5.6. Transform Coding (1)

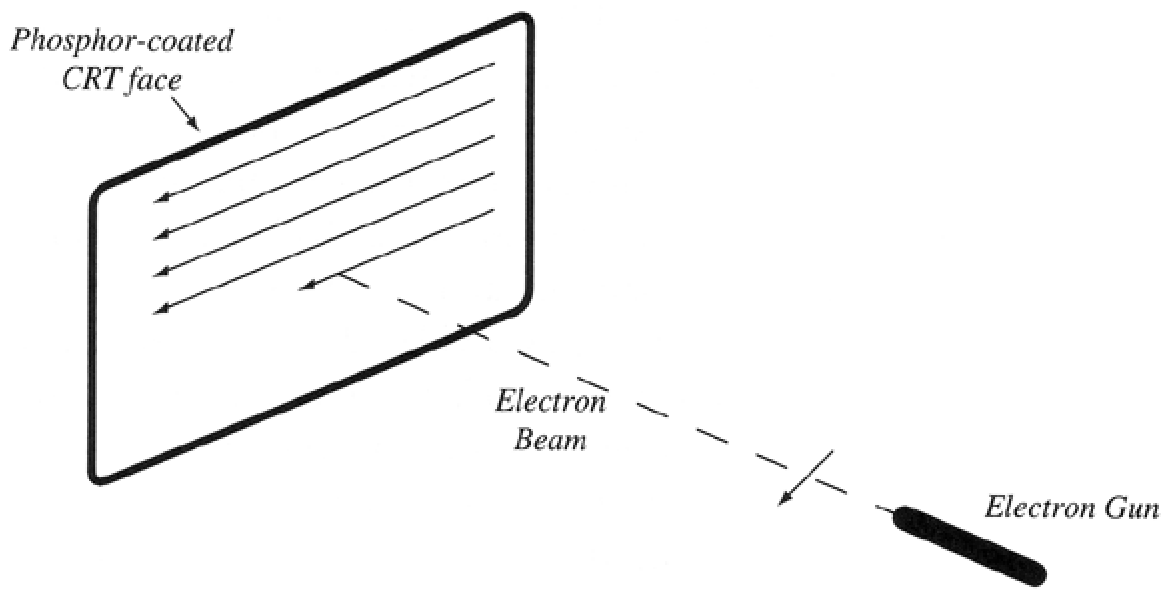
- Basic ideas:
  - The original source output sequence (e.g. audio sample values, image pixel) does not allow access to all resp. to the desired information characteristics.
  - Transform the information source output into another more appropriate representation resp. decompose the source output into proper components.
  - For that purpose apply a reversible transformation procedure that compacts most of the information of the source output sequence into a few elements of the transformed sequence. The transformation procedure has to ensure that the elements of the new sequence have differing (and better) statistics and perceptual importance.
  - For the reasons of simplicity and efficiency the length of single sequences is usually limited (e.g. 8 for 1-D data and 8x8=64 for 2-D data) denoting a division of the source output into blocks.

Transform Coding (2)

- Basic ideas: (cont.)
  - Pure transformation (decomposition) is lossless.
  - Compression takes place by quantizing, subsampling or discarding elements of the new sequence that do not contain much resp. important information.
- We consider:
  - color space transformation
  - transformation to frequency space (DCT, Wavelet)

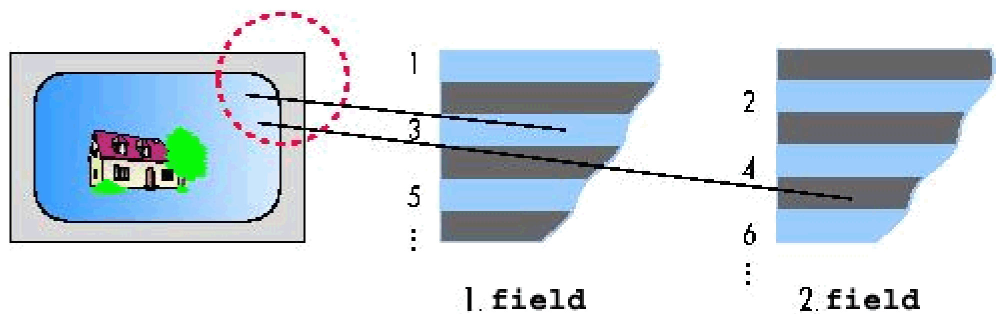
5.6.2. On TVs, fields, and frames... (1)

- Basics:
  - black and white television picture:
    - phosphor on the television screen is excited using an electron beam with modulated intensity to generate the image
    - electron beam traces screen line by line
    - image generated by traversal of beam has to be updated rapidly enough for persistence of vision to make the image appear stable



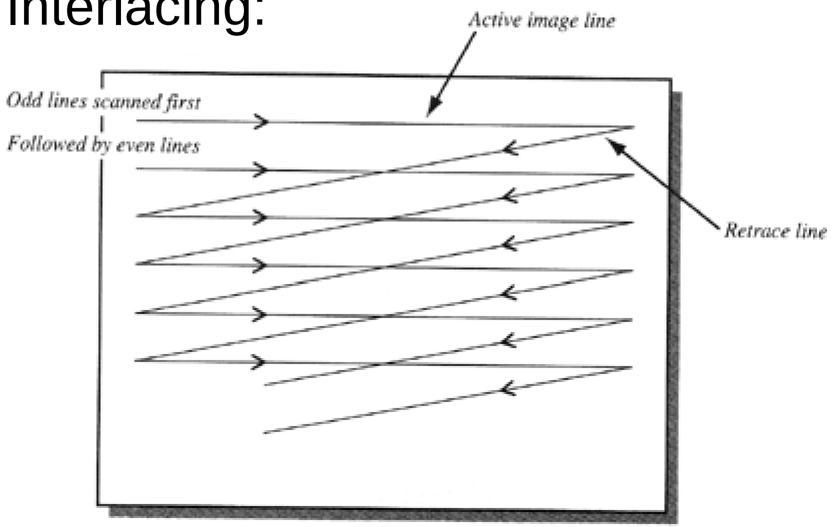
On TVs, fields, and frames... (2)

- Basics: (cont.)
  - color television picture:
    - same process but three electron beams for the three basic phosphor colors red, green, blue trace screen at the same time
  - One television picture ("frame") is build of two interlaced "fields" which are traced one after the other (in time) by the electron beams but which are nested concerning the lines on the screen (in space) => interlacing technique
  - Interlacing: 1. field: odd field, 2. field: even field

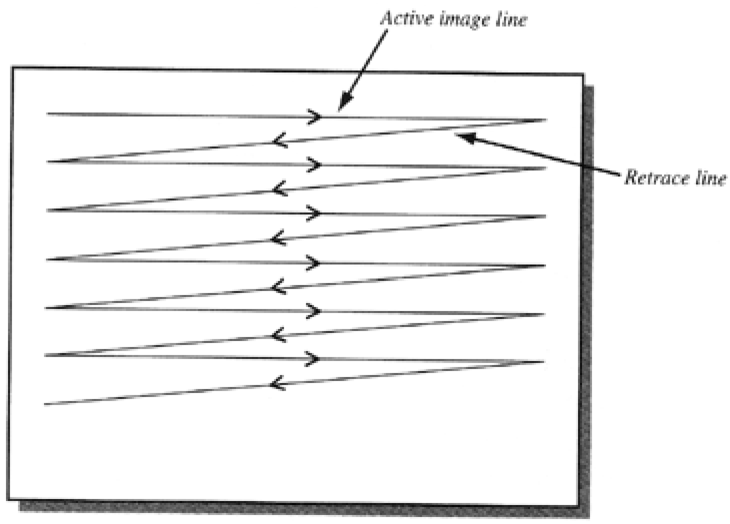


On TVs, fields, and frames... (3)

- Interlacing:



Interlaced Scan



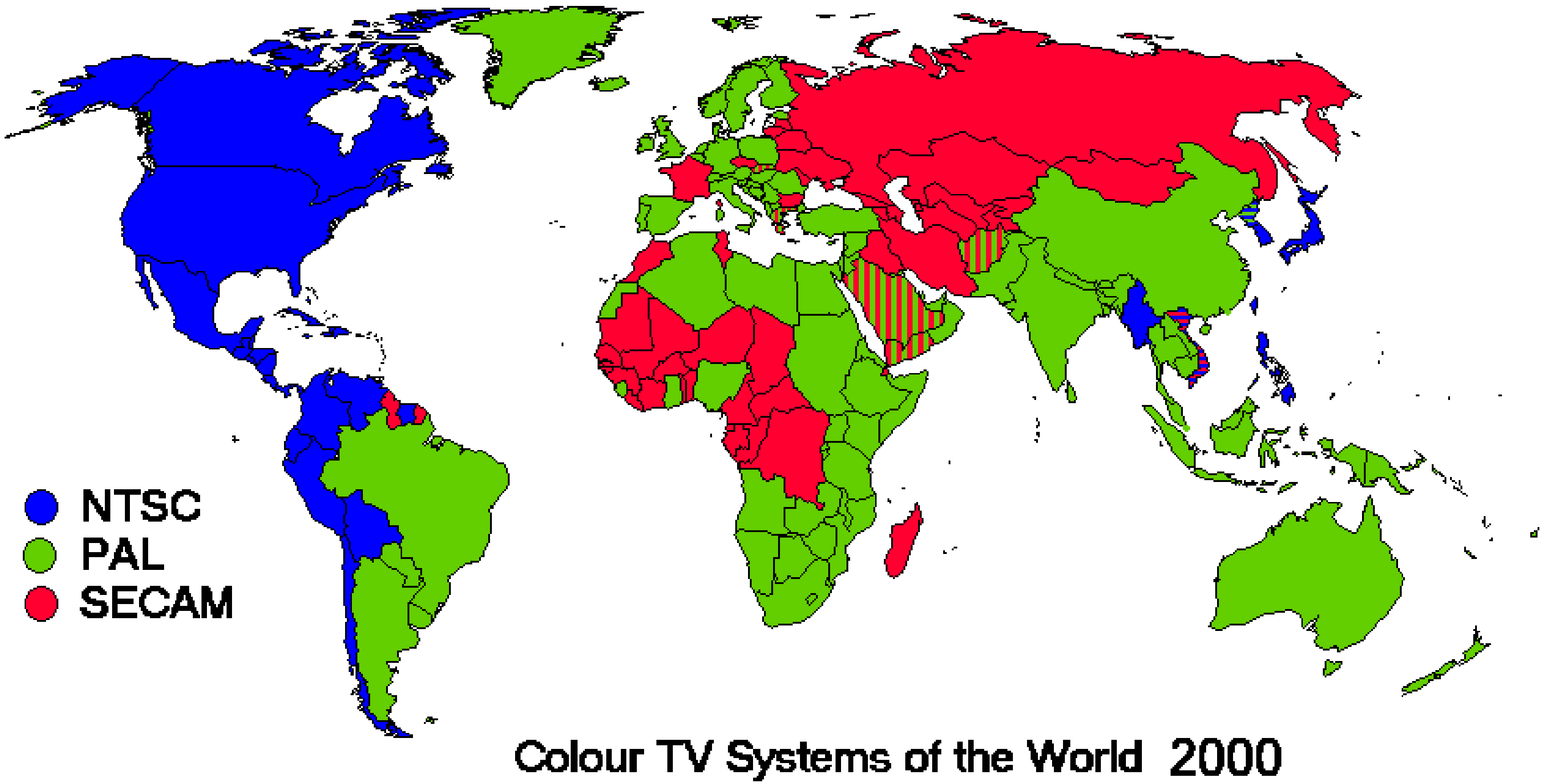
Progressive Scan

- Problems:
  - fields represent different points in time
    => a full picture (a frame) with moving objects shows up with horizontal combs
  - viewing a tv signal on a computer requests for techniques like Bob or Weave
  - the compression of frames consisting of fields has to pay attention to that situation!

On TVs, fields, and frames... (4)

- Common standards for video signals / video signal representation
  - PAL (Phase Alternating Lines)
    - common in many European countries
    - 625 lines
    - 25 full frames per second (50 fields per second)
    - component or composite color
  - NTSC (National Television Systems Committee)
    - common in America and Japan
    - 525 lines
    - 29.97 full frames per second (59.94 fields per second)
    - component or composite color
  - D1, ITU-R B.601-5
    - digital video standard
    - 720x576 for PAL (25 fps) => 10.368 10^6 pixel per second
    - 720x480 for NTSC (30 fps) => 10.368 10^6 pixel per second
    - component color

On TVs, fields, and frames... (5)



5.6.3. Color representation, transformation, and subsampling (1)

- Eye
  - light-sensitive cells of the retina: rods and cones
  - rods
    - approximately 120 million
    - supply brightness (luminance) information
    - very sensitive to light
    - high spatial resolution
  - cones
    - approximately 6,5 million
    - supply color (chrominance) information
    - three types depending on the wavelength (red 600nm, green 540nm, blue 450nm)

→

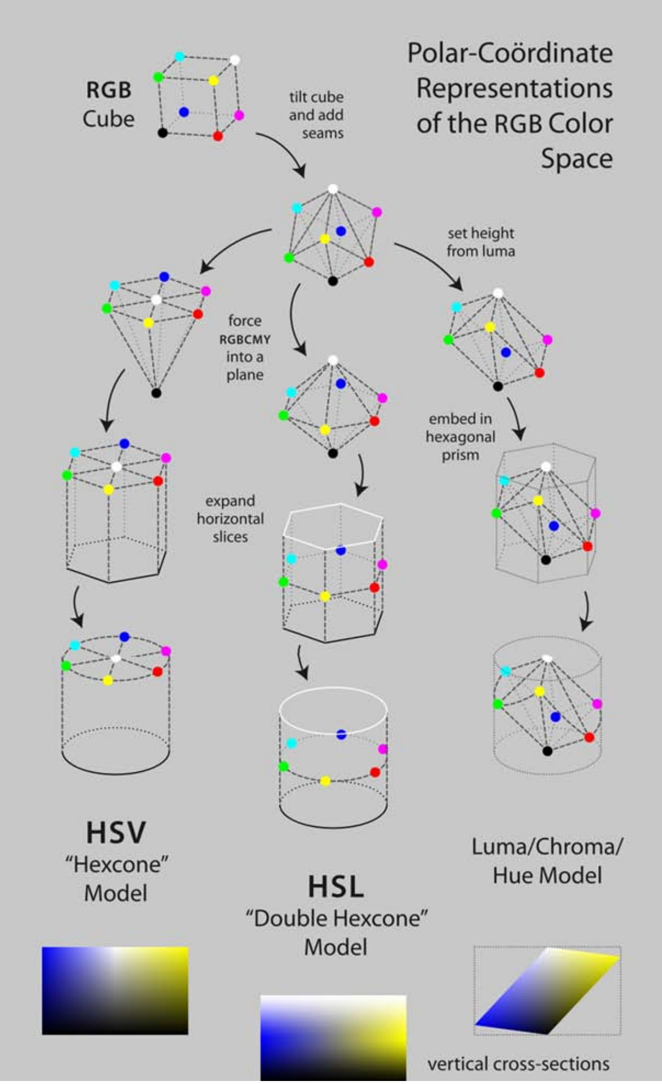
  - Color resolution is lower than brightness resolution
  - The human eye is much more sensitive to changes of the luminance than for changes of the wavelength!
  - An appropriate color representation should pay attention to that.

Color representation, transformation, and subsampling (2)

- Technical development
  - The technical evolution in the area of computer and monitor techniques resulted in the development of various different schemes for color representation (physically and perceptually based color models). Having in mind the three types of cones, most color models deal with three basic colors, e.g.
  - RGB
    - monitor system
    - each pixel consists of three types of phosphor dots
    - three electron guns
    - not perceptually based
  - HLS
    - perceptually based
    - Hue: sensory impression of a pure spectral color (wave length)
    - Saturation: mixture of the color tone with grey
    - Luminance: overall brightness

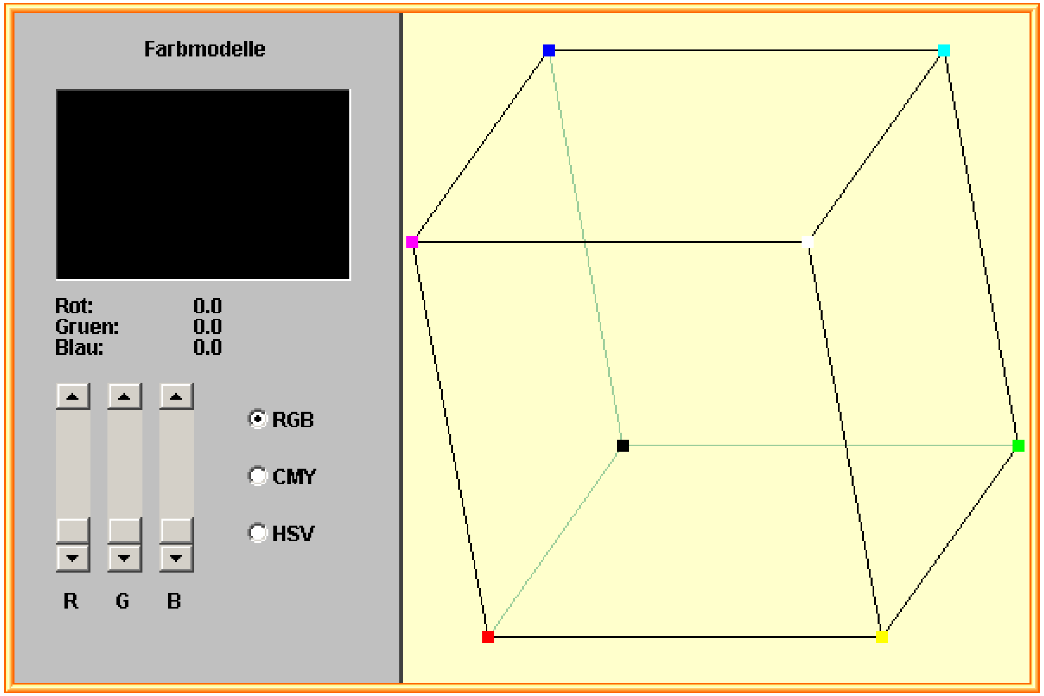
HSL / HSV

- **HSL** and **HSV** are two related representations of points in an RGB color model that attempt to describe perceptual color relationships more accurately than RGB, while remaining computationally simple. *HSL* stands for **h**ue, **s**aturation and **l**ightness, while *HSV* stands for **h**ue, **s**aturation and **v**alue.
- *HSL* is often also called **HLS** or **HSI**, and *HSV* is often also called **HSB**, with *I* standing for *intensity* or *B* for *brightness* as the third coordinate;their definitions are less standardized, and they occasionally refer to some other similar model.



Color Model Applet

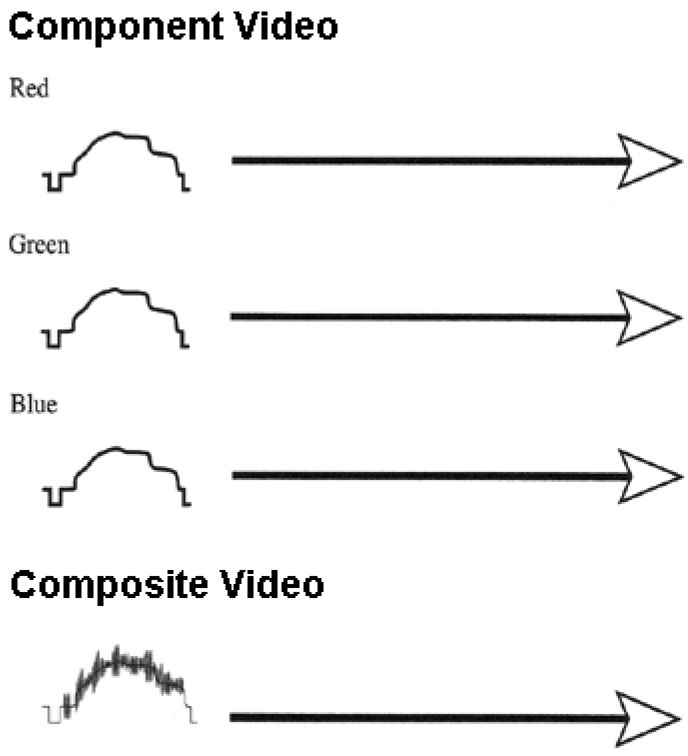
- Link:
  http://haya.informatik.uni-kl.de:11000/data/lectures/014_MMS_WS_2008_2009/050_Compression/0760a_color_2.htm



(c) 1996,1997 Universität Oldenburg

Color representation, transformation, and subsampling (3)

- Technical development (cont.)

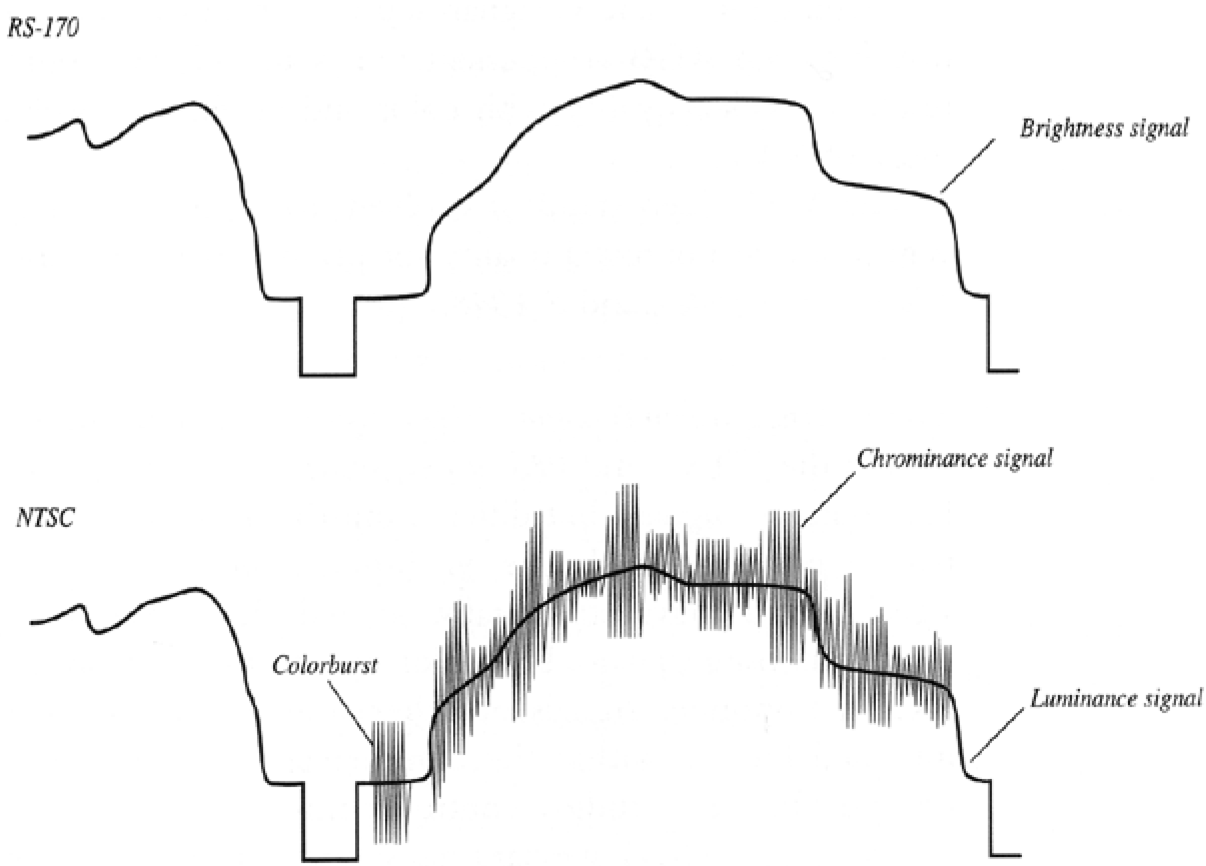


Color representation, transformation, and subsampling (4)

- Technical development (cont.)
  - YUV
    - generated with the invention of the color television to maintain compatibility with the b/w television
    - basic element of the video standards PAL and NTSC
    - "perceptually based"
    - brightness information Y is separated from color difference signals U and V
  - YC_rC_b
    - deduced from and similar to YUV to use with the digital video standard
  - Both YUV and YC_rC_b can be subsampled in their color components to achieve an optimal fitting to perceptional needs.

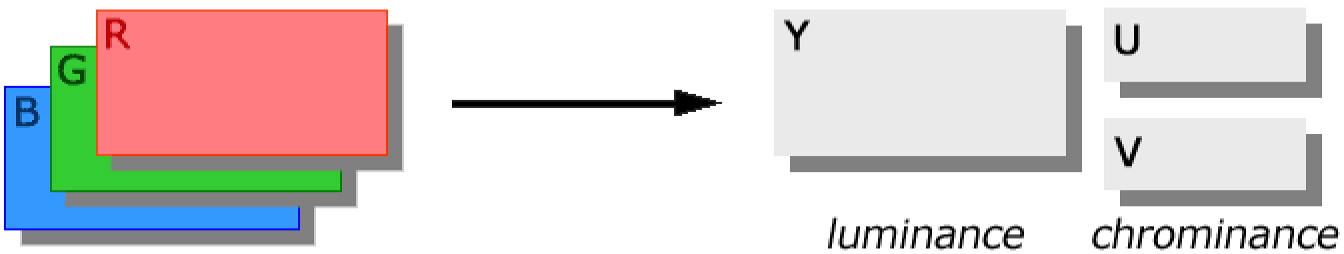
Color representation, transformation, and subsampling (5)

- Technical development (cont.)



Color representation, transformation, and subsampling (6)

- Color space transformation
  - Example: RGB → YUV



  - PAL-Norm color representation:
    Y = 0,30 R + 0,59 G + 0,11 B
    U = 0,493 (B – Y) = -0,10 R - 0,29G + 0,44 B
    V = 0,877 (R – Y) = 0,62 R – 0,52 G – 0,10 B
- In general the color space transformation is lossless
- Since the human perception of color is much less accurate than the perception of luminance, the chrominance data will often be subsampled

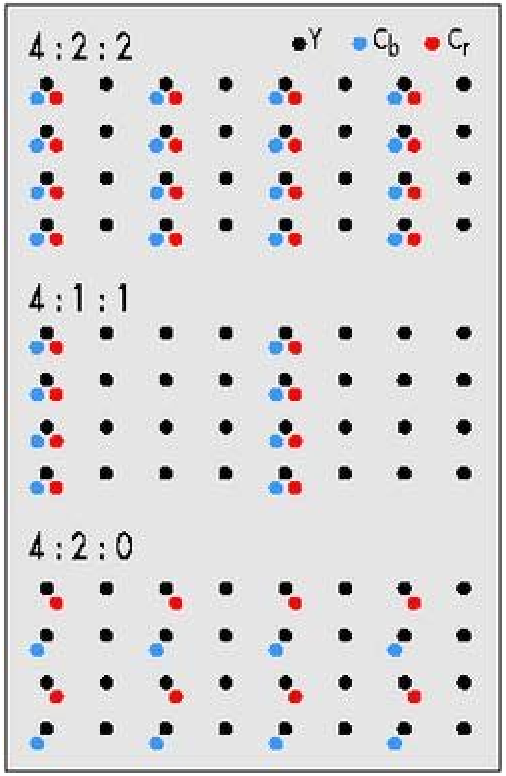
Color representation, transformation, and subsampling (7)

- Color sampling ratio for YUV
  - a:b:c notation for sampling ratio according to ITU-R (former CCIR) recommendation 601-2:
    - Encoding parameters of digital television for studios

4:4:4	no loss of information
4:2:2	2:1 horizontal downsampling, no vertical downsampling (= 4 Y samples for 2 U and 2 V samples per scanline)
4:1:1	4:1 horizontal downsampling, no vertical downsampling (= 4 Y samples for 1 U and 1 V samples per scanline)
4:2:0	2:1 horizontal downsampling, 2:1 vertical downsampling (= 4 Y samples for 2 U or 2 V samples per scanline, U and V are processed on alternating scanlines only)

Color representation, transformation, and subsampling (8)

- Color sampling ratio for YUV (cont.)



5.6.4. Frequency Transforms (1)

History:

- Fourier Series
  - Jean Baptiste Joseph Fourier
  - Representation of periodic continuous functions in terms of a series of sines and cosines.
  - Any periodic function *f*(*t*) with period *T*, i.e.

$$f(t) = f(t + nT) \quad n = \pm 1, \pm 2, \ldots$$

can be represented as

$$f(t) = \sum_{n=-\infty}^{\infty} c_n \cdot e^{jn\omega_0 t} = \sum_{n=-\infty}^{\infty} c_n \cdot (\cos(n\omega_0 t) + j\sin(n\omega_0 t)) \quad j = \sqrt{-1}, \omega_0 = \frac{2\pi}{T}$$

with

$$c_n = \frac{1}{T} \int_0^T f(t) e^{-jn\omega_0 t} dt$$

[Alternative introduction to frequency transformation]

Frequency Transforms (2)

History: (cont.)

- Fourier Series: gain, interpretation $f(t) = \sum_{n=-\infty}^{\infty} c_n \cdot e^{jn\omega_0 t} \quad \omega_0 = \frac{2\pi}{T}$
  - How does a signal change its amplitude over time (space), how does it vary in time (space)? → representation *f*(*t*) (*f*(*x*))
  - The coefficients $\{c_n\}_{n=-\infty}^{\infty}$ are a representation of the signal in the basis $\{e^{jn\omega_0 t}\}_{n=-\infty}^{\infty}$
    - The basis functions differ from each other in how fast they fluctuate in a given time interval,
      e.g. $e^{j2\omega_0 t}$ fluctuates twice as fast as $e^{j\omega_0 t}$
    - Fluctuation is measured in terms of frequency (unit: Hz)
    - The coefficients give a measure of the different amounts of fluctuations present in the signal, it provides a frequency profile of the signal
  - Each representation emphasizes a different aspect of the signal!

Frequency Transforms (3)

History: (cont.)

- Fourier Transform
  - Representation of non periodic continuous functions in terms of a series of sines and cosines by using the periodic extension of the function.
  - Fourier Transform:

$$F(\omega) = \int_{-\infty}^{\infty} f(t) \cdot e^{-j\omega t} dt$$

    How does the signal fluctuate at different frequencies?

  - Inverse Fourier Transform:

$$f(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(\omega) \cdot e^{j\omega t} d\omega$$

    How is the signal composed by components that fluctuate at different frequencies?

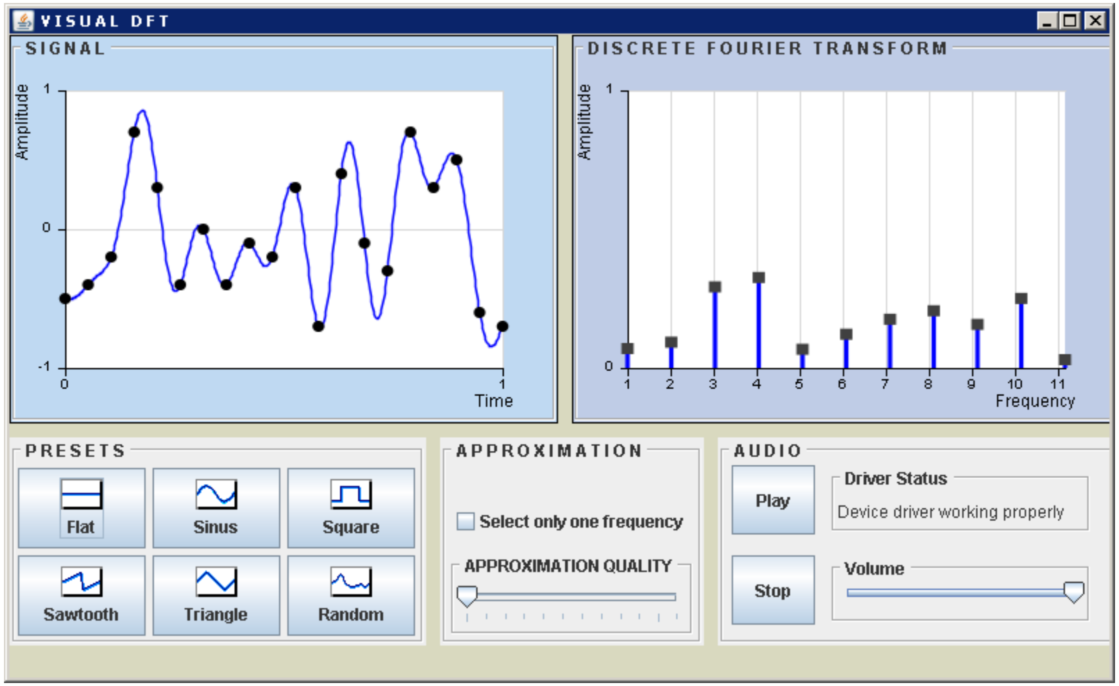
Frequency Transforms (4)

History: (cont.)

- Discrete Fourier Series (DFS)
  - Representation of periodic discrete (sampled) functions representing a periodic sequence.
- Discrete Fourier Transform (DFT)
  - Representation of discrete (sampled) functions representing sequences with finite length.
  - uses DFS (the finite length sequence is one period of a periodic sequence)
  - powerful algorithm "Fast Fourier Transform" (FFT)
  - DFT is usually not applied for compression due to the unhandy imaginary portion in the frequency domain and the fact that Discrete Cosine Transformation (DCT) performs better.

Frequency Transforms (5) - Applet

- Link:
  http://haya.informatik.uni-kl.de:11000/data/lectures/014_MMS_WS_2008_2009/050_Compression/0870_frequency_transforms_5_applet.htm



TECHNISCHE UNIVERSITÄT KAISERSLAUTERN

ICSY Integrated Communication Systems

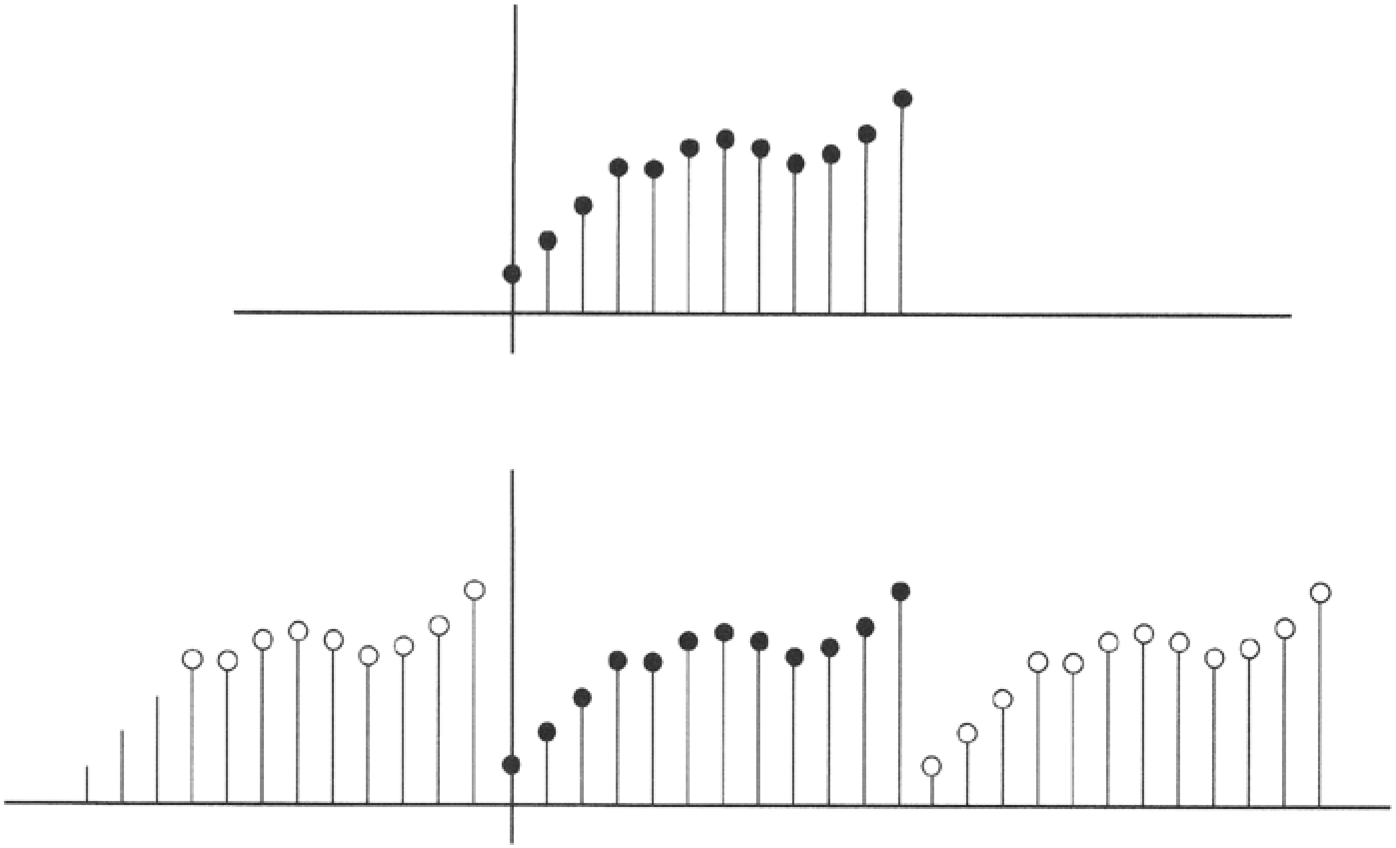
http://www.icsy.de

Frequency Transforms (6)

- Statistical view of the transform process:
  - Consider the changes in statistics between the original and the transformed sequences.
  - Result: It can be shown that we can get the maximum amount of compaction (compact most of the information of a source output sequence into a few elements of the transformed sequence) if we use a transform that decorrelates the input sequence; that is, the sample-to-sample correlation of the transformed sequence is zero.
  - Since an optimal transform in the above sense will be data dependent, it has a small practical area of application => Karhunen-Loève Transformation (KLT).
  - The most popular practical frequency transform to use in compression schemes is the DCT.

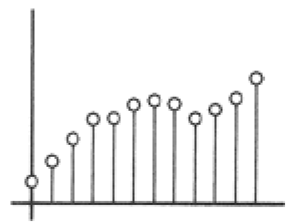
Frequency Transforms (7a)

- **The Discrete Cosine Transformation (DCT) in 1D: (cont.)**
  - Discrete Fourier Transform of a sequence:

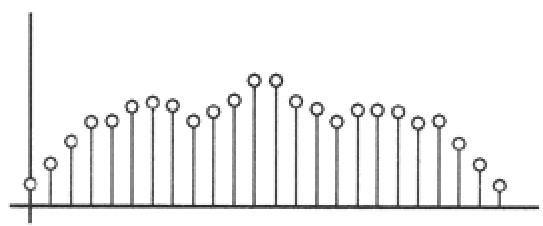


Frequency Transforms (7b)

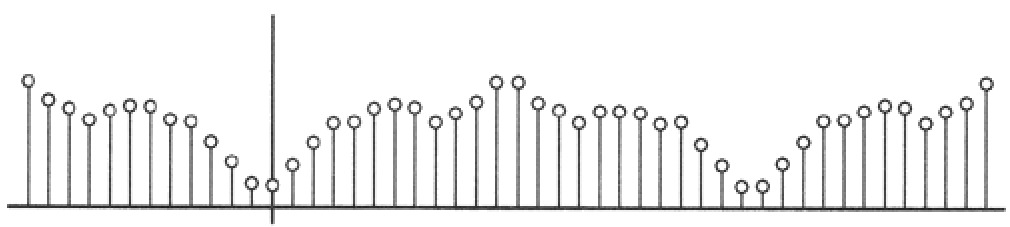
- **The Discrete Cosine Transformation (DCT) in 1D: (cont.)**
  - Discrete Cosine Transform of a sequence:



(a)



(b)



Frequency Transforms (8)

- **Discrete Transforms in 1D:**
  - Forward transform: $\theta_n = \sum_{i=0}^{N-1} x_i \cdot a_{n,i}$

    Compute the sequence $\{\theta_n\}$ from the source sequence $\{x_n\}$.
  - Inverse transform: $x_n = \sum_{i=0}^{N-1} \theta_i \cdot b_{n,i}$

    Recover the sequence $\{x_n\}$ from the transformed sequence $\{\theta_n\}$.
  - Matrix notation: $\theta = \mathbf{A} \cdot x$, $x = \mathbf{B} \cdot \theta$, **A** is the transform matrix
  - Assumption: linear transforms, orthonormal transforms that is **$B=A^{-1}=A^T$**
  - Interpretation: $\begin{pmatrix} x_0 \\ x_1 \end{pmatrix} = \begin{pmatrix} a_{00} & a_{10} \\ a_{01} & a_{11} \end{pmatrix} \cdot \begin{pmatrix} \theta_0 \\ \theta_1 \end{pmatrix} = \theta_0 \cdot \begin{pmatrix} a_{00} \\ a_{01} \end{pmatrix} + \theta_1 \cdot \begin{pmatrix} a_{10} \\ a_{11} \end{pmatrix}$
    - rows of A are the basis vectors of the transform
    - elements of the transformed sequence are the transform coefficients

Frequency Transforms (9)

- **The Discrete Cosine Transformation (DCT) in 1D:**
  - The rows of transformation matrix are obtained as a function of cosines:

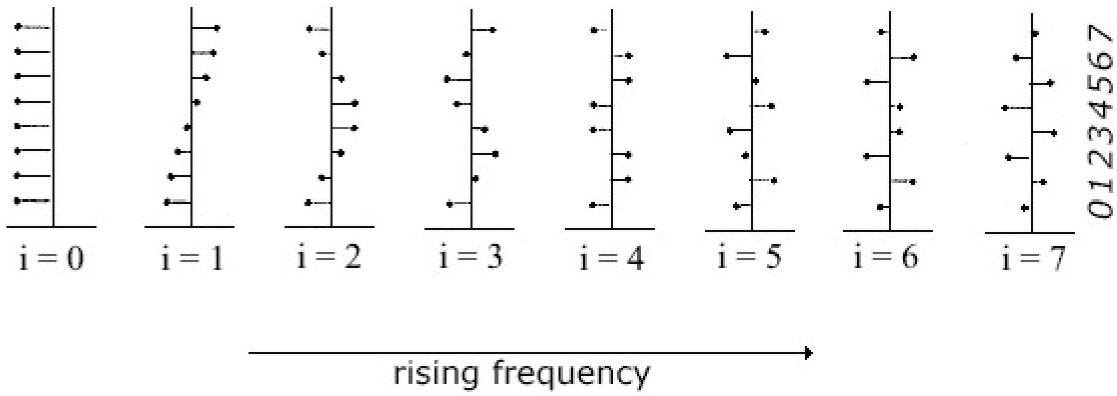
$$(\mathbf{A})_{i,j} = \begin{cases} \sqrt{\frac{1}{N}} \cos \frac{(2j+1)i\pi}{2N} & i = 0,\ j = 0,1,\ldots,N-1 \\ \sqrt{\frac{2}{N}} \cos \frac{(2j+1)i\pi}{2N} & i = 1,2,\ldots,N-1,\ j = 0,1,\ldots,N-1 \end{cases}$$

  - The DCT and the DFT are closely related: The DCT is obtained from the DFT by mirroring the original N point sequence to obtain a 2N-point sequence and then take the first N points of the resulting 2N-point DFT.

[Alternative introduction to DCT]

Frequency Transforms (10)

- **The Discrete Cosine Transformation (DCT) in 1D: (cont.)**
  - The basis vectors for *N*=8:



Frequency Transforms (11)

- **Discrete Transforms in 2D:**
  - General forward transform:

$$\Theta_{k,l} = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} x_{i,j} \cdot a_{i,j,k,l}$$

  - All two-dimensional transforms in use today are separable transforms; that is, transform a two-dimensional block by first taking the transform along one dimension (e.g. the rows) and then repeating the operation along the other direction (transform column-by-column of the intermediately resulting matrix):

$$\Theta_{k,l} = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} x_{i,j} \cdot a_{k,i} \cdot a_{l,j}$$

  - Assumption: linear transforms, orthonormal transforms

Frequency Transforms (12)

- **Discrete Transforms in 2D: (cont.)**
  - Interpretation:
    given *N*x*N* transformation **A**, define:

    $$\boldsymbol{\alpha}_{i,j} = \left(\left(\mathbf{A}\right)_{i\bullet}\right)^T \bullet \left(\left(\mathbf{A}\right)_{j\bullet}\right)$$

    then the inverse transform 2D (e.g. for *N*x*N*=2x2) can be written as

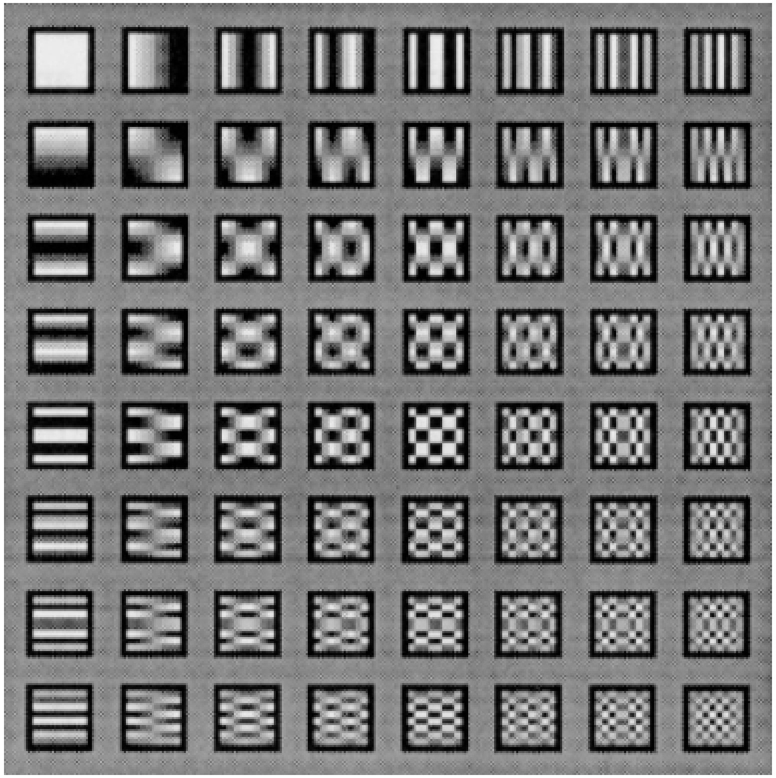
    $$\begin{pmatrix} x_{00} & x_{01} \\ x_{10} & x_{11} \end{pmatrix} = \Theta_{00} \cdot \boldsymbol{\alpha}_{0,0} + \Theta_{01} \cdot \boldsymbol{\alpha}_{0,1} + \Theta_{10} \cdot \boldsymbol{\alpha}_{1,0} + \Theta_{11} \cdot \boldsymbol{\alpha}_{1,1}$$

  - $\boldsymbol{\alpha}_{i,j}$ are the basis vectors of the transform
  - elements of the transformed sequence are the transform coefficients

  - the coefficient corresponding to $\alpha_{0,0}$ is called the DC coefficient
  - all other coefficients are called AC coefficients
  - all elements of $\alpha_{0,0}$ are the same, hence the DC designation

Frequency Transforms (13)

- **The Discrete Cosine Transformation (DCT) in 2D:**
  - The basis vectors for *N*x*N*=8x8:



Frequency Transforms (14)

- **The Discrete Cosine Transformation (DCT) - Summary:**
  - **representation of the information as a weighted superposition of increasing frequencies (describing finer details) - this corresponds to human perception**
  - assumption: existing intense correlation (time/space) of the source output (i.e. small changes between neighboring samples)
    - blocking operation supports exploitation of the correlation
    - applicable for natural photos, less/not applicable for vector graphics, two colored pictures, synthetic pictures
  - weights of high frequency basis functions will be small or nearly zero → information can be represented with less data in frequency space
  - weights/coefficients of high frequency basis functions describe fine details → can be quantized at a coarser level
  - The result of the DCT is typically quantized: Coefficients are divided by defined divisors to reduce precision. Typically small divisors are used for low frequency coefficients, higher divisors for high frequency coefficients. The choice of divisors defines the amount of loss of information.

Frequency Transforms (15)

- **The Discrete Cosine Transformation (DCT) - 2D-Quantization:**
  - The original image representation.
  - Result of the DCT: nearly all information of the picture is represented by low frequency; the rest is nearly null.
  - Result of the quantization: the precision of the values has been reduced; most values are exactly null now.
  - Result of the decompression process.

Frequency Transforms (16)

Wavelet Transformation/Decomposition - Motivation (using Haar wavelets):

- Example: one-dimensional image, resolution 4 pixels, pixel values [9 7 3 5]
  1. average pixels pairwise together to get a new lower-resolution (resolution 2) image with pixel values [8 4]
  2. store detail coefficients to capture the missing information [1 -1]
  3. repeat process on lower resolution image: image with resolution 1, pixel values [6] and detail coefficients [2]
- result:

resolution	averages	detail coefficients
4	[9 7 3 5]	
2	[8 4]	[1 -1]
1	[6]	[2]

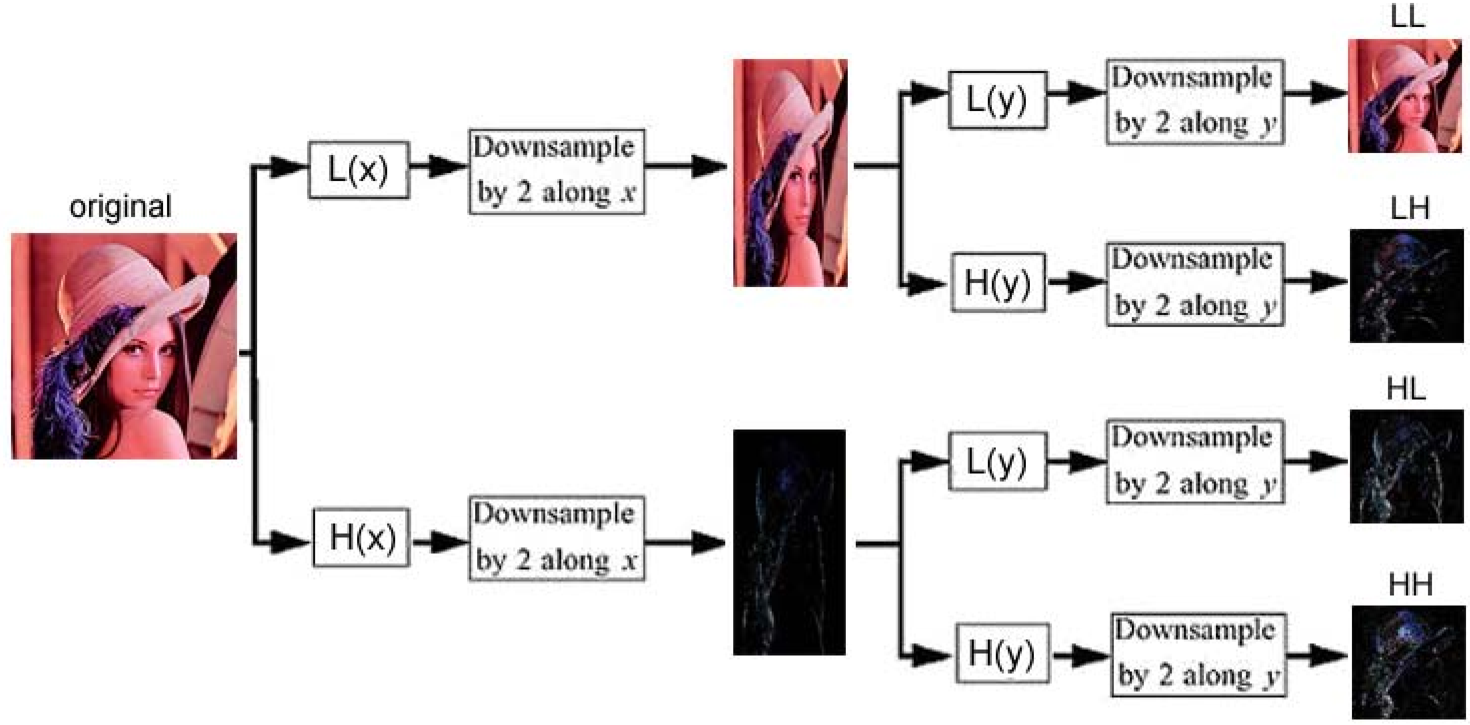
  – resolution averages detail coefficients 4 [9 7 3 5] 2 [8 4] [1 -1] 1 [6] [2]
- interpretation:
  – wavelet transform/decomposition := [6 2 1 -1]
  (overall average of the original followed by details in order of increasing resolution)
  – new image by averaging (low-pass filtering) and downsampling
  – detail image (coefficients) by high-pass filtering and downsampling

Frequency Transforms (17)

- **Wavelet Transformation/Decomposition:**
  - General decomposition procedure (2D): choose an appropriate set of wavelet basis functions defining a special filter pair (low-pass and high-pass) which serves for:
    - low- and high-pass filter information together enable lossless reconstruction
    - the resulting filtered sequences can be subsampled by 2 with no loss of information
  - for every row: process low- and high-pass filter and subsample by 2
  - for every (altered) column: process low- and high-pass filter and subsample by 2
  - repeat previous step on double low-pass filtered "subimage"
  - multi-scale/multi-resolution analysis

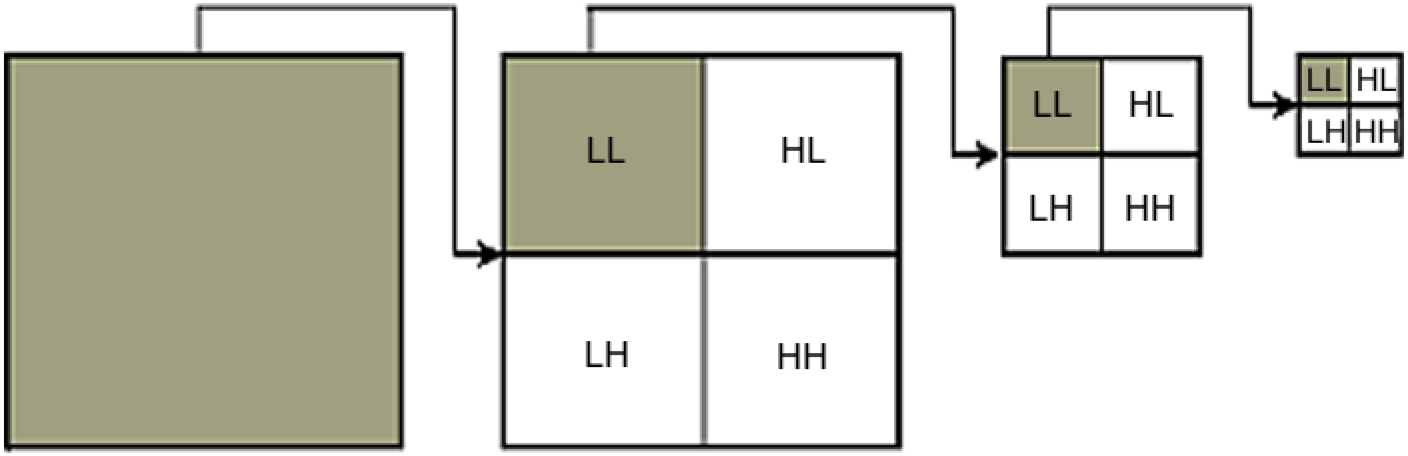
Frequency Transforms (18)

- **Wavelet Transformation/Decomposition: (cont.)**
  - General decomposition procedure (2D, one decomposition step):
    - L(x): low-pass filter on row x, L(y): low-pass filter on column y
    - H(x): high-pass filter on row x, H(y): high-pass filter on column y



Frequency Transforms (19)

- **Wavelet Transformation/Decomposition: (cont.)**
  - General decomposition procedure (2D): recursive decomposition steps



Frequency Transforms (20)

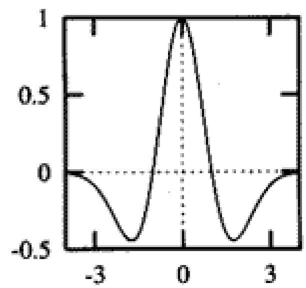
- **Wavelet Transformation/Decomposition: (cont.)**
  - General decomposition procedure (2D): recursive decomposition steps

Frequency Transforms (21)

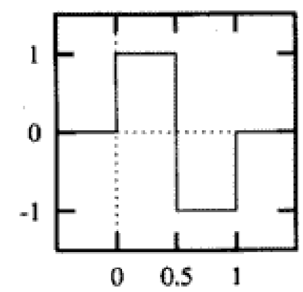
- **Wavelet Transformation/Decomposition: (cont.)**
  - General composition procedure (2D), reconstruction:
    - starting point is the smallest LL-version with difference information on the same level
    - add one zero row between every two rows in all four subimages
    - process mirrored low- and high-pass filter to columns of new subimages
    - add LL' and LH', add HL' and HH'
    - add one zero column between every two columns in the two intermediate images
    - process mirrored low- and high-pass filter to rows of new subimages
    - add the two images => result is LL image on the next level
    - process procedure until original or desired resolution is reached

Frequency Transforms (22)

- **Wavelet Transformation/Decomposition: (cont.)**
  - wavelets should be simple
  - *w(x) ≠ 0* only for x "near" 0
  - Wavelet examples:



Mexican Hat Wavelet



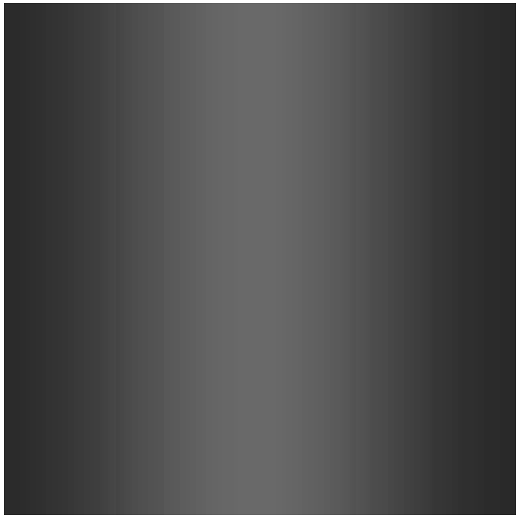
Haar Wavelet

$$w(x) = (1 - x^2)e^{-x^2/2}$$

$$w(x) = \begin{cases} 1 \text{ if } 0 \leq x < 0.5 \\ -1 \text{ if } 0.5 \leq x < 1 \\ 0 \text{ otherwise} \end{cases}$$

Frequency Transforms (23)

- **Wavelet Transformation/Decomposition: (cont.)**
  - Wavelet compression Example + Info:
    - Decompression video example (original from University of Bristol, UK)



    - More information: Amaras Wavelet page
      http://www.amara.com/current/wavelet.html

Frequency Transforms (24)

- **Wavelet Transformation versus Discrete Cosine Transformation:**
  - DCT is an operation that works on blocks, wavelet decomposition works on entire domain (e.g. picture), it is a global operation
  - DFT (DCT) contains only cosine and sinus waves, wavelet transformation supports many different wave functions (once they are chosen, the transformation is a simple repeated application of low- and high-pass filter operations)
  - wavelet decomposition delivers spatial frequency information, DCT coefficients affect entire domain
  - wavelet decomposition directly supports pictures at several resolution levels
  - the quantization step is similar for both transformations