Set 9: Image Compression

Lecturer Arto Kaarna

Lappeenranta University of Technology (LUT)
School of Engineering Science (LENS)
Machine Vision and Pattern Recognition (MVPR)

Arto.Kaarna@lut.fi
http://www.lut.fi/web/en/school-of-engineering-science/research/machine-vision-and-pattern-recognition

Contents

- Introduction and Background
- Image Coding
- Image Compression
- Wavelet Transform

Introduction

- Data compression is a process of reducing the amount of data required to represent the given information.
- Data redundancy is a central issue in image compression.

$$R_D = 1 - \frac{1}{C_R}, C_R = \frac{n_1}{n_2} = \frac{original}{compressed}$$

- If $n_1 = n_2, C_R = 1 \ and \ R_D = 0$ — *no redundancy*
- If $n_1 \gg n_2, C_R \to \infty \ and \ R_D \to 1$ — *high redundancy*
- If $n_1 \ll n_2, C_R \to 0 \ and \ R_D \to -\infty$ — *undesirable*

- Redundancy types: coding, interpixel (spatial), psychovisual redundancies

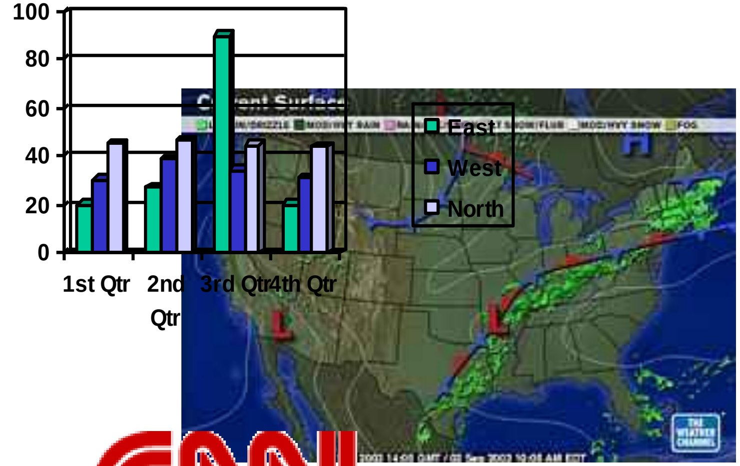
Introduction

1) Storage: local, cloud, services
2) Transmission: wired/wireless, application?
3) Data access, 1990 -> 201x
    - Capacities : 100MB -> TeraBytes
    - Seek time : 15 milliseconds -> 10 milliseconds
    - Transfer rate: 1MB/sec ->1 GB/sec
    - Price $/MB : 4 -> 0.0000283

- Compression improves overall response time in some applications

Background: Digital Data

- Everyday an enormous amount of information need to be stored, processed, and transmitted
- Sources of data: multiple, e.g.
    - Financial data,
    - Reports, Inventory,
    - Cable TV,
    - Online Ordering and tracking



- Processing of Digital Data
    - Coding, compression
    - Data analytics: patterns, classes/clusters

Background: Data from Imaging

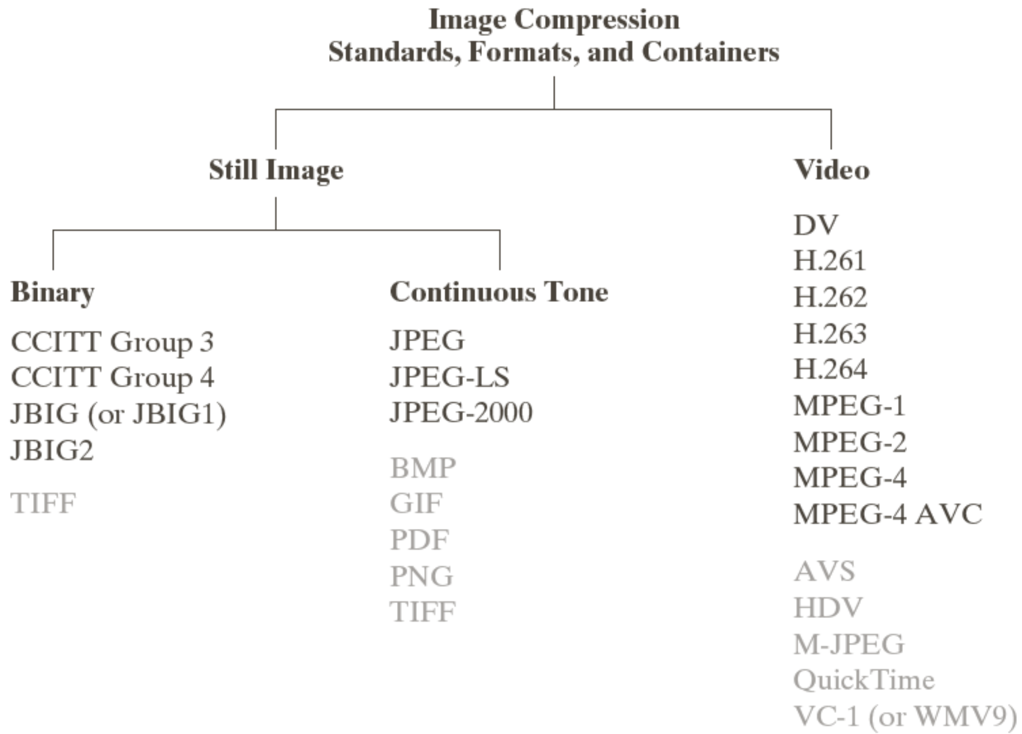
- Sources
  - Digital camera, video camera
  - Image scanner
  - Ultra-sound (US), Computer Tomography (CT), Magnetic resonance image (MRI), digital X-ray (XR), Infrared
  - Etc.
- Application areas, e.g.
  - Medical imaging
  - Remote sensing
  - Consumer applications: digital TV, mobile terminals, etc.

Basic Concepts: Data Rates

- Still images:
  - One page of A4 format at 600 dpi is > 100 MB
  - One color image in a digital camera generates 10-30 MB
  - Scanned 3”×7” photograph at 300 dpi is 30 MB
- Digital TV, HDTV/4K/8K
  - Resolutions: 1920×1080/3840×2160/7680×4320
  - Bitplanes: 8-12, Frame rate: 30-300
- Digital cinema 4K×2K×3×12 bits/pel = 48 MB/frame or 1 GB/sec or 70 GB/min
- Spectral images: $n * m * p * r$, per satellite per day, multiple scanners, continuous scanning

Compression standards

- Still images and video sequencies

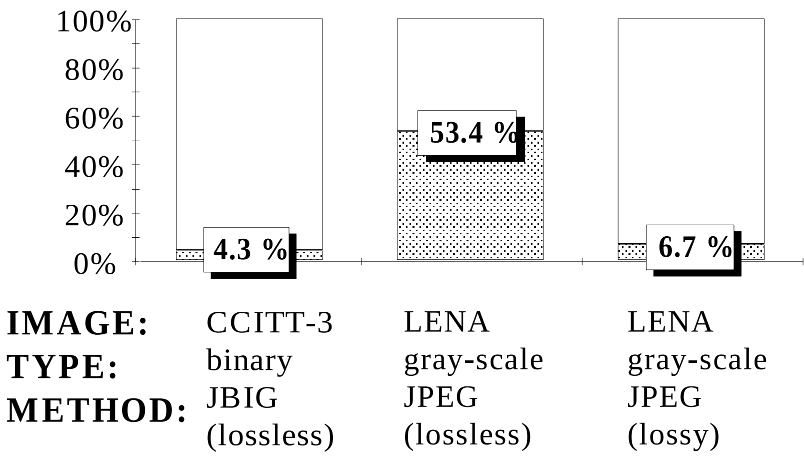


Basic Concepts

- Why do we need to compress images?
- Image types
- Parameters of digital images
- Lossless vs. lossy compression
- Quality measures: rate, distortion, computational, perceptual, etc.
- Compression is an enabling technology for HDTV, mobile communications, etc.

Basic Concepts: Approaches

- Lossless compression: reversible, information preserving
  - text compression algorithms,
  - binary images, palette images.

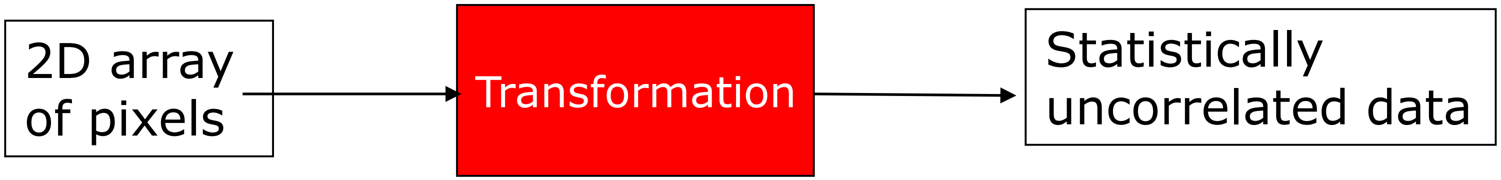


- Lossy compression: irreversible
  - grayscale, color, video.

- Near-lossless compression:
  - medical imaging, remote sensing.

Basic Concepts: Approaches

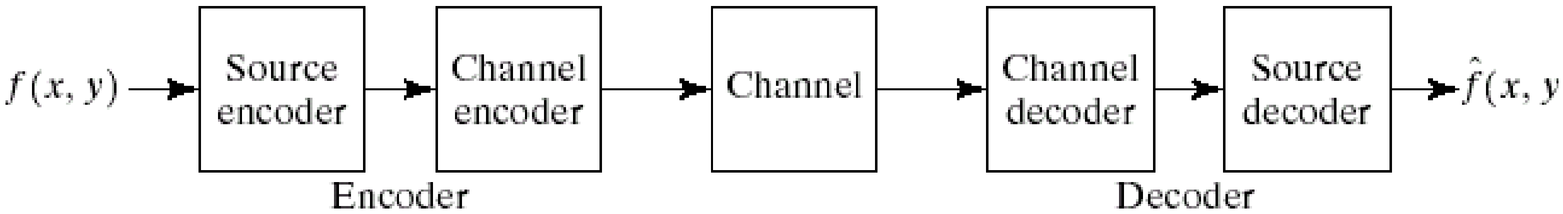
- Data compression is the art and science of representing information in a compact form: we want to remove redundancy from the data



- Data is a sequence of symbols taken from a discrete alphabet
- Still image data: a collection of 2-D arrays (one for each color plane) of values representing intensity (color) of the point in corresponding spatial location (pixel)

Basic Concepts: Approaches

- The coding depends both on the source and for the channel

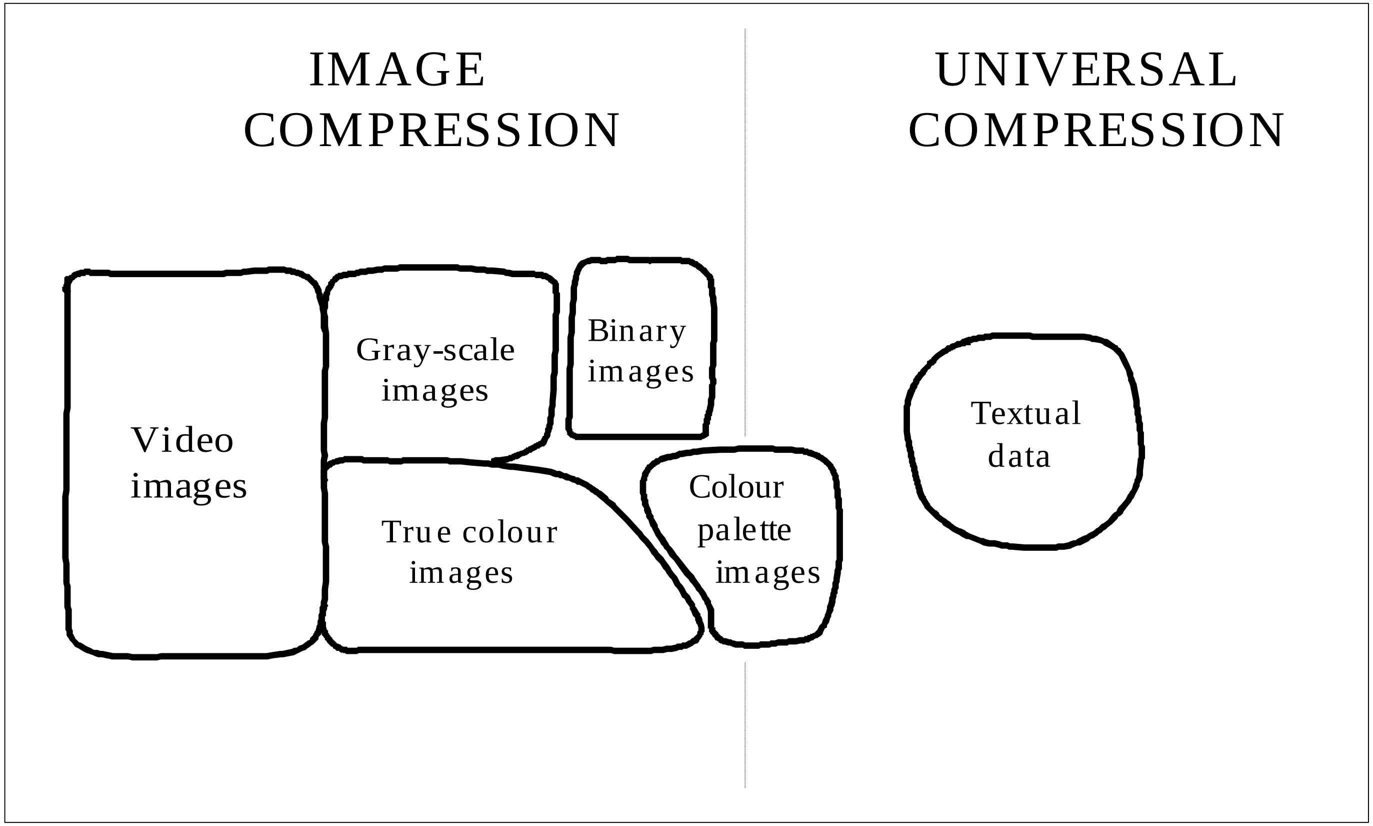


- The source encoder is responsible for removing redundancy (coding, inter-pixel, psycho-visual)
- The channel encoder ensures robustness against channel noise or other errors.

Basic Concepts: Approaches

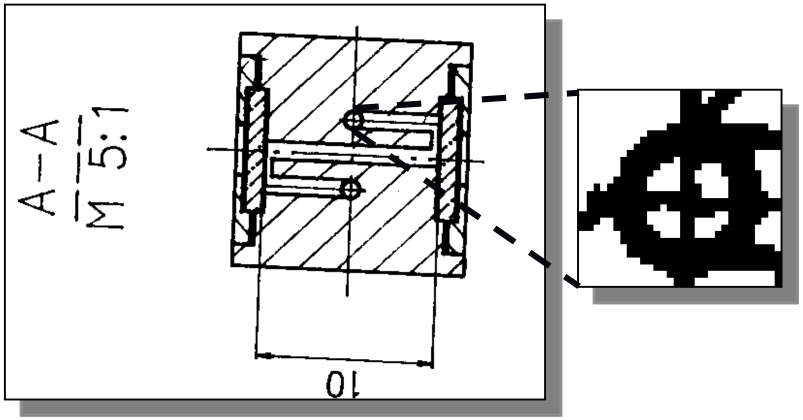
- Statistical redundancy:
  1. Spatial correlation
     a) Local: pixels at neighboring locations have similar intensities
     b) Global: reoccurring patterns
  2. Spectral correlation – between color planes
  3. Temporal correlation – between consecutive frames (for videos, store only the parts that change)
- Tolerance to fidelity:
  1. Perceptual redundancy
  2. Limitation of rendering hardware

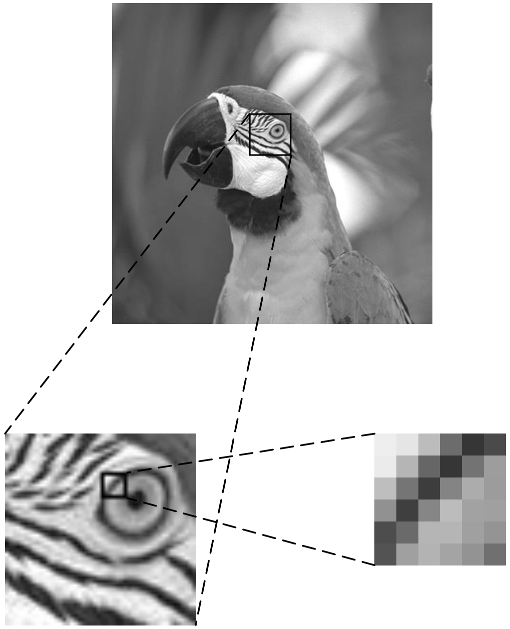
Basic Concepts: Approaches



- Why specific algorithms for images are needed?

Basic Concepts: Image types





Y-U-V components

Basic Concepts: Color

$$Y = 0.3 \cdot R + 0.6 \cdot G + 0.1 \cdot B$$

$$U = B - Y$$

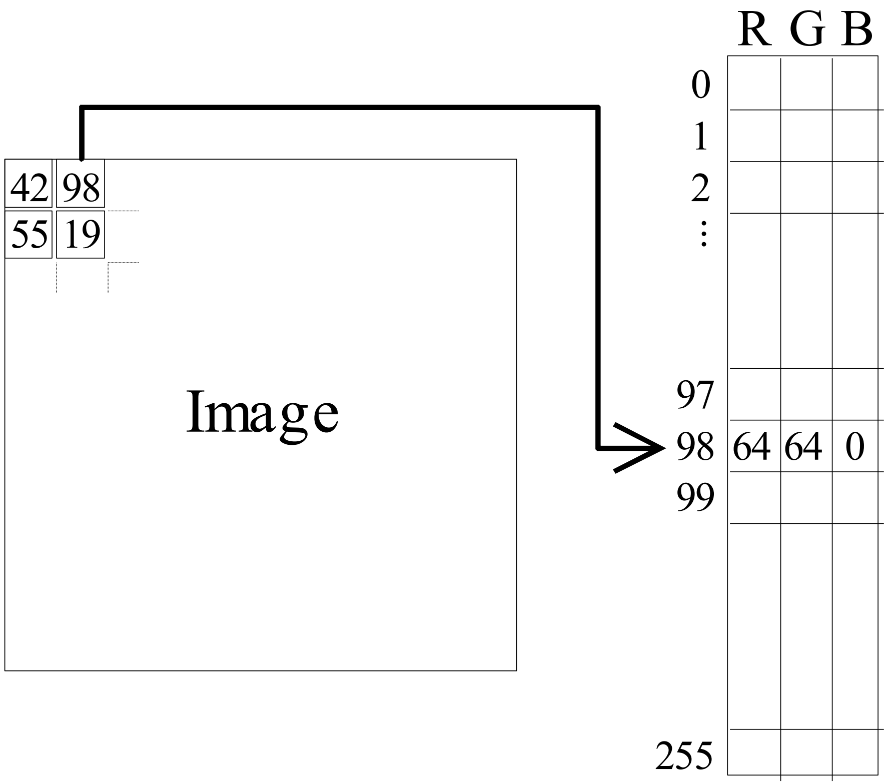
$$V = R - Y$$

R, G, B -- red, green, blue
Y -- the luminance
U,V -- the chrominance components

Most of the information is collected to the *Y* component, while the information content in the *U* and *V* is less.

Look-up Tables in Compression

- Benefits, limitations?



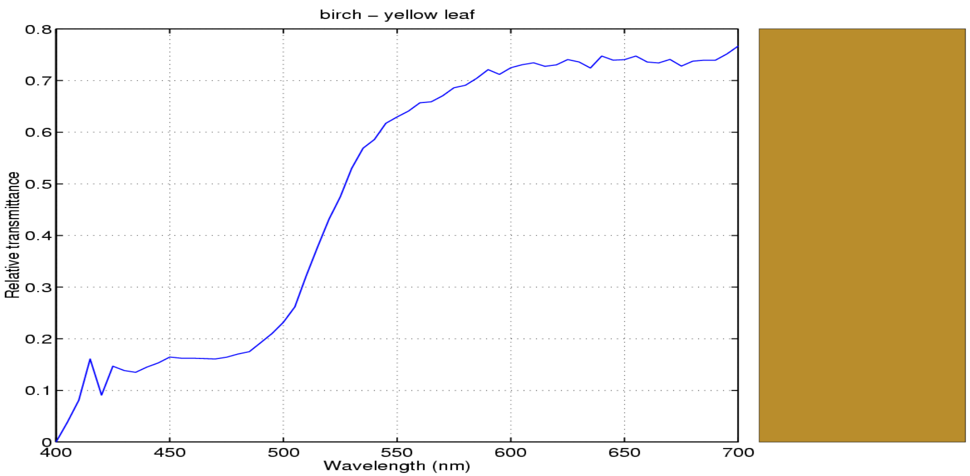
RGB look-up-table

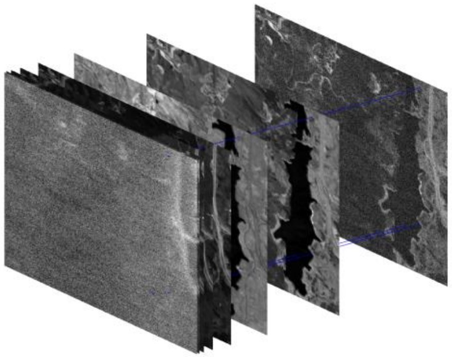
$$[R, G, B] = LUT[index]$$

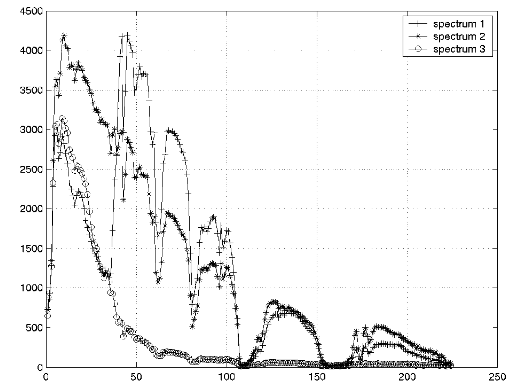
E.g.: $[64{,}64{,}0] = LUT[98]$

Spectral Images

- Spectral image: *n* components according to wavelengths.
- Three components R, G, B => “usual” color image.







Quality Measures

- Bit rate: How much data per pixel?
- Compression ratio: How much smaller?
- Computation time: How fast?
- Distortion: How much error in the presentation?
- Bit rate bits/pixel:

$$\frac{\text{size of the compressed file}}{\text{pixels in the image}} = \frac{D}{N}$$

- Compression ratio:

$$C = \frac{\text{size of the original file}}{\text{size of the compressed file}} = \frac{N \cdot k}{D}$$

k = the number of bits per pixel in the original image
D/N = the bit rate of the compressed image

Quality Measures

- The data redundancy is computed as $R = 1 - \frac{1}{C}$
- If the size of the image is $M * N$, the intensities of the image are r_k, and k:th intensity appears n_k times, then the probability for a symbol is

$$p_k(r_k) = \frac{n_k}{NM}$$

- The average number of bits required for each pixel is

$$L_{avg} = \sum_{k=0}^{L-1} l(r_k) p_k(r_k)$$

where $l(r_k)$ is the number of bits needed to represent r_k

Quality Measures

- Mean average error (MAE): $$\mathrm{MAE} = \frac{1}{N}\sum_{i=1}^{N}\left|y_i - x_i\right|$$
- Mean square error (MSE): $$\mathrm{MSE} = \frac{1}{N}\sum_{i=1}^{N}\left(y_i - x_i\right)^2$$
- Signal-to-noise ratio (SNR): $$\mathrm{SNR} = 10 \cdot \log_{10}\left[\sigma^2 / \mathrm{MSE}\right]$$
  - decibels
- Peak-signal-to-noise ratio (PSNR): $$\mathrm{PSNR} = 10 \cdot \log_{10}\left[A^2 / \mathrm{MSE}\right]$$
  - *A* is the amplitude of the signal, for an 8-bit signal $$A = 2^8 - 1 = 255$$

Quality Measures

- Visual quality metrics
  - Computational metrics developed such that they match with subjects, e.g. visual quality

Rating scale of the Television Allocations Study Organization. (Frendendall and Behrend.)

Value	Rating	Description
1	Excellent	An image of extremely high quality, as good as you could desire.
2	Fine	An image of high quality, providing enjoyable viewing. Interference is not objectionable.
3	Passable	An image of acceptable quality. Interference is not objectionable.
4	Marginal	An image of poor quality; you wish you could improve it. Interference is somewhat objectionable.
5	Inferior	A very poor image, but you could watch it. Objectionable interference is definitely present.
6	Unusable	An image so bad that you could not watch it.

Coding: General Aspects

- Coder and decoder computation complexity
- Memory requirements
- Fixed rate or variable rate
- Error resilience (sensitivity)
- Symmetric or asymmetric
- Decompress at multiple resolutions, decompress at various bit rates; decompress with lost data?
- Standard or proprietary (application based)
- In coding, $C_R = 2$ to 10 can be expected.
- Make use of coding redundancy and inter-pixel redundancy.

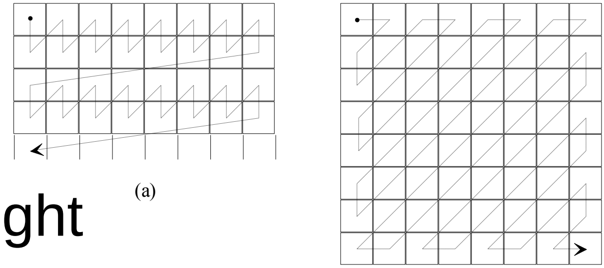
Coding: Role of modeling

- Modeling and coding:
  - How, and in what order the image is processed?
  - What are the symbols (pixels, blocks) to be coded?
  - What is the statistical model of these symbols?
- Requirement:
  - Uniquely decodable: different input => different output.
  - Instantaneously decodable: the symbol can be recognized after its last bit has been received

Coding: Role of modeling

- Segmentation:
  - Local (pixels) or global (fractal compression)
  - Compromise: block coding
- Order of processing:
  - In what order the blocks (or pixels) are processed?
  - In what order are pixels inside the block processed?

Coding: Role of modeling



(a)

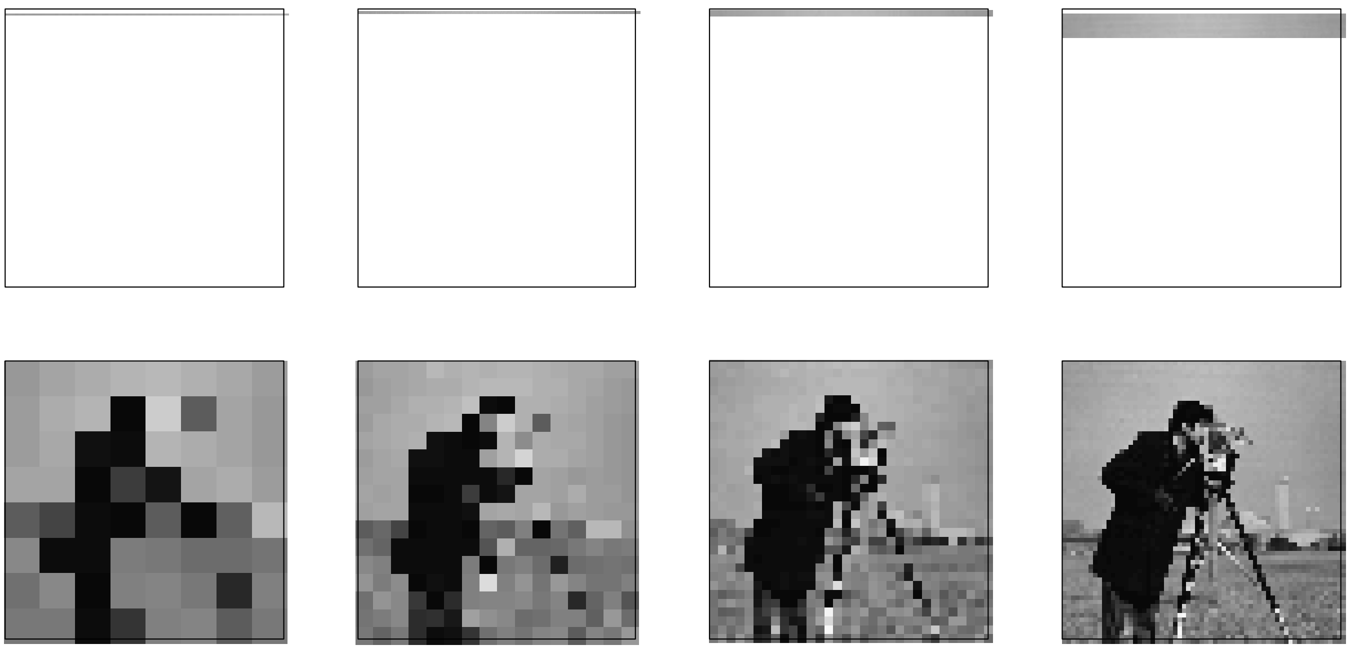
(b)

- Order of processing:
  - Row-major order: top-to-down, left-to-right
- Zigzag scanning:
  - Pixel-wise processing (a)
  - DCT-transformed block (Discrete Cosine Transform) (b)
- Progressive modeling:
  - The quality of an image quality increases gradually as data are received
  - For example in pyramid coding: first the low resolution version, then increasing the resolution

Progressive reconstruction

- Reconstruction quality

0.1 % 0.5 % 2.1 % 8.3 %



Statistical Modeling

- Set of symbols (alphabet) $S = \{s_1, s_2, \dots, s_N\}$
  - N is number of symbols in the alphabet.
- Probability distribution of the symbols: $P = \{p_1, p_2, \dots, p_N\}$
- The (Shannon) **entropy** H (and the average number of bits) of an information source S is defined as

$$H = -\sum_{i=1}^{N} p_i \cdot \log_2(p_i)$$

- The amount of information in symbol s_i, i.e., the number of bits to code or code length for the symbol s_i:

$$H(s_i) = -\log_2(p_i)$$

Statistical Modeling

- Static modeling:
  - Static model (code table)
  - One-pass method: encoding
  - ASCII data: *p('e')=10%, p('t')=8%*
- Semi-adaptive modeling
  - Two-pass method: (1) analysis, (2) encoding
- Adaptive (or dynamic) modeling:
  - Symbol by symbol on-line adaptation during coding/encoding
  - One-pass method: analysis and encoding

Statistical Modeling

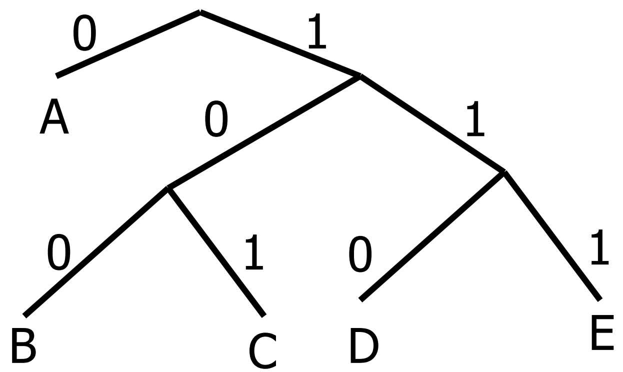
- Context modeling:
  - Spatial dependencies between the pixels
  - For example, what is the most probable symbol after a known sequence of symbols?
- Predictive modeling (for coding prediction errors):
  - Prediction of the current pixel value
  - Calculating the prediction error
  - Modeling the error distribution
  - E.g. differential pulse code modulation (DPCM)

Huffman Coding

- INIT:
  - Put all nodes in an OPEN list and keep it sorted all times according to their probabilities
- REPEAT
  a) From OPEN pick *two* nodes having the *lowest* probabilities, create a parent node of them
  b) Assign the sum of the children’s probabilities to the parent node and inset it into OPEN
  c) Assign code 0 and 1 to the two branches of the tree, and delete the children from OPEN

Huffman Coding

Symbol	p_i	$-log_2(p_i)$	Code	Subtotal
A	15/39	1.38	0	1*15
B	7/39	2.48	100	3*7
C	6/39	2.70	101	3*6
D	6/39	2.70	110	3*6
E	5/39	2.96	111	3*5
Total:				87 bits



Binary tree

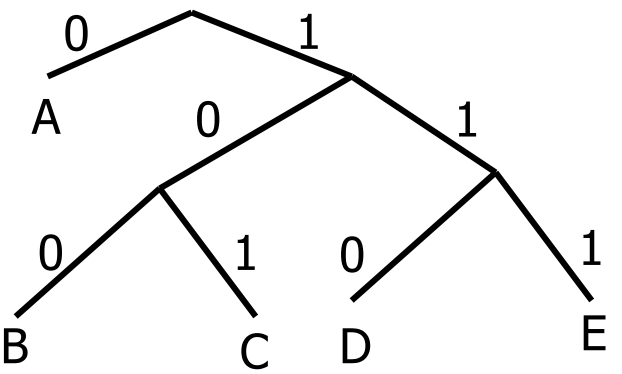
$H = 2.19$ bits

$L = 87/39 = 2.23$ bits

Huffman Coding

- Bit codes: A – 0; B – 100; C – 101; D – 110; E - 111
- Input bit stream (22 bits):

  1000100010101010110111



- Extracted codes:

  100 0 100 0 101 0 101 0 110 111
- Decoded message:

  B A B A C A C A D E

Huffman Coding

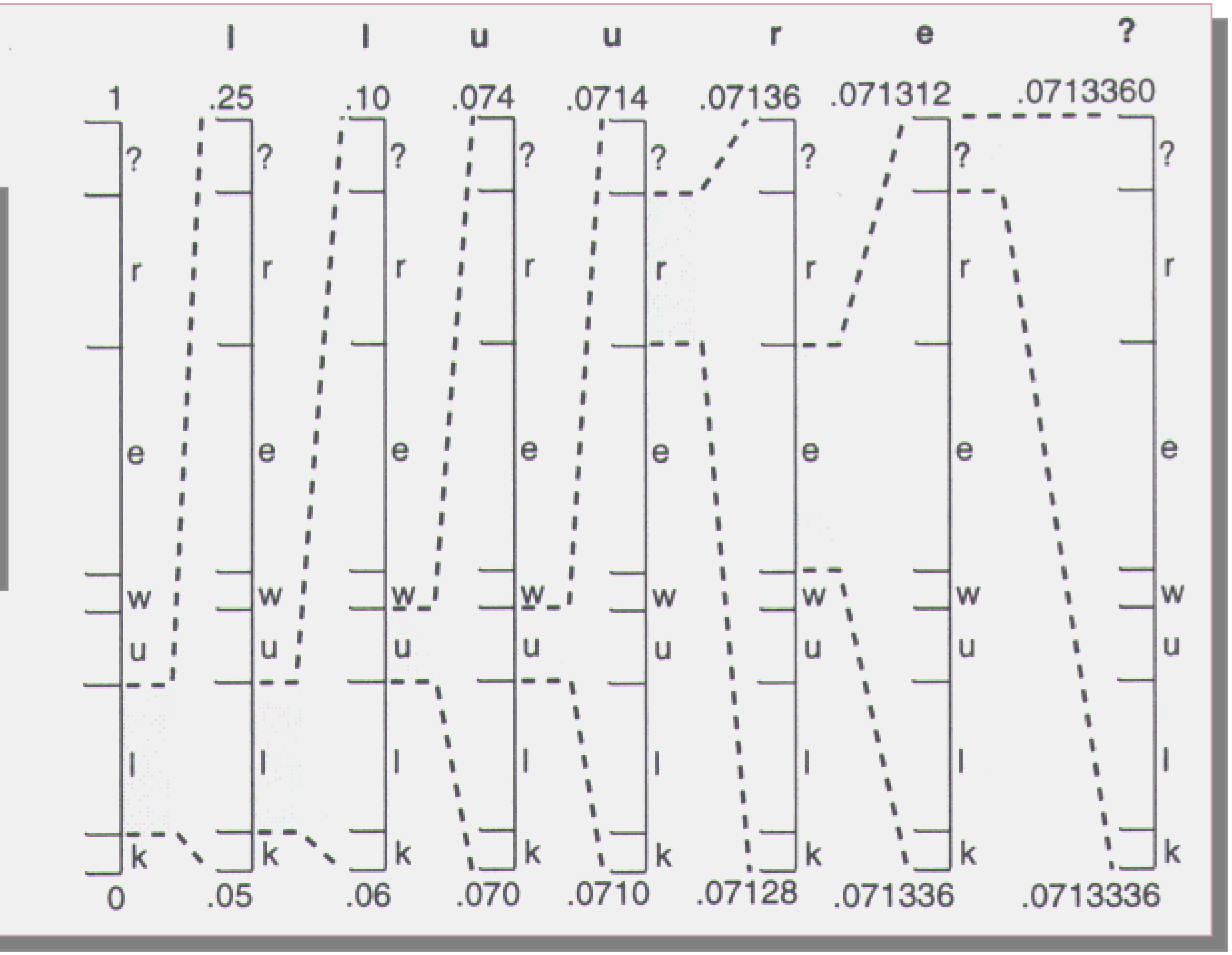
- Optimum code for a given data set requires two passes
- Code construction complexity *O(N log N)*
- Fast lookup table based implementation
- Requires at least one bit per symbol
- Average codeword length is within one bit of zero-order entropy (Tighter bounds are known): $H \leq R \leq H + 1$ bit
- Susceptible to bit errors

Arithmetic Coding

- Alphabet extension (blocking symbols) can lead to coding efficiency
- How about treating entire sequence as one symbol!
- Not practical with Huffman coding
- Arithmetic coding allows you to do precisely this
- Basic idea: map data sequences to sub-intervals in [0,1) with lengths equal to the probability of corresponding sequence
- QM-coder is an arithmetic coding tailored for binary data
  - 1) Huffman coder: $H \leq R \leq H + 1$ bit/pel
  - 2) Block coder: $H_n \leq R_n \leq H_n + {}^1/_n$ bit/pel
  - 3) Arithmetic coder: $H \leq R \leq H + 1$ bit/message

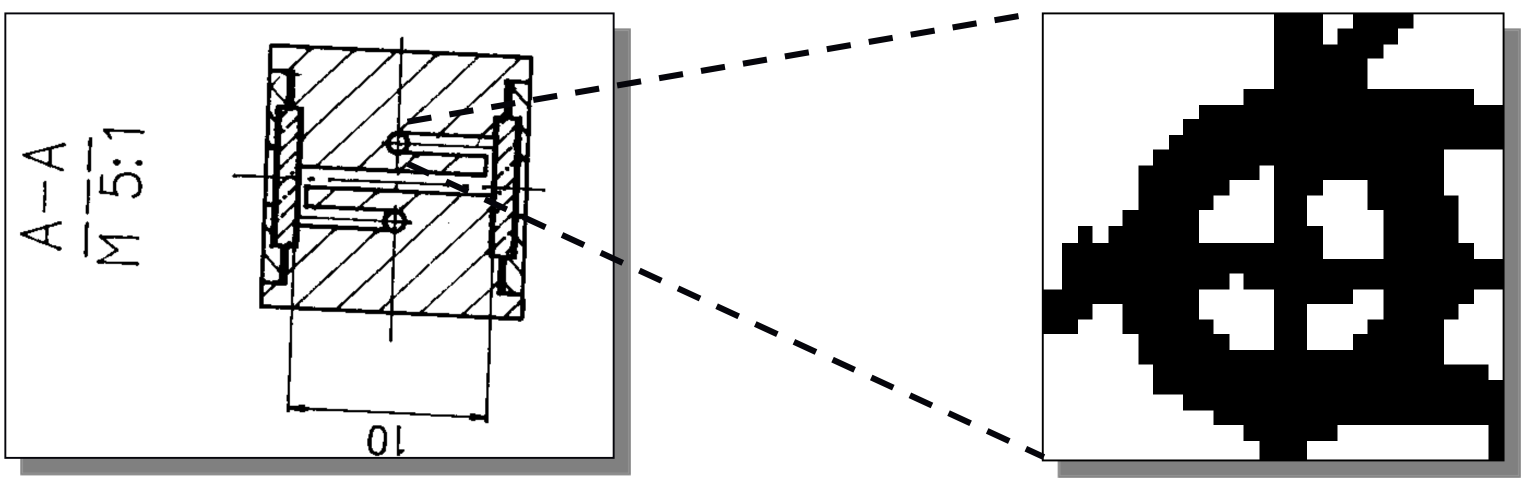
Arithmetic Coding

s_i	p_i	**Subinterval**
k	0.05	[0.00,0.05)
l	0.20	[0.05,0.25)
u	0.10	[0.25,0.35)
w	0.05	[0.35,0.40)
e	0.30	[0.40,0.70)
r	0.20	[0.70,0.90)
?	0.10	[0.90,1.00)



Binary Images

- Binary images consist only of two colors, black and white
- The probability distribution of the alphabet is often very skewed, e.g. *p(white)=0.98*, and *p(black)=0.02*
- Moreover, the images usually have large homogenous areas of the same color

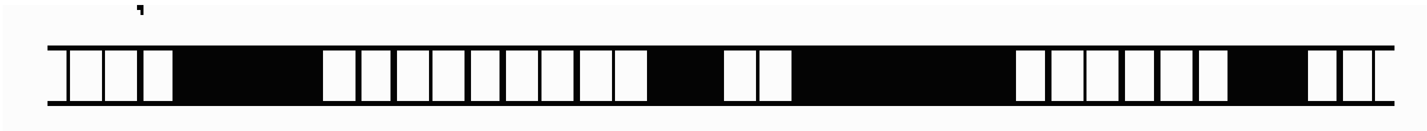


Binary Images

- Run-length encoding
- Predictive encoding
- CCITT group 3 and group 4 standards
  - READ code (2D, Relative Element Address Designate)
- Block transform coding
- JBIG, JBIG2 (Joint Bilevel Image Experts Group)
  - Standard by CCITT and ISO
  - Context-based compression pixel by pixel
  - QM-coder (arithmetic coder)

Binary Images: RLE

- Pre-processing method, good when one symbol occurs with high probability or when symbols are dependent
- Count how many repeated symbol occur
- Source ’symbol’ = length of run

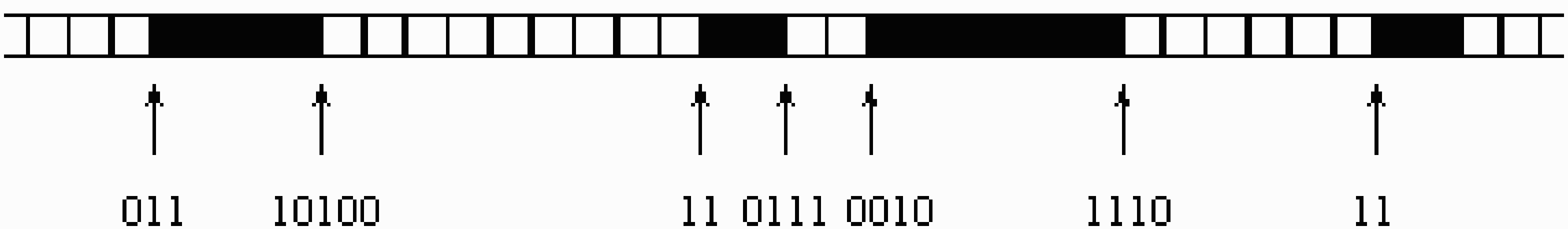


- Example: …, 4b, 9w, 2b, 2w, 6b, 6w, 2b, ...

Binary Images: RLE

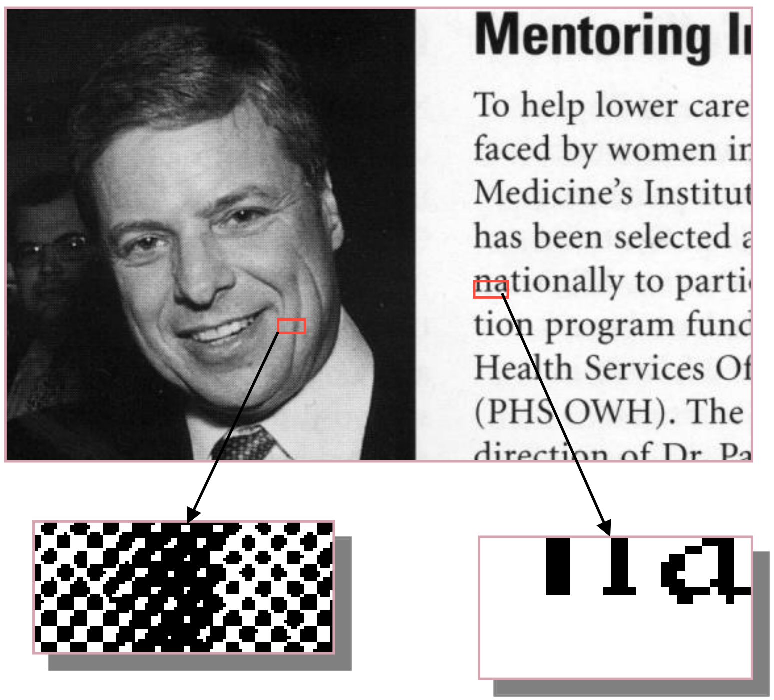
- Huffman code table

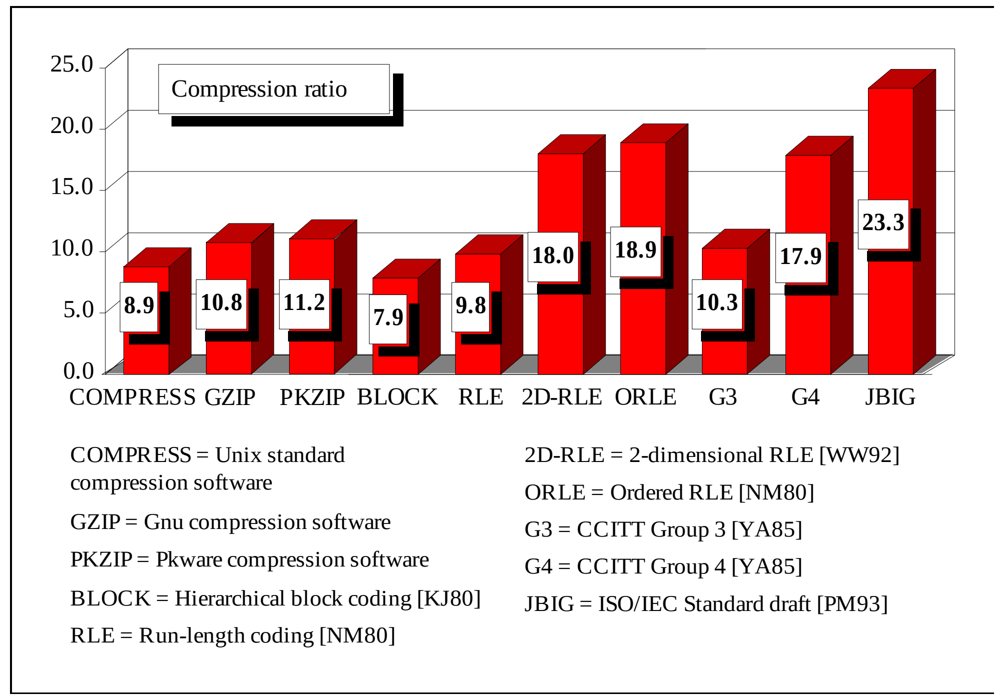
n	white runs	black runs
0	00110101	0000110111
1	000111	010
2	0111	11
3	1000	10
4	1011	011
5	1100	0011
6	1110	0010
7	1111	00011
8	10011	000101
9	10100	000100
10	00111	0000100
11	01000	0000101
12	001000	0000111



JBIG2

- Bilevel (binary) documents, like fax documents.
- Both graphic (line art) and pictures (halftone).





Continuous Tone Images

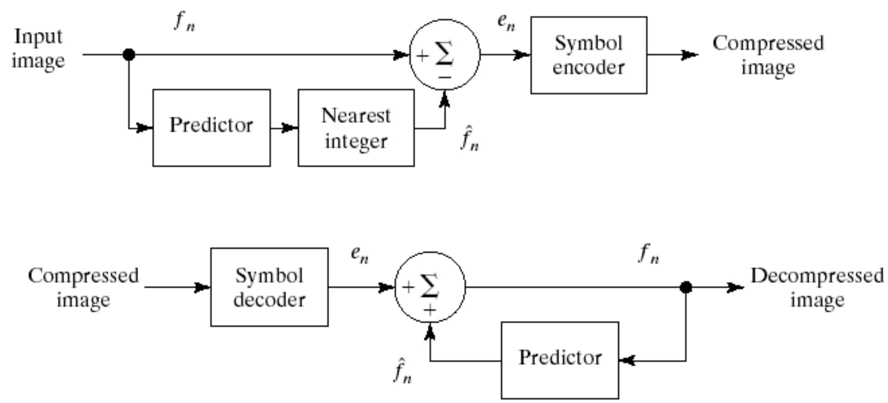
- Lossless and near-lossless compression
  - Bit-plane coding: to bit-planes of a grayscale image
    - A series of binary images
  - Lossless and near-lossless JPEG
    - Joint Photographic Experts Group: https://jpeg.org/index.html
    - Pixel by pixel by predicting the current pixel on the basis of the neighboring pixels
    - Prediction errors coded by Huffman or arithmetic coding (QM-coder)

Continuous Tone Images

- Prediction to remove inter-pixel redundancy

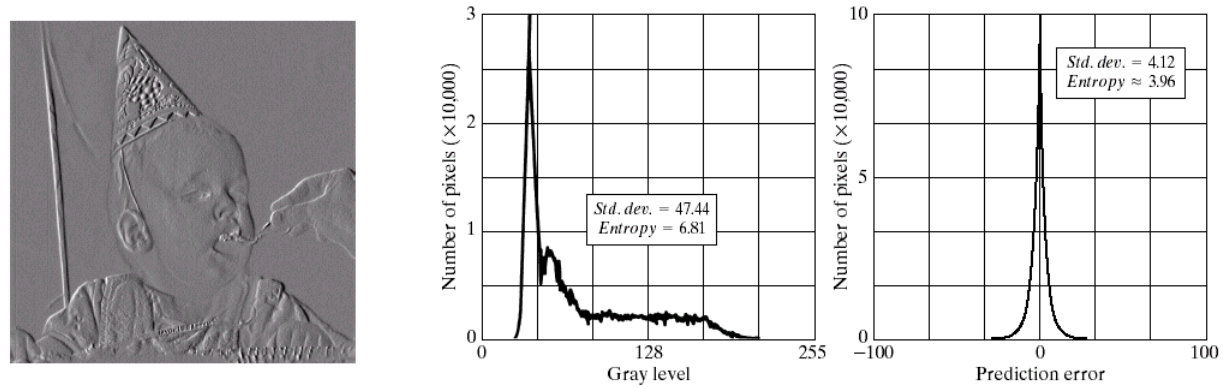
a
b

FIGURE 8.19 A lossless predictive coding model: (a) encoder; (b) decoder.



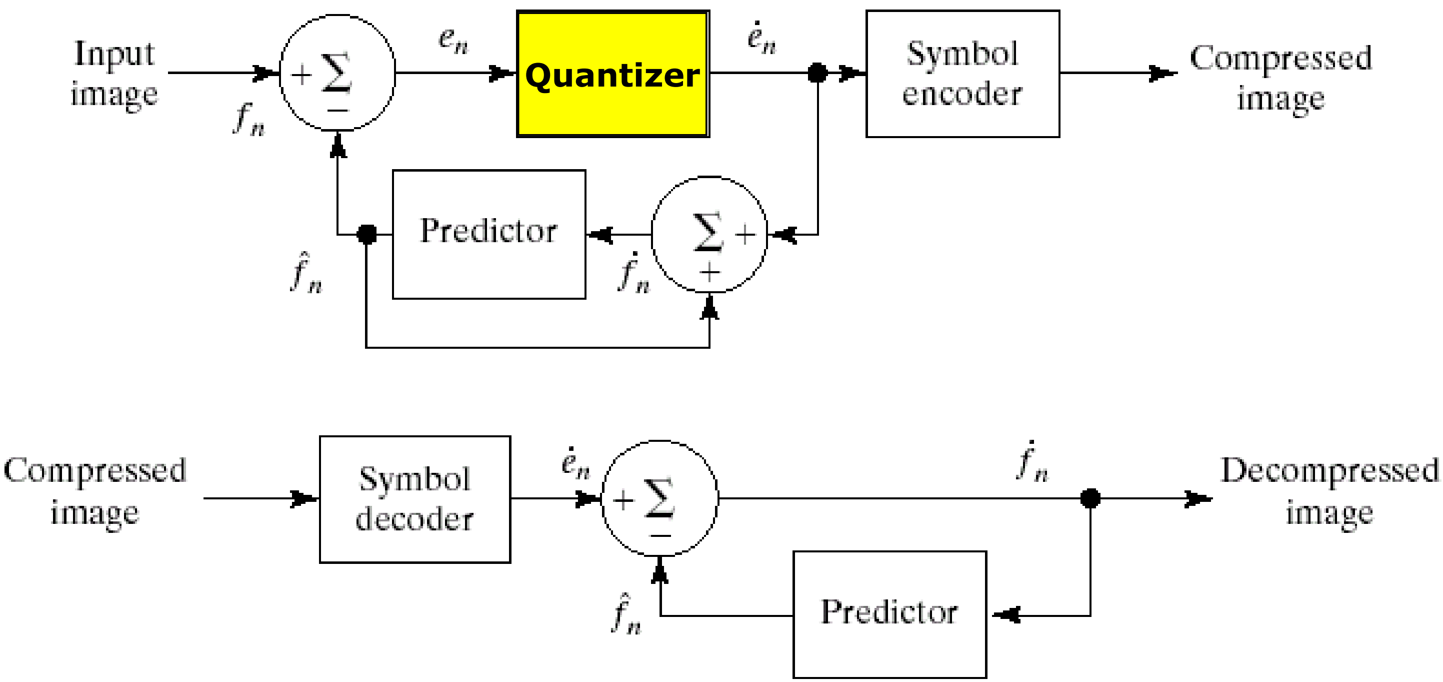
a
b c

FIGURE 8.20 (a) The prediction error image resulting from Eq. (8.4-9). (b) Gray-level histogram of the original image. (c) Histogram of the prediction error.



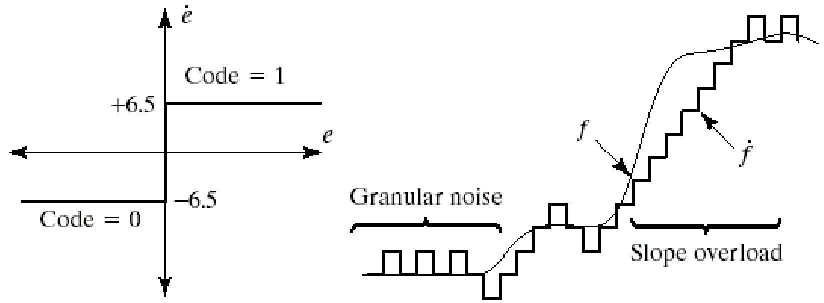
Continuous Tone Images

- Prediction in lossy compression



Continuous Tone Images

- Delta modulation

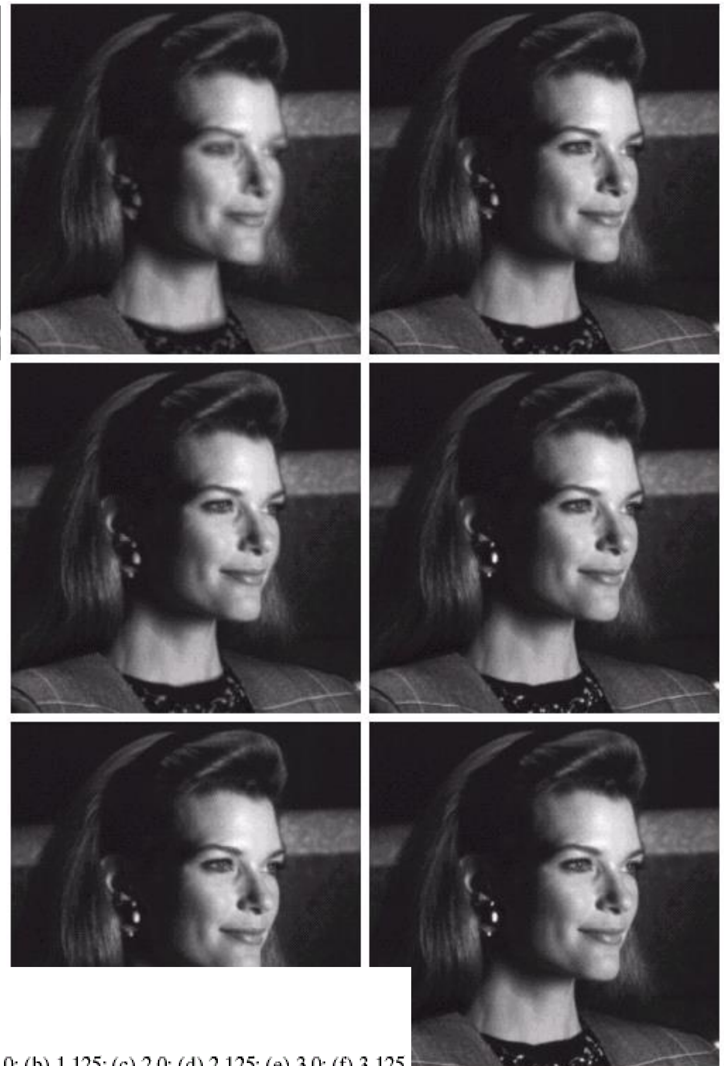


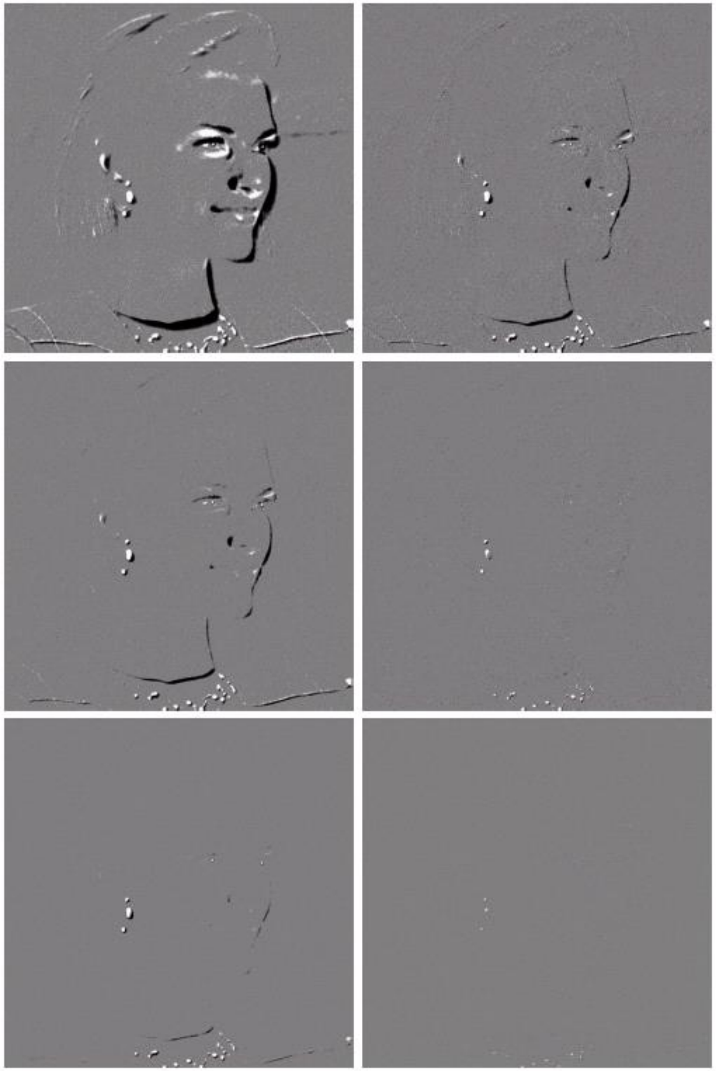
a b
c

FIGURE 8.22 An example of delta modulation.

Input		Encoder				Decoder		Error
n	f	$\hat{f}$	e	$\dot{e}$	$\dot{f}$	$\hat{f}$	$\dot{f}$	$[f - \dot{f}]$
0	14	—	—	—	14.0	—	14.0	0.0
1	15	14.0	1.0	6.5	20.5	14.0	20.5	–5.5
2	14	20.5	–6.5	–6.5	14.0	20.5	14.0	0.0
3	15	14.0	1.0	6.5	20.5	14.0	20.5	–5.5
.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.
14	29	20.5	8.5	6.5	27.0	20.5	27.0	2.0
15	37	27.0	10.0	6.5	33.5	27.0	33.5	3.5
16	47	33.5	13.5	6.5	40.0	33.5	40.0	7.0
17	62	40.0	22.0	6.5	46.5	40.0	46.5	15.5
18	75	46.5	28.5	6.5	53.0	46.5	53.0	22.0
19	77	53.0	24.0	6.5	59.6	53.0	59.6	17.5
.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.

Continuous Tone Images, DPCM



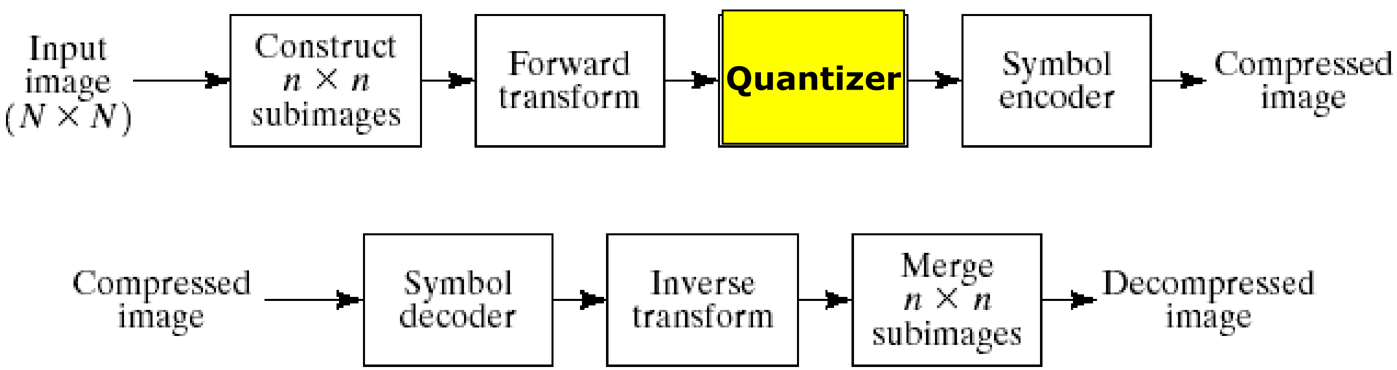


a b
c d
e f

FIGURE 8.26 DPCM result images: (a) 1.0; (b) 1.125; (c) 2.0; (d) 2.125; (e) 3.0; (f) 3.125 bits/pixel.

Continuous Tone Images

- A reversible linear transform (such as Fourier Transform) is used to map the image into a set of transform coefficients; These coefficients are then quantized and coded.
- The goal of transform coding is to decorrelate pixels and pack information into small number of transform coefficients.

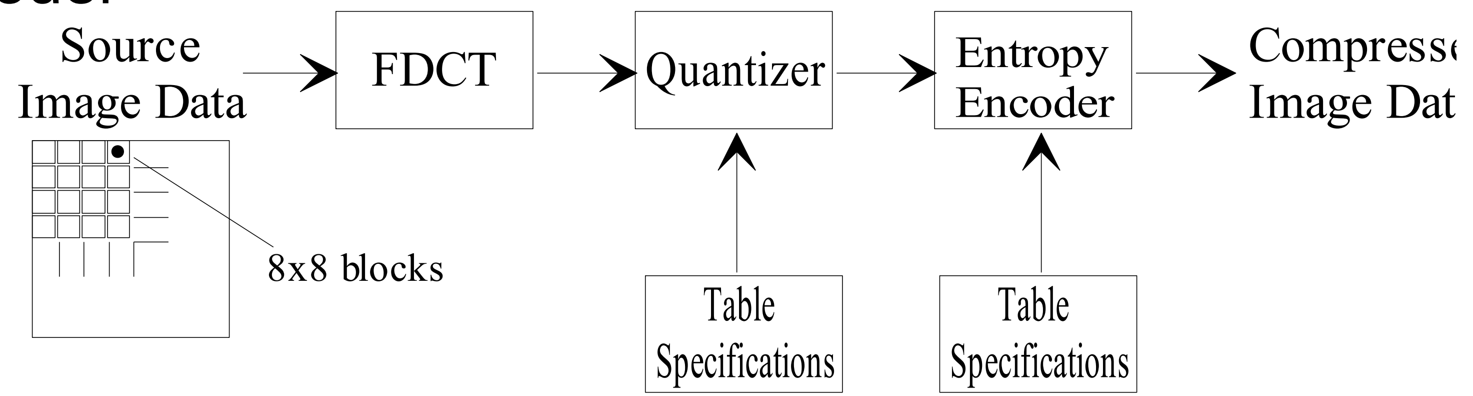


Continuous Tone Images: JPEG

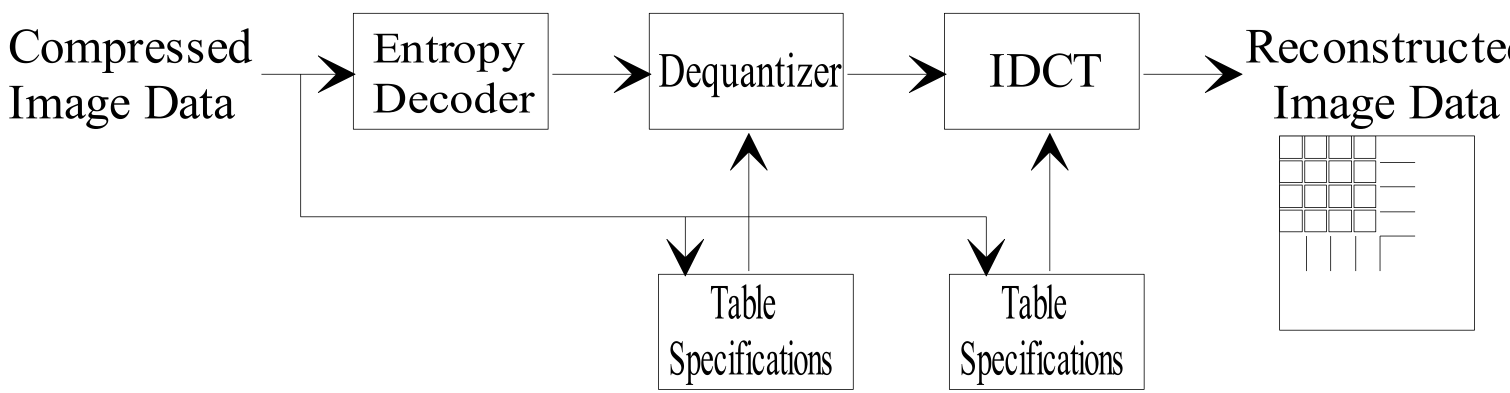
- Vector quantization: codebooks
- JPEG (Joint Photographic Experts Group)
- Lossy coding of continuous tone still images, both color and grayscale
- Based on Discrete Cosine Transform (DCT):
  1) Image is divided into block N×N.
  2) The blocks are transformed with 2-D DCT.
  3) DCT coefficients are quantized.
  4) The quantized coefficients are encoded.

Continuous Tone Images: JPEG

- Encoder

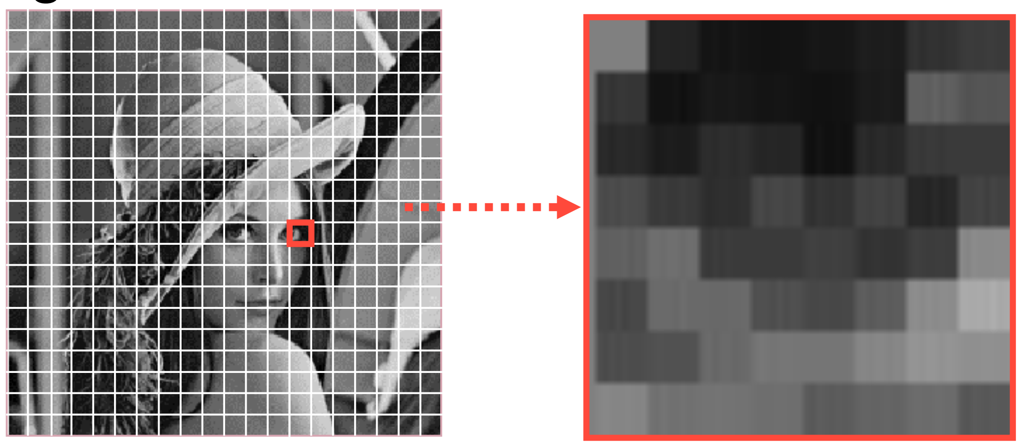


- Decoder

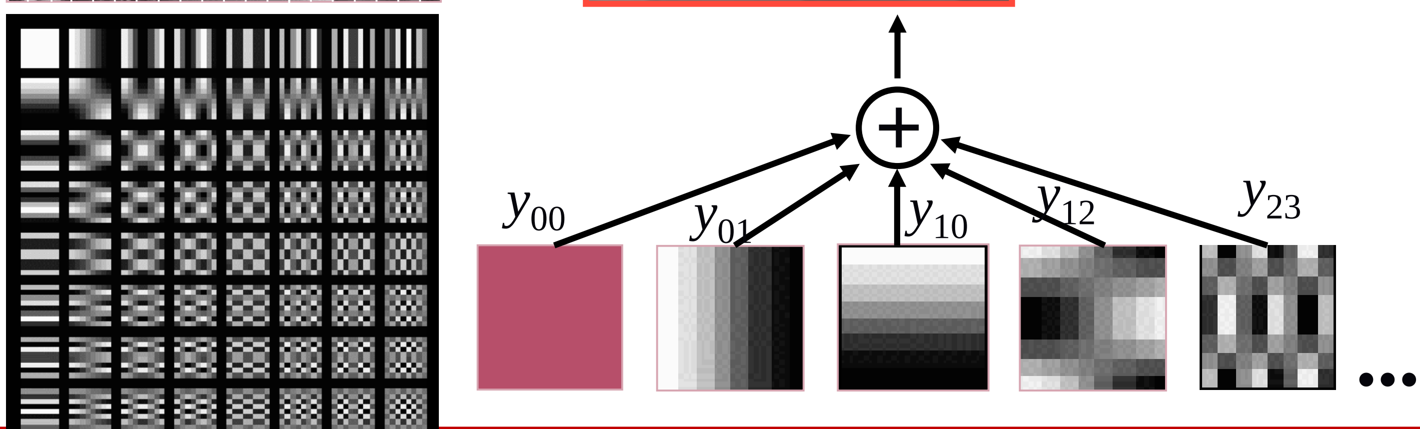


Continuous Tone Images: JPEG

- Input image

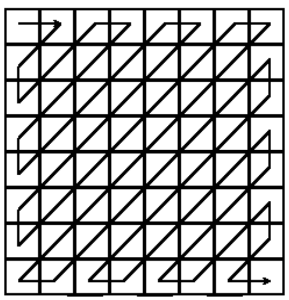


- DCT



Continuous Tone Images: JPEG

- Converting a 2-D matrix into a 1-D array, so that the frequency (horizontal and vertical) increases in this order and the coefficients variance are decreasing in this order.



0	1	5	6	14	15	27	28
2	4	7	13	16	26	29	42
3	8	12	17	25	30	41	43
9	11	18	24	31	40	44	53
10	19	23	32	39	45	52	54
20	22	33	38	46	51	55	60
21	34	37	47	50	56	59	61
35	36	48	49	57	58	62	63

DC Direct current

Matlab: y=dct(x);

AC Alternating current

$$\hat{T}(u,v) = round\left[\frac{T(u,v)}{Z(u,v)}\right]$$

Original block									Transformed block							
139	144	149	153	155	155	155	155		235.6	-1.0	-12.1	-5.2	2.1	-1.7	-2.7	1.3
144	151	153	156	159	156	156	156		-22.6	-17.5	-6.2	-3.2	-2.9	-0.1	0.4	-1.2
150	155	160	163	158	156	156	156		-10.9	-9.3	-1.6	1.5	0.2	-0.9	-0.6	-0.1
159	161	162	160	160	159	159	159		-7.1	-1.9	0.2	1.5	0.9	-0.1	0.0	0.3
159	160	161	162	162	155	155	155		-0.6	-0.8	1.5	1.6	-0.1	-0.7	0.6	1.3
161	161	161	161	160	157	157	157		1.8	-0.2	1.6	-0.3	-0.8	1.5	1.0	-1.0
162	162	161	163	162	157	157	157		-1.3	-0.4	-0.3	-1.5	-0.5	1.7	1.1	-0.8
162	162	161	161	163	158	158	158		-2.6	1.6	-3.8	-1.8	1.9	1.2	-0.6	-0.4

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

JPEG vs. JPEG2000

- Fractal coding
- Pyramid coding
- Wavelet transform, e.g. JPEG 2000

JPEG: 0.25 bpp

JPEG2000: 0.25 bpp

Continuous Wavelet Transform

- CWT is analysing signal $f(x)$ as with a transform

$$W_{\psi}(s,t) = \int_{-\infty}^{\infty} f(x)\psi_{s,t}(x)dx$$

where $\psi_{s,t}(x) = \frac{1}{\sqrt{s}}\psi\left(\frac{x-t}{s}\right)$

- The inverse transform has the prefect reconstruction feature

$$f(x) = \frac{1}{C_{\psi}} \int_{0}^{\infty} \int_{-\infty}^{\infty} W_{\psi}(s,t) \frac{\psi_{s,t}(x)}{s^2} dtds$$

with the admissibility condition $C_{\psi} = \int_{-\infty}^{\infty} \frac{|\Psi(\mu)|^2}{|\mu|} d\mu < \infty$

Discrete Wavelet Transform

- The signal $f(x)$ as a combination of expansion functions is

$$f(x) = \sum_{k} \alpha_k \varphi_k(x)$$

  where $\varphi_k(x)$ are the expansion functions.
- In DWT the $\varphi_k(x)$ discretized as

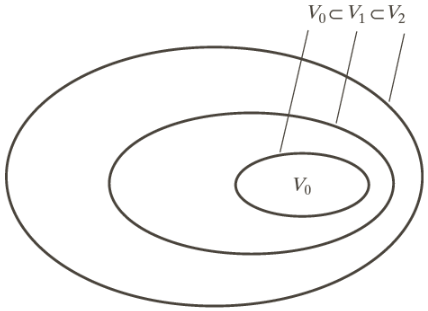
$$\varphi_{j,k}(x) = 2^{\frac{j}{2}} \varphi\left(2^j x - k\right), \qquad j,k \in Z, \varphi(x) \in L^2(R)$$

- E.g. Haar scaling function

$$\varphi(x) = \begin{cases} 1, & 0 \leq x < 1 \\ 0, & \text{otherwise} \end{cases}$$

Discrete Wavelet Transform

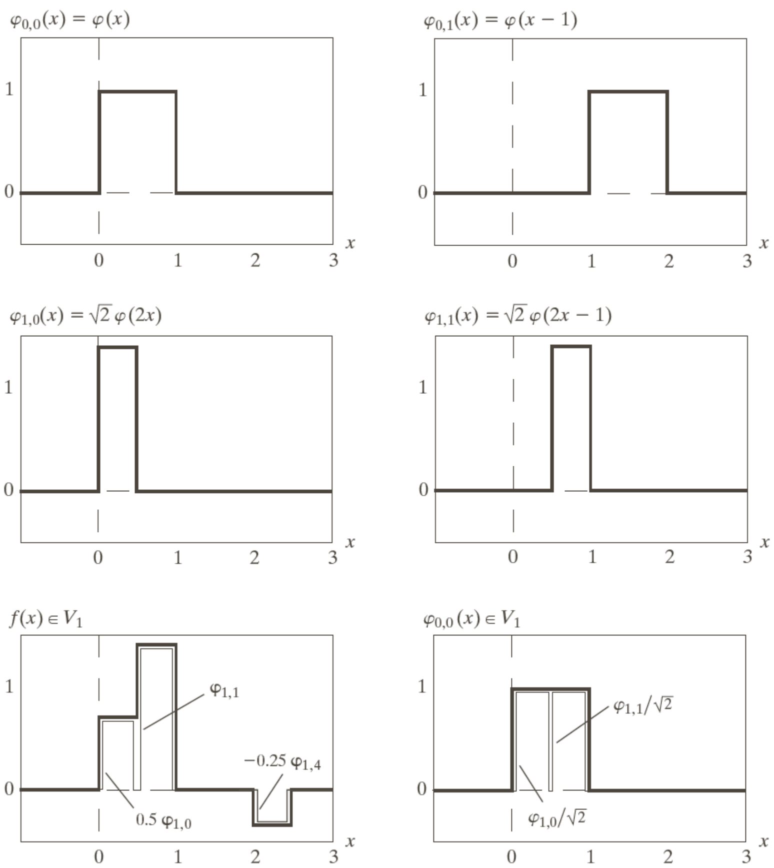
- Haar transform, scaling functions in V_0 and V_1



- The wavelet function $\psi(x)$ describes the differences

  $V_{j+1} = V_j \oplus W_j$

  $\langle \varphi_{j,k}, \psi_{j,l}(x) \rangle = 0$



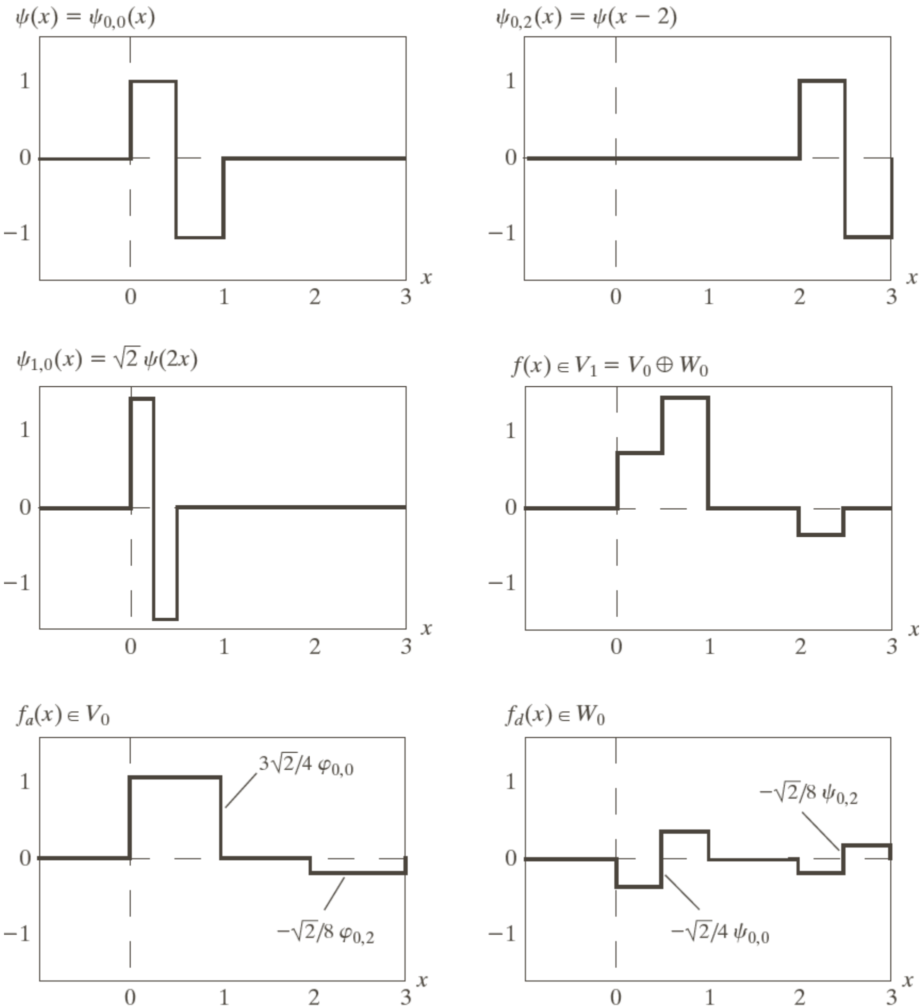
Discrete Wavelet Transform

- E.g. Haar wavelet function is

$$\psi(x)=\begin{cases}1, & 0\le x<0.5\\ -1, & 0.5\le x<1\\ 0, & \text{elsewhere}\end{cases}$$

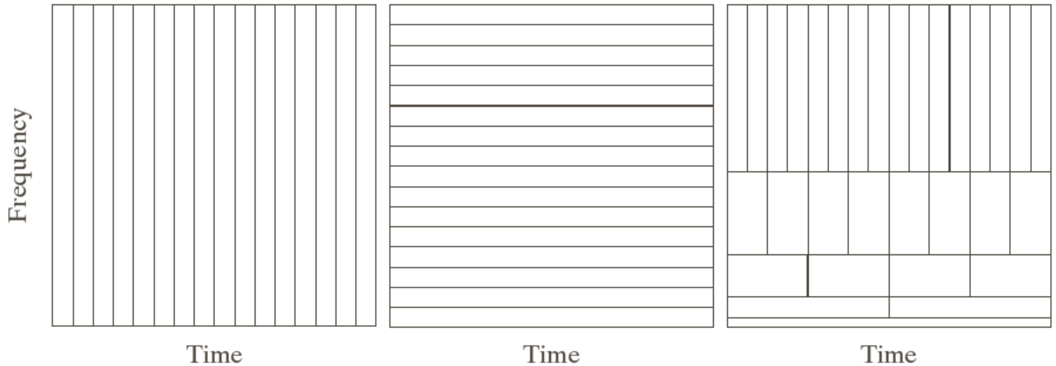
- As the result

$$f(x)=f_a(x)+f_d(x)$$

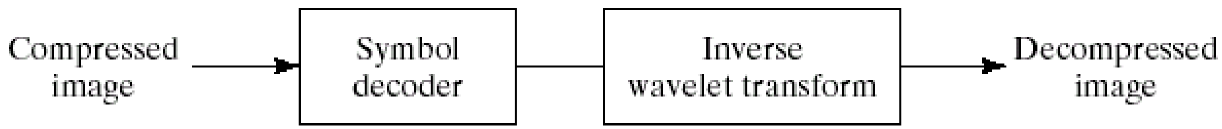


Wavelet Compression

- FT vs. WT: time-frequency tilings (time domain, FD, WD)

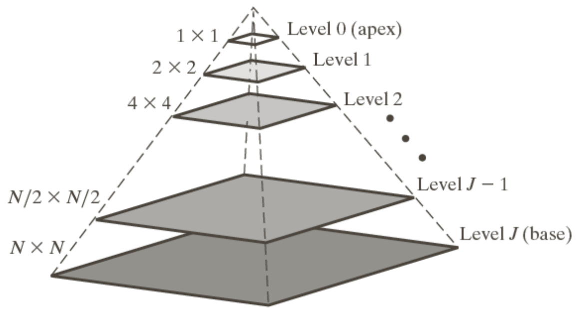


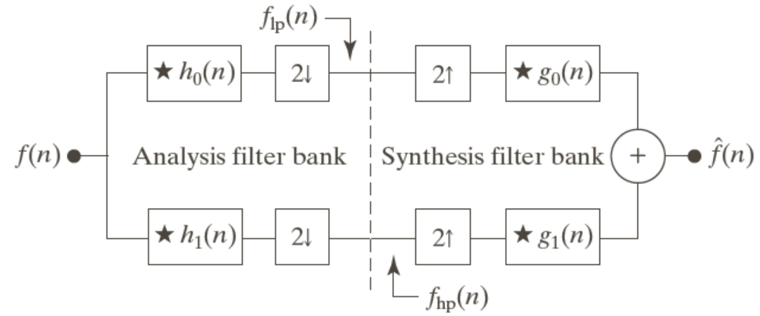
- In practice

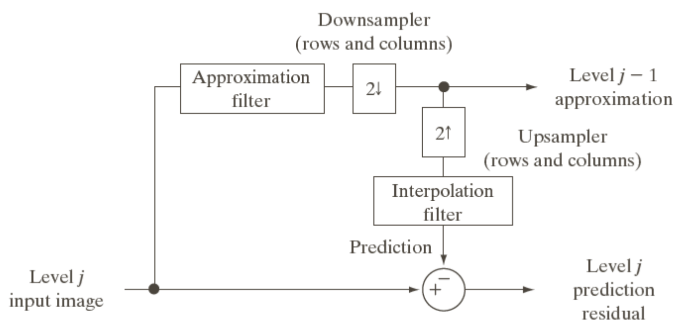


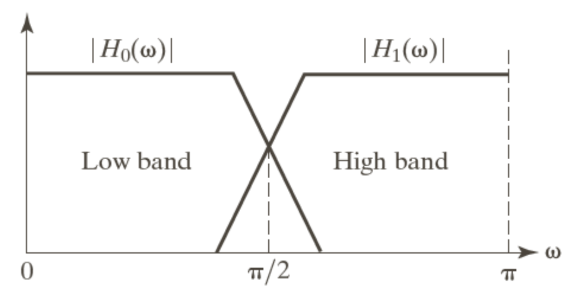
Wavelet Compression

- Wavelet transform is constructing an approximating image pyramid and differences (details) between them

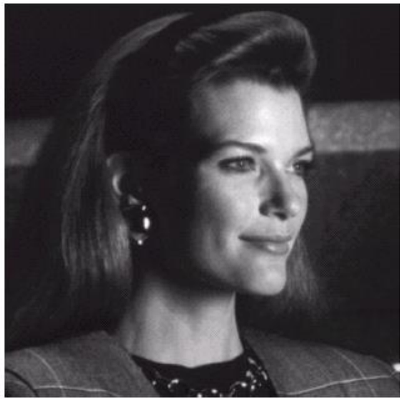








Wavelet Compression



Put a pixel in each quadrant-→ No size change

1	2
3	4

a	b
c	d

a =(x1+x2+x3+x4)/4

b =(x1+x2-x3-x4)/4

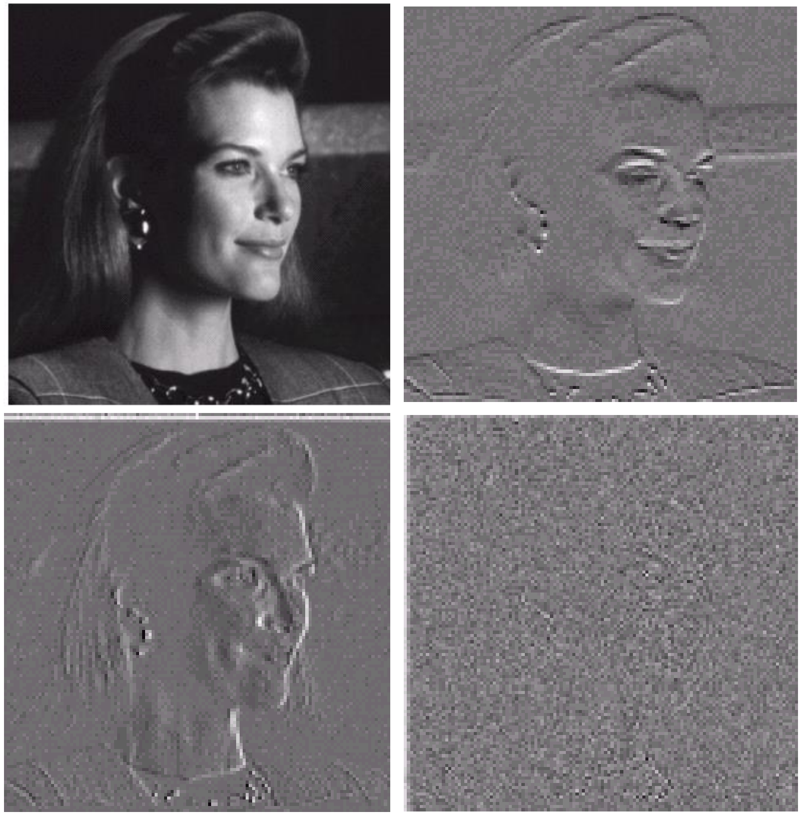
c =(x1+x3-x2-x4)/4

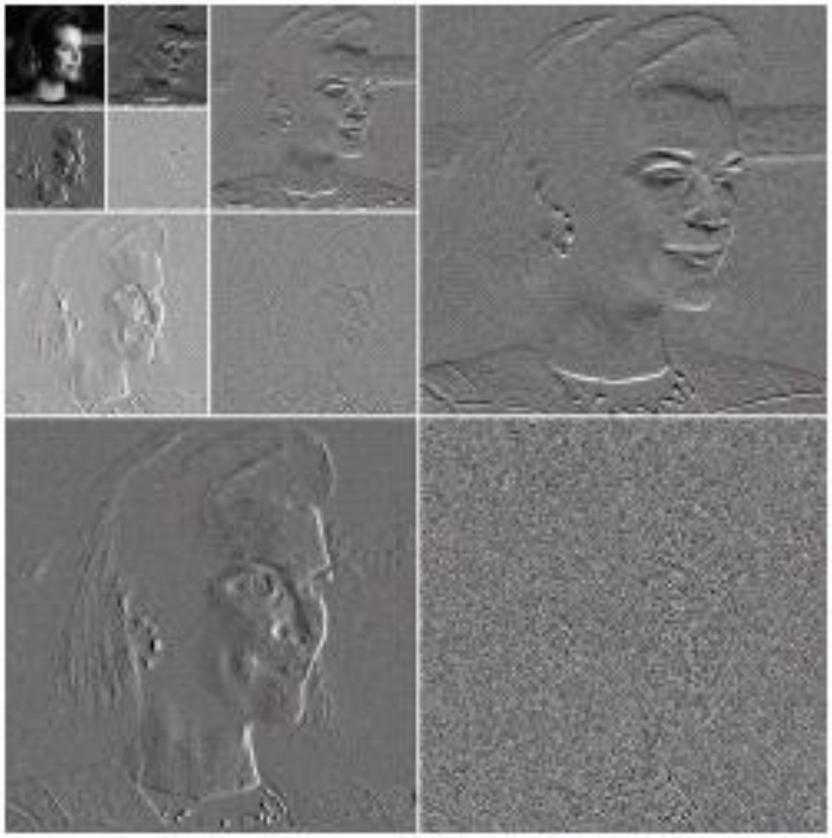
d =(x1+x4-x2-x3)/4

$$\begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} = \frac{1}{4} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & 1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} \begin{bmatrix} x1 \\ x2 \\ x3 \\ x4 \end{bmatrix}$$

$$\begin{bmatrix} x1 \\ x2 \\ x3 \\ x4 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & 1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix}$$

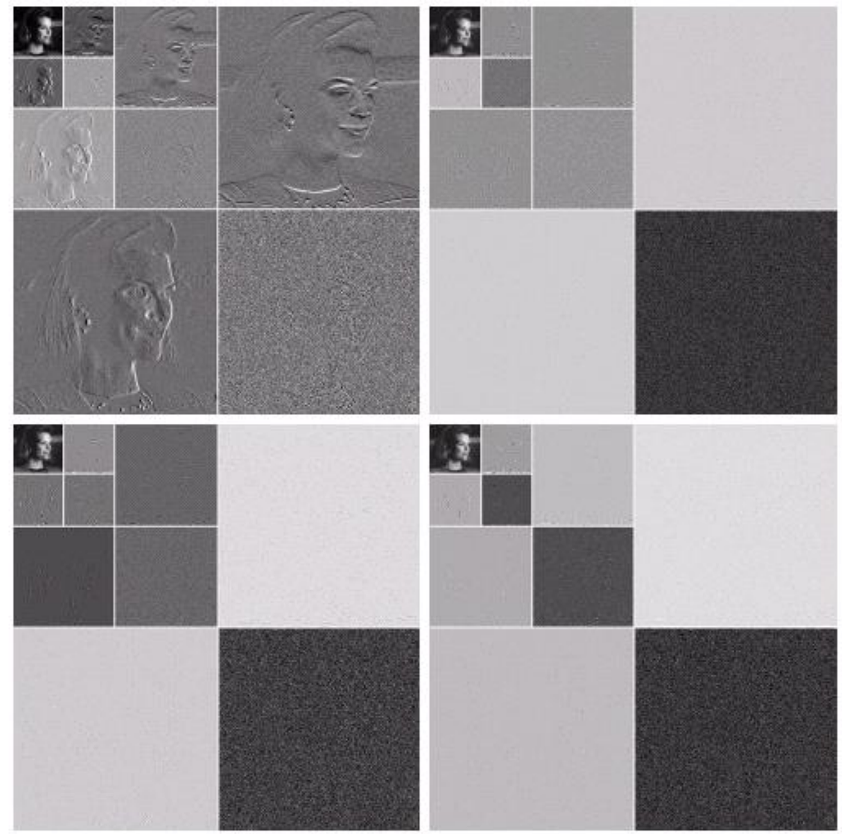
Wavelet Compression





Wavelet Compression

- Large set of filters defined with various features
  - Orthonormal
  - Biorthogonal (orthogonal)
  - Overcomplete (frame)

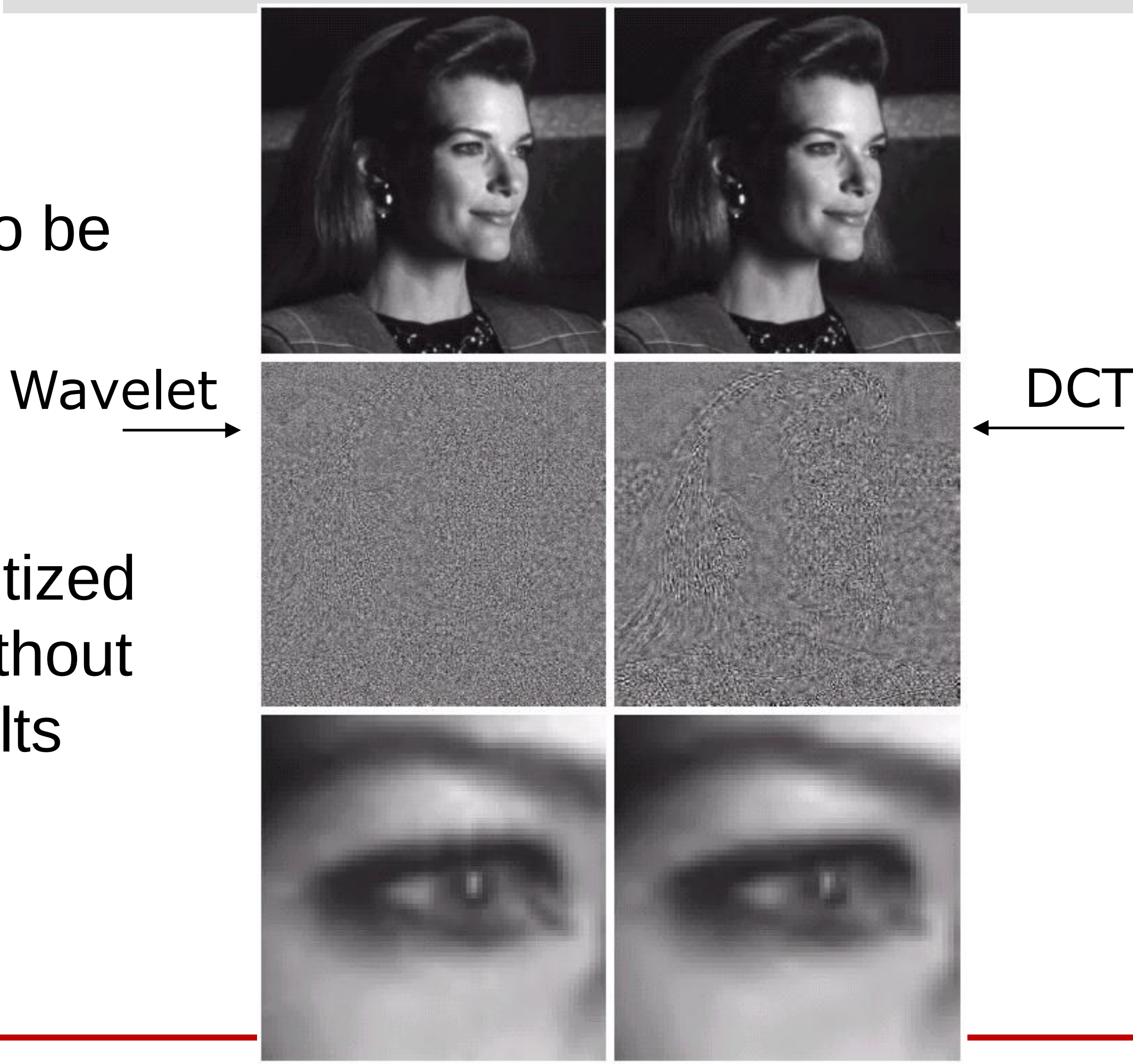


a b
c d

FIGURE 8.42 Wavelet transforms of Fig. 8.23 with respect to (a) Haar wavelets, (b) Daubechies wavelets, (c) symlets, and (d) Cohen-Daubechies-Feauveau biorthogonal wavelets.

Wavelet Compression

- High Frequency coefficients tend to be very small -> 0
- They can be quantized very effectively without distorting the results



Video Sequences

- Video images are three-dimensional generalization of still images (spatial correlation) where the third dimension is time (spatial and temporal correlation)
- Each frame of a video sequence can be compressed by any image compression algorithm
- Motion JPEG (M-JPEG)
  - Images separately JPEG coded
- MPEG (Moving Pictures Expert Group)
  - Temporal correlations used
  - Two basic techniques:
    - Block based motion compensation
    - DCT based compression

Video Sequences

- Compression ratios for various channels

Channel	Bit rate	NTSC TV 168 Mb/s	HDTV 933 Mb/s	Film quality 2300 Mb/s
PC LAN	30 kb/s	5,600:1	31,000:1	76,000:1
Modems	56 kb/s	3,000:1	17,000:1	41,000:1
ISDN	64 - 144 kb/s	1,166:1	6,400:1	16,000:1
T-1, DSL	1.5 Mb/s	112:1	622:1	1,500:1
Ethernet	10 Mb/s	17:1	93:1	230:1
T-3	42 Mb/s	4:1	22:1	54:1
Fiber optic	200 Mb/s	1:1	5:1	11:1

MPEG-family

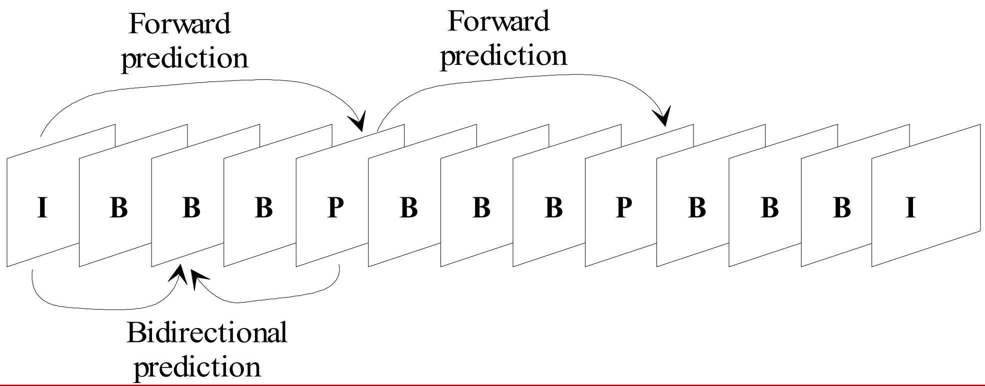
- MPEG-1 (1992): VideoCD
- MPEG-2 (1994): DVD, digital TV, SVCD
  - about 50:1 compression, typically 3-10 Mbps
- MPEG-3: was abandoned
- MPEG-4 (1999+): DivX (starting from Version 5)
  - designed specially for low-bandwidth
- MPEG-7 (>1998):
  - searching and indexing of a/v data, using Description Tools

MPEG-1: Blocks

- The pictures are divided into 16x16 macroblocks, each consisting of four 8x8 elementary blocks
- The choice of the prediction method is chosen for each macroblock separately
- The intra-coded blocks are quantized differently from the predicted blocks:
  - Intra-coded blocks contain information in all frequencies and are quantized differently from the predicted blocks
  - The predicted blocks, contain mostly high frequencies and can be quantized with more coarse quantization tables.

MPEG-1: Inter-block Prediction

- Bidirectional prediction, Forward prediction, Backward prediction; Intra coding
  - I: *Intra pictures* are coded as still images by DCT
  - P: *Predicted pictures* are coded with reference to a past picture. The difference between the prediction and the original picture is then compressed by DCT
  - B: *Bidirectional pictures*, the prediction can be made both to a past and a future frame. Bidirectional pictures are never used as reference

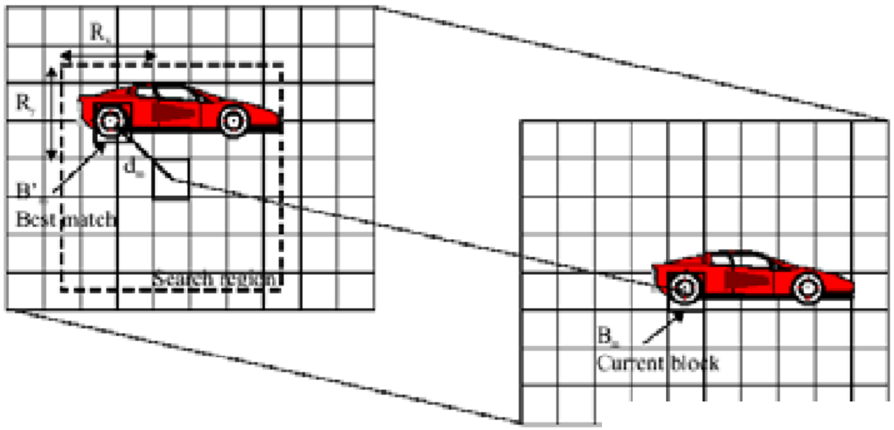


MPEG-1: Motion

- The prediction block in the reference frame is not necessarily in the same coordinates than the block in the current frame
- Because of motion in the image sequence, the most suitable predictor for the current block may exist anywhere in the reference frame
- The *motion estimation* specifies where the best prediction (best match) is found
- *Motion compensation* consists of calculating the difference between the reference and the current block

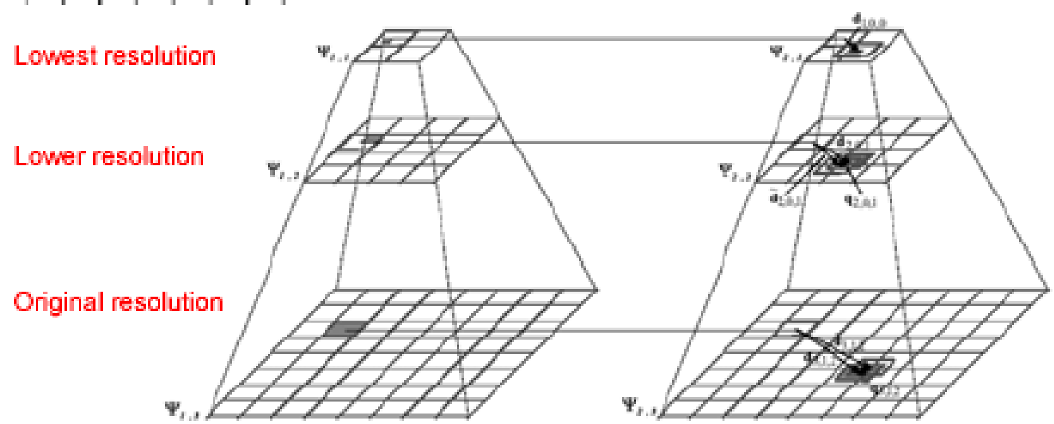
MPEG-1: Motion

- Exhaustive search block matching.



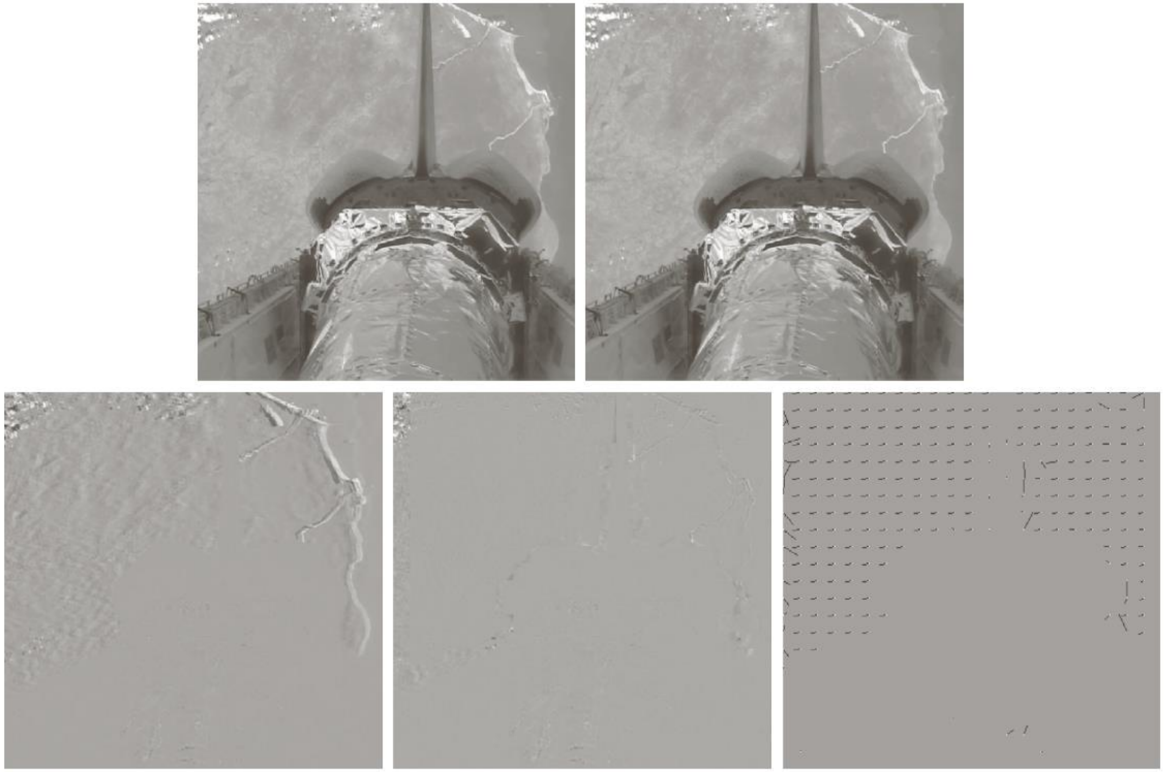
Slow !

- Hierarchical block matching.



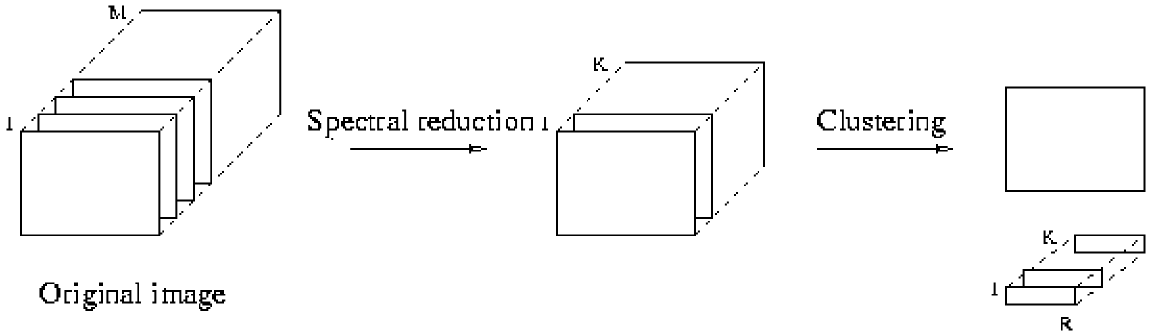
MPEG-1: Motion

- Motion compensation; two frames (13 frames apart), difference, motion compensation applied, motion vectors



Spectral Images

- Approach 1



- Approach 2
  - Clustering replaced with transform coding, quantization, and arithmetic coding.
- Results in lossy compression.

Summary

- Compression approaches contain lots of details in
  - Modelling
  - Transforms
  - Algorithms, Implementations
- Not any one solution good for various types of data
  - Images: Binary, color, spectral, video
  - Text, other data
  - Application?
- Still, how to handle the exponential increase in the amount of data?