



Multimedia Systems

WS 2009/2010

Exercise Course, February 2, 2010

Dipl.-Inf. Simon Schwantzer

University of Kaiserslautern, Germany
Integrated Communication Systems Lab

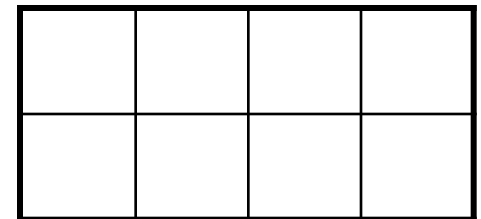
Email: schwantzer@informatik.uni-kl.de

Color/Chroma Subsampling

- a:b:c-Notation

- a

- number of columns as reference value
 - usually four, scarce 3

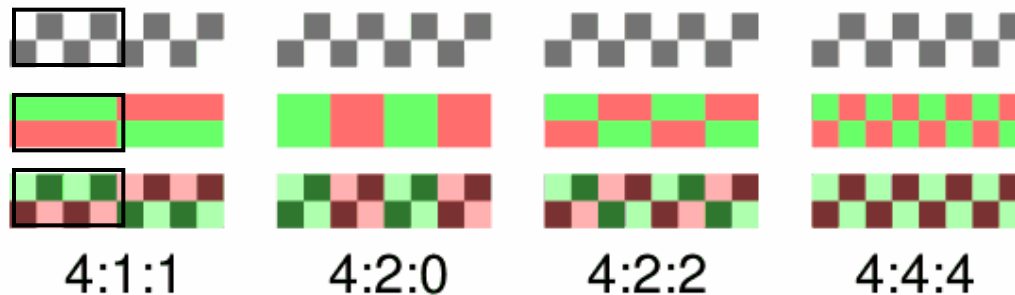


- b

- number of C_R/C_B elements in odd rows

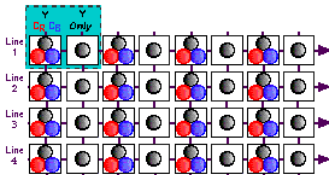
- c

- number of C_R/C_B elements in even rows



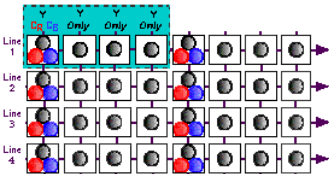
Color/Chroma Subsampling

- Examples



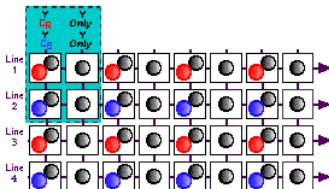
4:2:2

sampling is used in ITU-R BT601, D-1, D-5, Ampex DCT, Digital Betacam, Digital-S, and DVCPRO50 formats.



4:1:1

sampling is used in 525/59.94 ("NTSC") DV and DVCAM, and in both 525/59.94 and 625/50 ("PAL") DVCPRO



4:2:0

co-sited 4:2:0 sampling is used in 625/50 ("PAL") DV and DVCAM formats, JPEG



Fourier Series

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.

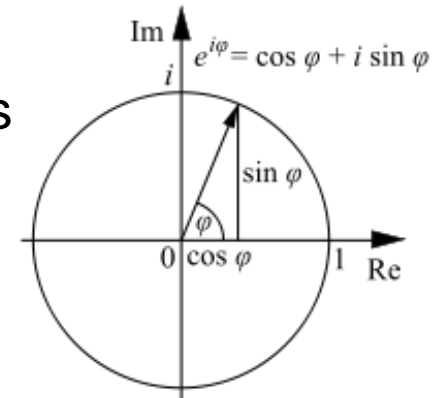
$$f(t) = f(t + nT) \quad n = \pm 1, \pm 2, \dots$$

can be represented as

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

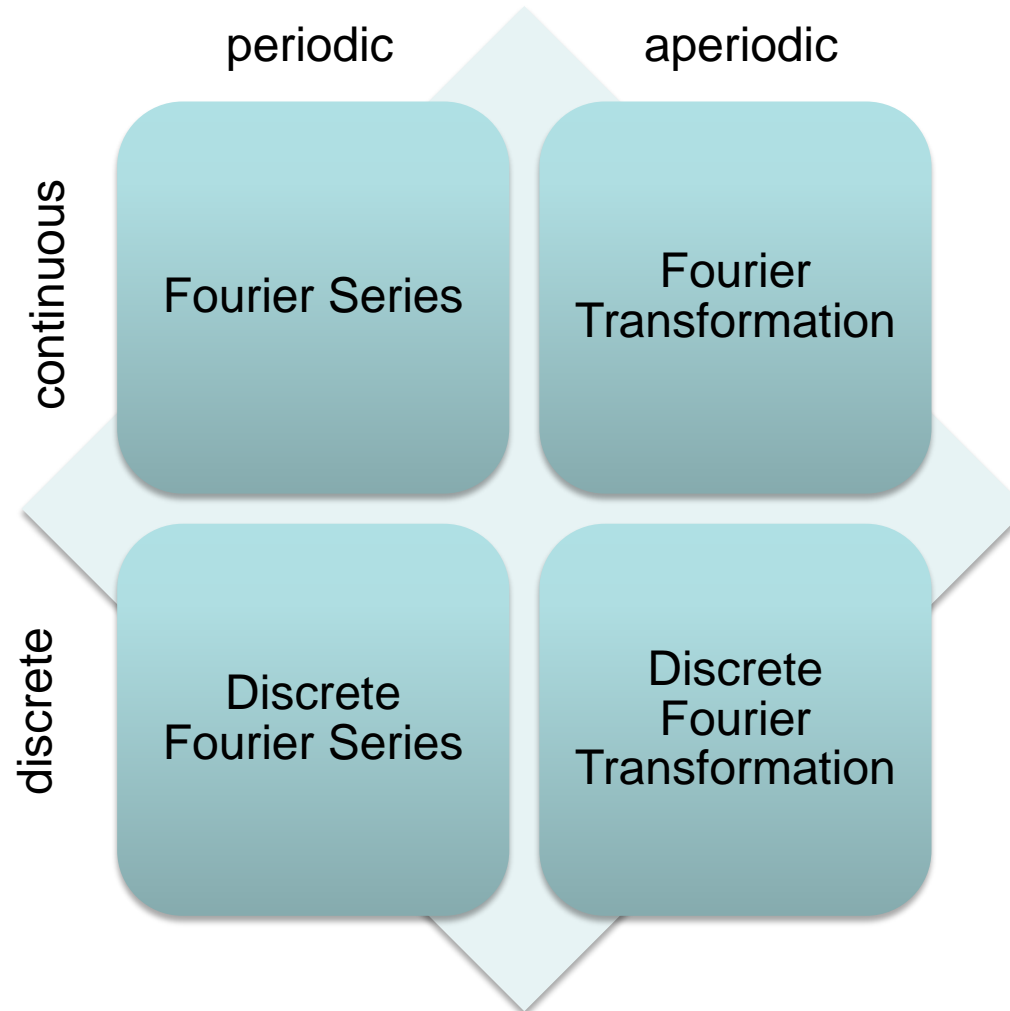
with

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





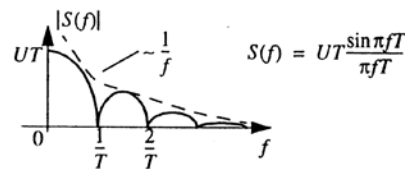
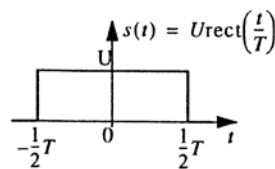
FS – FT – DFS – DFT



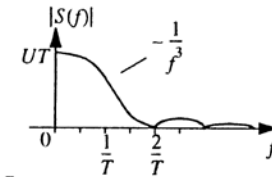
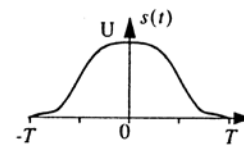
Fourier Transformation

Examples:

a) Rechteck-Impuls

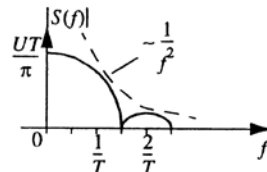
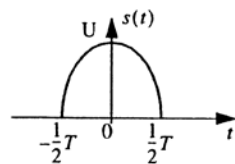


c) Kosinusquadrat-Impuls



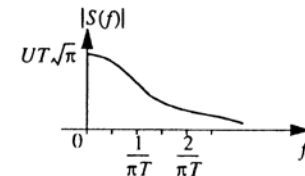
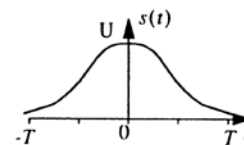
$$S(f) = U \int_{-T}^T \left(\cos\left(\frac{\pi t}{T}\right) \right)^2 e^{-j2\pi f t} dt = \dots = \frac{UT}{1 - (2fT)^2} \frac{\sin 2\pi f T}{2\pi f T}$$

b) Kosinus-Impuls



$$S(f) = U \int_{-T/2}^{T/2} \cos\left(\frac{\pi t}{T}\right) e^{-j2\pi f t} dt = \dots = \frac{UT}{\pi} \frac{\cos \pi f T}{1 - (2fT)^2}$$

d) Gauß-Impuls

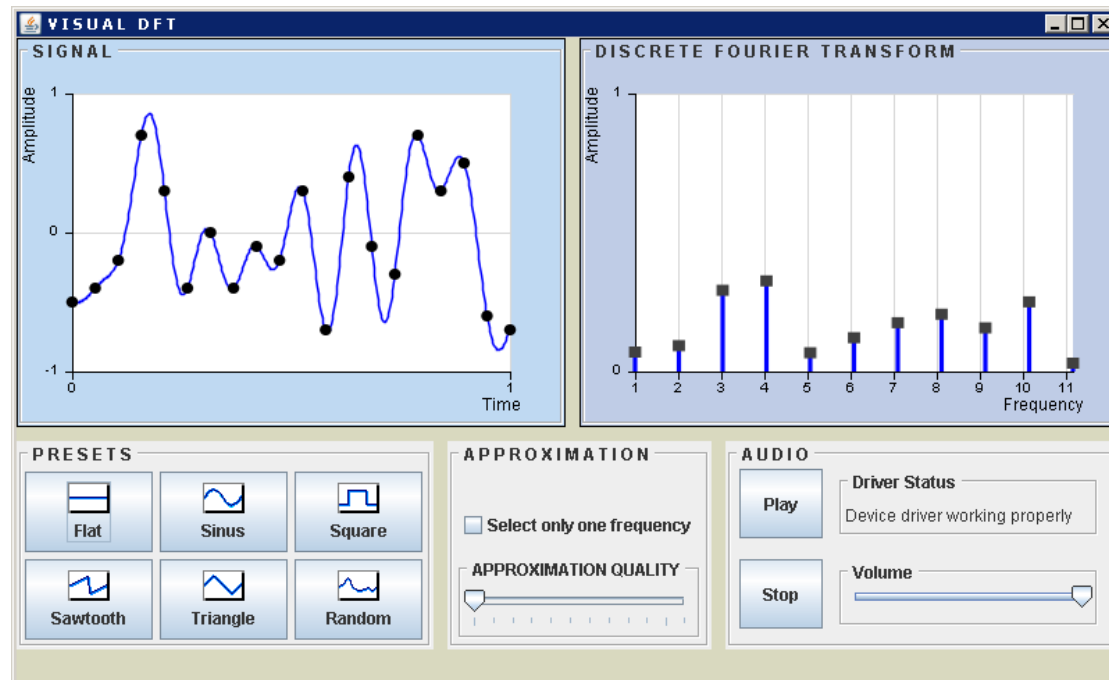


$$s(t) = U e^{-(t/T)^2} \rightarrow S(f) = UT \sqrt{\pi} e^{-(\pi f T)^2}$$

Frequency Transforms (5) - Applet

- Link:

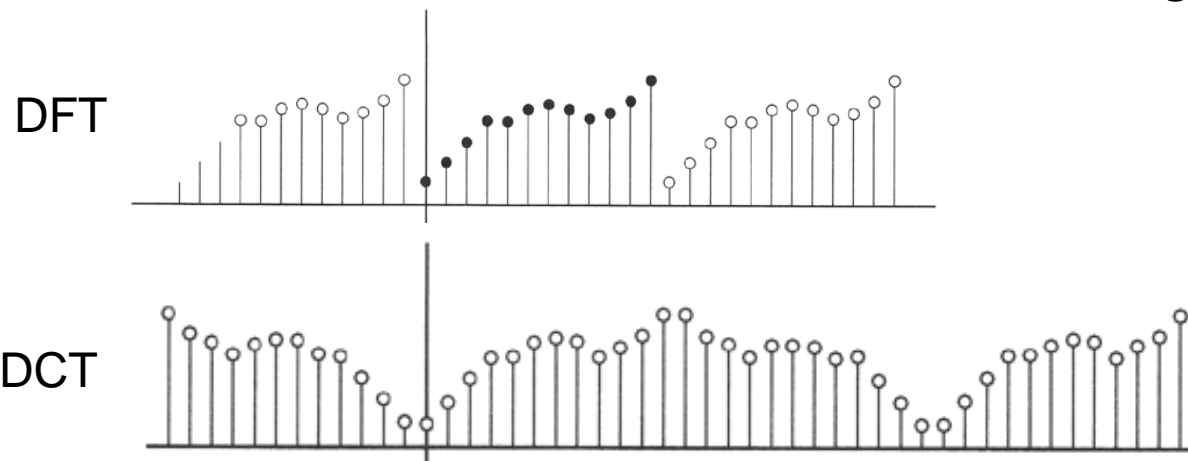
http://haya.informatik.uni-kl.de:11000/data/lectures/014_MMS_WS_2008_2009/050_Compression/0870_frequency_transforms_5_applet.htm





DFT - DCT

- DCT calculation does include handling of complex numbers
- In DCT the periodicity is realized with mirroring



- DCT with DFT
 - mirroring the original N point sequence to $2N$ -point sequence
 - take the first N points of the resulting $2N$ -point DFT

Image Compression - JPEG

- 1) select color space
- 2) color-subsampling
- 3) DCT
- 4) quantization
- 5) coding

- color space
 - greyscale
 - RGB
 - YC_bC_r

$$\begin{pmatrix} Y \\ C_b \\ C_r \end{pmatrix} = \begin{pmatrix} 0.299 & 0.587 & 0.114 \\ -0.1687 & -0.3313 & 0.5 \\ 0.5 & -0.4187 & -0.0813 \end{pmatrix} * \begin{pmatrix} R \\ G \\ B \end{pmatrix}$$

$$\begin{pmatrix} R \\ G \\ B \end{pmatrix} = \begin{pmatrix} 1.0 & 0.0 & 1.402 \\ 1.0 & -0.34414 & -0.71414 \\ 1.0 & 1.772 & 0.0 \end{pmatrix} * \begin{pmatrix} Y \\ C_b \\ C_r \end{pmatrix}$$

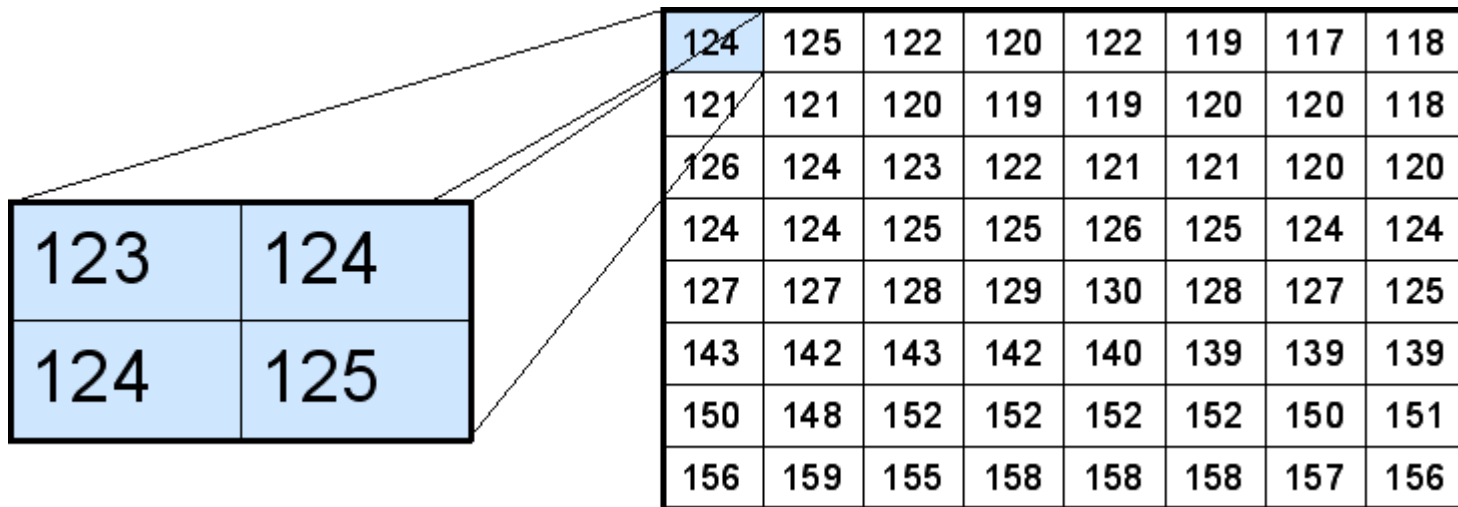
- The following example shows an 8x8 pixel image section
 - 8-bit numbers represent greyscales

124	125	122	120	122	119	117	118
121	121	120	119	119	120	120	118
126	124	123	122	121	121	120	120
124	124	125	125	126	125	124	124
127	127	128	129	130	128	127	125
143	142	143	142	140	139	139	139
150	148	152	152	152	152	150	151
156	159	155	158	158	158	157	156

Image Compression - JPEG

- 1) select color space
- 2) color-subsampling
- 3) DCT
- 4) quantization
- 5) coding

- Assuming a given 16x16 block with C_b or C_r
- The eye can sense local differences only poorly
- Averaged values of 2x2 sections



123	124
124	125

124	125	122	120	122	119	117	118
121	121	120	119	119	120	120	118
126	124	123	122	121	121	120	120
124	124	125	125	126	125	124	124
127	127	128	129	130	128	127	125
143	142	143	142	140	139	139	139
150	148	152	152	152	152	150	151
156	159	155	158	158	158	157	156

Image Compression - JPEG

- 1) select color space
- 2) color-subsampling
- 3) DCT
- 4) quantization
- 5) coding

- 8x8 block with 8-bit values
- thus $P=8$; left shift of all items corresponds to -128
- this is the source matrix X

- DCT
 - 1) normalize values
 - 2) enforce $N \times N$ blocks
 - 3) apply 2-D DCT

-4	-3	-6	-8	-6	-9	-11	-10
-7	-7	-8	-9	-9	-8	-8	-10
-2	-4	-5	-6	-7	-7	-8	-8
-4	-4	-3	-3	-2	-3	-4	-4
-1	-1	0	1	2	0	-1	-3
15	14	15	14	12	11	11	11
22	20	24	24	24	24	22	23
28	31	27	30	30	30	29	28

Image Compression - JPEG

- 1) select color space
- 2) color-subsampling
- 3) DCT
- 4) quantization
- 5) coding

- 8x8 block contains only values between -128 and 127
- Application of the 2-D DCT leads to the transformed matrix $D = A \cdot X \cdot A^T$

- DCT
 - 1) normalize values
 - 2) enforce NxN blocks
 - 3) apply 2-D DCT

39,88	6,56	-2,24	1,22	-0,37	-1,08	0,79	1,13
-102,4	4,56	2,26	1,12	0,35	-0,63	-1,05	-0,48
37,77	1,31	1,77	0,25	-1,50	-2,21	-0,10	0,23
-5,67	2,24	-1,32	-0,81	1,41	0,22	-0,13	0,17
-3,37	-0,74	-1,75	0,77	-0,62	-2,65	-0,13	0,76
5,98	-0,13	-0,45	-0,77	1,99	-0,26	1,46	0,00
3,97	5,52	2,39	-0,55	-0,05	-0,84	-0,52	-0,13
-3,43	0,51	-1,07	0,87	0,96	0,09	0,33	0,01

Image Compression - JPEG

- 1) select color space
- 2) color-subsampling
- 3) DCT
- 4) quantization
- 5) coding

- Quantization

- select table Q
- quantify DCT-coefficients

$$l_{ij} = \left\lfloor \frac{\varphi_{ij}}{q_{ij}} + 0.5 \right\rfloor$$

- store results in L

- select a quantization table Q with items q_{ij} according to JPEG recommendation
- quantify according to stated formula and obtain table L with quantified DCT-coefficients

2	1	0	0	0	0	0	0
-9	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Image Compression - JPEG

- 1) select color space
- 2) color-subsampling
- 3) DCT
- 4) quantization
- 5) coding

- Coding

- DC-coefficient
- AC-coefficients

0	0					
1	-1	1				
2	-3	-2	2	3		
3	-7	...	-4	4	...	7
4	-15	...	-8	8	...	15

- DC-coefficient

		2	1	...
		-9	0	...
-1	3			
-7	0
...		

- coding table: 16 categories with increasing number of entries
- code difference to preceding DC-coefficient:
 - here: 3
 - 3 is in group 2 and is on position 4 there (thus bitword 11)
 - assuming Huffman-Code for group 2 is 2-bit long (e.g. 01)
 - resulting sequence for DC: 0111

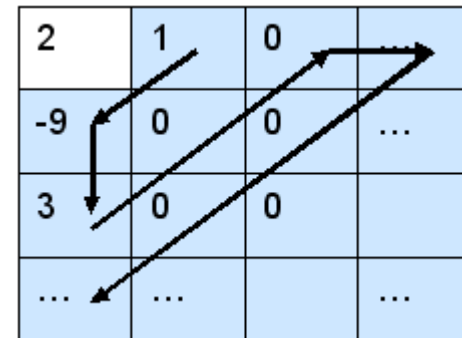
Image Compression - JPEG

- 1) select color space
- 2) color-subsampling
- 3) DCT
- 4) quantization
- 5) coding

- Coding
 - DC-coefficient
 - AC-coefficients

(Z,C)	Code	(Z,C)	Code
(0,0)EOB	1010	(1,1)	1100
(0,1)	00	(1,2)	11011
(0,2)	01
(0,3)	100	(16,0)ZRL	1111111001
...

- AC-coefficients are traversed in zick-zack and coded absolutely



- The (Z,C) table contains Huffman codes for AC-sequences
 - Z: number of preceding zeros
 - C: category of the AC coefficient
 - For the example the results are:
 - 1 -> (0,1) -> 00 1
(2nd position in 1st category)
 - -9 -> (0,4) -> 1011 0110
 - 3 -> (0,2) -> 01 11
 - 0...0 -> (0,0) -> 1010

Image Compression - JPEG

- Result

- 8x8 section coded with 23 bit
0111 0011 0110 1100 1111 010
- i.e. in average 23/64 instead of 8 bit per pixel
- the reconstructed block has only small errors [values in %]

0,8	2,4	0,0	0,8	1,6	0,8	1,7	0,8
0,0	0,0	0,8	0,8	0,0	1,7	1,7	0,0
4,0	2,4	2,4	2,5	1,7	2,5	2,5	2,5
0,0	0,0	1,6	2,4	3,2	3,2	3,2	3,2
2,4	2,4	0,8	0,0	1,5	0,0	0,8	1,6
1,4	0,7	2,1	1,4	0,7	0,7	0,7	1,4
1,3	2,7	0,7	0,7	1,3	2,0	0,7	2,0
1,9	0,0	1,9	0,6	0,6	1,3	1,3	0,6