Multimedia networked applications: standards, protocols and research trends

Maria Teresa Andrade
FEUP / INESC Porto
mandrade@fe.up.pt ; maria.andrade@inescporto.pt
http://www.fe.up.pt/~mandrade/ ; http://www.inescporto.pt

Multimedia traffic

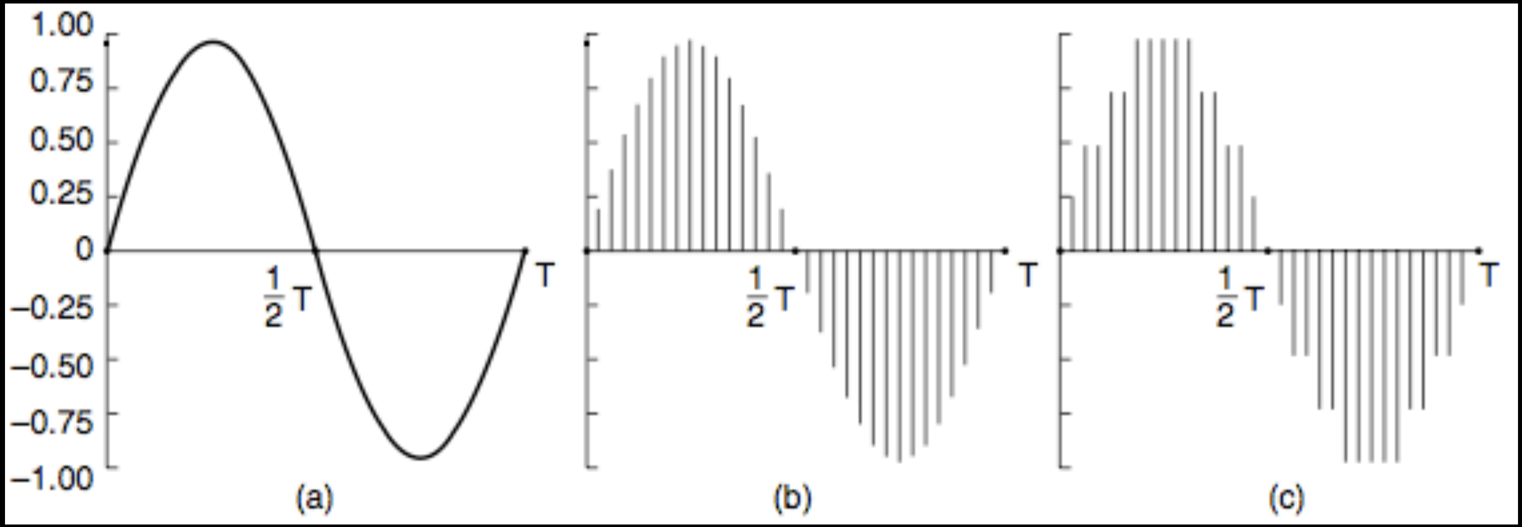
- Multimedia traffic
  - digital content sources
    - principles of media compression
  - media compression standards
    - JPEG
    - MPEG2, MPEG4
    - H264/AVC)

Digital sources

- digital sources are obtained by digitizing analogue sources
- two essential aspects must be considered when digitizing
  - sampling
    - the continuous signal is partitioned into discrete quantities in time, in space or in both dimensions
      - temporal and spatial samples
    - to allow reconstruction of the original signal, the sampling frequency must be greater than or equal to the Nyquist frequency (fs≥2*fmax)
  - quantization
    - each discrete sample is represented in a finite value scale
  - the frequency with which samples are obtained and the number of values used for quantizing, dictates the fidelity of the digital copy and the required bandwidth

Digital sources

- sampling at a rate less than fs introduces aliasing when recovering the analogue signal
- quantization introduces an error, seen as a distortion, when reconstructing the analogue signal



a) original analogue signal b) digital values, floating point c) integer digital values, 8 bits

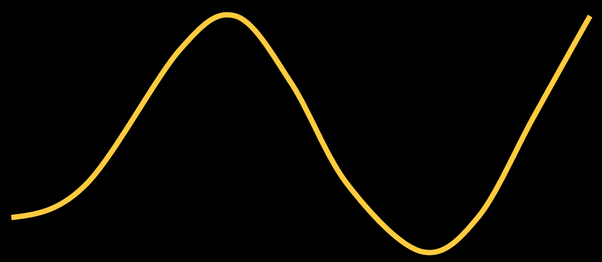
Digital Sources

- quantization error - what it is and how to reduce it
  - difference between the real value of the analogue signal at the instant when the sample is obtained and the value of its digital representation on a finite value scale
  - to reduce it,
    - use a scale as large as possible :-)
      - but this increases the bandwidth :-(
  - use dithering
    - dither is a controlled error signal added to the sampled signal
    - in audio, it takes advantage of the fact that the human ear is much more sensible to distortion due to patterns then to random noise
    - in image it allows to mask the quantization error in systems with a reduced color space

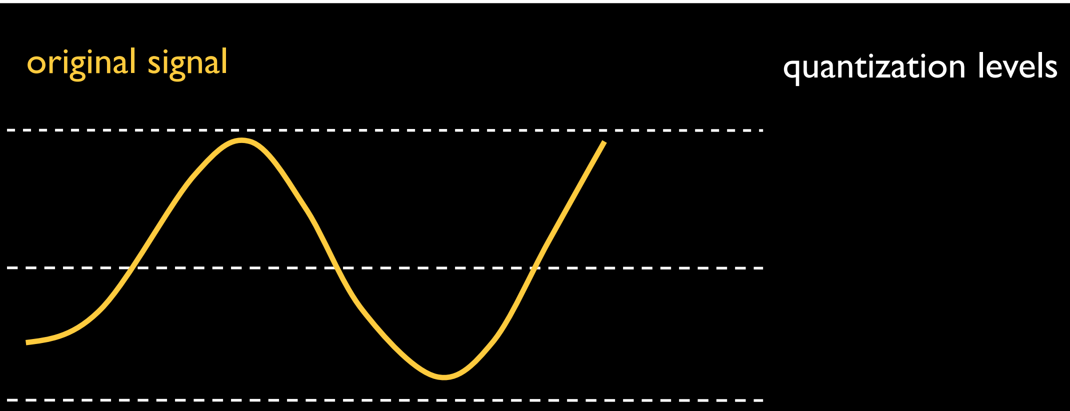
Digital sources - dither

Digital sources - dither

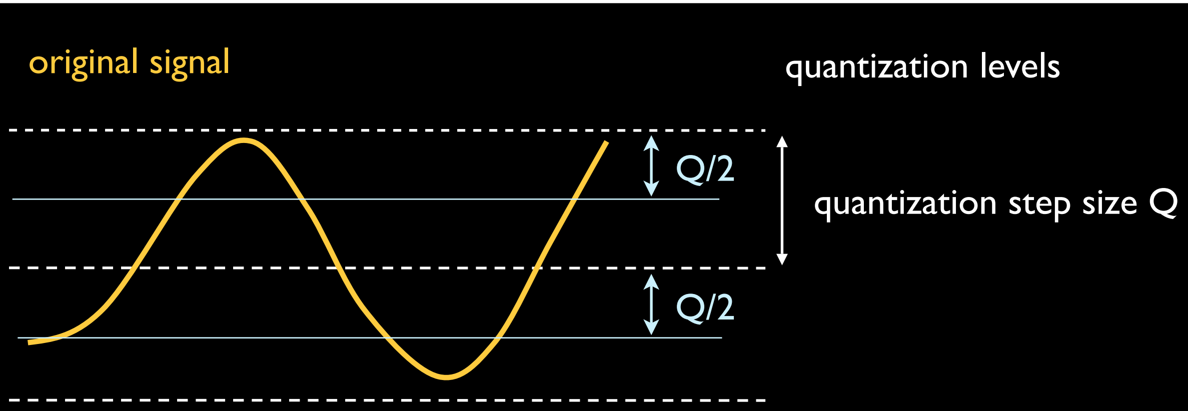
original signal



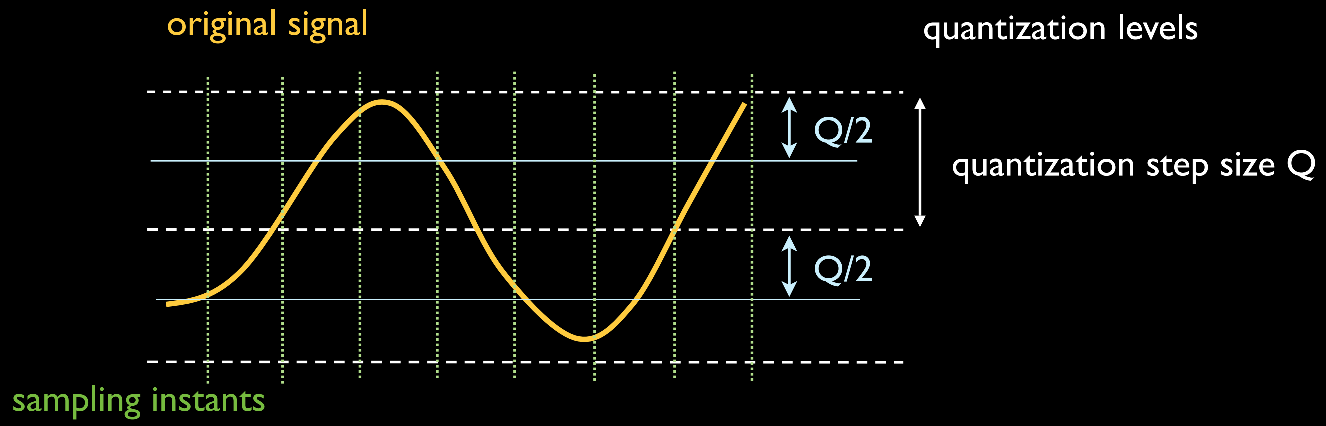
Digital sources - dither



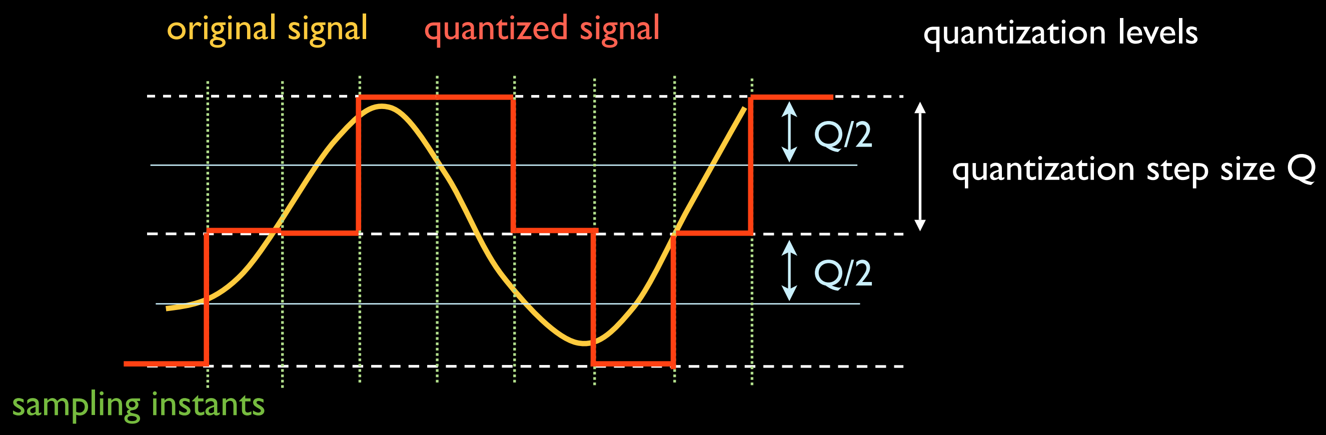
Digital sources - dither



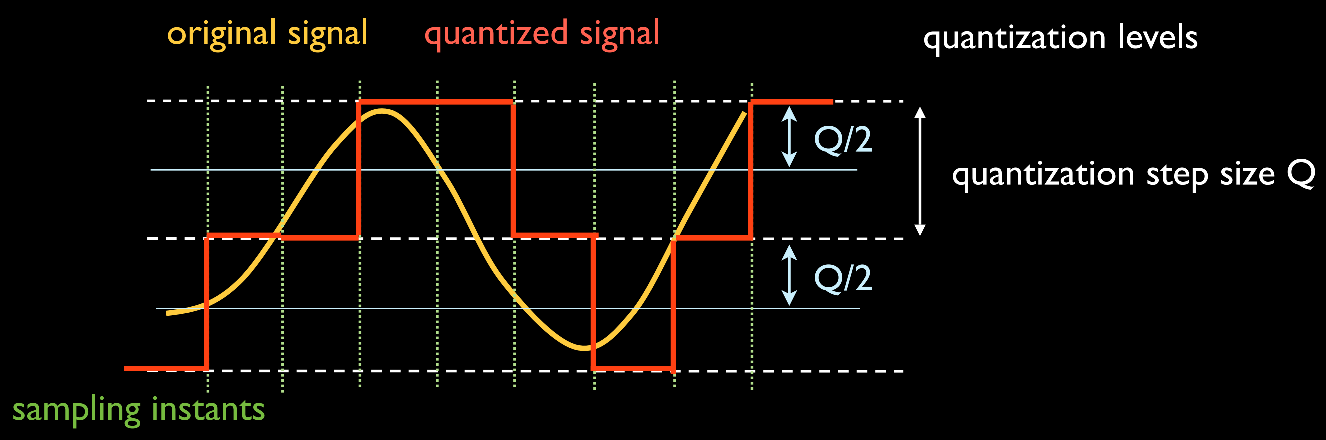
Digital sources - dither



Digital sources - dither

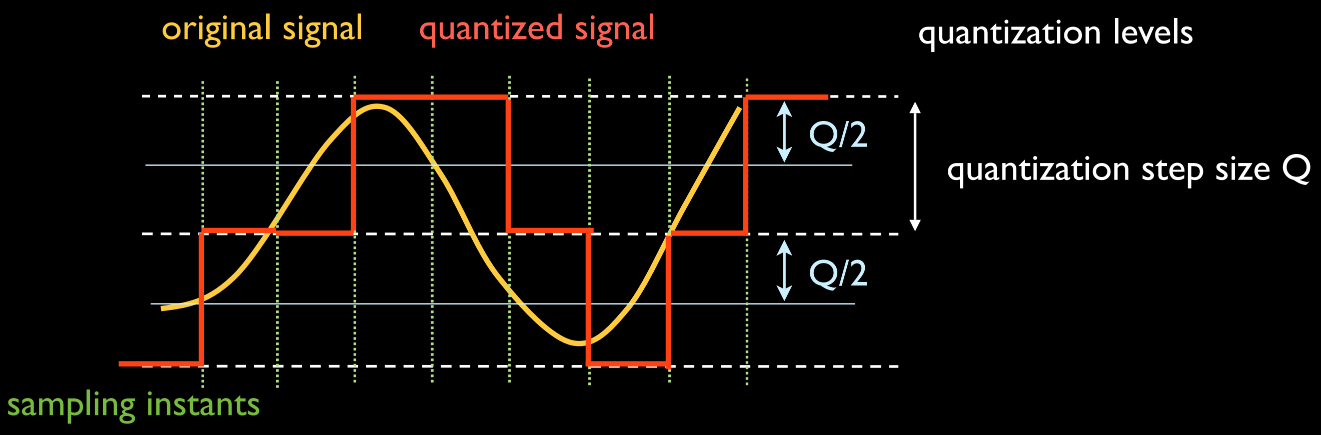


Digital sources - dither

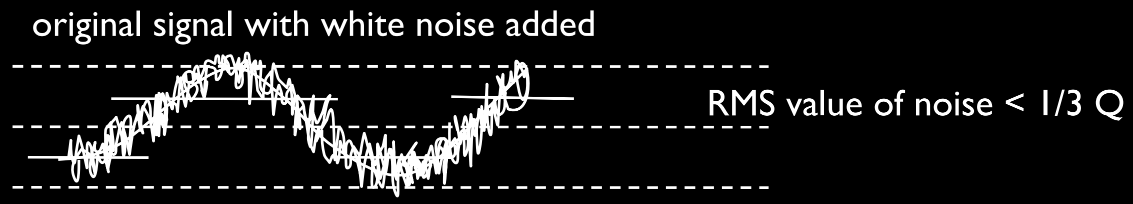


input signal (sinusoid) is digitized as a square wave - very poor approximation

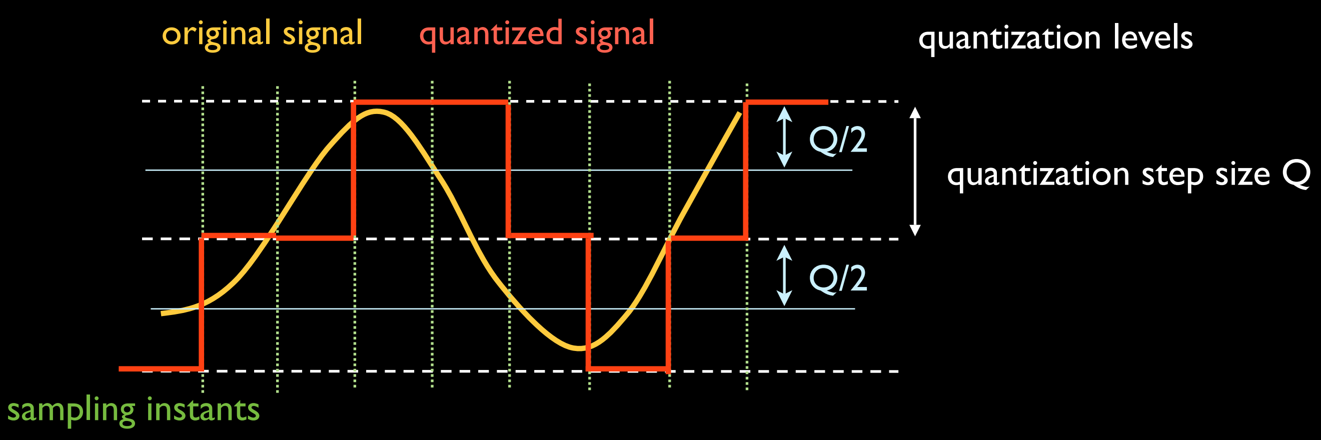
Digital sources - dither



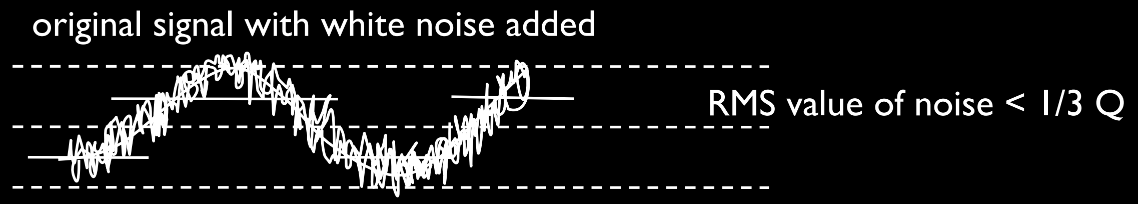
input signal (sinusoid) is digitized as a square wave - very poor approximation

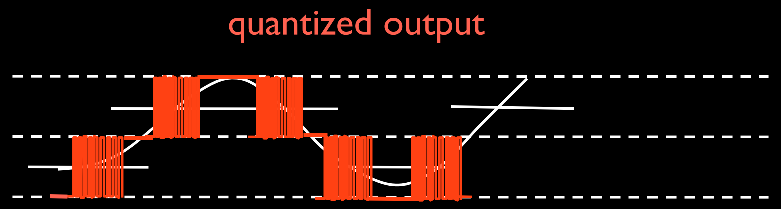

original signal with white noise added
RMS value of noise < 1/3 Q

Digital sources - dither



input signal (sinusoid) is digitized as a square wave - very poor approximation

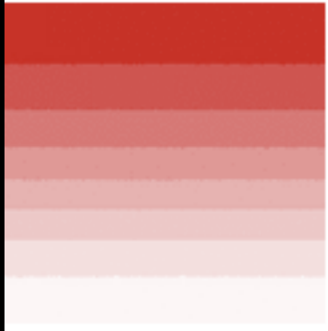

original signal with white noise added
RMS value of noise < 1/3 Q



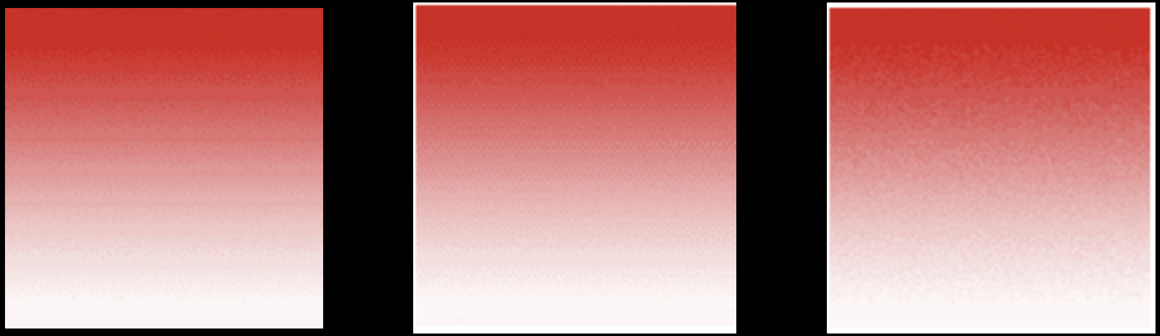
Digital sources - dither

- Examples of images with reduced color space with and without dithering

256 colors



16 colors



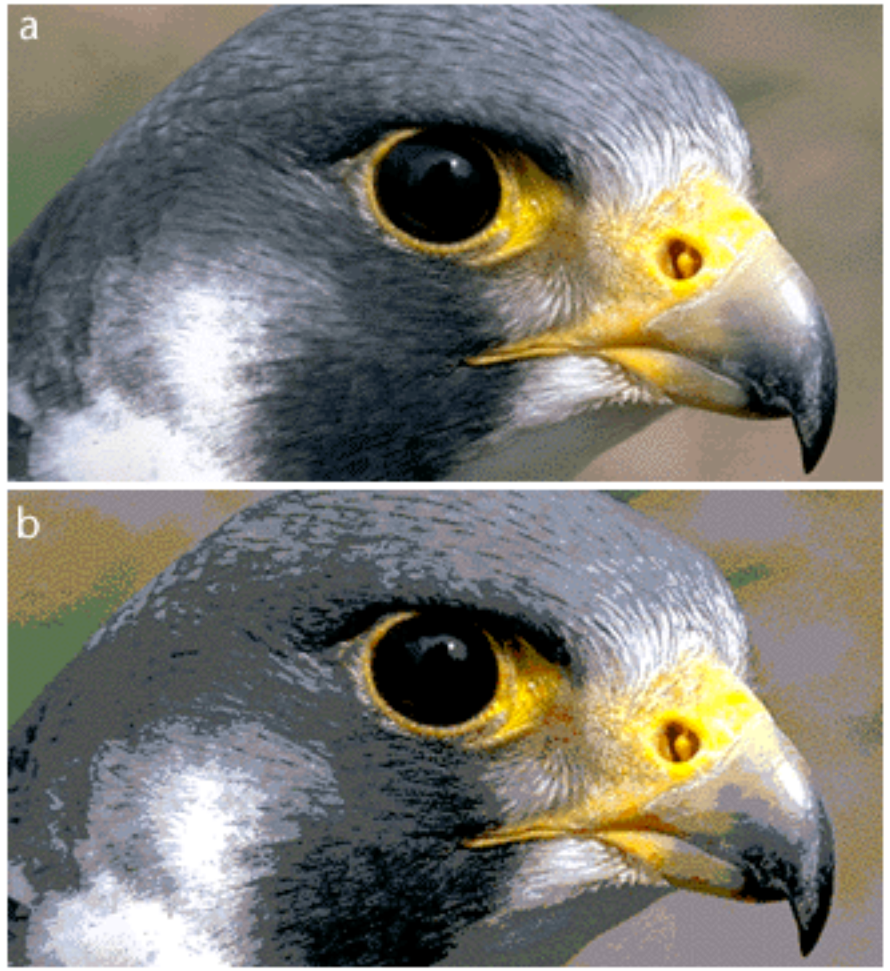
16 colors with different types of dithering

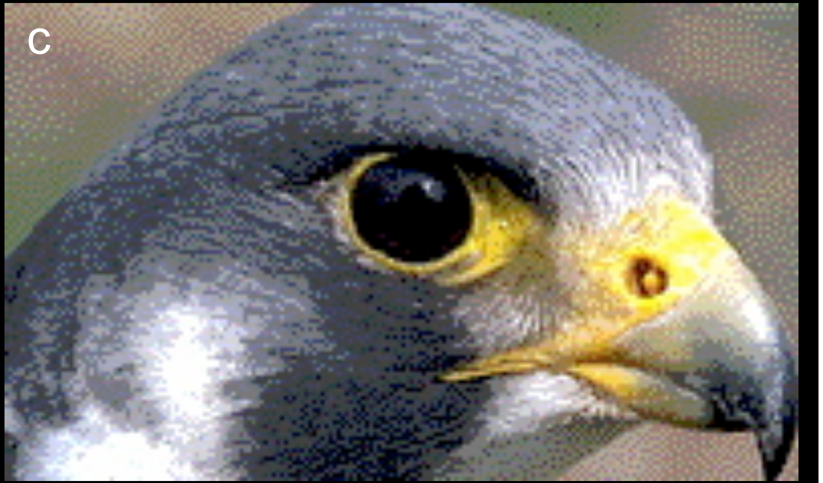
Digital sources - dither

a) original true color (24 bits)

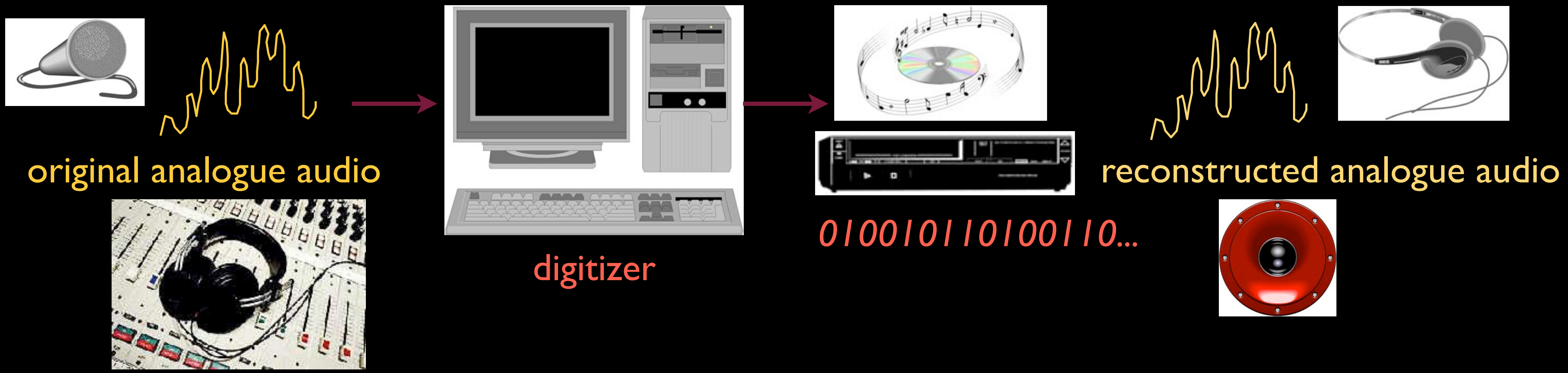
b) copy with 8 bits, 256 colors

c) copy with 8 bits, 256 colors and dither





Digital sources - audio



original analogue audio

digitizer

0100101110100110...

reconstructed analogue audio

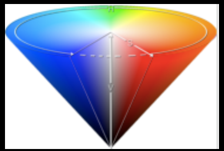
- uncompressed digital audio formats
  - PCM (Pulse Code Modulation), 16 bits / sample
    - used in studio / professional environments, stored in DAT
    - in the consumer market, used in the CD's
    - transmitted between equipments using the AES/EBU standard
  - many proprietary or de-facto standards on the Internet
    - (.wav): Windows Waveform
    - AIFF (.aif): Apple and SGI
    - AU (.au): Sun and NeXT

Digital sources - video

- each image, from a sequence captured from a camera is divided into picture elements (pixels), containing information of brightness and color
  - a set of pixels in the horizontal axe is a line
  - a set of vertically consecutive lines is an image
  - a set of timely consecutive images is a video signal
- initially the sampling was done directly on the composite video signal (PAL or NTSC)
  - brightness and color were processed the same way
  - did not benefit from the characteristics of the human visual system (HVS)
- the HVS is more sensible to brightness high-frequencies than to variations in the color space
  - it is thus possible to reduce the number of color samples without noticeable degradation

Digital sources - video

- An important aspect to look at when digitizing video is the color space and the color sub-sampling rate
  - different spaces for representing color
    - some are more efficient from the perceptual point of view (closer to the HVS)
    - others are more efficient in technological terms (more compact representations, more correlation between components, ..)
    - RGB, HSV / HSL / HSI / HSB (Hue, Saturation,Value/Lightness/Intensity/ Brightness),YUV /YIQ / YCrCb (Y=luminance; UV, IQ, CrCb - color difference)



- ITU-R Rec. 601 is a standard that samples luminance (brightness) and color components separately
  - sampling frequency of 13.5 MHz; space color YCrCb
  - relation between number of Y and Cr/Cb samples expressed as x:y:z

Digital sources - video

- The sampling scheme is expressed as x:y:z
  - x is the relative number of luminance (Y) samples
  - y is the relative number of chrominance (Cr and Cb) samples in the odd lines
  - z is the number of chrominance (Cr and Cb) samples in the even lines
  - ex., 4:2:0 indicates that for each 4 samples of luminance (Y) there are 2 chrominance samples (1 Cr and 1 Cb) only in the odd lines
    - relative compression of 1:4 of the color signals
    - for each block of 2x2 pixels, there are 4 samples of luminance (Y) and only 1 sample of each color signal (Cr and Cb)
    - format used in broadcast TV
  - ex., in 4:2:2 for each 4 Y samples there are 2 chrominance samples (1 Cr and 1 Cb) both in the odd and even lines
    - relative compression of 1:2 of the color signals

Digital sources - video

- Other video formats concerning the spatial resolution
  - HHR Half Horizontal resolution
    - used in MPEG-2
  - SIF Source Intermediate Format
    - used in MPEG-1
  - CIF Common Input Format e QCIF Quarter CIF
    - used in MPEG-1, videoconferencing, Internet
  - SDT Standard Television (480 lines x 640/704 pixels)
    - used in digital TV and DVDs
  - HDTV High Definition TV
    - 1152 / 1080 lines x 1440 pixels interlaced (1152i or 1080i in the States)
    - 720 lines x 1280 pixels progressive, 720p

Digital sources - video

HHR

Y 360 pels

576 pels
480

Cr 180

288
240

Cb 180

SIF

Y 360 pels

288 pels
240

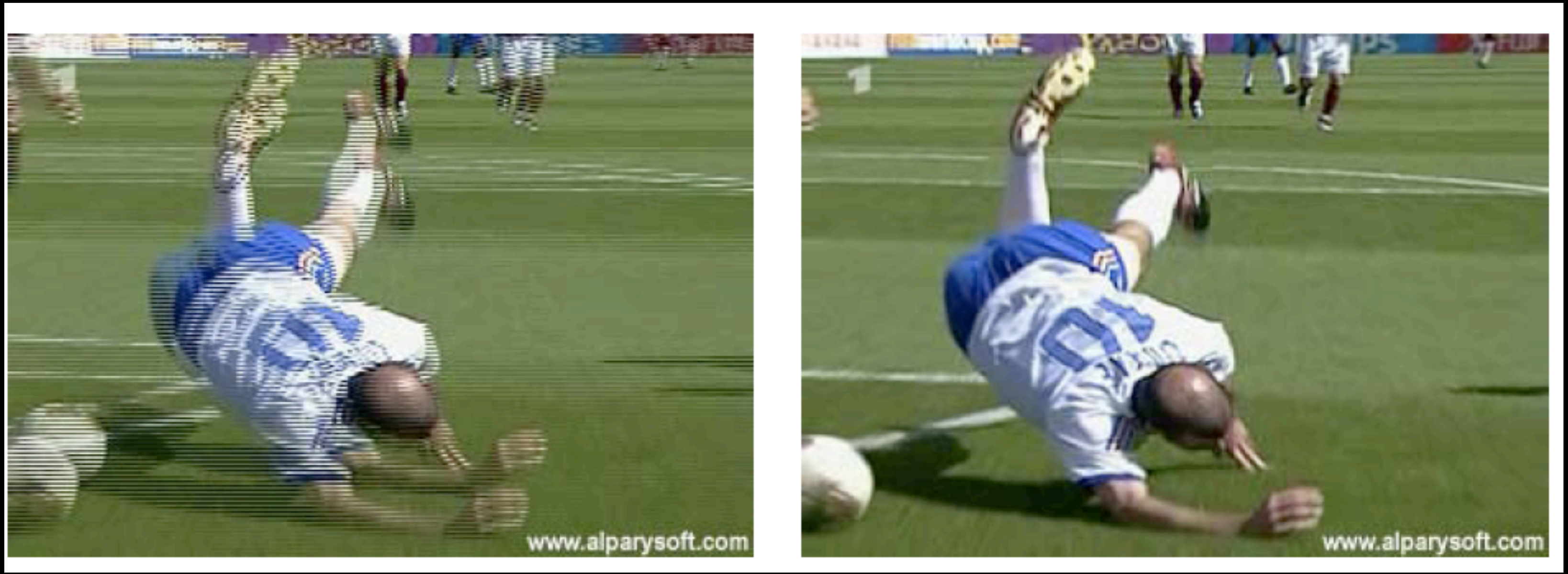
Cr 180

144
120

Cb 180

Digital sources - video

- Progressive versus interlaced
- analogue TV was interlaced
  - each picture is divided in two fields with half the lines (odd and even)
  - a trick to augment the refresh rate of the images on the screen without increasing the bandwidth
    - especially suitable for very bright images and high spatial detail
    - but problematic for fast movements



Principles of media compression

- compression algorithms can be lossy or lossless
  - lossless means that the original signal can be reconstructed without any distortion (ex. zip)
  - lossy means that some degradation will occur during reconstruction
    - amount of loss of information should be according to the application requirements
- loss is a distortion from the original data that can be measured through the SNR (Signal to Noise Ratio) or MSE (Mean Square Error)
  - it provides an indication of the distance between the original and the reconstructed signals

Principles of media compression

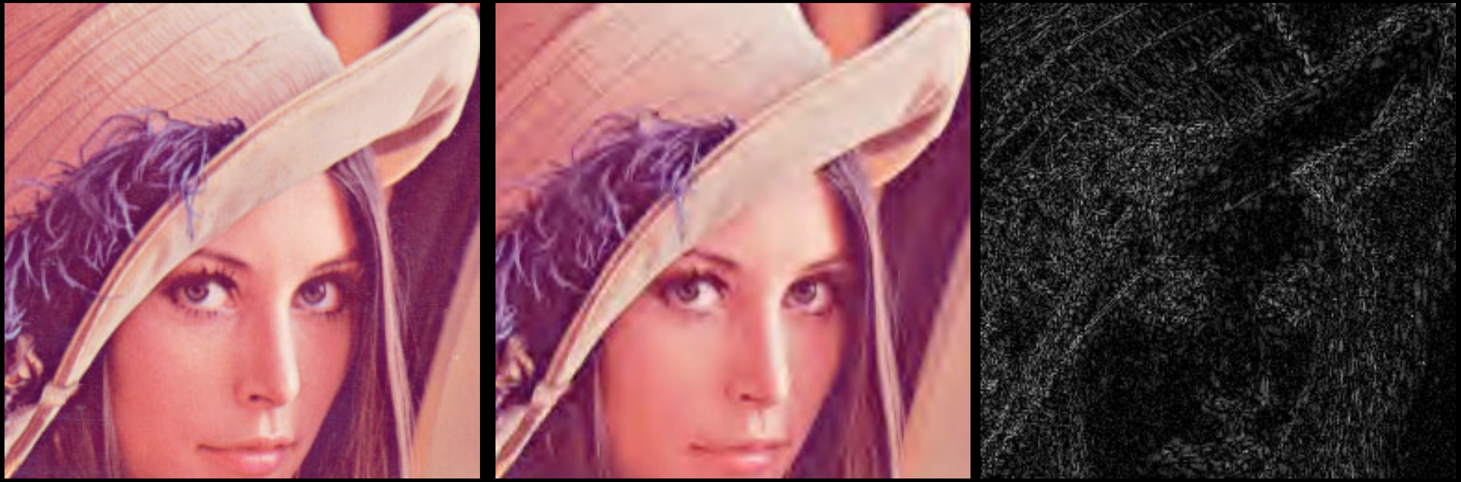
- important characteristics to look at when selecting a compression algorithm
  - efficiency
    - relation between compression rate and quality
    - compression can be measured by the relation of file sizes before and after compression
  - complexity
    - hardware or software only
    - amount of processing power consumed, number of operations per unit of time
    - real-time versus non rea-time
  - delay
    - how long is it necessary to wait before starting decoding a compressed signal? how much buffering is required?

Principles of media compression

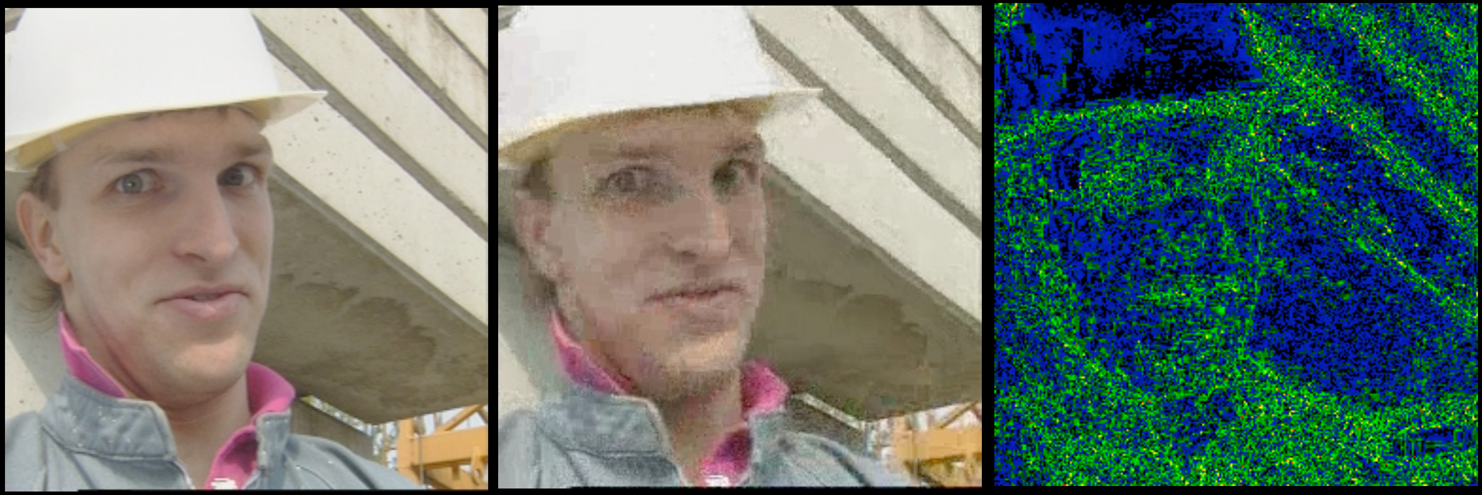
- It is thus important to be able to measure the quality or the efficiency (quality versus bit rate) of different compression schemes
  - objective measures
    - PSNR (peak signal to noise ratio), MSAD (mean sum of absolute differences), MSE (mean square error), ...
    - can be automated and performed in real-time
    - but sometimes are far away from the human perception of quality
    - measured in dB
      - a value of 50 dB indicates an almost perfect reconstruction
  - subjective measures
    - VQM (video quality metric)
      - use objective metrics that try to model the human perception
    - more difficult to perform
    - may use an audience to visualize sequences and grade them

Principles of media compression

- quality evaluation



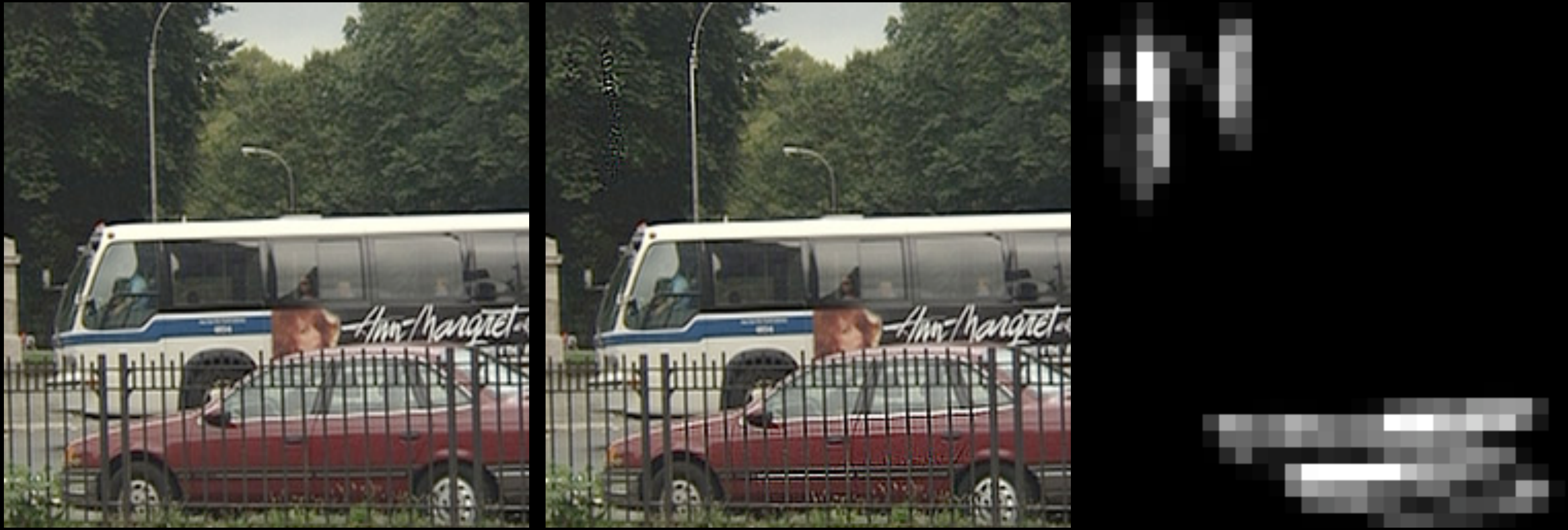
$$MSE = \frac{\sum_{M,N} [I_1(m,n) - I_2(m,n)]^2}{M * N}$$



$$PSNR = 10\log_{10}\left(\frac{R^2}{MSE}\right)$$

Principles of media compression

- quality evaluation



VQM - brighter blocks correspond to greater differences, hence less quality

Principles of media compression

- how is it possible to compress and still obtain a good representation of the original content?
  - taking advantage of redundancy that exists in the media signals
    - redundant data either duplicates information or brings no new information
  - taking advantage of the properties of the human perceptual system
    - perceptual redundancy (psychoacoustic and psychovisual)

Media compression - redundancy

- Given that
  - digital audio signal = succession of samples in time
  - digital image = square matrix of pixels (spatial samples)
  - video = sequence of images at a certain frequency in time
- neighbor samples in those signals (audio samples, pixels or images) are more or less correlated among them
  - i.e., part of their information is the same or very similar, thus, redundant

Media compression - redundancy

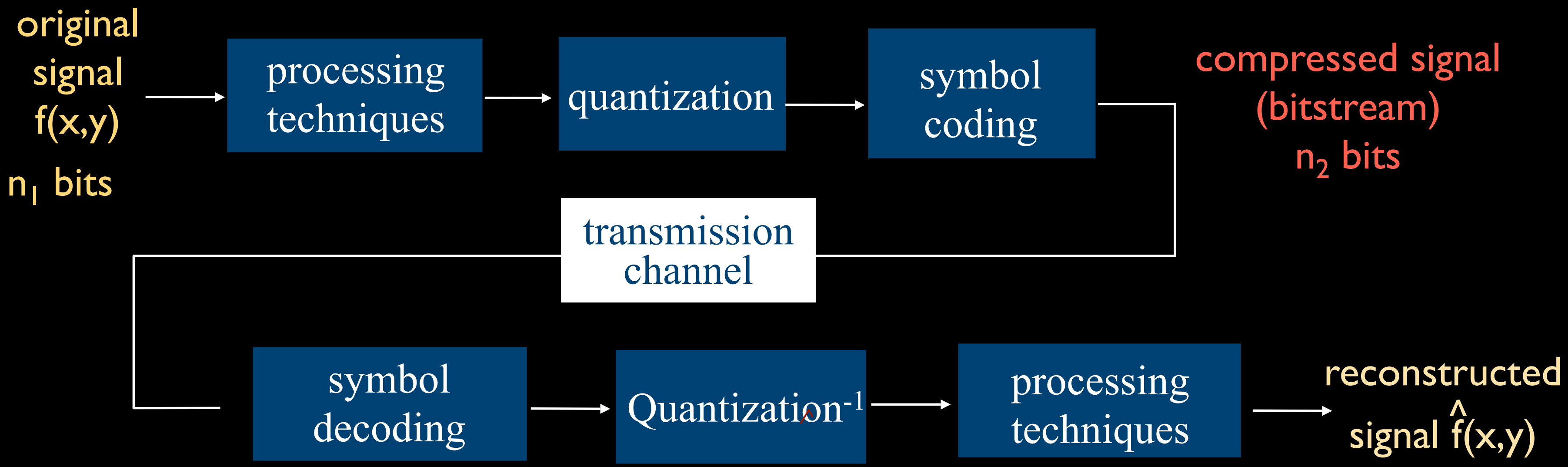
- several types of redundancy in audio and video signals
  - spectral redundancy
    - for example, duplication of information (redundancy!) between the RGB fundamental colors
      - hence the benefits of using other color spaces :-)
  - spatial redundancy
    - for example, between pixels of an image that are spatially co-located
  - temporal redundancy
    - for example, between two consecutive images of a video sequence with moderate motion; pixels belonging to the background of two images in a video sequence where only people are moving
  - statistical or coding redundancy
    - code words with length larger than necessary
    - code inefficiency

Media compression - redundancy

- Redundancy relative to the human perception system
  - part of the visual and audio information is not perceived by the human visual and auditory systems
    - for example, the human eye is much less sensitive to the high frequencies than to the low frequencies
    - also, it is less sensitive to the chrominance variations than to the brightness variations
    - the human ear cannot distinguish neighboring sounds in time when those signal have very different amplitudes or frequencies

Media compression - coding tools

- typical, high-level, codec block diagram



original signal f(x,y)

n_1 bits

processing techniques

quantization

symbol coding

compressed signal (bitstream)

n_2 bits

transmission channel

symbol decoding

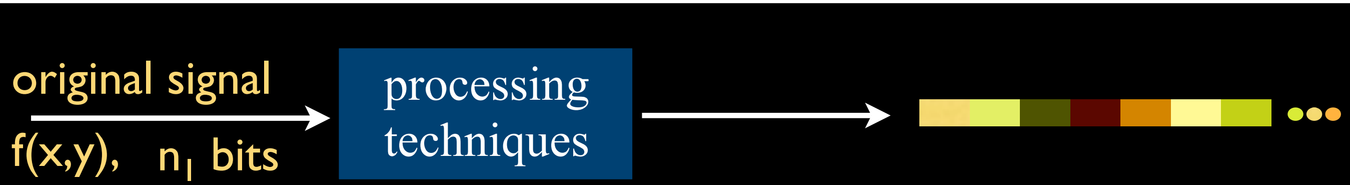
Quantization^{-1}

processing techniques

reconstructed signal $\hat{f}(x,y)$

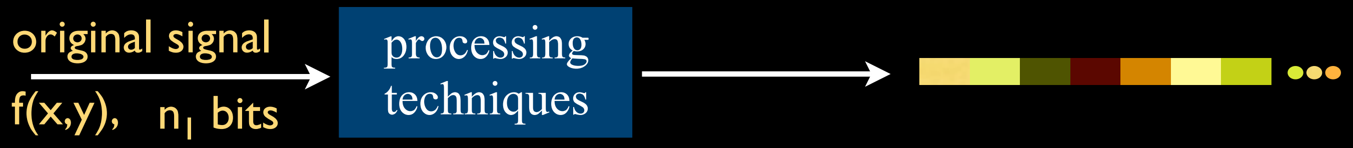
- each block tries to eliminate a certain type of redundancy
- overall compression ratio = n1/n2

Media compression - coding tools



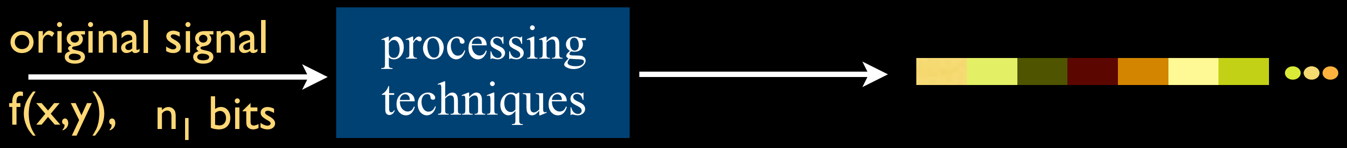
- the goal of this block is to eliminate (reduce) spatial and temporal redundancy
    - taking profit of the human perceptual redundancy
- it is assumed that the spectral redundancy has already been taken care off
    - color space already the appropriate one!
- incorporates a set of distinct techniques, in both the spatial-temporal and frequency domains
    - motion estimation/compensation and spatial transforms
- transforms the input signal into a non audible or non visioned format
    - sequence of symbols

Media compression - coding tools



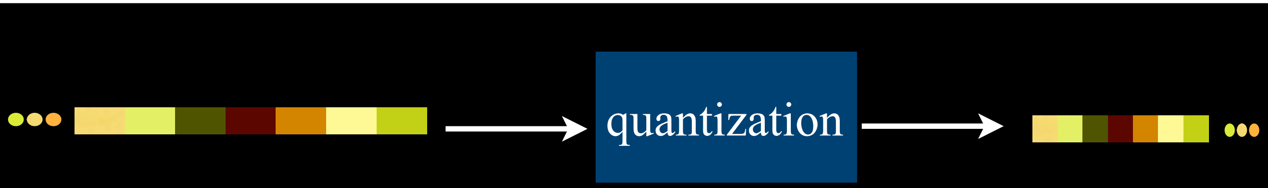
- motion estimation/compensation
  - eliminates temporal redundancy
  - divides the image into blocks of pixels
  - for each block, it searches for similar blocks in the neighbor images within a region
  - it selects the motion vectors (x,y coordinates) that point to the block that differs the less from the current block
    - that yields the smallest prediction error

Media compression - coding tools



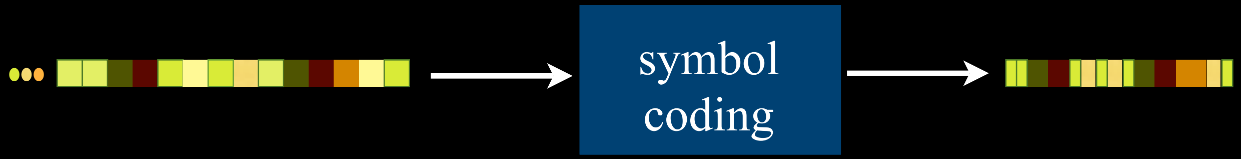
- spatial transform
  - translates the signal from the spatio-temporal domain into the frequency domain
  - the transform is applied to the prediction error, obtaining a set of transform coefficients
  - the energy of the prediction error becomes concentrated in a small number of coefficients
  - great part of the coefficients do not convey relevant information
    - they are thus redundant and can be eliminated!

Media compression - coding tools



- responsible for some compression (and distortion!), eliminating statistical redundancy
  - eliminates high frequency information (not well perceived by the HVS) by mapping or quantizing symbols at the output of the processing block into a set of discrete values
- the compression degree achieved depends on the number of levels of the quantizer and overall dynamic range
  - number of bits per word
  - quantizer step size
- a small number of levels or reduced dynamic range may result in severe distortion
  - depends on the type of content being compressed

Media compression - coding tools



- the symbol encoder, explores statistical or coding redundancy
- employs entropy coding, variable-length codes
    - RLC, Run Length Coding
    - SFC, Shannon-Fano coding
    - Huffman
    - arithmetic coding
- assigns shorter codewords to the symbols that occur more frequently

Media compression - coding tools

- overview conclusion
- compression is achieved by eliminating redundancy
  - spectral, between primary color components
    - converting to different color space where components are less correlated
  - temporal, between consecutive images in the case of video
    - using (motion compensated) predictive techniques
  - spatial, between spatially adjacent or close pixels of an image
    - applying a transform to the motion error thus concentrating energy
  - statistical or coding
    - using entropy coding
    - quantizing

Media compression - motion estimation

- motion estimation, ME, eliminates temporal redundancy between images
  - can achieve high degree of compression but it must have a good precision otherwise
    - errors in the decoded image will be significant, or
    - the gain will be small
      - if it does not calculate good motion vectors for each block, then the residual error will be large and thus the amount of information to be sent (the residual error + motion vector) will be significant
  - it is one of the most computationally intensive operations
    - 50-90% of the total compression system
- it is thus imperative to obtain good ME implementations to achieve good quality-bit rate-complexity codec

Media compression - motion estimation

- block-based technique, known as “Block Matching Algorithm” (BMA)
  - simpler to implement and simultaneously more efficient
  - each Y picture is divided into N × N blocks, designated as macroblocks (MB);
  - each MB of the current image is compared with candidate MBs of the reference image
    - usually within a window of pre-defined dimensions
  - the best match is the one that yields a smaller error/distortion
    - i.e., the motion vector (x,y coordinates) that points to a MB which differs the less from the current MB
  - the computed motion vector together with the prediction error is the information to be further processed and sent to the decoder
  - distortion is usually measured as the SAD

Media compression - motion estimation

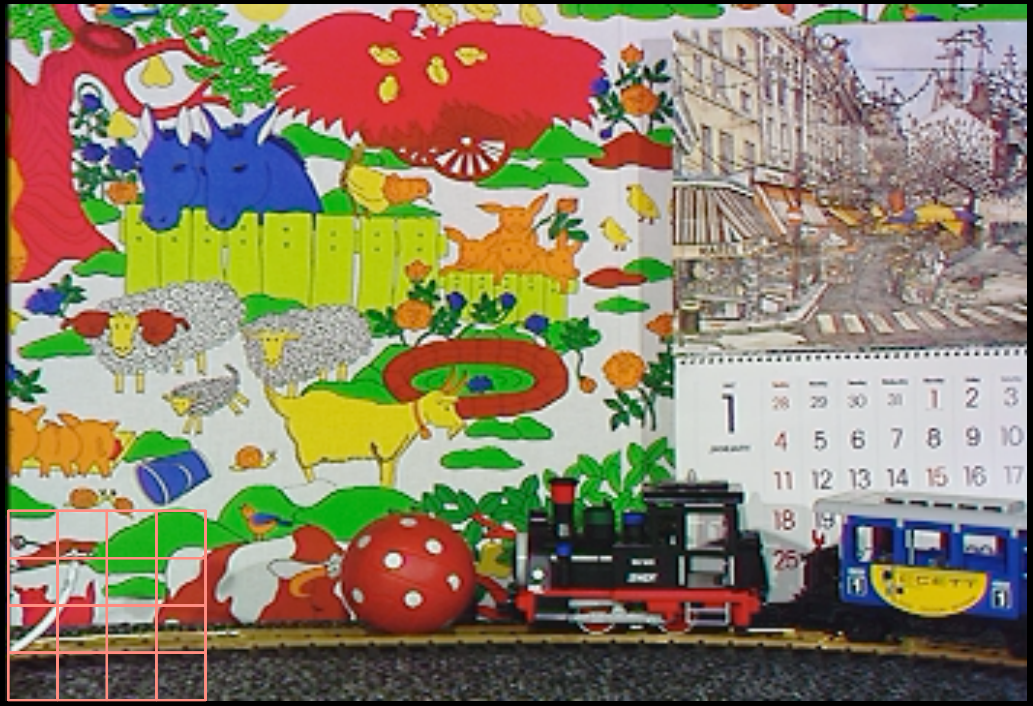


image 1, t

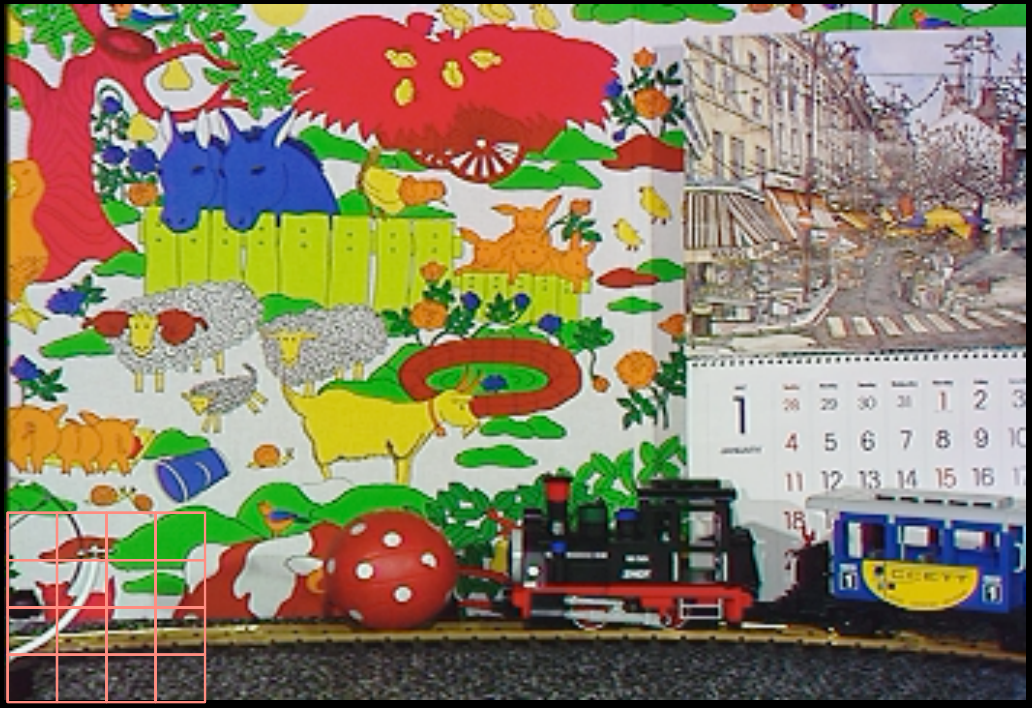


image 2, t+40ms

Media compression - motion estimation

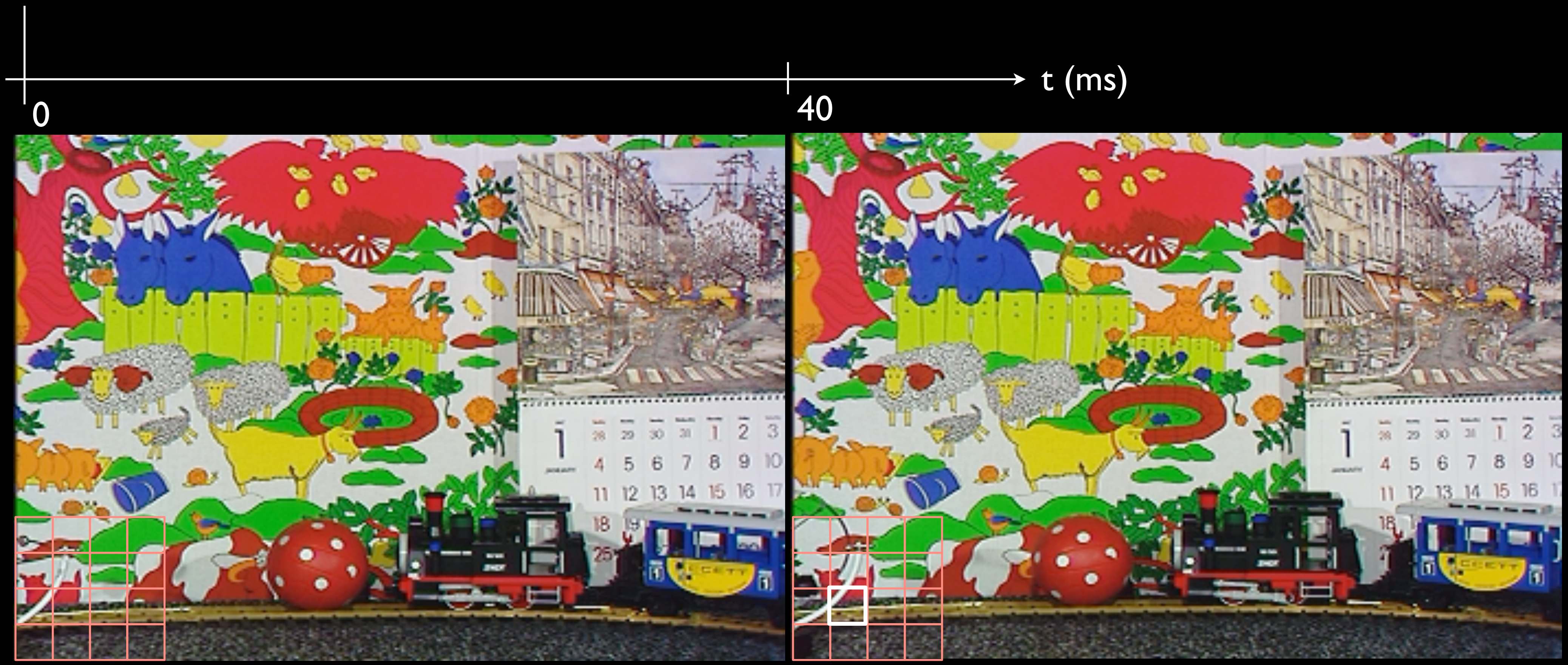
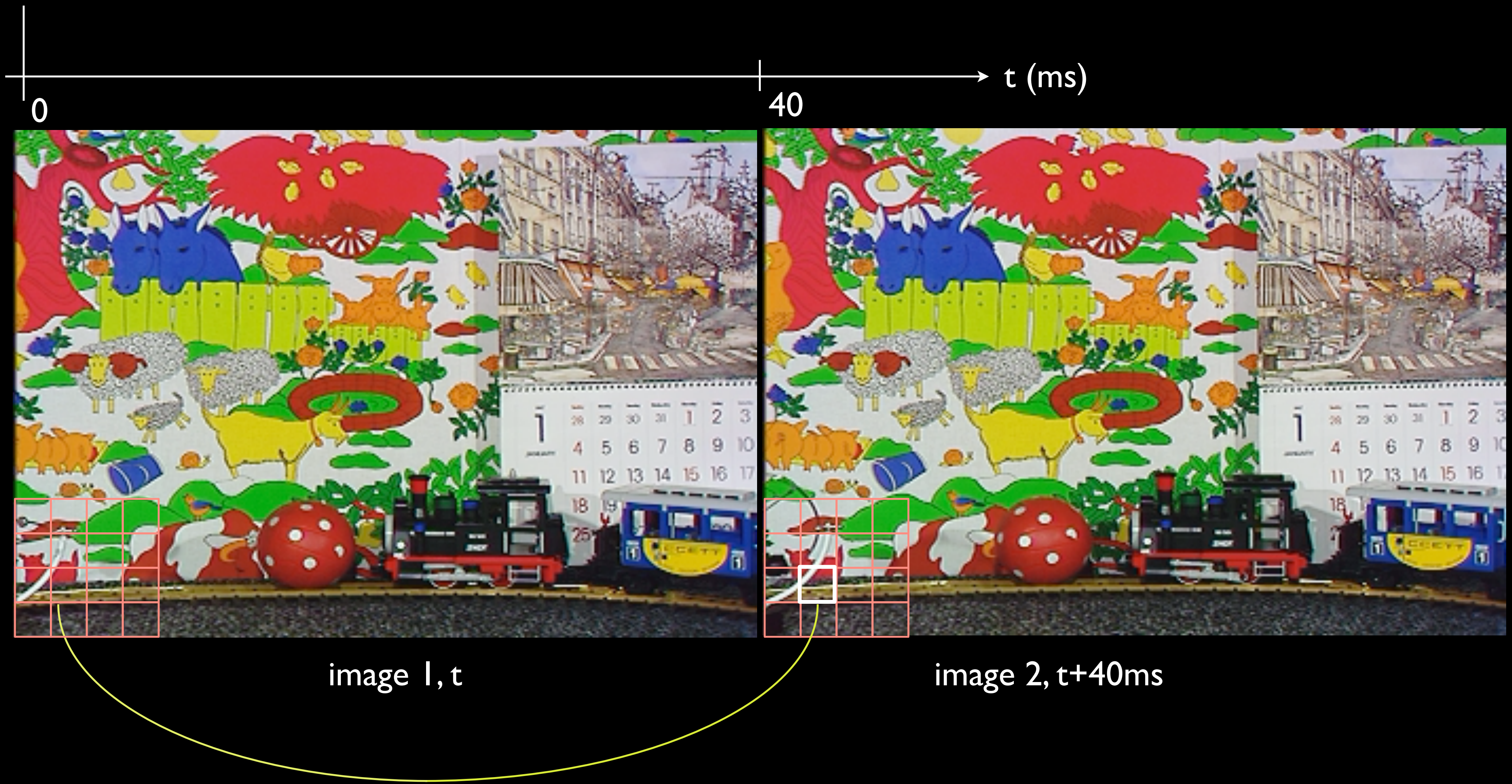


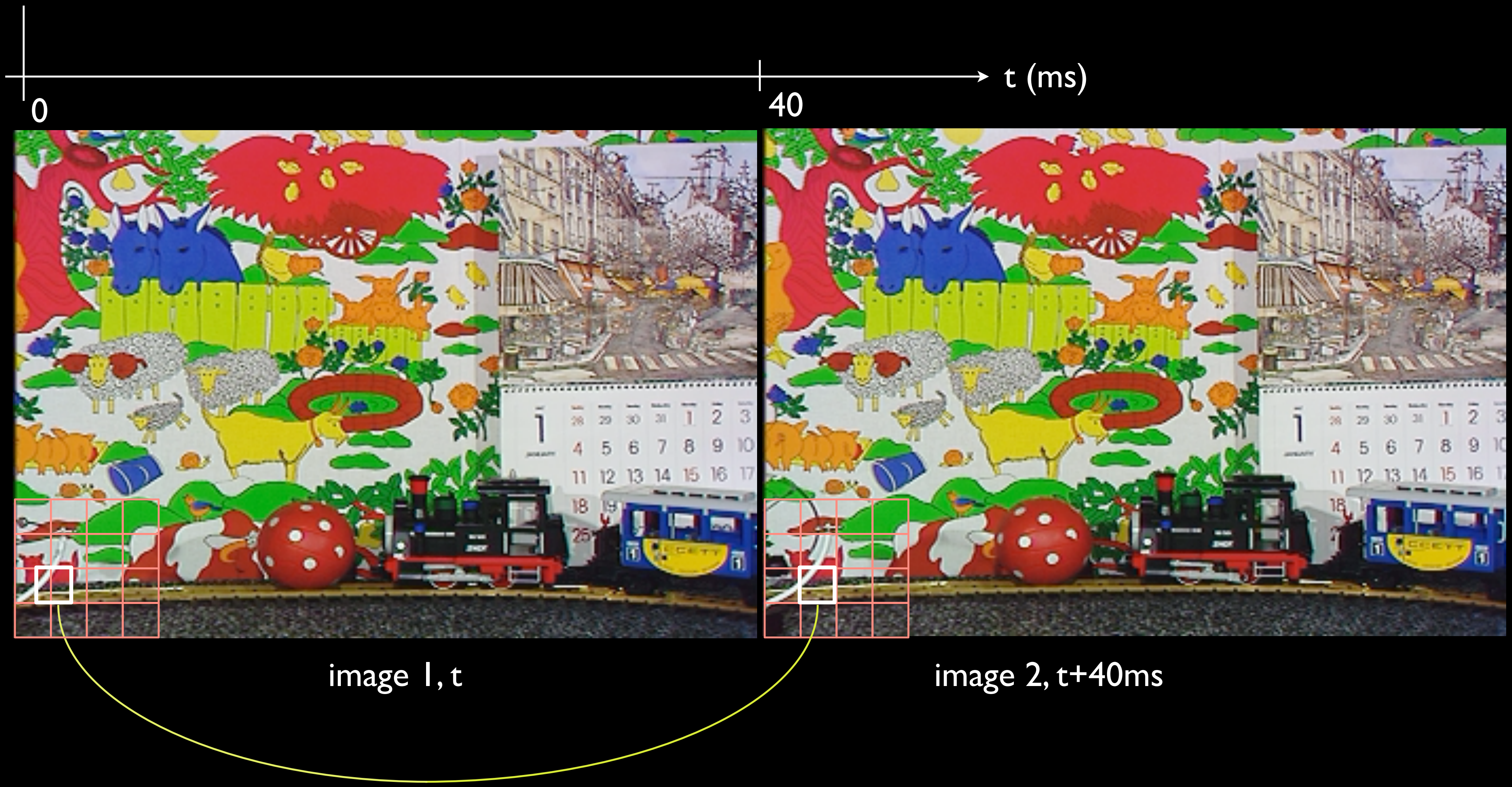
image 1, t

image 2, t+40ms

Media compression - motion estimation

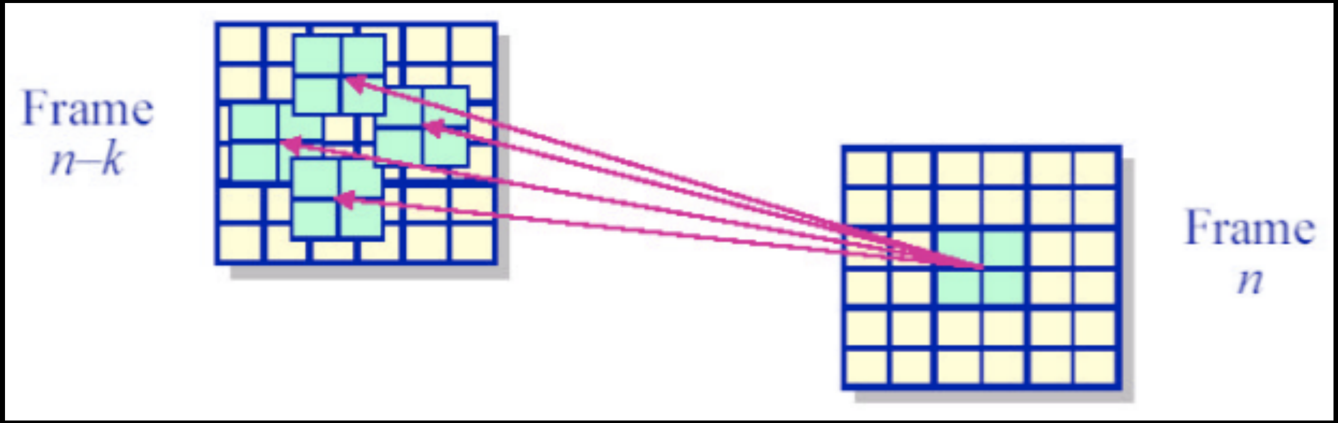


Media compression - motion estimation

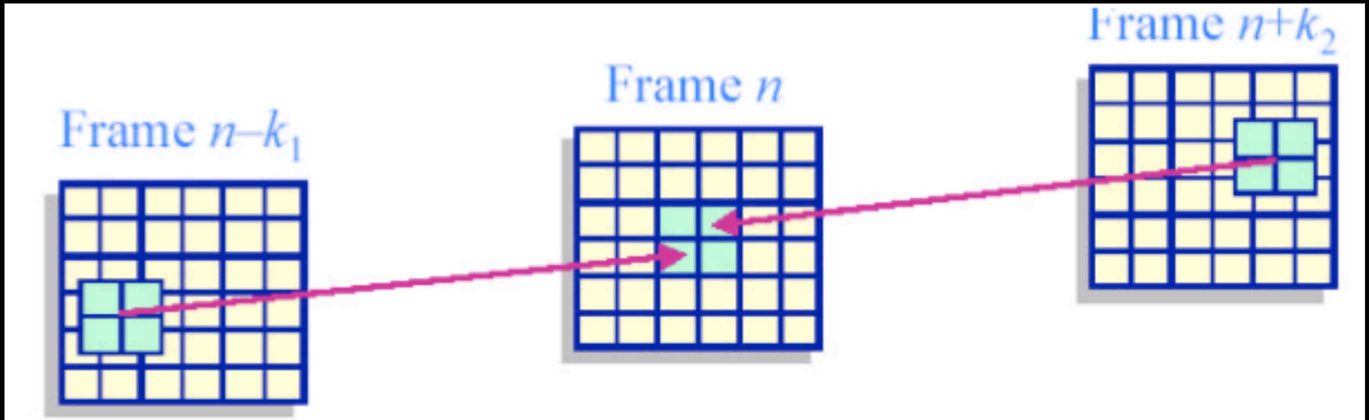


Media compression - motion estimation

- backwards estimation



- backwards and forward - bidirectional - estimation



Media compression - motion estimation

- parameters to consider
  - MB and search window size
  - search method
  - matching criteria
- generally smaller MB sizes lead to better results
  - smaller errors, best matching
  - but it loses efficiency, increasing the overhead
- generally, larger search windows lead to better results
  - more chances to find the best matching
  - but larger sizes, increase processing time and complexity
  - Depends on the distance to the reference picture

Media compression - motion estimation

- search methods
  - logarithmic search
  - three-step search
  - conjugate directions
  - hierarchical search
  - full search
- a block-based approach
  - reduces the overhead
  - suitable to regions that contain arbitrary shape moving objects
- translational model
  - simple (x,y coordinates) but does not treat efficiently rotations, zooming, etc

Media compression - motion estimation

- matching criteria

$$\mathrm{SAD}(m,n) = \sum_{i=0}^{N-1}\sum_{j=0}^{N-1} |c(i,j) - s(i+m, j+n)|$$

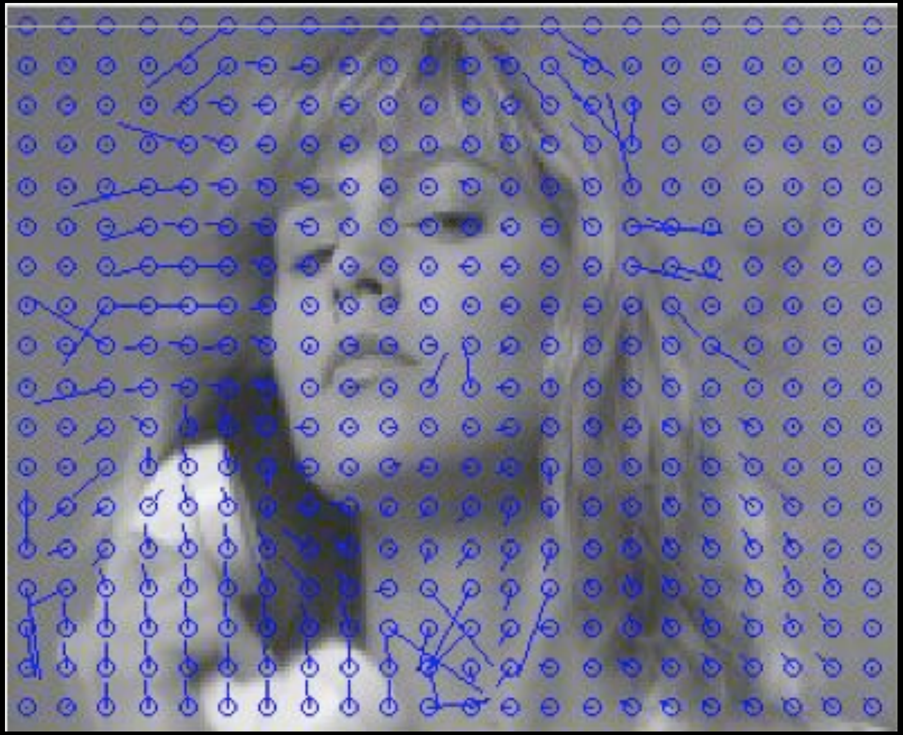
$$\mathrm{MV} = \{(u,v) \mid \mathrm{SAD}(u,v) \leq \mathrm{SAD}(m,n); \\ -p \leq m,n \leq p-1\}$$

- SAD(m, n) is the distortion of the candidate block standing in position (m, n);
- c(i,j), with 0≤ i ≤N- 1, 0≤ j ≤N-1, represents the pixels of the current block
- s(i,+m,j+n) represents the searched block
- [-p,p-1] is the search area
- N x N is the block size
- MV represents the motion vector associated to the current block that leads to the smallest SAD

Media compression - motion estimation

Media compression - motion estimation

Media compression - motion estimation



block-based motion vectors

Media compression - spatial transform

- values in the spatial-temporal domain are transformed into coefficients in the frequency domain
- the signal energy is compacted around the low frequency coefficients
- efficiency of compacting, and hence of the transform itself, is given by the number of coefficients that contain a large percentage of energy
  - if great part of the signal energy (more than 90%) is contained in a small number of coefficients, a great number of coefficients wont convey relevant information
    - they can be ignored thus leading to a significative compression rate

Media compression - spatial transform

- transforms can be applied:
  - to image blocks with n x n pixels
    - “Block-based transform coding”
  - to the complete image
    - “Frame-based transform coding”
- the former is the most used in popular compression scheme
- the methods “image-based” are more recent and are based on a decompositin of the image in the frequency domain

Media compression - spatial transform

- the application of a transform can be seen as an operation between matrices and vectors
  - an operator picks up an image at the input (bi-dimensional matrix) and produces a new image at the output, performing a point-to-point transformation
  - An operator is a function h(x, α, y, β), representing the degree of influence of the input position (x,y), in the output value positioned at (α,β)
  - because each image is a set of points, the application of the operator h upon a complete image f(x,y) with N x M points is expressed as

$$g(\alpha,\beta) = \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} f(x,y).h(x, \alpha, y, \beta)$$

Media compression - spatial transform

- it is a bi-dimensional convolution, that can be expressed in its matrix form
-
- g, vector with the transform image values
- g = Hf { H matrix representing the operation
- f, vector the original image values

- vector f is obtained by serializing the columns of the bi-dimensional input matrix
- matrix H is obtained taking the vectors $h^T_{\alpha,\beta}$ as its lines

Media compression - spatial transform

- selecting the best transform
- if the statistics of the input signal are known, it is possible to find the optimal transform for that signal
  - that's almost impossible given the non-stationary nature of video signals
  - it is chosen the transform that enables the subsequent processing to be more efficient
    - larger concentration of energy in a reduced number of coefficients
    - simple implementation that may allow realtime operation
  - in image and video algorithm the most popular is the DCT, Discrete Cosine Transform

Media compression - spatial transform

- usually the image is divided into blocks of 8x8 pixels, and the transform is applied to each block
    - the input image and its transform are vectors
    - the transform coefficients contain the energy of the input signal
        - there is always energy conservation but not always a uniform distribution of energy across the coefficients
    - the Karhunen-Loève transforms is usually the best one in theoretica terms
        - it achieves the best energy concentration
        - however efficiency depends on the input signal
            - complicates a lot its practical implementation

Media compression - spatial transform, DCT

- the Discrete Cosine Transform DCT
  - it consists in a sum of cosine functions with distinct frequencies
  - the output is a sum of functions
  - it can be obtained from the Discrete Fourier Transform DFT, which is very simple and widely used in signal processing
    - many studies done and efficient implementations available
  - but the DCT presents advantages over the DFT
    - its calculation does not involve complex numbers
    - it achieves a better concentration of energy for visual signals
    - it provides the best compromise complexity-efficiency

Media compression - spatial transform, DCT

forward DCT

$$S_{uv} = \frac{1}{4} C_u C_v \sum_{i=0}^{7} \sum_{j=0}^{7} s_{ij} \cos\frac{(2i+1)u\pi}{16} \cos\frac{(2j+1)v\pi}{16}$$

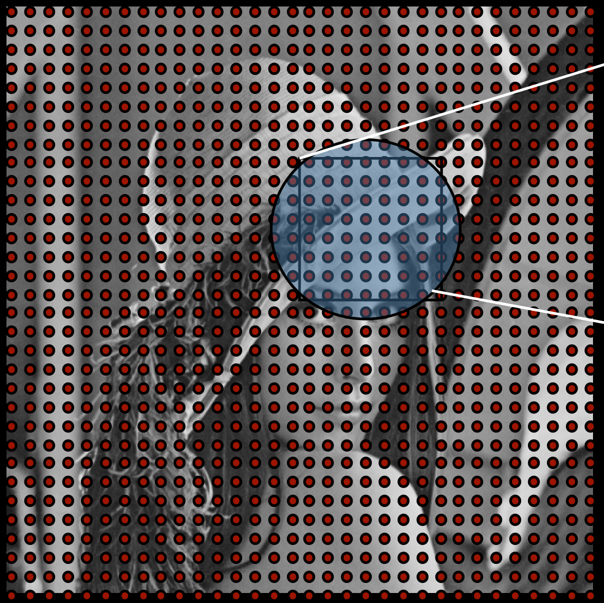
inverse DCT

$$s_{ij} = \frac{1}{4} \sum_{u=0}^{7} \sum_{v=0}^{7} C_u C_v S_{uv} \cos\frac{(2i+1)u\pi}{16} \cos\frac{(2j+1)v\pi}{16}$$

- Sμ,ν – coefficient corresponding to position (i, j)
- si,j – value of recovered image in position (i,j)
- Cμ, Cν – constants

Media compression - spatial transform, DCT

- applying the DCT to an 8 x8 image block



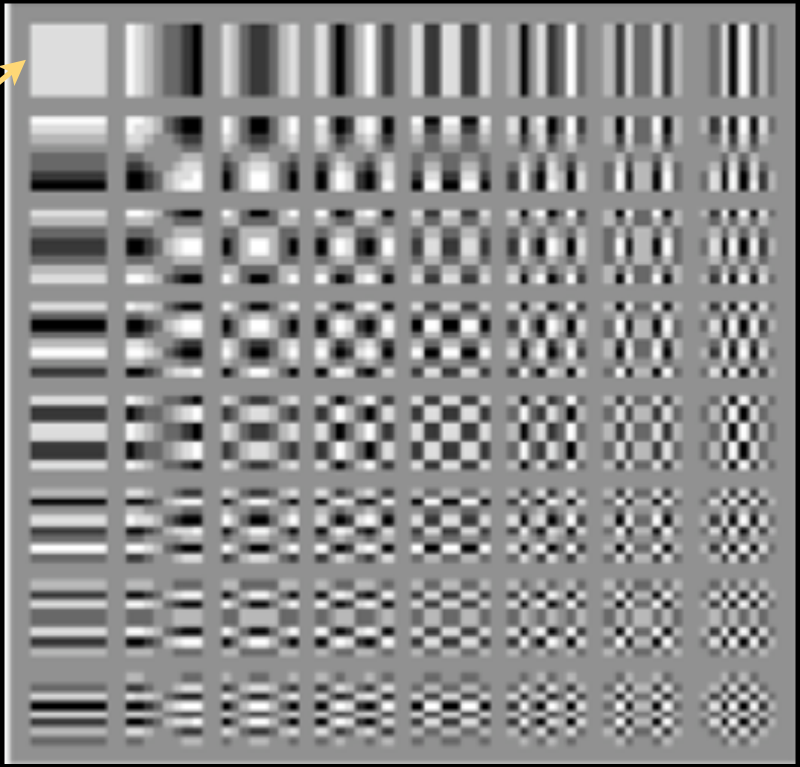
$S_{0,0}$ $S_{0,7}$

172	-18	15	-8	23	-9	-14	19
21	-34	24	-8	-10	11	14	7
-9	-8	-4	6	-5	4	3	-1
-10	6	-5	4	-4	4	2	1
-8	-2	-3	5	-3	3	5	6
4	-2	-4	6	-4	4	2	-1
4	-3	-4	5	6	3	1	1
0	-8	-4	3	2	1	4	0

$S_{7,0}$ $S_{7,7}$

Media compression - spatial transform, DCT

- basis of the DCT
  - each square block represents an image that would be obtained considering only the DCT coefficient corresponding to the block position and assuming all other coefficients have value zero



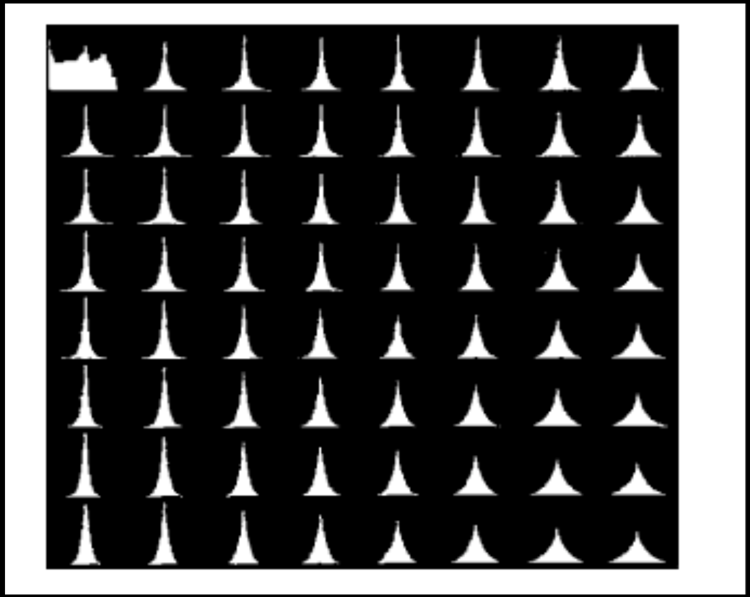
low frequencies

DC coefficient

high frequencies

Media compression - spatial transform, DCT

- energy distribution across DCT coefficients
  - amplitude histograms of an 8 x 8 DCT block of a natural image



  - the DC coefficient has a uniform distribution
  - all others have Laplace-like distribution

Media compression - spatial transform, DCT

- detail of a block and its relation to the number of retained DCT coefficients

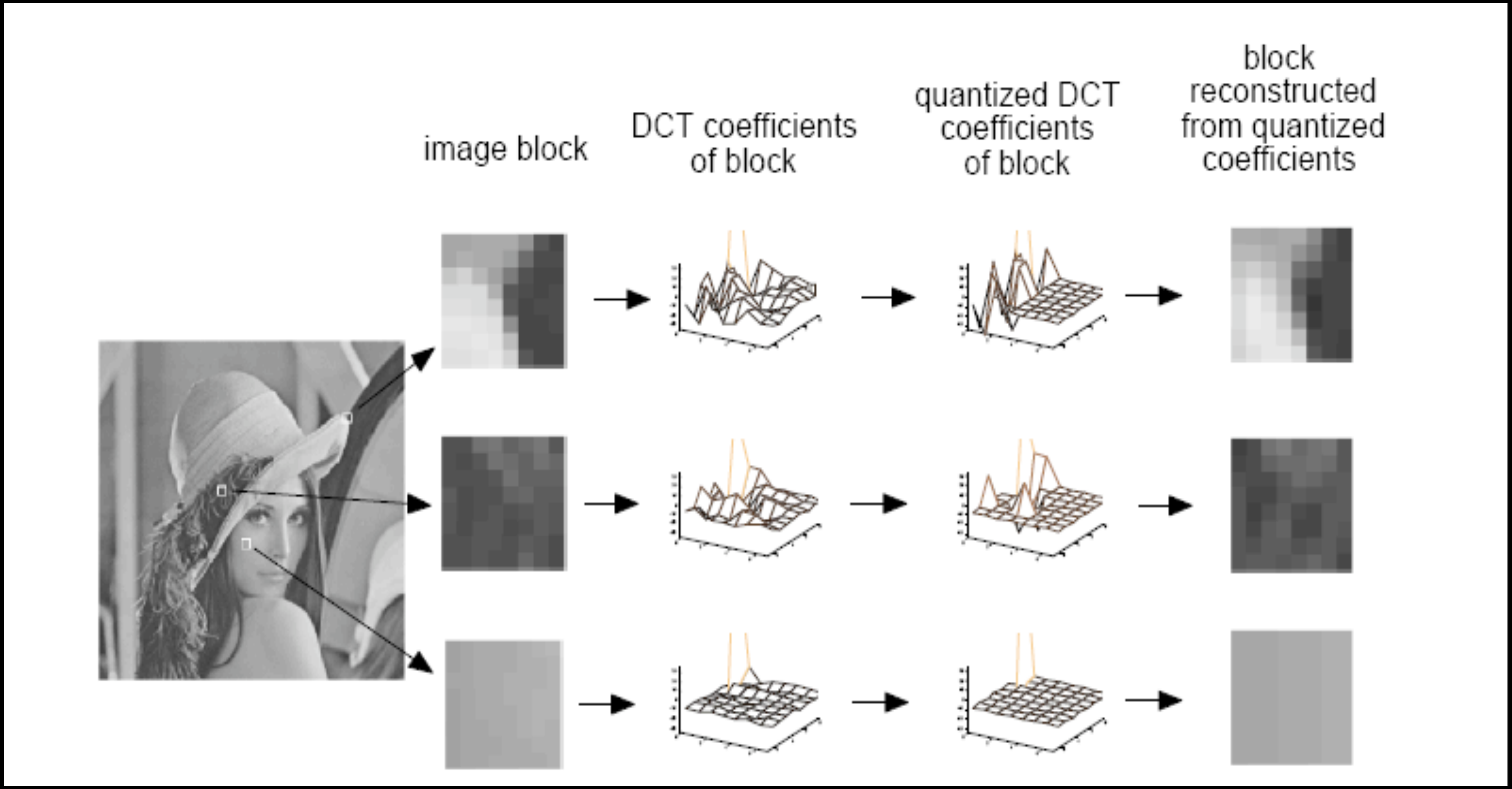


image block
DCT coefficients of block
quantized DCT coefficients of block
block reconstructed from quantized coefficients

Media compression - spatial transform, DCT

- efficiency of the DCT versus the size of the block
  - as the efficiency of the DCT depends on the type of image and as image varies in space, to better follow the local spatial statistics
    - it is used a strategy of variable size block
  - a larger block may
    - achieve better energy concentration and detail preservation in uniform regions where pixels are highly correlated
    - represent a larger area using less coefficients and hence greater compression rates
  - a smaller block may
    - be more convenient to represent areas with lots of spatial detail given that the correlation between points will be larger between points closer in space

Media compression - spatial transform, DCT

- advantages of DCT
  - coefficients exhibit generally a small correlation
    - it is thus possible to eliminate redundancy using simple techniques
      - a weighted distribution is made of the available quantization levels
      - coefficients that stand below a given threshold are eliminated
  - the bi-dimensional DCT is separable, i. e., it can be decomposed in two one-dimensional transforms (one in the vertical axe and the other one in the horizontal)
    - easy to implement, reduced complexity

Media compression - spatial transform

- image-based transforms
  - decomposition of the image in the frequency domain
  - explore a hierarchical relation and "self-similarity" of the different frequency bands in the same spatial localization
    - sub-band coding
    - Discrete Wavelet Transform, DWT
    - "Embedded Zerotree Wavelet", EZW
    - "Set Partitioning in Hierarchical Trees", SPIHT

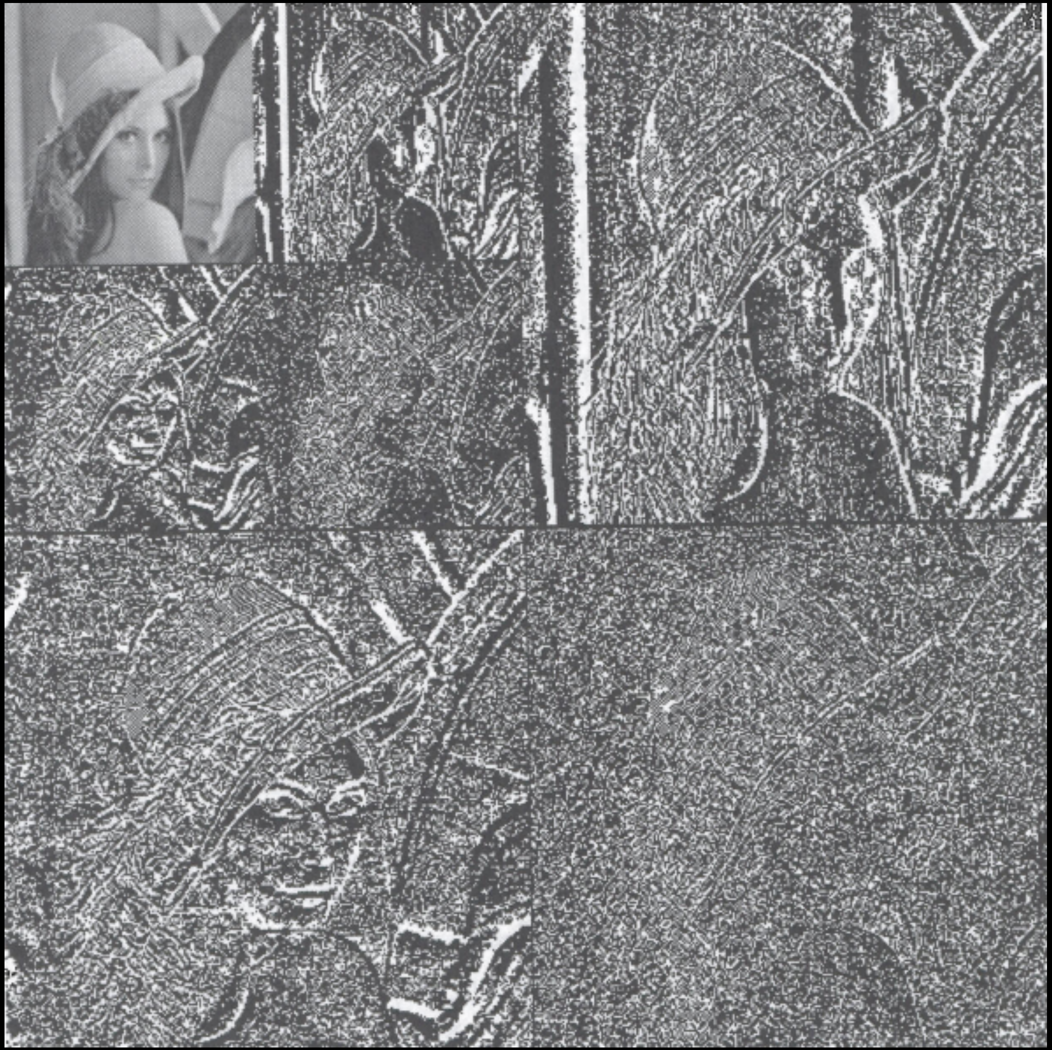
Media compression - spatial transform, DWT

- Discrete Wavelet Transform, DWT
  - based on the Fourier analysis, can be seen as a linear expansion of a series of continuous sine and cosine functions
  - it offers a very good resolution in the frequency domain
    - but it expands from $-\infty$ to $+\infty$, thus requiring infinite time and buffers to be calculated
      - practical implementations wont have full resolution in time
  - it is suitable to encode multi-resolution pictures
    - appropriate to the progressive transmission (images on the Web) or to video transmission in noisy/lossy environments
    - delivers different levels of the encoded signal with quality/resolution progressively better
      - each level can be used independently

Media compression - spatial transform, DWT

- transmission in lossy or variable bandwidth networks
  - if the instantaneous bandwidth is not sufficient to transmit the full resolution signal
    - only a resolution corresponding to the lower frequency is sent
    - when conditions change, resolutions corresponding to higher frequencies may be sent
  - the decoder may recover a low resolution signal using only the low frequency resolution signal
    - progressively it may improve the recovered signal resolution by incorporating detail from the higher frequencies

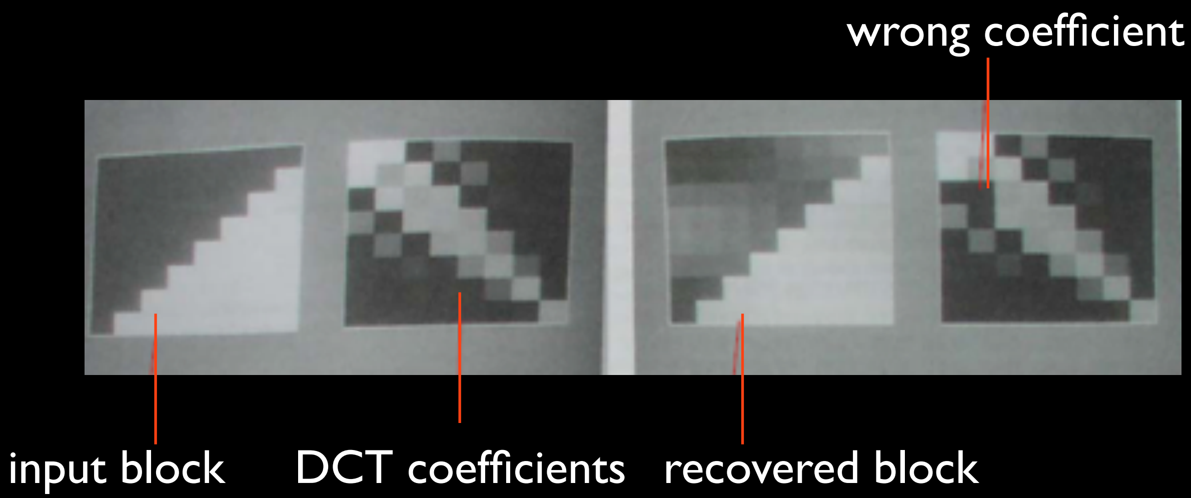
Media compression - spatial transform, DWT



- it is possible to extract a full resolution image; a 1/4th resolution image; a 1/16th resolution image; ...

Media compression - quantization

- DCT coefficient are quantized before being coded for transmission
  - quantization introduces an error on each coefficient
  - in the decoder, blocks are recovered by applying the inverse transform IDCT
- the effect of the error translates into extra frequency components and consequent visible artifacts



Media compression - quantization

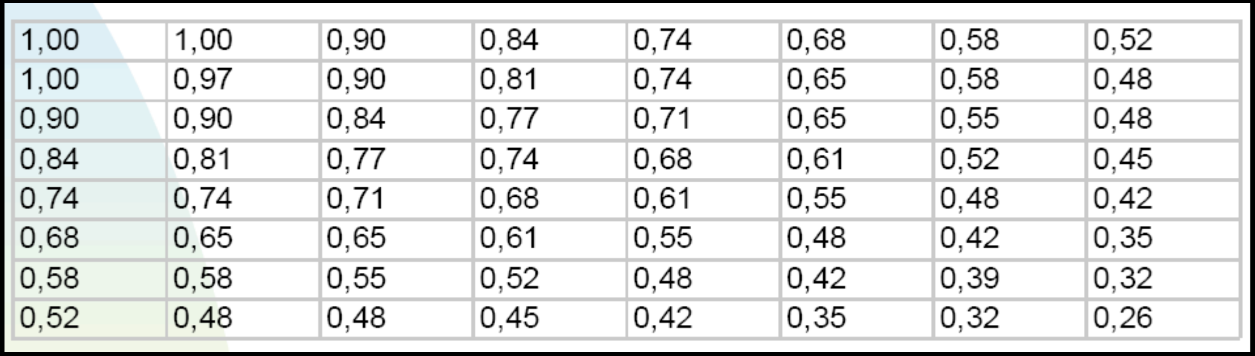
- quantization is done trying to reduce distortion and taking advantage of the
  - characteristics of the HVS
  - energy compacting properties of the DCT
- it adopts a variable strategy by which a larger weight is assigned to the lower frequencies
  - introduces more error in the high frequencies
  - to which our eyes are less sensitive
  - the gain (bit economy) is used to provide more resolution (more bits) to the low-frequencies
    - which contain the major part of the signal energy through the DCT!

Media compression - quantization

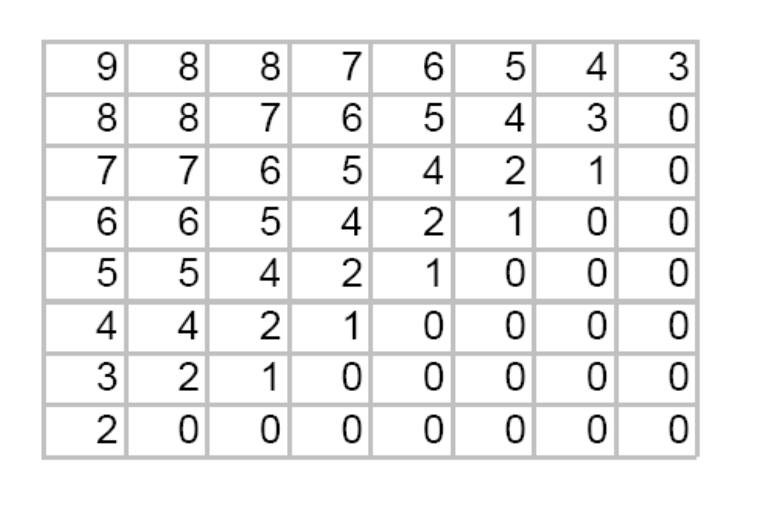
- each DCT coefficient is multiplied by a weight w(x,y) before quantization
    - $C(\mu,\nu) = w(x,y).c(x,y)$
  - the error in each coefficient is inversely proportional to the corresponding value of w(x,y)
  - in addition to this weight distribution it is necessary to decide how many bits to assign to each coefficient
    - coefficients with smaller weight should receive less bits

Media compression - quantization

- example of weight and bit allocation tables



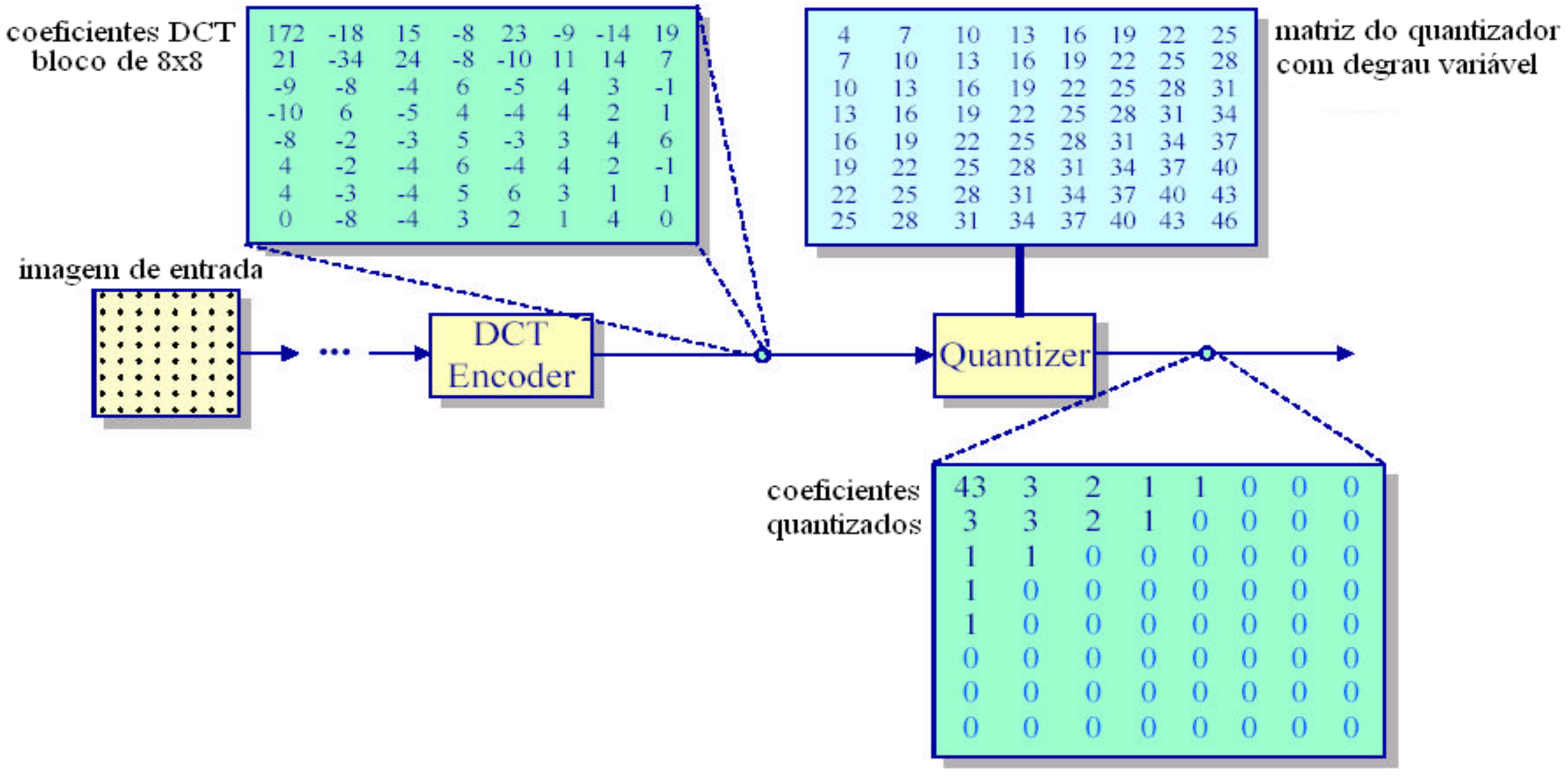
1,00	1,00	0,90	0,84	0,74	0,68	0,58	0,52
1,00	0,97	0,90	0,81	0,74	0,65	0,58	0,48
0,90	0,90	0,84	0,77	0,71	0,65	0,55	0,48
0,84	0,81	0,77	0,74	0,68	0,61	0,52	0,45
0,74	0,74	0,71	0,68	0,61	0,55	0,48	0,42
0,68	0,65	0,65	0,61	0,55	0,48	0,42	0,35
0,58	0,58	0,55	0,52	0,48	0,42	0,39	0,32
0,52	0,48	0,48	0,45	0,42	0,35	0,32	0,26



9	8	8	7	6	5	4	3
8	8	7	6	5	4	3	0
7	7	6	5	4	2	1	0
6	6	5	4	2	1	0	0
5	5	4	2	1	0	0	0
4	4	2	1	0	0	0	0
3	2	1	0	0	0	0	0
2	0	0	0	0	0	0	0

Media compression - quantization

- example of DCT ad quantization



coeficientes DCT
bloco de 8x8

172	-18	15	-8	23	-9	-14	19
21	-34	24	-8	-10	11	14	7
-9	-8	-4	6	-5	4	3	-1
-10	6	-5	4	-4	4	2	1
-8	-2	-3	5	-3	3	4	6
4	-2	-4	6	-4	4	2	-1
4	-3	-4	5	6	3	1	1
0	-8	-4	3	2	1	4	0

matriz do quantizador
com degrau variável

4	7	10	13	16	19	22	25
7	10	13	16	19	22	25	28
10	13	16	19	22	25	28	31
13	16	19	22	25	28	31	34
16	19	22	25	28	31	34	37
19	22	25	28	31	34	37	40
22	25	28	31	34	37	40	43
25	28	31	34	37	40	43	46

coeficientes
quantizados

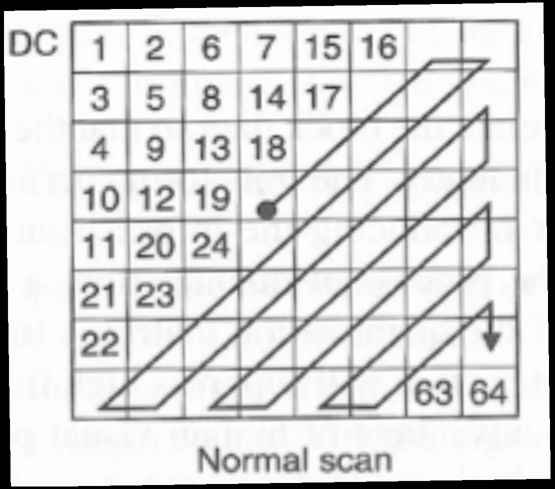
43	3	2	1	1	0	0	0
3	3	2	1	0	0	0	0
1	1	0	0	0	0	0	0
1	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Media compression - quantization

- relation with the content
  - generally video sequences have different characteristics
    - between images or even between different areas of the same image
    - a unique solution is not very efficient
  - different weight and bit allocation strategies can be used where more appropriate
    - that’s why the quantizer step size is usually a configurable parameter in encoders
    - enables to control the level of distortion being thus a central parameter in the overall “rate-distortion” strategy

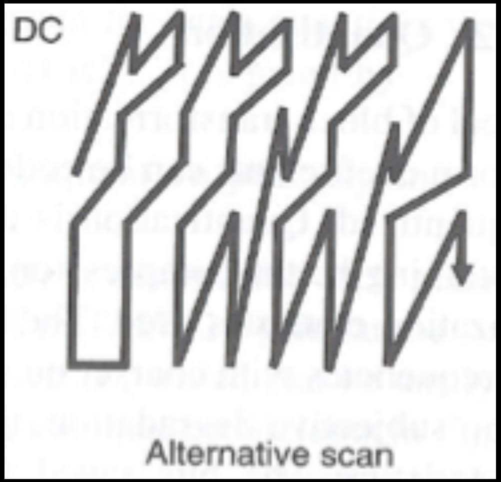
Media compression - entropy coding

- eliminates statistical redundancy
- how are transmitted the quantized coefficients?
  - using entropy encoding with run-length and variable length codes (VLC)
    - to increase the efficiency of VLC, coefficients are zig-zagged scanned



43	3	2	1	1	0	0	0
3	3	2	1	0	0	0	0
1	1	0	0	0	0	0	0
1	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

quantized coefficients



    - the number of consecutive equal values increases

Media compression - entropy coding

- Variable Length Coding (VLC)
- A simple way to code symbols with lossless compression is acieved by applying "Run-Length" codes
- whenever neighbor symbols have the same value, instead of repeatedly sending the same value, only the number of times that value has occurred together with that value (one single time) is sent
- data that repeats frequently are good candidates for this kibd of compression

Media compression - entropy coding

- Run Length Coding, RLC
- basic mode of operation
  - data are analyzed to detect successions of repeated symbols
  - repetitive symbols are replaced by a unique representative symbol followed by a special character and the number of times that symbol occurred
  - example:

"00 00 00 00 F0 F4 54 9F FF FF 45 45 45 45 45" →

→ "00 *3 F0 F4 54 9F FF FF 45 *4"

Media compression - entropy coding

- variable length coding
- the processed stream (fixed length) can be represented as a random variable with a probability of occurrence
- simple example: image *r* with 4 grey levels

symbol	probability	fixed length code	n1	variable length code	n2
r1	0.1875	00	2	11	2
r2	0.5	01	2	0	1
r3	0.1250	10	2	101	3
r4	0.1875	11	2	111	3

- average number of bits in fixed code = 2
- average number of bits in variable code = 3 x 0,1875 + 1 x 0,5 + 3 x 0,1250 + 2 x 0,1875 = 1,8125
- compression ration = 2 / 1,8125 = 1,103

Media compression - entropy coding

- entropy coding
- Shannon-Fano codes (SFC) are theoretically the most efficient
  - but not always lead to good results as they depend on the sets of symbols probability
  - and require a-priori knowledge of the symbol probabilities
- other codes overcome these problems
  - Huffman codes
  - arithmetic coding
  - Ziv-Lemple
  - range coding

Media compression - entropy coding

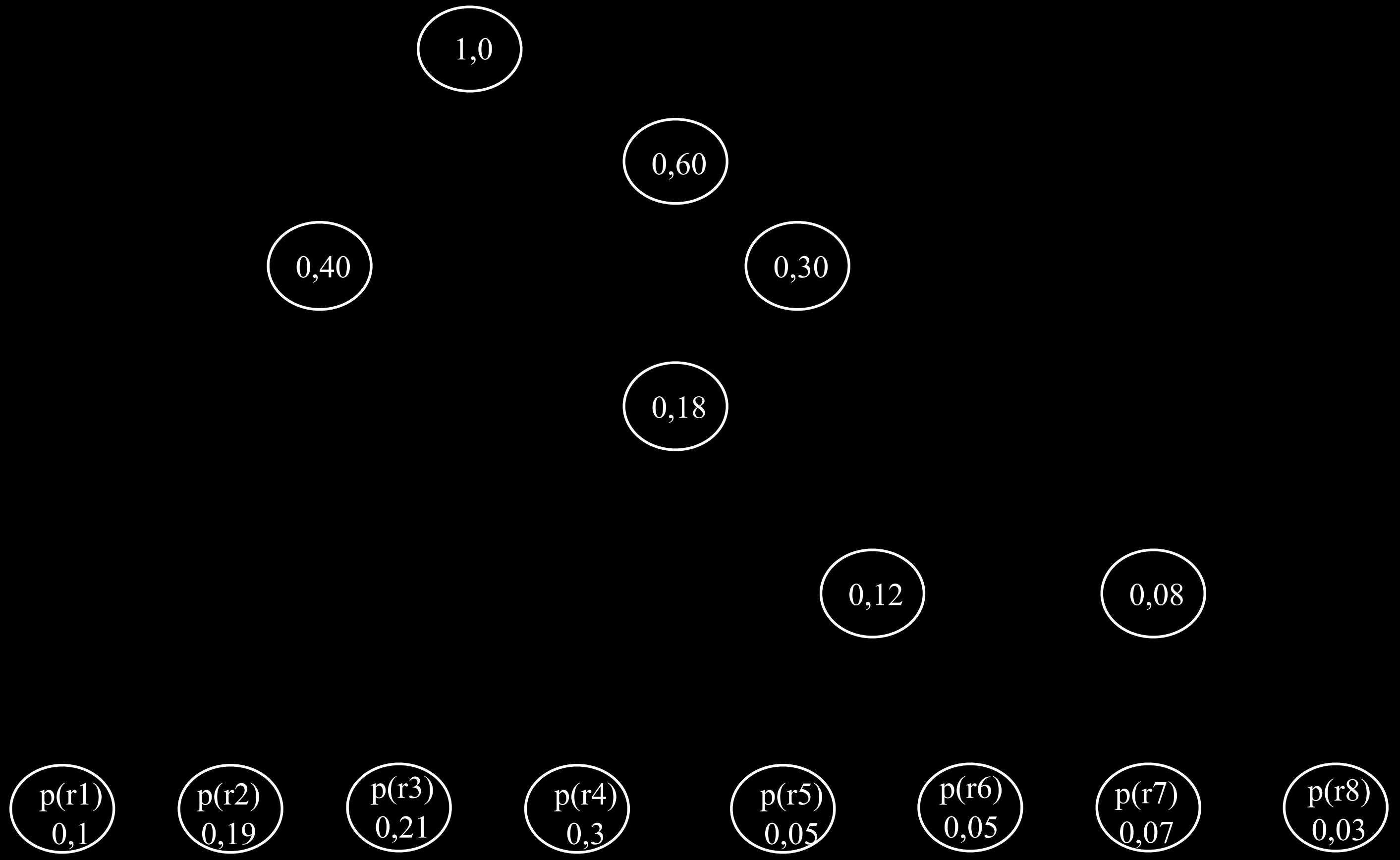
- Huffman codes
  - use a VLC code table obtained based on an estimative of the symbols probability of occurrence
  - uses a method to select the code assigned to each symbol guaranteeing that each code is "prefix-free"
    - no code is the prefix of another code!
  - the basic method consists in creating a binary tree, ordered according to the frequency of occurrence of symbols
    - the tree is created "bottom-up" and not "top-down" as usual

Media compression - entropy coding

- basic technique to create the Huffman binary tree
  - each symbol and associated weight (probability of occurrence) constitute the leaves of the tree (external nodes)
  - from bottom-up, nodes are associated in pairs creating a new node
    - the new node has a weight which is the sum of the weights of the 2 less probable nodes
  - superior branches are marked with “1”, others with “0”
  - the process continues until one single node remains

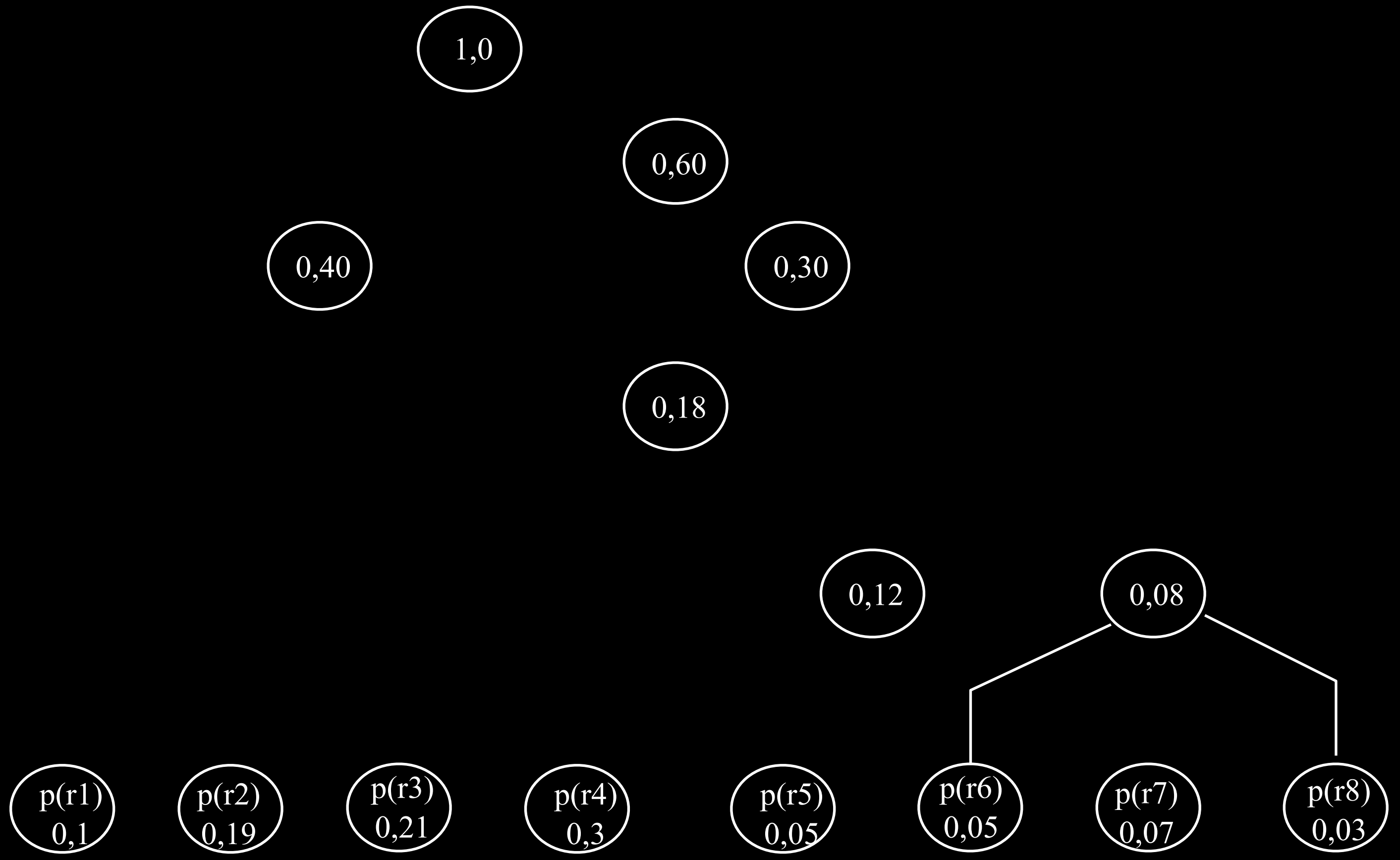
Media compression - entropy coding

- Huffman binary tree



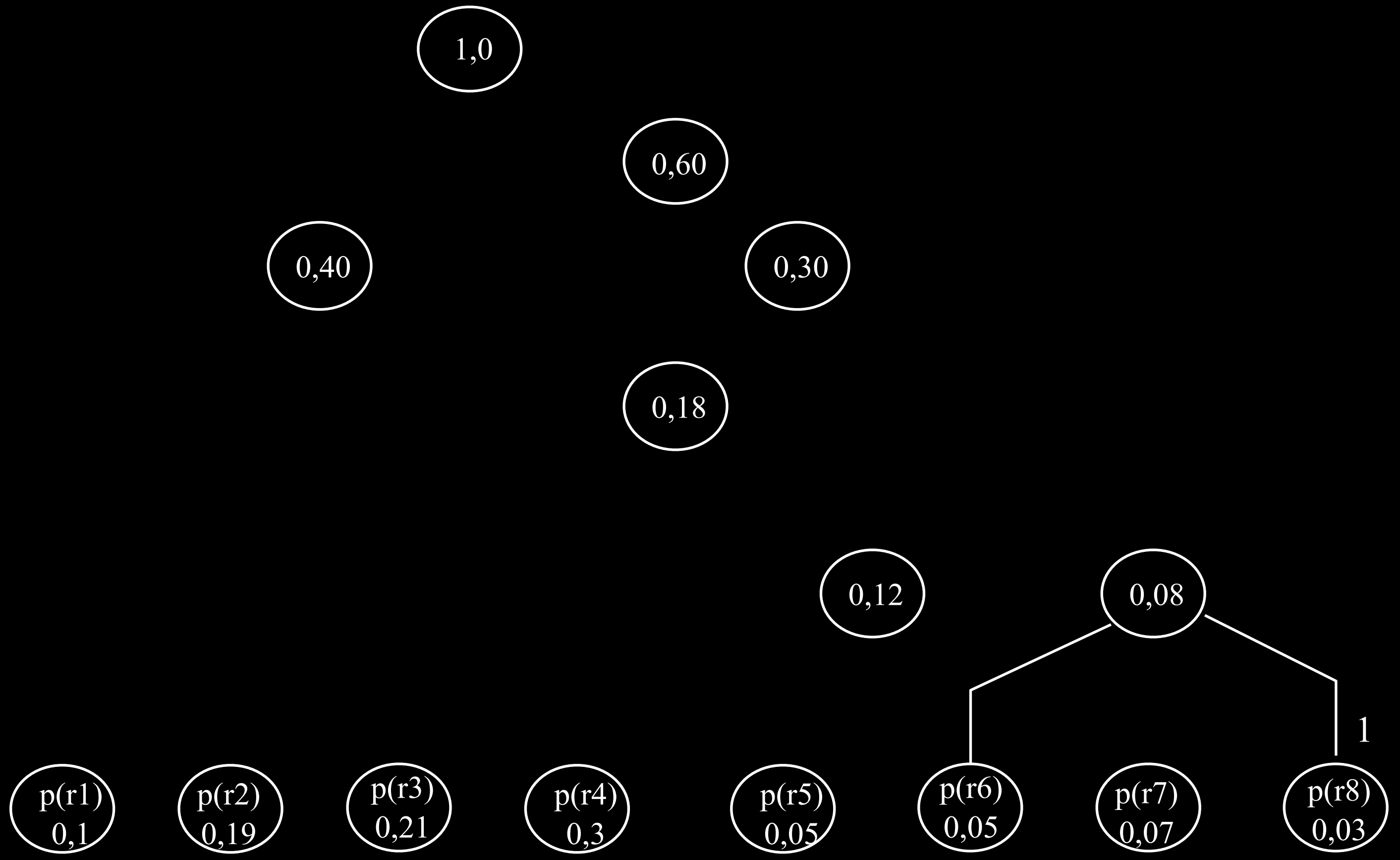
Media compression - entropy coding

- Huffman binary tree



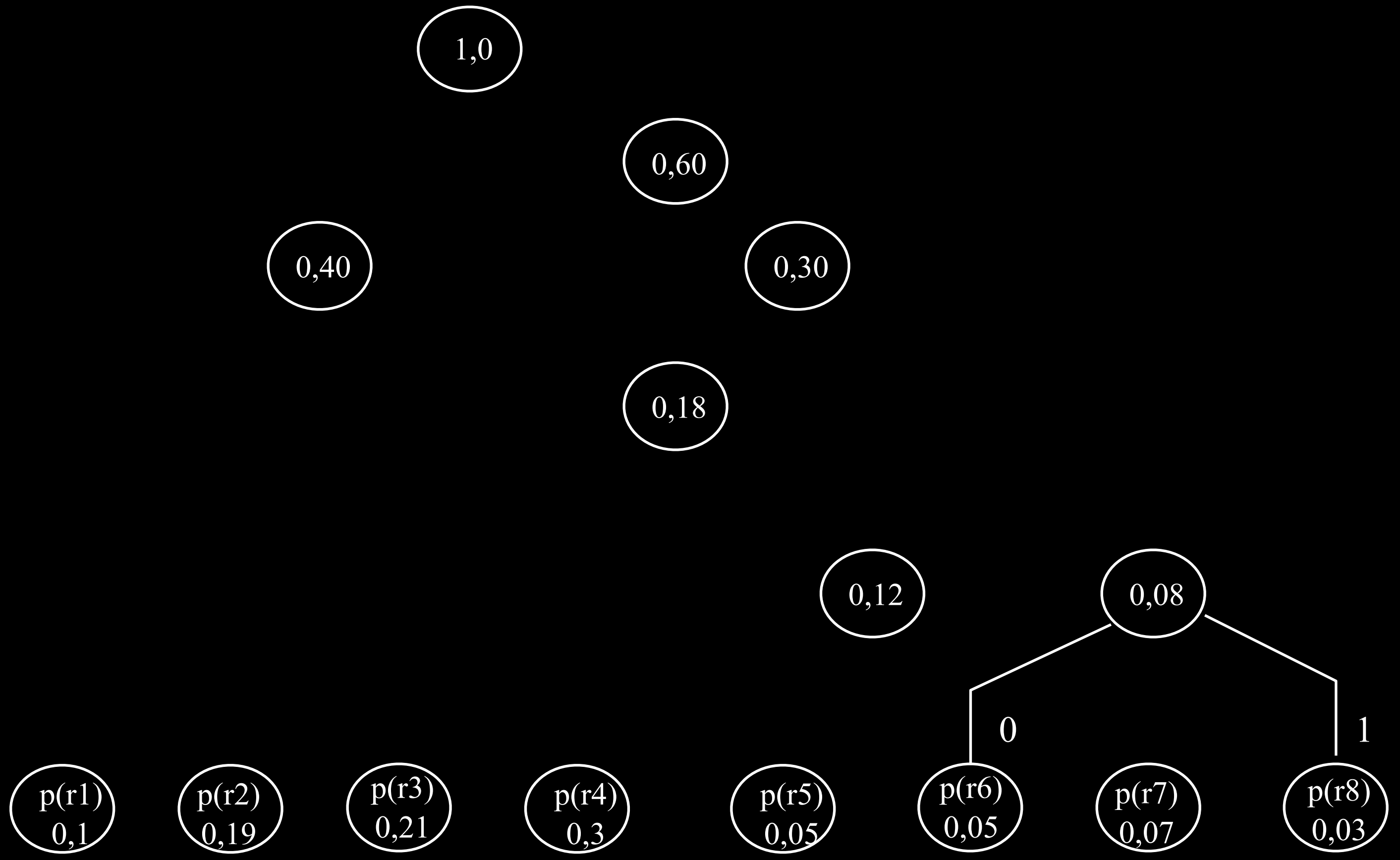
Media compression - entropy coding

- Huffman binary tree



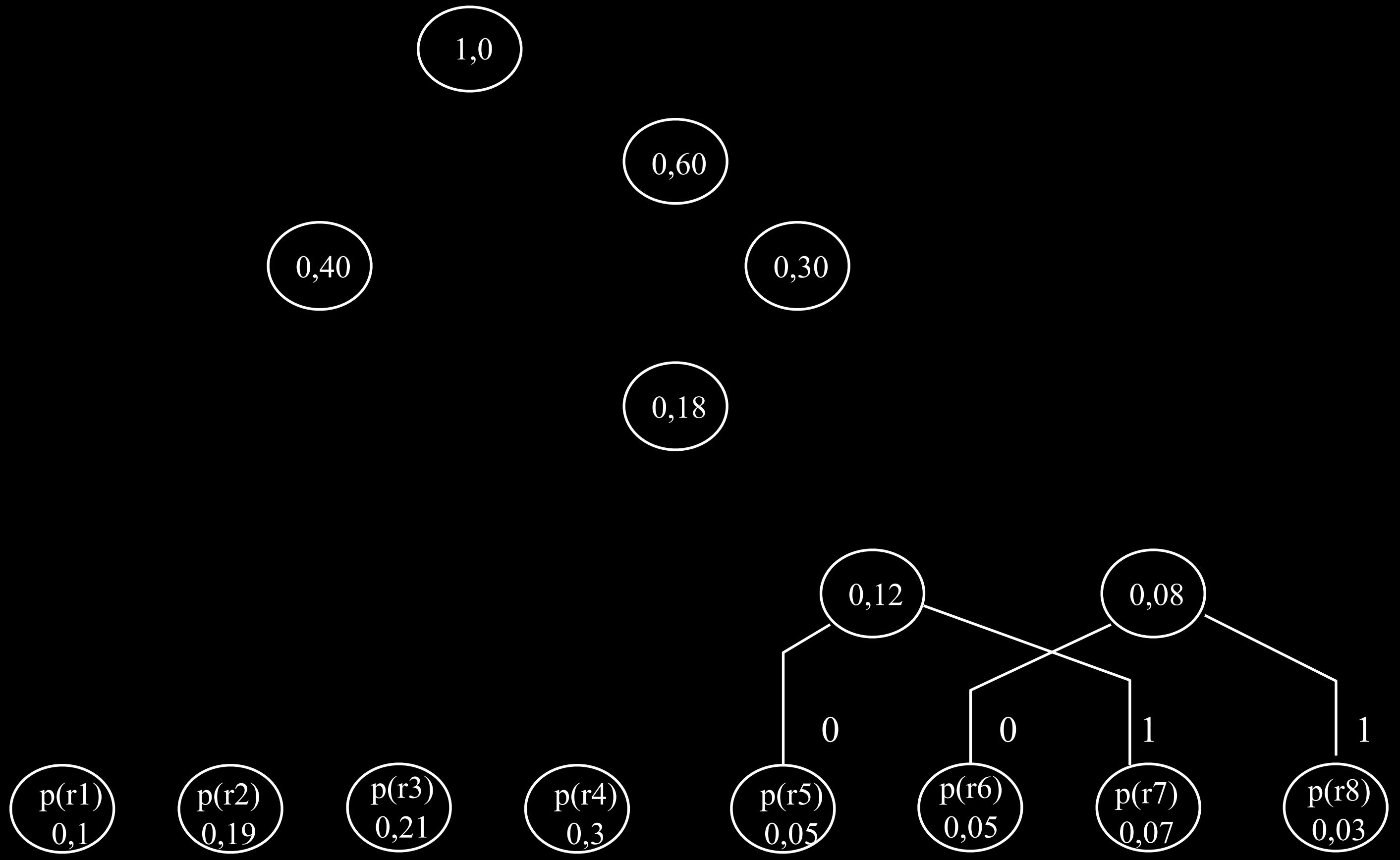
Media compression - entropy coding

- Huffman binary tree



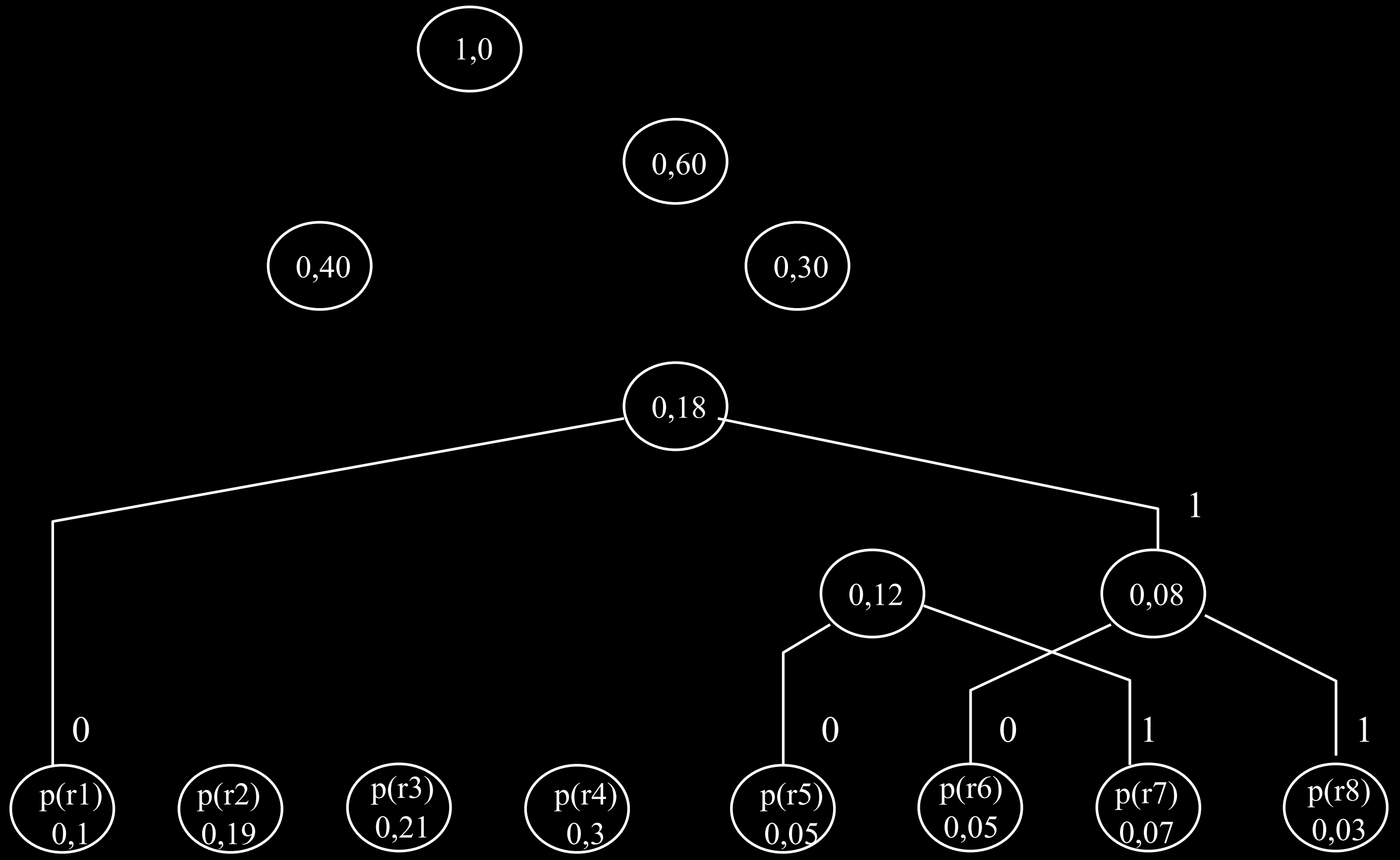
Media compression - entropy coding

- Huffman binary tree



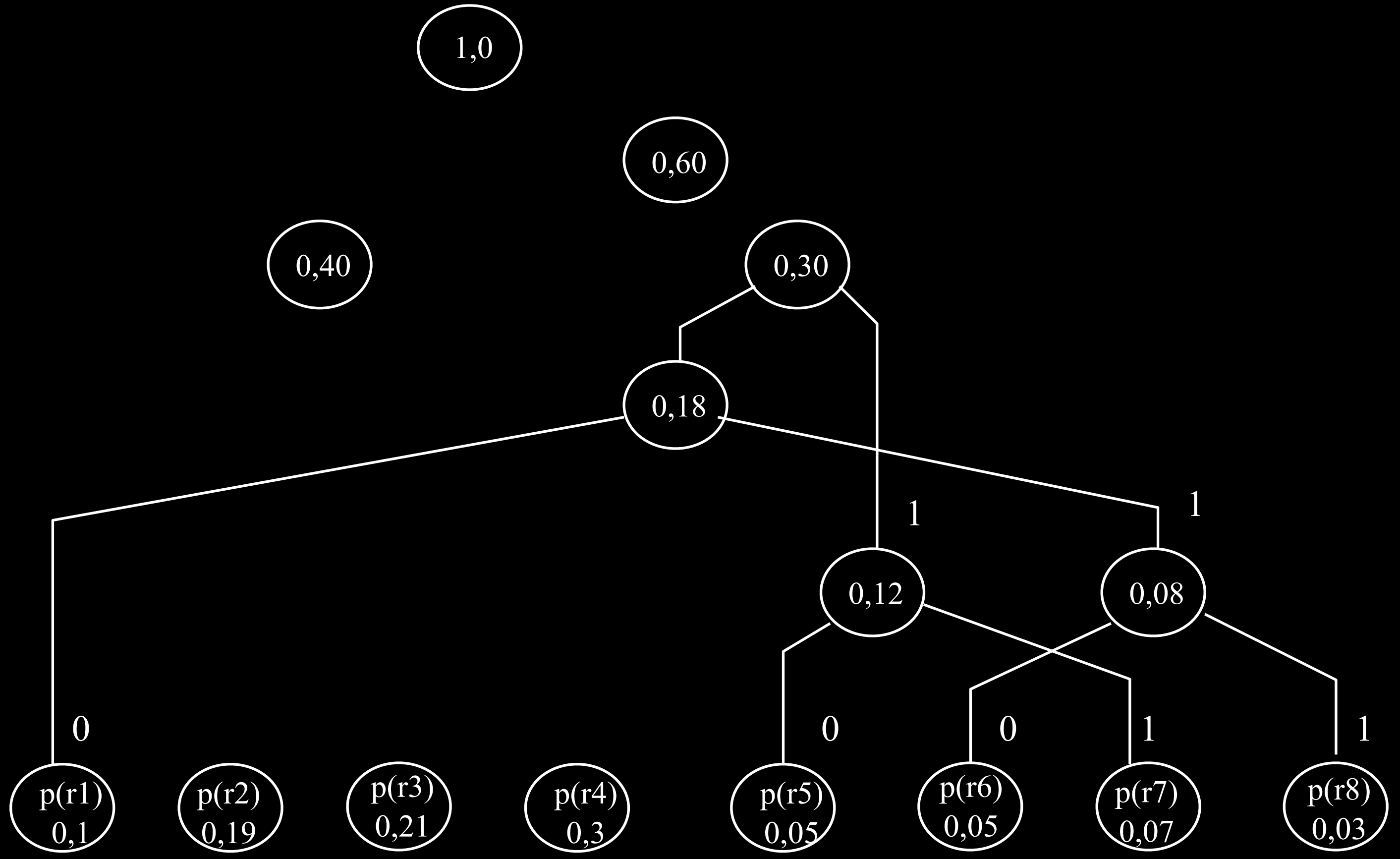
Media compression - entropy coding

- Huffman binary tree



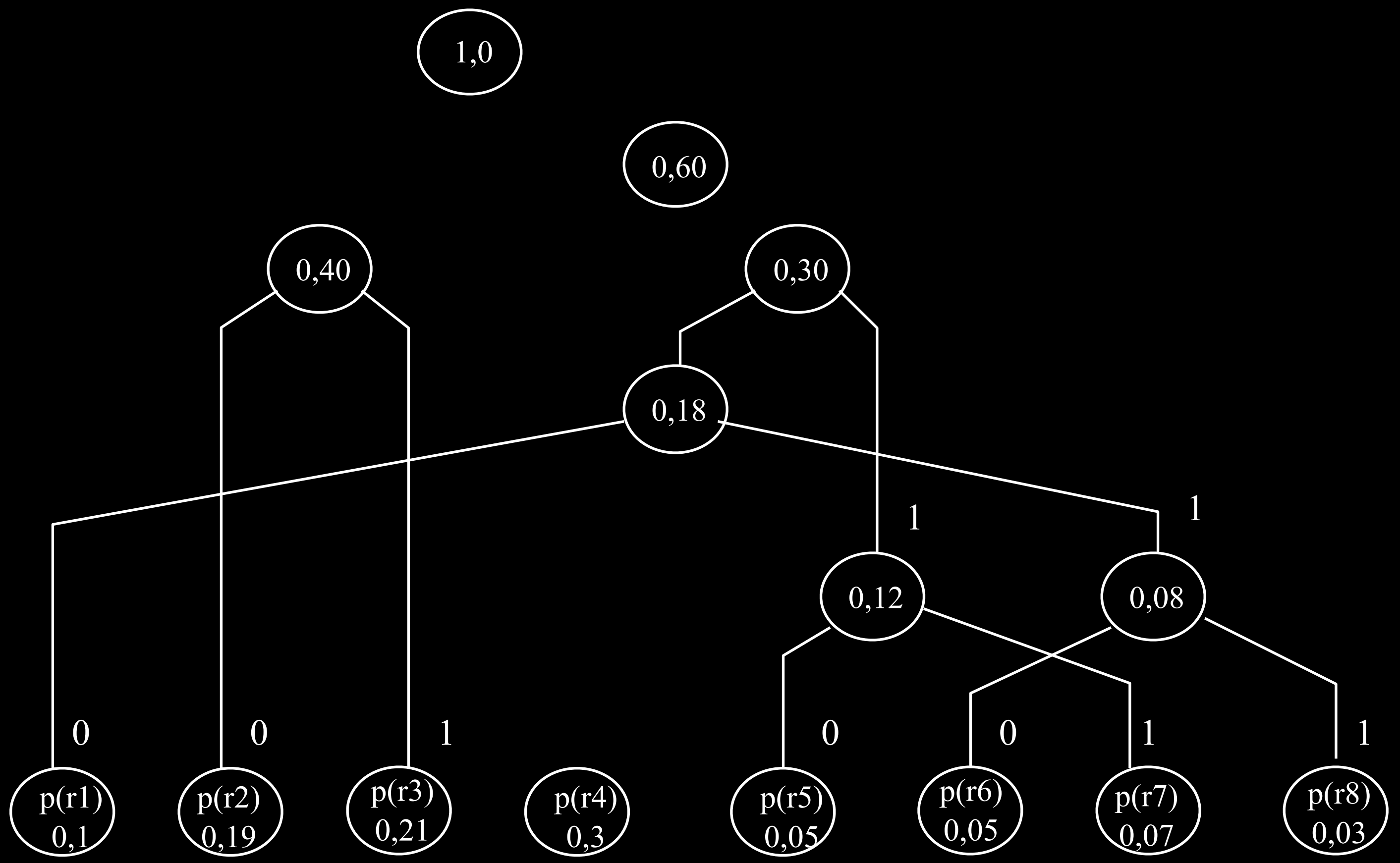
Media compression - entropy coding

- Huffman binary tree



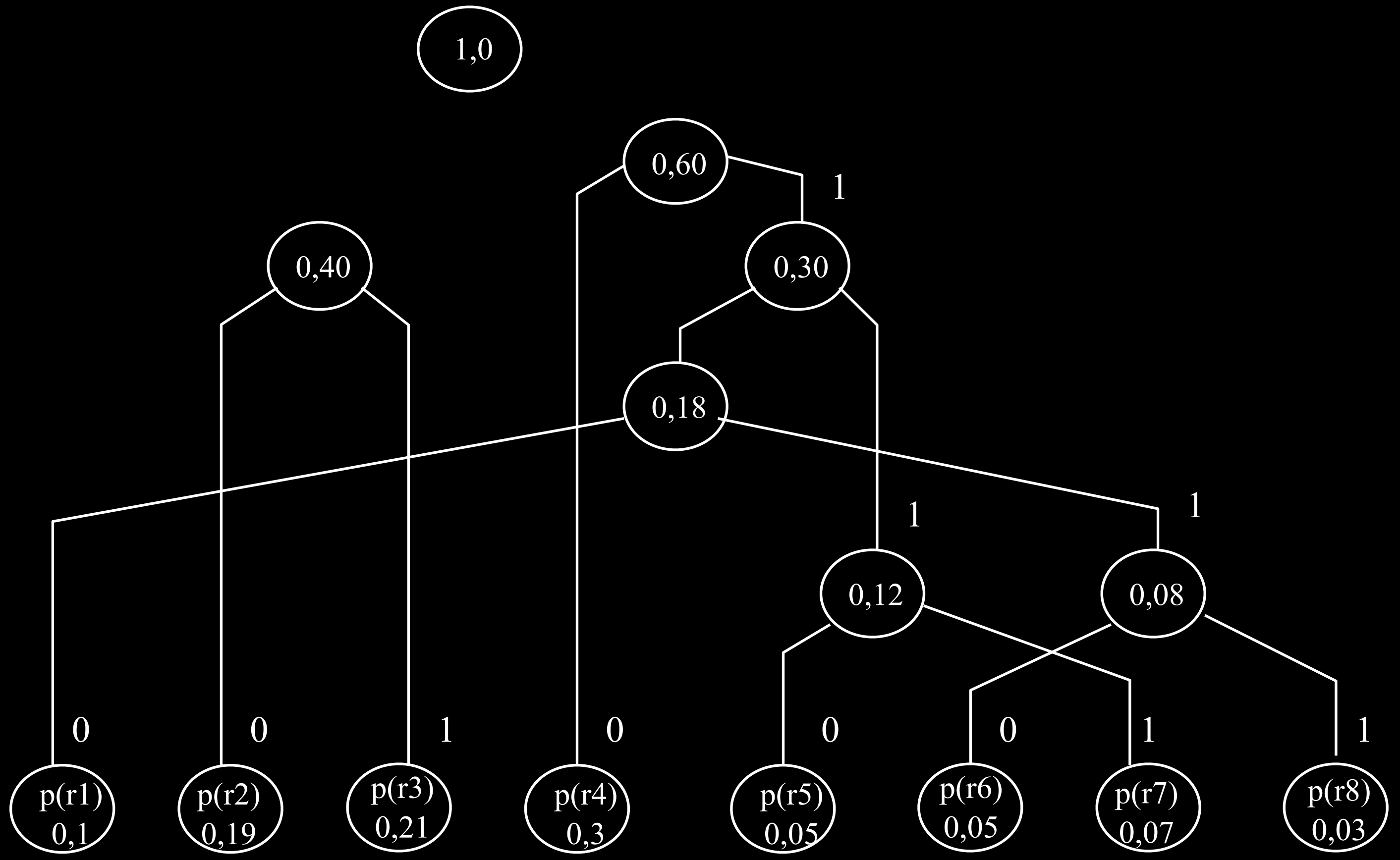
Media compression - entropy coding

- Huffman binary tree



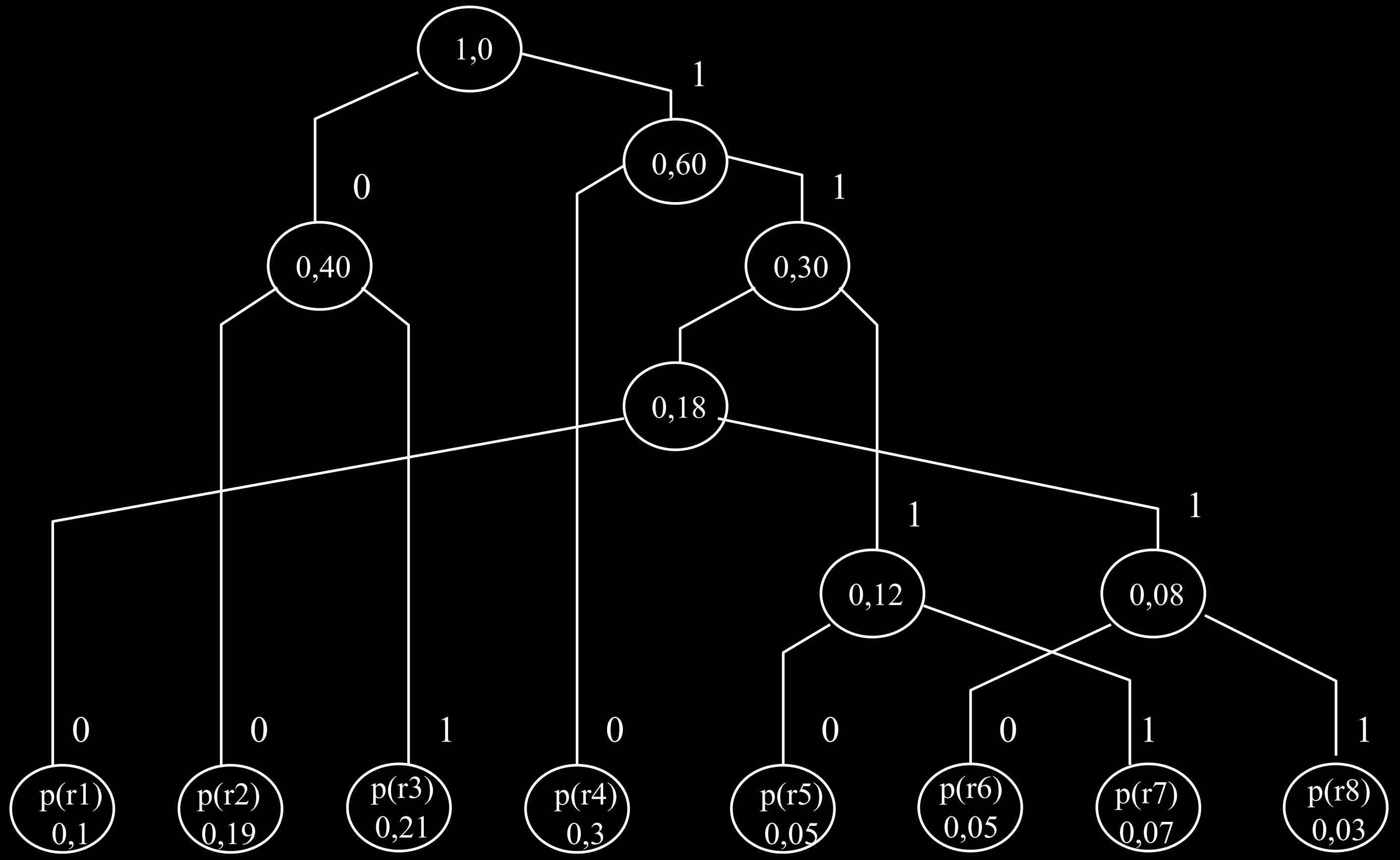
Media compression - entropy coding

- Huffman binary tree



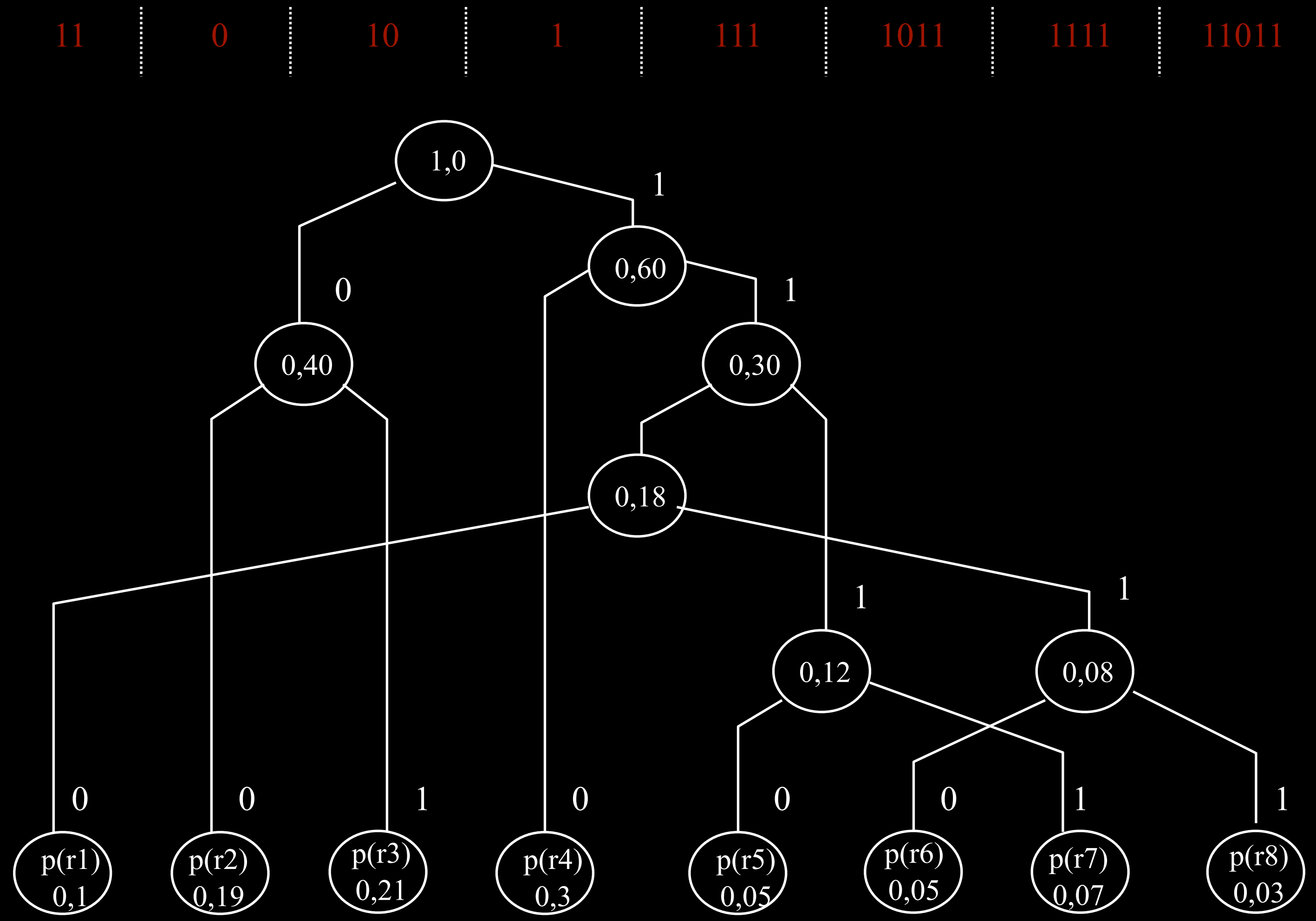
Media compression - entropy coding

- Huffman binary tree



Media compression - entropy coding

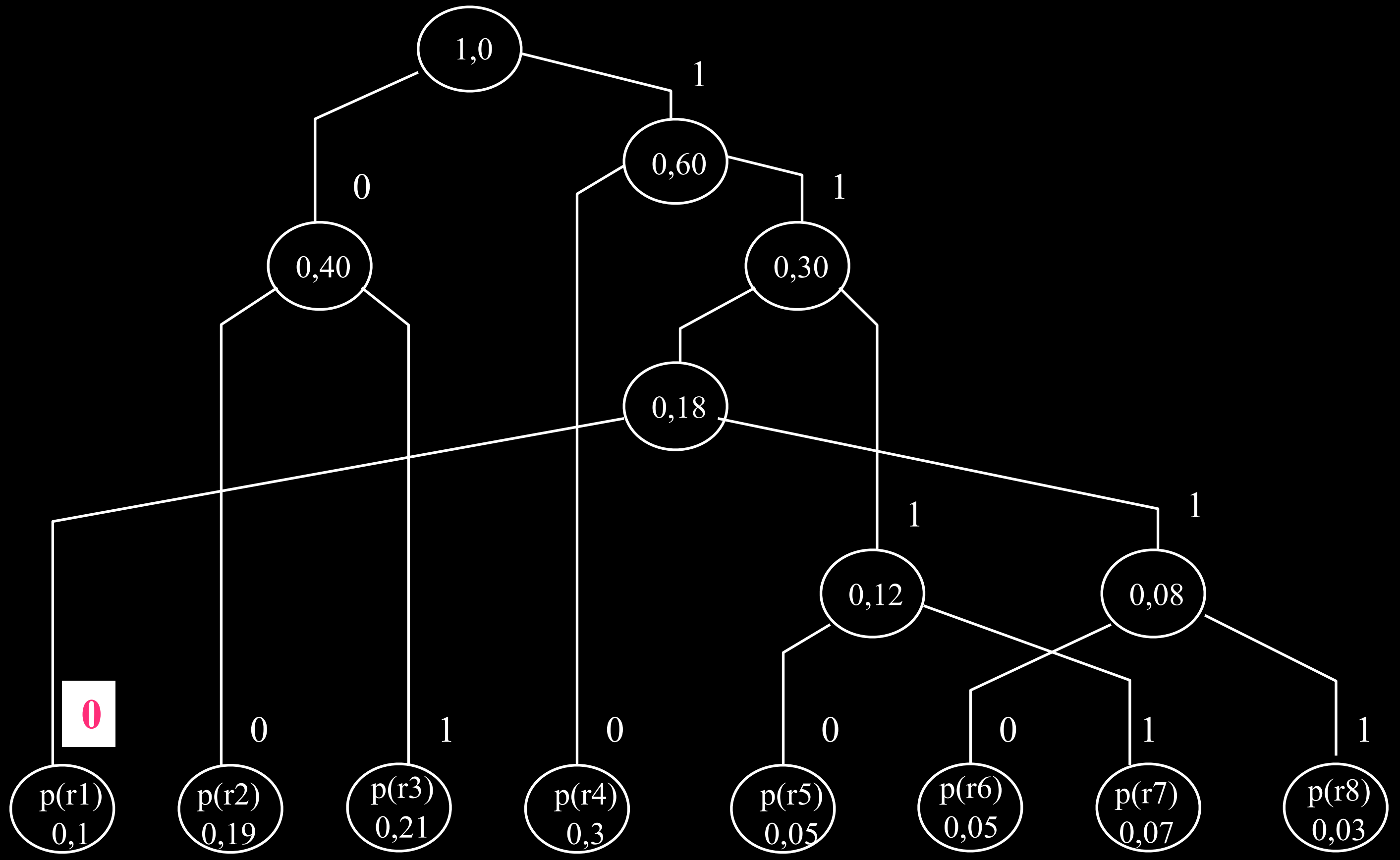
- Huffman binary tree



Media compression - entropy coding

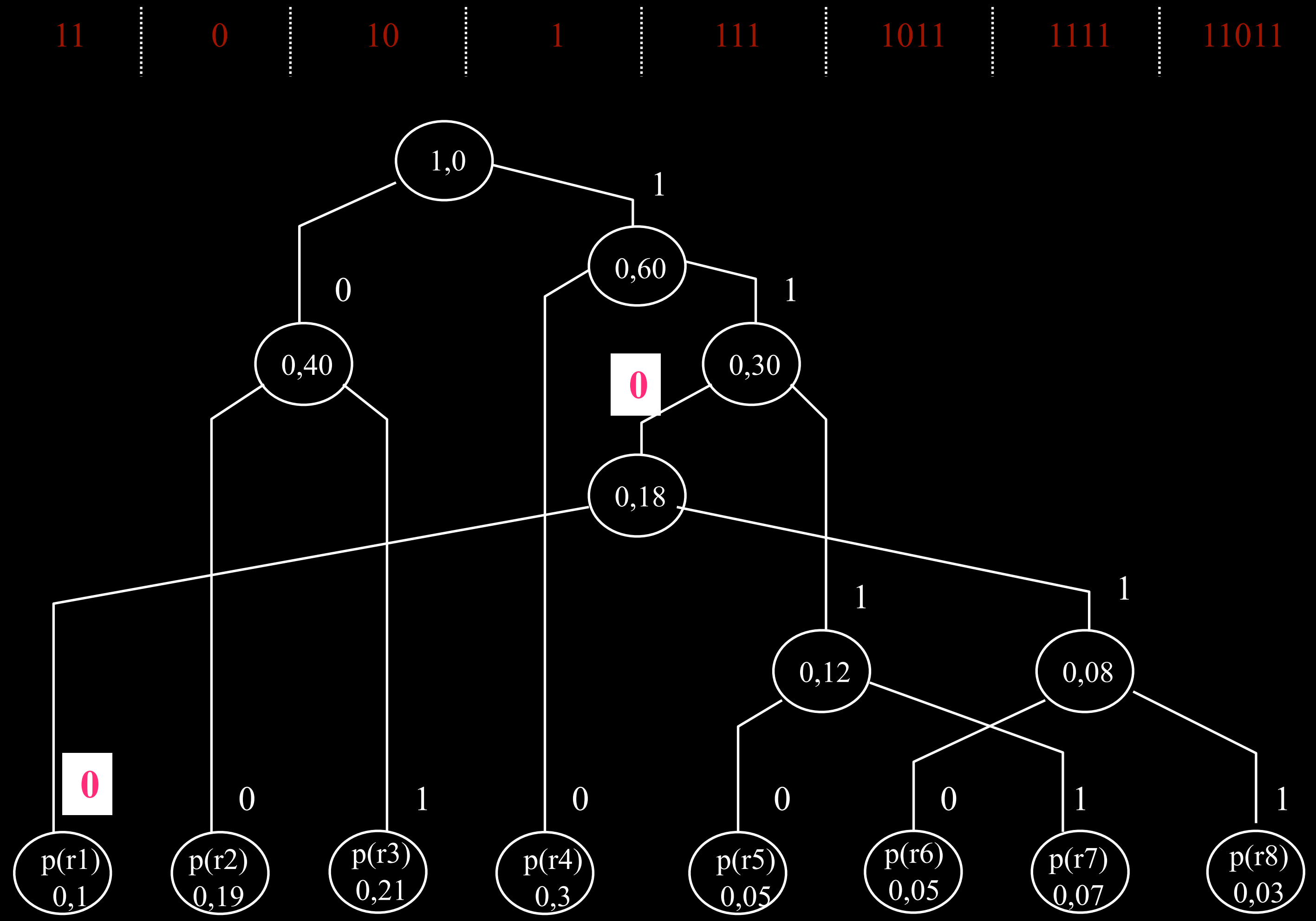
- Huffman binary tree

11 ┊ 0 ┊ 10 ┊ 1 ┊ 111 ┊ 1011 ┊ 1111 ┊ 11011



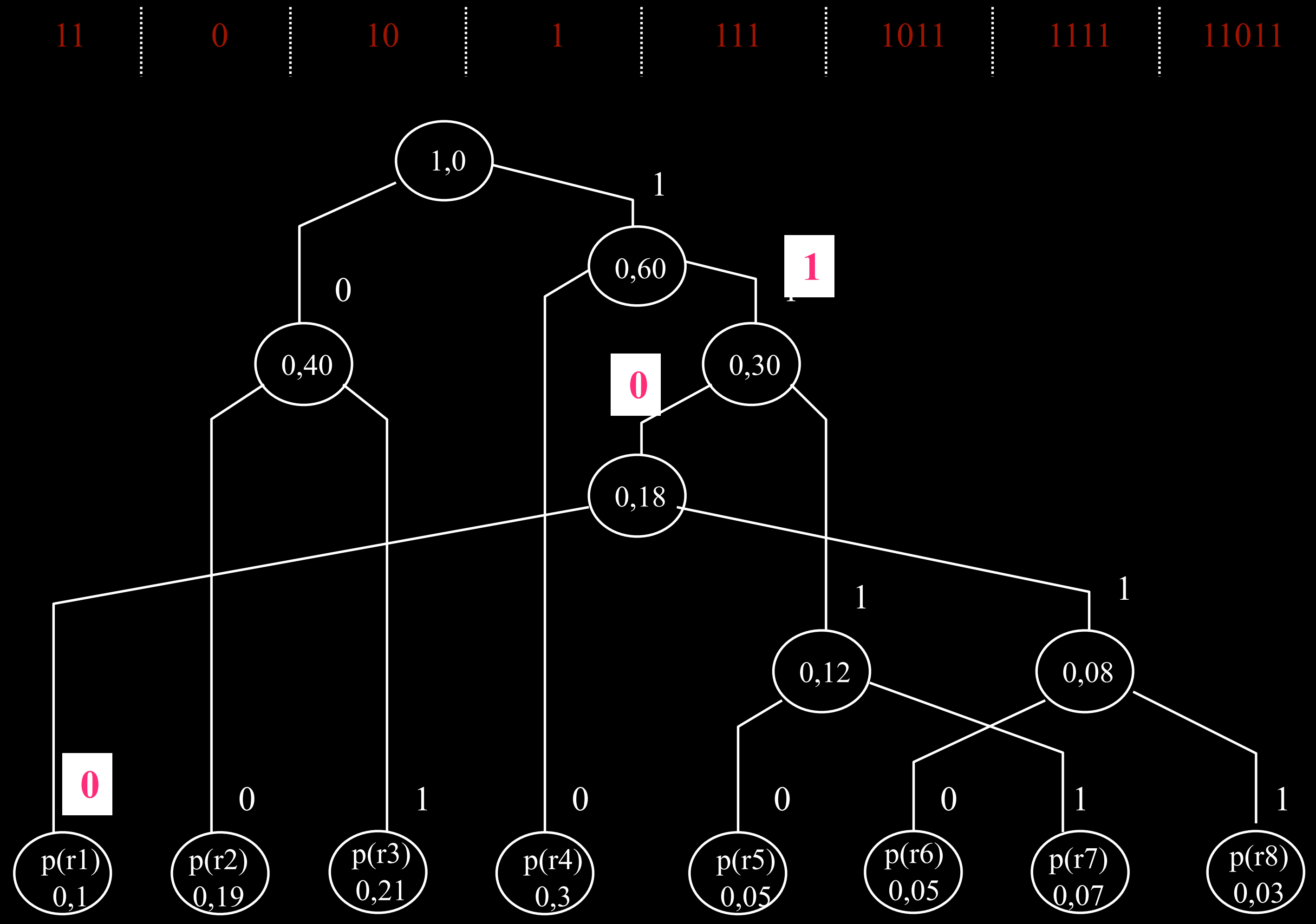
Media compression - entropy coding

- Huffman binary tree



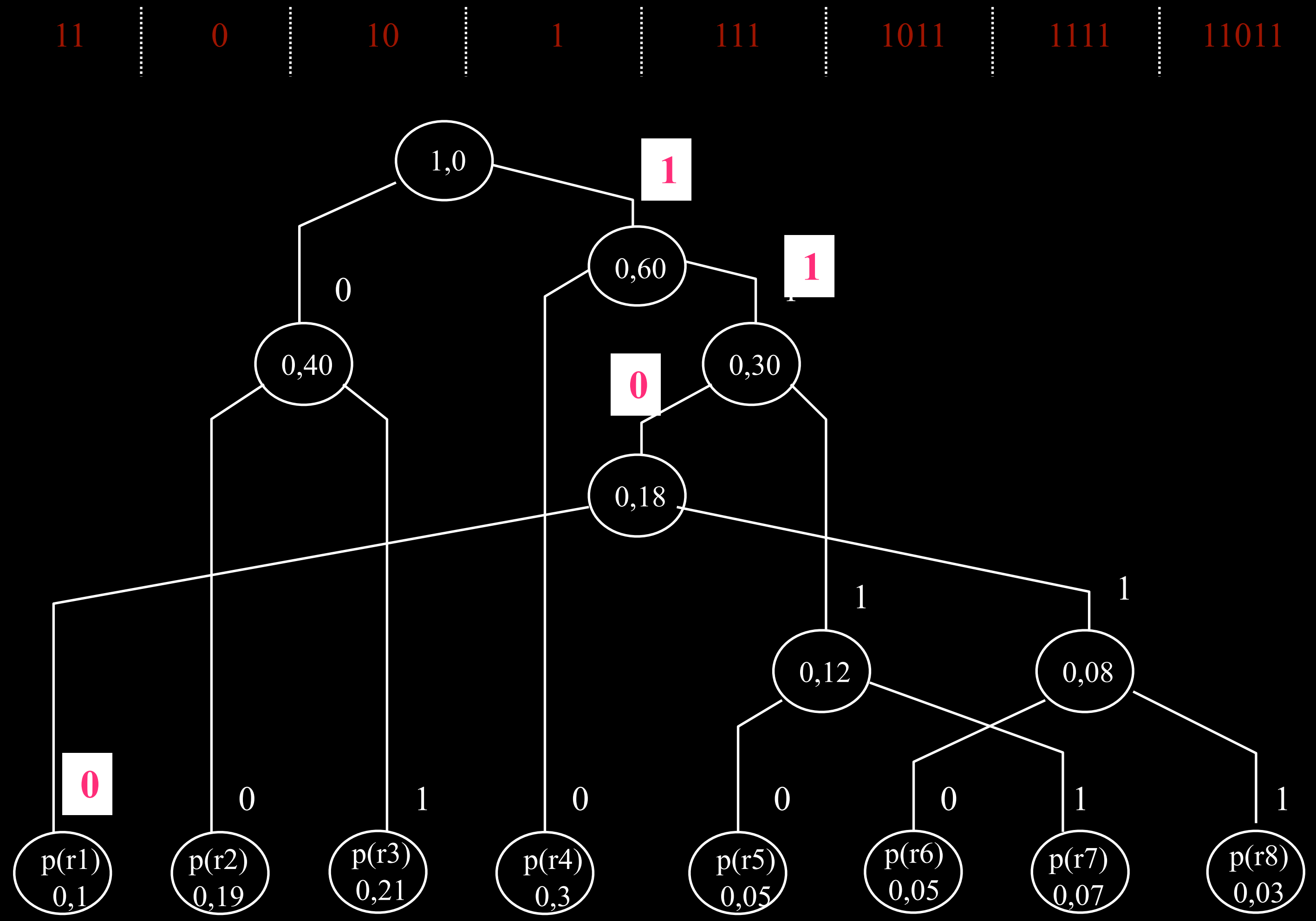
Media compression - entropy coding

- Huffman binary tree



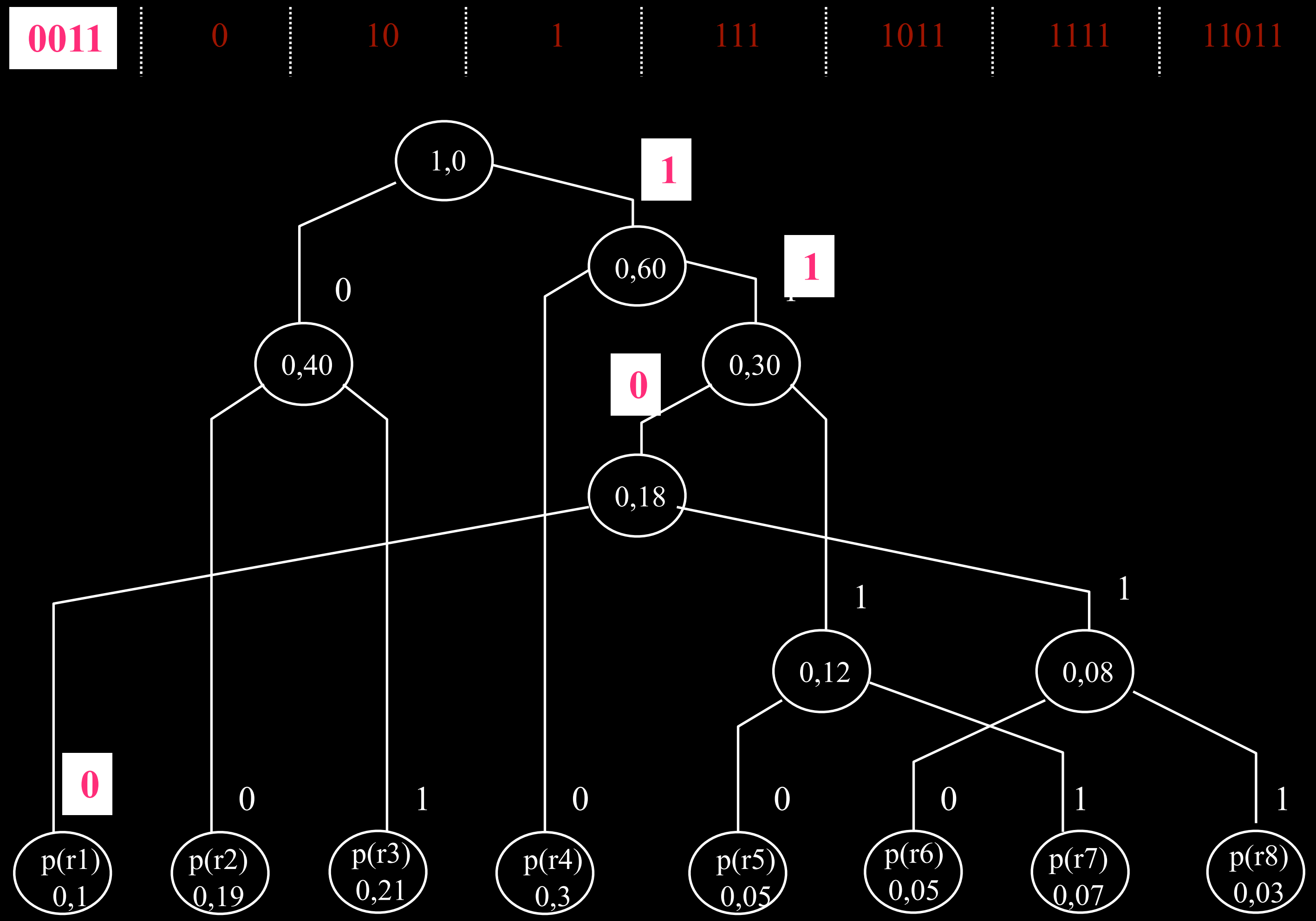
Media compression - entropy coding

- Huffman binary tree



Media compression - entropy coding

- Huffman binary tree



0011 | 0 | 10 | 1 | 111 | 1011 | 1111 | 11011

1,0

1

0,60

1

0

0,40

0

0,30

0,18

1

1

0,12

0,08

0

0

1

0

0

0

1

1

p(r1) 0,1

p(r2) 0,19

p(r3) 0,21

p(r4) 0,3

p(r5) 0,05

p(r6) 0,05

p(r7) 0,07

p(r8) 0,03

Media compression - entropy coding

- major characteristics of Huffman coding
  - optimal results if probability of occurrence of each symbol is a negative power of 2
    - but, in small sets of results this is not likely to happen
    - to overcome this, symbols may be "broken" before
  - the worst case of Huffman efficiency is when symbols have probabilities greater than 0,5
    - to overcome this, VLC is used before Huffman

Packing for transmission or storage

- After entropy encoding, the compressed stream is in its final format - a sequence of variable-length symbols - and ready to be transmitted or stored
- the final packaging for transmission or storage is specified by the systems part of the standards
  - MPEG-2 Systems Layer defines two possible formats
    - Program Stream, consisting of variable length packets, typically long, carrying a complete image
      - not suitable for error-prone environments
      - used for storage applications, namely DVD
    - Transport Stream, defines a multiplex of fixed-sized packets (188 bytes)
      - suitable for environments where errors occur frequently, such as radio channels
      - used in TV broadcasting
    - both syntaxes add an header with systems level information

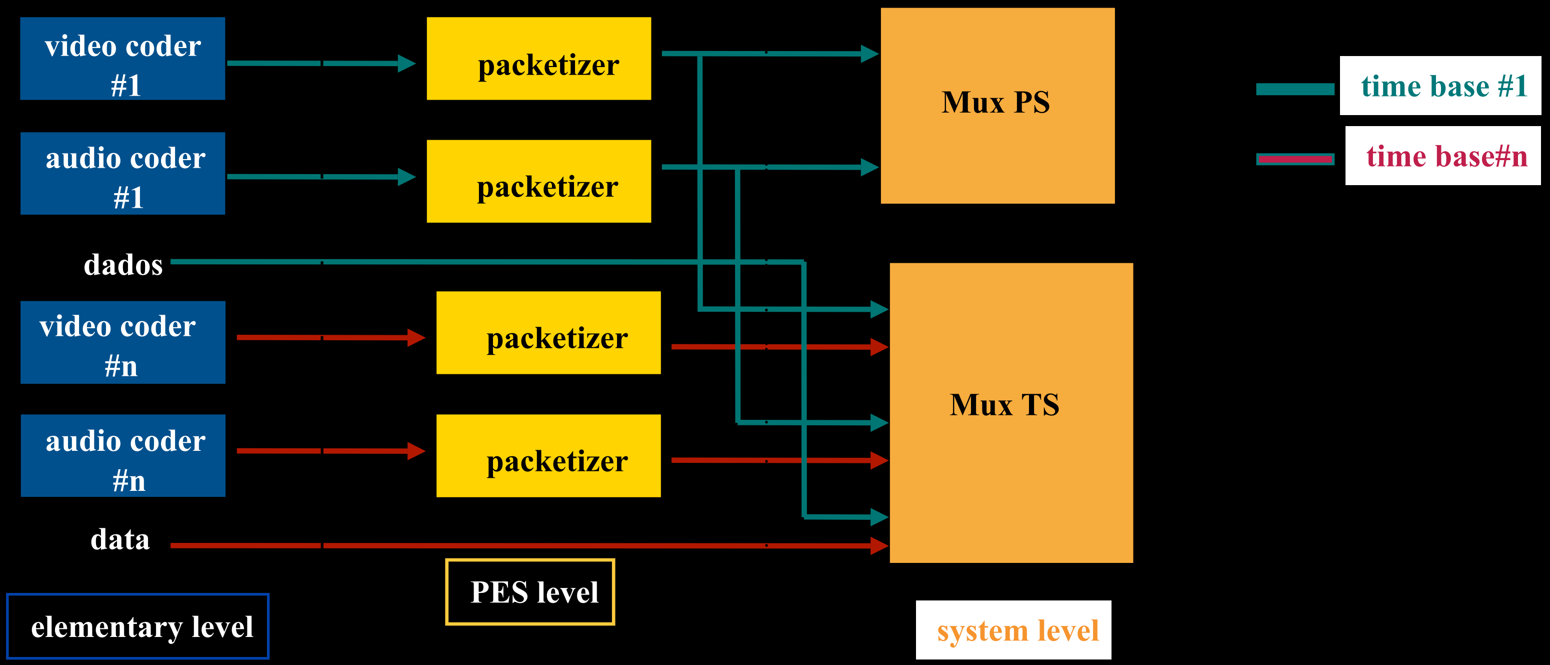
MPEG-2 Systems Layer

Defines the structure to group audio, video and data coded sources to be transmitted or stored

- two different structures or syntaxes are defined, each one with its own characteristics and field of application

  Program Stream (PS)

  Transport Stream (TS)



MPEG-2 Systems Layer

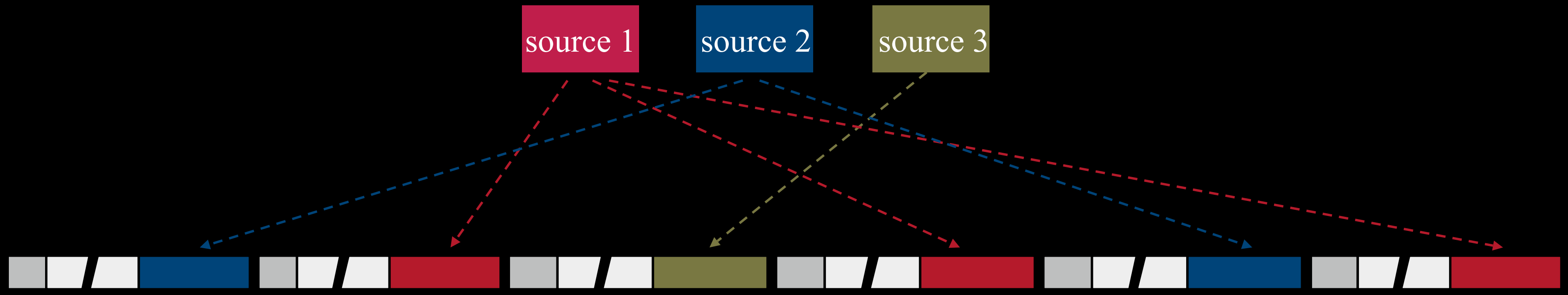
- there are two levels at the systems layer
  - elementary packetizer
    - it receives as input the audio or video coded sequence
    - it outputs Packetized Elementary Streams (PES) consisting of a header followed by the audio or video elementary data
      - each PES packet contains data from only one elementary source
  - multiplexer
    - receives one or more sequences of PES packets
    - outputs a sequence of packets, as a multiplex in time
      - each packet carries data from only one PES but the multiplexed stream is a sequence in time of packets from different PES
        - in the TS syntax packets are of fixed size and may carry audio, video and data elementary streams from different sources or channels
        - in the PS syntax, packets are of variable size and carry audio, video and data from only one source or channel

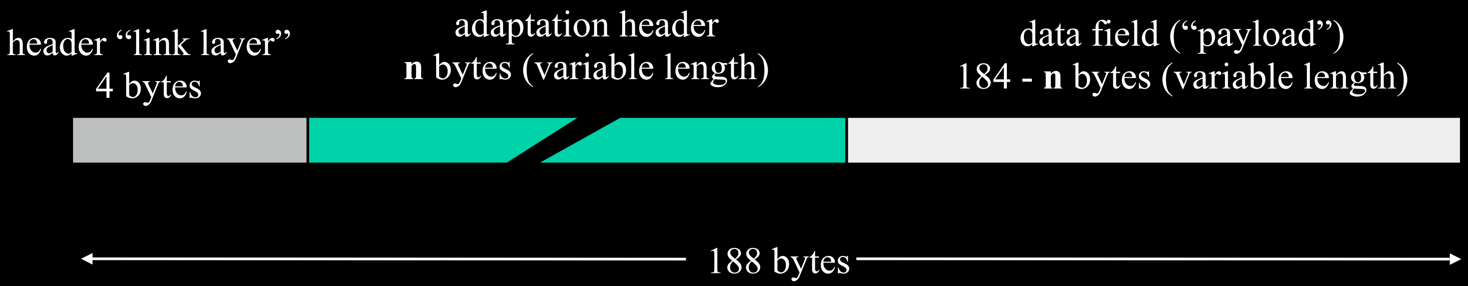
MPEG-2 Systems Layer

- the TS syntax allows to combine in the same multiplexed stream, sources with different time bases
  - it is thus suits the requirements of TV broadcasting, where multiple channels coming from disparate points are to be sent out together
  - the limited and fixed size of TS packets suits the use in error-prone environments
    - it is more efficient to implement error correction codes
    - if one packet is lost, only a small amount of coded information is lost
    - synchronization between sender and receiver is easily achieved (once you know the start of the packet)
  - however it introduces some overhead

MPEG-2 TS Systems Layer

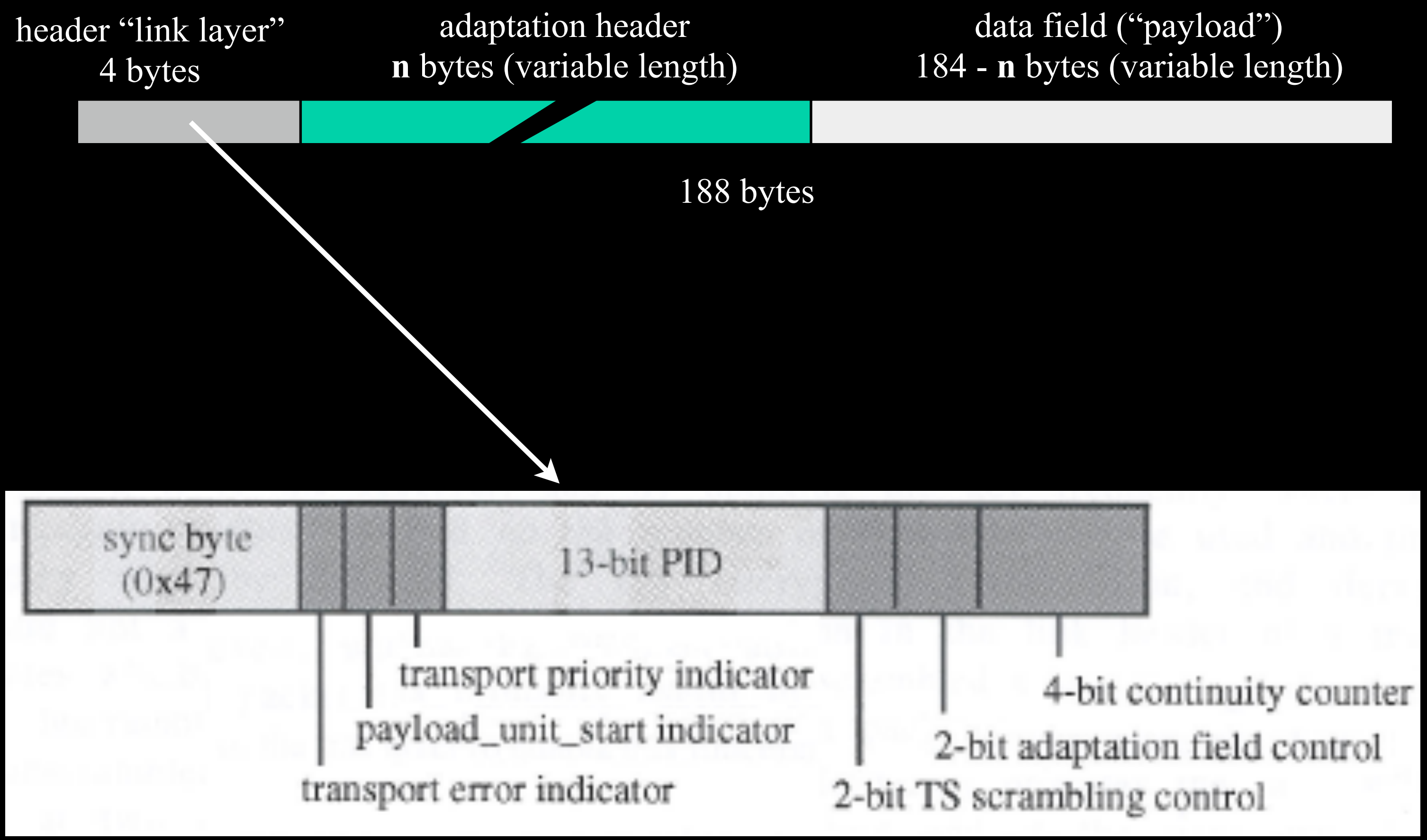
multiplexing level, TS format





MPEG-2 TS Systems Layer

functionality of the headers



header “link layer”
4 bytes

adaptation header
n bytes (variable length)

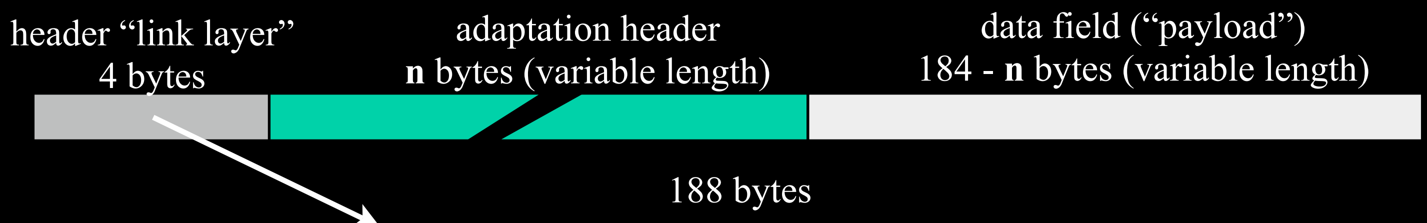
data field (“payload”)
184 - **n** bytes (variable length)

188 bytes

sync byte
(0x47)
13-bit PID
transport priority indicator
payload_unit_start indicator
transport error indicator
4-bit continuity counter
2-bit adaptation field control
2-bit TS scrambling control

MPEG-2 TS Systems Layer

functionality of the headers

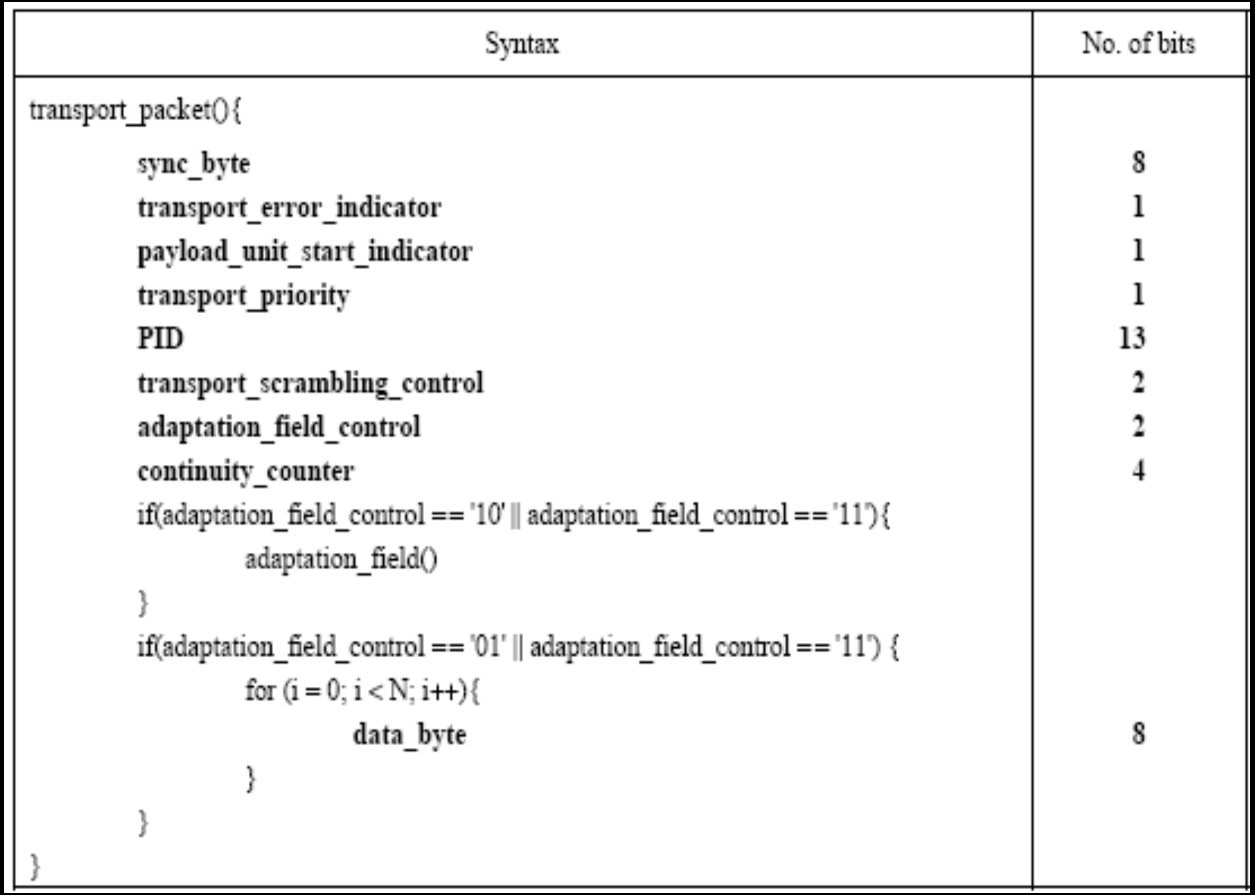


header “link layer”
4 bytes

adaptation header
n bytes (variable length)

data field (“payload”)
184 - **n** bytes (variable length)

188 bytes



Syntax	No. of bits
transport_packet(){	
sync_byte	8
transport_error_indicator	1
payload_unit_start_indicator	1
transport_priority	1
PID	13
transport_scrambling_control	2
adaptation_field_control	2
continuity_counter	4
if(adaptation_field_control == '10' || adaptation_field_control == '11'){	
adaptation_field()	
}	
if(adaptation_field_control == '01' || adaptation_field_control == '11') {	
for (i = 0; i < N; i++){	
data_byte	8
}	
}	
}	

MPEG-2 TS Systems Layer

- functionality included in the header "link layer"
  - packet synchronization
    - using the initial byte with always the value 0x47 ("sync byte")
  - packet identification
    - using the 13 bit field PID, Packet Identifier, that provides a mechanisms to insert and extract packet that carry data from a particular source
    - since the PID field occurs always in the same position within the packet header, extraction of packets belonging to a given elementary source after packet synchronization has been achieved, is done by simply filtering out from the multiplex those packets with the identified PID

MPEG-2 TS Systems Layer

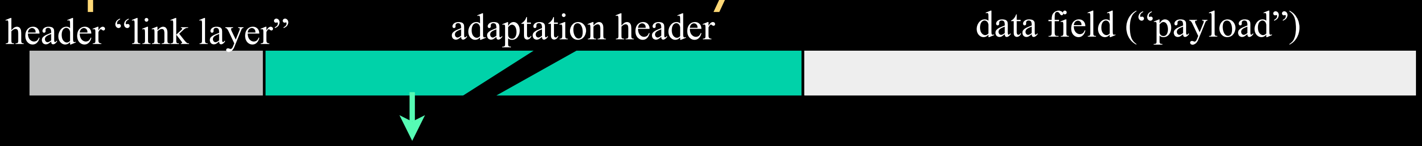
- "link layer" header functionality
  - error robustness using three mecanisms
    - field "**transport_error_indicator**", to signal that an error has occurred in the packet
    - 4 bit "**continuity_counter**"
      - in the transmitter side, the value of this counter is incremented from 0 to 15 for packets with the same PID that are transmitted
      - in the receiver, a discontinuity in this field, indicates that a loss has occurred. A report may be sent to the transmitter requesting the retransmission (not specified in the standard)
    - "**transport_priority**" field
      - a value of "1", indicates to the network equipment that data in the packet has priority

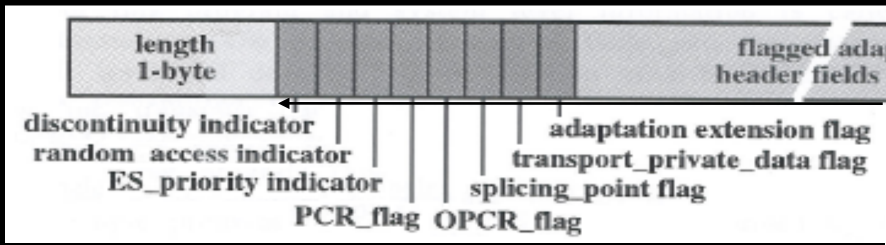
MPEG-2 TS Systems Layer

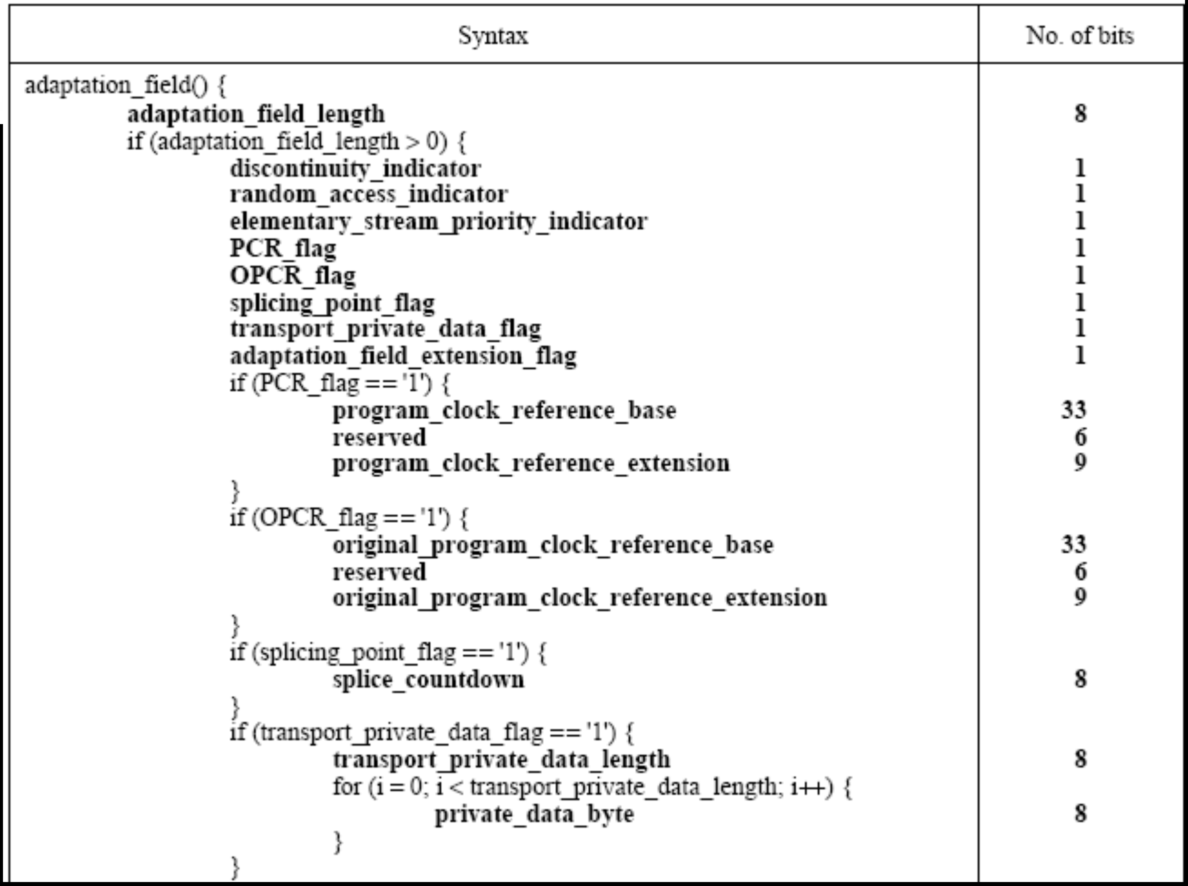
- "link layer" header functionality
  - data alignment
    - using the "**payload_unit_start_indicator**" field
    - a value of "1", indicates that the payload of this TS packet starts with the first byte of a PES packet (and thus with the start of an image or audio frame if the PES packet contains one entire image or audio frame)
  - conditional access
    - using the "**transport_scrambling_control**" e "**adaptation_field_control**" fields
      - "transport_scrambling_control" (2 bits) indicates whether the data in the TS packet is or not encrypted (00 means it is not)
      - "adaptation_field_control" signals the presence of an adaptation header with more data regarding conditional access

MPEG-2 TS Systems Layer

- Adaptation header functionality







Syntax	No. of bits
adaptation_field() {	
adaptation_field_length	8
if (adaptation_field_length > 0) {	
discontinuity_indicator	1
random_access_indicator	1
elementary_stream_priority_indicator	1
PCR_flag	1
OPCR_flag	1
splicing_point_flag	1
transport_private_data_flag	1
adaptation_field_extension_flag	1
if (PCR_flag == '1') {	
program_clock_reference_base	33
reserved	6
program_clock_reference_extension	9
}	
if (OPCR_flag == '1') {	
original_program_clock_reference_base	33
reserved	6
original_program_clock_reference_extension	9
}	
if (splicing_point_flag == '1') {	
splice_countdown	8
}	
if (transport_private_data_flag == '1') {	
transport_private_data_length	8
for (i = 0; i < transport_private_data_length; i++) {	
private_data_byte	8
}	
}	

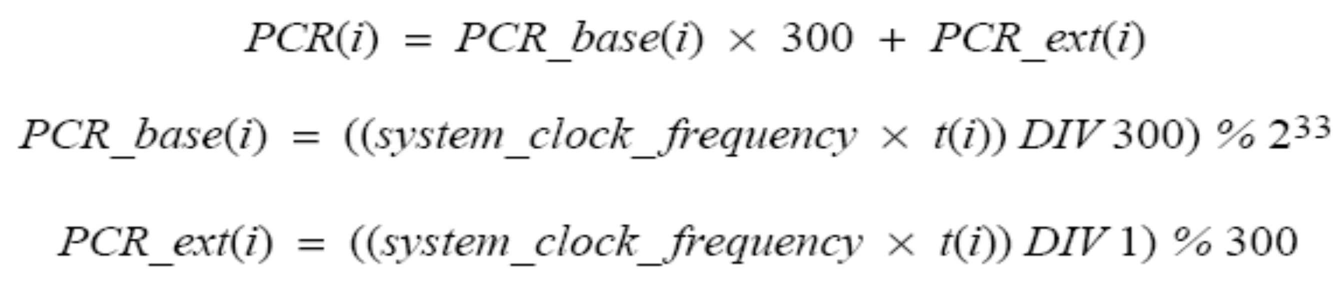
-

MPEG-2 TS Systems Layer

- Adaptation header functionality
  - random access
    - using the “random_access_indicator” field
      - very important in TB broadcasting to tune into a programme
        - the video decoder must start decoding on a I-frame
      - a value of “1” indicates that the TS packet contains the start of an I picture
        - when tuning in a new channel, all packet with the desired PID that do not have this field with value “1”, can be simply ignored

MPEG-2 TS Systems Layer

- Adaptation header functionality
  - Synchronization
    - implemented using the PCR (Program Clock Reference) and PCR_flag fields
      - the presence of a PCR is signaled by the value “1” of the field “PCR_flag”
    - each programme may have its own independent clock or time base, thus a Multi Program Transport Stream will carry diferent PCRs
      - each PCR value corresponds to samples of the time base inserted in the multiplex at most with intervals of 100 ms
      - coded as two fields
        - 33-bit field PCR_base and 9-bit field PCR_extension



$$PCR(i) = PCR\_base(i) \times 300 + PCR\_ext(i)$$

$$PCR\_base(i) = ((system\_clock\_frequency \times t(i))\ DIV\ 300)\ \%\ 2^{33}$$

$$PCR\_ext(i) = ((system\_clock\_frequency \times t(i))\ DIV\ 1)\ \%\ 300$$

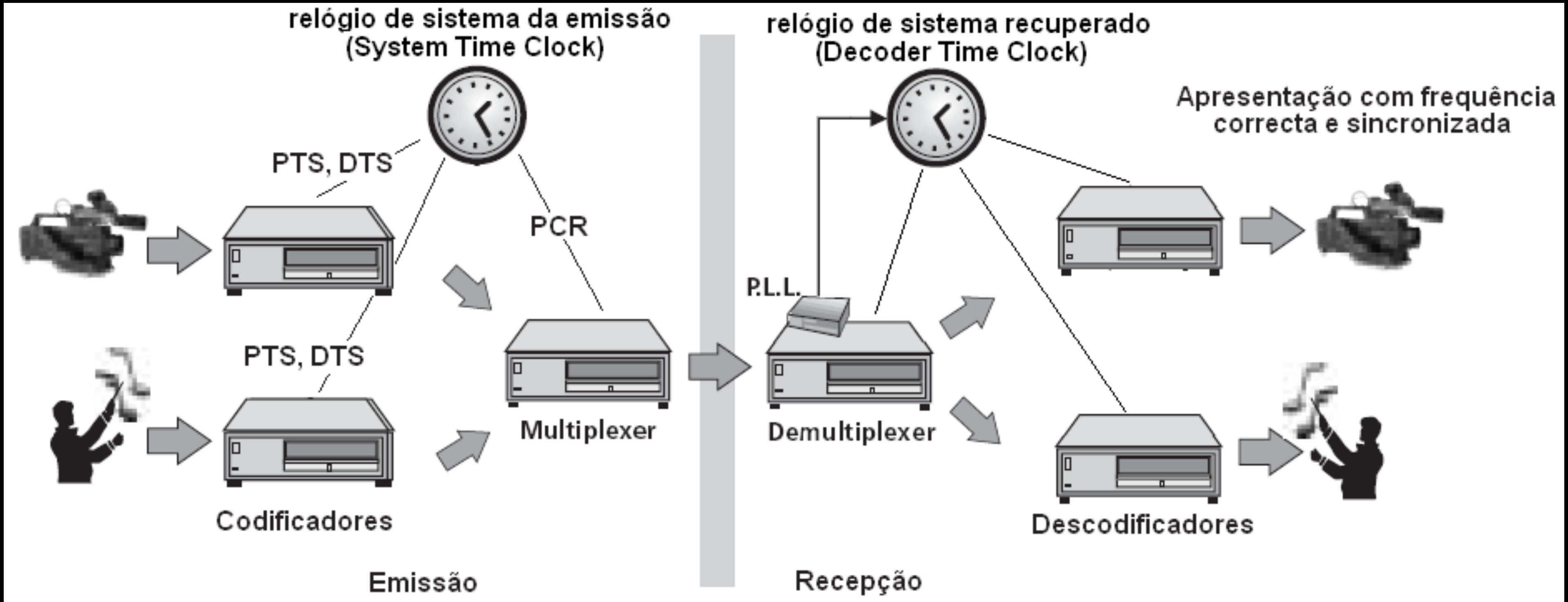
- receivers use these samples to regenerate a time base that follows (is synchronized with) the original time base
  - sender and receiver become synchronized

MPEG-2 TS Systems Layer

- Adaptation header functionality
  - Synchronization
    - in addition to the PCR time stamp, there are also other time stamps required to obtain synchronization of elementary streams (audio and video)
      - PTS (Presentation Time Stamp) and DTS (Decoding Time Stamps)
        - samples from the clock system just like the PCR
        - PTS and DTS are stored by the decoder and continuously compared against the regenerated time base (that was regenerated using the PCRs)
          - when values are equal, it indicates that the decoder must either decode and image or audio frame (DTS=time base) or present it (PTS=time base)
          - have a 90 KHz resolution

MPEG-2 TS Systems Layer

- Adaptation header functionality
  - Synchronization



MPEG-2 TS Systems Layer

- After the implementation of the systems layer functionality, the coded stream is ready to be transmitted
  - in IP networks, using the IETF protocol stack!

next: IETF protocol stack

- IETF protocols
  - signaling
    - RTSP, SDP, SIP, SAP
  - transport
    - TCP, UDP, RTP+RTCP
  - QoS assurances
    - RSVP, DiffServ