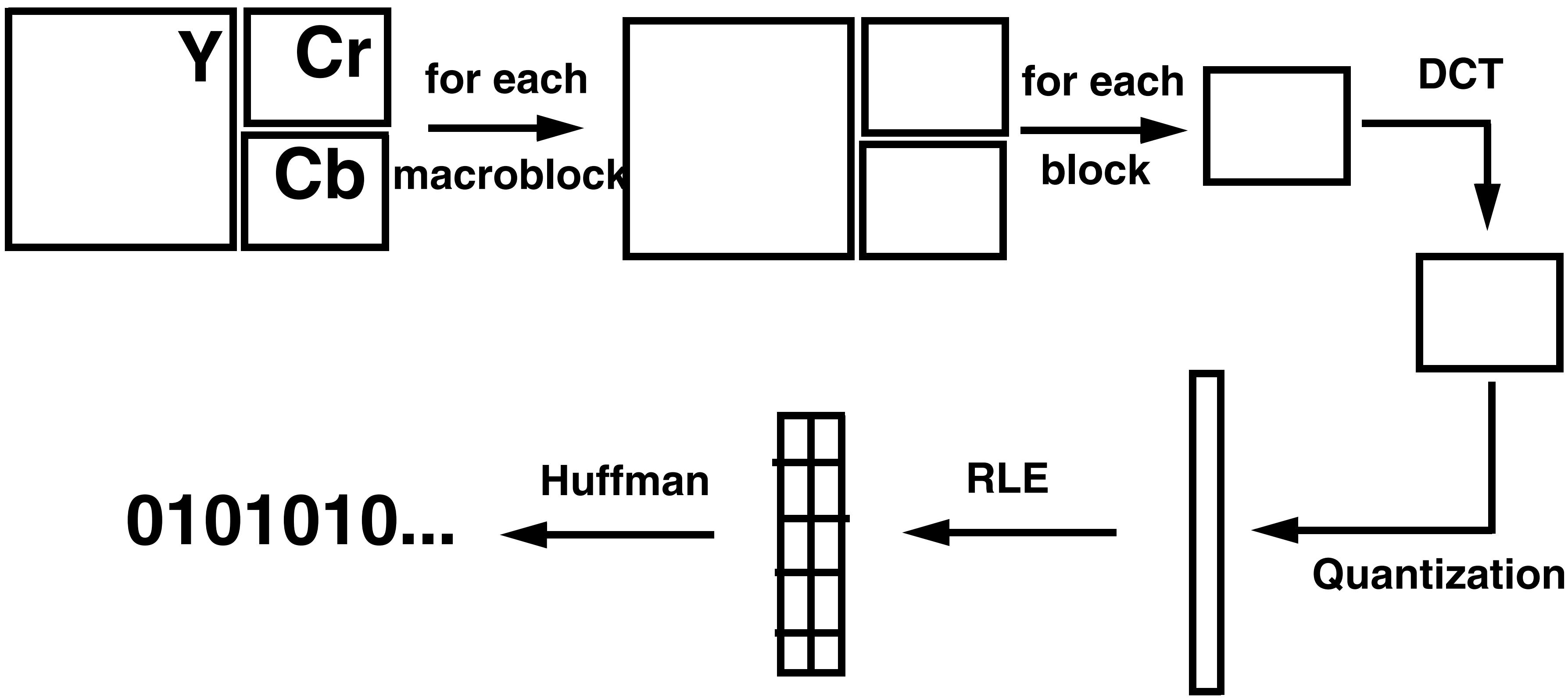
MPEG Video Standard

Intra-coded images (I-frames)

- self-contained without any references to other images, treated as still images; MPEG uses JPEG for I-frames
- The compression rate of I-frames is the lowest within MPEG.
- I-frames are points to random access in MPEG stream
- i-frames use 8x8 blocks defined within a macroblock, on these blocks DCT is performed. Quantization is by constant value for all DCT coefficients.

MPEG Video Standard

Intra-coded images (I-frames)



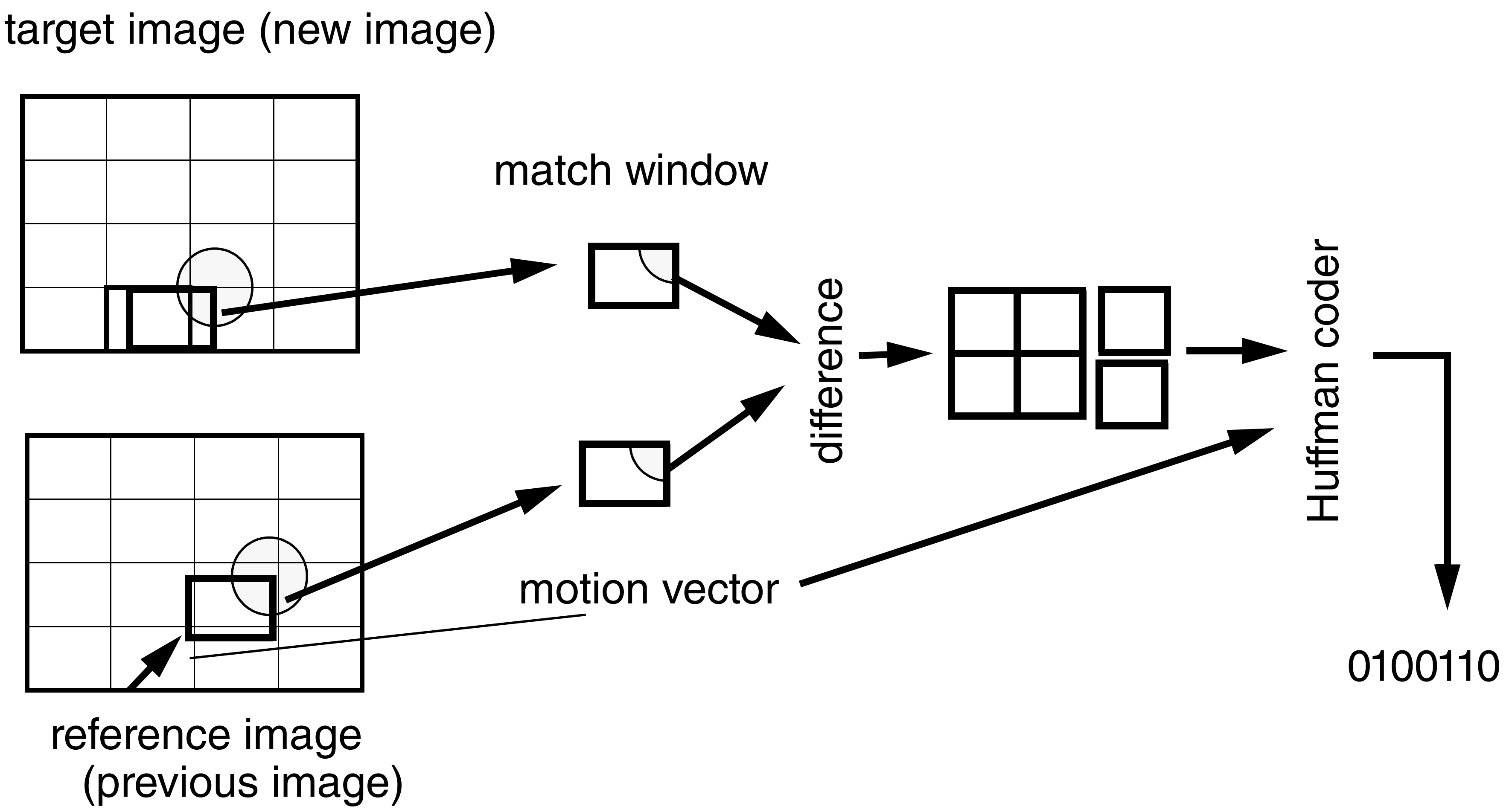
MPEG Video Standard

Predictive-coded Frame (P-frame)

- require information of the previous I-frame and/or previous P-frames for encoding and decoding
- coding of P-frames: utilize successive images in which areas do not change or are shifted
- temporal redundancy: determine the last P-frame or I-frame that is most similar to the block under consideration
- use motion estimation method at the encoder

MPEG Video Standard

motion estimation



MPEG Video Standard (P-Frame)

- motion vector—a pair (x-offset, y-offset) that specifies a block elsewhere in the frame
- block coded by specifying motion vector and error term between source block and motion vector block
- block can also be skipped - just use previous block at that location
- the coder must determine if a macroblock should be coded predictively or as a macroblock of an I-frame
- quantization value per macroblock, we want to vary quantization to fine tune compression

MPEG Video Standard

Motion Computation (P-Frames)

- look for match window within a given search window
  - match window; macroblock
  - search window: how far away are we willing to look
- methods:
  - 1. SSD correlation $\Sigma (x_i - y_i)^2$
  - 2. SAD correlation $\Sigma |x_i - y_i|$

MPEG Video Standard

P-Frame Processing:

- apply 2D DCT to macroblocks not reduced
- motion vector of adjacent macros often differs slightly
- the maximum size of the motion vector is not defined in the standard
- P-frames consist of I-frame macroblocks and predictive macroblocks
- P-frames are quantized and entropy encoded: RLE

MPEG Video Standard

B-Frames (Bi-directionally predictive-coded)

- require information of the previous and following I and/or P-frame
- motion vector from previous and future reference frame
- average of blocks from previous and future frame

MPEG Video Standard

D-Frames (DC-coded Frames)

- D-frames can be used for fast forward, fast rewind mode
- DC-parameters are DCT-coded, AC coefficients are neglected
- D-frames consist of the lowest frequencies of an image
- used only in MPEG-1

MPEG Video Standard

Decoding

- using B-frames, the order of images in an MPEG-encoded data stream differs from the actual decoding order:
- display order
  - type of frame: I B B B P B B B I B B B P
  - frame number: 1 2 3 4 5 6 7 8 9 10 11 12 13
- decoding order
  - type of frame: I P B B B I B B B P B B B
  - frame number: 1 5 2 3 4 9 6 7 8 13 10 11 12

MPEG Video Standard

Quantization

- AC coefficients of B- and P-frames are usually large values, I-frames have smaller values
- ==> MPEG quantization is adjusted
  if data rate increases over threshold => quantization enlarges the step size
  if data rate decreases => quantization is performed with finer granularity

How to Compare?

Part II Compression

- performance tradeoffs
  - encoding/decoding time and complexity
  - image size and frames per second
  - compression factor
  - image quality
- functionality tradeoffs
  - multiple resolutions
  - constant vs. variable bit rate
  - ease of editing

MPEG Performance

- decoding is relatively easy
  - real-time sw decoding possible on current platforms
- encoding is expensive
  - sw encoders on current workstations are still too slow for good quality and high compression factors

MPEG System Standard

- stream specification—interleaving of audio and video, packetizing, timing
- MPEG-1 program streams—variable length packets, 1-2kB up to 64 kB
- MPEG-2 transport streams—fixed length packets, 188bytes/ packet=> 4 ATM packets per transport stream packet
- timing/synchronization specified in streams (27 mHz virtual clock)

MPEG-2 Video Standard Overview

- concept similar to MPEG-1, but includes extensions to cover a wider range of applications
  - all digital transmission of broadcast TV quality video at coded bit rates between 4 and 9 Mbps
  - efficient for HDTV bit rates
- enhancements
  - addition of syntax for efficient coding of interlaced video (e.g., 16x8 block size motion compensation)
  - 10-bit DCT DC coefficients (MPEG-1 8 bits)
  - scalable extensions which permit the division of a continuous video signal into 2 or more coded bit streams representing the video at different resolutions, picture quality or picture rates

MPEG-2 Video Standard Overview

- MPEG-2 addresses applications using CCIR-601 recommendation: more than 15 million samples/sec
- MPEG-2 add to MPEG-1
    - more aspect ratios
    - 4:2:2, 4:4:4 macroblocks
    - progressive and interlaced frame coding
    - additional prediction modes (16x8 MC, field MC)
    - four scalable modes
    - improved picture quality (quantization, zig-zag scan)

MPEG-2 Video Standard

MPEG soon realized:

- no reason to restrict maximum coded bit rate to 10 Mbits/s; MPEG could support higher bit rates (80 to 100 Mbit/s)
- impossible to define a single standard satisfying all requirements
- most applications would only use a small subset of features offered. Hence, MPEG decided to adopt a toolkit-like approach: MPEG-2 is a collection of tools defined in such a way as to satisfy the requirements of specific major applications.

Scalable Bit Streams

four scalable modes—break MPEG-2 video into different layers mostly for purpose of prioritizing video data:

- spatial scalability—codes a base layer at lower sampling dimensions (resolutions) than the upper layers
- data partitioning—breaks the block of 64 quantized coefficients into 2 bit streams. The first, higher priority bit stream contains the more critical lower frequency coefficients and side information (DC values, motion vectors)

Scalable Bit Streams

- SNR scalability—channels are coded at identical sample rates, but with different picture quality (through quantization)
- temporal scalability—higher priority bit stream codes video at a lower frame rate, intermediate frames can be coded in a second bit stream using the first bit stream reconstruction as prediction

Profiles and Levels

- the range of coding support is divided into *profiles* and *levels*. For each profile/level, MPEG-2 provides the syntax for the coded bit stream and the decoding requirements.
- A profile is a defined subset of the entire bit stream syntax. Within a profile, a level is defined as a set of constraints imposed on the parameters of the bit stream (e.g., resolution, max. bit rate)
- profiles: simple, main, 4:2:2, SNR, spatial, high, multiview
- levels: low (SIF), main (CCIR 601), high-1440, high (HDTV)

Profiles and Levels, Nonscalable Modes

Levels		Profiles (nonscalable)		
		simple 4:2:0 (I, P)	**main 4:2:0 (I, B, P)**	**4:2:2 (I), (I, B, P)**
High	res. / rate	N/A	1920 x 1152 / 60	N/A
	Mbits/s	N/A	80	N/A
High-1440	res. / rate	N/A	1140 x 1152 / 60	N/A
	Mbits/s	N/A	60	N/A
Main	res. / rate	720 x 576 / 30	720 x 576 / 30	720 x 608 / 30
	Mbits/s	15	15	50
Low	res. / rate	N/A	352 x 288 / 30	N/A
	Mbits/s	N/A	4	N/A

Profiles and Levels

- tables show upper bounds for picture resolution, frame rate and bit rates. Bitrate data refer to the max. compressed bit rate supported by the input buffers of a decoder. Areas with N/A indicate no conformance restrictions for these variables.
- early implementations only support the main profile at the main level. The simple profile is a low-cost version of the main profile (without bidirectional prediction). Levels are related to resolution: main level (CCIR 601), high-1440 and high level (HDTV), low level (SIF).

Profile and Levels

- profiles and levels have a hierarchical relationship. The syntax supported by a higher profile/level includes all the syntactical elements of lower profile/levels. Decoders for a specific profile/level should be able to decode also bit streams of lower profiles/levels.
- Exception:
  Decoders of simple profile at main level are also required to decode main profile at low level bit streams (MPEG-1).

MPEG-2 4:2:2 Profile

Features:

- DC precision of intracoded blocks can be 8, 9, 10 or 11 bits.
- Number of bits in a macroblock is unconstrained.
- Excellent multigeneration performance (up to 8 generations).
- GOP selection is flexible and permits all encoded pictures to be I-pictures.
- For the same bit rate across several generations, MPEG 4:2:2 provides better image quality than Motion-JPEG.

Profiles and Levels, Scalable Modes

Levels		Profiles (scalable)			
		SNR 4:2:0	spatial 4:2:0	high 4:2:0, 4:2:2	multiview 4:2:2
High	enhancem. (auxiliary)	N/A	N/A	1920 x 1152 / 60	1920 x 1152 / 60
	lower (base)	N/A	N/A	960 x 576 / 30	1920 x 1152 / 60
	Mbits/s	N/A	N/A	100 (all layers) 80 (base + mid) 25 (base layer)	130 (all layers) 50 (auxiliary) 80 (base layer)
High-1440	enhancem. (auxiliary)	N/A	1440 x 1152 / 60	1440 x 1152 / 60	1920 x 1152 / 60
	lower (base)	N/A	720 x 576 / 30	720 x 576 / 30	1920 x 1152 / 60
	Mbits/s	N/A	60 (all layers) 40 (base + mid) 15 (base layer)	80 (all layers) 60 (base + mid) 20 (base layer)	100 (all layers) 40 (auxiliary) 60 (base layer)

Profiles and Levels, Scalable Modes

Levels		Profiles (scalable)			
		SNR 4:2:0	**spatial 4:2:0**	**high 4:2:0, 4:2:2**	**multiview 4:2:2**
Main	enhancem. (auxiliary)	720 x 576 / 30	N/A	720 x 576 / 30	720 x 576 / 30
	lower (base)	-	N/A	352 x 288 / 30	720 x 576 / 30
	Mbits/s	15 (all layers) 10 (base layer)	N/A	20 (all layers) 15 (base + mid) 4 (base layer)	25 (all layers) 10 (auxiliary) 15 (base layer)
Low	enhancem. (auxiliary)	352 x 288 / 30	N/A	N/A	352 x 288 / 30
	lower (base)	-	N/A	N/A	352 x 288 / 30
	Mbits/s	4 (all layers) 3 (base layer)	N/A	N/A	8 (all layers) 4 (auxiliary) 4 (base layer)

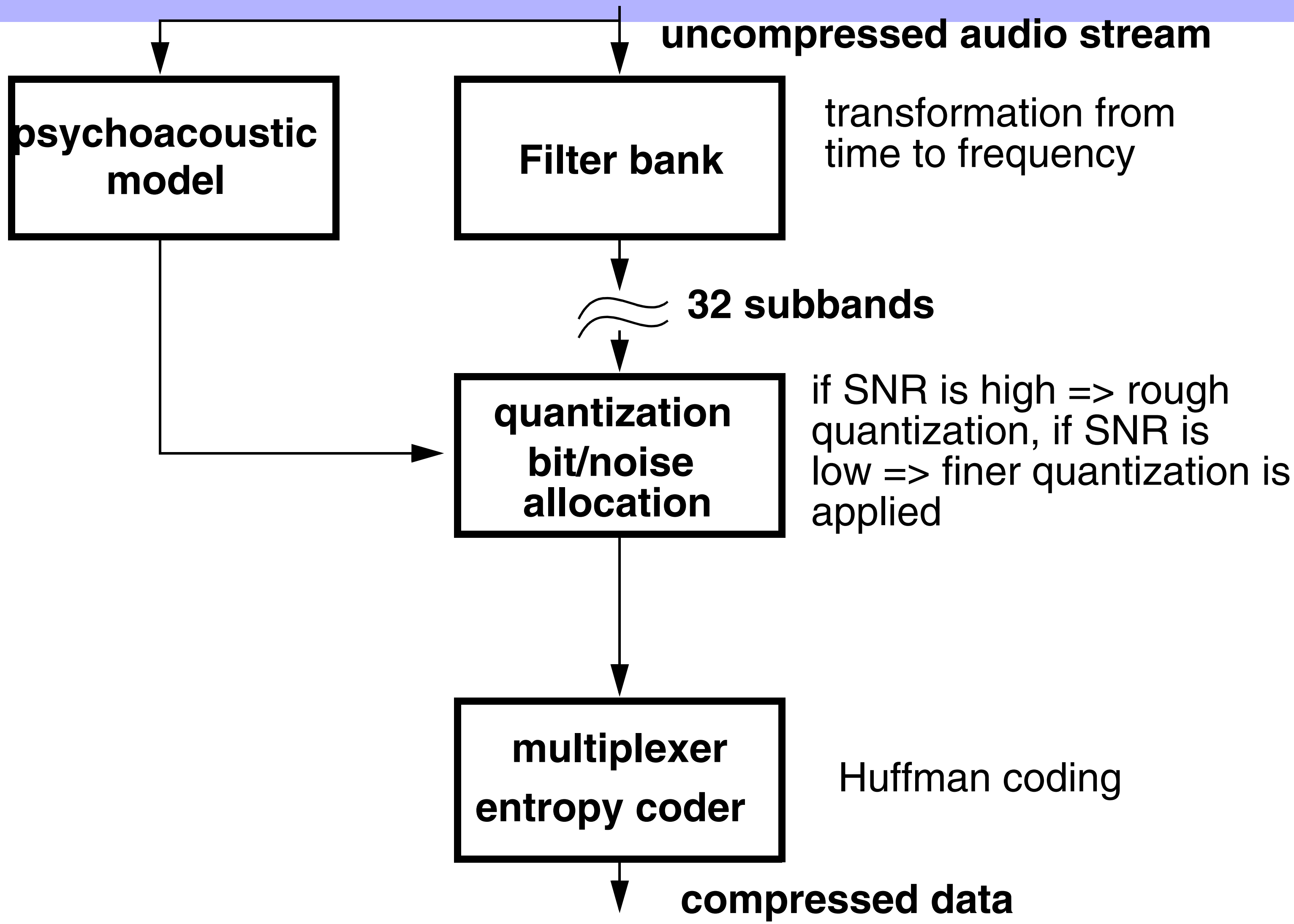
MPEG Audio

- ❑ exponential growth of MPEG coded audio material on the internet since 1995
- ❑ “mp3” the most searched for term in early 1999
- ❑ everybody is using MP3, not many know details of MPEG audio coding
- ❑ *MPEG-1/2 Layer-3 is called MP3*

Why MPEG-1/2 Layer-3?

- open standard
  - specification available for a fee to everybody
  - patents licensed on fair and reasonable terms
  - no single company "owns" the standard
  - public example source code available to implementers
  - format well defined
- availability of encoders and decoders
- supporting technologies—widespread use of sound cards; computers fast enough for audio decoding and even encoding, fast internet access; the spread of CD-ROM and DVD writers

MPEG Audio Standard (Encoding)



High Quality Audio Coding

- MPEG-1 Layer-3 has been defined in 1991
- since then, research on perceptual audio coding has progressed and codes with better compression efficiency became available: MPEG-2 Advanced Audio Coding (AAC) and other proprietary compression systems.
- basic task is to compress audio data in a way that
  - the compression is as efficient as possible (file size small)
  - the reconstructed (decoded) audios sounds exactly (or as close as possible) to the original audio data
  - requires low complexity (sw or inexpensive hw)
  - offers flexibility for different application scenarios.

A Basic Perceptual Audio Encoder

consists of the following building blocks:

- filter bank—is used to decompose the input signal into subsampled spectral components (time/frequency domain)
- perceptual model—using either the domain input signal and/or the output of the analysis filter bank, an estimate of the actual masking threshold is computed by rules from psychoacoustics
- quantization and coding—the spectral components are quantized and coded; noise introduced should be kept below threshold
- encoding of bitstream

A Basic Perceptual Audio Coder