Introduction to image coding

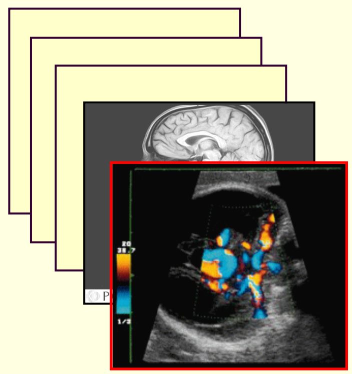
Image coding aims at reducing amount of data required for image representation, storage or transmission.

This is achieved by removing redundant data from an image, i.e. by using a transfomation of a 2-D pixel array into uncorrelated data set

Diagnostic imaging

A single biomedical image requires storage of hundreds of MB of data

Acquisition



Archivation (data bases)

Presentation

Why do we need to compress images?

Standard colour TV image in digital form is 640×576 pixel image, that takes up more than 1.1 MB of memory. 25 such images shown per second generate ~28 MB of data.

Audio CD disk can store ~25 s of such a film.

A two-sided DVD disc (17 GB) can store 10 min. of a film.

Image coding techniques

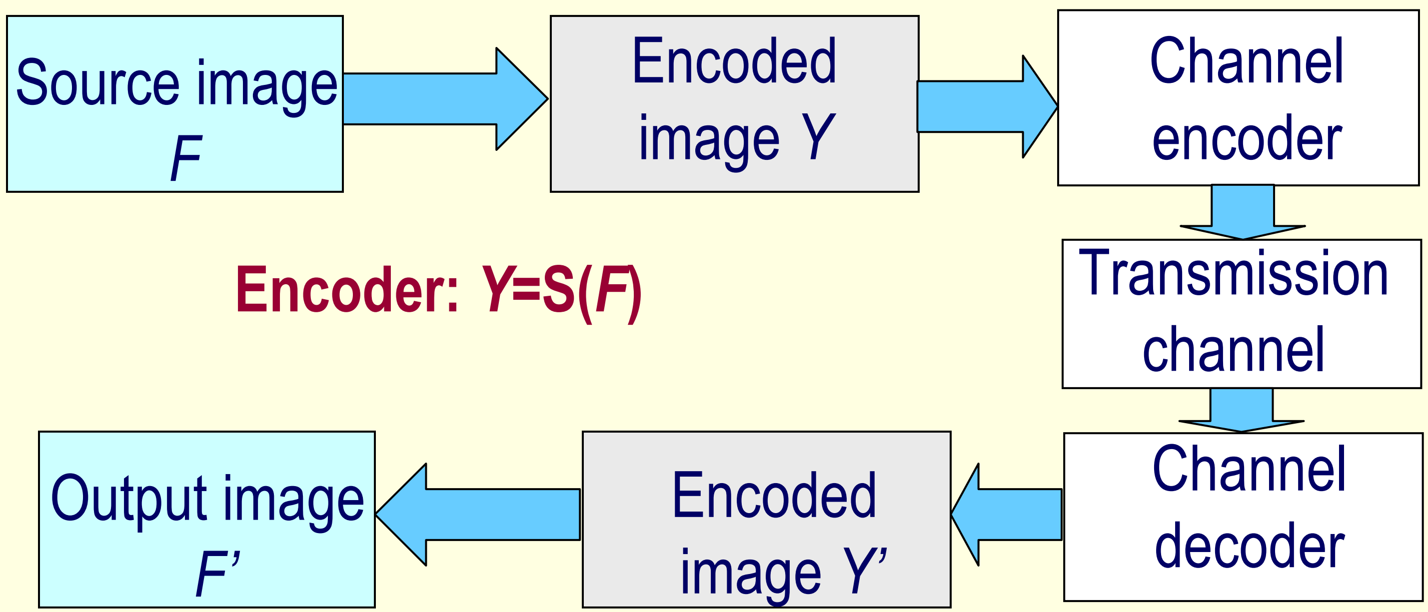
<u>Aim:</u> removing redundancy and keeping important information,

np. Al● h●s a c●t

- **lossless** based on statistical image properties e.g. image histogram → CR~ a few
- **lossy** based on spatial correlation between image pixels → CR~ a few
- **lossless** and **lossy** based on image transforms → CR~ tens-hundreds

CR - **Compression Ratio**

Simple image transmission channel



Encoder: Y=S(F)

Decoder: F'=T(Y')

Error-free channel: Y'=Y

If $F' = F$, (T = S^{-1}) then loss-less compression
If $F' \neq F$, (T $\neq S^{-1}$) then lossy compression

Data redundancy and compression ratio

Let n_1, n_2 denote data units carrying the same information.

Relative data redundancy is given by:

$$R_D = 1 - 1/C_R$$

where C_R – is termed the ***compression ratio:***

$$C_R = n_1 / n_2$$

for $n_1 = n_2$: $C_R = 1$; $R = 0$
for $n_1 >> n_2$: $C_R \rightarrow \infty$; $R \rightarrow 1$
for $n_1 << n_2$: $C_R \rightarrow 0$; $R \rightarrow -\infty$

Source entropy

Source ***entropy***: :

$$E_F = -\sum_{i=0}^{L-1} p(l_i)\log_2 p(l_i)$$

represents the average information per source output.

For data source given in the form of an image, $p(l_i)$ – is the probability of occurrence of gray level l_i.

Entropy can be interpreted as the average number of bits required for coding a single image pixel.

Entropy values for example images

E = 7.01

E = 5.31

Types of redundancies in digital images

- coding redundancy
- interpixel redundancy
- psychovisual redundancy

Coding redundancy

The average number of bits required to code each pixel:

$$m_{avg} = \sum_{i=0}^{L-1} m(l_i)\, p(l_i)$$

where $m(l_i)$ – wordlength representing gray level l_i

Entropy indicates the minimum average number of bits per pixel that are required to represet an image.

Example

Variable-length code

l_k	$p(l_k)$	code 1	$m^1(l_k)$	code 2	$m^2(l_k)$
0	0.19	000	3	11	2
1	0.25	001	3	01	2
2	0.21	010	3	10	2
3	0.16	011	3	001	3
4	0.08	100	3	0001	4
5	0.06	101	3	00001	5
6	0.03	110	3	000001	6
7	0.02	111	3	000000	6

$m^1_{avg} = 3$, $m^2_{avg} = 2.7$, $E = 2.651$, $C_R = 1.11$, $R = 0.099$, thus ~ 10% of data in code 1 is redundant

Huffman coding

Source		Source reduction			
Symbol	$p(l_k)$	#1	#2	#3	#4
a2	0.4	0.4	0.4	0.4	0.6
a6	0.3	0.3	0.3	0.3	0.4
a1	0.1	0.1	0.2	0.3	
a4	0.1	0.1	0.1		
a3	0.06	0.1			
a5	0.04				

Huffman coding

Source		Source reduction			
Symbol	$p(l_k)$ kod	#1	#2	#3	#4
a2	0.4 1	1 0.4	1 0.4	1 0.4	0 0.6
a6	0.3 00	00 0.3	00 0.3	00 0.3	1 0.4
a1	0.1 011	011 0.1	010 0.2	01 0.3	
a4	0.1 0100	0100 0.1	011 0.1		
a3	0.06 01010	0101 0.1			
a5	0.04 01011				

Properties of the Huffman code

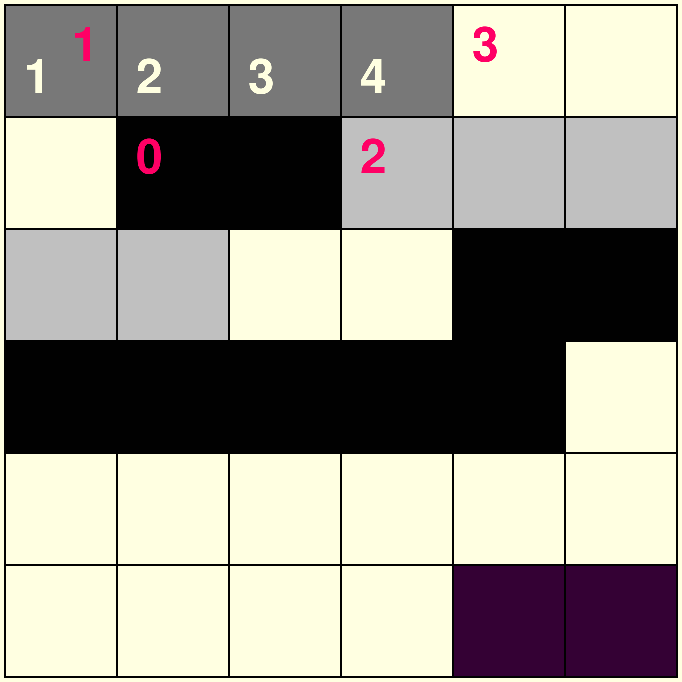
- variable-length code
- yields the smallest possible number of code symbols per source symbol
- is a memory-free code
- gives one-to-ne mapping between symbols and codes

Interpixel redundancy removal

Gray levels of adjacent pixels are strongly correlated.

Run length (RL) coding – an image is coded by symbol pairs (g_i, l_i) where g_i denotes i-th gray level and l_i is the run length

RL example (4 gray level image)



RL sequences: (1,4); (3,3); (0,2); (2,5); (3,2); (0,7); (3,11); (1,2)

8*3=24 bytes are needed for image coding

g – coded by 1 byte,
l – coded by 2 bytes

6x6=36 bytes

$C_R = 36/24 = 1.5$

Predictive coding

Given a sequence of symbols $\{s_i\}$:

1	2	5	7	2	-2	0	-5	-3	-1	1	-2	-7	-4	-2	1	3	4

we try to predict each consecutive symbol by adding 2 to the preceding symbol.

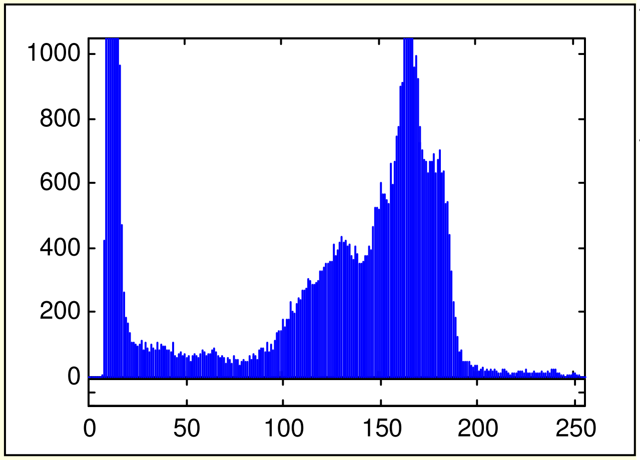
We compute the difference between the predicted and true symbol values in a sequence:

$$e_i = s_i - (s_{i-1} + 2)$$

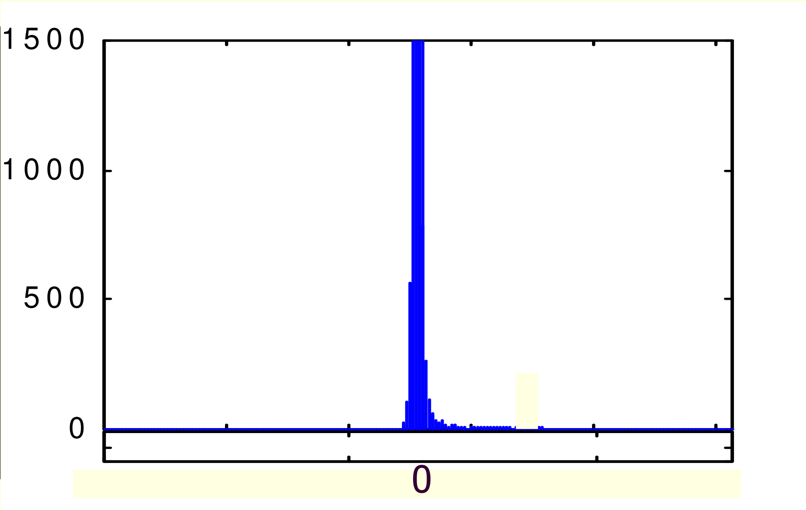
1	-1	1	0	-7	-4	0	-7	0	0	0	-5	-7	1	0	1	0	-1

Note that this error sequence contains many -1,0,1 symbols and other symbols are less frequent

Predictive coding



$\{s_i\}$



$\{e_i\} = \{s_i - (s_{i-1} + 2)\}$

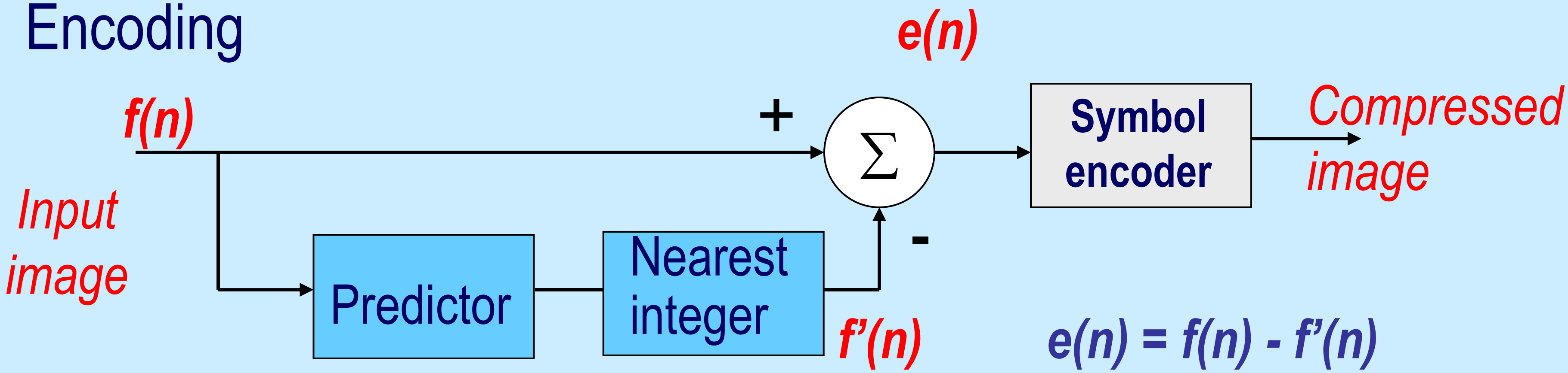
Large entropy, i.e. on average many bits required to code symbols

Small entropy, i.e. on average many bits required to code symbols

Predictive coding is used in MP3 audio compression standard.

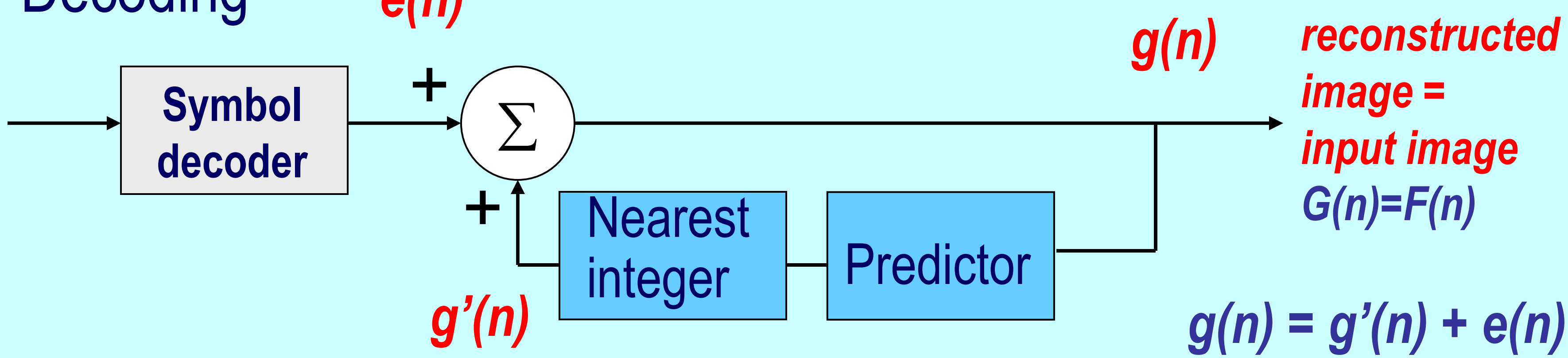
Lossless predictive coding

Encoding



$$e(n) = f(n) - f'(n)$$

Decoding



$$g(n) = g'(n) + e(n)$$

Differential Pulse Code Modulation

$$f'(n) = round\left[\sum_{i=1}^{m} a_i f(n-i)\right]$$

Lossless predictive coding - example

n	f_n	$f_n' = round[0.9 f(n-1)]$	$e_n = f_n - f_n'$	$g_n = e_n + g_n'$
0	**10**	-	-	**10**
1	**11**	round(9)=9	2	2+9=**11**
2	**15**	round(9.9)=10	5	5+10=**15**
3	**12**	round(13.5)=14	-2	-2+14=**12**
4	**17**	round(10.8)=11	6	6+11=**17**
5	**20**	round(15.3)=15	5	5+15=**20**
..	..	....	....	....

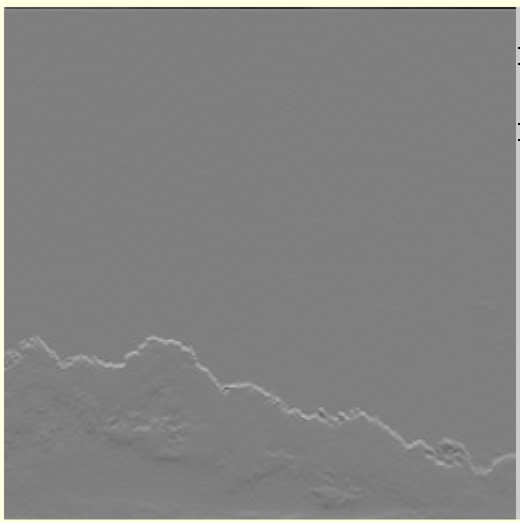
This pixel is transmitted first

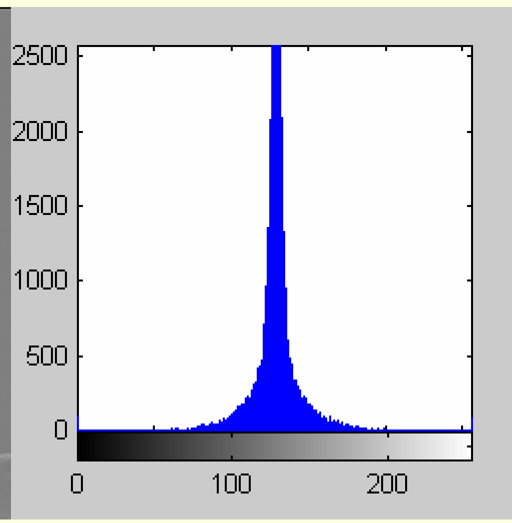
decoder

Predictive coding - example

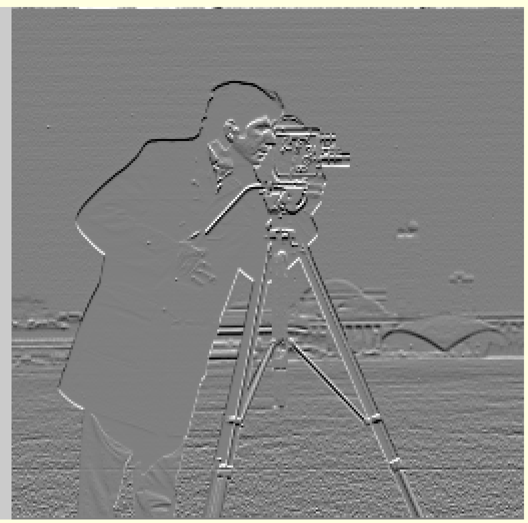
$$f'(n) = round\left[\sum_{i=1}^{m} a_i f(n-i)\right] = f(n-1) \qquad m=1, a_1=1$$

e(n)





e(n)



E = 2.44 (5.31)

C_R = 3.28 (1.51)

E = 5.06 (7.01)

C_R = 1.58 (1.14)

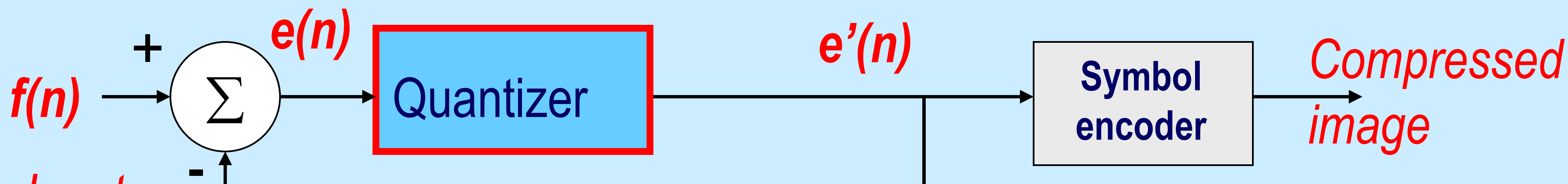
Lossless predictive coding algorithm

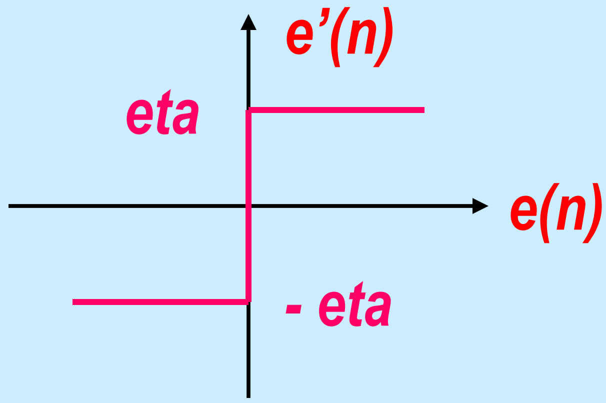
```
f[ i ],g[ i ] = array[1..N*N] of byte {image given as a vector}
e[ i ] = array[2..N*N] of integer {error image}
{1-st order predictor, a1=1}

{encoding}
for i := 2 to N*N do
begin
  e[ i ] := f[ i ] - f[ i-1 ];
  e[ i ] :=e [ i ] + 128;
  if e[ i ] > 255 then e[ i ] := 255;
  if e[ i ] < 0 then e[ i ] := 0;
end;
{decoding}
g[ 1 ] := f[ 1 ];
for i := 2 to N*N do g[ i ] := e[ i ] + g[ i-1 ];
```

Lossy predictive coding

Encoding





Quantizer

② $e(n) = f(n) - f'(n)$

③ $f''(n) = e'(n) + f'(n)$

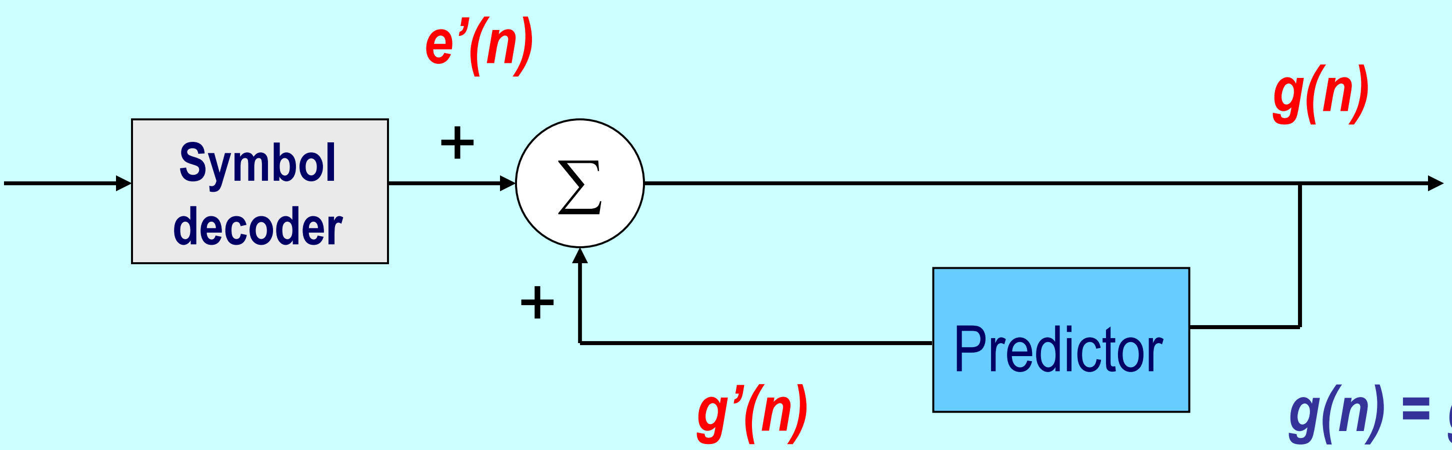
① $f'(n) = a_1 f''(n-1) = e'(n-1) + f'(n-1)$; 1-st order predictor, $a_1=1$

$f''(1) = f(1)$

Lossy predictive coding

Decoding

$g(1) = f(1)$



reconstructed image ≠ output image

$g(n) \neq f(n)$

$g(n) = g'(n) + e'(n)$

$g'(n) = g(n-1)$

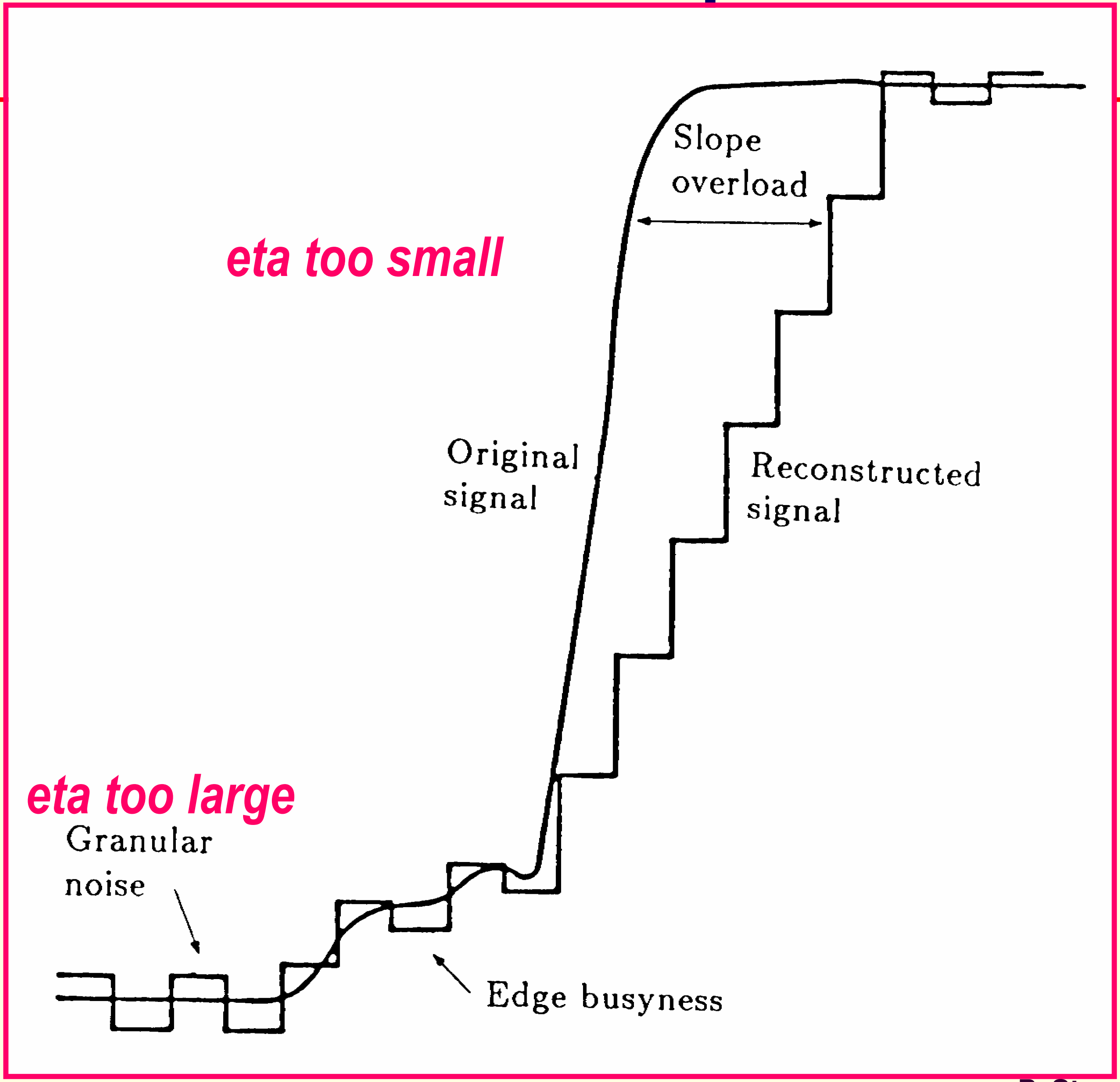
Lossy predictive coding - example

n	f_n	$f_n^{'}$	e_n	$e_n^{'}$	$f_n^{''} = e_n^{'} + f_n^{'}$	$g_n = g_n^{'} + e_n^{'}$	$f_n - g_n$
0	**10**	-	-		10	10	0
1	**11**	10	1	5	15	15	-4
2	**15**	15	0	5	20	20	-5
3	**12**	20	-8	-5	15	15	-3
4	**17**	15	2	-5	10	10	7
5	**20**	10	10	5	20	20	0
..			..		....	....	....

This pixel is transmitted first

decoder

Delta modulation example



Eta=1

Eta=10

Eta=20

Eta=40

Lossy predictive coding algorithm

```
f[ i ], f '[ i ],  g[ i ] = array[1..N*N] of byte {image given as vector}
e[ i ] = array[2..N*N] of integer {error image}
{1-st order predictor, a1=1, eta – quantizer constant}
{encoding}
f ' [ 2 ] := f [ 1 ];
for i := 2 to N*N do
begin
  e[ i ] := f[ i ] - f '[ i ];
  if e[ i ] >= 0  then  e[ i ] := eta  else  e[ i ] := -eta;
  f ' [ i+1 ] := e[ i ] + f ' [ i ];
end;
{decoding}
g :=  f ';
if g[ i ] < 0  then  g[ i ] := 0;  if g[ i ] > 255  then  g[ i ] := 255;
```

Psychovisual redundancy

Visual perception of humans does not work as a camera of a predefined characteristic.

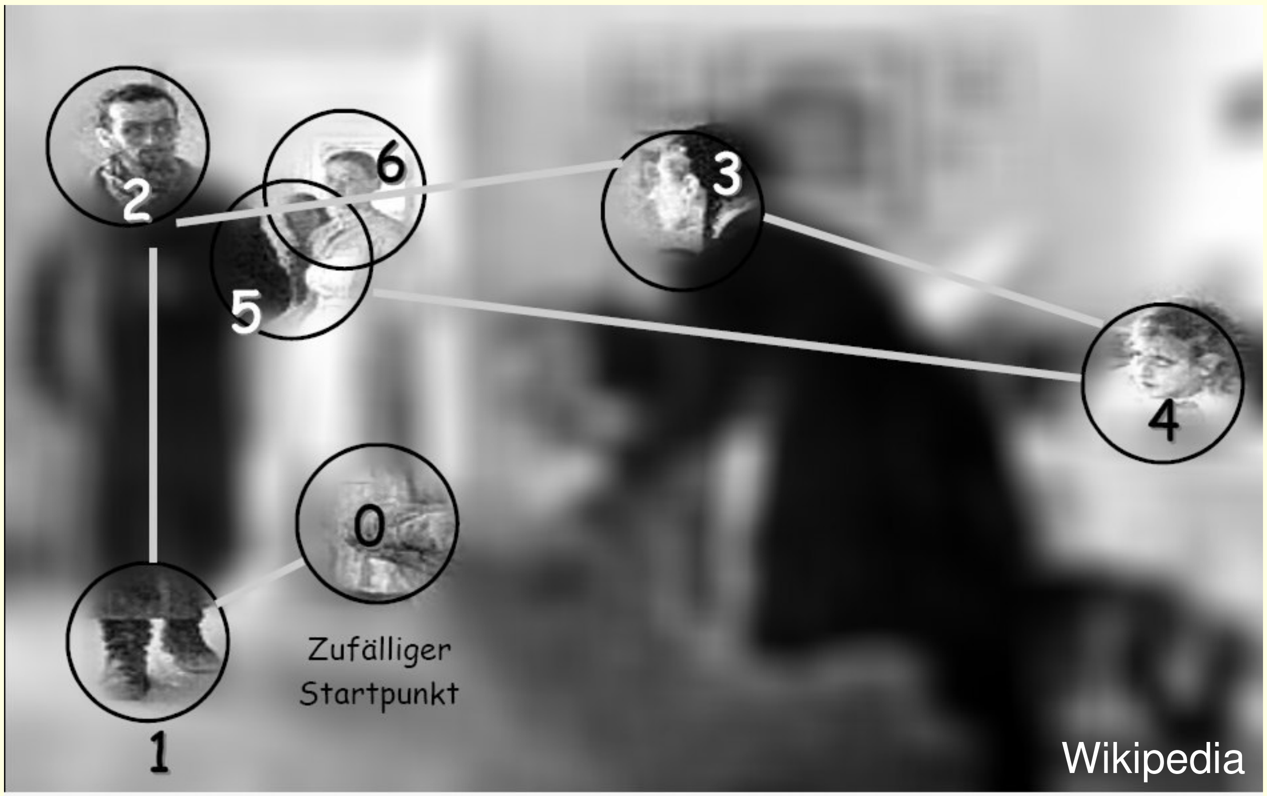


Bild 11: Foveale Ergänzung durch die ersten 6 Fixationen (nach Daten von *Yarbus, 1967*)

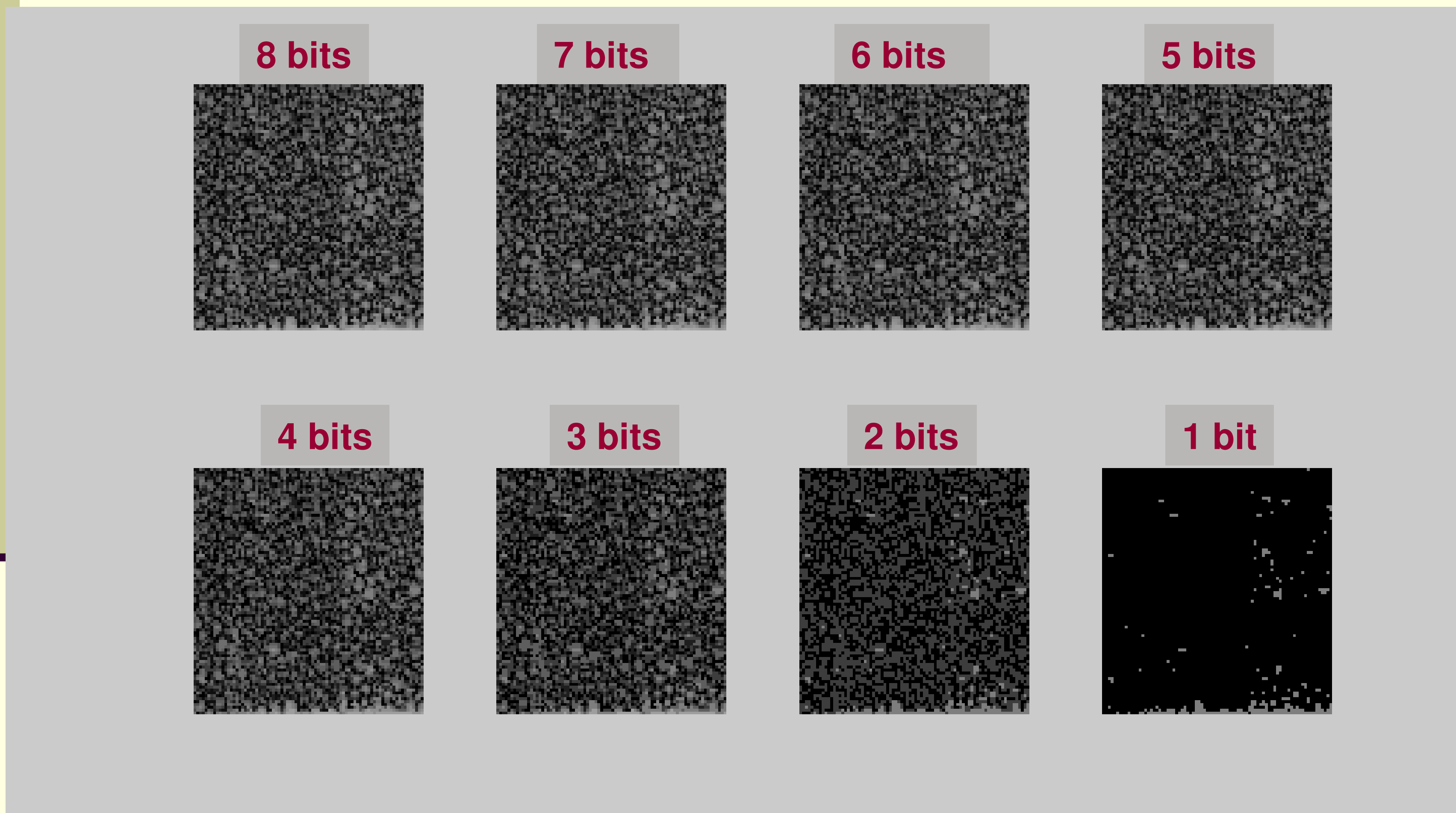
Psychovisual redundancy (example)

8 bits

5 bits

3 bits

Psychovisual redundancy examples of image textures



Transform coding

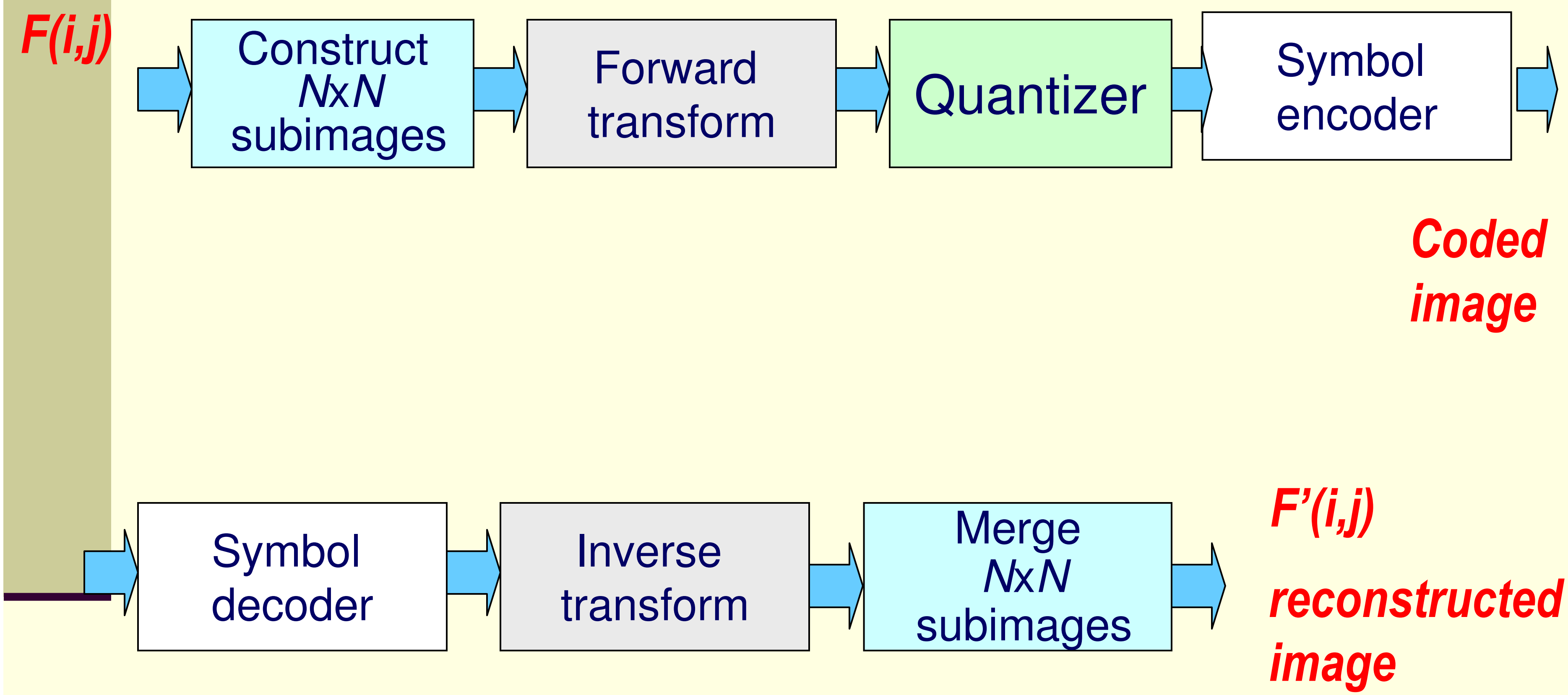


Image transforms

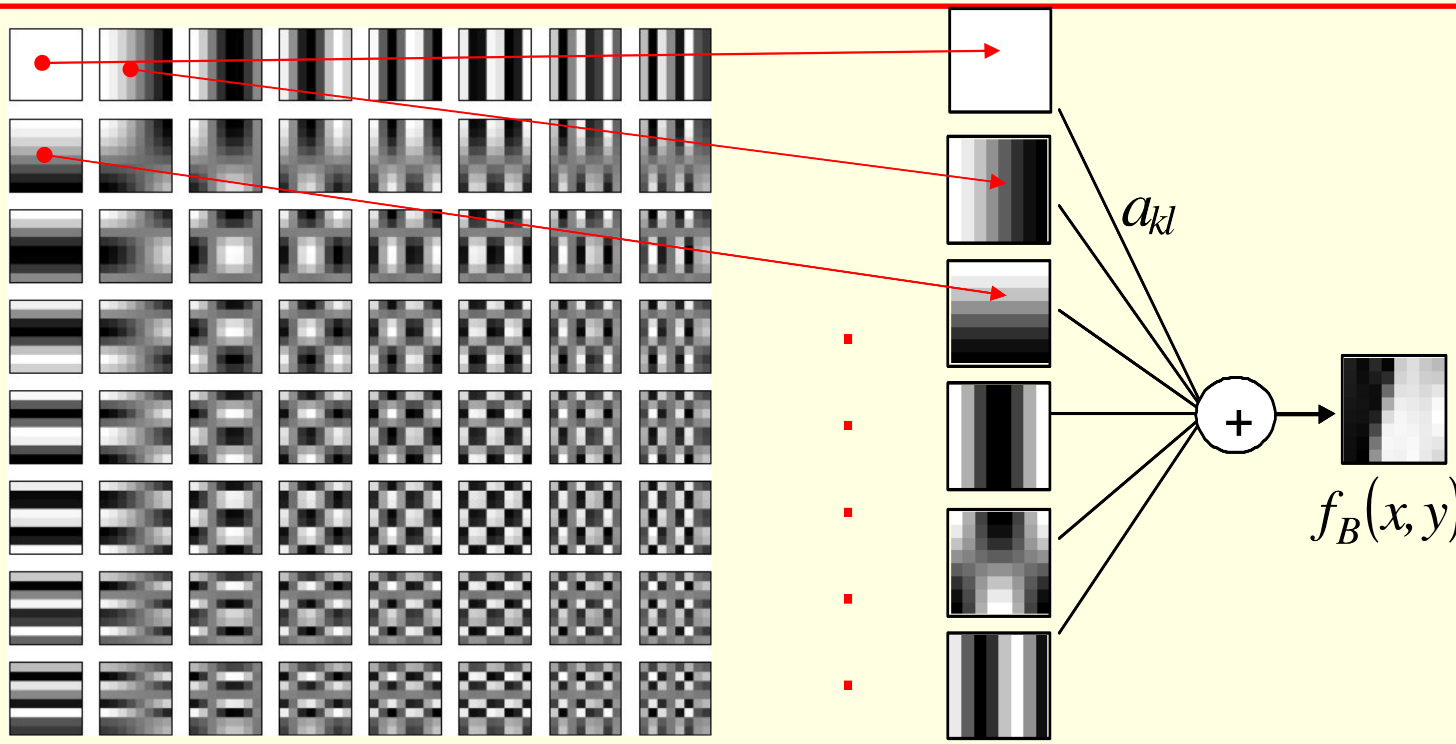
- **The Karhunen-Loeve expansion** - minimizes the mean square reconstruction error, computationally complex, no fast transform
- **Discrete Cosine Transform** (**DCT**) – high compression ratios achievable (fast vanishing of cosine coefficients), fast transform exists (used in JPEG)
- **The wavelet transform** – new compression technique, competitive or better than DCT, fast transform exists
- **MPEG 2000, JPEG 2000**

Image compression standards

- **JPEG** (***Joint Photographic Experts Group***) – a group of experts (engineers and scientists) working on developing new methods and standards for coding still images (www.jpeg.org)

- **MPEG** (***Moving Pictures Experts Group***) - a group of experts working on developing standards for coding audio and video signals (www.mpeg.org)

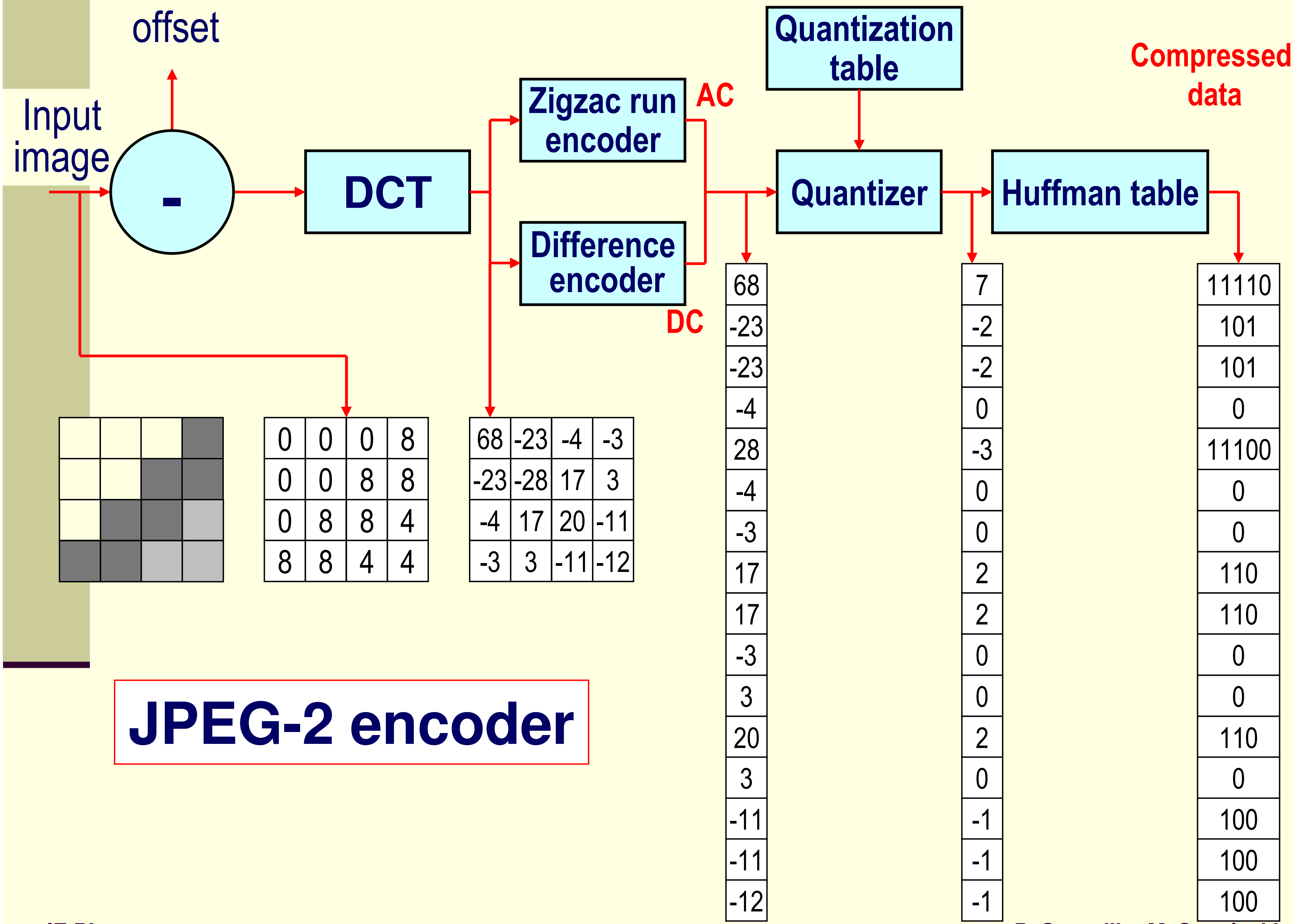
DCT basis functions



$$f_B(x,y)=\sum_{k=0}^{7}\sum_{l=0}^{7}a_{kl}\cos(k\omega_0 x)\cos(l\omega_0 y),\ \ for\ \ x,y=1,2,\ldots,7$$

Image 8x8 blocks are obtained by computing a linear combination of DCT basis functions

JPEG-2 encoder

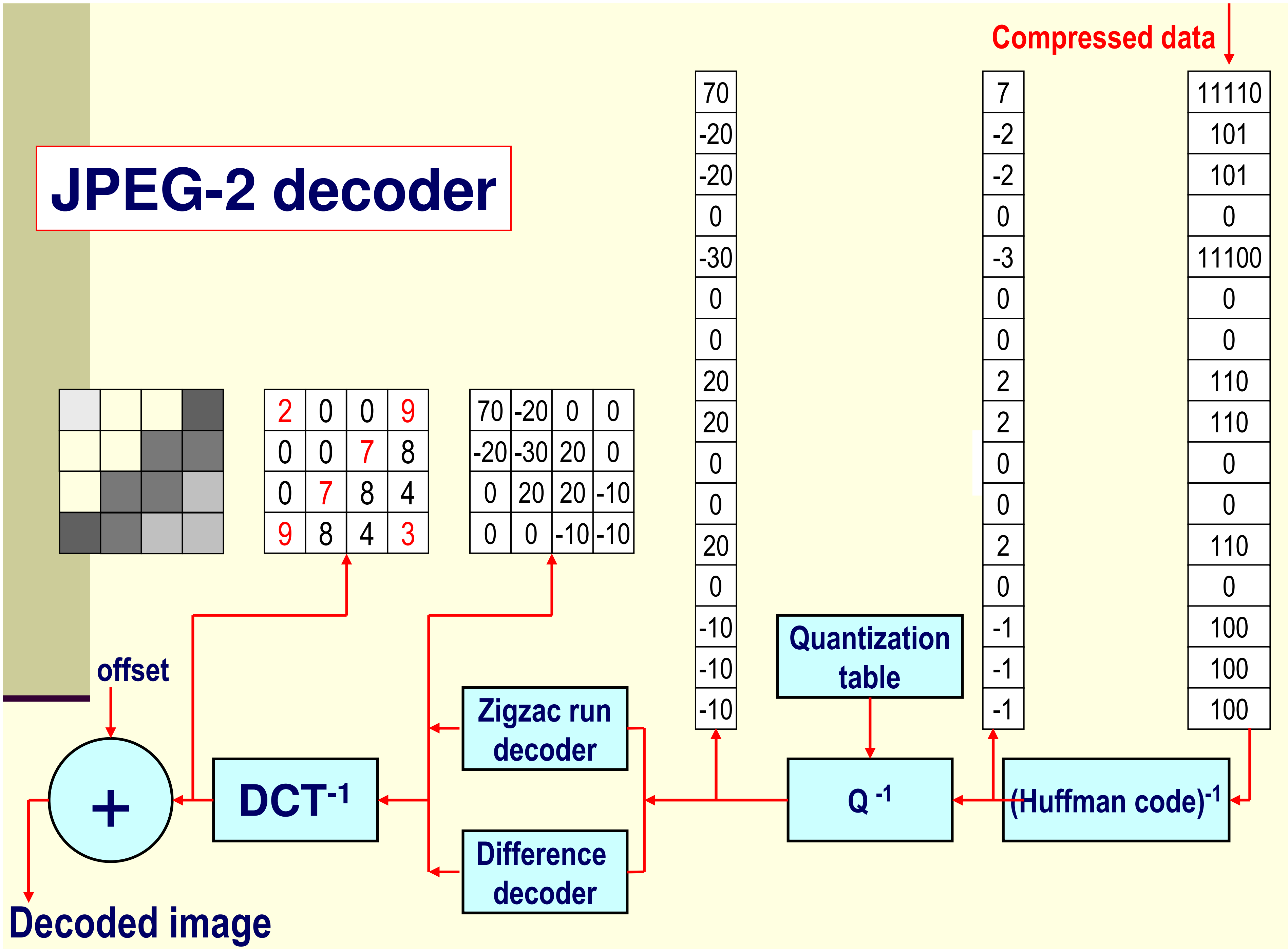


0	0	0	8
0	0	8	8
0	8	8	4
8	8	4	4

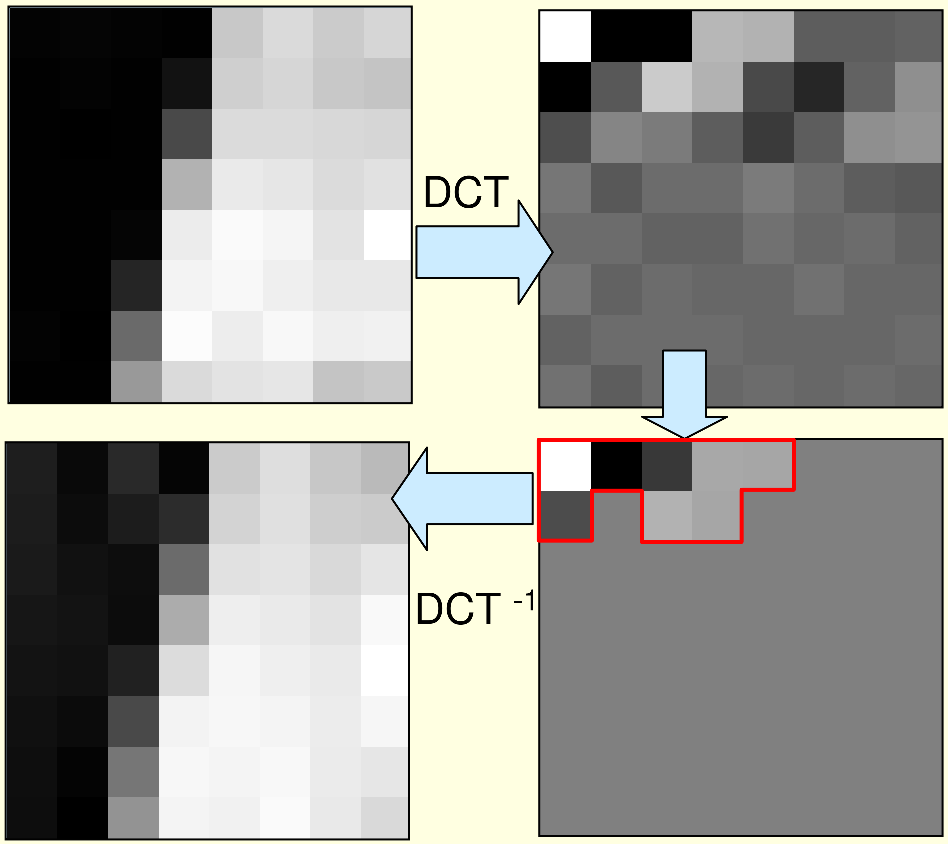
68	-23	-4	-3
-23	-28	17	3
-4	17	20	-11
-3	3	-11	-12

68	7	11110
-23	-2	101
-23	-2	101
-4	0	0
28	-3	11100
-4	0	0
-3	0	0
17	2	110
17	2	110
-3	0	0
3	0	0
20	2	110
3	0	0
-11	-1	100
-11	-1	100
-12	-1	100

JPEG-2 decoder



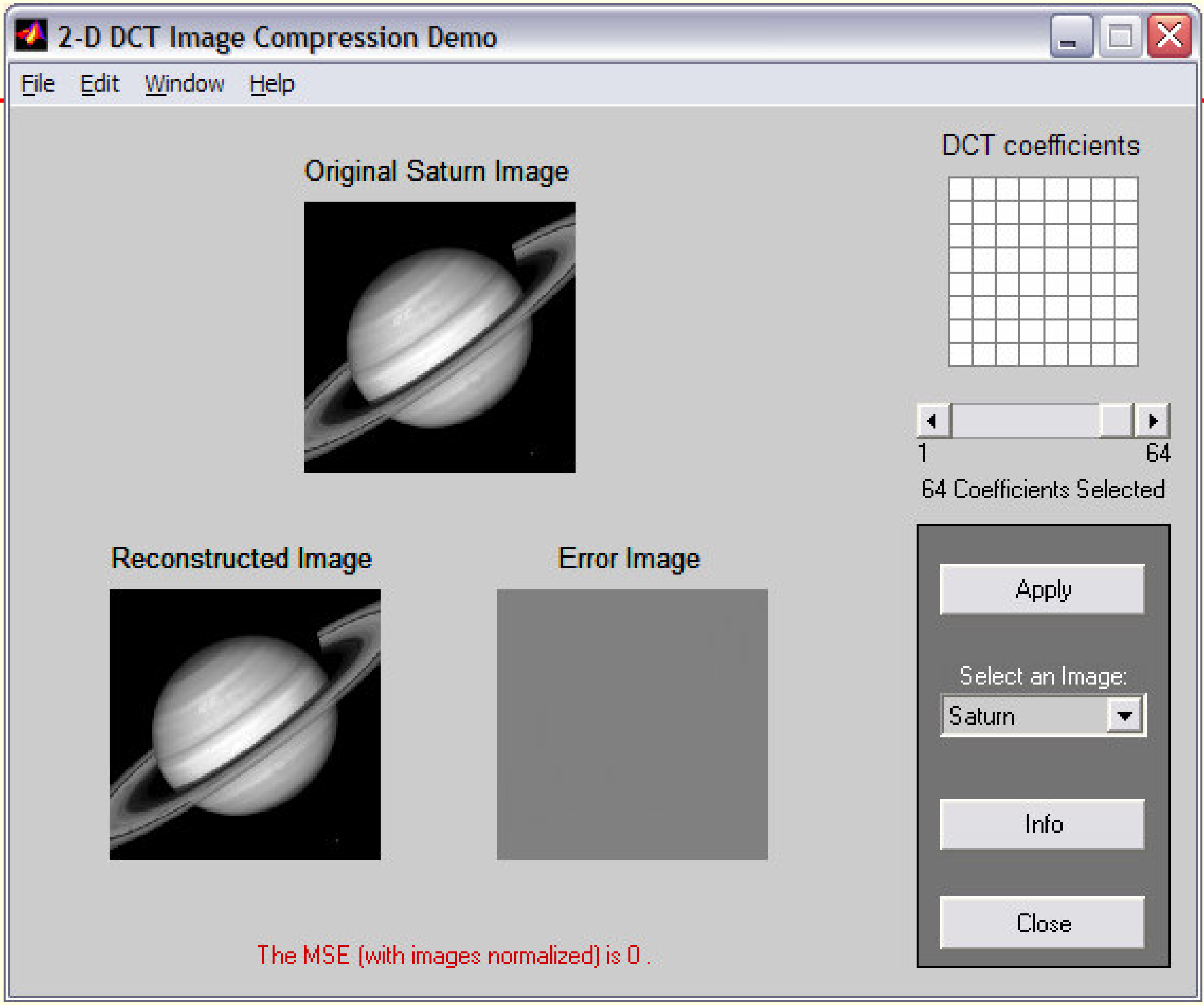
Coding example



Quantization of DCT coefficients

CR=8:1

MATLAB Demo – Discrete Cosine Transform



Fidelity criteria for lossy compression

***Root-mean-square error (RMS)*:**

$$RMS = \sqrt{\frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}\left[f(x,y) - \hat{f}(x,y)\right]^2}$$

***Mean-square signal-to-noise ratio (MSE)*:**

$$SNR = 10 log_{10}\left(\frac{\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}\left[f(x,y)\right]^2}{\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}\left[f(x,y) - \hat{f}(x,y)\right]^2}\right)$$

Fidelity criteria for lossy compression

Mean-square signal-to-noise ratio (MSE):

$$PSNR = 10\,log_{10}\left(\frac{MN\{max[f(x,y)]\}^2}{\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}\left[f(x,y)-\hat{f}(x,y)\right]^2}\right)$$

Most often used

Subjective fidelity criteria

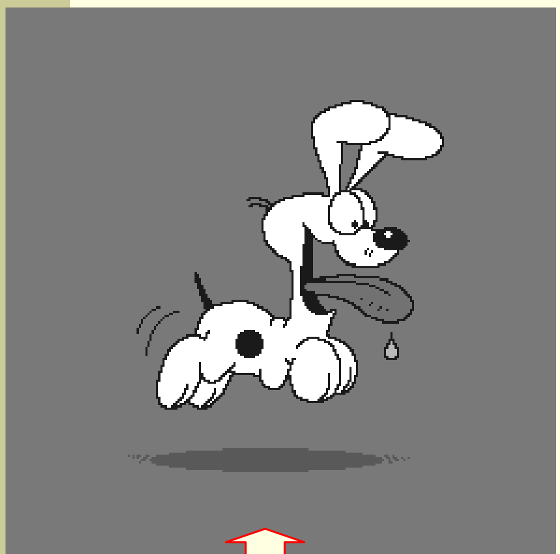
Objective criteria are mean measures evaluated for an entire image.

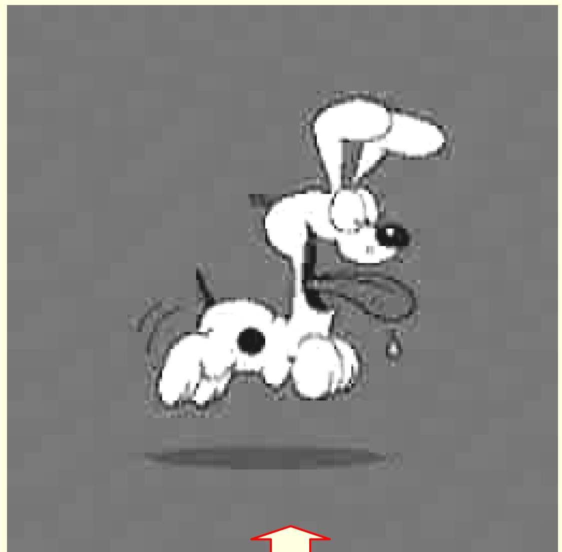
Subjective criteria do not need to match the obejctive criteria!

Subjective criteria:

- **arbitrary scale** (e.g. 0-5)
- **comparative** (comparison of the subjective perception of different coded images to the original).

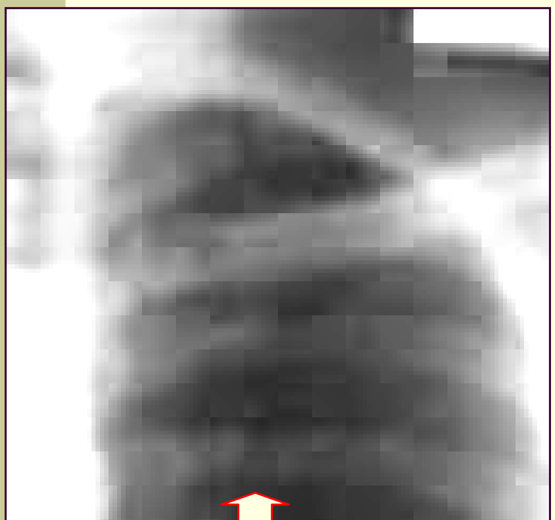
JPEG coding

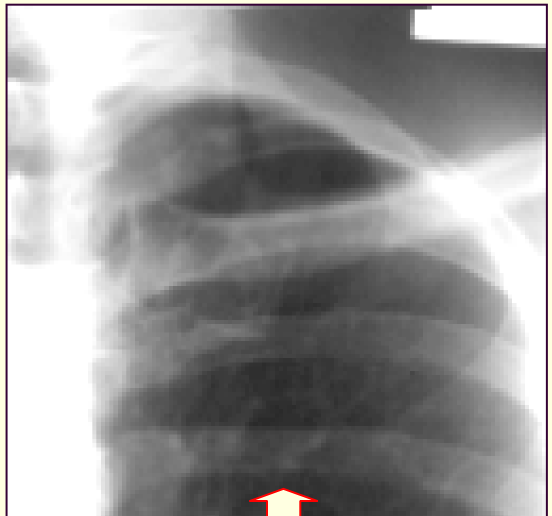


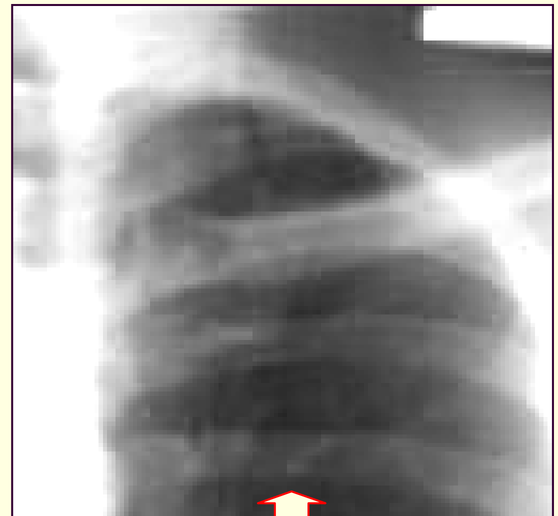


	Parameter	Value	Parameter	Value
Bitmap (8bpp)	**CR**	**20:1**	**CR**	**40:1**
	bpp	0.4	bpp	0.2
	MSE	257	MSE	367
	SNR	8.2 dB	SNR	6.6 dB
	PSNR	24 dB	PSNR	22.5 dB

JPEG coding

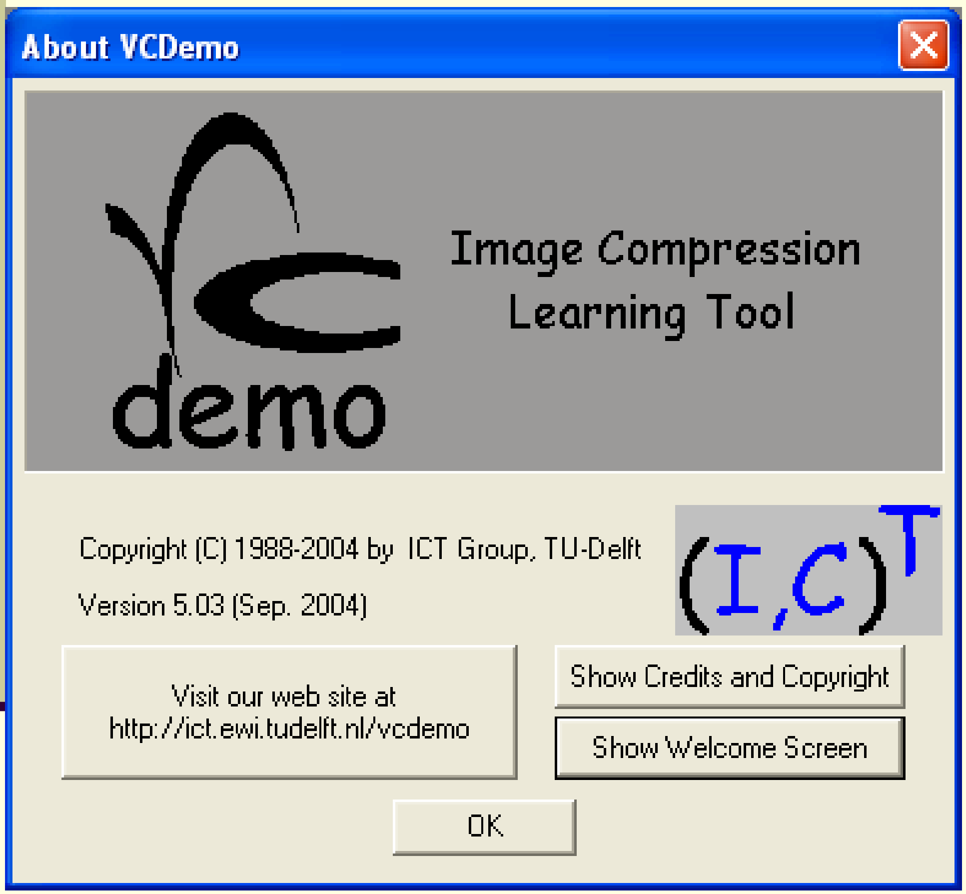


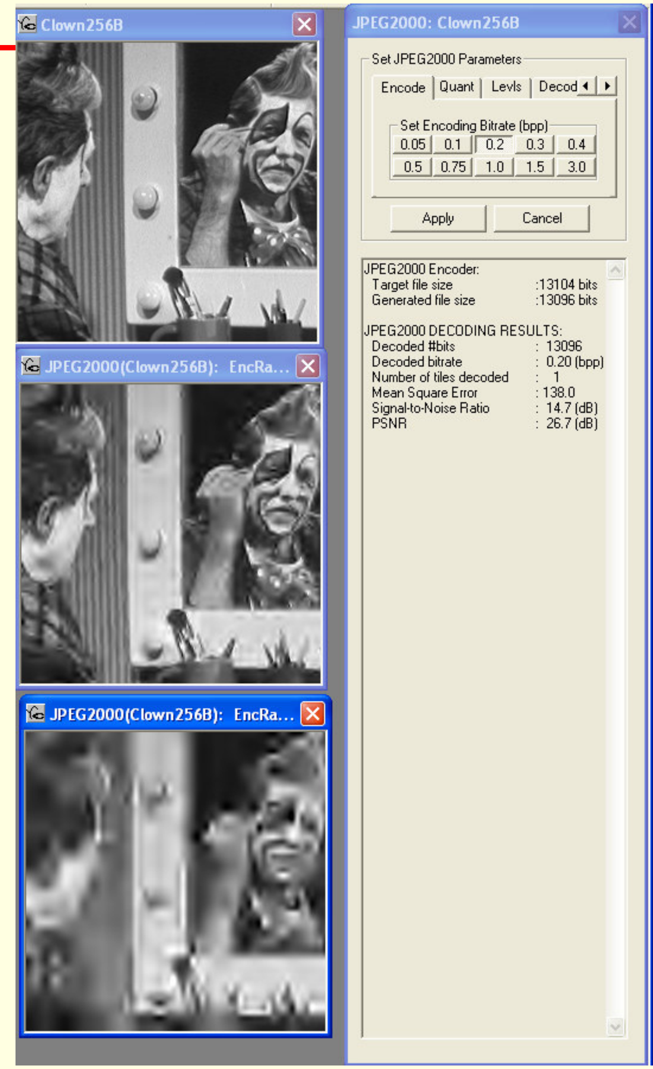




	Parameter	Value	Parameter	Value
Bitmap (8bpp)	**CR**	**15:1**	**CR**	**30:1**
	bpp	0.53	bpp	0.27
	MSE	4.6	MSE	12.2
	SNR	30 dB	SNR	25.7 dB
	PSNR	41.5 dB	PSNR	37.3 dB

VCDemo


About VCDemo
demo
Image Compression
Learning Tool
Copyright (C) 1988-2004 by ICT Group, TU-Delft
Version 5.03 (Sep. 2004)
(I,C)T
Visit our web site at
http://ict.ewi.tudelft.nl/vcdemo
Show Credits and Copyright
Show Welcome Screen
OK


Clown256B
JPEG2000(Clown256B): EncRa...
JPEG2000(Clown256B): EncRa...
JPEG2000: Clown256B
Set JPEG2000 Parameters
Encode
Quant
Levls
Decod
Set Encoding Bitrate (bpp)
0.05
0.1
0.2
0.3
0.4
0.5
0.75
1.0
1.5
3.0
Apply
Cancel
JPEG2000 Encoder:
Target file size :13104 bits
Generated file size :13096 bits
JPEG2000 DECODING RESULTS:
Decoded #bits : 13096
Decoded bitrate : 0.20 (bpp)
Number of tiles decoded : 1
Mean Square Error : 138.0
Signal-to-Noise Ratio : 14.7 (dB)
PSNR : 26.7 (dB)

Comparison of JPEG standards

Lena 512x512, 0.3 b/pixel (VCDemo)

JPEG (DCT)

JPEG 2000 (wavelet transform)

MPEG versions

Intra frame coding (DCT) and INTER frame coding (motion vectors DPCM)

Standard	MPEG-1	MPEG-2	MPEG-4
	1992	1994	1999
Applications	Digital storage on CD's	Digital TV (HDTV)	Video transmission via internet
Resolution	~ 288 x 352 (CIF)	~ 576 x 720 (1152 x 1440)	Depends on video standard
Number of frames per second	25 – 30 frames/s	50-60 frames/s (100-120 frames/s)	up to 60 frames/s
Bit Rate	1.5 Mb/s	~ 4 Mb/s (~ 20 Mb/s)	5 – 64 kb/s 4 Mb/s
Quality	VHS	NTSC/PAL	VHS/NTSC/PAL
Compression ratio	~ 20 - 30	~30-40	hundreds

DVD – Video

A standard based on DVD ROM (4.4 – 17 GB), resolution: PAL 720×576, 24 fps, 12 b/pixel, MPEG-2, aspect ratio 4:3, 16:9, bit rate 4 Mbp (max. 9.8 Mbp), 2 – 8 hours of video.

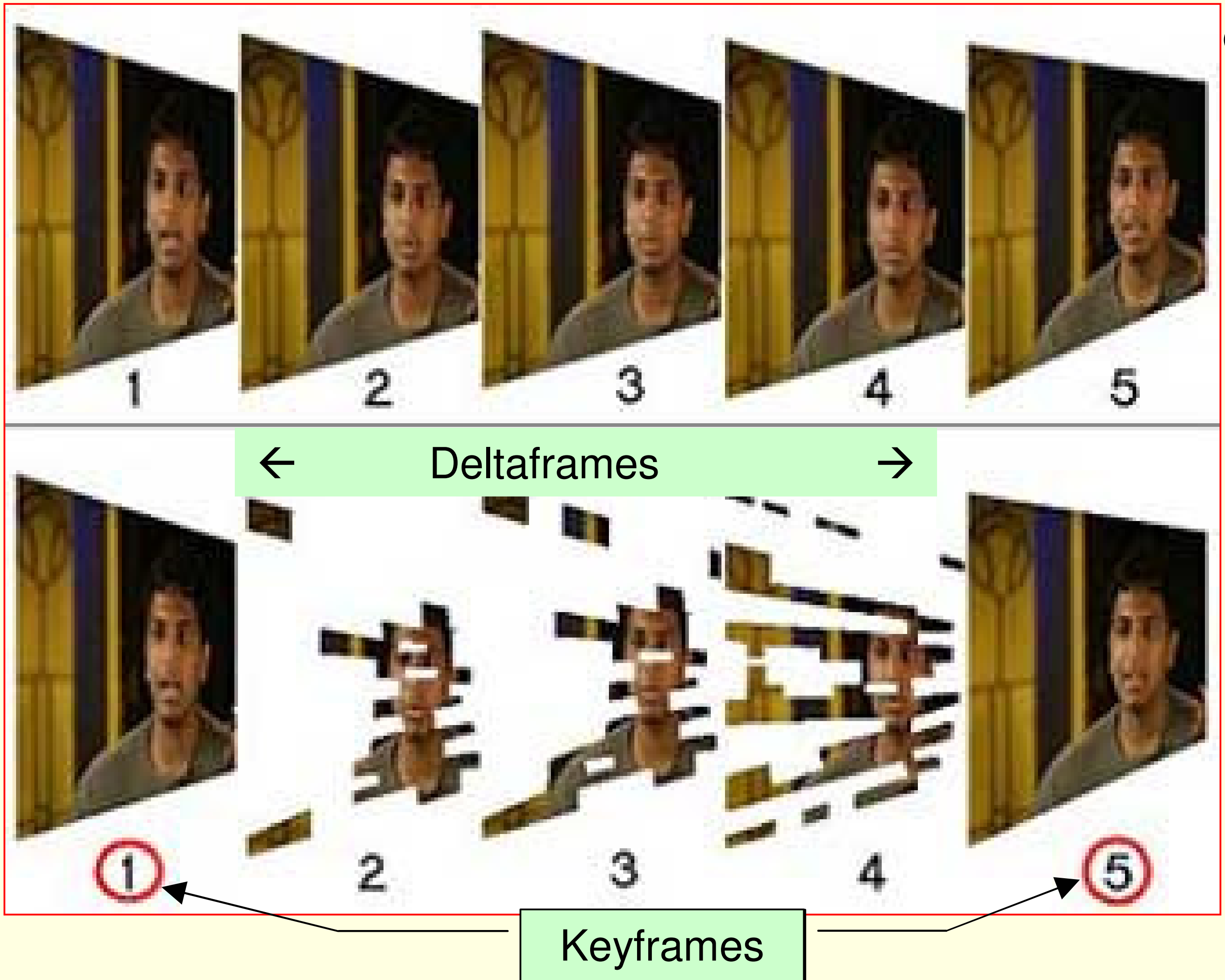
Video CD (VCD)

A standard based on CD ROM (~700 MB), resolution VHS 352×288, 24 fps, MPEG-1, aspect ratio 4:3, 70 min. of video

DivX

A fomat based on MPEG-4, very popular in the internet.

MPEG compression concept



©DivX.com

Comparison of video standards

HDTV 1920x1800

DVD 16:9

http://www.cs.tut.fi/~leopold/Ld/ResolutionComparison/

Comparison of video standards

TV PAL

VHS (VCD)