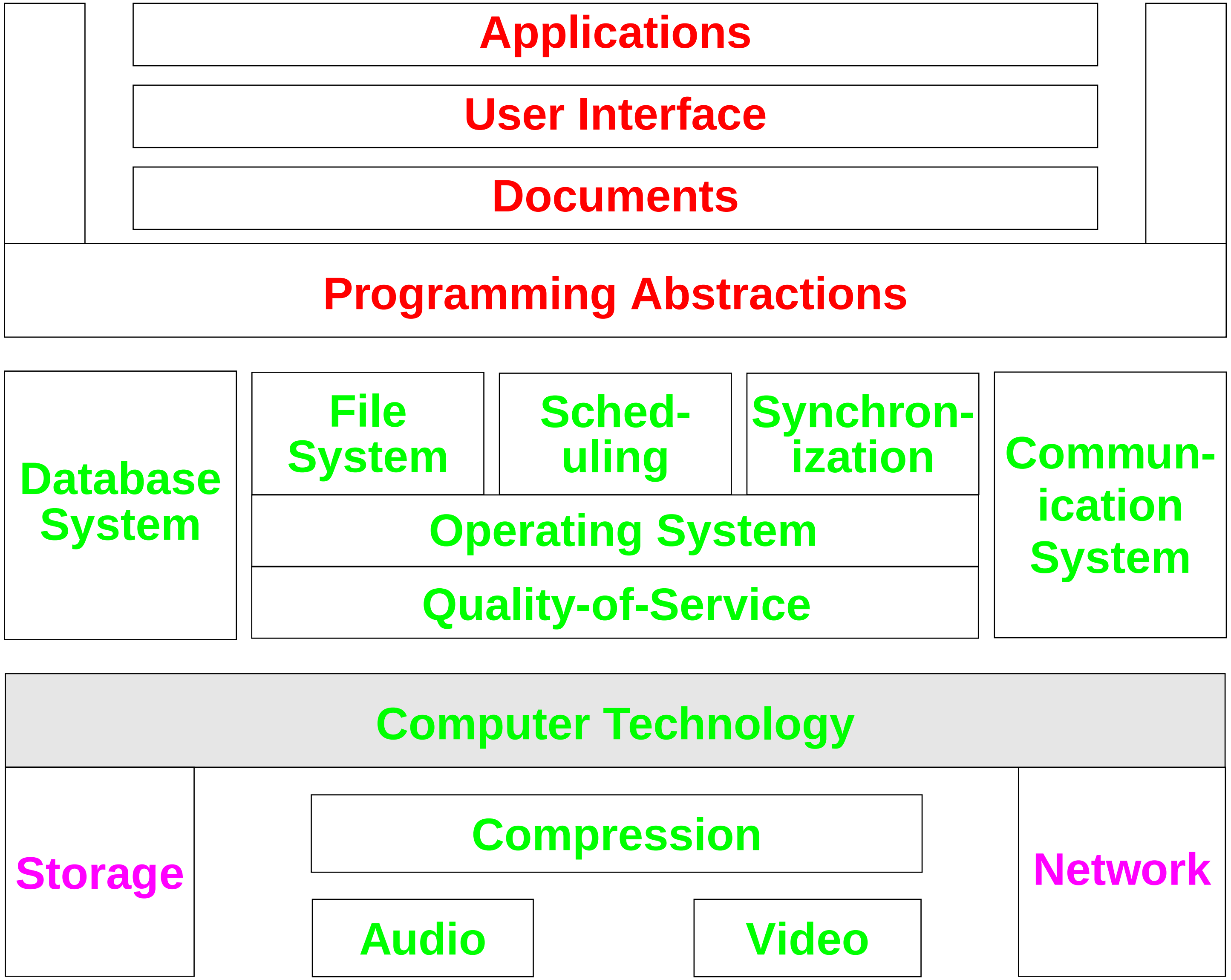
Multimedia

# Multimedia Systems: Compression Techniques

**Ralf Steinmetz**
**Lars Wolf**

**Darmstadt University of Technology**
**Industrial Process and System Communications**

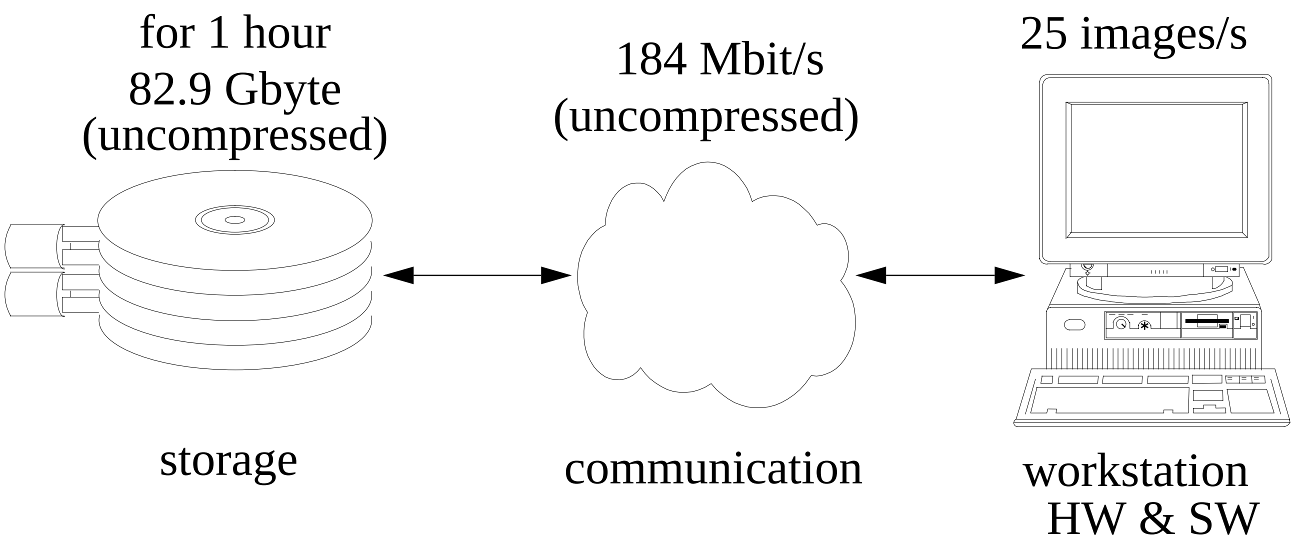
**Merckstraße 25 – 64283 Darmstadt – Germany**


Applications
User Interface
Documents
Programming Abstractions
Database System
File System
Sched-uling
Synchron-ization
Operating System
Quality-of-Service
Commun-ication System
Computer Technology
Storage
Compression
Audio
Video
Network

# CONTENT

# 1. Motivation

**Digital video in computing means:**

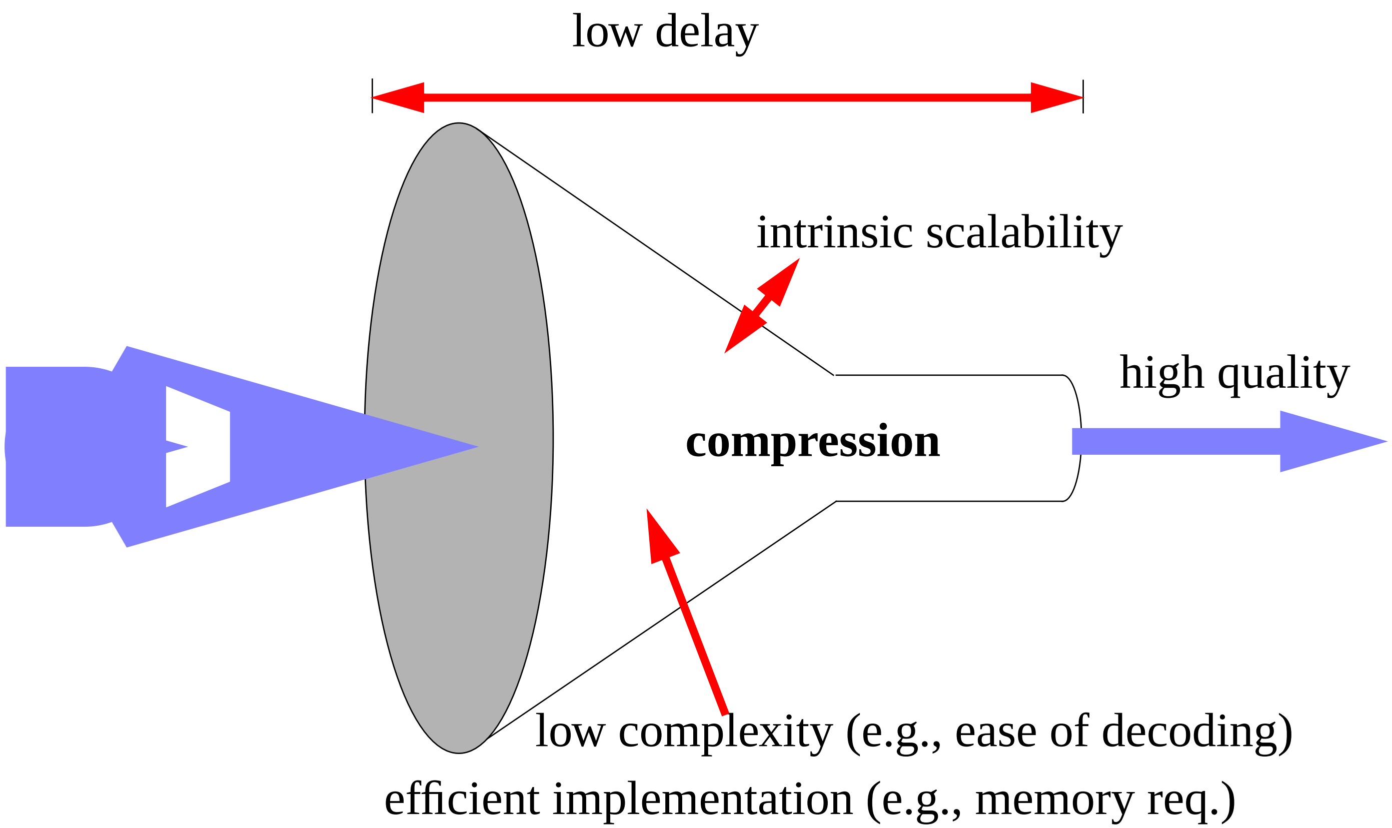



**with 3 bytes/pixel**
**640*480 pixels/frame**
**25 images/s**

**Hence, Compression is *necessary***

# 2. Requirements

## General



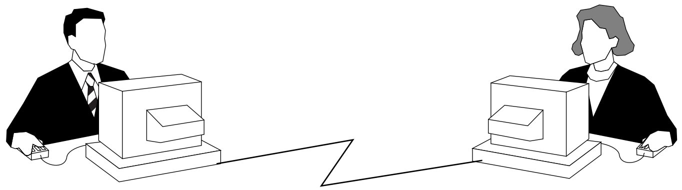
# Requirements

## *Dialogue and retrieval* mode requirements:

- Independence of frame size and video frame rate
- Synchronization of audio, video, and other media

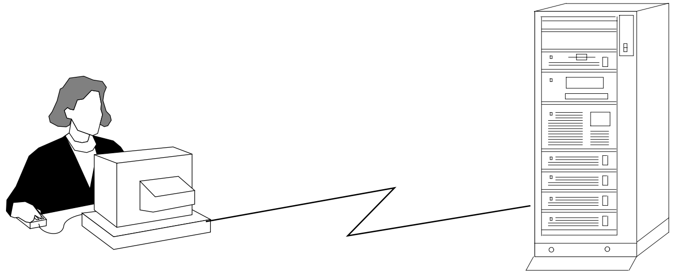
## *Dialogue* mode requirements:

- Compression and decompression in real-time (e.g. 25 frames/s)
- End-to-end delay < 150ms



## *Retrieval* mode requirements:

- Fast forward and backward data retrieval
- Random access within 1/2 s



**Software and/or hardware-assisted implementation requirements**

# 3. Fundamentals

## Categories

**entropy coding**
- ignoring semantics of the data
- lossless

**source coding**
- based on semantic of the data
- often lossy

**channel coding**
- adaptation to communication channel
- introduction of redundancy

**hybrid coding**
- entropy and source coding

# Categories and Techniques

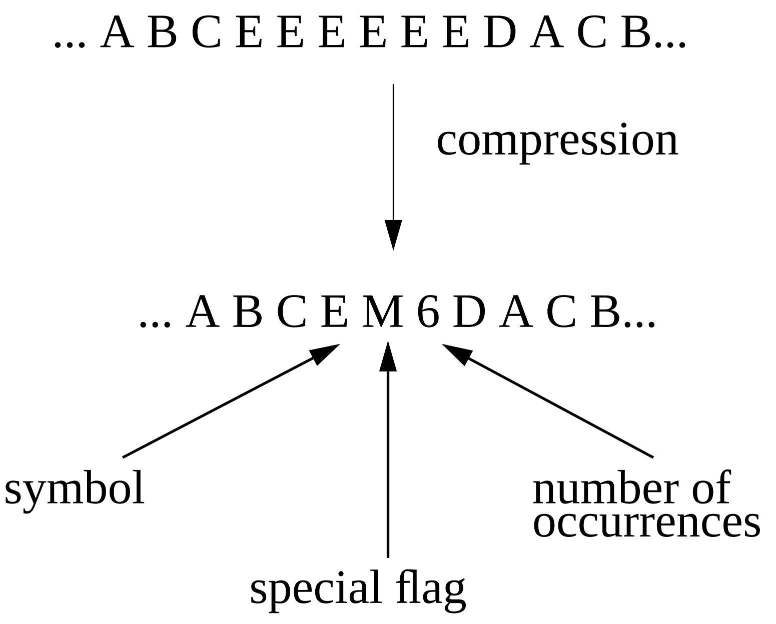
| | | |
|---|---|---|
| Entropy Coding | Run-Length Coding | |
| | Huffman Coding | |
| | Arithmetic Coding | |
| Source Coding | Prediction | DPCM |
| | | DM |
| | Transformation | FFT |
| | | DCT |
| | Layered Coding | Bit Position |
| | | Subsampling |
| | | Sub-Band Coding |
| | Vector Quantization | |
| Hybrid Coding | JPEG | |
| | MPEG | |
| | H.261 | |
| | proprietary: DVI RTV, DVI PLV,... | |

# Entropy Coding: Run-Length

## Assumption:

- Long sequences of identical symbols

## Example:



# Entropy Coding: Huffman

## Assumption:

- Some symbols occur more often than others
- E.g., character frequencies of the English language

## Fundamental principle:

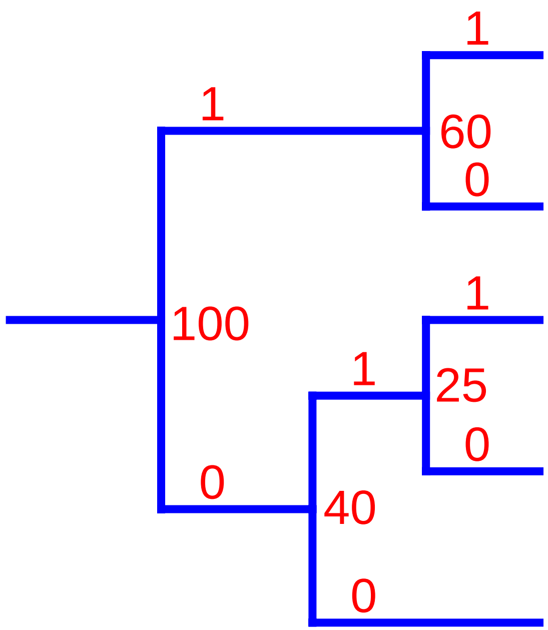
- Frequently occurring symbols are coded with shorter bit strings

# Huffman Coding

**Example:**

- Characters to be encoded: A, B, C, D, E
- Given probabilities of occurrence:
- p(A)=0.3, p(B)=0.3, p(C)=0.1, p(D)=0.15, p(E)=0.15

coding tree

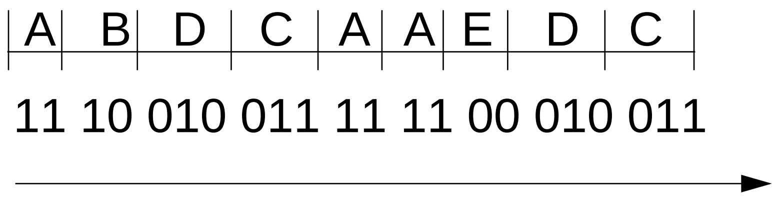



| probability | symbol | code |
|---|---|---|
| 30% | A | 11 |
| 30% | B | 10 |
| 10% | C | 011 |
| 15% | D | 010 |
| 15% | E | 00 |

# Entropy Coding: Huffman

## Table and example of application to data stream

| symbol | code |
|---|---|
| A | 11 |
| B | 10 |
| C | 011 |
| D | 010 |
| E | 00 |

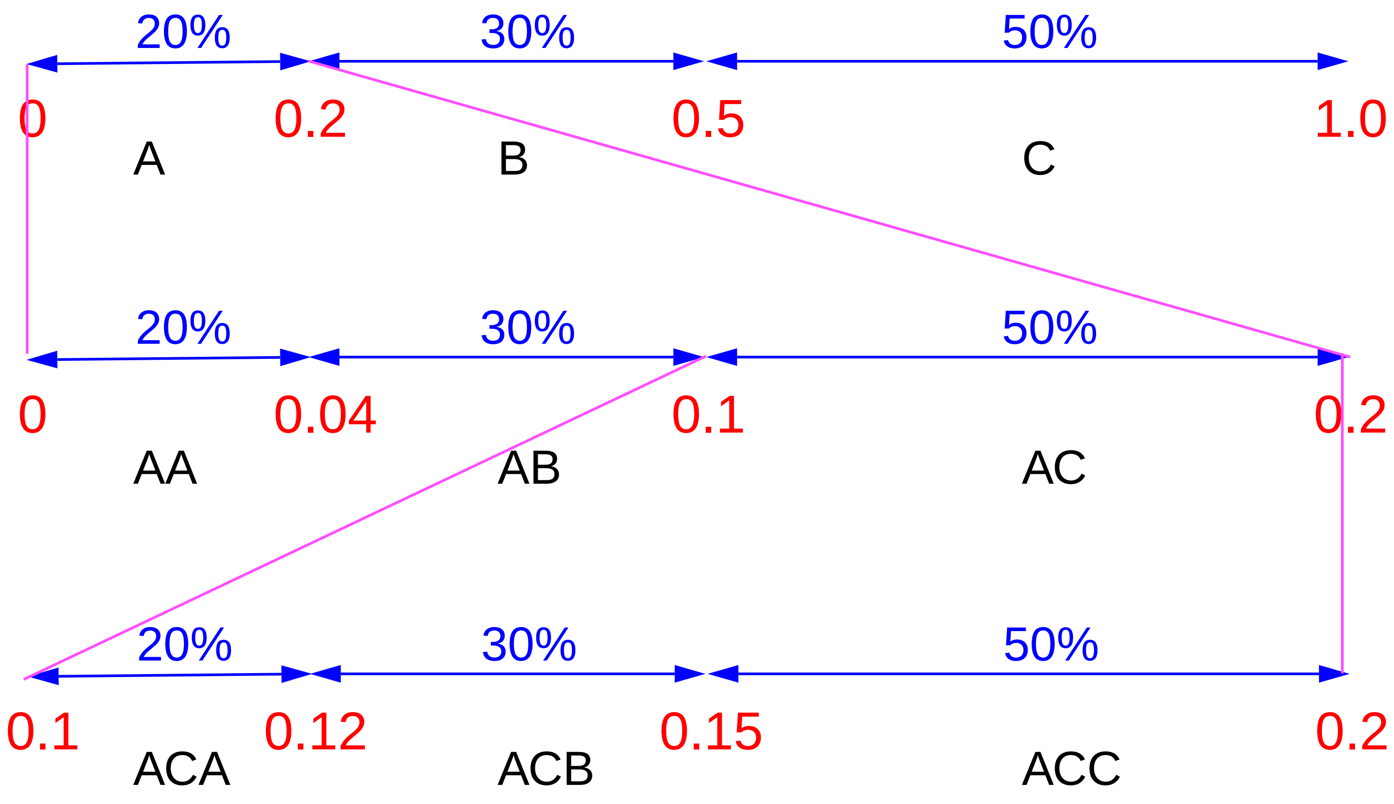


# Arithmetic Encoding (1)

**Example:**

- p(A)=0.2, p(B)=0.3, p(C)=0.5

Code generation



# Arithmetic Encoding (2)

## Table as result

| | lower bound | upper bound | output |
|---|---|---|---|
| A | 0 | 0.2 | - |
| AC | 0.1 | 0.2 | - |
| ACB | 0.12 | 1.15 | 1 |
| A | 0 | 0.2 | - |
| AA | 0 | 0.02 | 0 |

## Note

- some issues are subject to patents

# Source Coding: DPCM

**DPCM = Differential Pulse-Code Modulation**

**Assumptions:**

- Consecutive samples or frames have similar values
- Prediction is possible due to existing correlation

**Fundamental Steps:**

- Incoming sample or frame (pixel or block) is predicted by means of previously processed data
- Difference between incoming data and prediction is determined
- Difference is quantized

**Challenge: optimal predictor**

**Further predictive coding technique:**

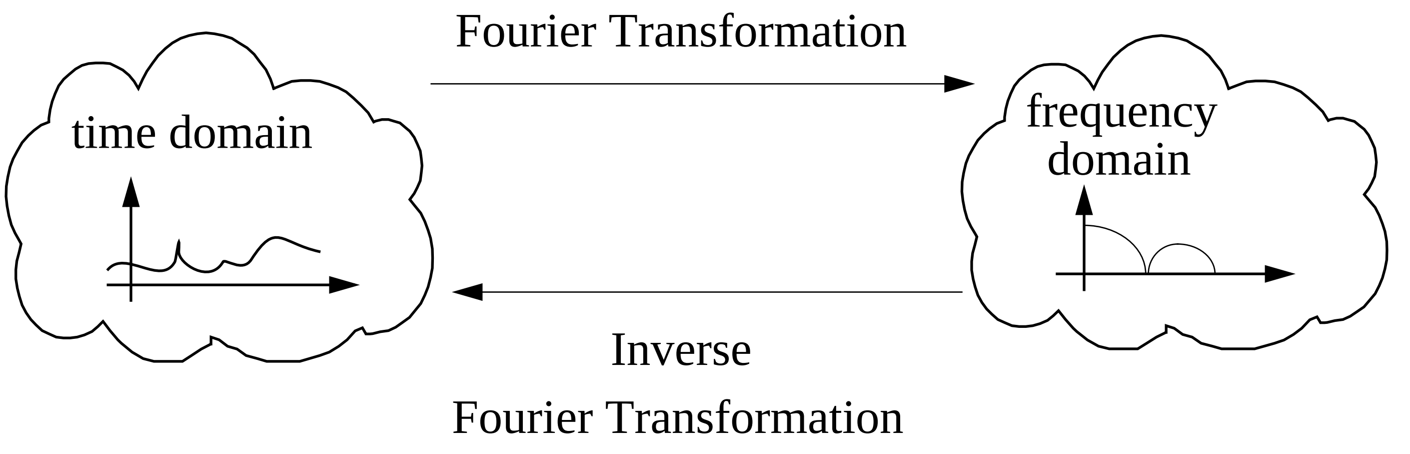
- Delta modulation (DM): 1 bit as difference signal

# Source Coding: Transformation

## Assumptions:

- Data in the transformed domain is easier to compress
- Related processing is feasible

## Example:




## FFT: Fast Fourier Transformation

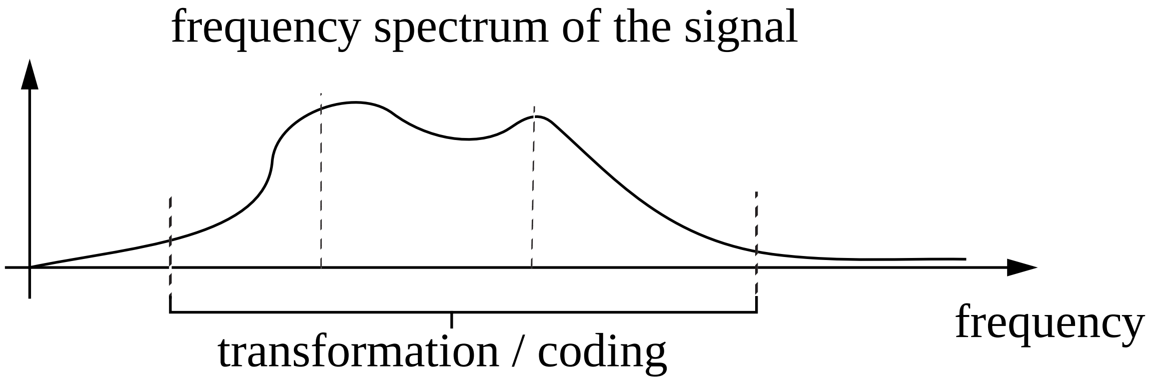
## DCT: Discrete Cosine Transformation

# Source Coding: Sub-Band

## Assumption:

- Some frequency ranges are more important than others

## Example:

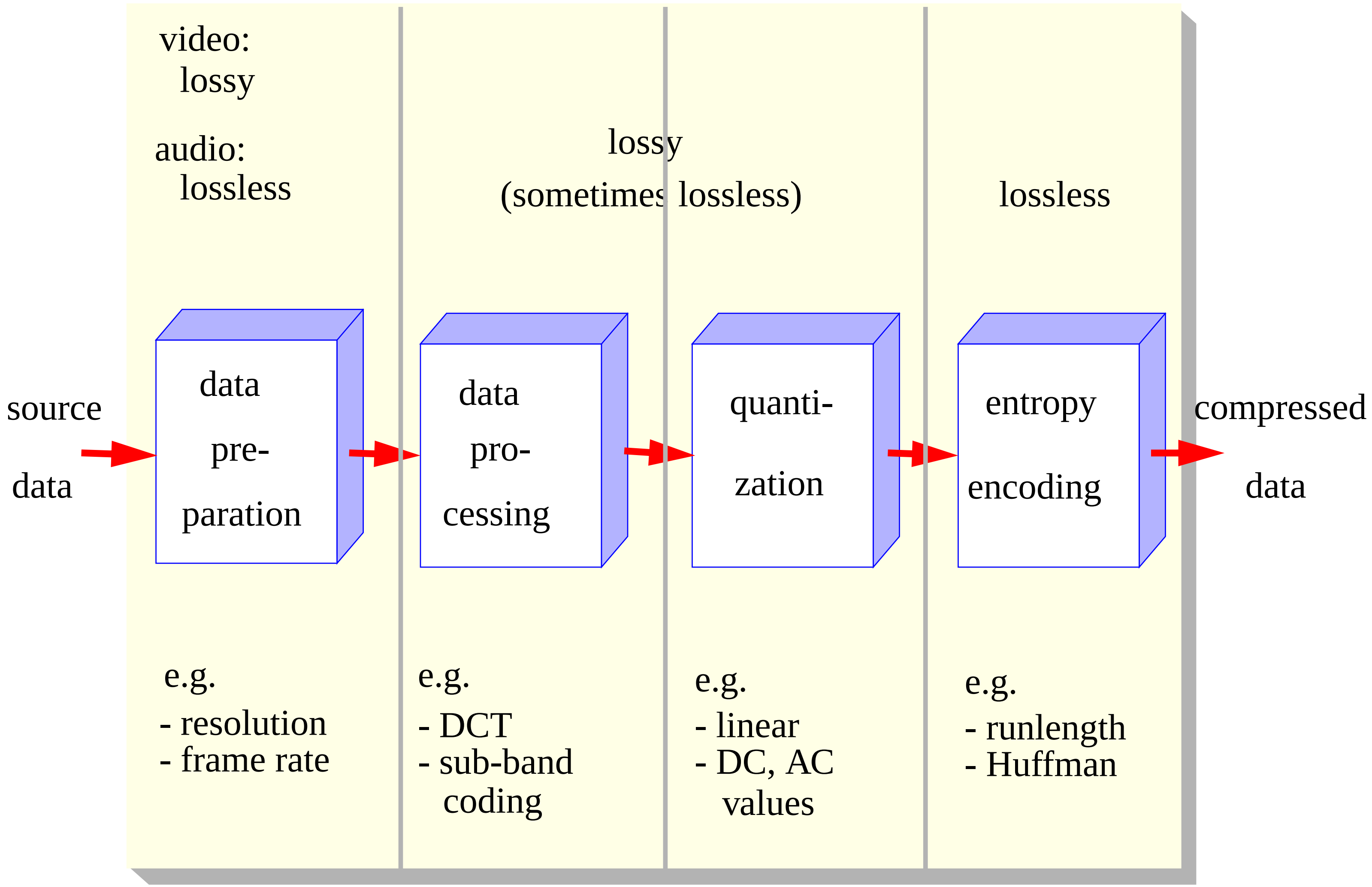



## Application:

- E.g., vocoder for speech communication

Multimedia

# 4. Basic Encoding Steps



# 5. Basic Audio Coding Schemes

## Background

- ITU driven activities

## G.711: PCM

- with 64 kbps

## G.722 differential PCM (DPCM)

- 48, 56, 64 kbps

## G.723

- Multipulse-maximum Likelihood Quatizer (MP-MLQ): 6,3 kbps
- Algebraic Codebook Excitation Linear Prediction (ACELP) 5,3 kbps
- application: speech

# Schemes for Video/Audio Conferencing

## G.728: Low Delay Code Excited Linear Prediction (LD-CELP)

- 16 kbps
- one-way end to end delay less than 2 msec (due to CODEC algorithm)
- complex algorithm
  - 16-18 MIPS in floating point required
  - appr. 40 MIPS whole encoding and decoding

## AV.253

- still “under consideration” at ITU
- 32 kbps

## IS-54

- VSELP
  - good for voice
  - bad for music
- 13 kbps (appr. 8 kbps voice + 5.05 kbps forward error correction FEC)
- driving force: Motorola (similar developments in Japan)

# Schemes for Mobile Telephone Networks

## RPE-LTP (GSM)

- Regular Pulse Excitation - Long-Term Predictor
- used in European GSM: speech
- 13 kbps

## GSM Half-Rate Coders

- 5.6 - 6.25 kbps
- quality and characteristics similar to RPE-LPT

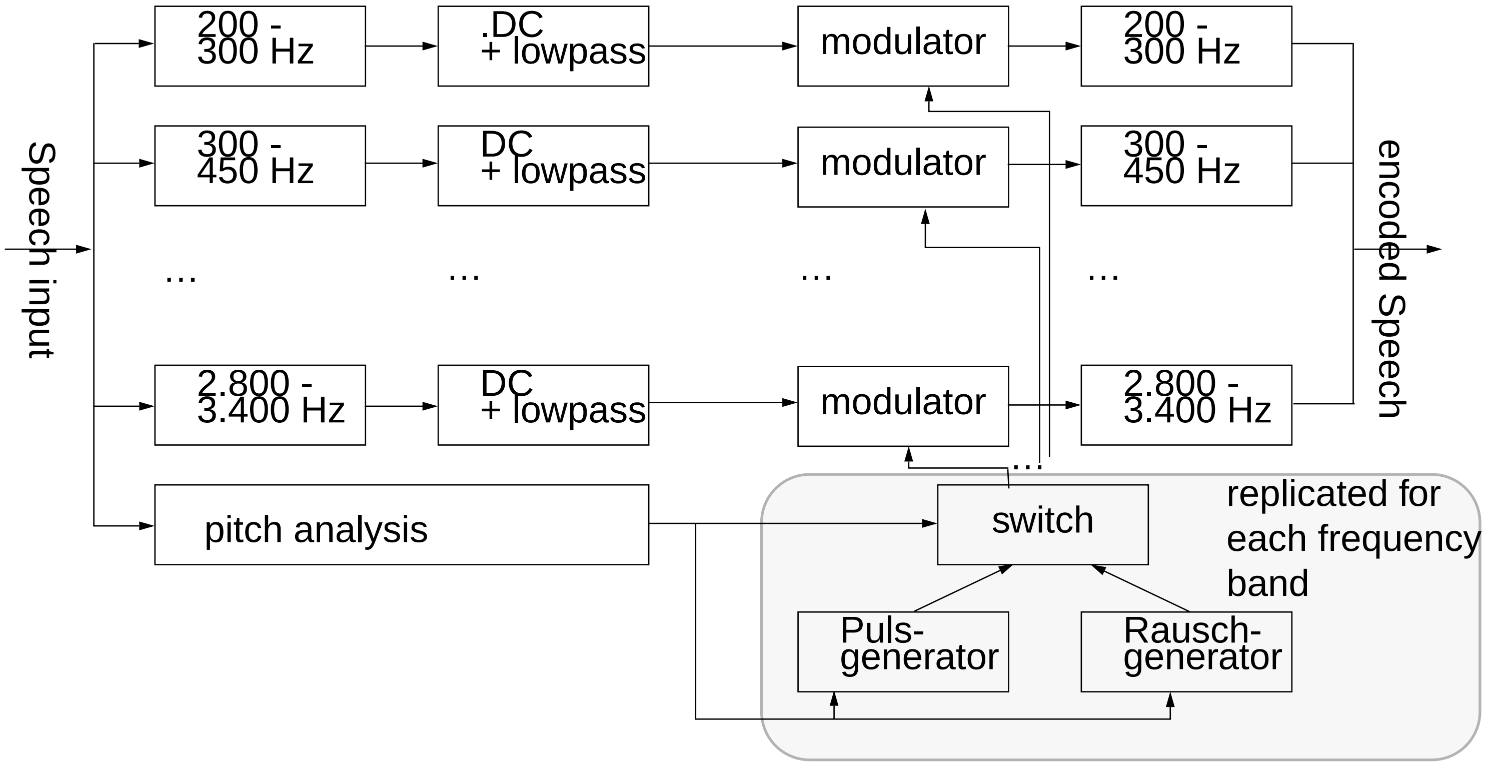
# Vocoder: e.g. Inmarsat IMBE Coder

## Improved Multiband Excitation Coder IMBE

- application: maritime satellite communications
- 4,15 kbps for voice (plus 2,25 kbps for channel coding)

## Principle: Vocoder

- (IMBE voiced and unvoiced individually for each frequency band)



# 6. JPEG

**“JPEG”: Joint Photographic Expert Group**

**International Standard:**

- For digital compression and coding of continuous-tone still images:
  - Gray-scale
  - Color
- Since 1992

**Joint effort of:**

- ISO/IEC JTC1/SC2/WG10
- Commission Q.16 of CCITT SGVIII

**Compression rate of 1:10 yields reasonable results (lt. Heinrichs, Multimedia im Netz)**

# JPEG

## Very general compression scheme

## Independence of:

- Image resolution
- Image and pixel aspect ratio
- Color representation
- Image complexity and statistical characteristics

## Well-defined interchange format of encoded data

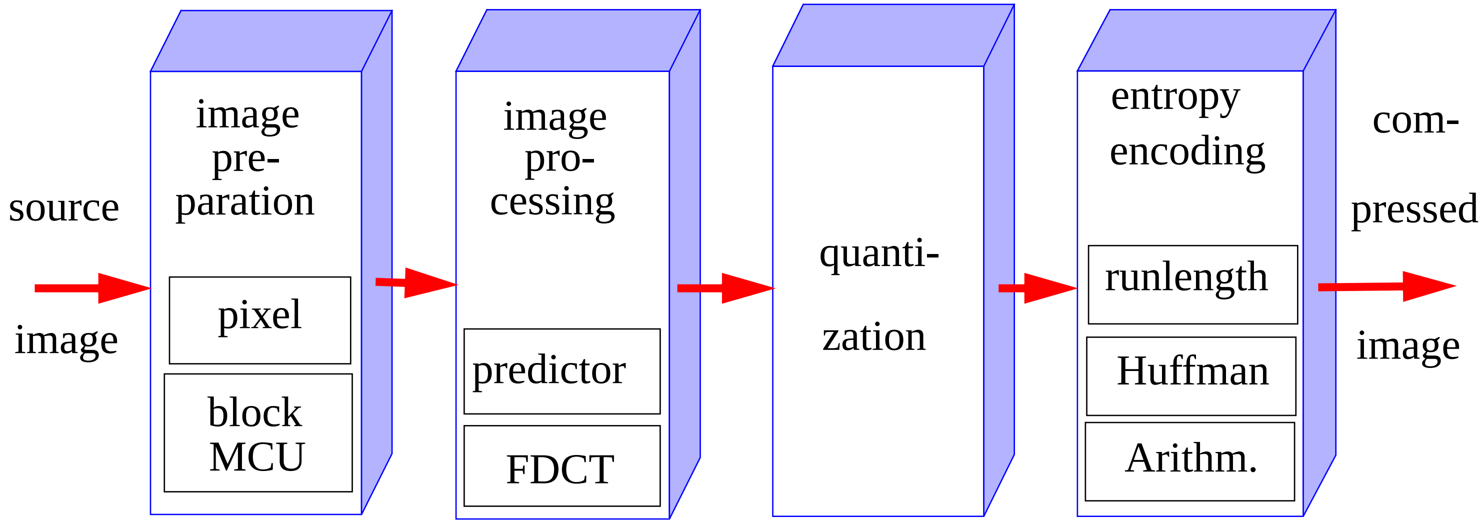
## Implementation in:

- Software only
- Software and hardware

## "MOTION JPEG" for video compression

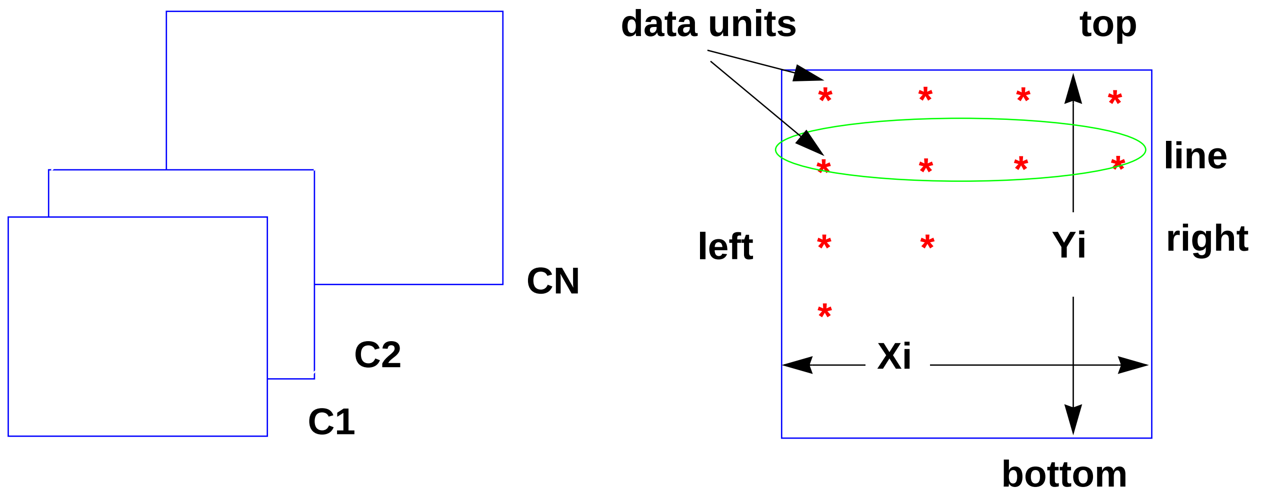
- Sequence of JPEG-encoded images

# JPEG - Compression Steps




**MCU: Minimum Coded Unit**
**FDCT: Forward Discrete Cosine Transformation**

# JPEG - Image Preparation




data units: samples in lossless mode, blocks with 8x8 pixels in other modes

## Planes:

- $1 \leq N \leq 255$ components $C_i$ (e.g., one plane per color)
- Different resolution of individual components possible

## Pixel resolution:

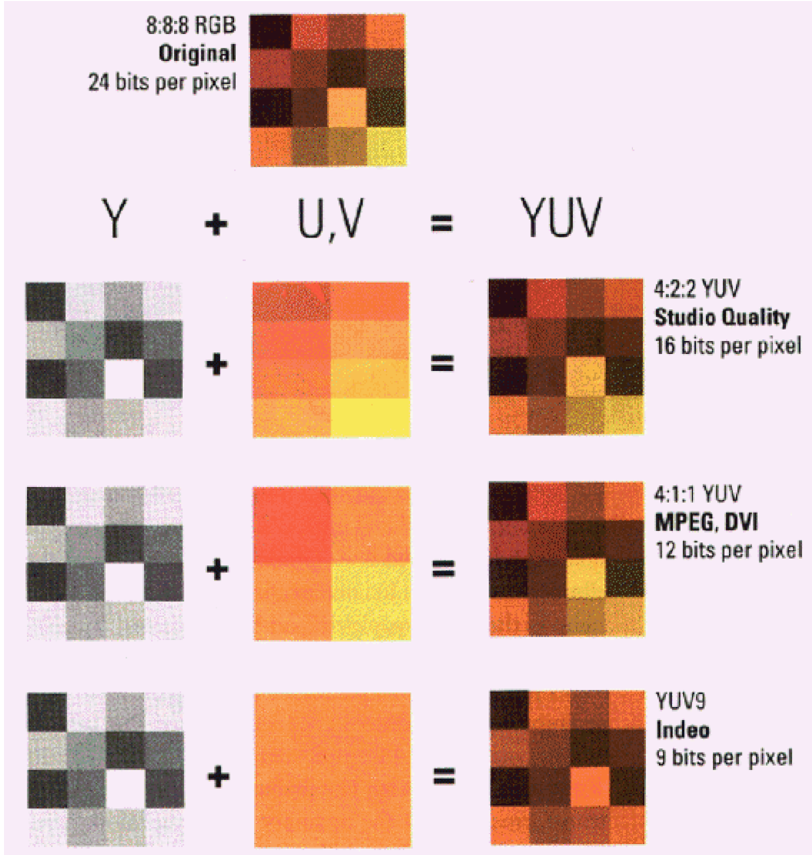
- 8 or 12 bit per pixel in lossy modes
- 2 to 16 bit per pixel in lossless mode

*Multimedia*

# JPEG - Image Preparation

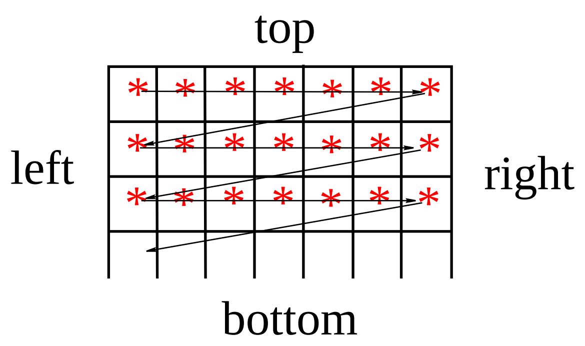
## Example 4:2:2 YUV, 4:1:1 YUV, and YUV9 Coding

- Luminance (Y):
  - brigthness
  - sampling frequency 13.5 MHz
- Chrominance (U, V):
  - color differences
  - sampling frequency 6.75 MHz

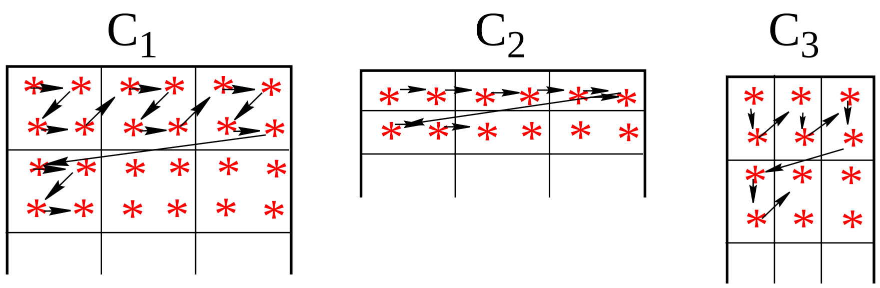


# JPEG - Image Preparation

## Non-interleaved encoding:




## Interleaved encoding:




## Minimum Coded Unit (MCU):

- Combination of interleaved data units of different components

# JPEG - 4 Modes of Compression

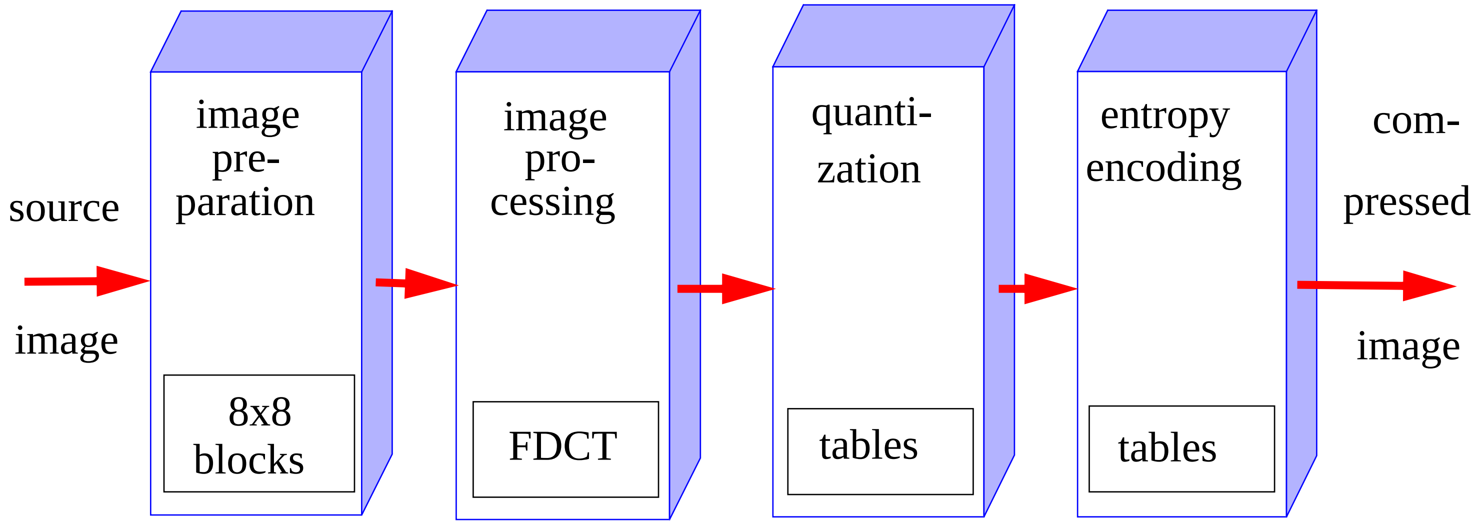
lossy sequential DCT-based mode
(baseline mode)

expanded lossy DCT-based mode

lossless mode

hierarchical mode

# JPEG - Baseline Mode




## Baseline mode is mandatory for all JPEG implementations:

- Often restricted to certain resolution
- Often only three planes with predefined color set-up

## Image preparation:

- Pixel resolution of p=8 bit
- 8 x 8 pixel blocks (data units)

# JPEG - Baseline Mode: Image Processing

***Forward Discrete Cosine Transformation (FDCT):***

$$S_{vu} = \frac{1}{4} C_u C_v \sum_{x=0}^{7} \sum_{y=0}^{7} s_{yx} \cos\frac{(2x+1)u\pi}{16} \cos\frac{(2y+1)v\pi}{16}$$

***with:***

***$c_u, c_v = \frac{1}{\sqrt{2}}$, for u, v= 0; else $c_u, c_v = 1$***

## Formula applied to each block for all $0 \leq u, v \leq 7$:

- Blocks with 8x8 pixel result in 64 DCT coefficients:
  - 1 DC-coefficient $S_{00}$: basic color of the block
  - 63 AC-coefficients: (likely) zero or near-by zero values

## Different significance of the coefficients:

- DC: most important
- AC: less important

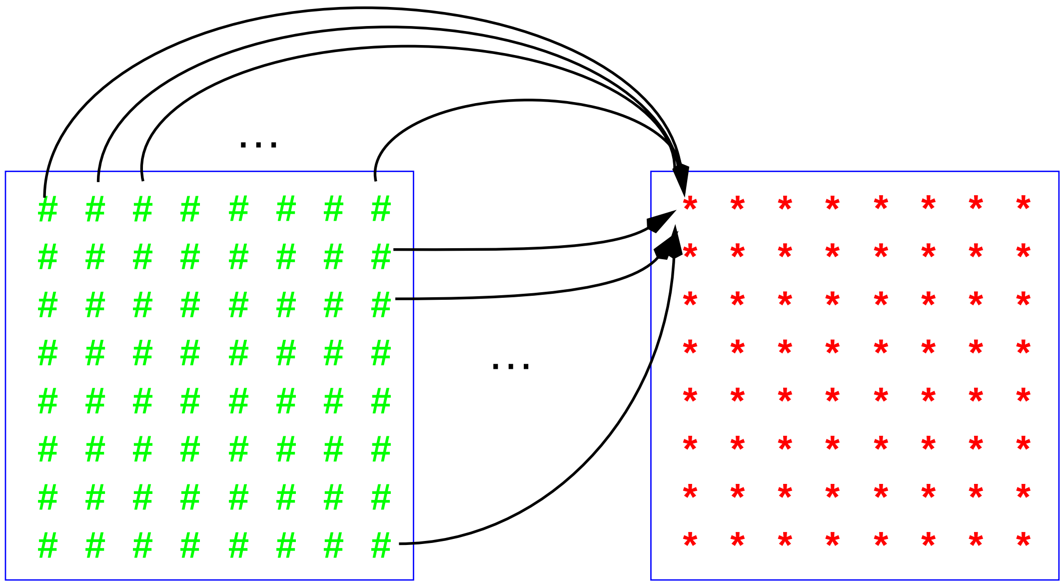
# JPEG – Baseline Mode: Image Processing

## FDCT transforms:

- blocks into blocks
- not pixels into pixels

## Example:

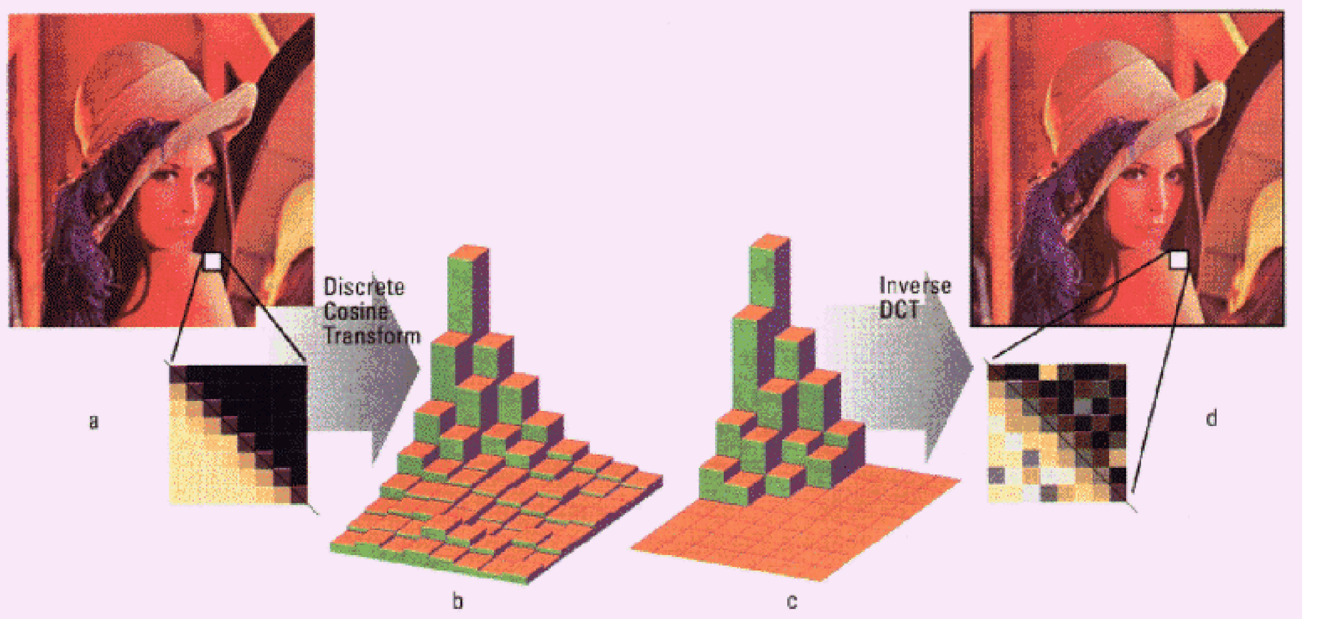
- Calculation of $S_{00}$



# JPEG - Baseline Mode: Quantization

**Use of quantization tables for the DCT-coefficients:**

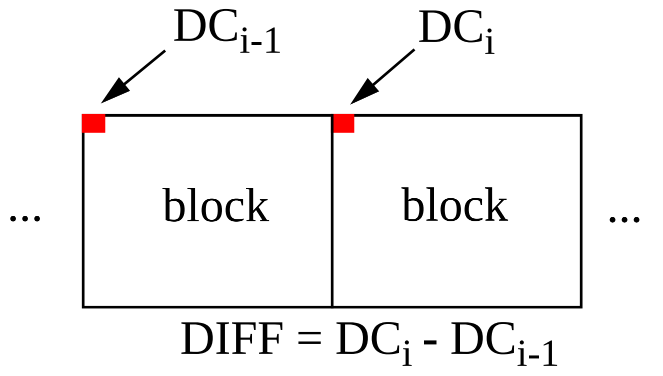
- Map interval of real numbers to one integer number
- Allows to use different granularity for each coefficient



# JPEG - Baseline Mode: Entropy Encoding

## DC-coefficients:

- Compute the differences:

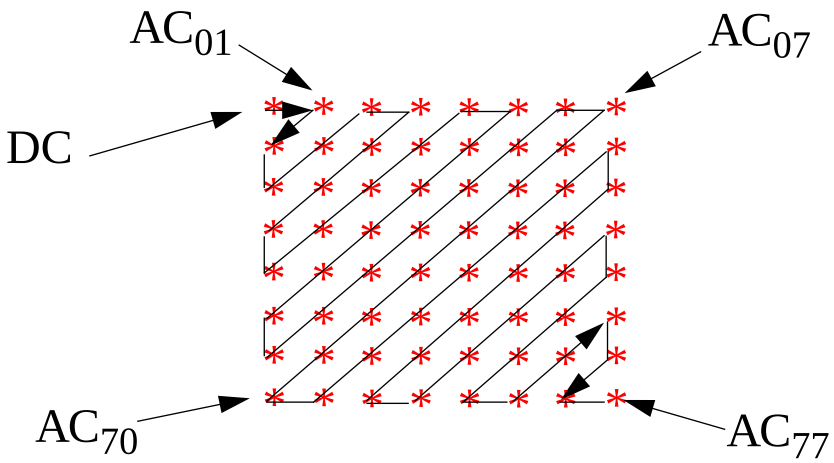



- Use differences instead of the $DC_i$ values

# JPEG - Baseline Mode: Entropy Coding

## 63 AC coefficients:

- Ordering in ‘zig-zag’ form



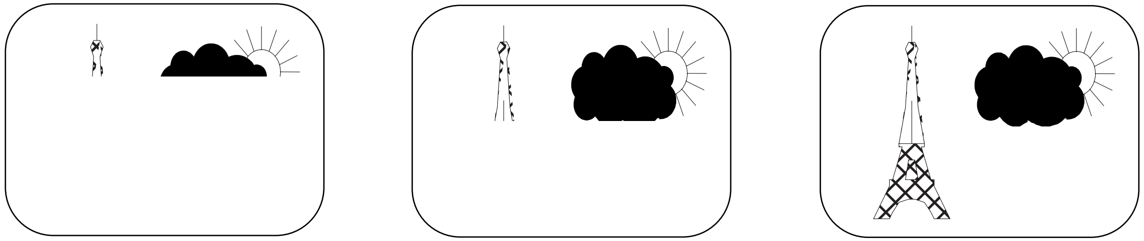

  - reason: coefficients in lower right corner are likely zero
- Huffman coding of all coefficients:
  Transformation into a code where amount of bits depends on frequency of respective value
- Subsequent runlength coding of zeros

# JPEG - Extended Lossy DCT-Based Mode

**Pixel resolution 8 to 12 bit**

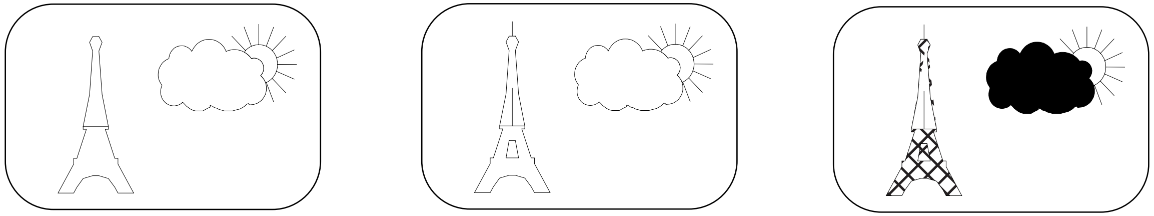
**Sequential image display:**

- Top ➨ bottom
- Good for small images and fast processing



**Progressive image display:**

- Coarse ➨ fine
- Good for large and complicated images



Multimedia

# JPEG - Extended Lossy DCT-Based Mode

**Principle:**

- Coefficients stored in buffer after quantization
- Order of pixel/block processing changed

**By spectral selection:**

- Selection according to importance of DC, AC value
- All DC values of whole image first
- All AC values in order of importance subsequently

**By successive approximation:**

- Selection according to position of bits
- First the most significant bit of all blocks
- Then the second significant bit of all blocks
- Until the least significant bit of all blocks

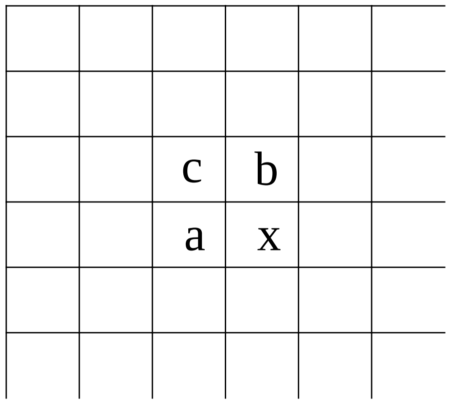
# JPEG - Lossless Mode

## Image preparation:

- On pixel basis (2-16 bit/pixel)

## Image processing:

- Selection of a predictor for each pixel




| code | prediction |
|---|---|
| 0 | no prediction |
| 1 | x=A |
| 2 | x=B |
| 3 | x=C |
| 4 | x=A+B+C |
| 5 | x=A+((B-C)/2) |
| 6 | x=B+((A-C)/2) |
| 7 | x=(A+B)/2 |

## Entropy coding:

- Same as lossy mode
- Code of chosen predictor and its difference to the actual value

# JPEG - Hierarchical Mode

**Coding of each image with several resolutions:**

- Image scaling
- Differential encoding
- First, coded with lowest resolution  image A
- Coded with increasing horizontal & vertical resolution  image A’
- Difference between both images is computed  B = A - A’
- Iteration for higher resolutions

**Features:**

- Requires more storage and higher data rate
- Fast decoding process
- Used for scalable video
- Similar to Photo-CD

# 7. H.261 (px64)

**Video codec for audiovisual services at p x 64kbit/s:**

- CCITT standard from 1990
- For ISDN
- With p=1,..., 30

**Technical issues:**

- Real-time encoding/decoding
- Max. signal delay of 150ms
- Constant data rate
- Implementation in hardware (main goal) and software

# H.261 - Image Preparation

**Fixed source image format**

**Image components:**

- Luminance signal (Y)
- Two color difference signals ($C_b$,$C_r$)

**Subsampling according to CCIR 601 (4:1:1)**

**Quarter Common Intermediate Format (QCIF) resolution:**

- Mandatory
- Y: 176 x 144 pixel
- At 29.97 frames/s appr. 9.115 Mbit/s (uncompressed)

# H.261 - Image Preparation

**Common Intermediate format (CIF) resolution:**

- Optional
- Y: 352 x 288 pixel
- At 29.97 frames/s appr. 36.46 Mbit/s (uncompressed)

**Layered structure:**

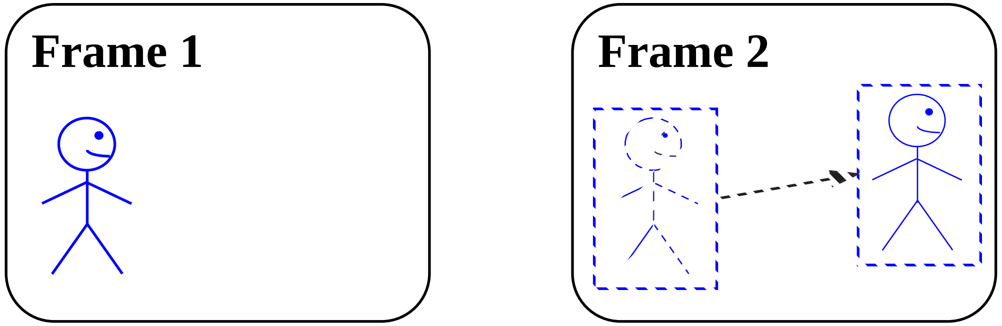
- Block of 8 x 8 pixels
- Macroblock of:
  - 4 Y blocks
  - 1 $C_r$ block
  - 1 $C_b$ block
- Group of blocks (GOBs) of 3 x 11 macroblocks
- Picture:
  - QCIF picture: 3 GOBs
  - CIF picture: 12 GOBs

# H.261 - Image Compression

## Intraframe coding:

- DCT as in JPEG baseline mode

## Interframe coding, motion estimation:




- Search of similar macroblock in previous image
- Position of this macroblock defines motion vector
- Search range is up to the implementation:
  - i.e., motion vector may always be 0

# H.261 - Image Compression

## Interframe coding, further steps:

- Results:
  - Difference between similar macroblocks
  - Motion vector
- Difference of macroblocks:
  - DCT if value higher than a specific threshold
  - No further processing if value less than this threshold
- Motion vector:
  - Components are coded yielding code words of variable length

## Quantization:

- Linear
- Adaptation of step size  constant data rate

# 8. Further ITU Video Schemes (H.263, H.3xx)

## H.263

- extension to H.261
- bitrate: H.263 approx. 2.5 x H.261

## Source Image Formats

<table>
<tr><th rowspan="2">Format</th><th rowspan="2">Pixels</th><th colspan="2">H.261</th><th colspan="2">H.263</th></tr>
<tr><th>Encoder</th><th>Decoder</th><th>Encoder</th><th>Decoder</th></tr>
<tr><td>SQCIF</td><td>128 x 96</td><td colspan="2">optional</td><td colspan="2">required</td></tr>
<tr><td>QCIF</td><td>176 x 144</td><td colspan="2">required</td><td colspan="2">required</td></tr>
<tr><td>CIF</td><td>352 x 144</td><td colspan="2">optional</td><td colspan="2">optional</td></tr>
<tr><td>4CIF</td><td>704 x 576</td><td colspan="2" rowspan="2">not defined</td><td colspan="2" rowspan="2">optional</td></tr>
<tr><td>16CIF</td><td>1408 x 1152</td></tr>
</table>

# H.263

## Differences of H.263 compared to H.261

- optinal PB-frames (2 combined pictures: 1 B- & 1 P-Frame)
- optional overlapped block motion compensation
- optional motion vector pointing outside image
- half pel motion compensation (instead of full pel)
- JPEG is the still picture mode
- no included error detection and correction
- ..

# H.320, H.32x Family

## H.320 specifies (as overview) videophone for ISDN

## H.310

- adapt MPEG 2 for communication over B-ISDN (ATM)

## H.321

- define videoconferencing terminal for B-ISDN (instead of N-ISDN)

## H.322

- adapts H.320 for guaranteed QoS LANs (like ISO-Ethernet)

## H.323

- videoconferencing over non-guaranteed LANs

## H.324

- Terminal for low bit rate communication (over V.34 Modems)

# 9. MPEG-1

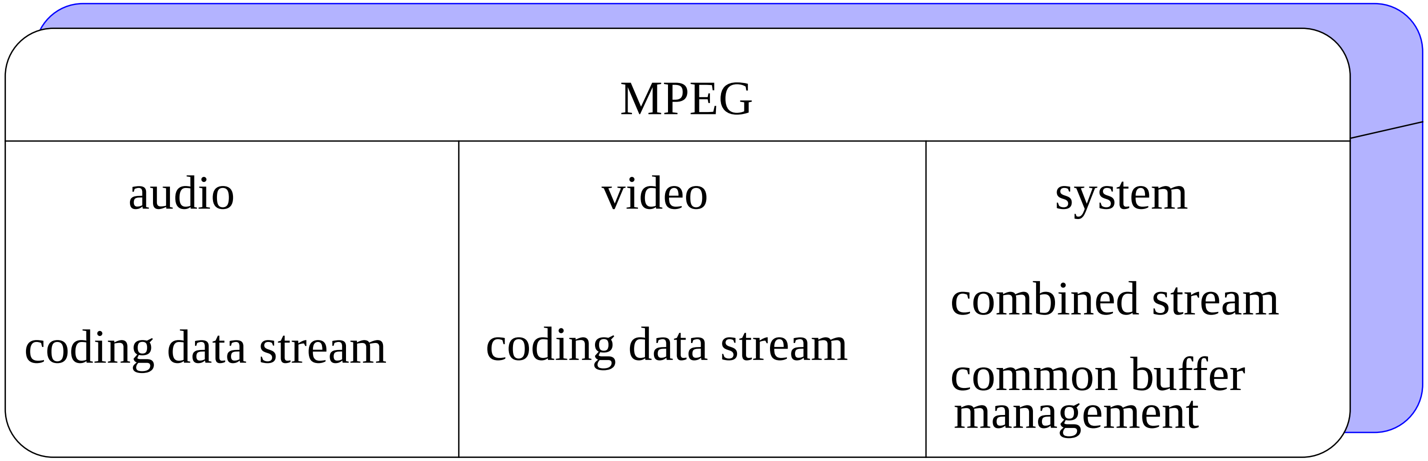
## Motion Picture Expert Group (MPEG):

- ISO/IEC JTC1/SC29/WG11
- ISO IS 11172 since 3/93
- Based on experiences with JPEG and H.261
- Target: coding at about 1.5 Mbit/s for
  - Audio and
  - Video

### Evolution:

- MPEG-1 was starting point
- additional MPEG standards follow
  - MPEG-2:
    - higher data rates for high-quality video
  - MPEG-4:
    - lower data rates for e.g. mobile communication
    - additional functionality
      based on analysis of image contents

# MPEG - Features




**Consideration of other standards:**

- JPEG
- H.261

**Symmetric and asymmetric compression**

**Constant data rate, should be < 1856 kbit/s**

**Target rate about 1.5 Mbit/s**

# MPEG - Video: Preparation Step

## Fixed image format

## Color subsampling:

- Y, $C_r$, $C_b$
- 4:2:0

## Resolution:

- Should be at most 768 x 576 pixel
- 8 bit/pixel in each layer (i.e., for Y, $C_r$, $C_b$)
- 14 pixel aspect ratios
- 8 frame rates

## No user defined MCU like JPEG

## No progressive mode like JPEG

# MPEG - Video: Processing Step

**4 types of frames:**

**I-frames (intra-coded frames):**

- Like JPEG
- Real-time decoding demands

**P-frames (predictive coded frames):**

- Reference to previous I- or P-frames
- Motion vector
  - MPEG does not define how to determine the motion vector
  - difference of similar macroblocks is DCT coded
- DC and AC coefficients are runlength coded

# MPEG - Video: Processing Step

## B-frames (bi-directional predictive coded frames):

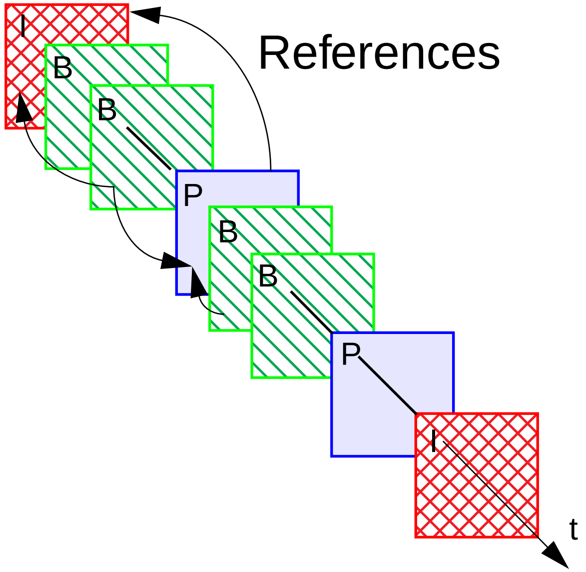
- Reference to previous and subsequent (I or P) frames
- Interpolation between macro blocks

## D-frames (DC-coded frames):

- Only DC-coefficients are DCT coded
- For fast forward and rewind

# MPEG - Video Coding

## Sequence of I-, P-, and B-frames:




- I-Frames (Intracoded)
- P-Frames (Predictive Coded)
- B-Frames (Bidirectionally Coded)
- (D-Frames (DC Coded))

## Sequence:

- Defined by application
- E.g., I B B P B B P B B I B B P B B P B B…
- Order of transmission is different: I P B B ...

# MPEG - Video: Implications

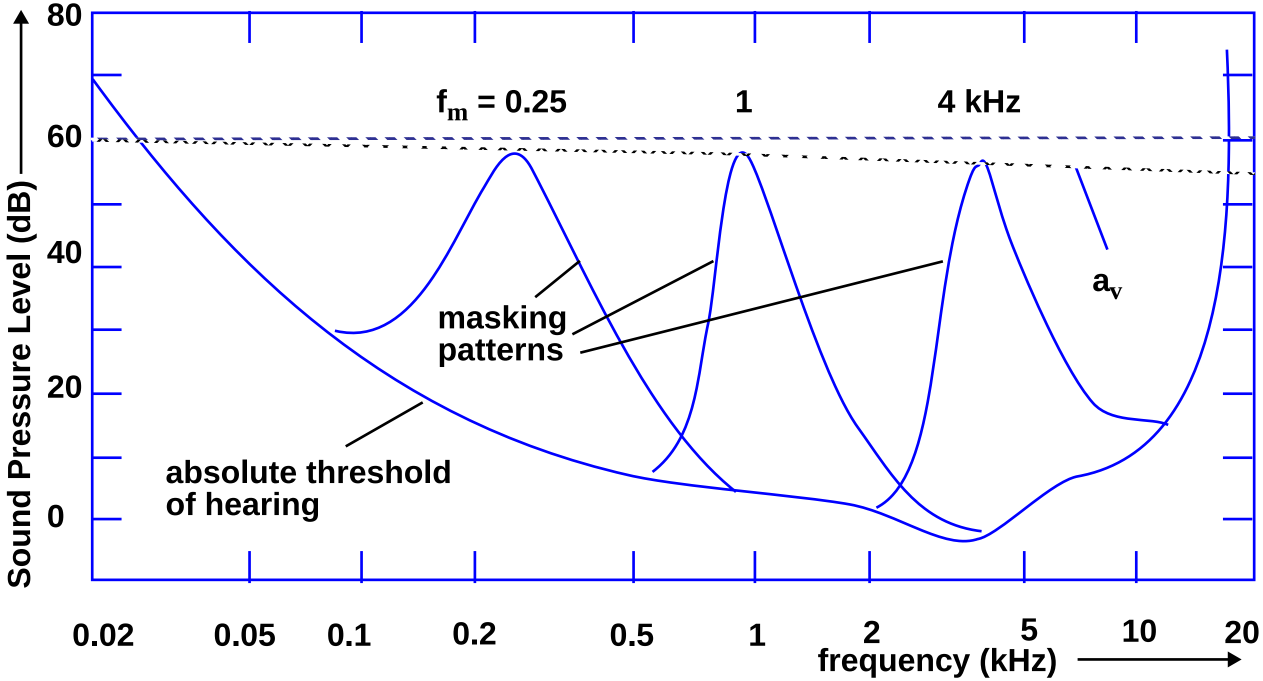
## Random access

- at I-frames
- at P-frames: i.e. decode previous I-frame first
- at B-frame: i.e. decode I and P-frames first

## Editing

- decoded data
  - loss of quality (encode -> decode -> encode -> ...)
  - application of all video editing functions
- encoded data (previous to entropy encoding)
  - preservation of quality
  - transition effects as function in the DCT domain
  - morphing, non-block conform overlay very difficult
- encoded data
  - preservation of quality
  - today: too complex, if possible, i.e. need for entropy decoding

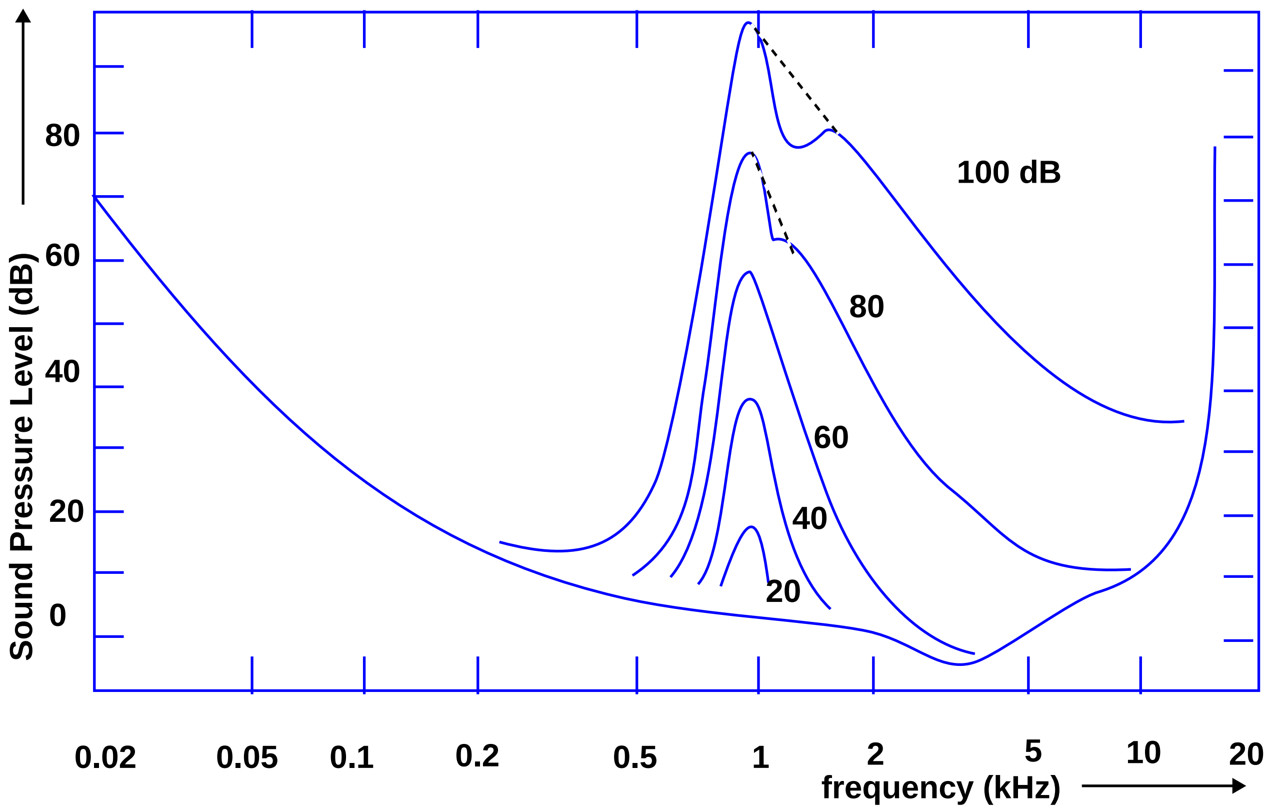
# MPEG - Audio Coding: Fundamentals




## Masking threshold in the frequence domain

- narrowband random noise
- depends on frequency

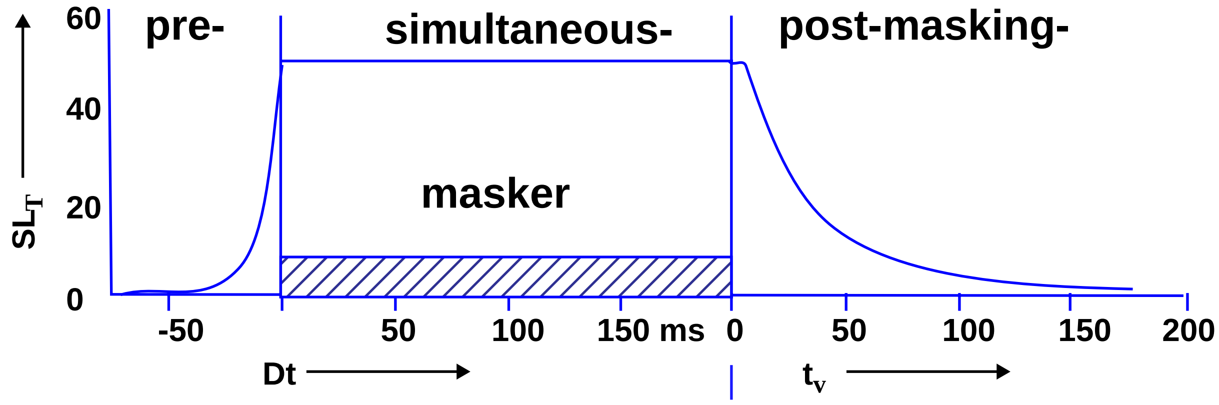
# MPEG - Audio Coding: Fundamentals




## Masking threshold in the frequence domain

- narrowband random noise
- depends on amplitude

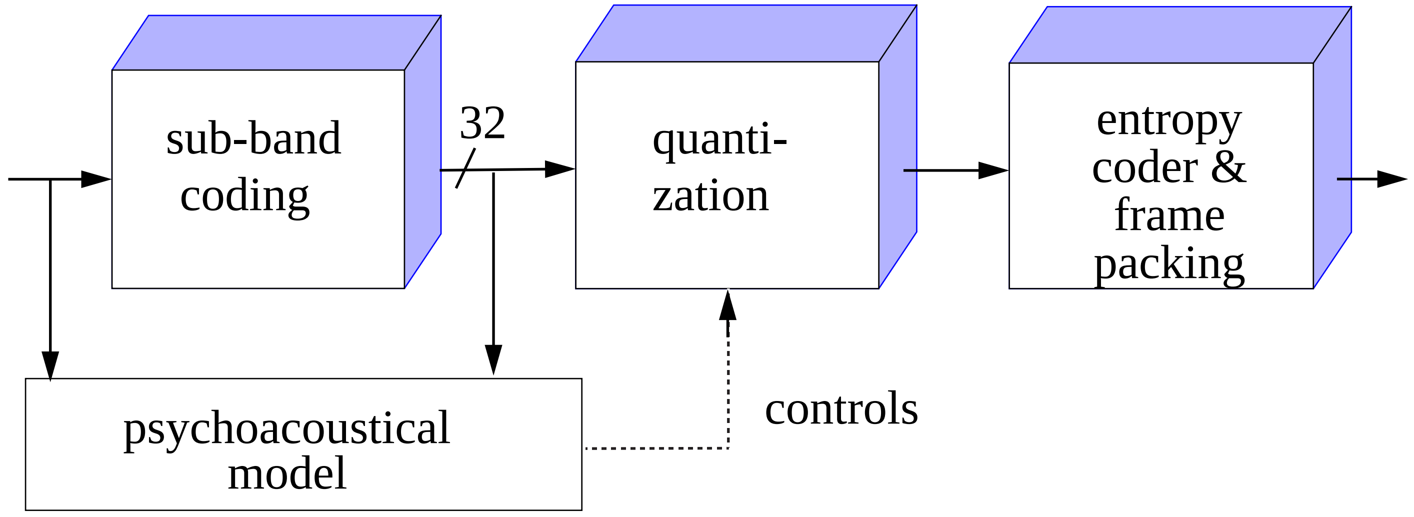
# MPEG - Audio Coding: Fundamentals




## Masking in Time Domain

- after and before the event
- depends on (to some extent) amplitude

# MPEG - Audio Coding




## Audio channel:

- Between 32 and 448 kbit/s
- In steps of 16 kbit/s

## Definition of 3 layers of quality

- Layer 1: max. 448 Kbit/s
- Layer 2: max. 384 Kbit/s (most often used, also as MUSICAM in DAB)
- Layer 3: max. 320 Kbit/s

# MPEG - Audio Coding

## Compatible to encoding of CD-DA and DAT:

- Sampling rates:
  - 32 kHz
  - 44,1 kHz
  - 48 kHz
- Sampling precision:
  - 16 bit/sample

## Audio channels:

- Mono (single, 1 channel)
- Stereo (2 channels)
  - dual channel mode (independent, e.g., bilingual)
  - optional: joint stereo (exploits redundancy and irrelevancy)

# MPEG - Audio Coding: Application

## DAB Digital Audio Broadcasting

- uses MPEG layer 2
- data compression also known as “MUSICAM”
  (Masking pattern adapted Universal Subband Integrated Coding And Multiplexing)

## Delays

- max. of 30 ms encoding
- max. of 10 ms decoding
- based on VLSI

# MPEG - Audio and Video Data Streams

## Audio Data Stream Layers:

1. Frames
2. Audio access units
3. Slots

## Video Data Stream Layers:

1. Video sequence layer
2. Group of pictures layer
3. Single picture layer
4. Slice layer
5. Macroblock layer
6. Block layer

# Follow-Up MPEG Standards

## MPEG-2

- 1993: Committee Draft
- 1994: Draft International Standard
  - MPEG-2 Audio, MPEG-2 Video in March 1994, MPEG-2 System in June 1994
- Late 1994: International Standard
- Commercial MPEG-2 realizations available: digital TV

## MPEG-3

- Initially HDTV
- MPEG-2 scaled up to subsume MPEG-3

## MPEG-4

- Originally targeted at very low bit rates (less than 128 Kbit/s)
- Additional functionality based on analysis of image contents

# 10. MPEG-2

## From MPEG-1 to MPEG-2

- Improvement in quality
  - from VCR to TV to HDTV
- No CD-ROM based constraints
  - higher data rates

## Data rates

- MPEG-1: about 1.5 Mbit/s
- MPEG-2: 2-100 Mbit/s

## Also later known as H.262

# MPEG-2 Video

## Inclusion of interlaced video format

## Increase resolution, more than CCIR 601

## Defined as:

- 5 profiles (simple, main,..)
- 4 levels (with increasing resolution,...)

## Other additional features

- DCT coefficients may be coded with a non-linear quatization function

# MPEG-2 Video: Scaling

## Motivation

- analog: contiuous decrease in quality if errors occur
- digital: need for tolerance whenever error occur, i.e **scaling**

## Option: Spatial scaling

- reduction of resolution
- approach
  - image sampled with half resolution, then MPEG algorithms applied, output processd with better FEC (**base layer**)
  - Image decoded, substracted from original, to difference MPEG algorithms applied, output processd with worseFEC (**enhanced layer**)

## Option: Signal to Noise SNR scaling

- noise introduced by
  - quantization errors and visible block structures
- approach
  - **Base layer**: DCT output, more significant bits encoded with better FEC
  - **Enhanced layer**:DCT output, less significant bits encoded with worse FEC

# MPEG-2 Video Profiles und Levels

| | | | | | |
|---|---|---|---|---|---|
| High Level<br>1920 pixels/line<br>1152 lines | | ≤ 80 Mbit/s | | | ≤ 100 Mbit/s |
| High-1440 Level<br>1440 pixels/line<br>1152 lines | | ≤ 60 Mbit/s | | ≤ 60 Mbit/s | ≤ 80 Mbit/s |
| Main Level<br>720 pixels/line<br>576 lines | ≤ 15 Mbit/s | ≤ 15 Mbit/s | ≤ 15 Mbit/s | | ≤ 20 Mbit/s |
| Low Level<br>352 pixels/line<br>288 lines | | ≤ 4 Mbit/s | ≤ 4 Mbit/s | | |
| LEVELS<br>and<br>PROFILES | Simple<br>Profile | Main<br>Profile | SNR<br>Scalable<br>Profile | Spatial<br>Scalable<br>Profile | High<br>Profile |
| | No B-frames | B-frames | B-frames | B-frames | B-frames |
| | 4:2:0 | 4:2:0 | 4:2:0 | 4:2:0 | 4:2:0 or 4:2:2 |
| | Not Scalable | Not Scalable | SNR<br>Scalable | SNR Scalable<br>or Spatial<br>Scalable | SNR Scalable<br>or Spatial<br>Scalable |

# MPEG-2 Audio

## Up to

- 5 full bandwidth channels (surround system)
  - left and right front
  - center (in frot)
  - left and right back with (x and y = 0.71)

$$\text{Left for Stereo} = \text{Left\_f} + x\text{Center} + y\text{Left\_b}$$

$$\text{Right for Stereo} = \text{Right\_f} + x\text{Center} + y\text{Rigtht\_b}$$

- and 7 multilingual/commentary channels

## Improved quality at or below 64 kbit/s

## Compatible to MPEG-1

- all MPEG-1 audio format can be processed by MPEG-2
- Only 3 MPEG-2 audio codec will not provide backward compatibility (in the range between 256 - 448 Kbit/s)

# MPEG-2 System

## Steps

1. audio and video combined to “Packetized Elementary Stream (PES)“
2. PES(es) combined to “Program Stream” or “Transport Stream”

## Program stream:

- Error-free environment
- Packets of variable length
- One single stream with one timing reference

## Transport stream:

- Designed for “noisy“ (lossy) media channels
- Multiplex of various programs with one or more time bases
- Packets of 188 byte length

## Conversion between Program and Transport Streams possible

# 11. MPEG-4

## Goals

**MPEG-4 (ISO 14496) Originally:**

- Targeted at systems with very scarce resources
- To support applications like
  - Mobile communication
  - Videophone and E-mail
- Max. data rates and dimensions (roughly):
  - Between 4800 and 64000 bits/s
  - 176 columns x 144 lines x 10 frames/s

**Further demand:**

- To provide enhanced functionality
  to allow for analysis and manipulation of image contents

# MPEG-4: Components

## Concept

- API for coded representation of objects
- not algorithm itself

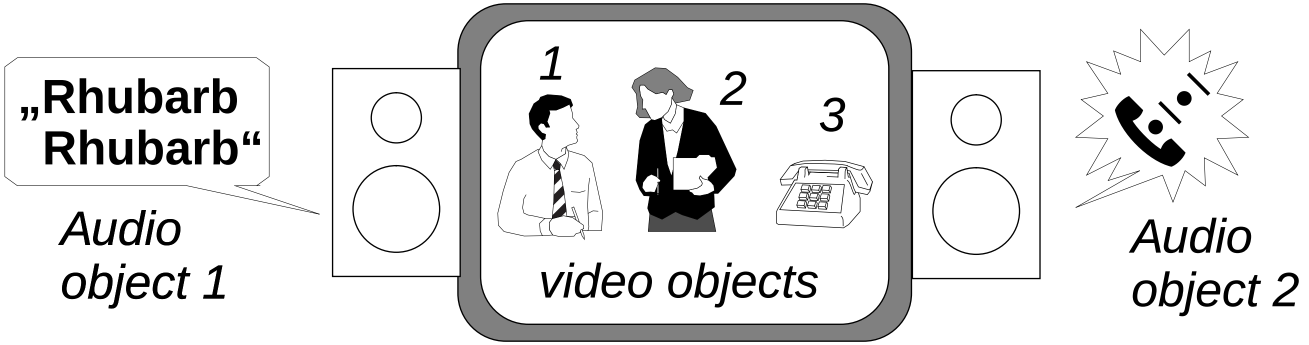
## Components

- definition of the object interface (as kind of API)
- mechanism to combine objects
  - to construct the compression algorithms & profiles
- specification how to download new objects
- rules (syntax) how to parse all mentioned above

# MPEG-4: Basic Idea

## Code and manipulate individual objects:

- Video objects
- Audio objects




## Include:

- Natural objects
- Synthectic objects

## Coding steps:

- Analyse image contents => identify objects
- Code individual objects independently of each others

# MPEG-4: Elements of Standard

## Element 1: Set of coding tools

- To support
  - efficient compression
  - object-based operations
  - scalability
  - error robustness
- User can select appropriate tools from toolset
- Toolset is extendable

## Element 2: Syntax for descriptions

- MSDL: MPEG-4 Syntactic Description Language
- To describe individual objects
- and operations on objects:
  - decoding
  - manipulation
  - combination

# MPEG-4: Standardization

**Standardization is currently under way:**

- 1993: Work on MPEG-4 started
- 1998 (?): International MPEG-4 standard finished

**Parts of coding toolset will be drawn from existing standards:**

- MPEG-1, MPEG-2, H.261, H.263

# 12. Wavelets

## Motivation

**DCT problems:**

- at high compression ratio ⇒ block structure becomes visible
- scaling as add-on ⇒ additional effort
- DCT function is fixed ⇒ can not be adapted to source data

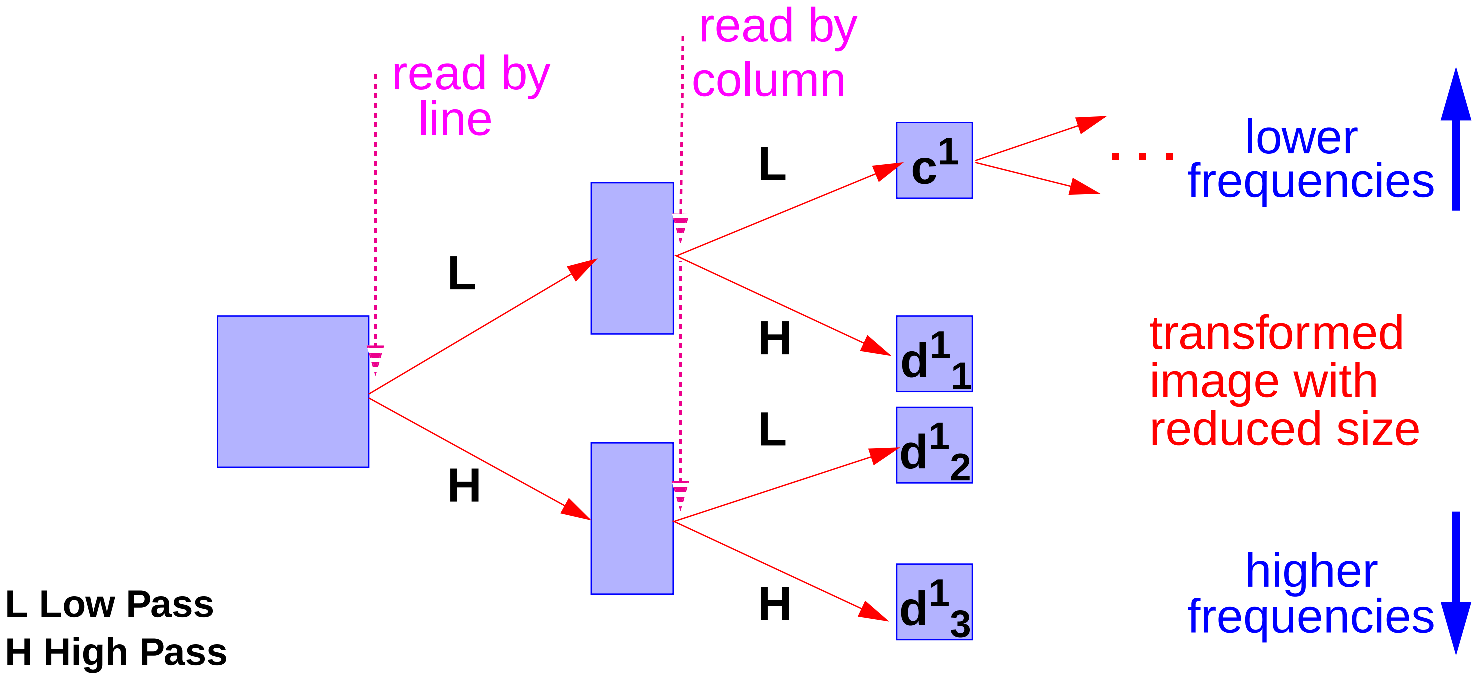
**Goal for developing Wavelets:**

- to get a better identification of which data is relevant to human perception
  ⇒ higher compression ratio
- iterative operation on whole image
  ⇒ overcomes visible block structures & introduces inherent scaling

# Wavelets: Principle

## Idea:

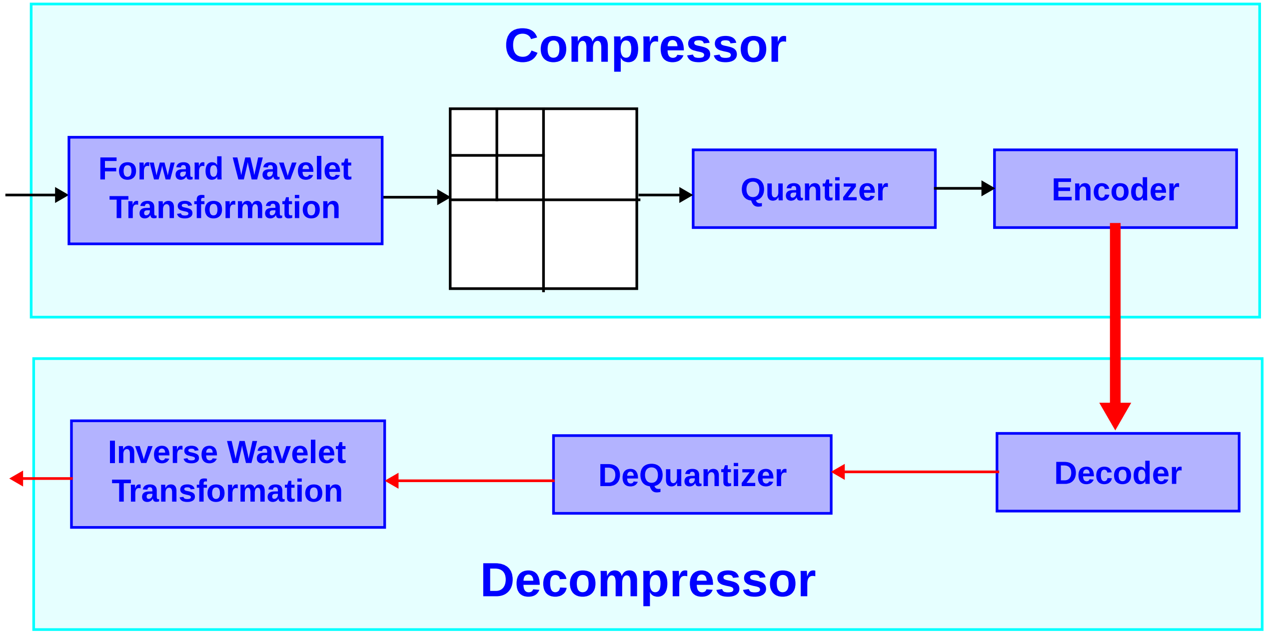
- To split image recursively by using high and low pass filters




## Next Steps:

- Quantization (according to importance) of transformed images
- Entropy encoding

# Wavelet Compression / Decompression



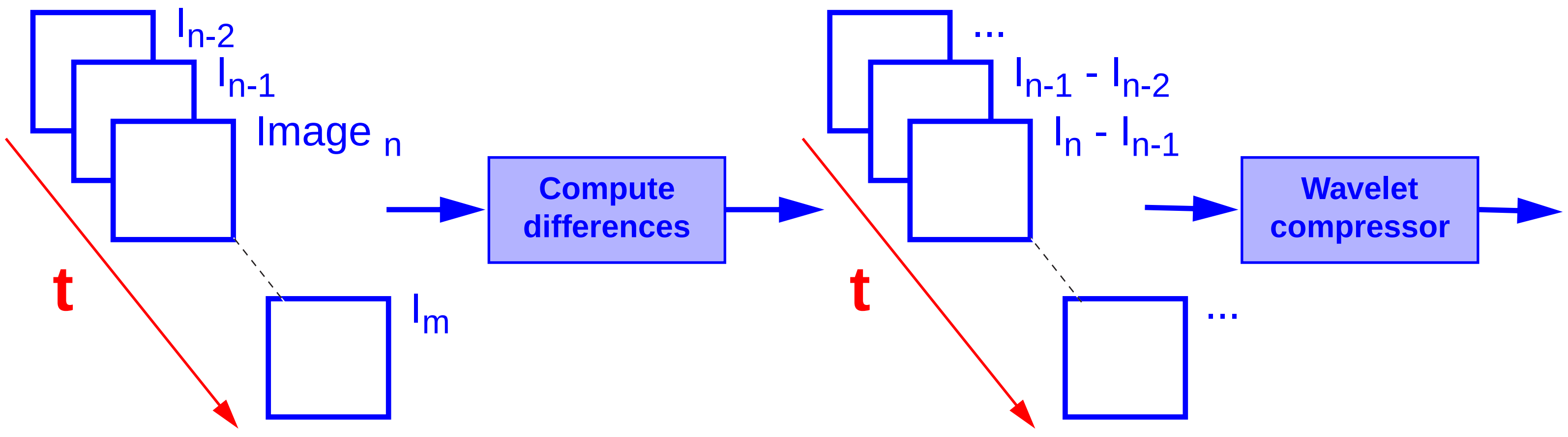

- Wavelets are various functions, i.e. Function Class
- Application of Algorithm of Mallat for archieving efficiency

# Wavelets: Further Issues

## Edge detection reduces high frequencies:

- first extract detected edges
- then apply wavelets to such a filtered image

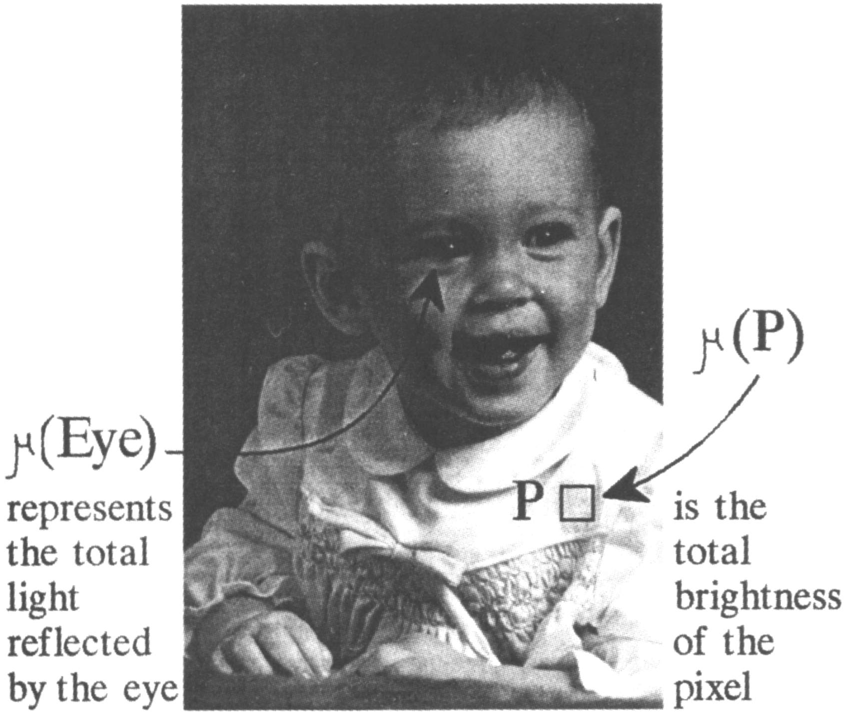
## Application to video:



# 13. Fractal Image Compression

## Idea:

- To search for self-similarities in image by using affine transformations and brightness-correction
- To build a function μ with image as fix point by coding similarity information



# Fractal Image Decompression

# 14. Conclusion

## JPEG:

- Very general format with high compression ratio
- SW and HW for baseline mode available

## H.261:

- Established standard by telecom world
- Preferable hardware realization

## MPEG family of standards:

- Video and audio compression for different data rates
- Asymmetric (focus) and symmetric

## Proprietary systems: e.g. DVI Product

- Migration to the use of standards

## Next steps: wavelets, fractals, models of objects