Information theoretic basics (1)

- information content I_i of symbol x_i with probability $P(x_i)$

$$I_i = ld \frac{1}{P(x_i)} \; [bit]$$

(Claude E. Shannon, 1948)

- information content is reciprocal to the probability of symbols
  - the more surprising a symbol is, the higher its information content
  - highly probable symbols contribute less information
- information content is expressed in binary units

Information theoretic basics (2)

- entropy H of source symbols x_i $(i=1,\ldots,n)$

$$H = \sum_{i=1}^{n} P(x_i)\, I_i \qquad [bit/symbol]$$

- entropy of source H denotes the minimum average number of bits to code a message comprised of symbols x_i

- for statistically independent symbols x_i, it is not possible to provide a lossless message with less bits than given by H, multiplied with the number of transmitted symbols

- it is necessary to model the actual code to match the information content of individual symbols in order to provide an efficient code

Entropy example

Symbols	•	• A ■	• B ■	• C ■	• D ■
Probabilities	• a)	• 1/4	• 1/4	• 1/4	• 1/4
	• b)	• 1/2	• 1/4	• 1/8	• 1/8

Entropy

$$H_a = H_{max} = 4 \cdot \frac{1}{4} \cdot ld\ 4 = 2\ \ [bit/symbol]$$

$$H_b = \frac{1}{2} ld\ 2 + \frac{1}{4} ld\ 4 + 2 \cdot \frac{1}{8} ld\ 8 = 1.75\ \ [bit/symbol]$$

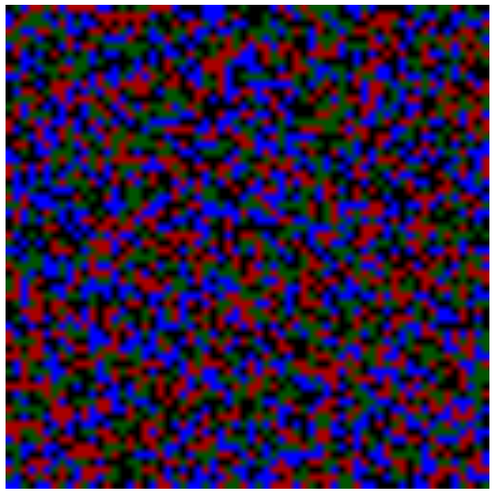
Redundancy of source

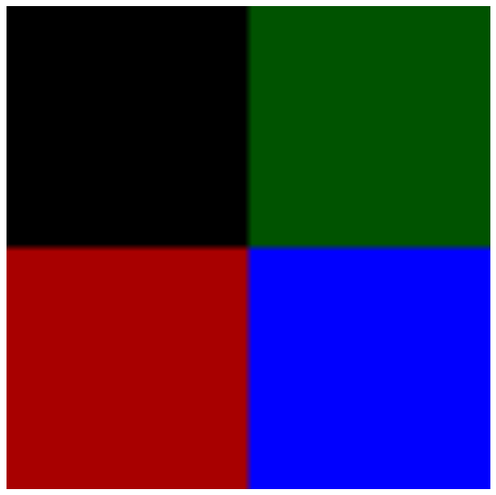
$$R_a = H_{max} - H_a = 0$$ (redundancy free source)

$$R_b = H_{max} - H_b = 0.25\,[bit/symbol]$$

Entropy example (2)

$$H_a = 2\ bit/symbol$$



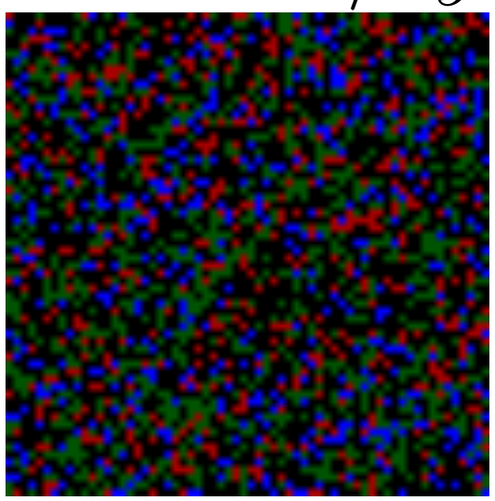


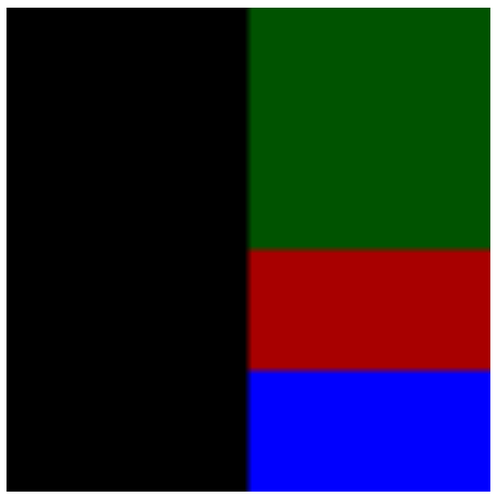
What about these two distributions below ?

Probabilities are equal to examples a and b.

On the first glance, the formally calculated entropy would be equal to the random patterns on the left side.

$$H_b = 1.75\ bit/symbol$$





Symbols in the examples on the right are obviously correlated.

- **Entropy formula for uncorrelated source symbols does not apply in such cases.**
- **Dependencies not always obvious, hence success of coding dependent on modeling.**

Summary entropy

- H - Entropy of a source
- $H = H_{max}$, if symbols are equally distributed ($P_i = 1/n$)
- the more uneven the symbol distribution, the lower the average information content
- $H = 0$, if $P_i = \begin{cases} 1.0 & i = j \\ 0.0 & i \neq j \end{cases}$ one symbol only(!)
- $0 \leq H \leq H_{max}$ for statistically independent symbols
- H is symmetric, order of symbols irrelevant

Redundancy

- Coding Redundancy

$$R_{cod} = L_{aver.} - H \quad [bit/symbol]$$

with $L_{aver.}$ - mean code word length (over whole message)

H - source entropy

- Inter-symbol Redundancy

$$R_{inter} = H - H_{cond} \quad [bit/symbol]$$

with H_{cond} - conditional entropy

Variable Length Codes (VLC)

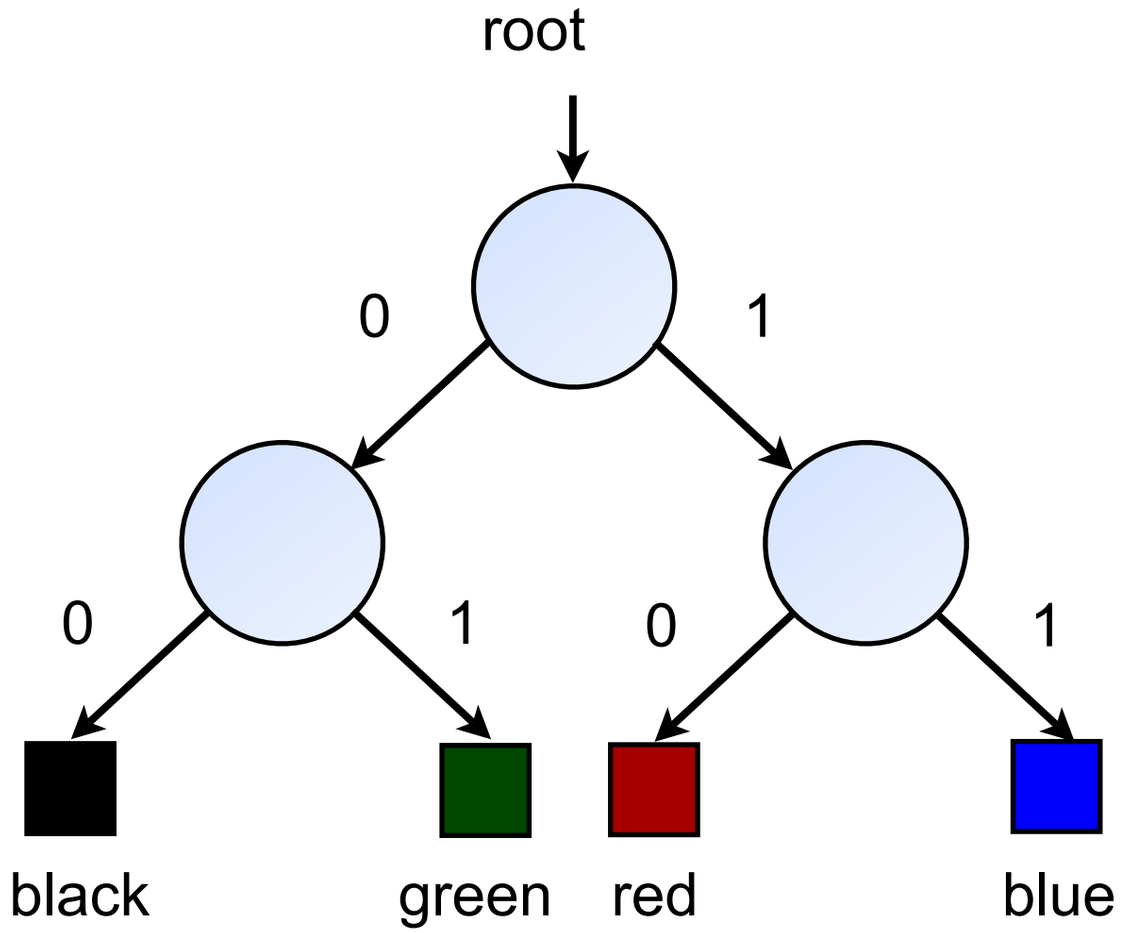
- Example: Morse Code (Samuel F. B. Morse, 1843)
  - one of the first transmission codes for systems with just two states
  - introduced for the telegraph system, where initially only „on/off“ electrical signals could be transmitted
  - combination of short pulses (‚dit‘, ‚◦‘) and 3x longer pulses (‚dah‘, ‚-‘)
  - separation of code letters and code words by pauses (3x, 7x of short pulse length)
  - some letters:

Symbol	A	B	C	D	E	F	G	H	I	...
Code	◦-	-◦◦◦	-◦-◦	-◦◦	◦	◦◦-◦	--◦	◦◦◦◦	◦◦	

  - properties:
    - frequent symbols have a shorter code
    - length of code words is retrieved from the pause
    - no symbols of the same length share the same prefix
    - interpretation by code word length and ‚dit‘, ‚dah‘ - count

Representation of symbols as code tree

- each binary branch is given a code word element, either 0 or 1
- leaves of the tree correspond to the represented symbols
- tree annotation is not unique, inversion of branch bits still leads to a valid code
- shown code is valid, every path from root to leaves is distinct



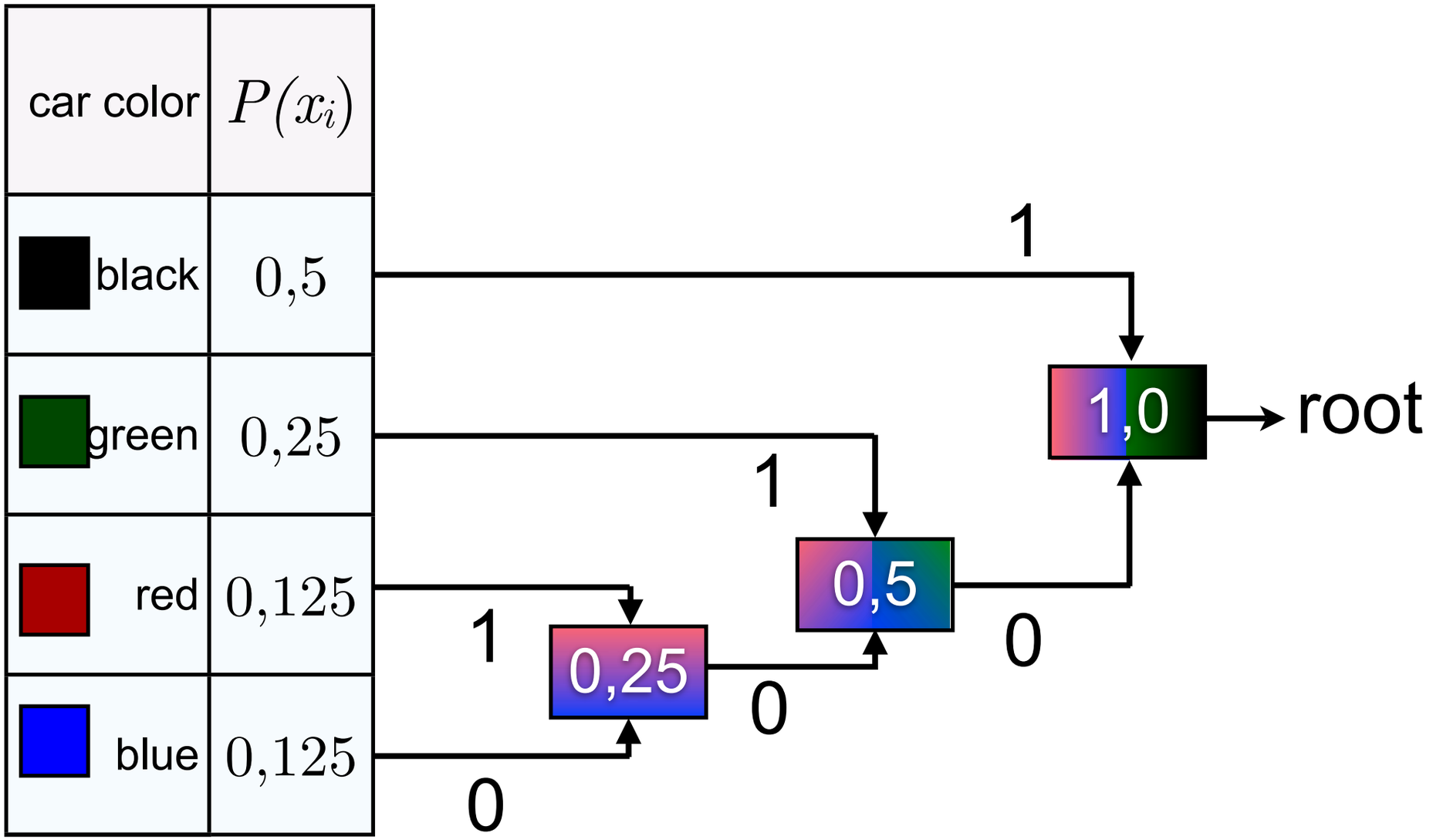
Huffman Code (optimal binary valued code)

- Construction algorithm: (Huffman D. 1952)
  1. find two symbols x_j, x_k with lowest probability within input symbols x_i, where $p_i = P(x_i)$, $p_j \leq p_k \leq p_i$ with $i = (1,2,...,n)$, $i \neq j$, $i \neq k$
  2. assign a new symbol x_{jk} to the compound of x_j, x_k
     - 2.1. remove x_j, x_k from input symbols ($n = n - 2$)
     - 2.2. assign 0 to x_j ,1 to x_k, remember x_j, x_k as leaves of x_{jk}
     - 2.3. add x_{jk} with probability $p_{jk} = p_j + p_k$ to input symbols x_i ($n = n + 1$)
  3. return to 1) until $p_{jk} = 1.0$
  4. collect leaves of the tree, generate code bits and lengths along the path from root to leaves
- Features
  - bottom-up method, starting with the leaves of the tree
  - optimum code with code words of binary length $l_i \in \mathbb{N}$
  - despite optimum code, the coding redundancy R_{cod} is only zero in case $p_i = \frac{1}{2^n}$, $n \in \mathbb{N}$
  - minimum code length l_i is 1 bit, hence even when the entropy H_{src} is less than one bit per symbol, all code words contain at least one bit

Huffman Code - Example

- Construction by using least probabilities, compound symbol represents the former individual symbols or compounds
- code words / lengths can be obtained by following a path from root to the leaves

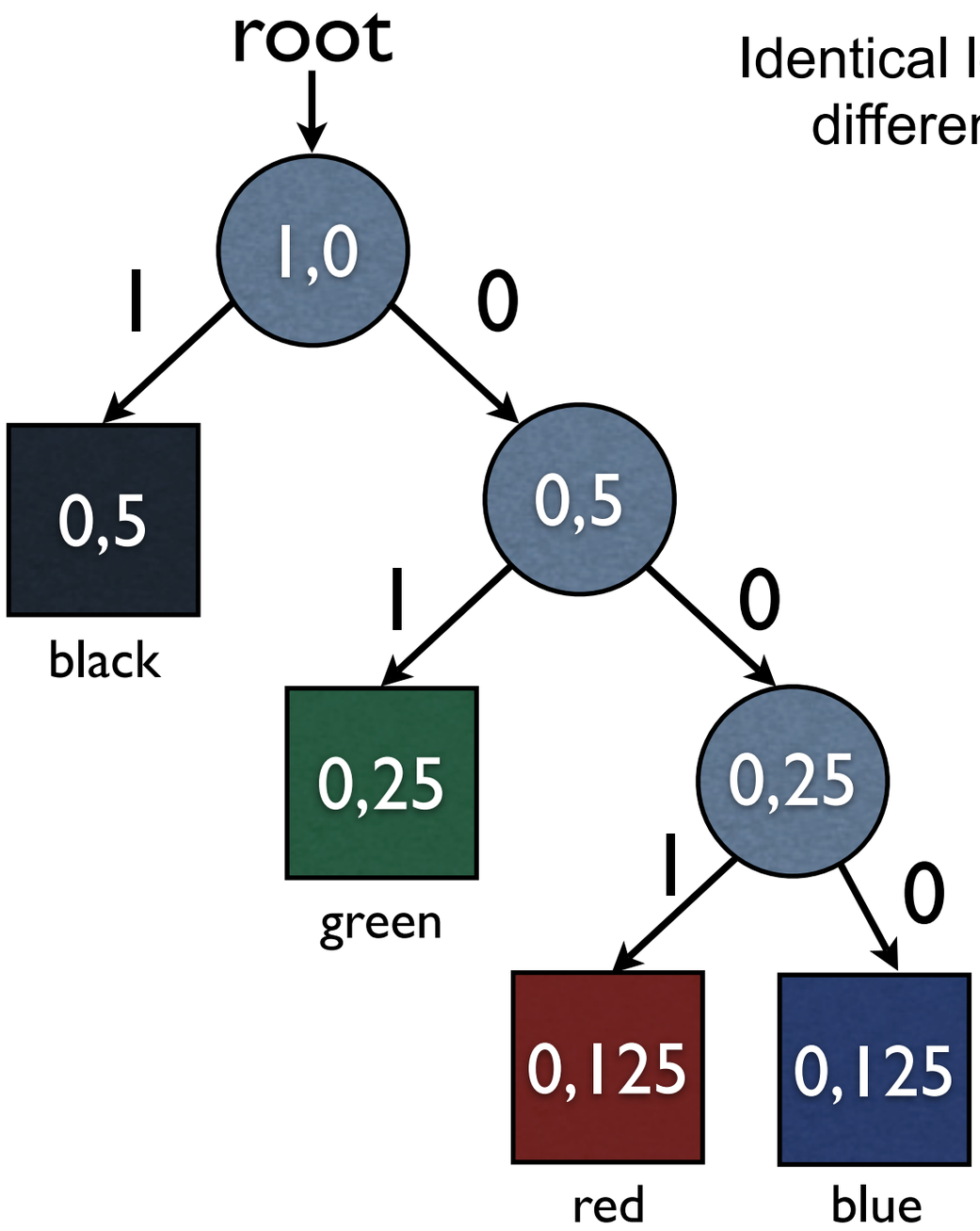
car color	$P(x_i)$
black	$0{,}5$
green	$0{,}25$
red	$0{,}125$
blue	$0{,}125$

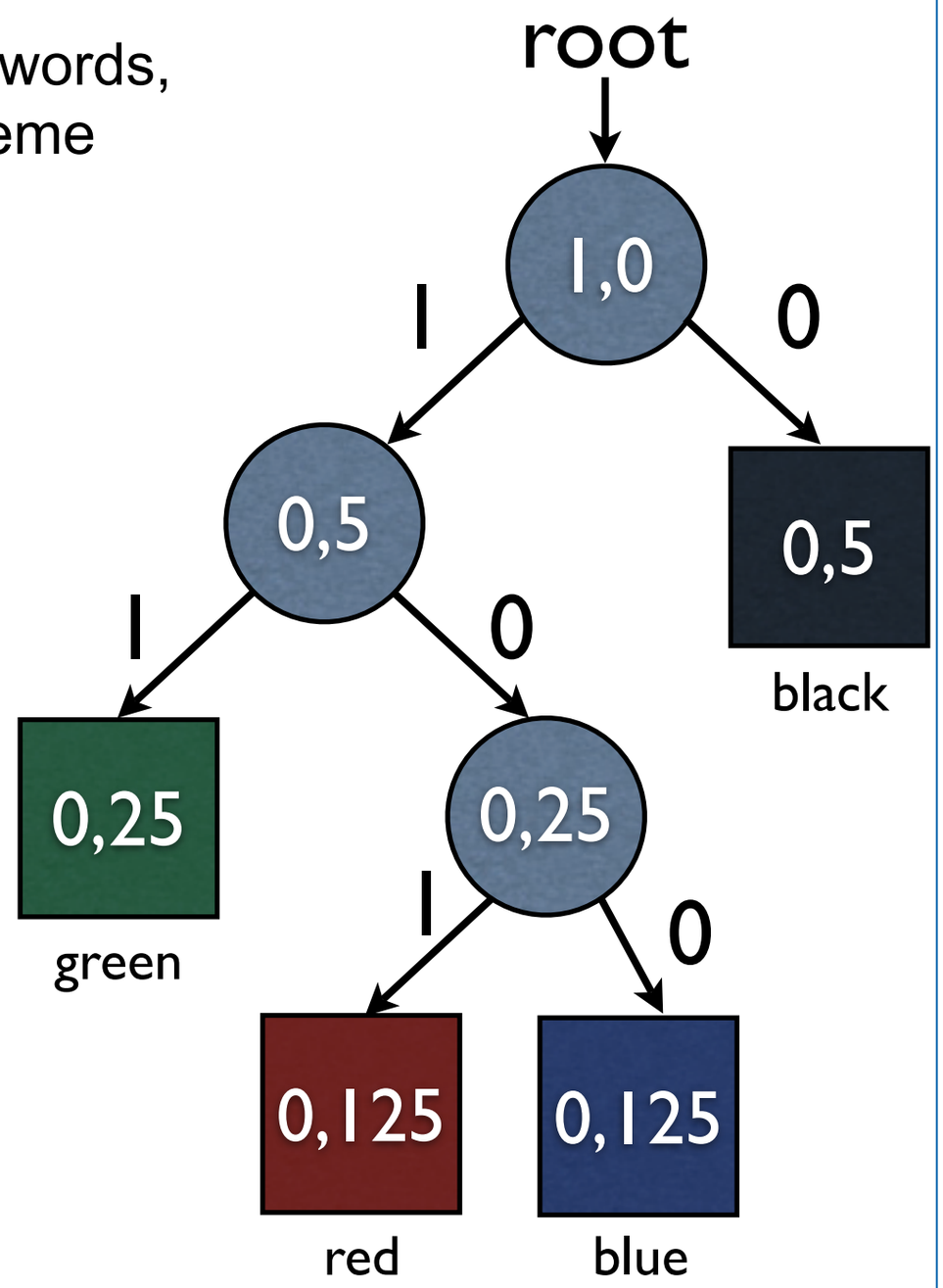


c_i	l_i
1	1
01	2
001	3
000	3

VLC tree options/properties (1)

Identical length of code words,
different naming scheme

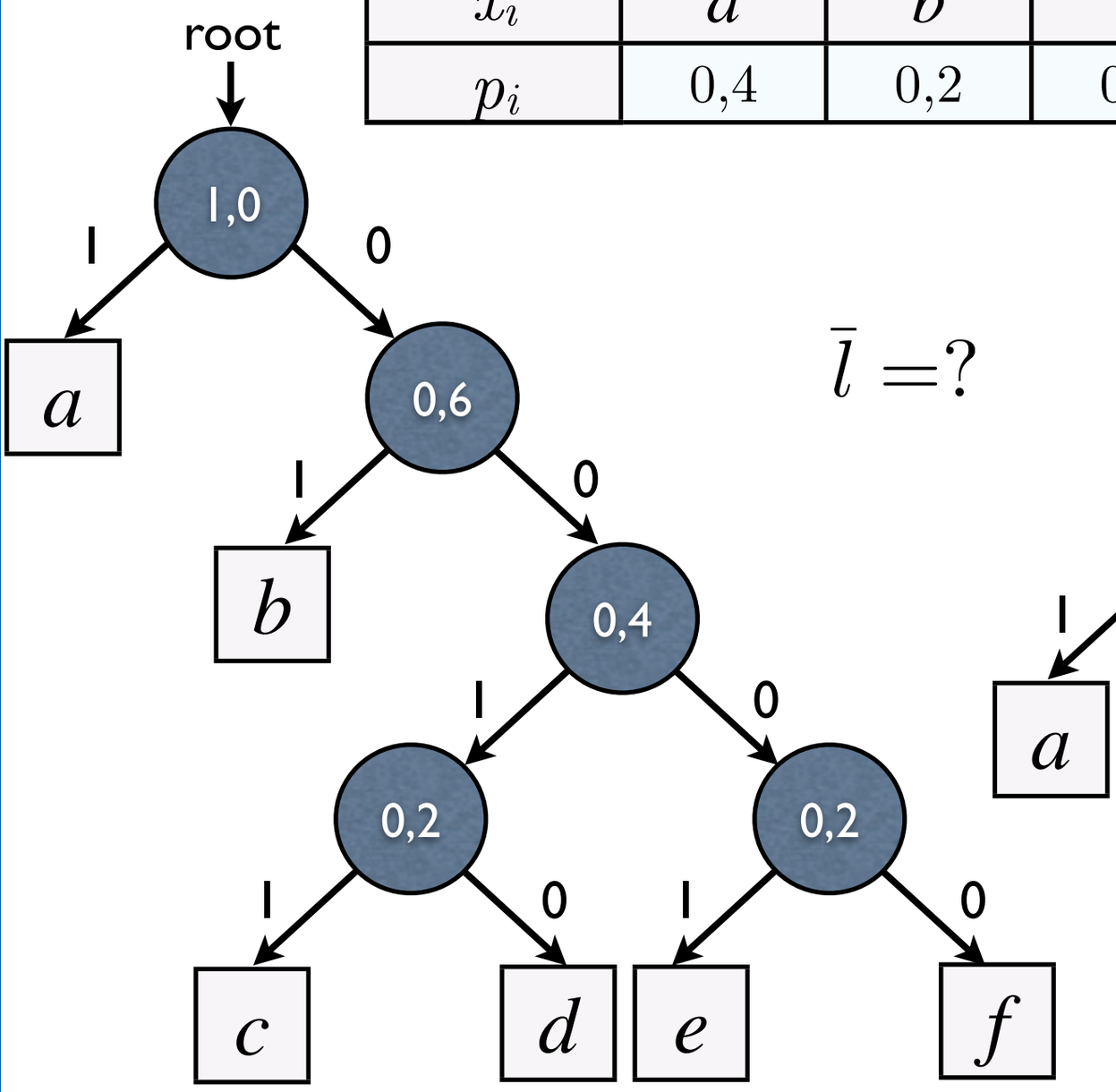




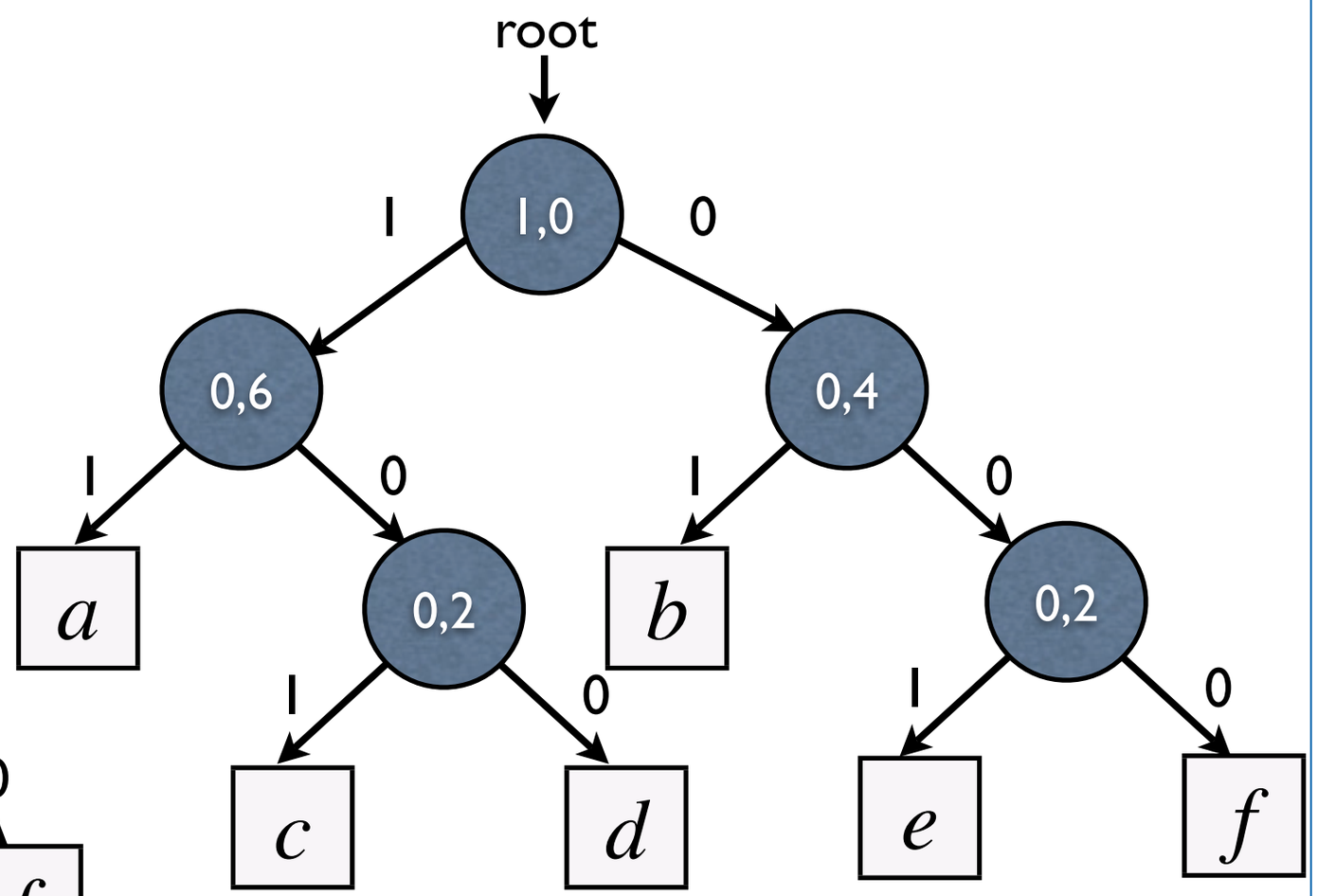
$$\bar{l} = ?$$

VLC tree options/properties (2)

i	1	2	3	4	5	6
x_i	a	b	c	d	e	f
p_i	0,4	0,2	0,1	0,1	0,1	0,1



$\bar{l} = ?$



- different length of individual code words, different naming scheme

Summary: Variable Length Codes

- Huffman Codes are optimal bit based codes for a given probability distribution
- Advantages
  - fast encoding and decoding possible by using look-up tables
  - prefix codes
    - have an implicit length
    - do not need further control information
    - have no restrictions on symbol count and allow random symbol order
  - can be inter-mixed with fixed length codes in a single bit stream
- Drawbacks
  - $\bar{l} \geq 1.0$, even with $H_{src} < 1.0\ Bit/symbol$
  - if signal statistics change (i.e. non stationary input signals)
    - compression gets sub-optimal due to fixed code trees
    - adaptive coding feasible but code tree regeneration (at each symbol) is complex and with varying tree adaptation algorithms not necessarily unique
    - in most cases, either the code tree is transmitted within data stream or known a priori in form of a constant table in both encoder and decoder
  - coding redundancy R_{cod} is only zero in case $p_i = \frac{1}{2^n}, n \in \mathbb{N}$

Precoding method overview

precoding

run length coding

- strings and blocks of identical symbols are substituted by a smaller number of symbols from a new alphabet

dictionary coding

- arbitrary symbol strings are mapped into new symbols of another alphabet
- match arbitrary symbol strings against an encoder maintained dictionary
- encode match parameters instead of explicit string coding

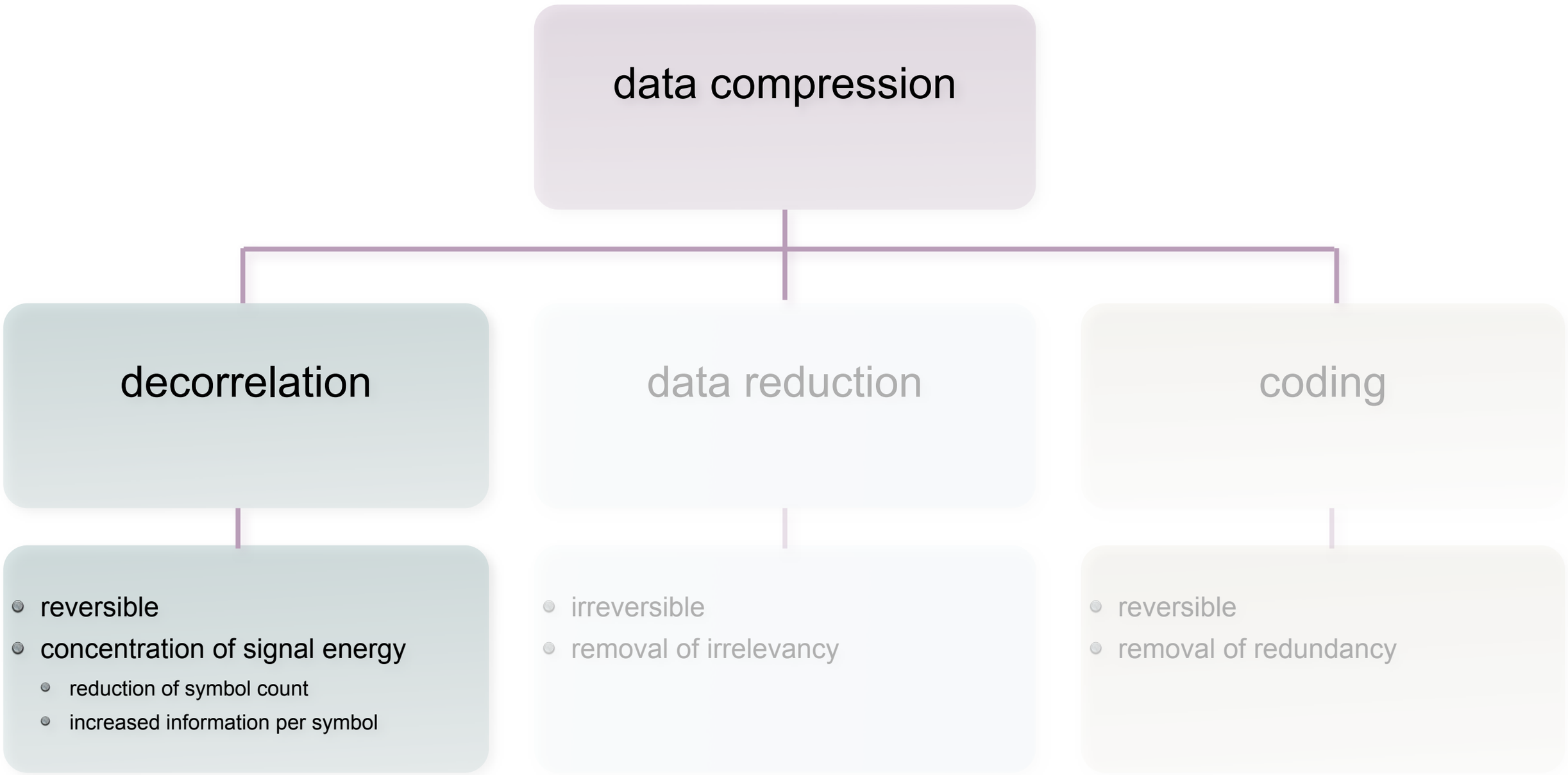
block sorting

- strings of symbols are sorted to match other sets in the current scope more closely in order to increase correlation

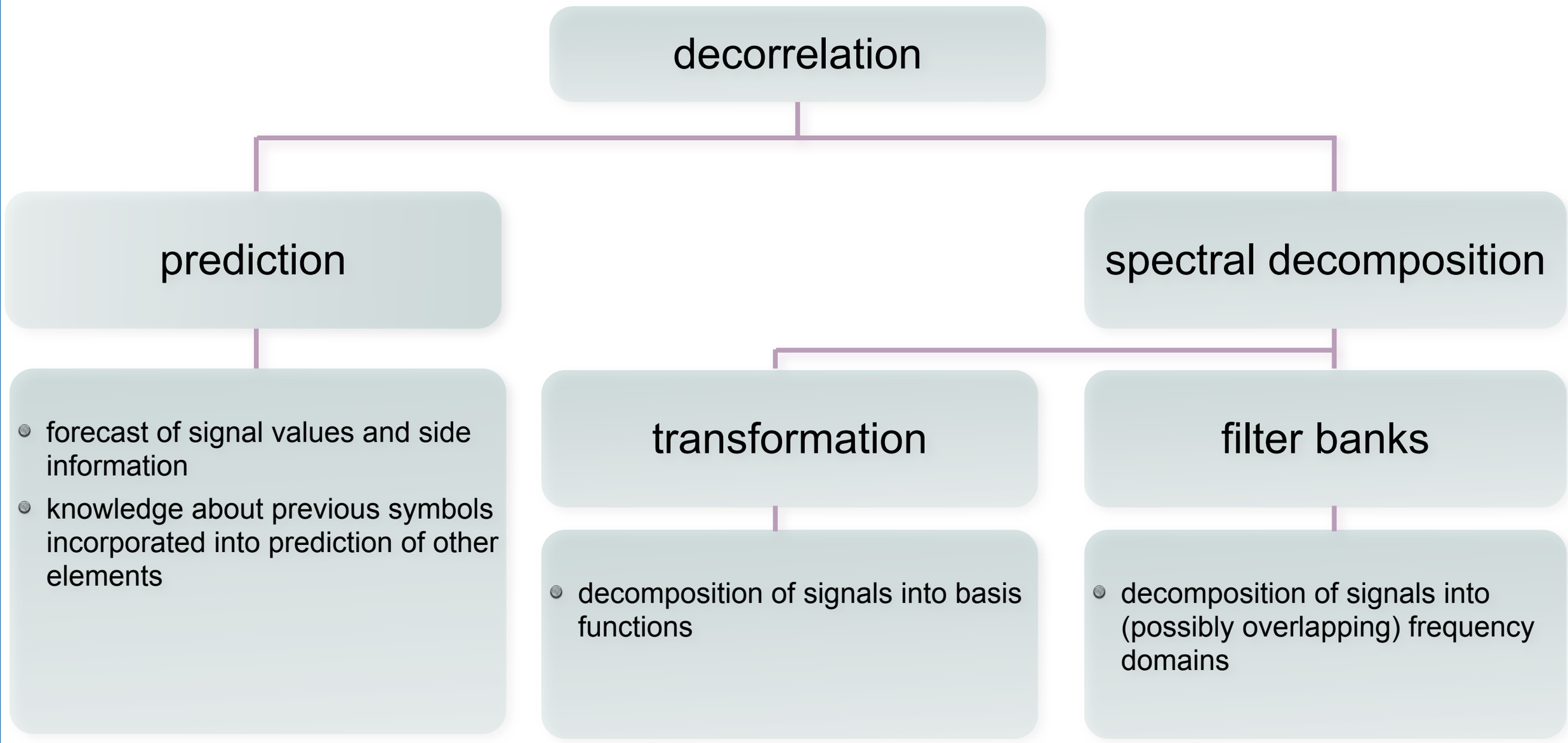
other

- any method providing information about symbol relationship can aid in reduction of inter-symbol redundancy

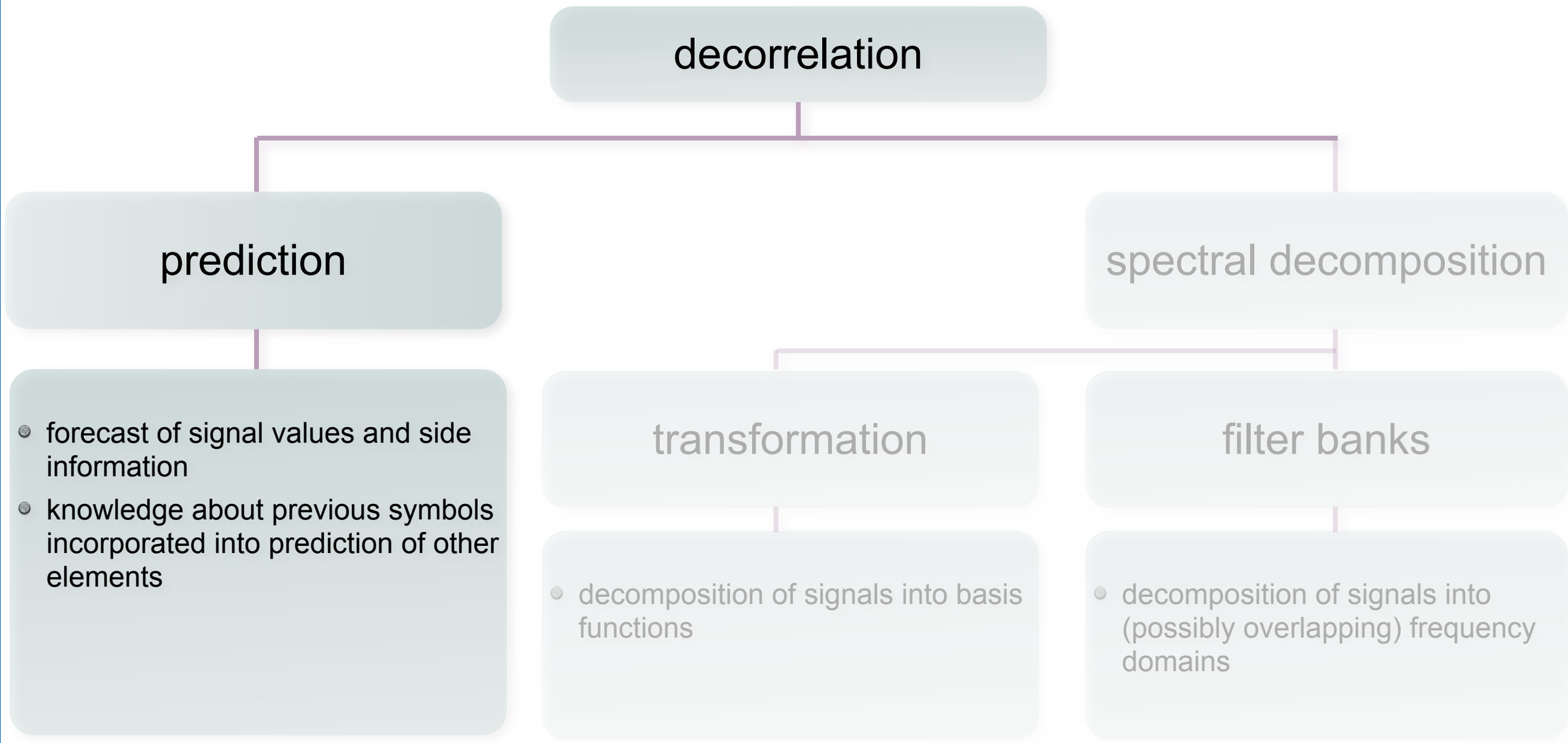
General data compression tools



Tools for concentration of signal energy / information

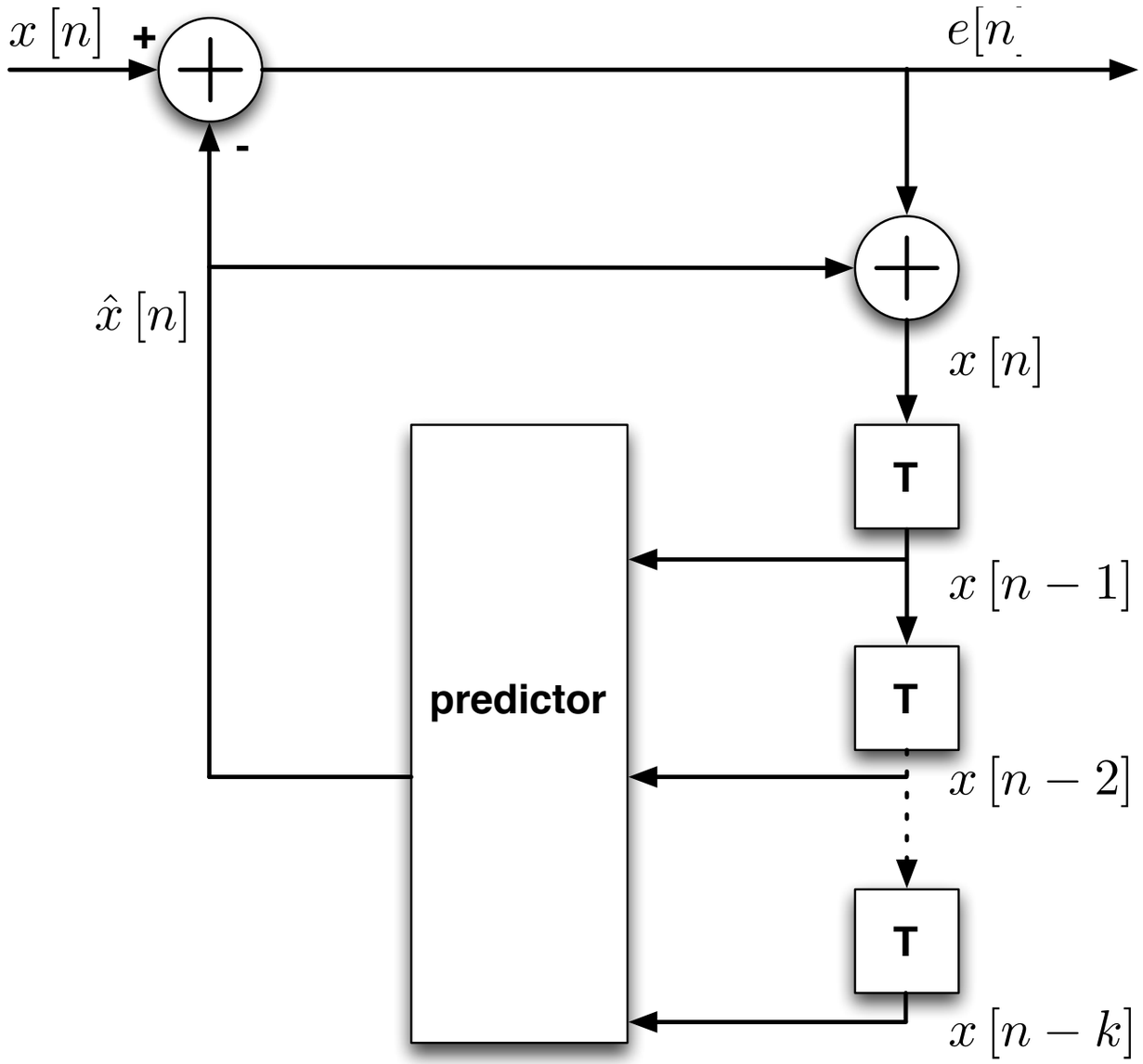


Tools for concentration of signal energy / information

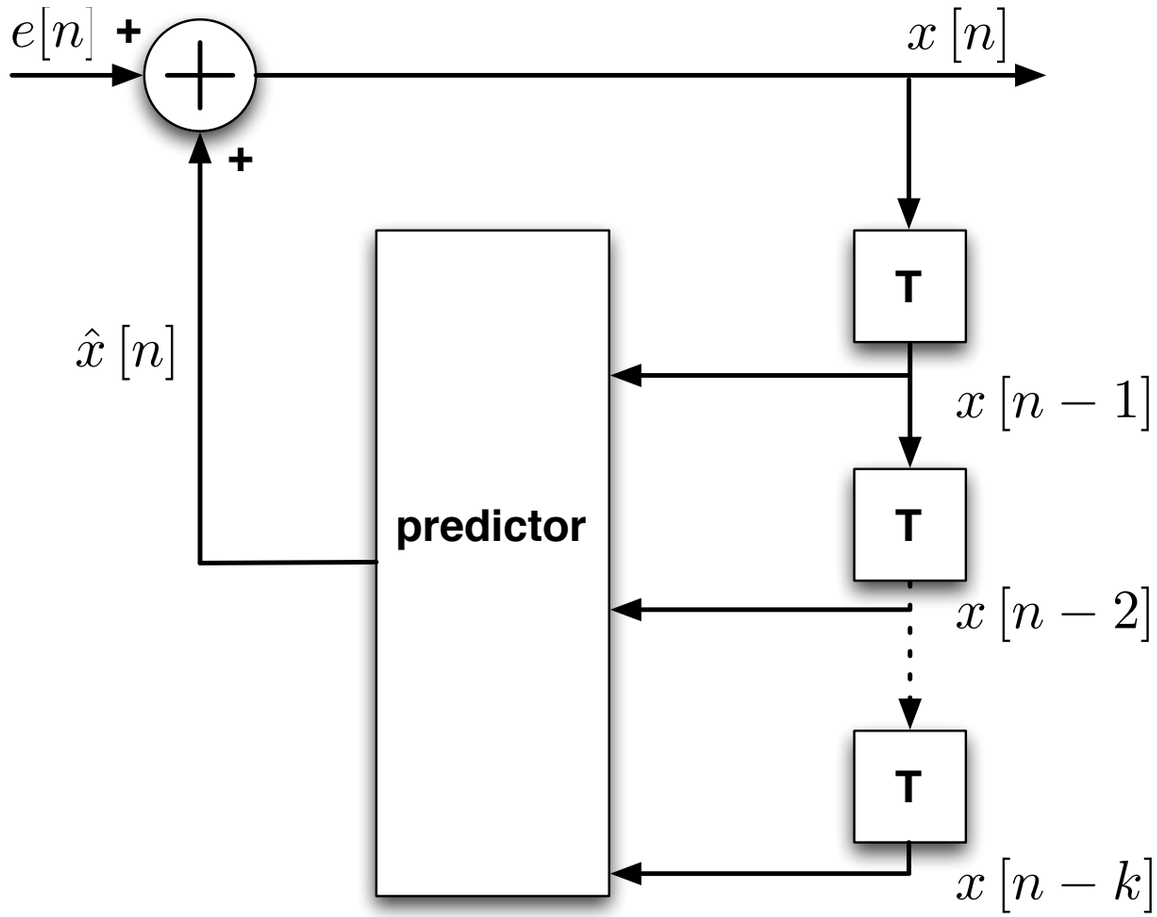


Prediction

Transmitter

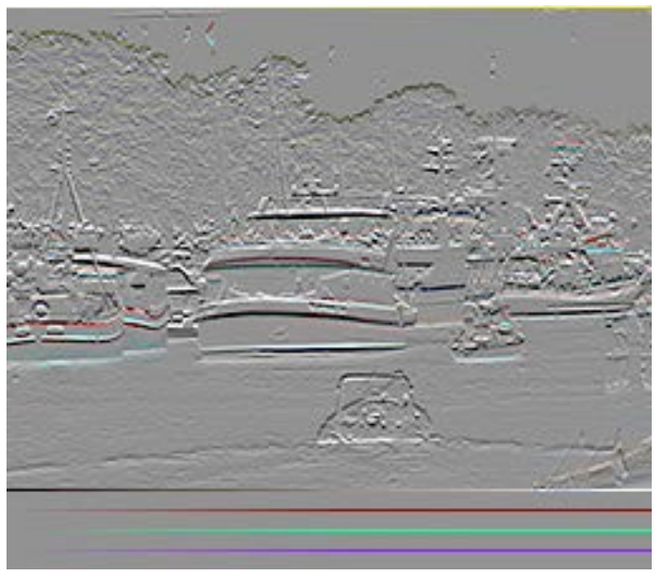


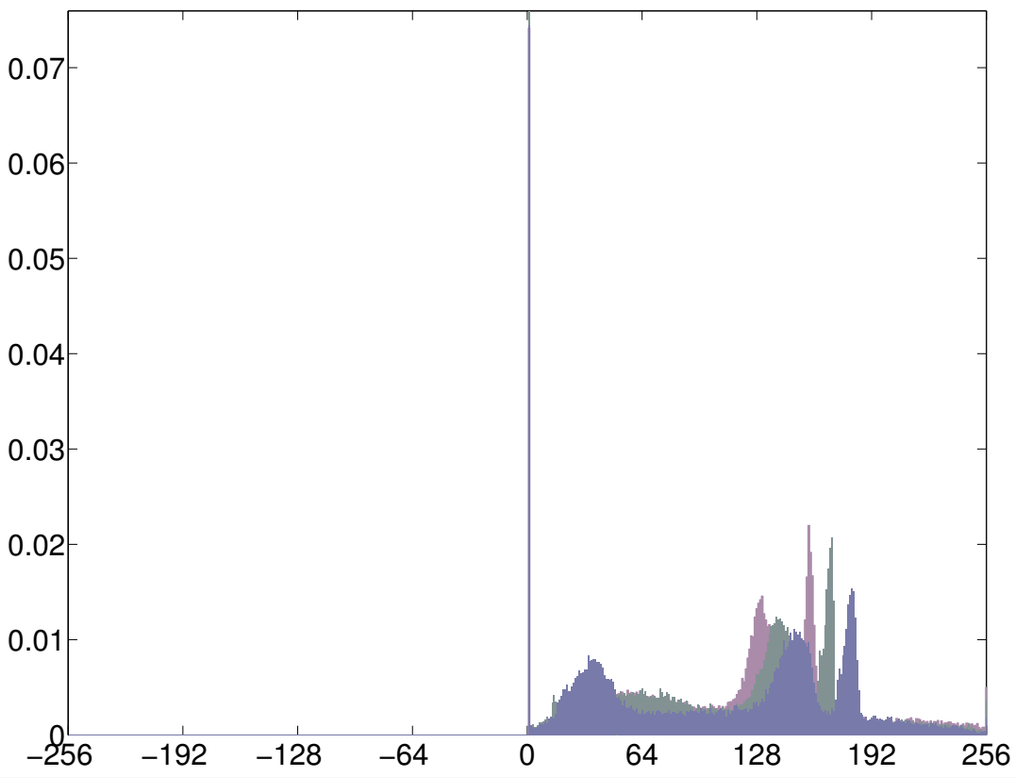
Receiver

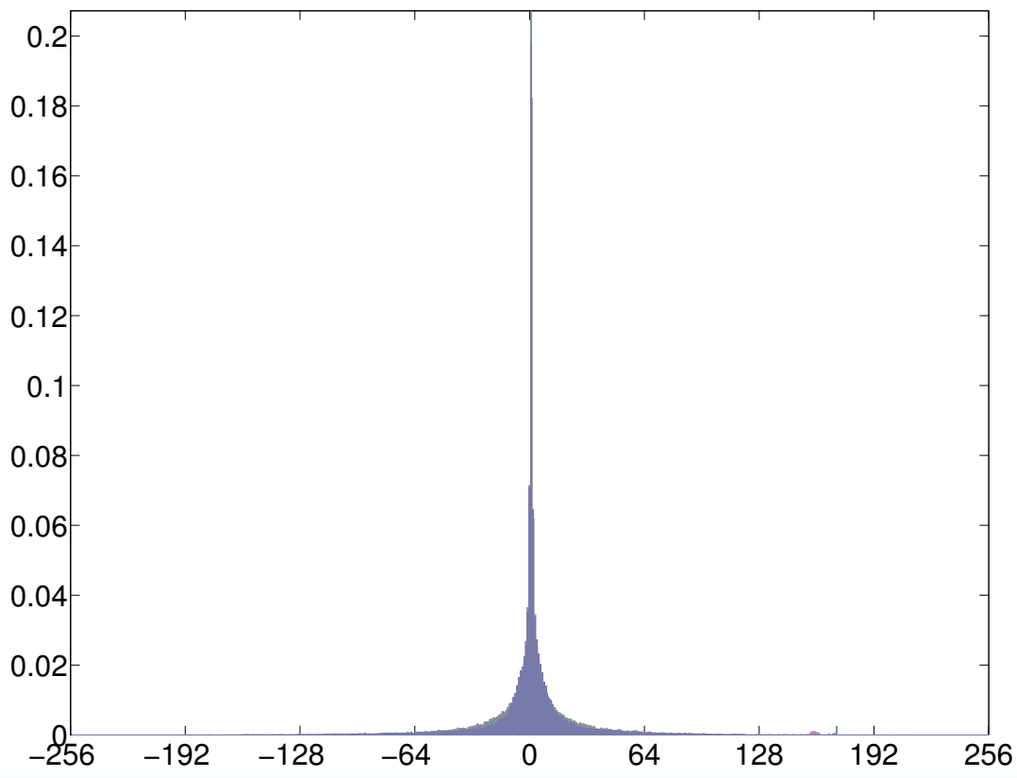


Example: $e[x,y] = s[x,y] - s[x,y-1]$

Histogram reveals concentration of signal energy towards 0







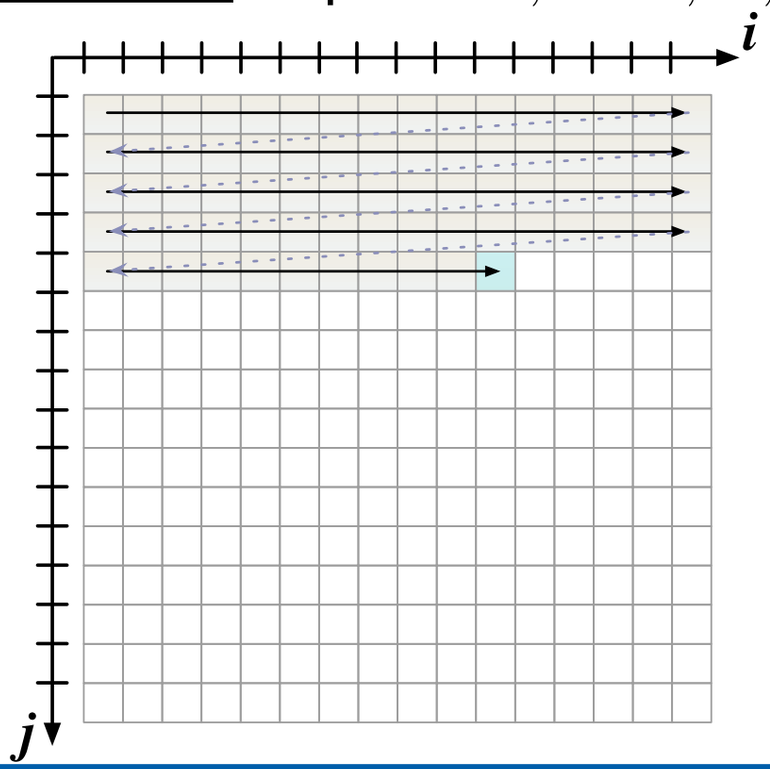
2D prediction

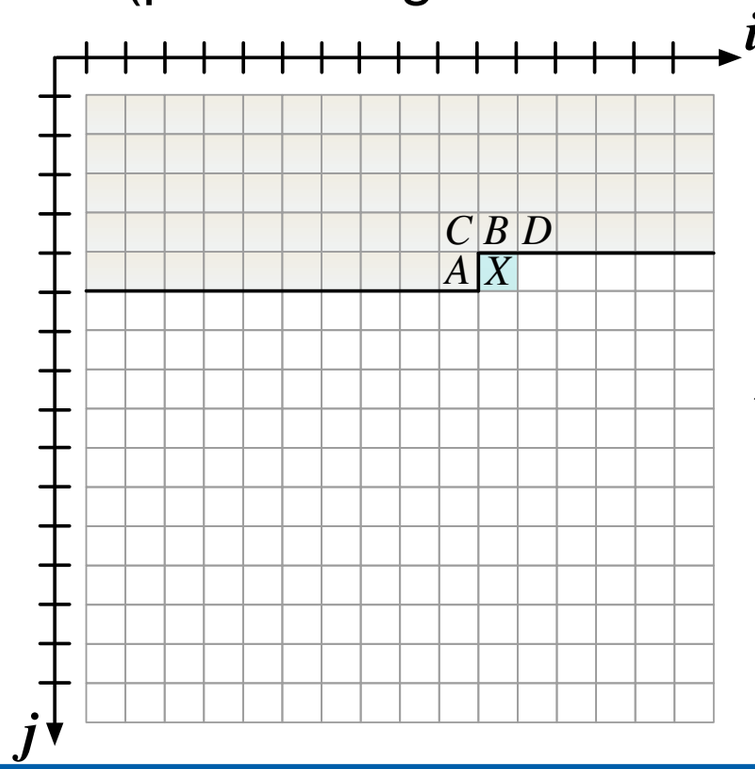
- two-dimensional signals can be predicted in the same way as 1D signals, however
  - correlation between direct neighbors (vertical, horizontal) higher than for predictors of greater distance

- general 2D linear prediction $\hat{x}[i,j] = \sum_k \sum_l a_{k,l}\, x[i-k, j-l]$

- for grayscale intensity $x[\,i,j\,]$ at position (i,j)

- causal prediction requires $k,l > 0;\ k,l \in \mathbb{N}$ (processing in raster scan order)

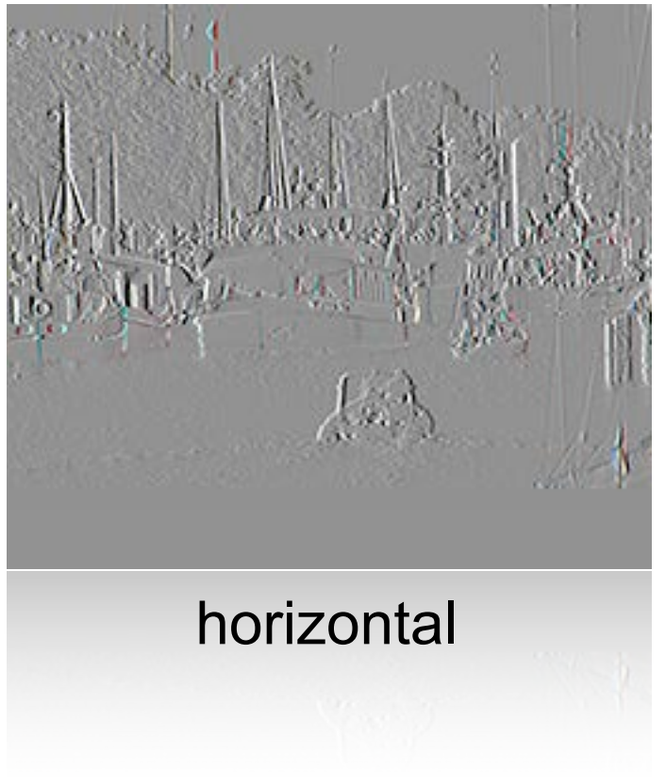




$$\hat{X} = f(A,B,C,D)$$

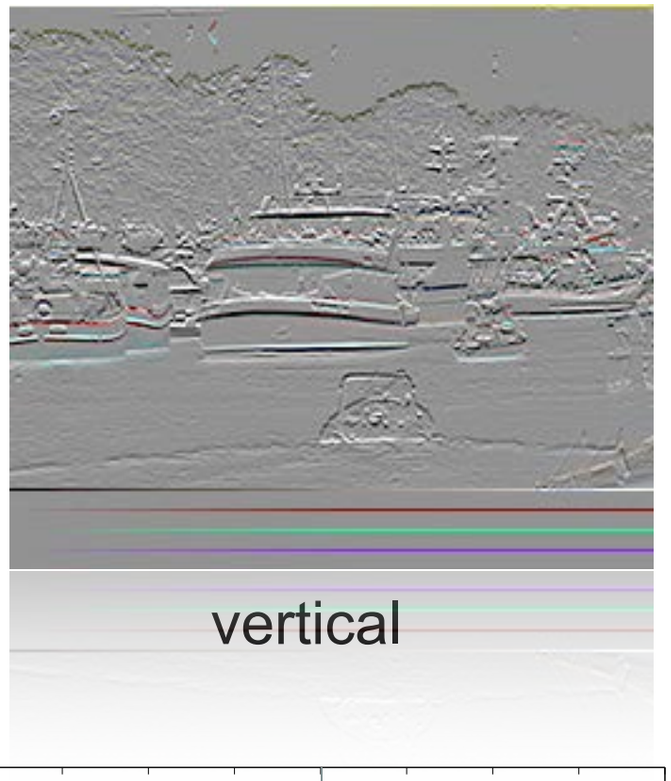
2D prediction example

$\hat{x}[i,j] = x[i,j] - x[i-1,j]$



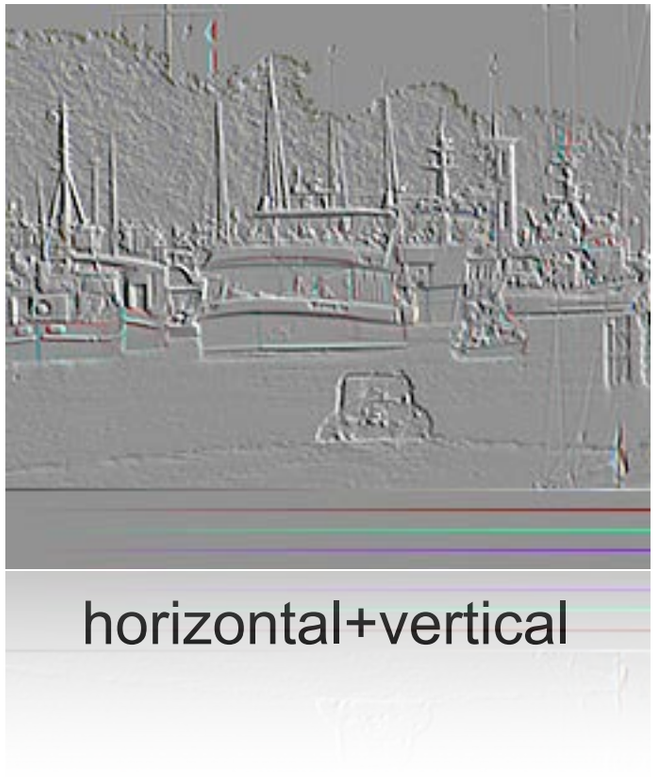
horizontal

$\hat{x}[i,j] = x[i,j] - x[i,j-1]$

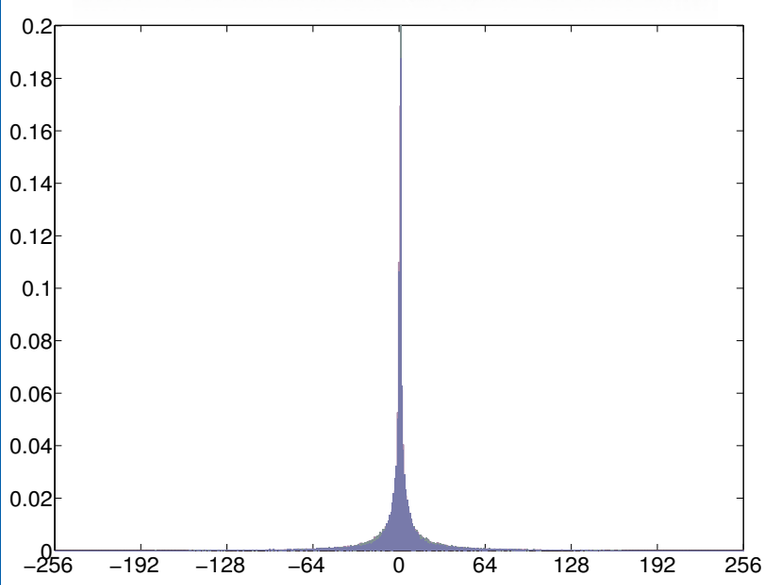


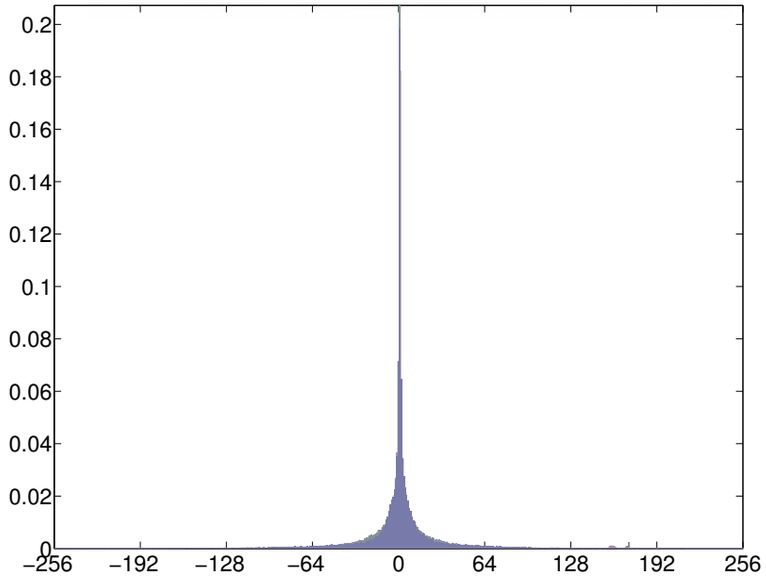
vertical

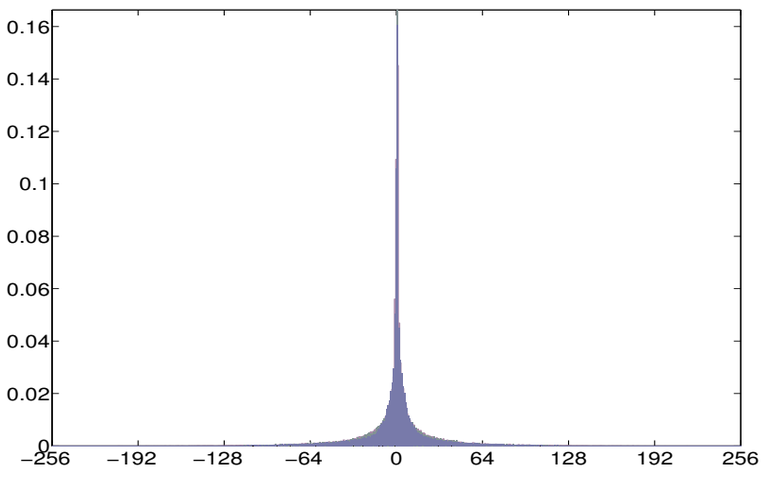
$\hat{x}[i,j] = x[i,j] - \frac{1}{2}\Big(x[i,j-1] + x[i-1,j]\Big)$



horizontal+vertical







Non linear prediction

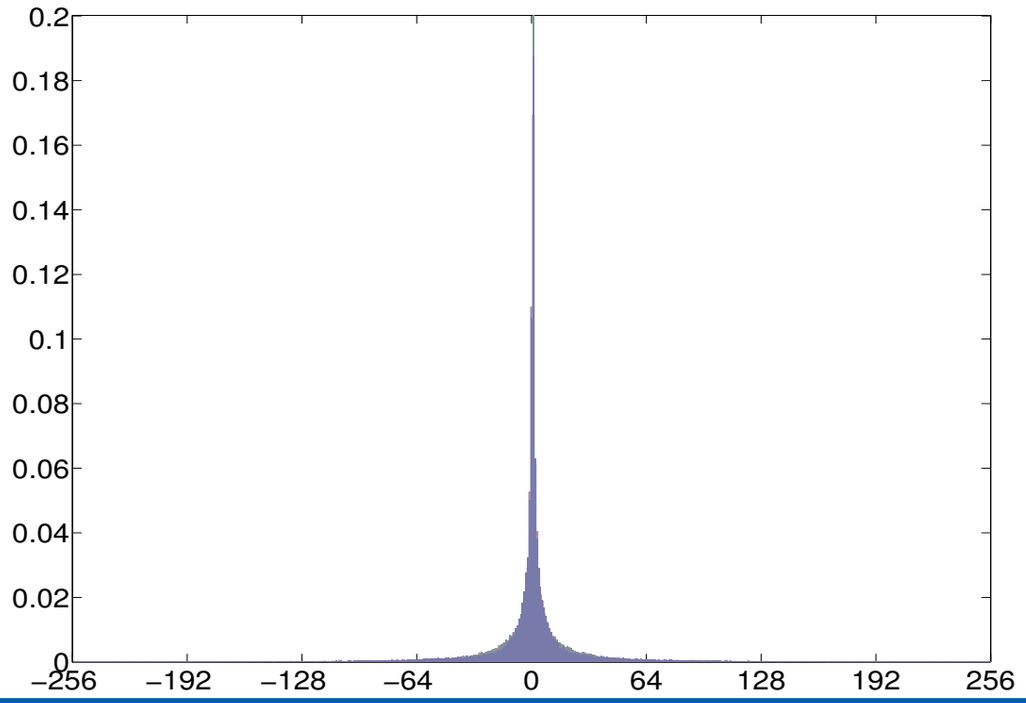
- conditional choice of predictors, their weights and prediction algorithm
- context adaptive selection based on
  - texture
  - edges
  - median of nearest symbols
  - ...
- example(1): JPEG-LS $\hat{x} = off[cx] + \begin{cases} min(A,B) & \text{for} \quad C > max(A,B) \\ max(A,B) & \text{for} \quad C < min(A,B) \\ A - C + B & \text{else} \end{cases}$
  - adaptation of offset off dependent on context cx can improve accuracy of prediction
- example(2): Median prediction variants

$$\hat{x} = A + B + D - max(A,B,D) - min(A,B,D)$$
$$\hat{x} = A + B + C - max(A,B,C) - min(A,B,C)$$

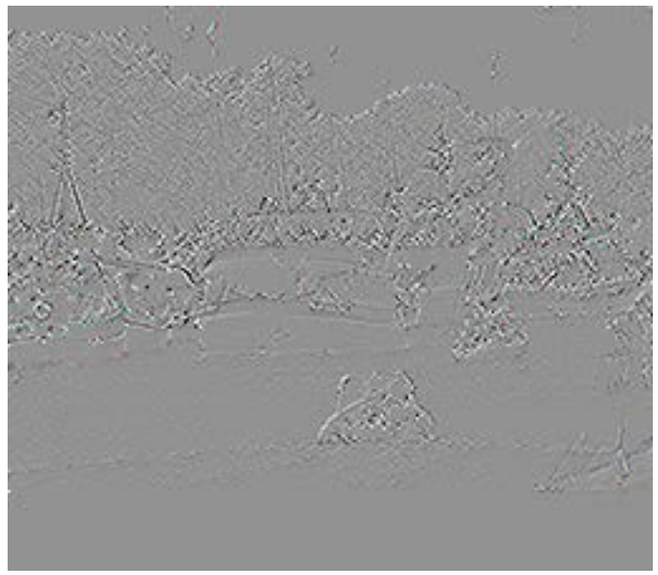
  - Median prediction is extensively used in side information coding (e.g. motion vectors)

2D non linear prediction example

horizontal linear $\hat{x}[i,j] = x[i,j] - x[i-1,j]$

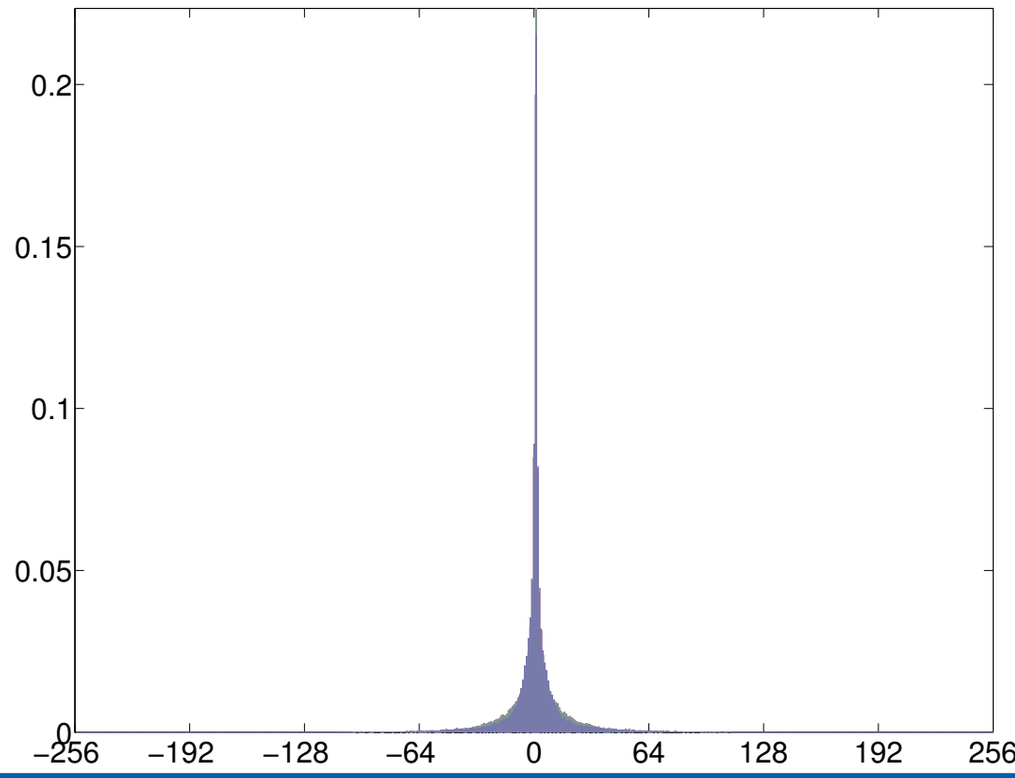


JPEG-LS*



*off[cx]=0

non linear prediction from 3 samples



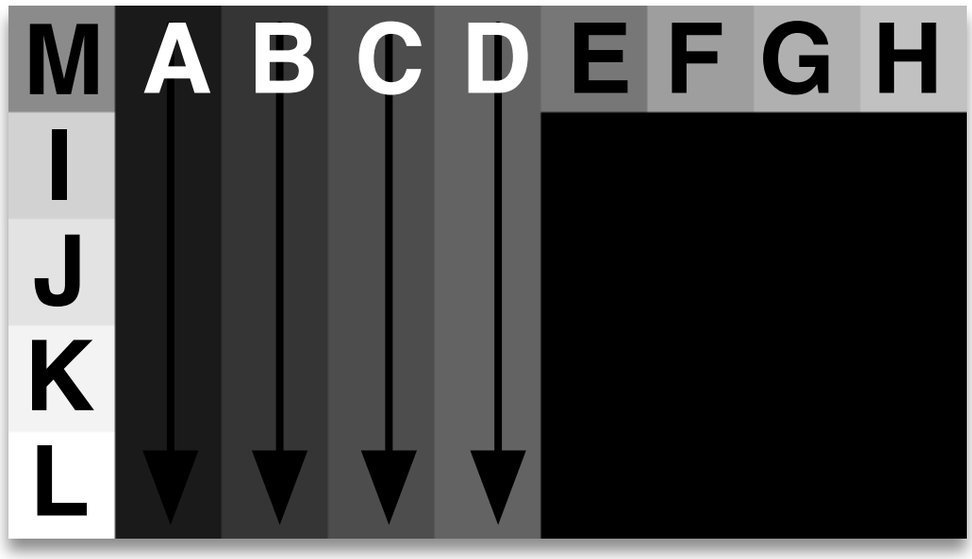
Spatial block prediction

- spatial prediction of more than a single symbol
- common approaches: (e.g. MPEG-4/ASP*, H.264)
  - vertical, horizontal, DC prediction

- prime example: H.264 FrExt
  - 4x4 block prediction (luma), 9 modes
  - 8x8 block prediction (luma), 9 modes
  - 16x16 block prediction (luma), 4 modes
  - 8x8 block prediction** (chroma), 4 modes

- transmission of chosen prediction mode plus quantized prediction error
  - non linear, non obvious decision of best mode for each block
    - prediction error quantization in transform space
    - best mode provides optimal quality with respect to bit rate (➠ rate-distortion problem)
  - side information generated (potentially correlated, subject to prediction)
  - variable amount of side information (➠ rate-distortion problem)

4x4 vertical prediction



**YCbCr4:2:0

*MPEG-4/ASP in DCT domain

Temporal block prediction

- high correlation between individual frames in image sequences and video
  - temporal block prediction by copying content from previously transmitted frames into new frames
  - object displacement between individual frames handled by motion compensation

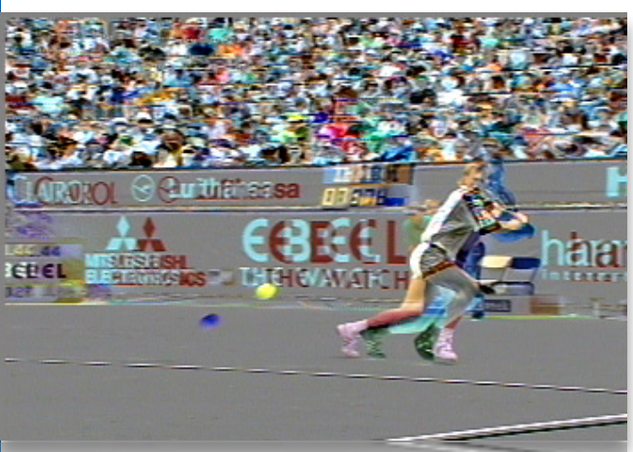
➠ recursion between frames introduces inter-frame dependencies

➠ high complexity in optimal mode selection and motion search

➠ generated side information: motion vector field, block partitioning, prediction direction

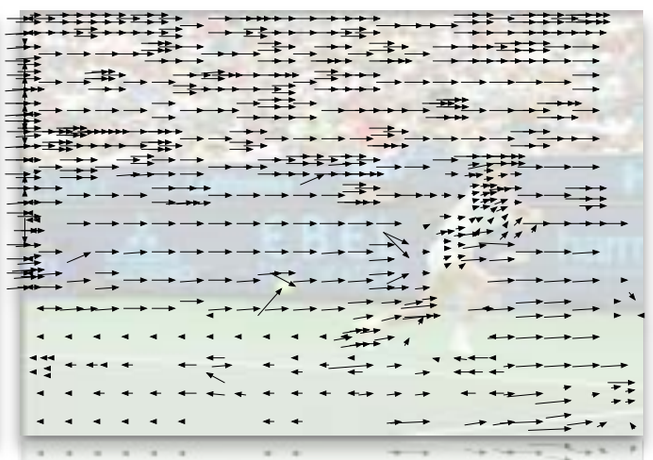
➠ considerable amount of correlated side information, subject to prediction

frame difference w/o MC

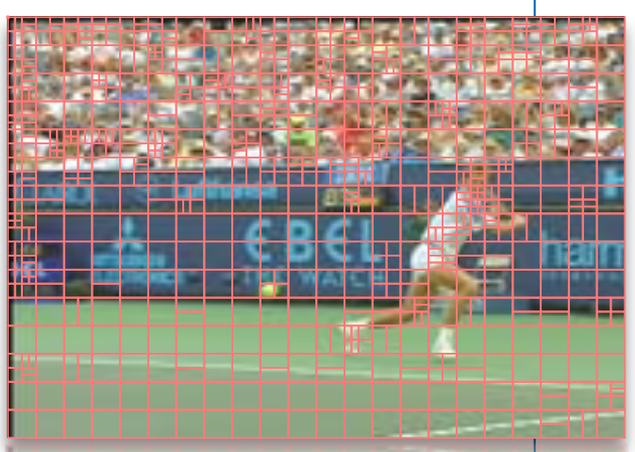


frame difference with MC

motion vector field



block partitioning



Tools for concentration of signal energy / information

- **decorrelation**
  - **prediction**
    - forecast of signal values and side information
    - knowledge about previous symbols incorporated into prediction of other elements
  - **spectral decomposition**
    - **transformation**
      - decomposition of signals into basis functions
    - **filter banks**
      - decomposition of signals into (possibly overlapping) frequency domains

example: relevant signal information concentrated in few coefficients

original image

16x16 block transform (DCT-II)

re-grouping by frequency

8x8 block transform (DCT-II)

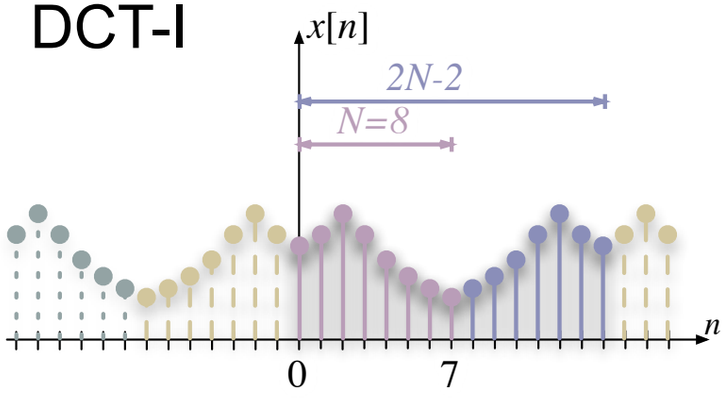
re-grouping by frequency

Discrete Cosine Transform

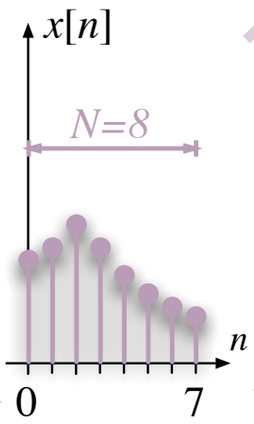
- real valued signal transform from space/time domain to frequency domain [ANR74]
- concept
  - the discrete fourier transform (DFT) provides energy compaction
    - suitability for compression is limited due to complex spectral components (in typical cases)
  - however: DFT of real even (symmetric) signals results in purely real valued spectrum
  - key idea is to generate a symmetric signal from the input signal whose DFT results in a real spectrum
- 8 DCT variants available
  - differences are in the way of symmetric extension
    - direct extension (mirroring around $x[0], x[N-1]$)
    - half shifted extension (mirroring including $x[0], x[N-1]$)
    - even/odd mirroring (+/-)
  - DCT-I to DCT-IV perform either direct or half shifted extension
  - DCT-V to DCT-VIII perform direct extension one one, half shifted extension on the other boundary (rarely used in practice)

Example: Symmetric extension DCT-I...DCT-IV

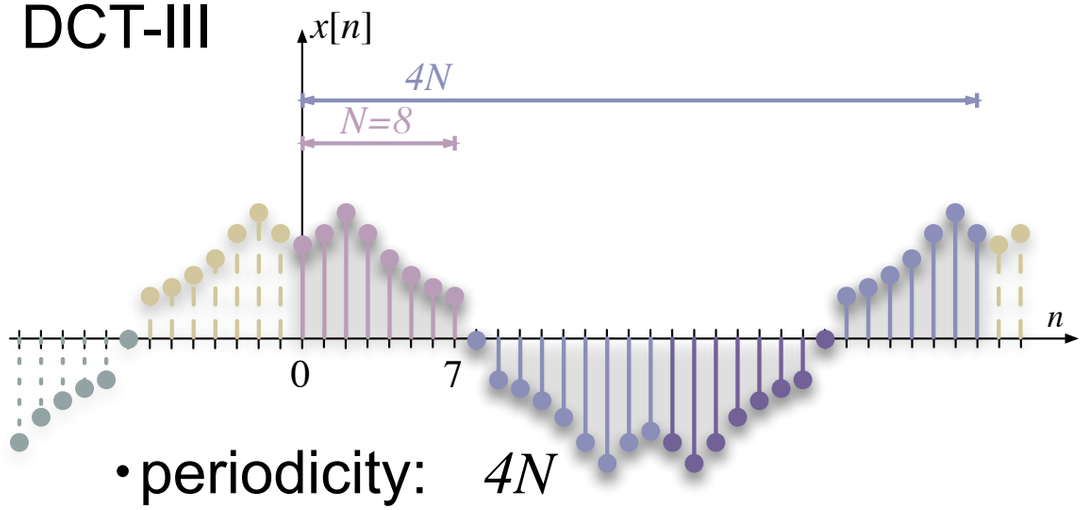
DCT-I



- periodicity: $2N-2$
- symmetry: $n=0;\ n=N-1$

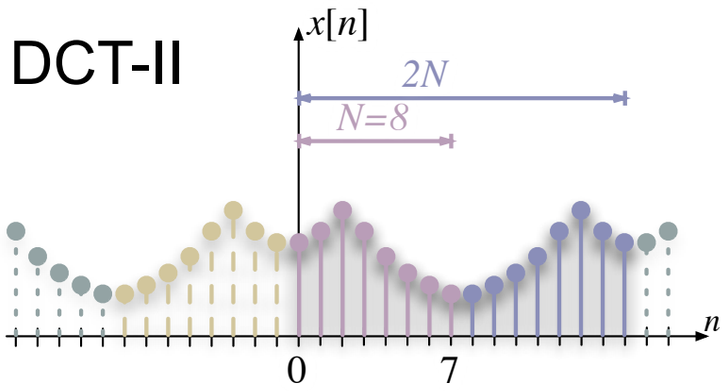


DCT-III



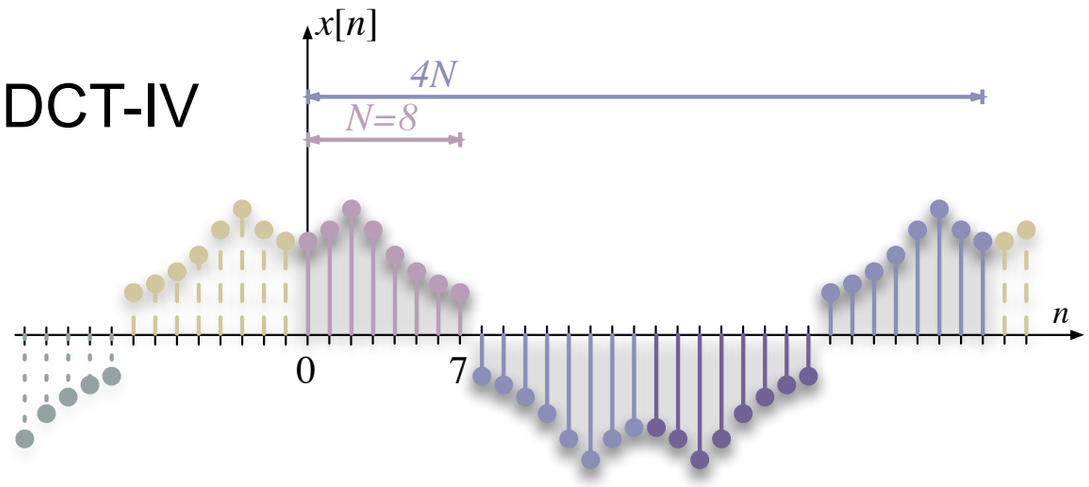
- periodicity: $4N$
- symmetry: $n=0;\ n=2N$

DCT-II



- periodicity: $2N$
- symmetry: $n=-0.5;\ n=N-0.5$

DCT-IV



- periodicity: $2N$
- symmetry: $n=-0.5;\ n=2N-0.5$

DCT-I...DCT-IV transform kernels

- Transform kernels for forward and inverse transform are the basis functions
  - i.e. cosine functions of different frequency
- Type I

$$\sqrt{\frac{2}{N}}\, w[n]w[k] \cos\left(\frac{kn\pi}{N}\right) \qquad k,n = 0,1,\ldots,N$$

- Type II

$$\sqrt{\frac{2}{N}}\, w[k] \cos\left(\frac{k(n+0.5)\pi}{N}\right) \qquad k,n = 0,1,\ldots,N-1$$

- Type III

$$\sqrt{\frac{2}{N}}\, w[n] \cos\left(\frac{(k+0.5)n\pi}{N}\right) \qquad k,n = 0,1,\ldots,N-1$$

- Type IV

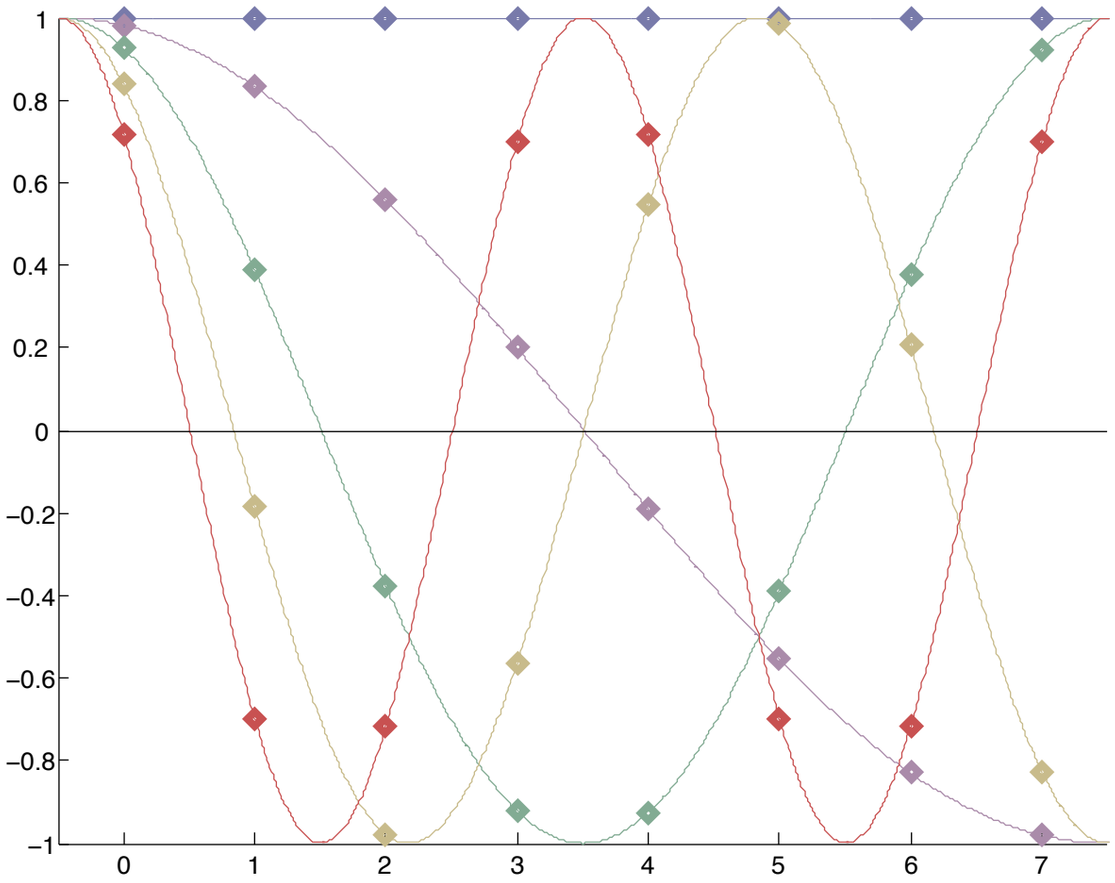
$$\sqrt{\frac{2}{N}} \cos\left(\frac{(k+0.5)(n+0.5)\pi}{N}\right) \qquad k,n = 0,1,\ldots,N-1$$

  - factors $\sqrt{\frac{2}{N}}$ and weights $w[m]$ make the DCT orthogonal and orthonormal

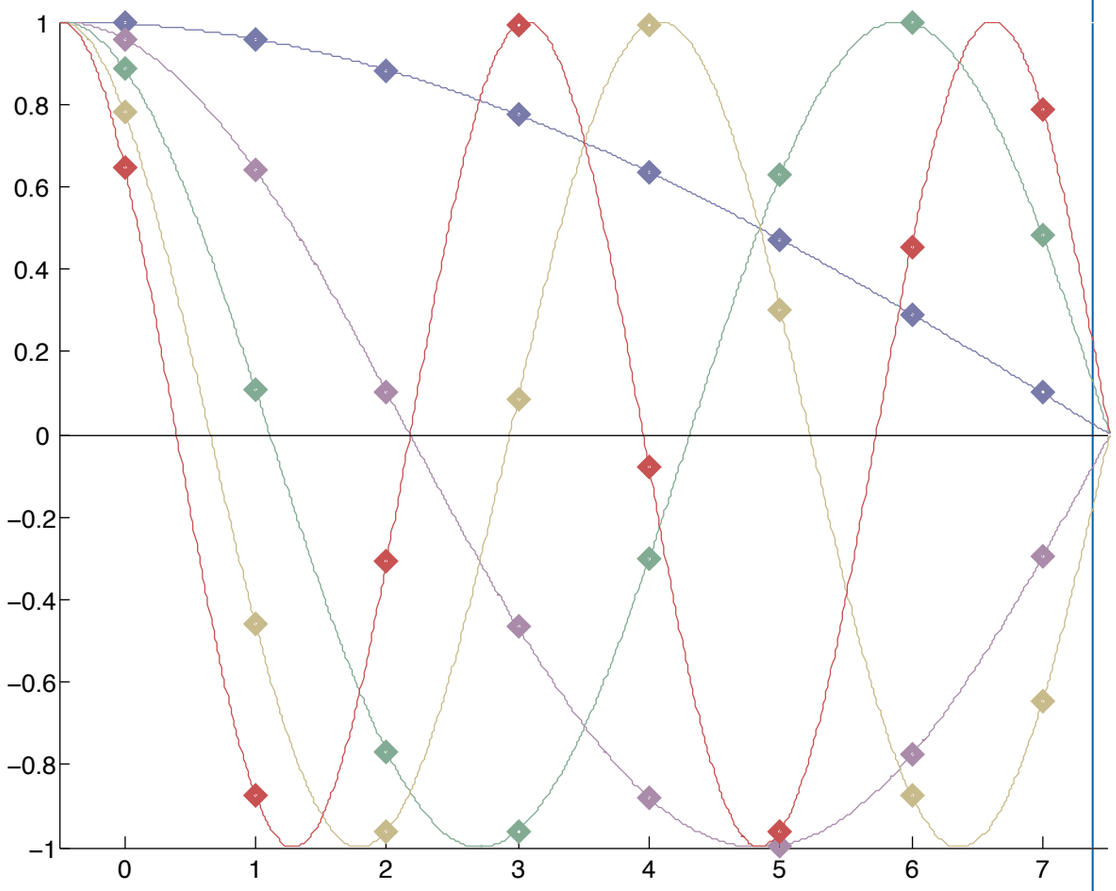
$$w[m] = \begin{cases} 0 & \text{if } m < 0 \text{ or } m > N \\ \frac{1}{\sqrt{2}} & \text{if } m = 0 \text{ or } m = N \\ 1 & \text{if } 0 < m < N \end{cases}$$

Example: 1D DCT basis functions

DCT-II



DCT-IV



k= 0
1
2
3
4

- example shows first 5 (non normalized) basis functions of 1D transform with N=8
- DCT-II has constant as basis function k=0 → image/video coding
- DCT-IV has low frequency wave as basis function k=0 → audio coding

DCT-II transform

- DCT-II offers favorable energy concentration in images [RY90]
  - first base function is a constant, important for flat image regions
  - for first order stationary Markov-processes near optimal decorrelation (AR(1) models)

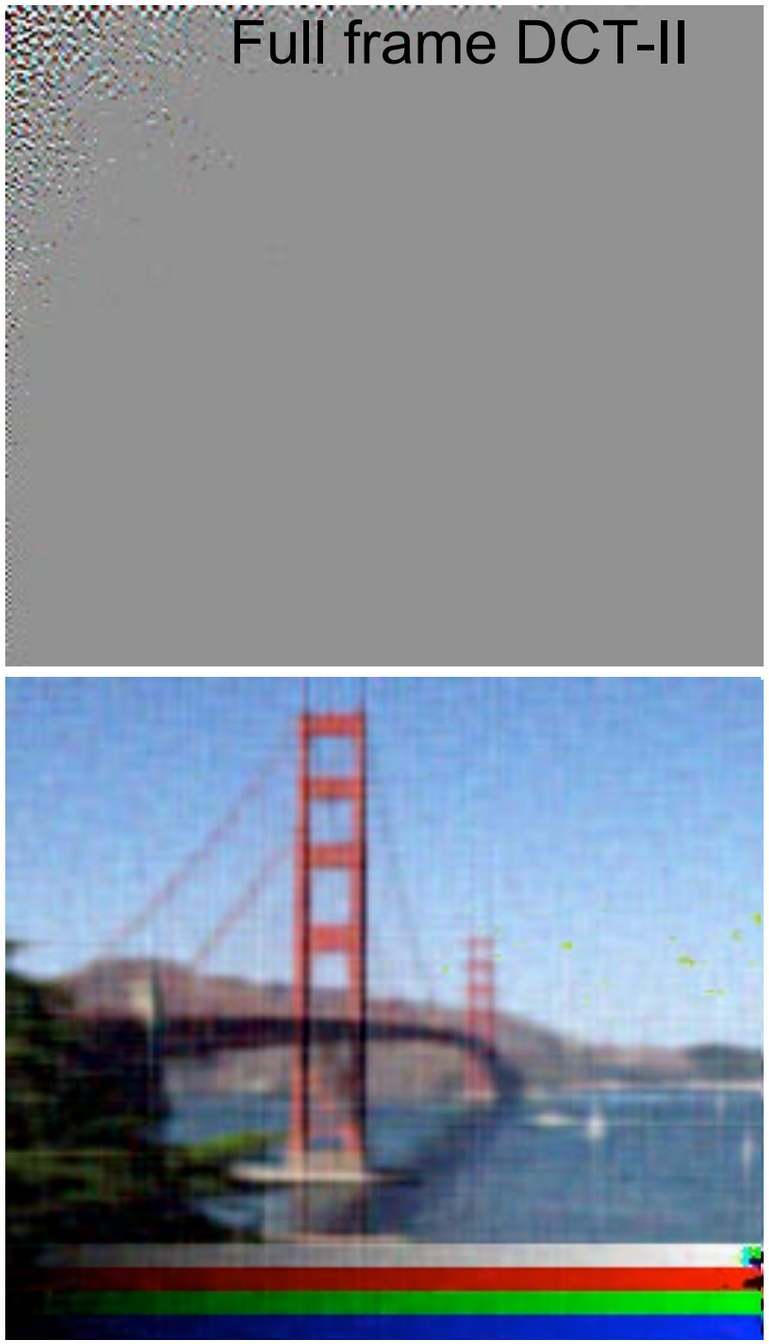
- DCT-III is the inverse transform of DCT-II
- used in all relevant standards for image/video compression (JPEG,MPEG,H.264*)
- Commonly used block size for image coding is 8x8

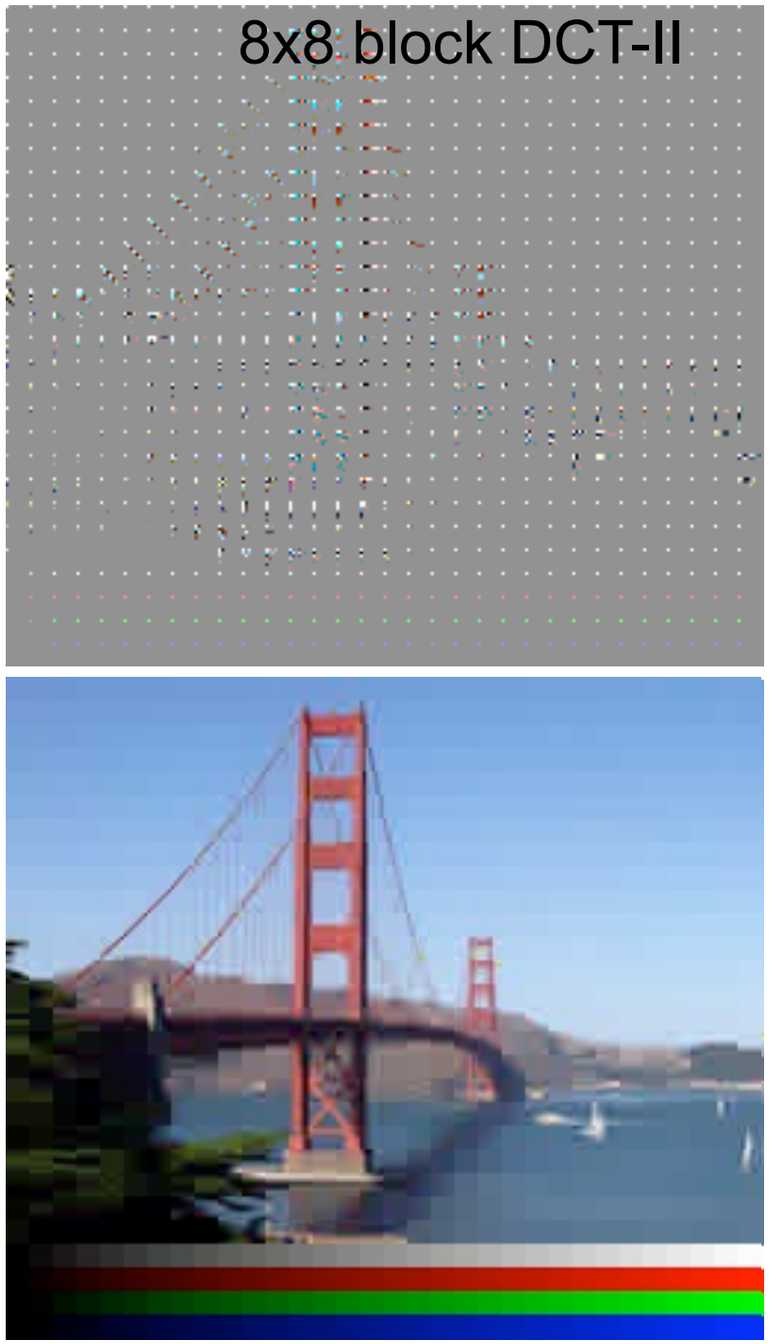
- Coefficients are real valued, perfect reconstruction requires FP calculation
  - integer arithmetic possible, however with limited accuracy
  - IEEE standard 1180 enforces maximum permissible drift among different implementations

- Special variants
  - shape adaptive DCT (SADCT) in MPEG-4 Part 2
  - 4x4 / 8x8 integer DCT in H.264 / MPEG-4 AVC

* heavily quantized form

Reason for block based DCT





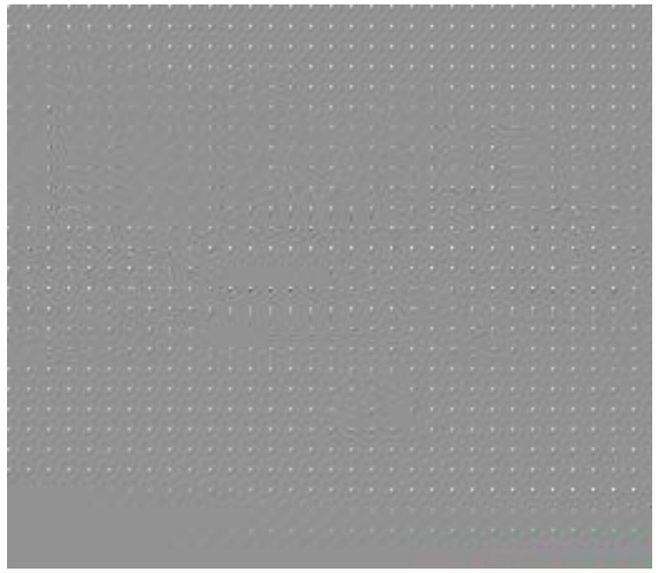
- **Quantization errors are distributed over size of basis functions.**

• quant: $q[x,y] = \begin{cases} 0 & \text{for } |X[x,y]| < 50 \\ X[x,y] & \text{for } |X[x,y]| \geq 50 \end{cases}$

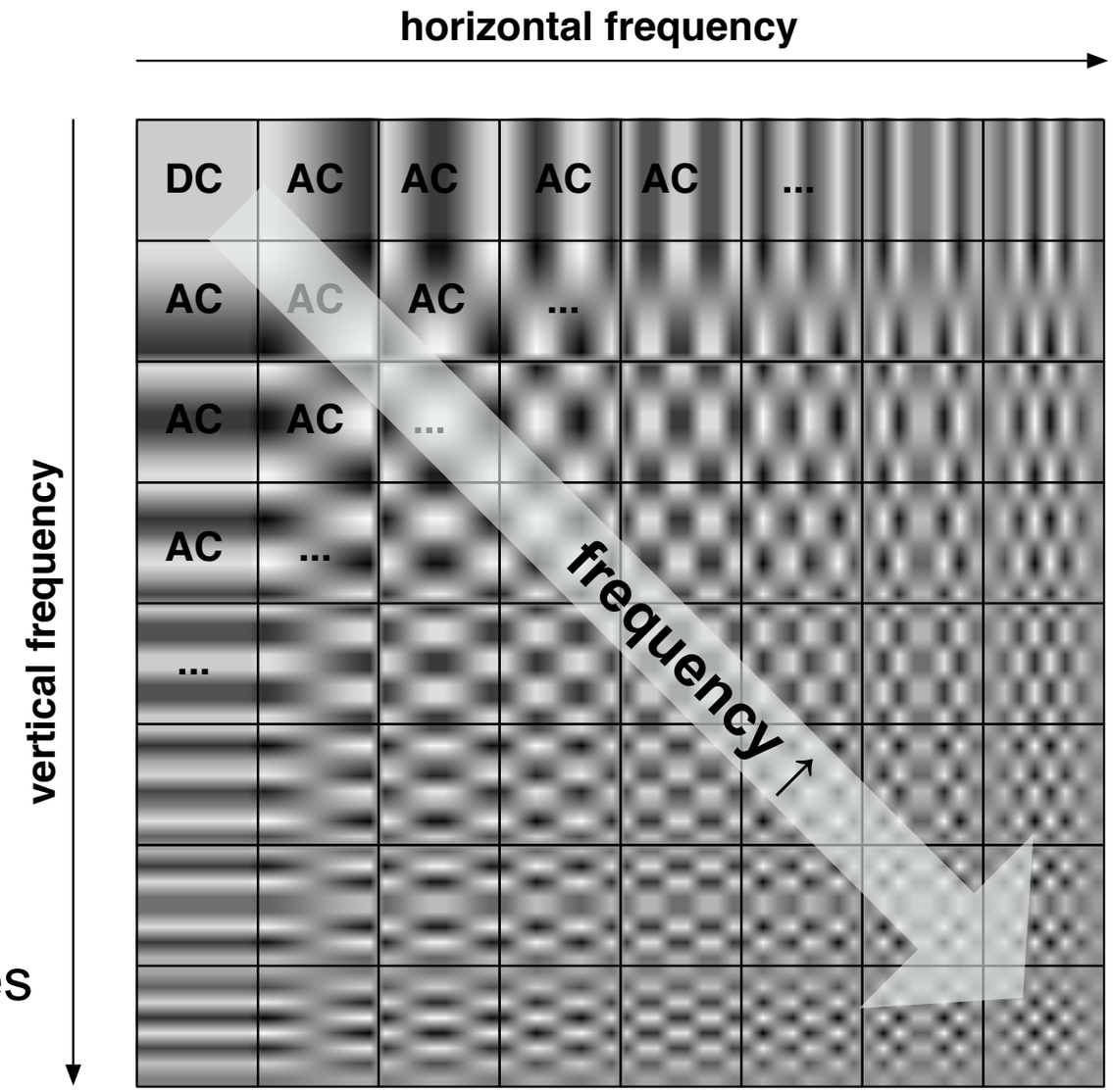
DCT-II of images

input image

8x8 block DCT-II



8x8 basis functions



- Images are divided into blocks of NxN samples
- every block is transformed individually
- transformed blocks consist of DC and AC components
- frequency progression within block in diagonal direction

2D DCT-II / DCT-III formal for NxN samples

- DCT (DCT-II)

$$S[u,v] = \frac{2}{N}\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} w[x]w[y]s[x,y]cos\frac{(2x+1)u\pi}{2N}cos\frac{(2y+1)v\pi}{2N}$$

with

$u, v, x, y = 0, 1, \ldots, N-1$

x, y coordinates in spatial domain

u, v coordinates in frequency domain

$$w[x], w[y] = \begin{cases} \frac{1}{\sqrt{2}} & \text{for } u, v = 0 \\ 1 & \text{else} \end{cases}$$

- IDCT (DCT-III)

$$s[x,y] = \frac{2}{N}\sum_{u=0}^{N-1}\sum_{v=0}^{N-1} w[u]w[v]S[u,v]cos\frac{(2x+1)u\pi}{2N}cos\frac{(2y+1)v\pi}{2N}$$

with

$u, v, x, y = 0, 1, \ldots, N-1$

x, y coordinates in spatial domain

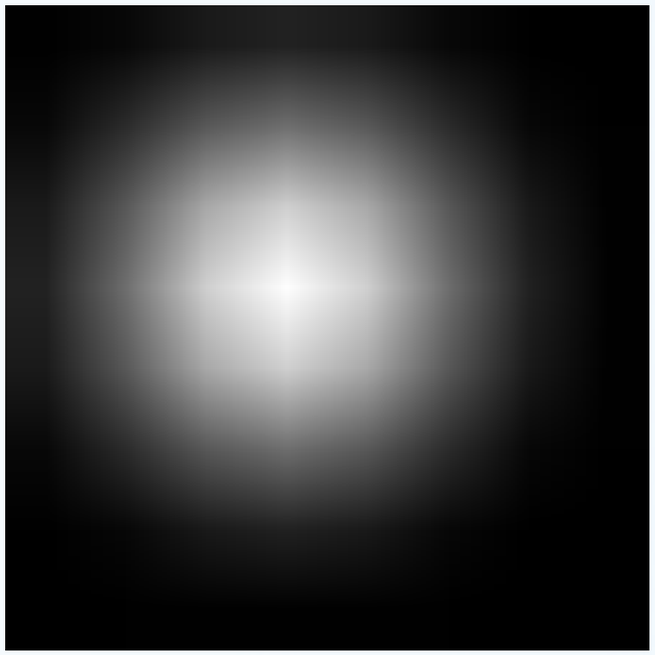
u, v coordinates in frequency domain

$$w[u], w[v] = \begin{cases} \frac{1}{\sqrt{2}} & \text{for } u, v = 0 \\ 1 & \text{else} \end{cases}$$

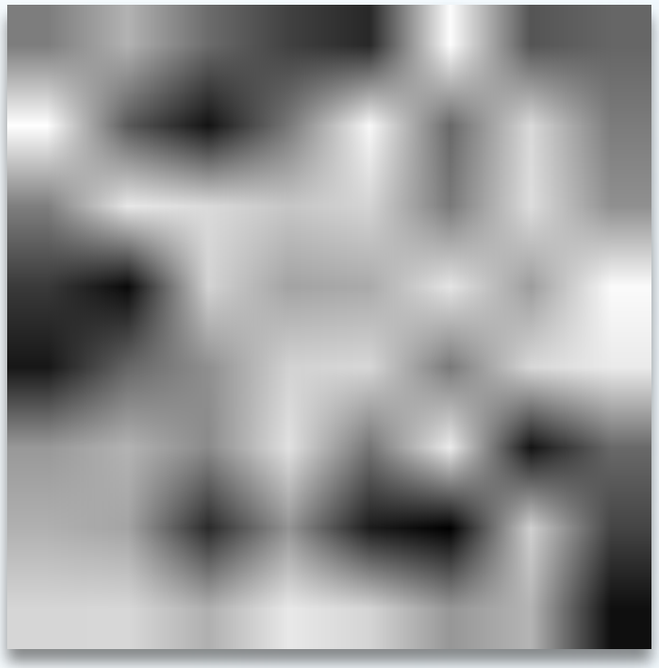
8x8 DCT examples: energy concentration

- spatial domain

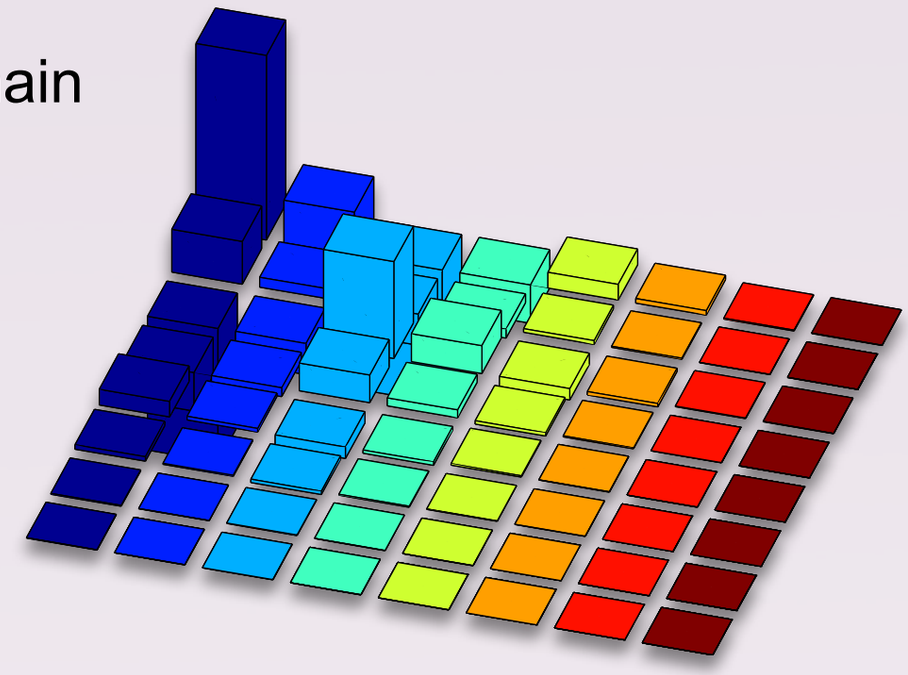
2D Gaussian

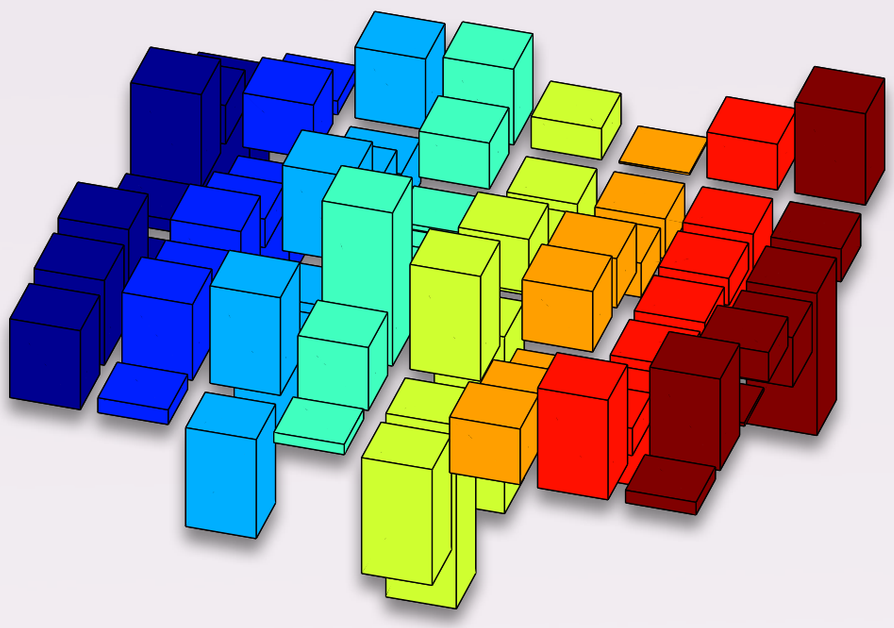


2D Noise



- DCT domain





References

- [SNT] Müller, E.: Statistische Nachrichtentheorie, winter semester lecture http://www.int.uni-rostock.de
- [DSV2] Müller, E.: Digital Signal Processing 2, winter semester lecture, http://www.int.uni-rostock.de
- [MAL92] Malvar H.S., „Signal Processing with Lapped Transforms“ (Artech House: Norwood MA, 1992).
- [MAL99] Malvar, H.S., "A Modulated Complex Lapped Transform And Its Applications to Audio Processing". Proc. International Conference on Acoustics, Speech and Signal Processing, 1999.
- [MAL03] Malvar, H.S., "Fast Algorithm for the Modulated Complex Lapped Transform", IEEE Signal Processing Letters, Vol. 10, No. 1, 2003.
- [ANR74] Ahmed, N.; Natarjan, T.; Rao, K.R., „Discrete cosine transform“. IEEE Transactions on Computers, 23:90–3, 1974
- [RY90] Rao, K.R., Yip, P., „Discrete Cosine Transform : Algorithms, Advantages, Applications“. Academic Press, San Diego, USA, 1st edition, 1990
- [FLI93] Fliege, N.: „Multiraten-Signalverarbeitung“. Verlag Teubner Stuttgart, 1993
- [STR05] Strutz, T.: „Bilddatenkompression“. Verlag Vieweg Wiesbaden, 2005