Image compression

Václav Hlaváč

Czech Technical University in Prague
Czech Institute of Informatics, Robotics and Cybernetics
160 00 Prague 6, Jugoslávských partyzánů 1580/3, Czech Republic
http://people.ciirc.cvut.cz/hlavac, vaclav.hlavac@cvut.cz
also Center for Machine Perception, http://cmp.felk.cvut.cz
Courtesy: Tomáš Svoboda, Jan Kybic

Outline of the talk:

- Redundance, irrelevance; 1D and 2D.
- Image compression procedure.
- Entropy and compression.
- Optimal coding.
- Compression of segmented images.
- Lossy compression in image domain.
- Compression in transform domain. E.g., JPEG, Wavelets compression.

Image compression, introduction

- The aim: is reducing the amount of data needed to represent the image. The used amount is measured, e.g. in bits.
- Usage: for data transmission or storage.
- Why does 2D compression differs from a 1D one?

- A digitized image is treated as a 2D structure (a matrix) of random samples.
- The compression goal from the procedural point of view: The aim is to tranform the digital image (the matrix of intensities or 3 matrices with color components for color image) into another representation, in which the data are less dependent statistically (roughly: less correlated).

What makes image compression possible?

- Images are not random as noise usually is.
- Images are redundant and predictable.
- Intensities are distributed non-uniformly.
- Color channels are statistically dependent.
- Pixel values are spatially correlated.
- Human vision system does not perceive all details.

Reading resources

- Anil Jain: “Fundamentals of Digital Image Processing”, 1989.
- M. Šonka, V. Hlaváč, R. Boyle R.: “Image Processing, Analysis, and Machine Vision”, 2015, 4th edition.
- T. Svoboda, J. Kybic, V. Hlaváč: “Image Processing, Analysis, and Machine Vision, A MATLAB Companion”, 2007. `http://visionbook.felk.cvut.cz`

Downsampling, motivating image compression

- Reduce the size (spatial resolution) of the image.
- Lossy, simple, often appropriate (limited monitor resolution, web).
- High-quality interpolation (B-splines) helps.

Downsampling, example (1)

Original size, 3456×5184, $859\,\mathrm{kB}$ (stored as JPEG with quality 75).

Downsampling, example (2)

Downsampled $2\times$, 1728×2592, 237 kB.

Downsampling, example (3)

Downsampled $4\times$, 864×1296, 75 kB.

Downsampling, example (4)

Downsampled $8\times$, 432×648, 27 kB.

Downsampling, example (5)

Downsampled $16\times$, 216×324, 10 kB.

Downsampling, example (6)

Downsampled $16\times$, 216×324, 10 kB, bicubic interpolation.

Downsampling, example (7)

Downsampled $32\times$, 108×162, 4.2 kB.

Downsampling, example (8)

Downsampled $32\times$, 108×162, 4.2 kB, bicubic interpolation.

Redundance, irrelevance

- Redundance in coding
  - The basic principle: a less frequent data item (symbol) is coded by a shorter code word than a more frequent one.
  - Optimal coding: Huffman coding and arithmetic coding.
- Redundance among pixels, it is modeled and only residuum to the model is coded because it exhibits a smaller variance. Different models, e.g.:
  - Linear integral transforms, e.g., Fourier, cosine, wavelets.
  - Predictive compression, e.g. a linear combination of a few preceding values.
  - Data-saving image generating models, e.g. fractals.
- Irrelevance from the human perception point of view
  - E.g. some intensity levels, color or frequencies (typically high frequencies) are not represented.

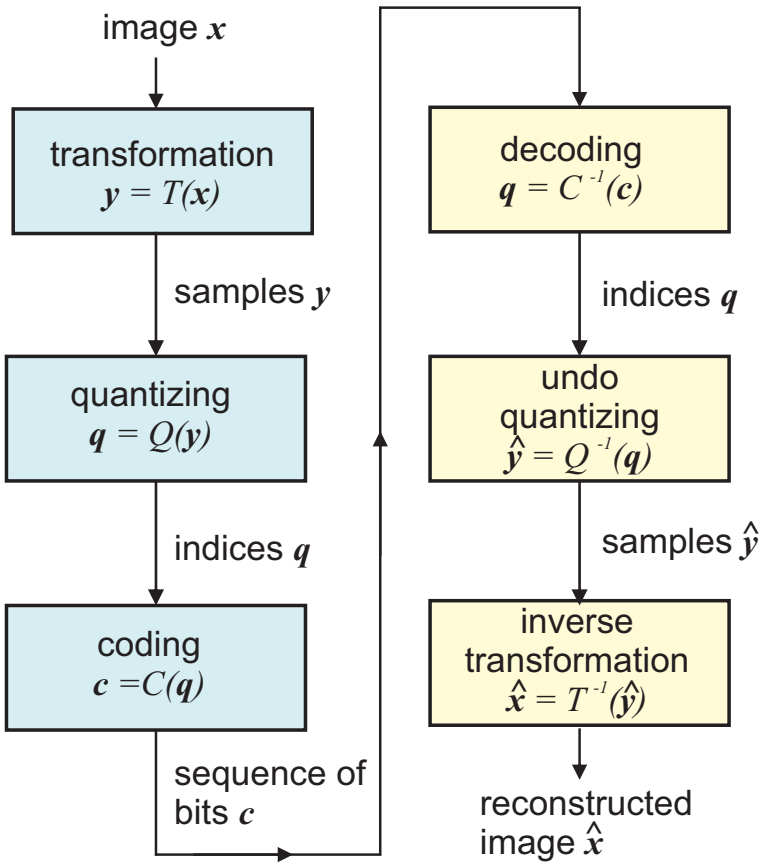
Taxomonmy of image compression methods

1. Based on data interpretation $\Rightarrow$ image segmentation is needed.
   - Methods are dependent on data semantics.
   - Higher compression ratios are achieved.
   - Decompression does not reconstruct the input image fully.
2. Without data interpretation $\Rightarrow$ redundant and irrelevant knowledge is removed.
   - Compression can be used for any image, regardless of its semantics.
   - Statistical redundancy and (possibly) irrelevance for human viewing is explored.

Two classes of methods without interpretation – lossy and lossless ones

1. Lossless methods
   - Only the statistical redundancy is removed/supressed.
   - A full reconstruction of the original signal/image is possible.
2. Lossy methods
   - Irrelevant information is removed.
   - Such information is removed which is unimportant in a given context (e.g. high frequencies, details in intensity, which is unobservable by a human eye).
   - Only partial reconstruction of the original signal/image is possible.

Image compression and its backward reconstruction



- Transformation T reduces redundance and is often invertible.

  E.g., cosine transformation, run length encoding (RLE).

- Quantizing Q removes irrelevance and is not invertible.

  E.g., neglecting cosine transformation coefficient matching to high frequencies.

- Coding C and decoding C^{-1} are invertible and lossless.

Information theory and redundance

- Entropy in physics (thermodynamics) estimates the energy of the system available to perform work. The work can be estimated from the "order in a system". The entropy is a measure of disorderliness in a system. There is a relation to the second thermodynamic theorem.
- The concept was introduced by a German physicist Rudolf Clausius in 1850 (1822-1888, one of thermodynamics founders).

Entropy in information theory, Claude Shannon, 1948

$$H_e = -\sum_i p_i \log_2 \, p_i \quad \text{[bits]} \, ,$$

where p_i is the probability of i-th symbol occurrence in the message.

Entropy, two examples

Let only two symbols a, b occur in the message.

Example 1 $p(\mathsf{a}) = p(\mathsf{b}) = \frac{1}{2}$

$$H = -\left(\frac{1}{2}\log_2\frac{1}{2} + \frac{1}{2}\log_2\frac{1}{2}\right) = \left(\frac{1}{2}\cdot 1 + \frac{1}{2}\cdot 1\right) = 1$$

Example 2 $p(\mathsf{a}) = 0,99;\ p(\mathsf{b}) = 0,01$

$$\begin{aligned} H &= -\left(0,99\log_2 0,99 + 0,01\log_2 0,01\right) \\ &= -\left(0.99\cdot(-0,0145) + 0,01\cdot(-6,6439)\right) \\ &= 0,0144 + 0,0664 = 0,0808 \end{aligned}$$

Entropy and a grayscale image

Let the image has G gray levels, $k = 0 \ldots G-1$ with the probability $P(k)$.

$$\text{Entropy} \quad H_e = -\sum_k P(k) \log_2 P(k) \quad \text{[bits]} \,,$$

Let b be the minimal number of bits, which can represent the used number of quatization levels.

$$\text{Information redundance} \qquad r = b - H_e \,.$$

Estimate of the entropy from the image histogram

Image has a histogram $h(k)$, $0 \leq k \leq 2^b - 1$. M, N are image sizes.

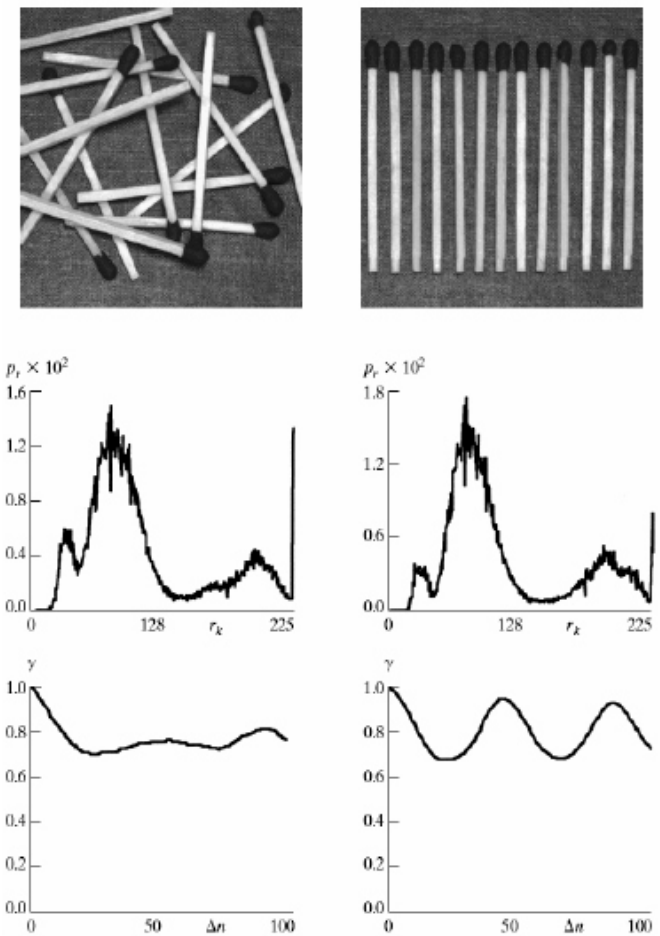
The estimate of the probability $\hat{P} = \dfrac{h(k)}{M\ N}$.

The estimate of the entropy $\hat{H}_e = -\sum_{k}^{2^b-1} \hat{P}(k) \log_2 \hat{P}(k)$ [bits]

Note:

The entropy estimate is overoptimistic. The entropy is lower in reality because there are statistical interdependencies among pixels (redundance).

Illustration of the inter-pixel redundance



Histogram

Normalized autocorrelation

Courtesy: R.C. Gonzalez, R.E. Woods: Digital Image Processing, 2nd Edition, 2002, p. 415

Three definition of the compression ratio

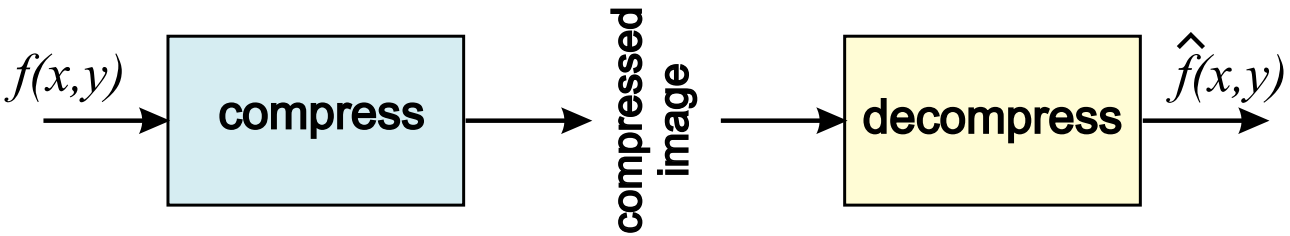
The definition is based on

1. redundance (measured by entropy) $K = \frac{b}{\hat{H}_e}$
2. memory saving
$$\kappa = \frac{n_1}{n_2} = \frac{\text{length of the message before compression}}{\text{length of the message after compression}}$$
3. relative memory saving $R = 1 - \frac{1}{\kappa}$

Example 1: $n_1 = n_2 \Rightarrow \kappa = 1,\ R = 0.$

Example 2: $n_1 : n_2 = 10 : 1 \Rightarrow \kappa = 10,\ R = 0,9 = 90\%.$

Assessing image compression fidelity



- The reconstructed image (estimate) $\hat{f}(x,y) = f(x,y) + e(x,y)$, where $f(x,y)$ is the original image and $e(x,y)$ is the error (or residuum) after the compression.
- The question under discussion is: How close is $f(x,y)$ to $\hat{f}(x,y)$?
- Assessment criteria for reconstruction fidelity:
  - Subjective: based on the human observer, used in television; a cheap and practical one = a difference image.
  - Objective: calculated mathematically. The aim is to substitute subjective methods.

Measurement of the compression loss

$u_1, \ldots, u_n$ is the input sequence and
$u_1', \ldots, u_n'$ is the lossy compressed sequence.

- Mean Square Error (MSE)

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^{n} \left(u_i - u_i' \right)^2$$

- Signal to noise ratio (SNR)

$$\text{SNR} = 10 \log_{10} \frac{P^2}{\text{MSE}^2} \, [\text{dB}] \, ,$$

where P is the interval of input sequence values,
$P = \max\{u_1, \ldots, u_n\}$ - $\min\{u_1, \ldots, u_n\}$.

Measurement of the compression loss (2)

- Peak-signal to noise ratio (PSNR)

$$\mathrm{PSNR} = 10 \log_{10} \frac{M^2}{\mathrm{MSE}^2},$$

where M is the maximal interval of input sequence values, e.g. 256 for 8 bit range and 65356 for 16 bit range.

SNR and PSNR are used mainly in applications. The expression for MSE serves as an auxiliary value for the SNR and PSNR definitions.

Huffman coding, from the year 1952

- Input: a message, i.e. symbols with the probability of their occurrence.
- Output: the optimally coded message.
- Prefix code, i.e. no code word can be a prefix of any other code word. It allows decoding without knowing the length of individual words corresponding to coded symbols.
- Procedure: the binary (Huffman) tree is created in a bottom-up manner based on the probability of symbols occurrence. This tree serves for message encoding.
- An integer number of bits per coded symbol.
- Let b be the average number of bits per symbol. Let L be the average length of a coded word.

$$H(b) \leq L \leq H(b) + 1$$

Huffman coding, example

Input: messages are composed of 8 symbols with the known probability of occurrence.

s_0	s_1	s_2	s_3	s_4	s_5	s_6	s_7
0.12	0.26	0.3	0.15	0.1	0.03	0.02	0.02

$s_2 = 2$ 0.3

$s_1 = 1$ 0.26

$s_3 = 3$ 0.15

$s_0 = 0$ 0.12

$s_4 = 4$ 0.1

$s_5 = 5$ 0.03

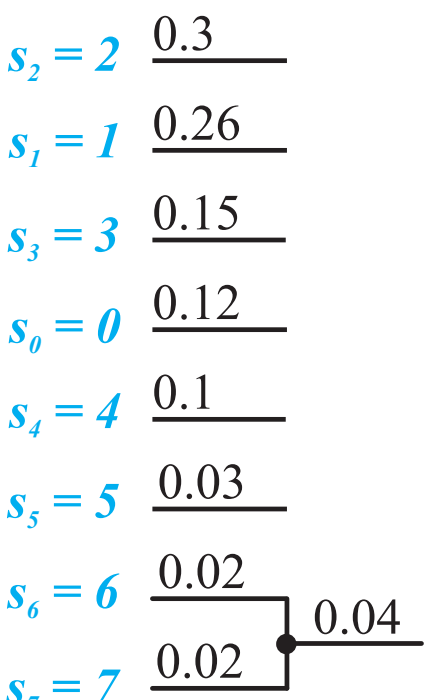
$s_6 = 6$ 0.02

$s_7 = 7$ 0.02

Huffman coding, example

Input: messages are composed of 8 symbols with the known probability of occurrence.

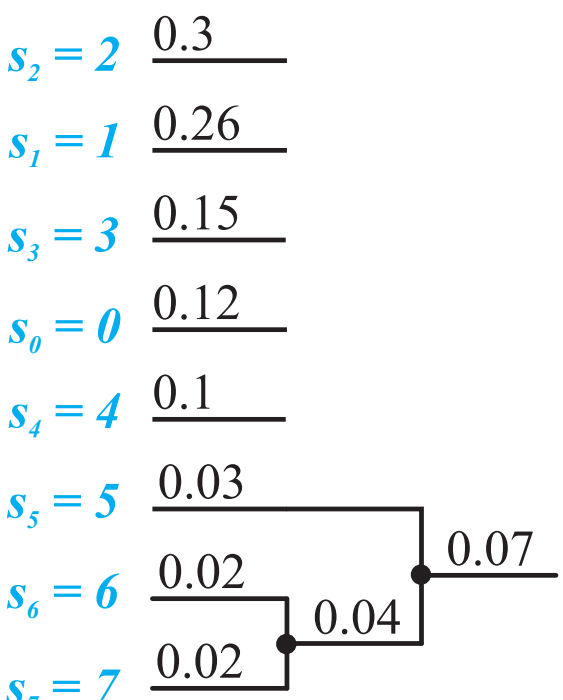
s_0	s_1	s_2	s_3	s_4	s_5	s_6	s_7
0.12	0.26	0.3	0.15	0.1	0.03	0.02	0.02



Huffman coding, example

Input: messages are composed of 8 symbols with the known probability of occurrence.

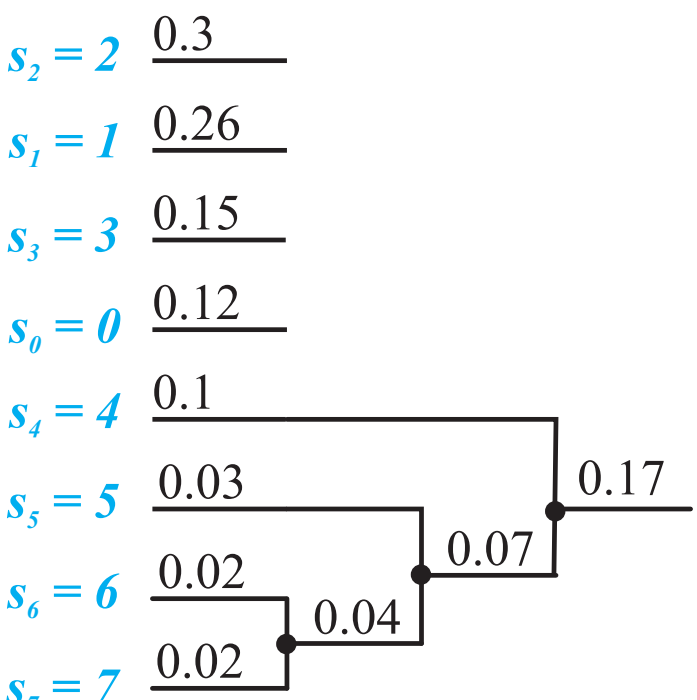
s_0	s_1	s_2	s_3	s_4	s_5	s_6	s_7
0.12	0.26	0.3	0.15	0.1	0.03	0.02	0.02



Huffman coding, example

Input: messages are composed of 8 symbols with the known probability of occurrence.

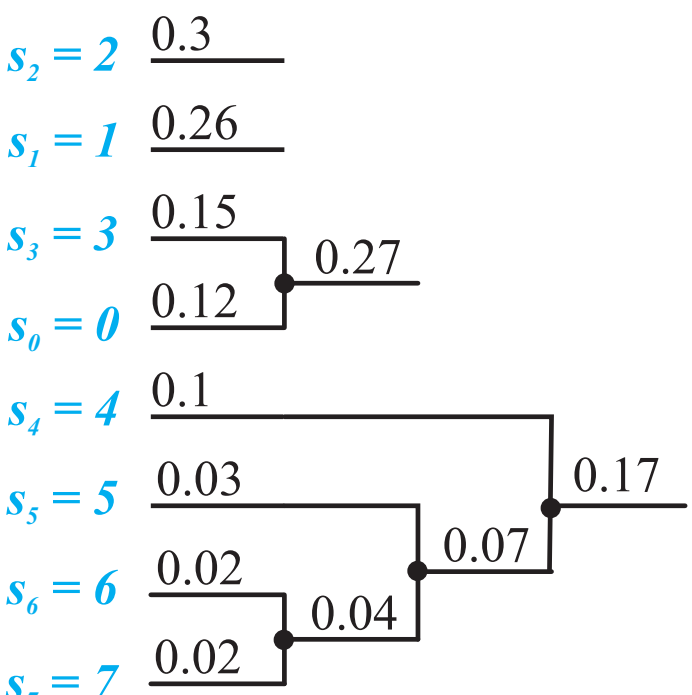
s_0	s_1	s_2	s_3	s_4	s_5	s_6	s_7
0.12	0.26	0.3	0.15	0.1	0.03	0.02	0.02



Huffman coding, example

Input: messages are composed of 8 symbols with the known probability of occurrence.

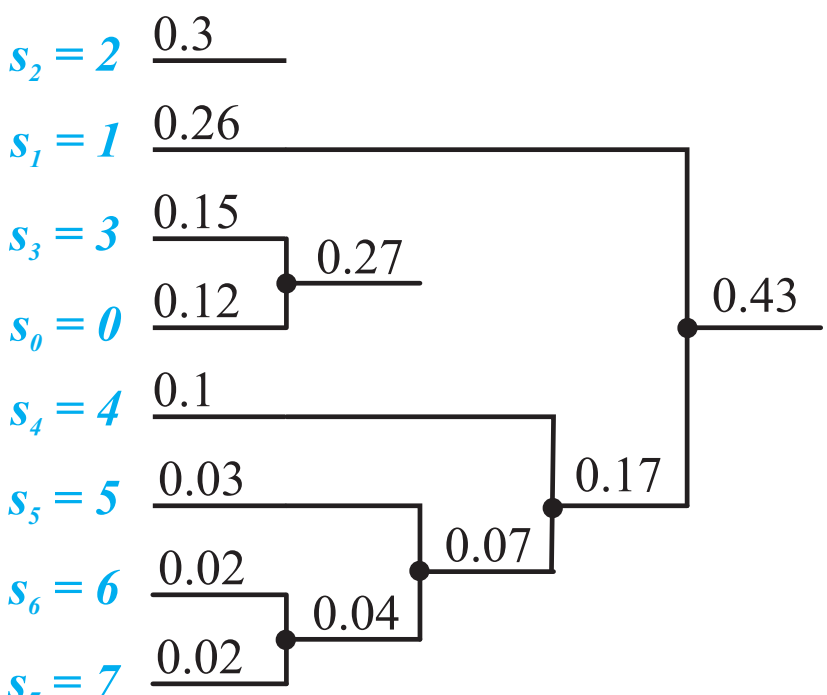
s_0	s_1	s_2	s_3	s_4	s_5	s_6	s_7
0.12	0.26	0.3	0.15	0.1	0.03	0.02	0.02



Huffman coding, example

Input: messages are composed of 8 symbols with the known probability of occurrence.

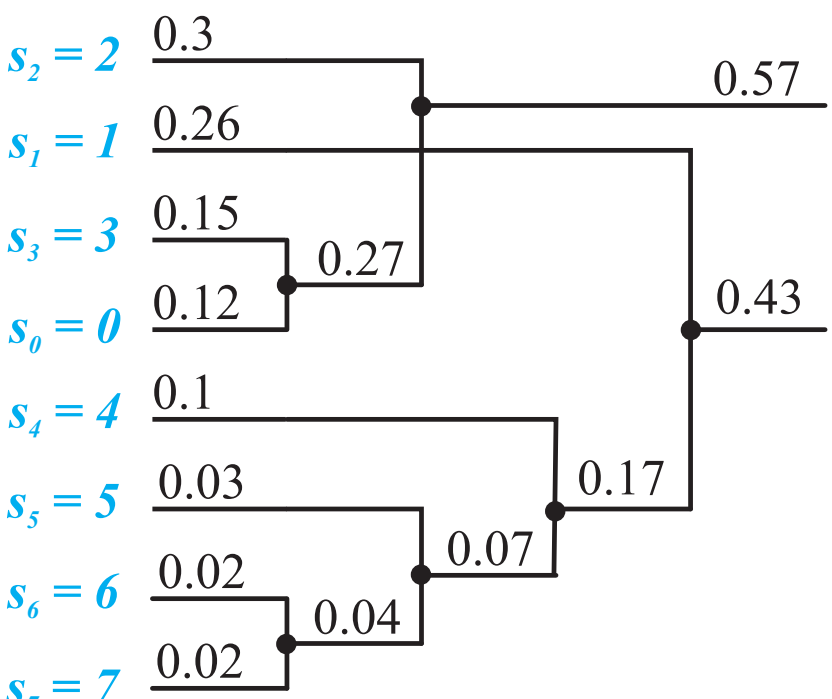
s_0	s_1	s_2	s_3	s_4	s_5	s_6	s_7
0.12	0.26	0.3	0.15	0.1	0.03	0.02	0.02



Huffman coding, example

Input: messages are composed of 8 symbols with the known probability of occurrence.

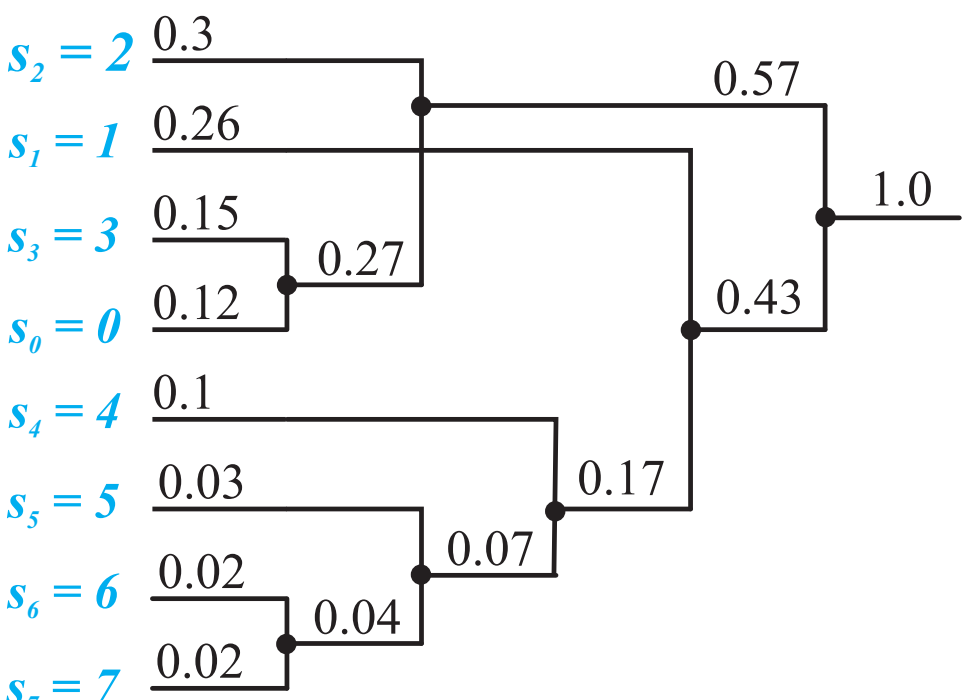
s_0	s_1	s_2	s_3	s_4	s_5	s_6	s_7
0.12	0.26	0.3	0.15	0.1	0.03	0.02	0.02



Huffman coding, example

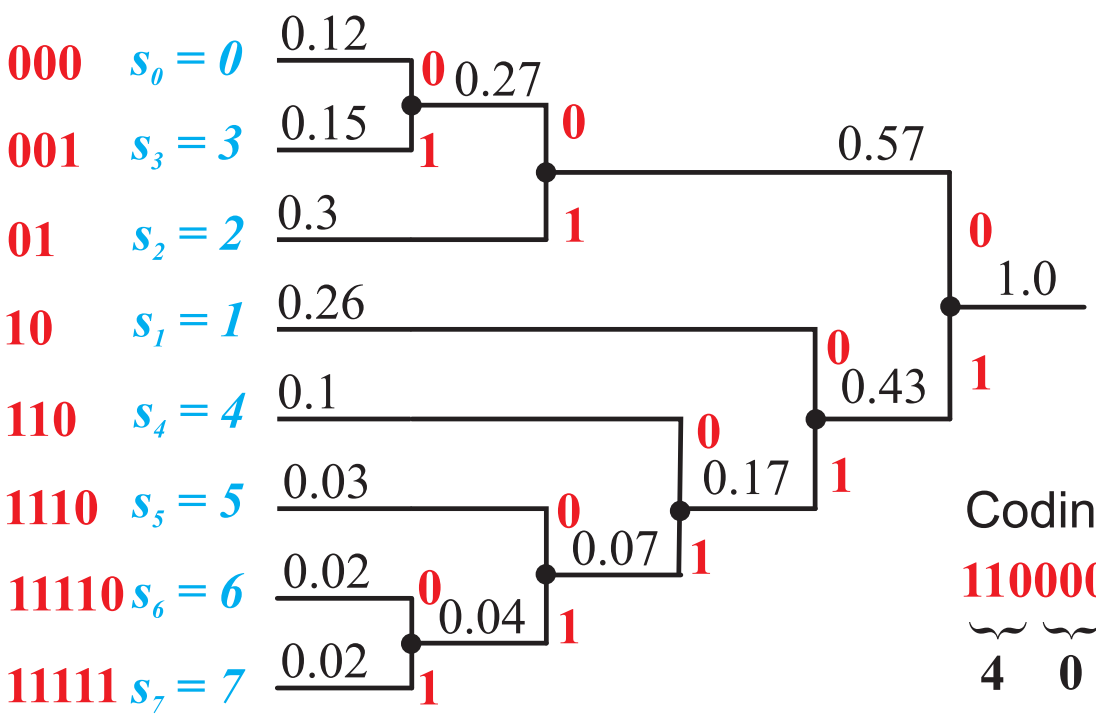
Input: messages are composed of 8 symbols with the known probability of occurrence.

s_0	s_1	s_2	s_3	s_4	s_5	s_6	s_7
0.12	0.26	0.3	0.15	0.1	0.03	0.02	0.02



Huffman coding, example
re-ordering of the tree

- Reordering is needed to have the tree without crossing branches.
- Coding: either 0 or 1 at branching points.



Coding/decoding

110000011010

4 0 2 1 1

Compression of segmented (interpreted) image data

Same methods can be used for binary images because they can be treated as a result of segmentation. Ones in the image correspond to objects and zeroes to background (or vice-versa).

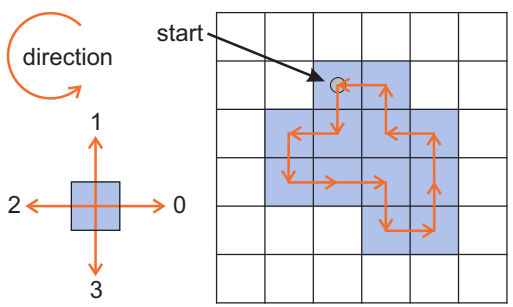
Taxonomy of methods:

- Chain code representation of a region boundary, a special case of the polygonal representation (lossless compression).
- Region boundary approximation by a polygonal curve, called also curve vectorization (lossy compression).
- Run length encoding of regions (lossless compression).
- Corner compression (lossless), allows set and a few other operations with compressed images, by M.I. Schlesinger 1986, there is a V. Hlaváč's separate talk on it.

Chain code of a region boundary

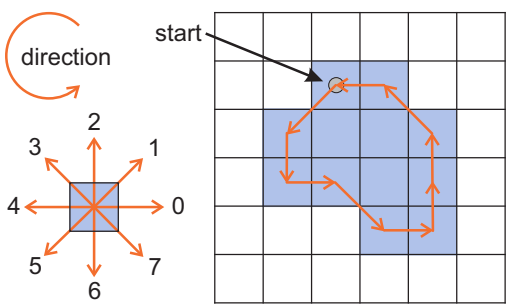
- The chain code (H. Freeman 1961) is a special case of the region boundary polygonal representation. The individual polygonal segments are of length 1 of the used neighborhood relation (4, 8, 6 -neighbors).
- A starting point is given, e.g. the most top-left pixel.
- The anti-clockwise direction is assumed while traversing the region boundary.
- Fast implementation: a 3×3 neighborhood and look into 256-lookup table.
- Disadvantage: the chain code depends on a starting point.

4-neighborhood



Chain code: 3 2 3 0 0 3 0 1 1 2 1 2

8-neighbourhood

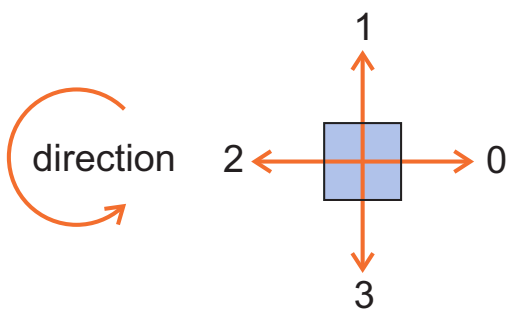


Chain code: 5 6 0 7 0 2 2 3 4

Derivative dd of the chain code

- Derivative dd (also the first difference) of the chain code yields the rotation invariance up to 90^o for 4-neighborhood or up to 45^o for 8-neighborhood.
- Derivative dd = the number of direction changes in counterclockwise direction needed to rotate from the old direction d_{old} to the new direction d_{new}.

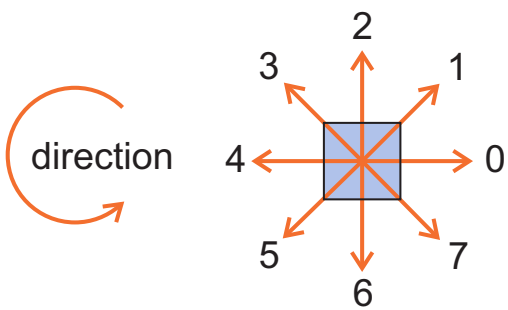
4-neighborhood



if $d_{\text{new}} \geq d_{\text{old}}$ then $dd = d_{\text{new}} - d_{\text{old}}$

if $d_{\text{new}} < d_{\text{old}}$ then $dd = 4 + d_{\text{new}} - d_{\text{old}}$

8-neighborhood

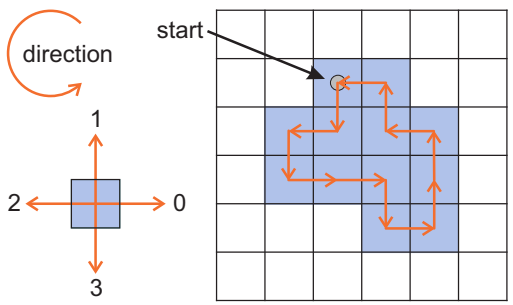


if $d_{\text{new}} \geq d_{\text{old}}$ then $dd = d_{\text{new}} - d_{\text{old}}$

if $d_{\text{new}} < d_{\text{old}}$ then $dd = 8 + d_{\text{new}} - d_{\text{old}}$

Chain code derivative dd, example

4-neighborhood



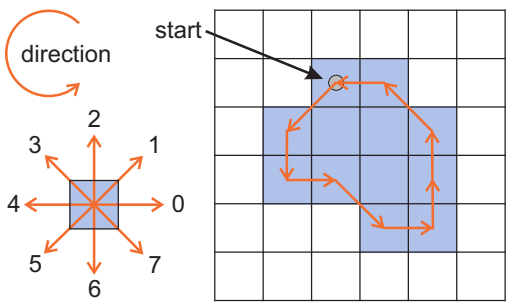
if $d_{\text{new}} \geq d_{\text{old}}$ then $dd = d_{\text{new}} - d_{\text{old}}$

if $d_{\text{new}} < d_{\text{old}}$ then $dd = 4 + d_{\text{new}} - d_{\text{old}}$

Chain code: 3 2 3 0 0 3 0 1 1 2 1 2

Derivative dd: 3 1 1 0 3 1 1 0 1 3 1 1

8-neighborhood



if $d_{\text{new}} \geq d_{\text{old}}$ then $dd = d_{\text{new}} - d_{\text{old}}$

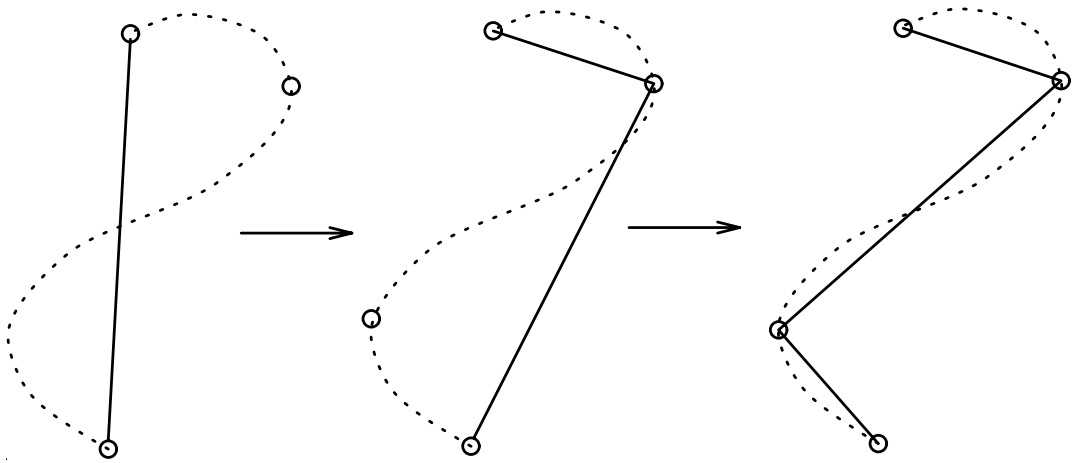
if $d_{\text{new}} < d_{\text{old}}$ then $dd = 8 + d_{\text{new}} - d_{\text{old}}$

Chain code: 5 6 0 7 0 2 2 3 4

Derivative dd: 1 2 7 1 2 0 1 1 1

Region boundary approximation by polygons

Ramer (1972), Douglas-Peucker (1973) recursive algorithm



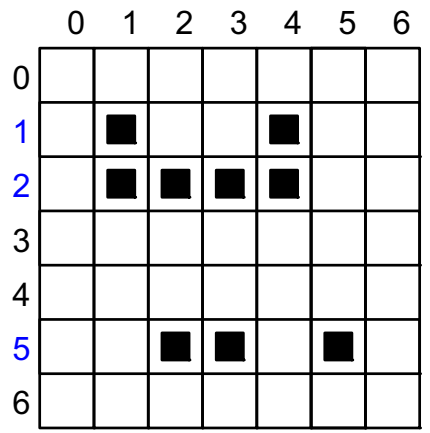
Urs Ramer: An iterative procedure for the polygonal approximation of plane curves. In Computer Graphics and Image Processing. Volume 1, Issue 3, pp. 244-256, 1972.

David Douglas, Thomas Peucker, Algorithms for the reduction of the number of points required to represent a digitized line or its caricature, The Canadian Cartographer 10(2), 112–122 (1973)

Run Length Encoding (RLE) of regions

- The code is composed of a list of lists.
- The outer list contains several inner lists describing only lines containing black pixels.
- Each inner list represents the situation in a single line with black pixel. The first number is the line number (in our example in blue).
- The remaining elements of inner lists are pairs of numbers. The first number in each pair is a column number in which a contiguous sequence of pixels starts. The second number is the column number in which it finishes.
- RLE is used by FAX (CCITT Group 3).

Example:



Run Length encoding:

((11144)(214)(52355))

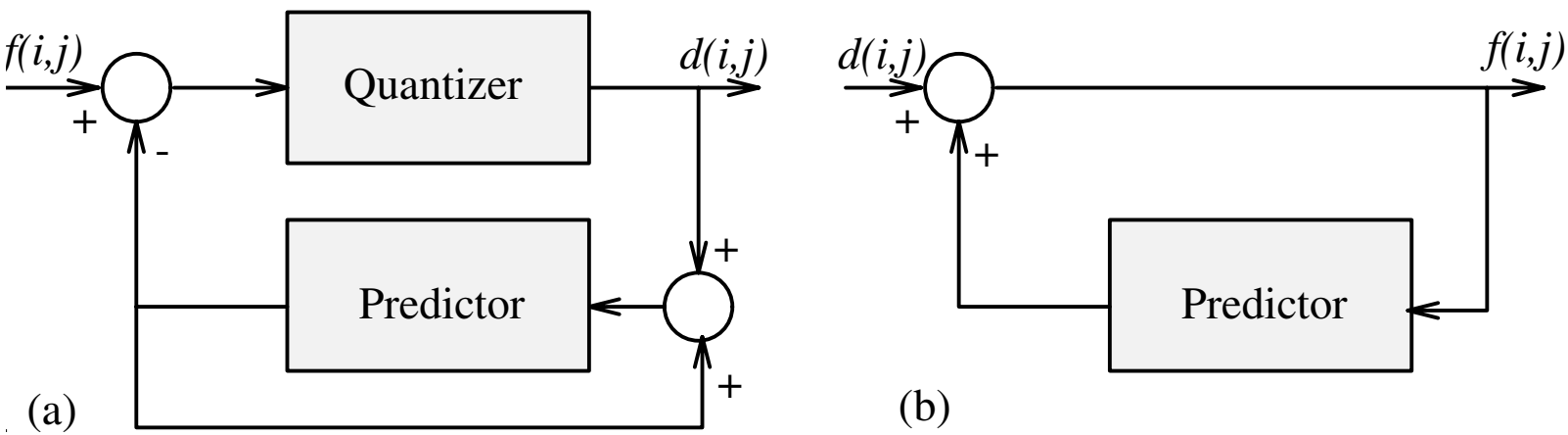
Lossy compression, taxonomy of approaches

There are three main approaches to lossy image compression:

1. Reducing the data in the original image domain, i.e. by removing many image pixels and filling the missing data by inpainting.
2. Using a predictor which approximates a pixel value from a few "past" samples. The method can be lossless if the whole residuum between the predictor and real pixel is stored, otherwise it can be lossy. Example: Digital Pulse Coding Modulation.
3. Reducing the data in a transform domain (e.g. discrete cosine transform or wavelet transform). The remaining compressed data are used to reconstruct the original image.

Predictive compression – the idea

- The idea is to find a mathematical model, which allows predicting the pixel value from several values in a local neighborhood.
- The difference (prediction error) between the correct and the predicted value for each pixel and a few model parameters for the entire image are stored/transmitted.
- A compression occurs because the prediction error exhibits lower statistical variance than the original data.

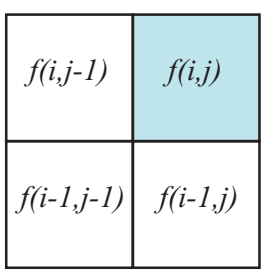


Digital pulse coding modulation (1)

- Let $f(i,j)$ be the image. Statistical dependencies in the image are estimated using the autocorrelation function $R(i,j,k,l) = \mathcal{E}(f(i,j)\, f(k,l)) = f\, f^{\top}$.
- The mathematical model of a predictor $\hat{f}(i,j)$ is sought.
- The prediction error is $d(i,j) = \hat{f}(i,j) - f(i,j)$.
- Let assume, e.g., a simple linear predictor of the third order

$$\hat{f}(i,j) = a_1\, f(i,j-1) + a_2\, f(i-1,j-1) + a_3\, f(i-1,j)\,,$$

where a_1, a_2, a_3 are its parameters.



Digital pulse coding modulation (2)

- How are the parameters a_1, a_2, a_3 of the predictive model going to be estimated?
- By solving a statistical optimization task. The stationary random process f with zero mean value is assumed,

$$e = \mathcal{E}\left([\tilde{f}(i,j) - f(i,j)]^2 \right) ,$$

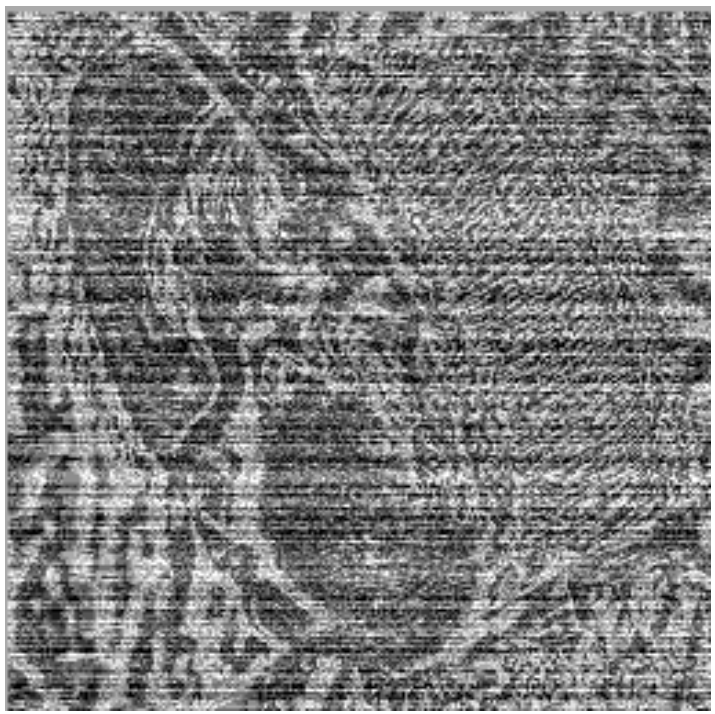
as well as the predictor of the third order,

$$\begin{aligned} a_1\, R(0,0) + a_2\, R(0,1) + a_3\, R(1,1) &= R(1,0) \\ a_1\, R(0,1) + a_2\, R(0,0) + a_3\, R(1,0) &= R(1,1) \\ a_1\, R(1,1) + a_2\, R(1,0) + a_3\, R(0,0) &= R(0,1) \end{aligned}$$

where $R(m,n)$ is the autocorrelation function of a special form $R(\alpha,\beta) = R(0,0)\exp(-c_1\alpha - c_2\beta)$.

DPCM – example, $K = 3.8$

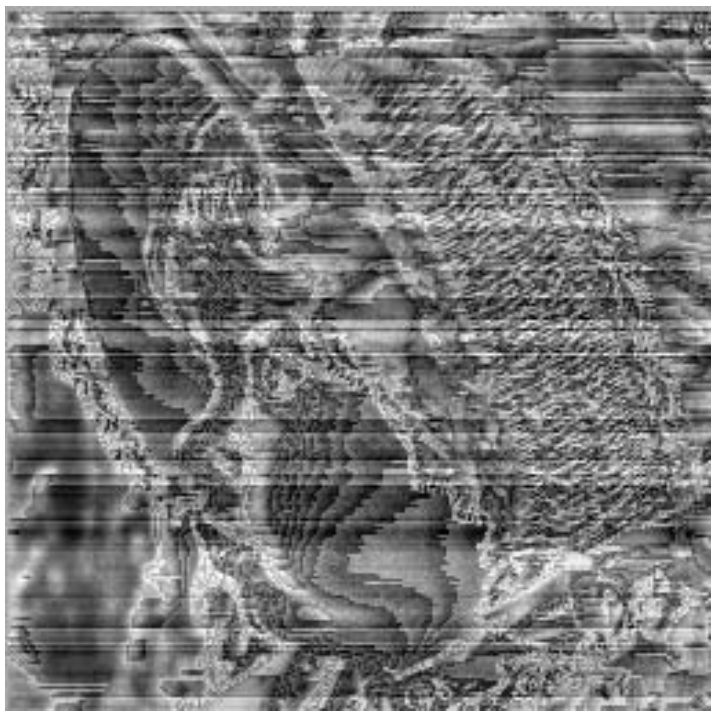
After reconstruction $K = 3.8$.



Predictor error.

DPCM – example, $K = 6.2$

After reconstruction $K = 6.2$.



Predictor error.

JPEG, introduction

- JPEG (Joint Photographic Expert Group) was standardized in the year 1992.
- JPEG is used both for gray level and color images. Color images are first converted from the RGB color space to YUV color space, in which the U, V matrices are stored with the half resolution of the matrix Y ($\approx$ image intensity).
- There is both lossless and lossy compression in JPEG working on different principles.
- The first generation of JPEG (.jpg) from 1992 uses discrete cosine transformation (DCT) to remove redundance a irrelevance applied in 8×8 neighborhoods. DCT coefficients are converted to a 1D vector, are Run Length Encoded (RLE) and coded optimally by Huffman coding.
- The second generation JPEG2000 (.jp2) from the year 2000 removed redundance and irrelevance using the wavelet transform. Coding is performed in individual bit planes separately using the arithmetic coding.

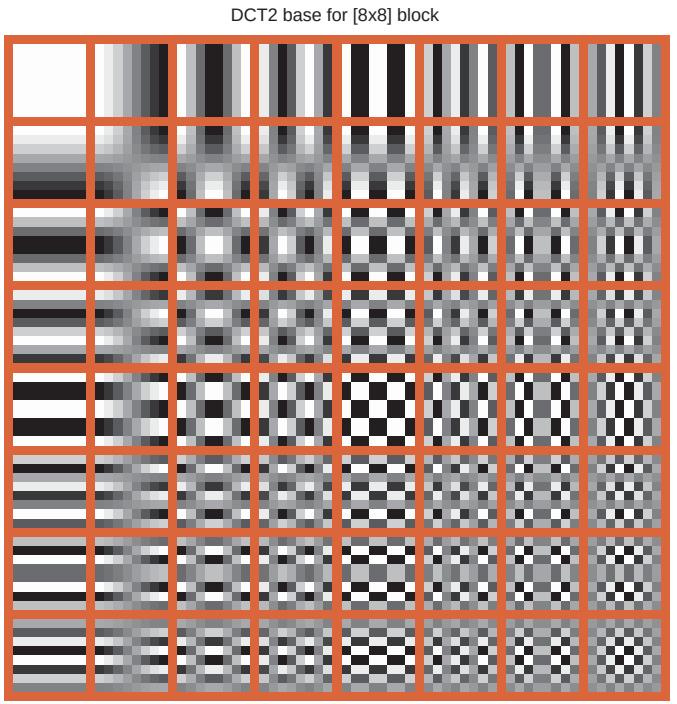
Why DCT was used in JPEG?

- DCT is periodic implicitly. No troubles with discontinuities occur.
- DCT apporximates PCA (Principal Component Analysis, Karhunen-Loeve expansion), which is optimal from the mean square error (MSE, energy) point of view.
- DCT has fixed basis functions. In the PCA case, the basis functions need to be calculated for each image again and again.
- The image is divided into non-overlapping blocks of the size 8×8. The data in each block are compressed independently each of the other.

DCT, basis functions

64 fixed basis functions are used.

- Each block of the image of the size 8×8 is expressed as a linear combination of basis functions.
- While compressing, 64 weights of linear combinations are calculated.
- Weights are thresholded. The threshold value provides the degree of compression, i.e. selects desired irrelevance.



Example, cameraman

image block

image intensities

	1	2	3	4	5	6	7	8
1	185	187	184	183	189	186	185	186
2	185	184	186	190	187	186	189	191
3	186	187	187	188	190	185	189	191
4	186	189	189	189	193	193	193	195
5	185	190	188	193	199	198	189	184
6	191	187	162	156	116	30	15	14
7	168	102	49	22	15	11	10	10
8	25	19	19	26	17	11	10	10

Example, cameraman, DCT

image intensities

	1	2	3	4	5	6	7	8
1	185	187	184	183	189	186	185	186
2	185	184	186	190	187	186	189	191
3	186	187	187	188	190	185	189	191
4	186	189	189	189	193	193	193	195
5	185	190	188	193	199	198	189	184
6	191	187	162	156	116	30	15	14
7	168	102	49	22	15	11	10	10
8	25	19	19	26	17	11	10	10

coefficients of the DCT2

	1	2	3	4	5	6	7	8
1	1117	114	10	7	19	−2	−7	2
2	459	−119	−20	−11	−16	−4	3	0
3	−267	−3	24	8	1	6	4	−1
4	50	107	−9	−1	11	−6	−7	3
5	52	−111	−22	−2	−16	−2	5	−3
6	−38	39	46	19	2	0	4	3
7	−17	39	−46	−26	8	−5	−10	2
8	30	−46	28	22	−9	2	7	−1

Example, cameraman, 100 % a 50 %

100 % of most significant DCT2 coeffs

50 % of most significant DCT2 coeffs

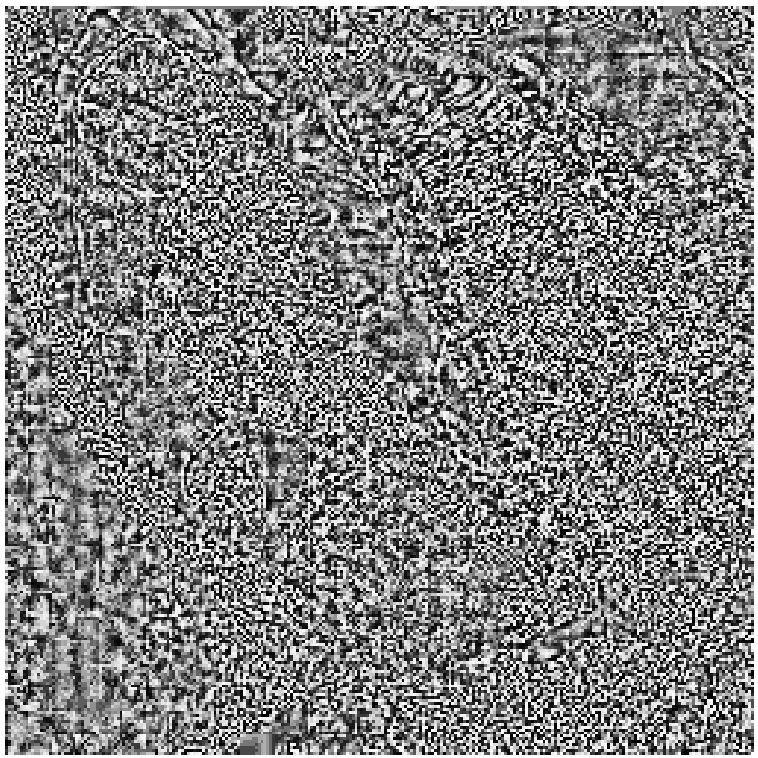
Example, cameraman, 20 % a 5 %

20 % of most significant DCT2 coeffs

5 % of most significant DCT2 coeffs

JPEG – example, $K = 3.8$

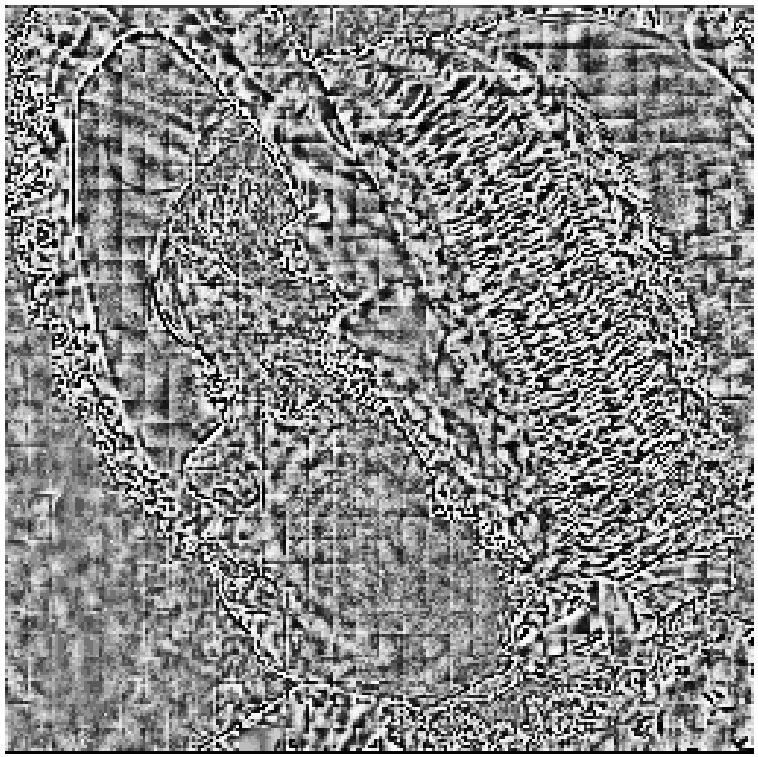
After reconstruction $K = 3.8$.



Predictor error.

JPEG – example, $K = 4.2$

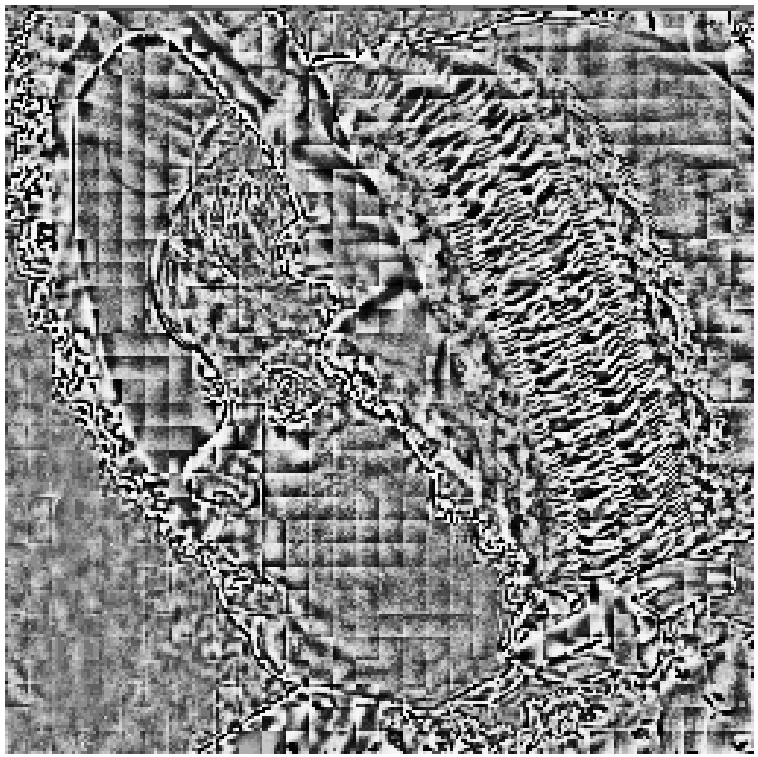
After reconstruction $K = 4.2$.



Predictor error.

JPEG – example, $K = 5.6$

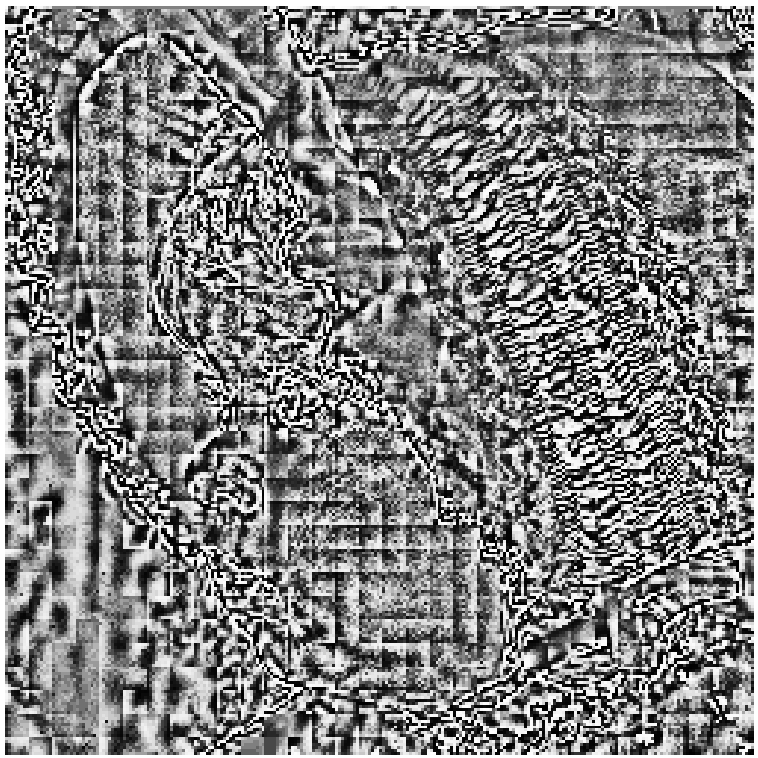
After reconstruction $K = 5.6$.



Predictor error.

JPEG – example, $K = 10.2$

After reconstruction $K = 10.2$.



Predictor error.