ALBERT-LUDWIGS-UNIVERSITÄT FREIBURG
INSTITUT FÜR INFORMATIK
Lehrstuhl für Mustererkennung und Bildverarbeitung
Prof. Dr.-Ing. Hans Burkhardt

Am Flughafen 17
Geb. 052, Zi 01-029
D-79085 Freiburg
Tel. 0761 - 203 - 8260

Literatur zur Vorlesung
Algorithmen zur Digitalen Bildverarbeitung

Literatur

[1] W.K. Pratt. *Digital Image Processing.* Wiley Interscience, 2. edition, 1991.

[2] Rafael C. Gonzalez und Richard E. Woods. *Digital Image Processing.* Addison-Wesley, 1993.

[3] F.M. Wahl. *Digitale Bildsignalverarbeitung—Berichtigter Nachdruck.* Springer Verlag, 1989.

[4] B.K.P. Horn. *Robot Vision.* Mc Graw-Hill, 1987.

[5] H. Heuser und H. Wolf. *Algebra, Funktionalanalysis und Codierung.* Teubner Verlag, 1986.

[6] L.A. Zadeh und C.A. Desoer. *Linear System Theory.* Mc Graw-Hill, 1963.

[7] N.R. Bracewell. *The Fourier Transform and its Applications.* Mc Graw-Hill, 1986.

[8] A. Papoulis. *Signal Analysis.* Mc Graw-Hill, 1984.

[9] A. Papoulis. *Probability, Random Variables, and Stochastic Processes.* Mc Graw-Hill, 1984.

[10] M. Schwartz. *Information Transmission, Modulation, and Noise.* Mc Graw-Hill, 1981.

[11] Anil K. Jain. *Fundamentals of Digital Image Processing.* Prentice-Hall, Inc., 1989.

[12] B. Jähne. *Digitale Bildverarbeitung.* Springer-Verlag, 4. edition, 1997.

Legendre - Polynome

Allgemeine Form:

$$L_n(t) = \frac{1}{2^n n!} \frac{d^n}{dt^n}((t^2-1)^n)$$

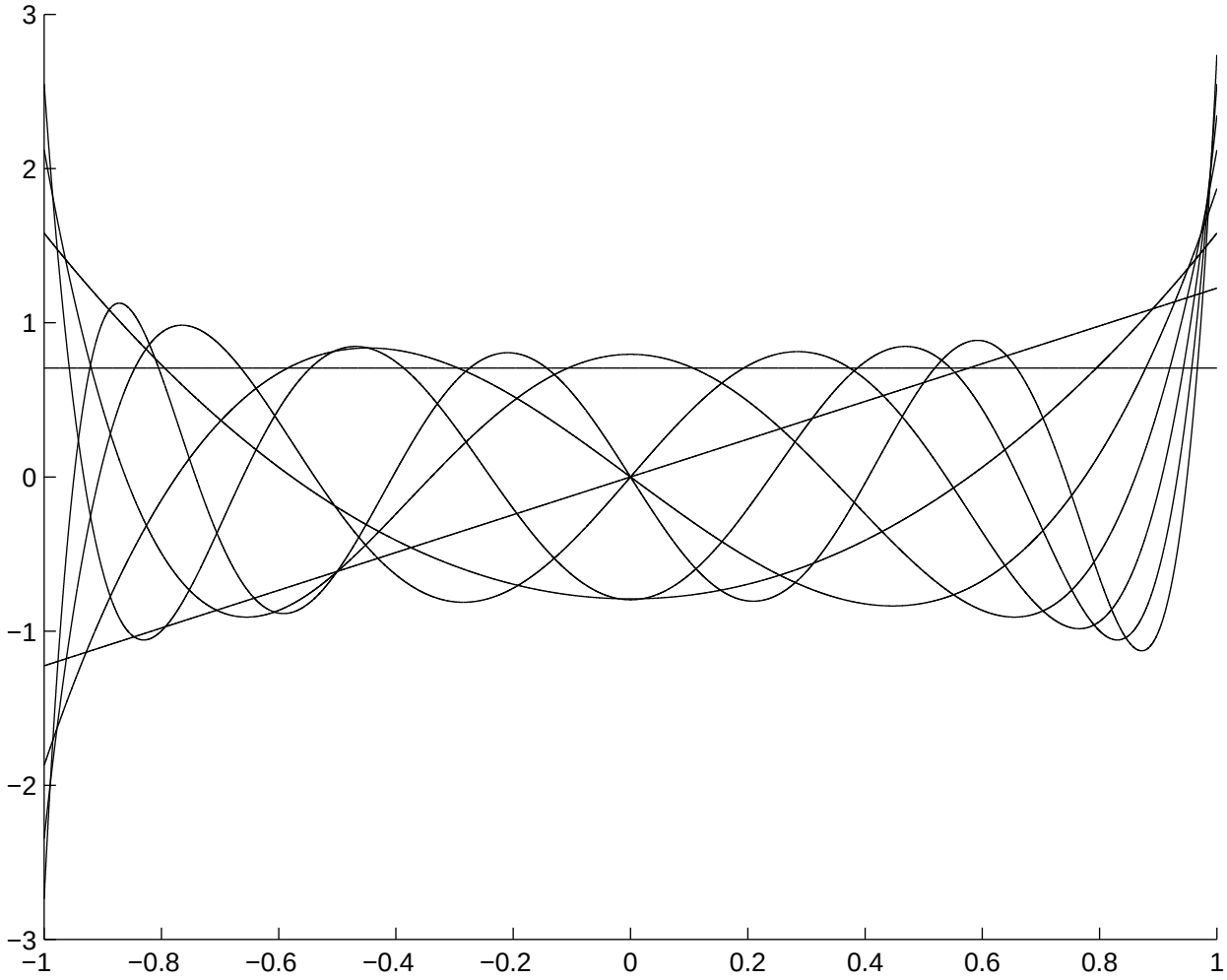
normiert:

$$l_n(t) = \sqrt{\frac{2n+1}{2}} L_n(t)$$

Explizit für $n = 0, \ldots, 2$:

$$l_0(t) = \sqrt{\tfrac{1}{2}}, \quad l_1(t) = \sqrt{\tfrac{3}{2}}t, \quad l_2(t) = \sqrt{\tfrac{5}{8}}(3t^2-1), \ \ldots$$

Funktionsgraphen von $l_0, \ldots, l_6$:



Verallgemeinerte Fourierreihe mit Polynomen

Bestapproximation der Funktion $x(t) = \sin(\frac{\pi}{2}(t+1))$ durch die Fourierreihe

$$y(t) = \sum_{i=0}^{2} \alpha_i \cdot l_i(t),$$

mit

$$\|x(t) - y(t)\|^2 = \|x(t) - \sum_i \alpha_i l_i(t)\|^2 \qquad \text{minimal}$$

wobei $l_0, \ldots, l_2$ die ersten drei Legendre-Polynome sind:

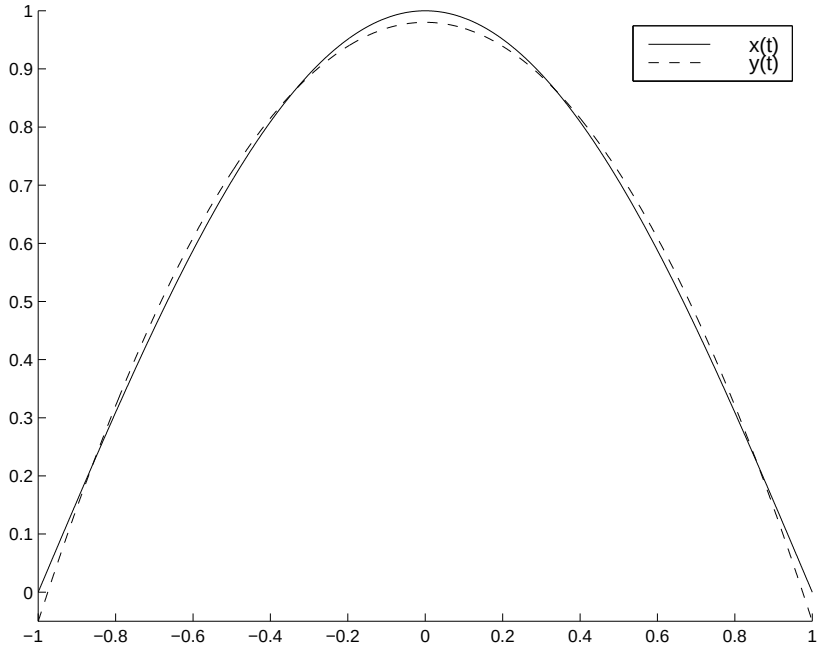
$$l_0(t) = \sqrt{\frac{1}{2}}, \quad l_1(t) = \sqrt{\frac{3}{2}}t, \quad l_2(t) = \sqrt{\frac{5}{8}}(3t^2 - 1)$$

Berechnung der Fourier-Koeffizienten:

$$\begin{aligned}
\alpha_0 &=< l_0(t), x(t) > &= \frac{1}{\sqrt{2}} \int_{-1}^{+1} \sin\left(\frac{\pi}{2}(t+1)\right) dt = \frac{1}{\sqrt{2}}\frac{4}{\pi} \\
\alpha_1 &=< l_1(t), x(t) > &= 0 \\
\alpha_2 &=< l_2(t), x(t) > &= \sqrt{\frac{5}{8}}\frac{8}{\pi}(1 - \frac{12}{\pi^2})
\end{aligned}$$

Fourierreihe:

$$y(t) \;=\; c_0\sqrt{\frac{1}{2}} + c_1\sqrt{\frac{3}{2}}t + c_2\sqrt{\frac{5}{8}}(3t^2 - 1) = \frac{2}{\pi} + \frac{5}{\pi}(1 - \frac{12}{\pi^2})(3t^2 - 1)$$



Pseudo-Inverse

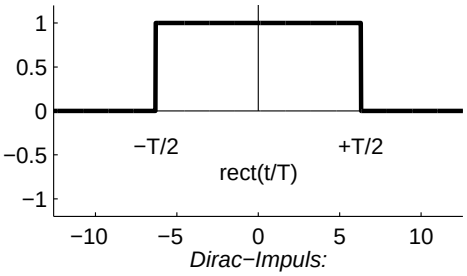
Algorithmus 1 Matlab-Programm zur Berechnung der Pseudo-Inversen mit Hilfe der Singulärwertzerlegung

```
function X = pinv(A,tol)
%PINV   Pseudoinverse.
%   X = PINV(A) produces a matrix X of the same dimensions
%   as A' so that A*X*A = A, X*A*X = X and A*X and X*A
%   are Hermitian. The computation is based on SVD(A) and any
%   singular values less than a tolerance are treated as zero.
%   The default tolerance is MAX(SIZE(A)) * NORM(A) * EPS.
%
%   PINV(A,TOL) uses the tolerance TOL instead of the default.
%
%   See also RANK.

%   Copyright (c) 1984-97 by The MathWorks, Inc.
%   $Revision: 5.6 $  $Date: 1997/04/08 06:27:27 $

[U,S,V] = svd(A,0);
[m,n] = size(A);
if m > 1
   s = diag(S);
elseif m == 1
   s = S(1);
else
   s = 0;
end
if nargin < 2
   tol = max(m,n) * max(s) * eps;
end
r = sum(s > tol);
if (r == 0)
   X = zeros(size(A'));
else
   s = diag(ones(r,1)./s(1:r));
   X = V(:,1:r)*s*U(:,1:r)';
end
```

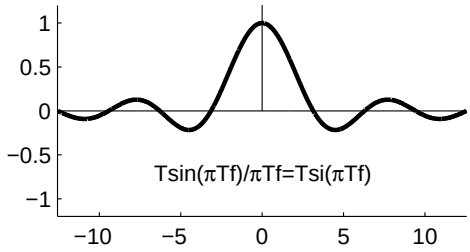
Fourierkorrespondenzen

$x(t)$ ○—● $\tilde{x}(f)$

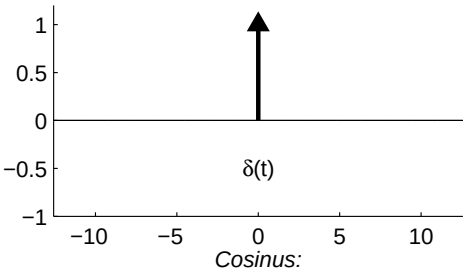
Rechteck−Impuls:

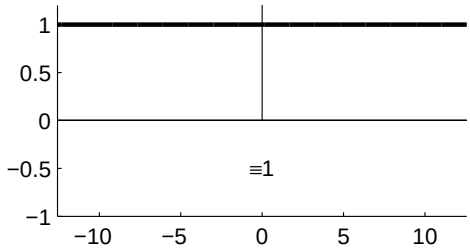


Spalt− oder si−Funktion:

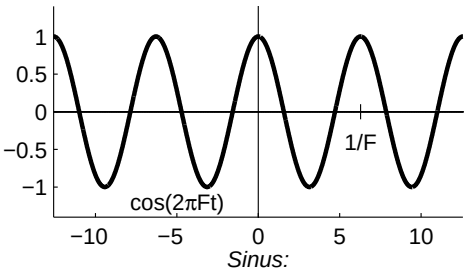


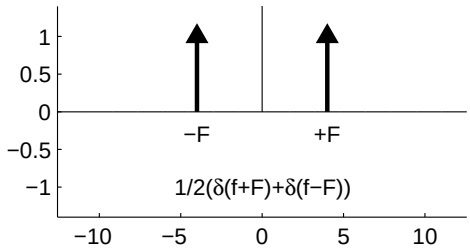
Dirac−Impuls:



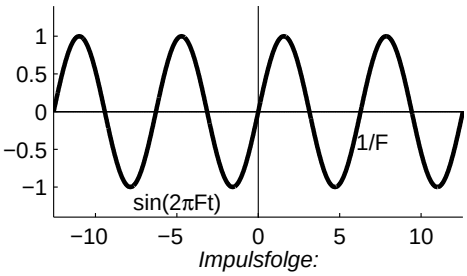


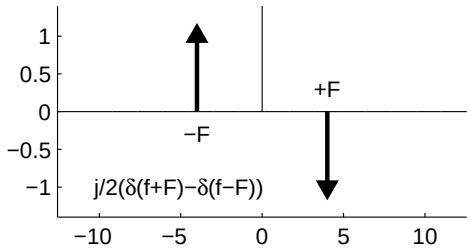
Cosinus:



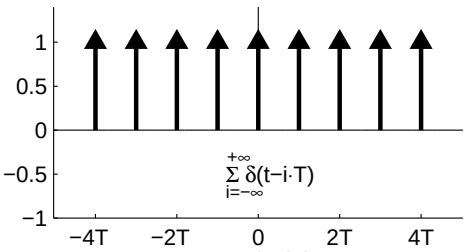


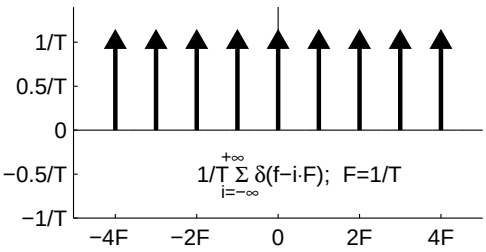
Sinus:



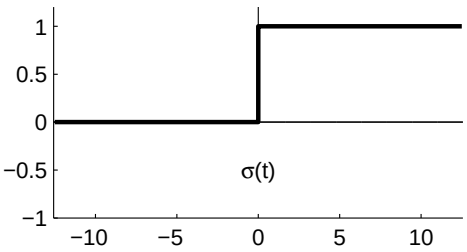


Impulsfolge:



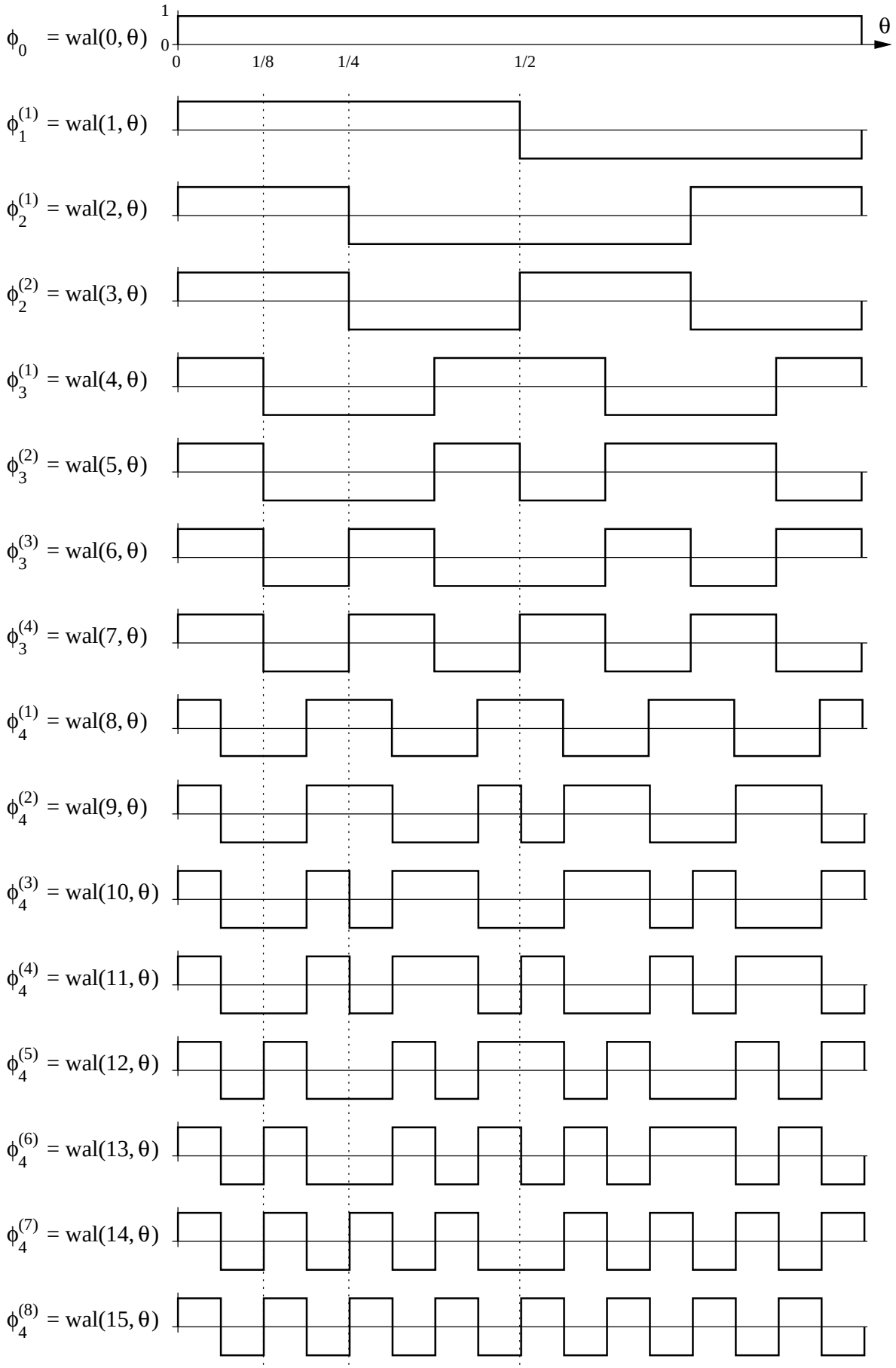


Sprungfunktion:

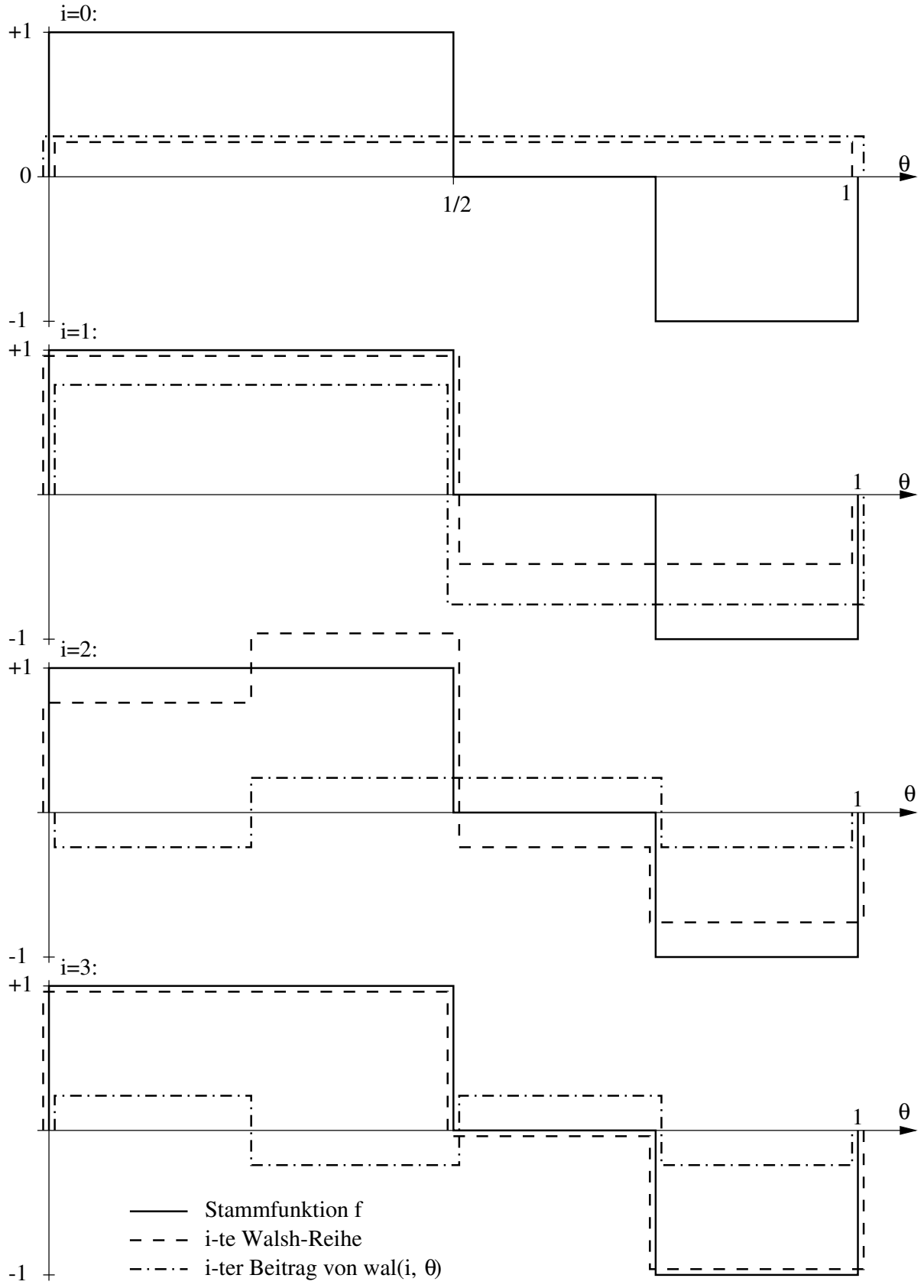


$\pi\delta(2\pi f)+1/j2\pi f$

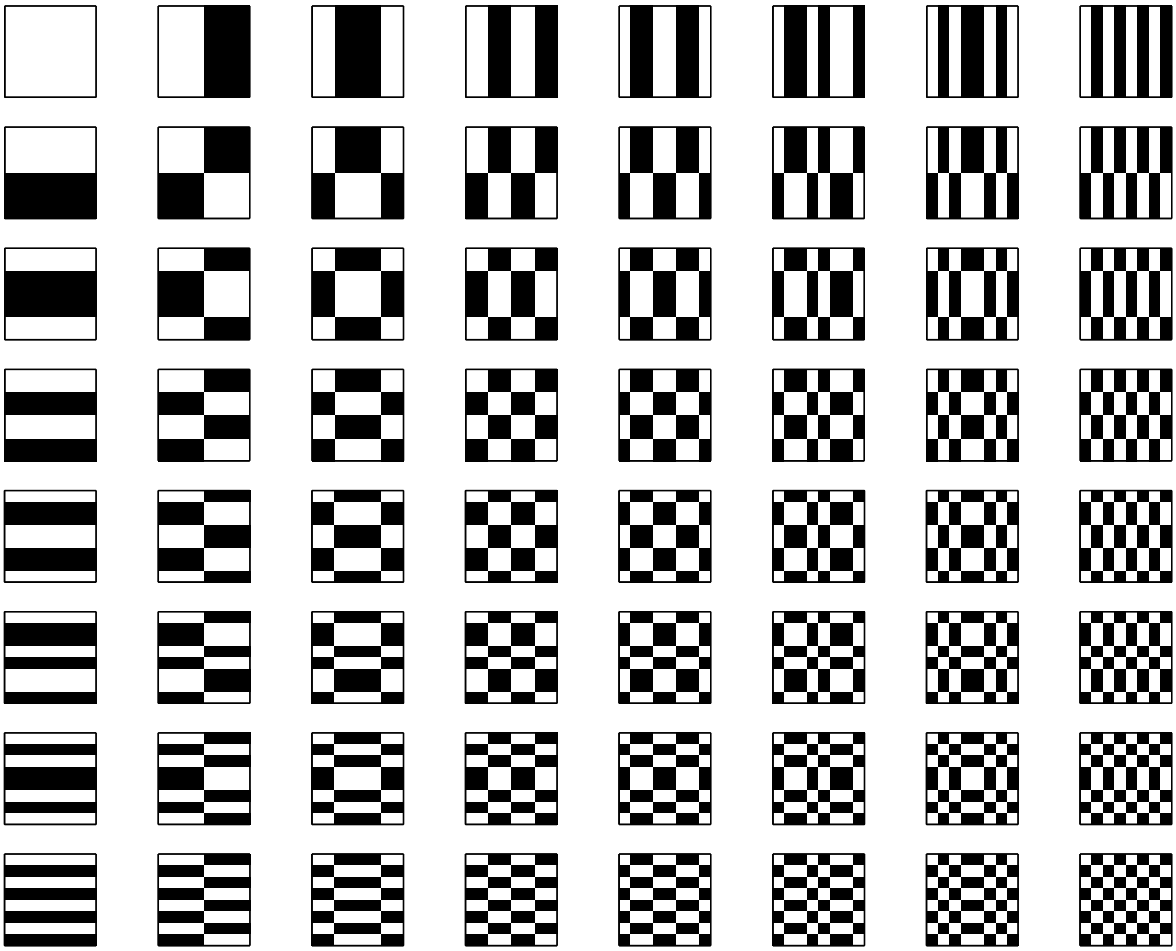
Walsh - Funktionen



Walsh-Approximation für eine diskrete Funktion

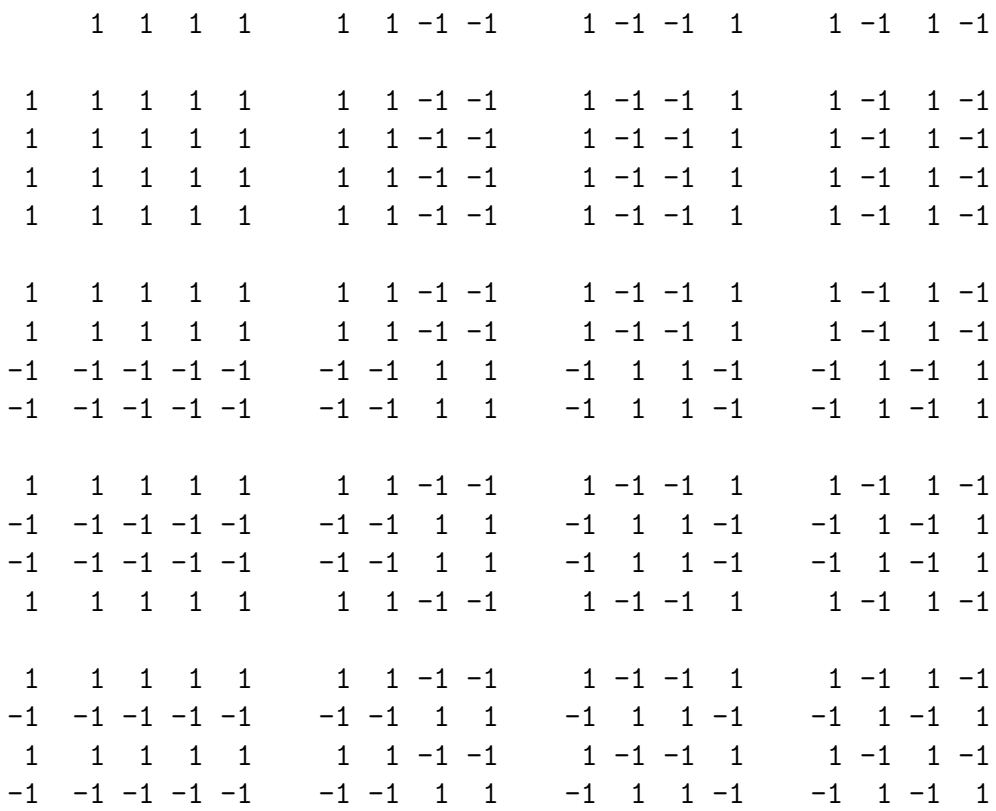


Zweidimensionale Walsh-Funktionen



□ = +1 ■ = −1

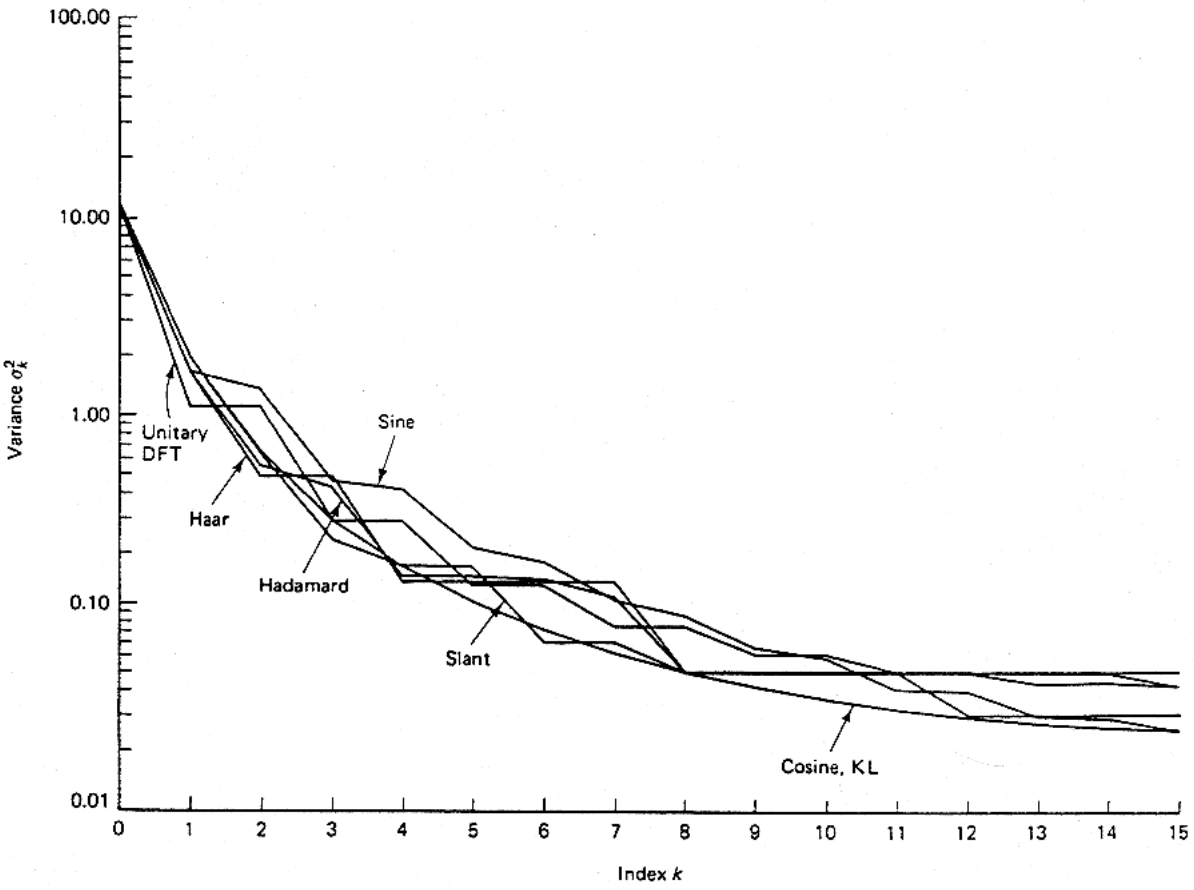
Zweidimensionale Walsh-Funktionen (2)



$$W = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{pmatrix}$$

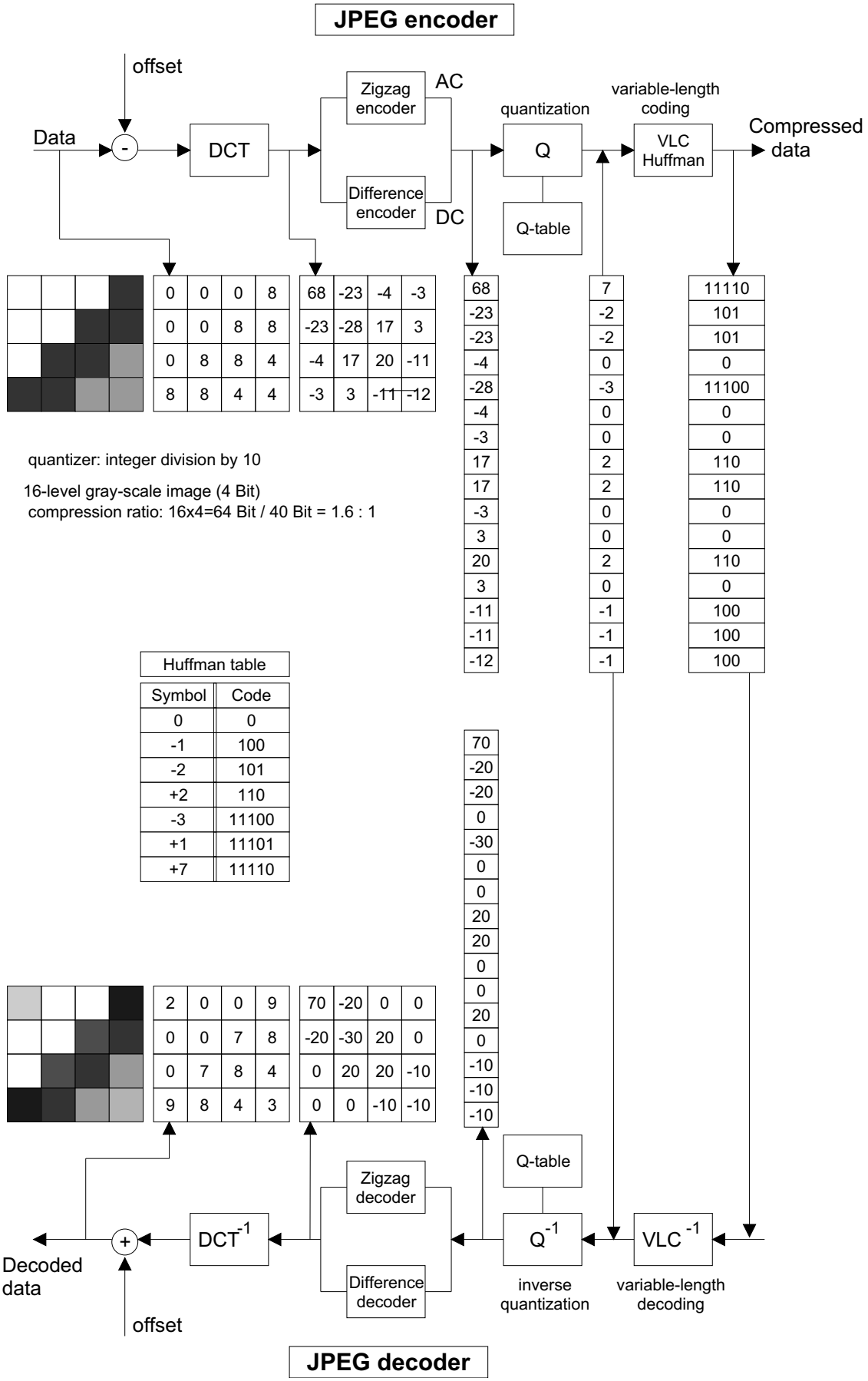
Varianzvergleich der Fourierkoeffizienten unitärer Transformationen

$\downarrow k$ \ Transform	KL	Cosine	Sine	Unitary DFT	Hadamard	Haar	Slant
0	12.442	12.406	11.169	12.406	12.406	12.406	12.406
1	1.946	1.943	1.688	1.100	1.644	1.644	1.904
2	0.615	0.648	1.352	0.292	0.544	0.487	0.641
3	0.292	0.295	0.421	0.139	0.431	0.487	0.233
4	0.171	0.174	0.463	0.086	0.153	0.144	0.173
5	0.114	0.114	0.181	0.062	0.152	0.144	0.172
6	0.082	0.083	0.216	0.051	0.149	0.144	0.072
7	0.063	0.063	0.098	0.045	0.121	0.144	0.072
8	0.051	0.051	0.116	0.043	0.051	0.050	0.051
9	0.043	0.043	0.060	0.045	0.051	0.050	0.051
10	0.037	0.037	0.067	0.051	0.051	0.050	0.051
11	0.033	0.033	0.040	0.062	0.051	0.050	0.051
12	0.030	0.030	0.042	0.086	0.051	0.050	0.031
13	0.028	0.028	0.031	0.139	0.051	0.050	0.031
14	0.027	0.027	0.029	0.292	0.050	0.050	0.031
15	0.026	0.026	0.026	1.100	0.043	0.050	0.031

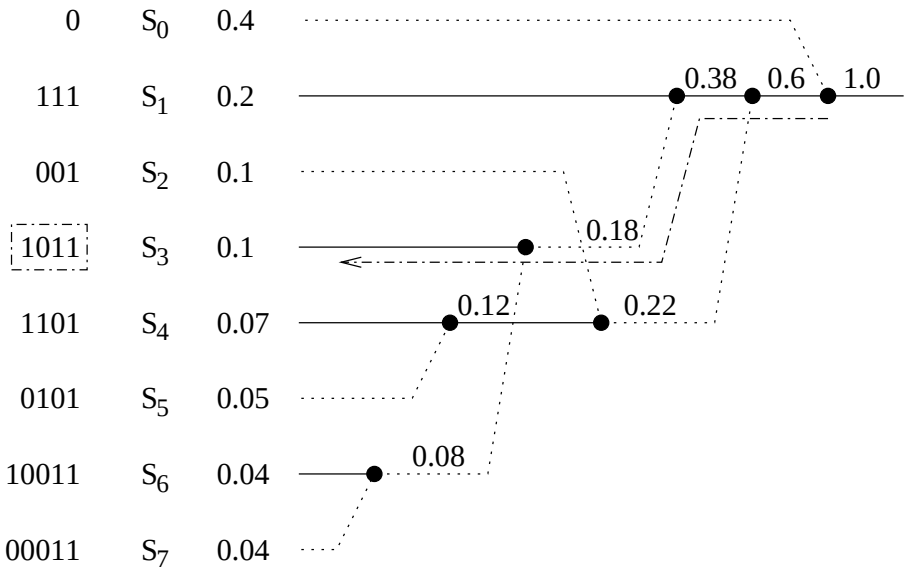


Varianz der Fourierkoeffizienten σ_k^2 verschiedener unitärer Transformationen für eine stationäre Markov-Sequenz mit $\delta = 0,95$ und $N = 16$. (Aus A. K. Jain: "Fundamentals of Digital Image Processing", Prentice Hall 1986)

Einzelbild-Kodierung mit JPEG

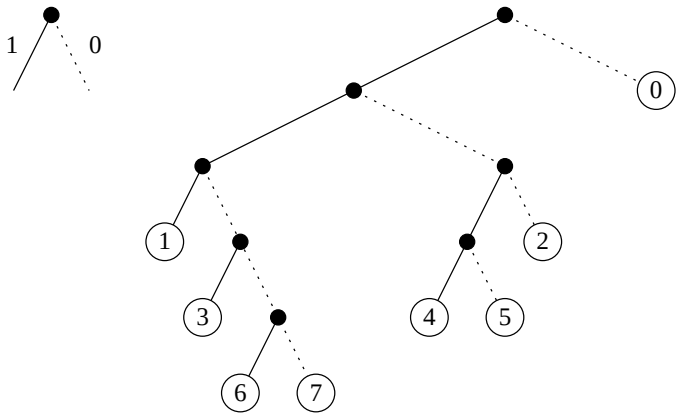


Huffman - Codierung



Fasse in jeder Stufe die zwei kleinsten Wahrscheinlichkeiten zusammen und führe zur Unterscheidung eine Bitstelle hinzu.

Decodierungsbaum

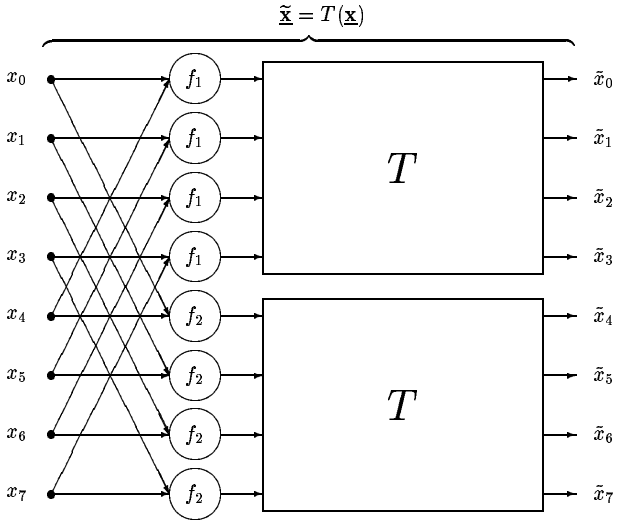


Der Huffman-Code minimiert die mittlere Codelänge:

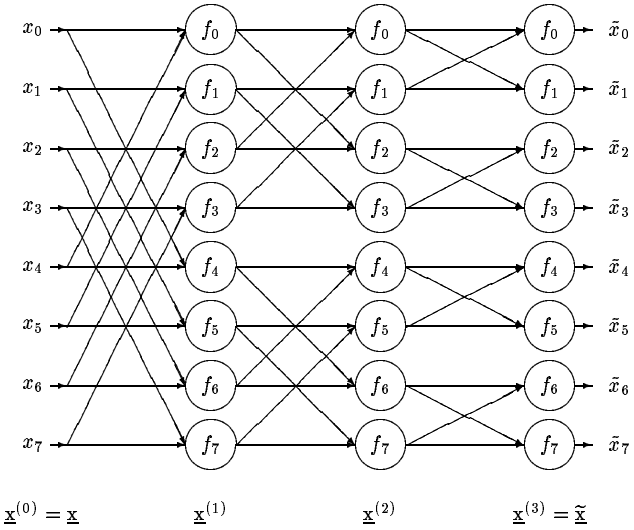
$$\begin{aligned} L_m &= \sum p_i l_i \\ &= 0.4 \cdot 1 + 0.2 \cdot 3 + 0.1 \cdot 3 + 0.2 \cdot 4 + 0.07 \cdot 4 + 0.05 \cdot 4 + 0.04 \cdot 5 + 0.04 \cdot 5 \\ &= 2.58 \end{aligned}$$

Ein Blockcode hätte zum Vergleich eine Codelänge von 3.

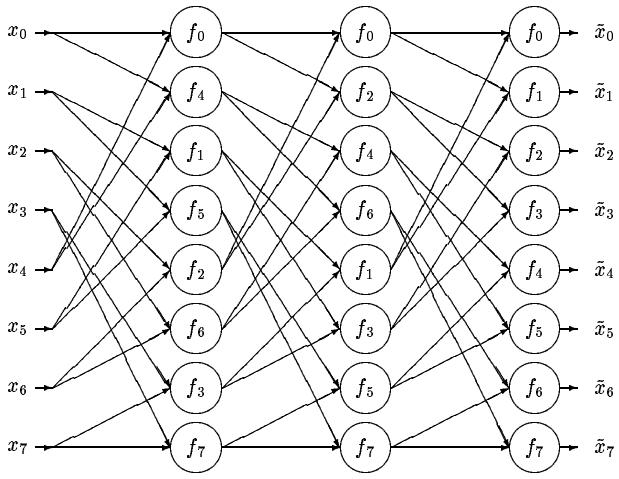
Schnelle Transformationen (1)



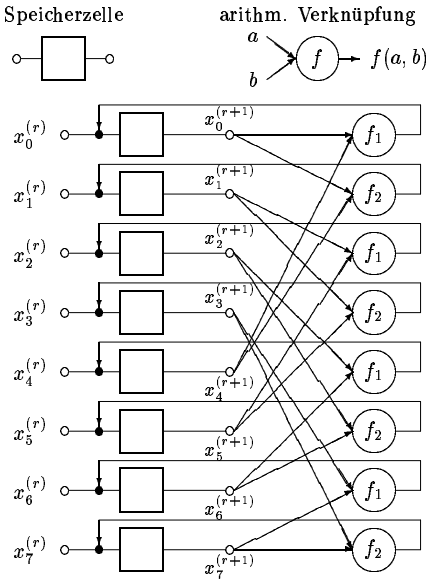
(a) Rekursive Definition der Transformation T



(b) In-Place-Signalflußgraph der schnellen Transformation T ($N = 8$)



(c) Homogener Signalflußgraph der schnellen Transformation T ($N = 8$)



(d) Allgemeiner paralleler Signalprozessor ($N = 8$)

Schneller CT-in-place-Algorithmus

Algorithmus 2 Schneller Basis-2 in-place Algorithmus in Matlab (bereitgestellt im WWW unter http://www.informatik.uni-freiburg.de/~lmb/lectures/index.html)

```
function X = ict( X )
%ICT Schneller ct-in-place-Algorithmus
%   X = ICT(X) berechnet beliebigen schnellen
%   Basis-2-Algorithmus.
%   Operatorwahl durch externe Funktionen f1(a,b) und
%   f2(a,b)

n = length( X );        % Dimension des Eingabevektors
ln = log2( n );
np2 = n;

for i = 1:ln,           % fuer jede Schicht
  np = np2;
  np2 = np/2;
  for m =1:np2,         % fuer jeden Knoten eines Teilbaumes
    for j1 = m:np:n,    % fuer jeden Teilbaum
      j2 = j1 + np2;
      t = f2( X(j1), X(j2) );
      X(j1) = f1( X(j1), X(j2) );
      X(j2) = t;
    end
  end
end
```

Schnelle Transformationen (2)

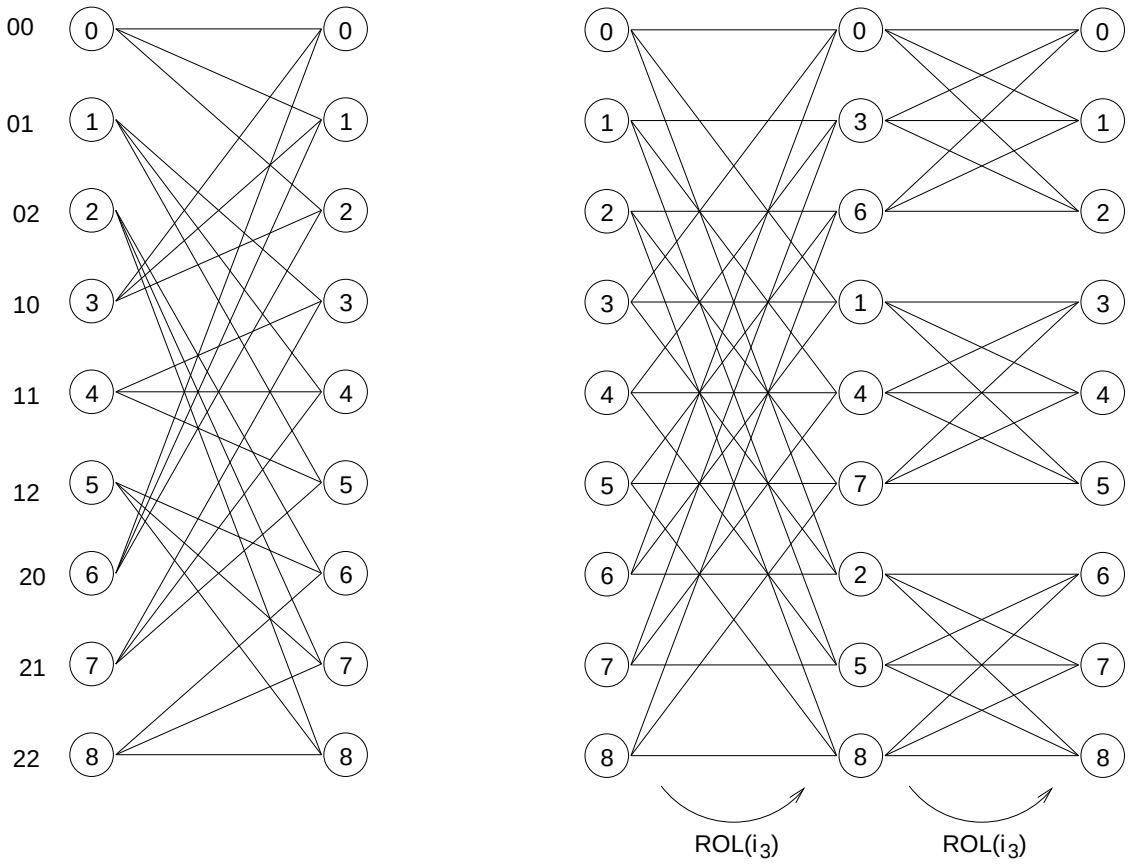


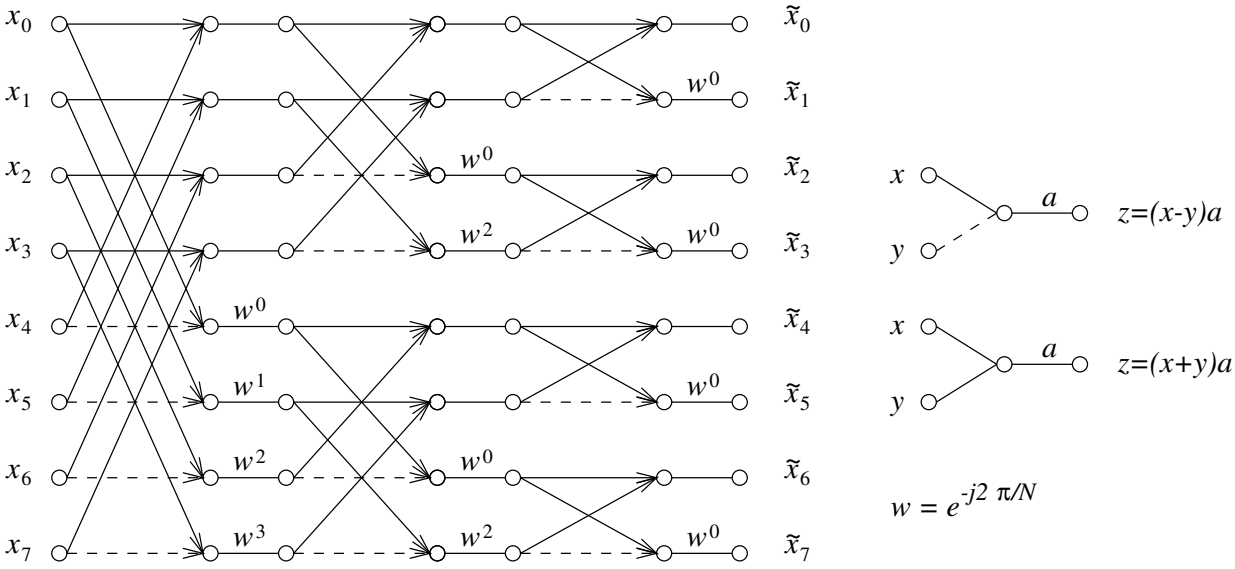
Abbildung 1: Zwei kanonische Formen des schnellen Signalflußgraphen für einen Basis-3-Algorithmus (B=3, N=9)

$$
\begin{array}{ll}
[0] & Z \leftarrow T \quad X; N \\
[1] & \rightarrow (1 > N \leftarrow (, \rho Z \leftarrow X) \div 2)/0 \\
[2] & Z \leftarrow, (T \;\; ((N \uparrow X) \;\; F1 \;\; (N \downarrow X)), [1.5](T \;\; ((N \uparrow X)) \;\; F2 \;\; ((N \downarrow X)))
\end{array}
$$

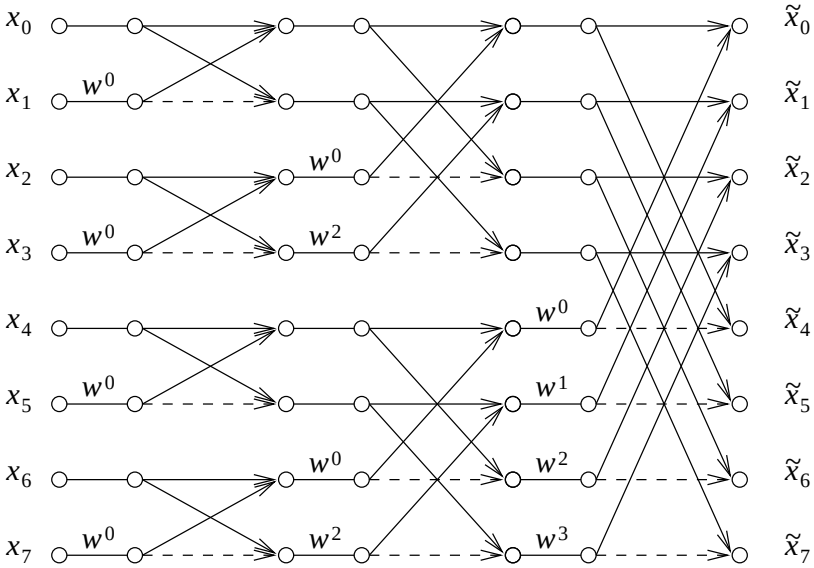
Rekursive Implementierung der verallgemeinerten schnellen Transformation T in APL (inkl. bit-reversal).

Signalflußgraphen der FFT

Sande-Tukey-Algorithmus



Cooley-Tukey-Algorithmus

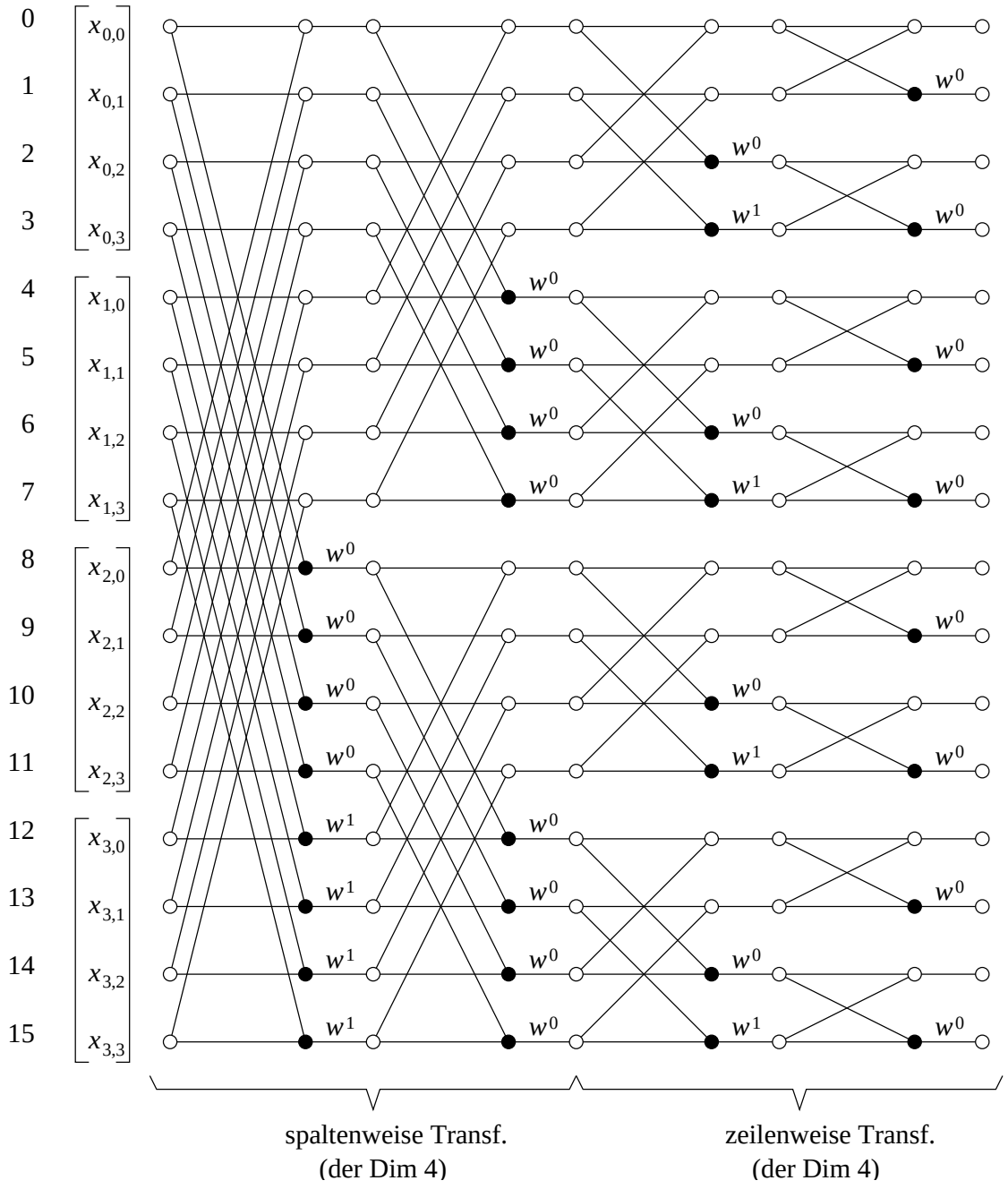


FFT Sande-Tukey-In-Place-Algorithmus

Funktionen in Matlab-Notation:

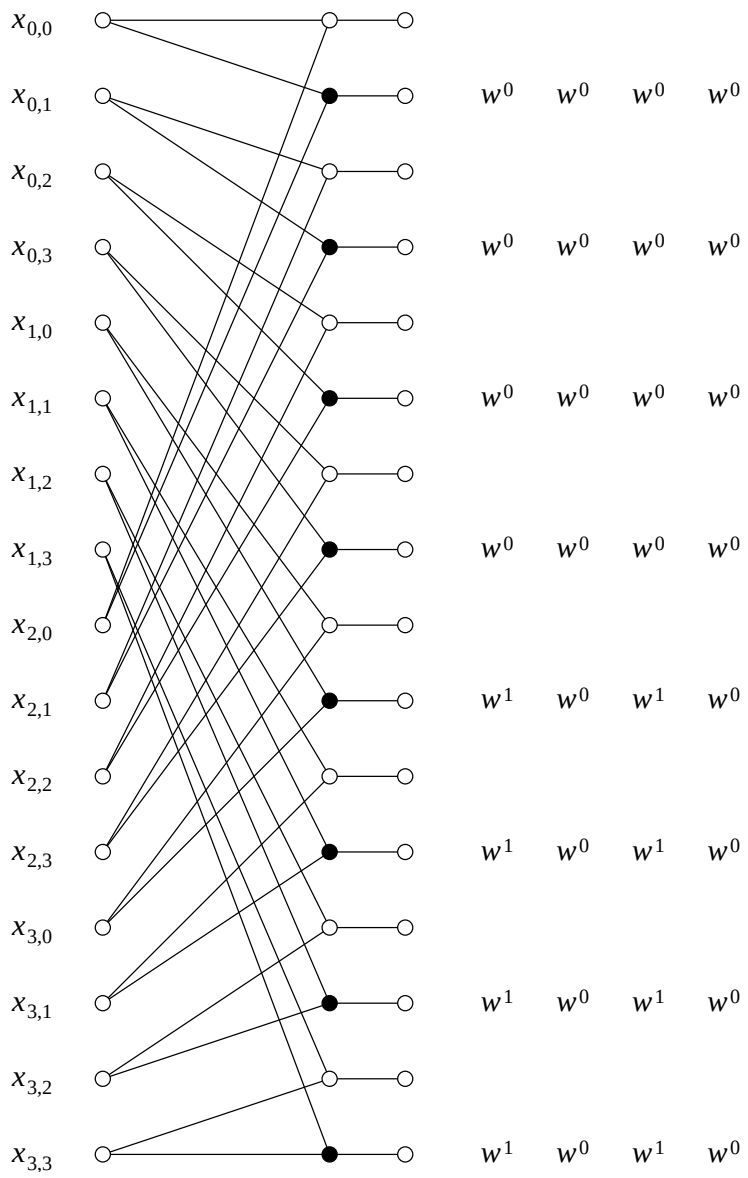
```
function result = FFT(X, N)
% result = FFT(X, N)
%   Eindimensionale FFT (Sande-Tukey-In-Place-Algorithmus)
LN = floor(log2(N));

NP2 = N;
for I=1:LN,
  NP = NP2;
  NP2 = NP/2;
  U = 1.0+ 0i;
  w = exp(i*(-pi/NP2));

  for M=1:NP2,
    for J1=M:NP:N,
      J2 = J1 + NP2;
      T = X(J1) - X(J2);
      X(J1) = X(J1) + X(J2);
      X(J2) = T * U;
    end;
    U=U*w;
  end;
end;
result = BR(X,N);


function result = BR(X, N)
% result = BR(X, N)
%   Fuehrt ein Bit-reversal auf dem Vektor X der Laenge N durch
NH = N / 2;
J = 1;
for I=1:N-1,
  if (I < J),
    T = X(J);
    X(J) = X(I);
    X(I) = T;
  end;
  K = NH;
  while (K < J),
    J = J-K;
    K = K/2;
  end;
  J=J+K;
end;
result = X;
return;
```

Eindimensionale Realisierung der zweidimensionalen FFT



$$w = e^{-j2\pi/4} = -j$$

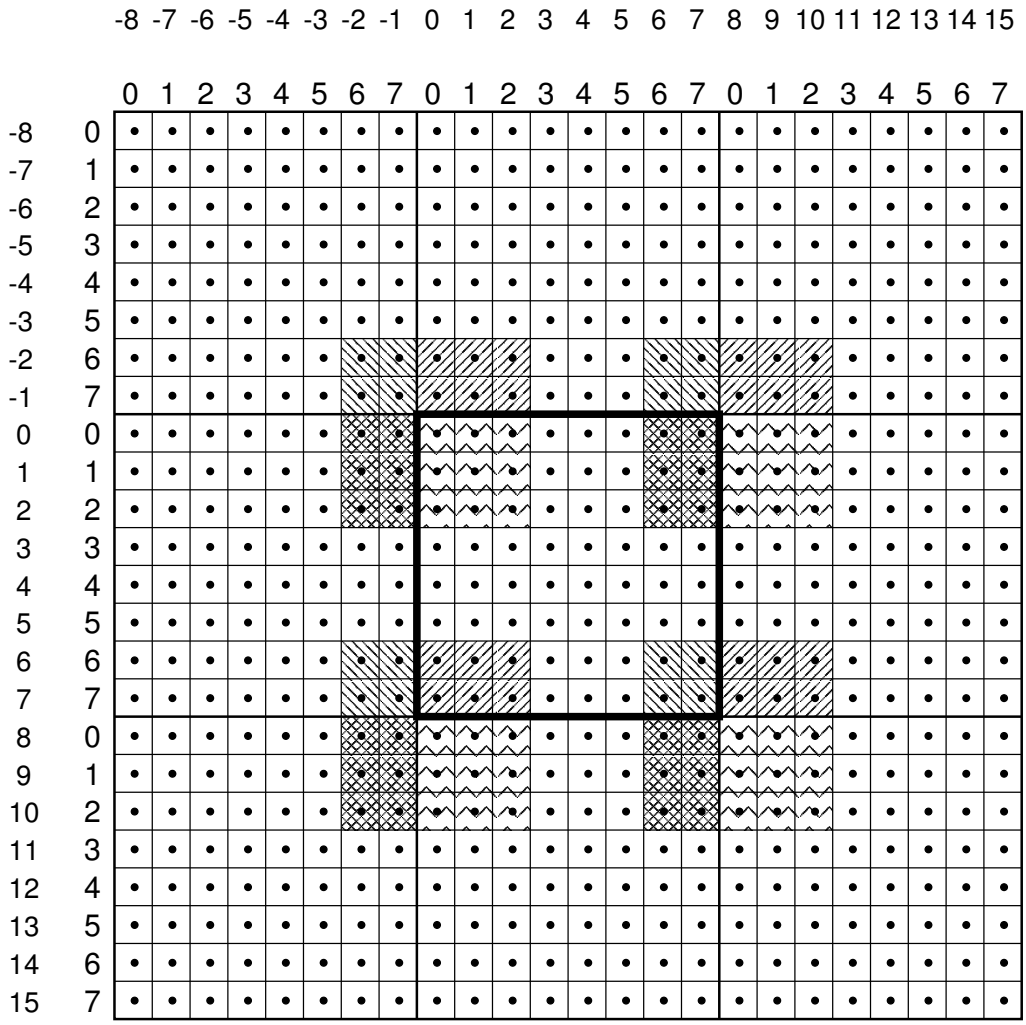
Homogene Struktur zur eindimensionalen Realisierung der zweidimensionalen FFT



Ein Beispiel zur 2D-Fourier & Walsh Transformation

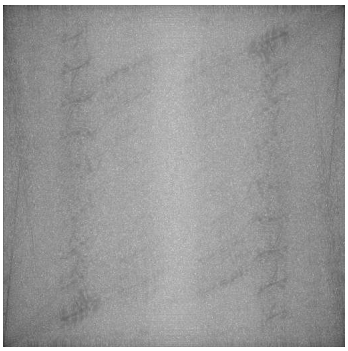
```
A =

     1    12    18    13     2    17    19    20
    16     5    17     4    20    14     7    16
     5    12    11     7     2    12     4     8
    19     3    10     2     8    12    16     8


>> fft2(A) =

   1.0e+02 *

  Columns 1 through 4

   3.4000             0.1112 + 0.2465i  -0.2900 - 0.0900i   0.0688 + 0.4465i
   0.4100 - 0.2100i  -0.1505 + 0.2654i  -0.2000 + 0.0200i   0.1554 + 0.1902i
  -0.1400            -0.0288 - 0.1826i  -0.5500 - 0.0100i  -0.0712 - 0.1426i
   0.4100 + 0.2100i   0.0846 + 0.0502i  -0.3200 + 0.2400i  -0.2495 - 0.1946i

  Columns 5 through 8

   0.1000             0.0688 - 0.4465i  -0.2900 + 0.0900i   0.1112 - 0.2465i
  -0.0500 + 0.0700i  -0.2495 + 0.1946i  -0.3200 - 0.2400i   0.0846 - 0.0502i
  -0.8800            -0.0712 + 0.1426i  -0.5500 + 0.0100i  -0.0288 + 0.1826i
  -0.0500 - 0.0700i   0.1554 - 0.1902i  -0.2000 - 0.0200i  -0.1505 - 0.2654i



>> walsh2(A) =

   340   -30     2   -20   -38    -4    68    10
    62   -28   -20   -34     4   -34     6   -12
    20   -18    30   -44   -30    40    -8     2
   -14    20    -4   -54   -56   -10   -26   -88
```

Periodische Fortsetzung des Spektrums eines 2D-Signal

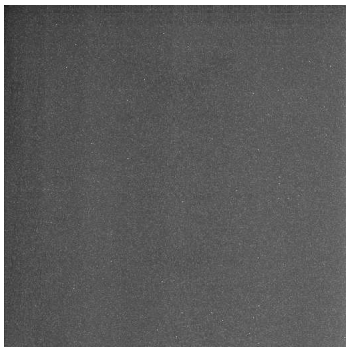


Beispiele für Fourier- und Walshtransformation (Originalbilder)

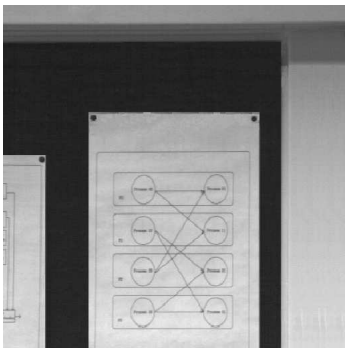
(a) Haus 512 × 512



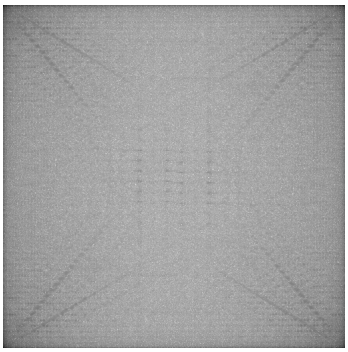
(b) Fourierspektrum (Haus)



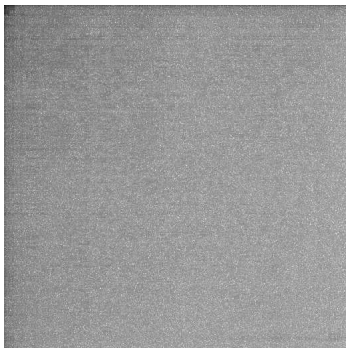
(c) Walshtranformierte (Haus)



(d) Poster 512 × 512

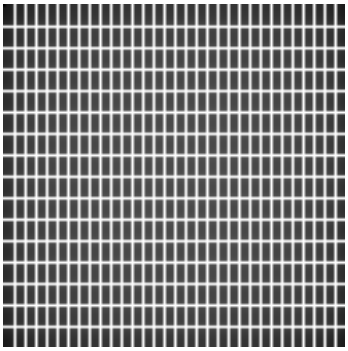


(e) Fourierspektrum (Poster)



(f) Walshtranformierte (Poster)

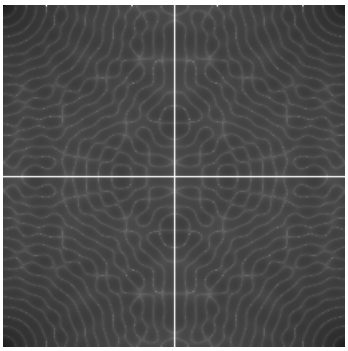
(g) Rechteck 128 × 128



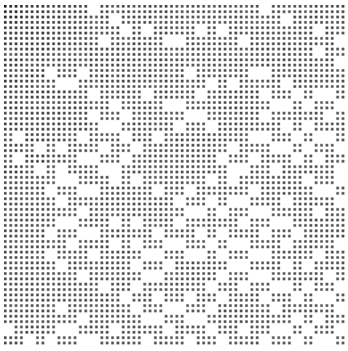
(h) Fourierspektrum (Rechteck)

(i) Walshtranformierte (Rechteck)

(j) Kreis 256 × 256

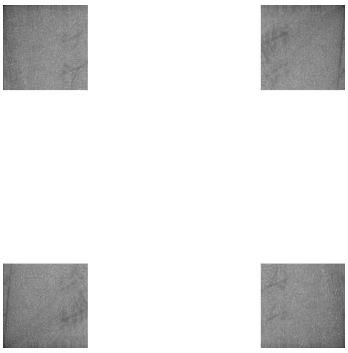


(k) Fourierspektrum (Kreis)

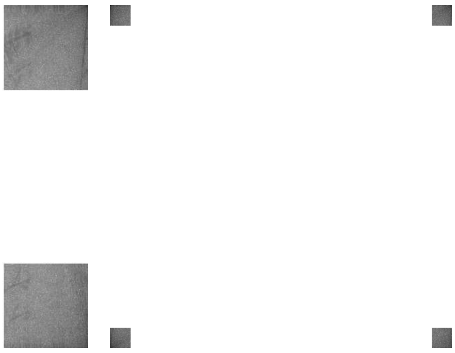


(l) Walshtranformierte (Kreis)

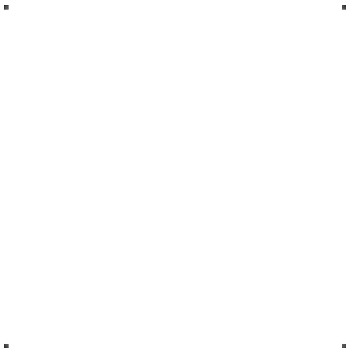
Fourier-Transformierte (Natürliche Szene)



(a) TP mit 256^2 Koeff.; Spektrum



(b) TP mit 64^2 Koeff.; Spektrum

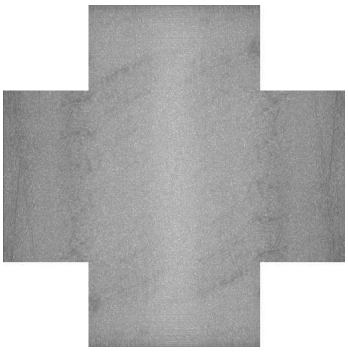


(c) TP mit 16^2 Koeff.; Spektrum

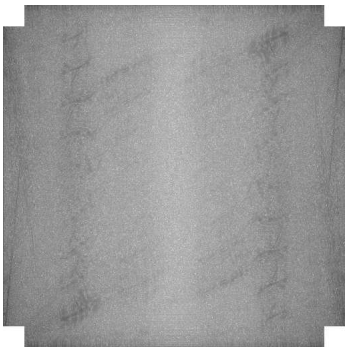
(d) TP-Synthese mit 256^2 Koeff.; RMSE=10,4

(e) TP-Synthese mit 64^2 Koeff.; RMSE=24,6

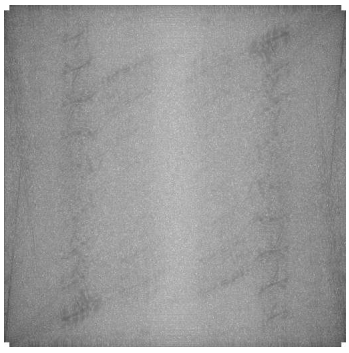
(f) TP-Synthese mit 16^2 Koeff.; RMSE=35,6



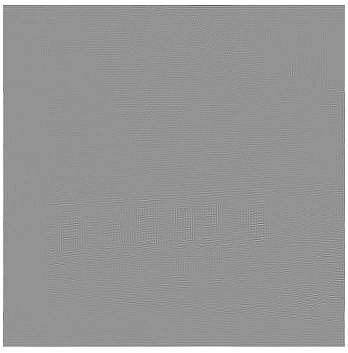
(g) HP mit 512^2-256^2 Koeff.; Spektrum



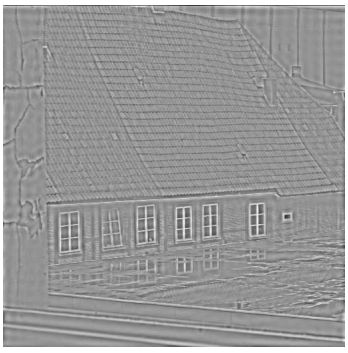
(h) HP mit 512^2-64^2 Koeff.; Spektrum



(i) HP mit 512^2-16^2 Koeff.; Spektrum



(j) HP-Synthese mit 512^2-256^2 Koeff.; RMSE=180,1

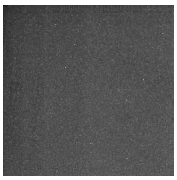


(k) HP-Synthese mit 512^2-64^2 Koeff.; RMSE=178,7

(l) HP-Synthese mit 512^2-16^2 Koeff.; RMSE=176,8

$$\text{RMSE} = \sqrt{\frac{\sum(X_{\text{org}}(i,j)-X_{\text{synth}}(i,j))^2}{NM}}$$

WALSH-Transformierte (Natürliche Szene)



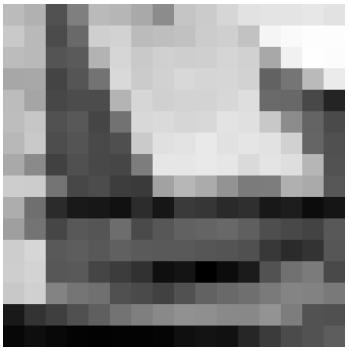
(a) TP mit 256^2 Koeff.; Transformierte

(b) TP mit 64^2 Koeff.; Transformierte

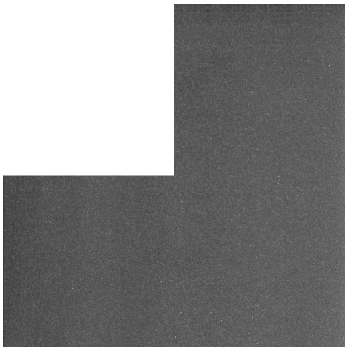
(c) TP mit 16^2 Koeff.; Transformierte

(d) TP-Synthese mit 256^2 Koeff.; RMSE=14,0

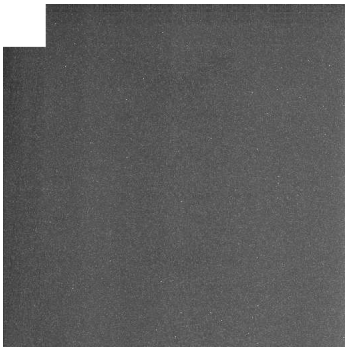
(e) TP-Synthese mit 64^2 Koeff.; RMSE=26,3



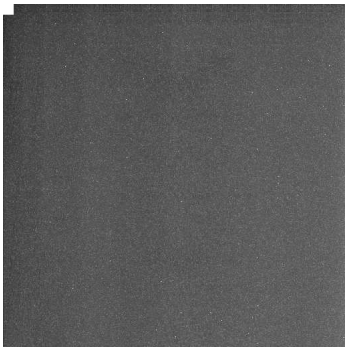
(f) TP-Synthese mit 16^2 Koeff.; RMSE=35,9



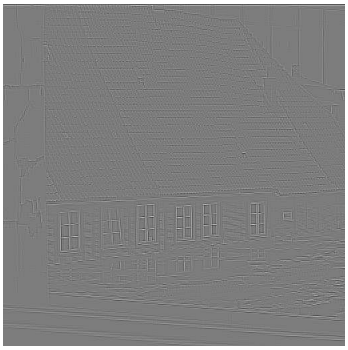
(g) HP mit $512^2 - 256^2$ Koeff.; Transformierte



(h) HP mit $512^2 - 64^2$ Koeff.; Transformierte



(i) HP mit $512^2 - 16^2$ Koeff.; Transformierte

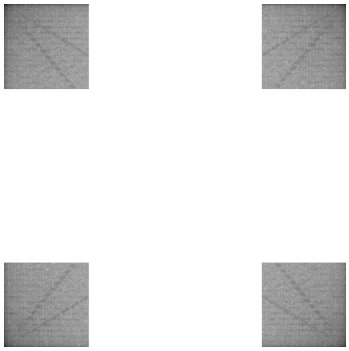


(j) HP-Synthese mit $512^2 - 256^2$ Koeff.; RMSE=179,8

(k) HP-Synthese mit $512^2 - 64^2$ Koeff.; RMSE=178,4

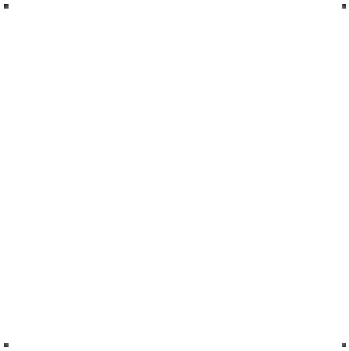
(l) HP-Synthese mit $512^2 - 16^2$ Koeff.; RMSE=176,7

Fourier-Transformierte (Künstliche Szene)

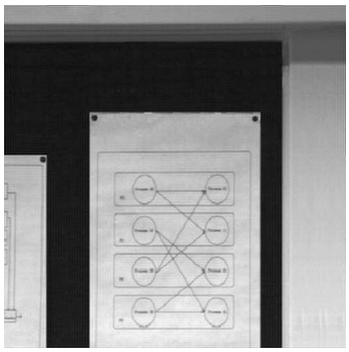


(a) TP mit 256^2 Koeff.; Spektrum

(b) TP mit 64^2 Koeff.; Spektrum

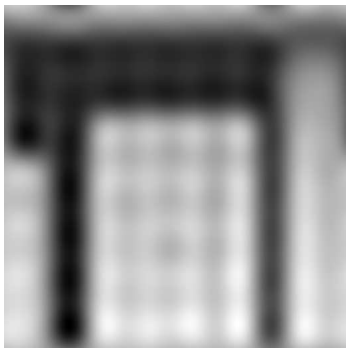


(c) TP mit 16^2 Koeff.; Spektrum

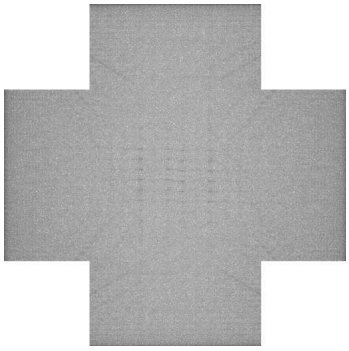


(d) TP-Synthese mit 256^2 Koeff.; RMSE=5,2

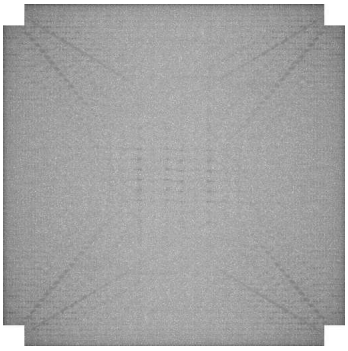
(e) TP-Synthese mit 64^2 Koeff.; RMSE=12,3



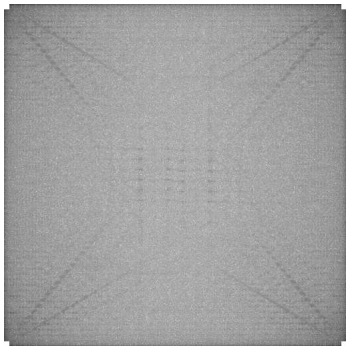
(f) TP-Synthese mit 16^2 Koeff.; RMSE=24,7



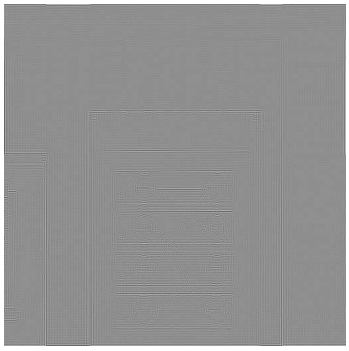
(g) HP mit 512^2-256^2 Koeff.; Spektrum



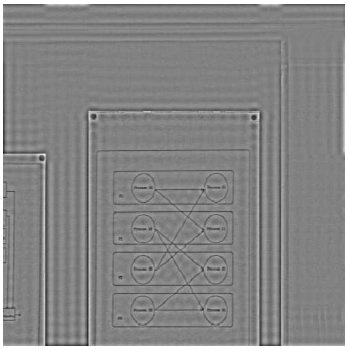
(h) HP mit 512^2-64^2 Koeff.; Spektrum



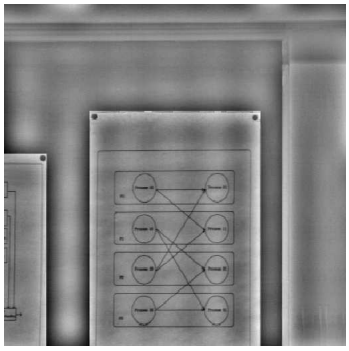
(i) HP mit 512^2-16^2 Koeff.; Spektrum



(j) HP-Synthese mit 512^2-256^2 Koeff.; RMSE=151,3

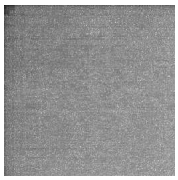


(k) HP-Synthese mit 512^2-64^2 Koeff.; RMSE=150,8



(l) HP-Synthese mit 512^2-16^2 Koeff.; RMSE=149,3

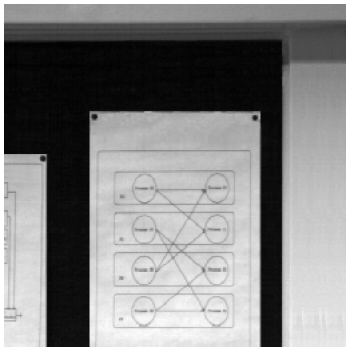
WALSH-Transformierte (Künstliche Szene)



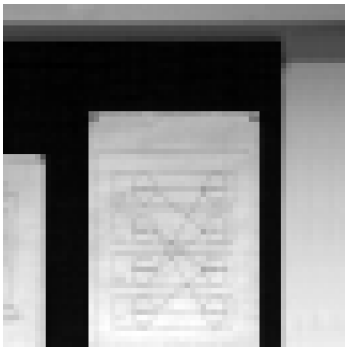
(a) TP mit 256^2 Koeff.; Transformierte

(b) TP mit 64^2 Koeff.; Transformierte

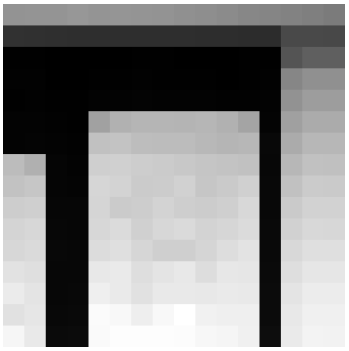
(c) TP mit 16^2 Koeff.; Transformierte



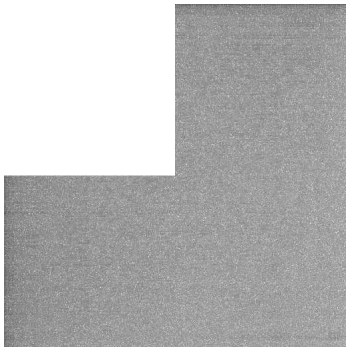
(d) TP-Synthese mit 256^2 Koeff.; RMSE=5,4



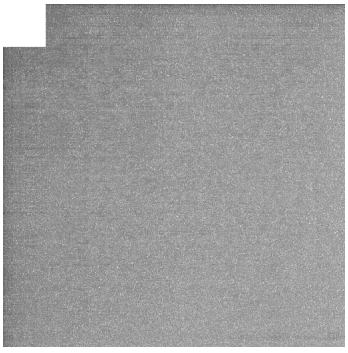
(e) TP-Synthese mit 64^2 Koeff.; RMSE=9,7



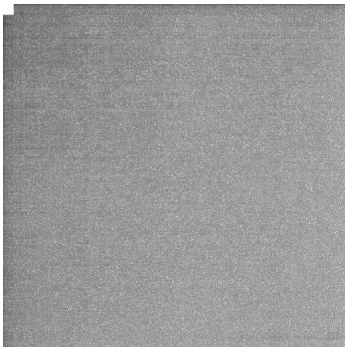
(f) TP-Synthese mit 16^2 Koeff.; RMSE=12,8



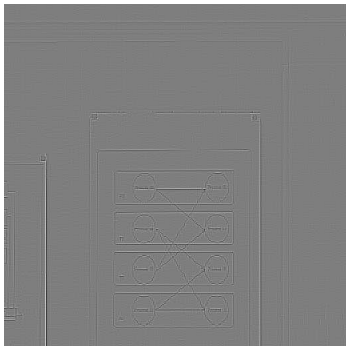
(g) HP mit 512^2-256^2 Koeff.; Transformierte



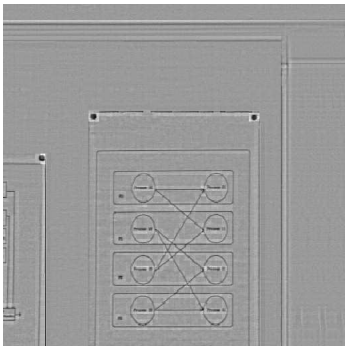
(h) HP mit 512^2-64^2 Koeff.; Transformierte



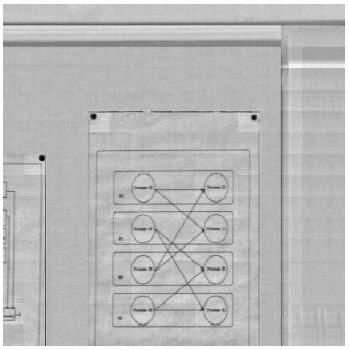
(i) HP mit 512^2-16^2 Koeff.; Transformierte



(j) HP-Synthese mit 512^2-256^2 Koeff.; RMSE=150,9



(k) HP-Synthese mit 512^2-64^2 Koeff.; RMSE=151,1



(l) HP-Synthese mit 512^2-16^2 Koeff.; RMSE=151,3

Fourier-Transformierte bei rechteckigen Strukturen

(a) TP mit 64^2 Koeff.; Spektrum

(b) TP mit 16^2 Koeff.; Spektrum

(c) TP mit 4^2 Koeff.; Spektrum

(d) TP-Synthese mit 64^2 Koeff.; RMSE=7,8

(e) TP-Synthese mit 16^2 Koeff.; RMSE=16,9

(f) TP-Synthese mit 4^2 Koeff.; RMSE=39,4

(g) HP mit $128^2 - 64^2$ Koeff.; Spektrum

(h) HP mit $128^2 - 16^2$ Koeff.; Spektrum

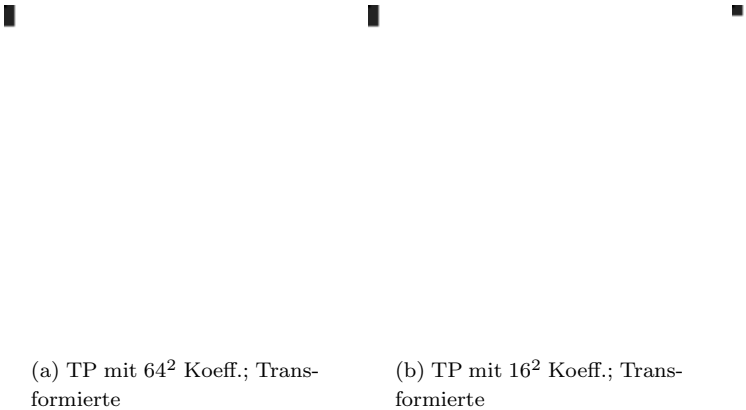
(i) HP mit $128^2 - 4^2$ Koeff.; Spektrum

(j) HP-Synthese mit $128^2 - 64^2$ Koeff.; RMSE=250,9

(k) HP-Synthese mit $128^2 - 16^2$ Koeff.; RMSE=250,4

(l) HP-Synthese mit $128^2 - 4^2$ Koeff.; RMSE=247,9

WALSH-Transformierte bei rechteckigen Strukturen



(a) TP mit 64^2 Koeff.; Transformierte

(b) TP mit 16^2 Koeff.; Transformierte

(c) TP mit 4^2 Koeff.; Transformierte

(d) TP-Synthese mit 64^2 Koeff.; RMSE=0.0

(e) TP-Synthese mit 16^2 Koeff.; RMSE=0.0

(f) TP-Synthese mit 4^2 Koeff.; RMSE=31,9

(g) HP mit $128^2 - 64^2$ Koeff.; Transformierte

(h) HP mit $128^2 - 16^2$ Koeff.; Transformierte

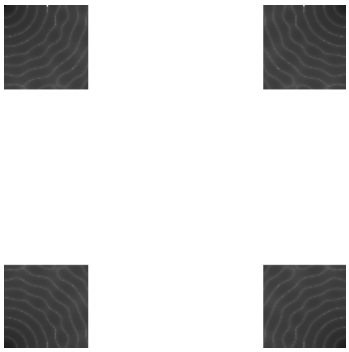
(i) HP mit $128^2 - 4^2$ Koeff.; Transformierte

(j) HP-Synthese mit $128^2 - 64^2$ Koeff.; RMSE=251,0

(k) HP-Synthese mit $128^2 - 16^2$ Koeff.; RMSE=251,0

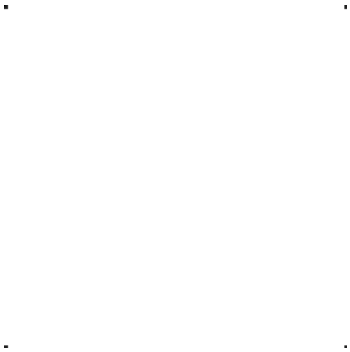
(l) HP-Synthese mit $128^2 - 4^2$ Koeff.; RMSE=249,0

Fourier-Transformierte bei Kreisstrukturen

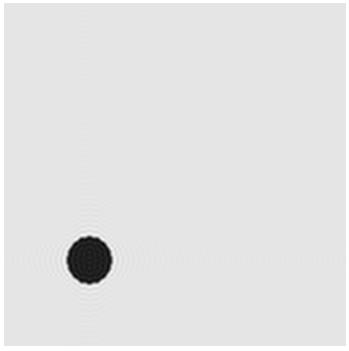


(a) TP mit 128^2 Koeff.; Spektrum

(b) TP mit 32^2 Koeff.; Spektrum

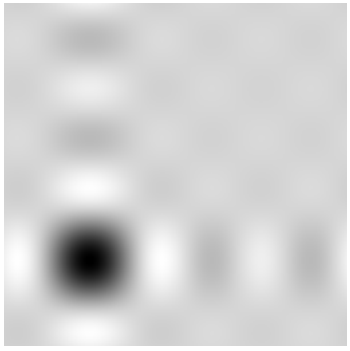


(c) TP mit 8^2 Koeff.; Spektrum

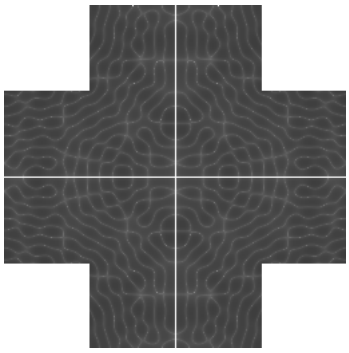


(d) TP-Synthese mit 128^2 Koeff.; RMSE=4,2

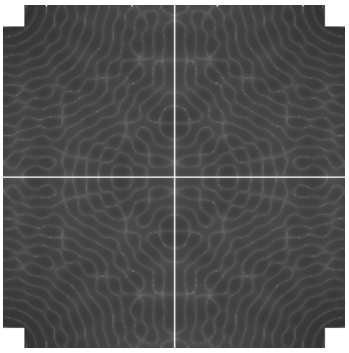
(e) TP-Synthese mit 32^2 Koeff.; RMSE=9,0



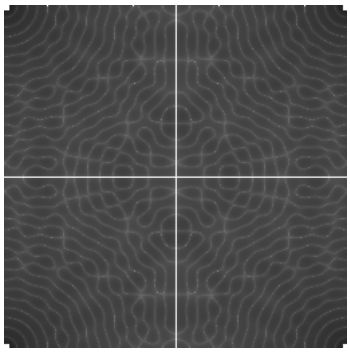
(f) TP-Synthese mit 8^2 Koeff.; RMSE=21,7



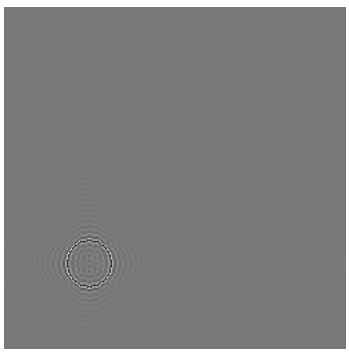
(g) HP mit 256^2-128^2 Koeff.; Spektrum



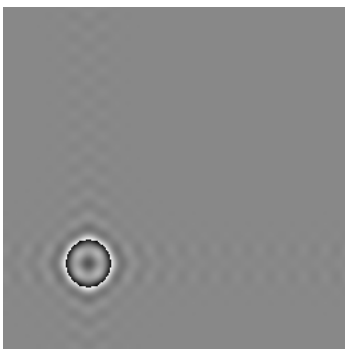
(h) HP mit 256^2-32^2 Koeff.; Spektrum



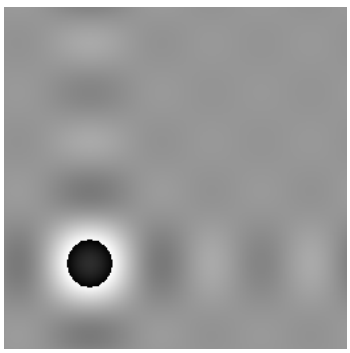
(i) HP mit 256^2-8^2 Koeff.; Spektrum



(j) HP-Synthese mit 256^2-128^2 Koeff.; RMSE=253,1



(k) HP-Synthese mit 256^2-32^2 Koeff.; RMSE=253,0



(l) HP-Synthese mit 256^2-8^2 Koeff.; RMSE=252,2

WALSH-Transformierte bei Kreisstrukturen

(a) TP mit 128^2 Koeff.; Transformierte

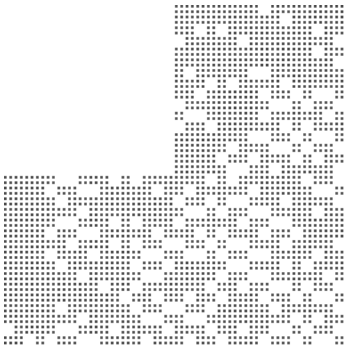
(b) TP mit 32^2 Koeff.; Transformierte

(c) TP mit 8^2 Koeff.; Transformierte

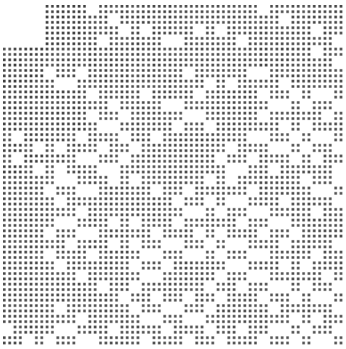
(d) TP-Synthese mit 256^2 Koeff.; RMSE=5,8

(e) TP-Synthese mit 32^2 Koeff.; RMSE=10,5

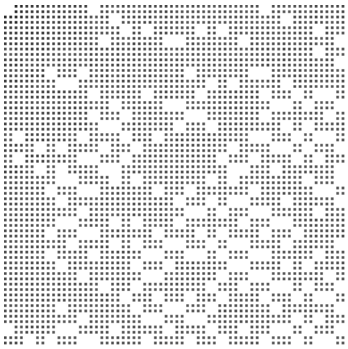
(f) TP-Synthese mit 8^2 Koeff.; RMSE=26,8



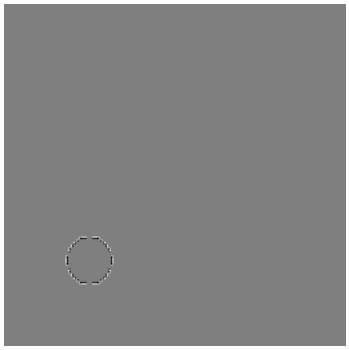
(g) HP mit 256^2-128^2 Koeff.; Transformierte



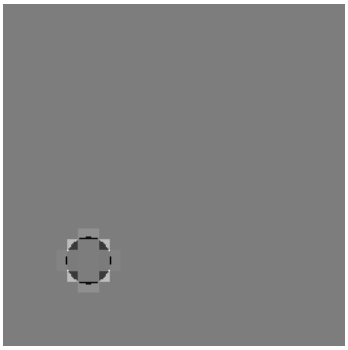
(h) HP mit 256^2-32^2 Koeff.; Transformierte



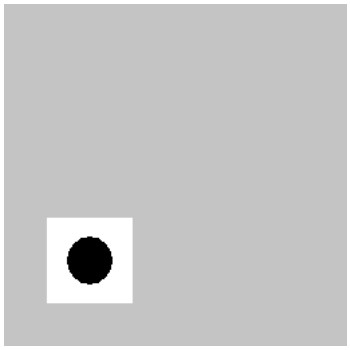
(i) HP mit 256^2-8^2 Koeff.; Transformierte



(j) HP-Synthese mit 256^2-128^2 Koeff.; RMSE=253,1

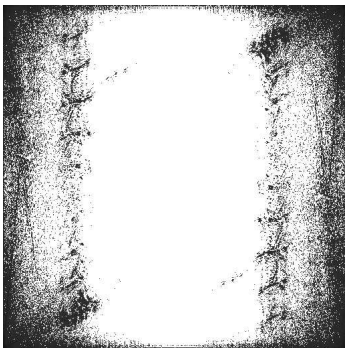


(k) HP-Synthese mit 256^2-32^2 Koeff.; RMSE=253,0

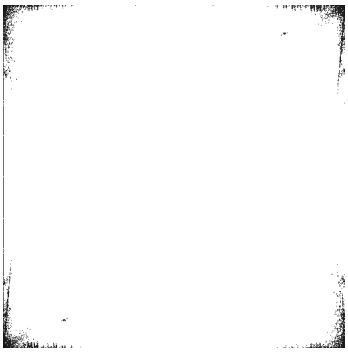


(l) HP-Synthese mit 256^2-8^2 Koeff.; RMSE=251,8

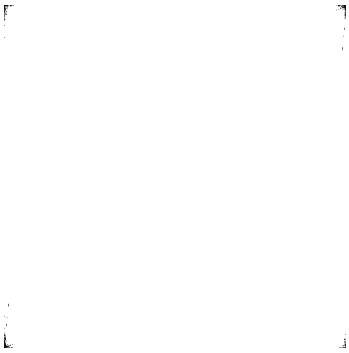
Fourier-Transformation mit den N größten Koeffizienten



(a) 256^2 größte Koeffizienten



(b) 64^2 größte Koeffizienten

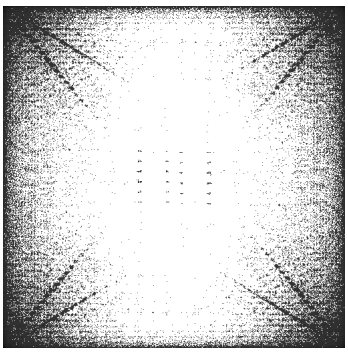


(c) 16^2 größte Koeffizienten

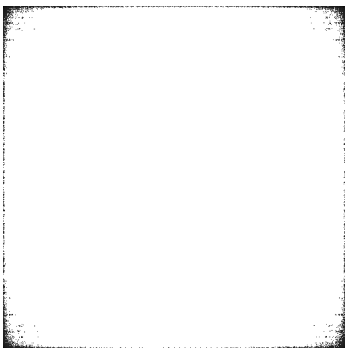
(d) 256^2 größte Koeffizienten; RMSE=5,29

(e) 64^2 größte Koeffizienten; RMSE=17,8

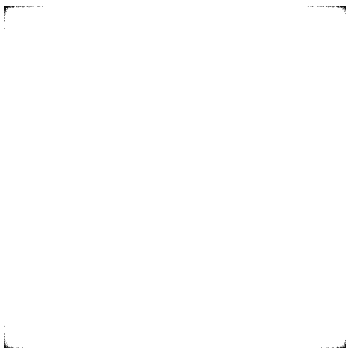
(f) 16^2 größte Koeffizienten; RMSE=30,4



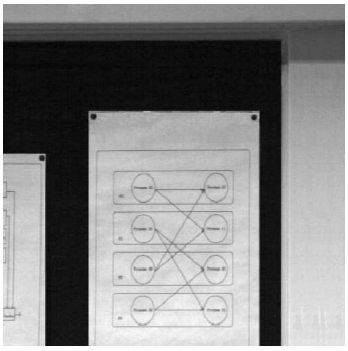
(g) 256^2 größte Koeffizienten



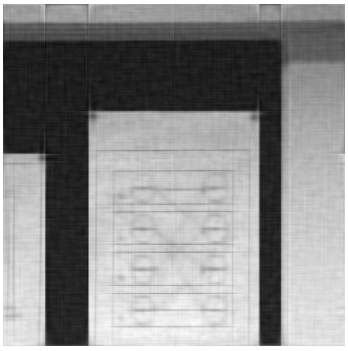
(h) 64^2 größte Koeffizienten



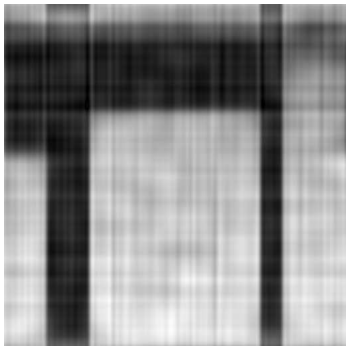
(i) 16^2 größte Koeffizienten



(j) 256^2 größte Koeffizienten; RMSE=2,04

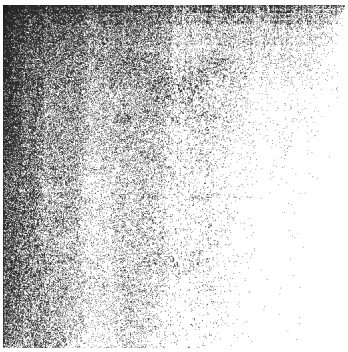


(k) 64^2 größte Koeffizienten; RMSE=6,85

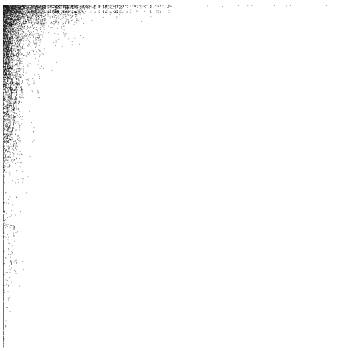


(l) 16^2 größte Koeffizienten; RMSE=15,9

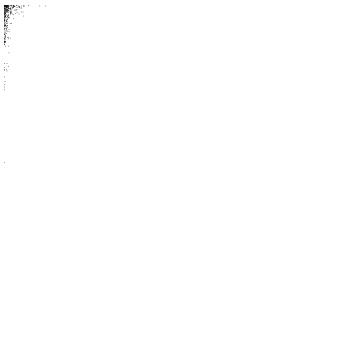
WALSH-Transformation mit den N größten Koeffizienten



(a) 256^2 größte Koeffizienten



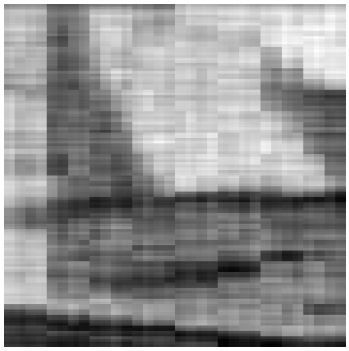
(b) 64^2 größte Koeffizienten



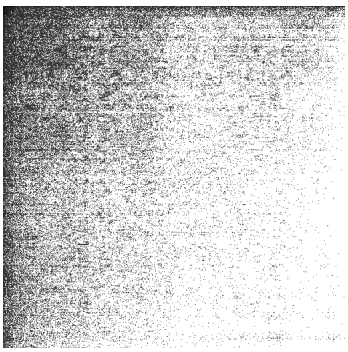
(c) 16^2 größte Koeffizienten

(d) 256^2 größte Koeffizienten; RMSE=6,26

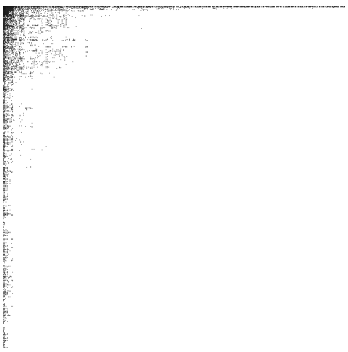
(e) 64^2 größte Koeffizienten; RMSE=19,5



(f) 16^2 größte Koeffizienten; RMSE=31,4

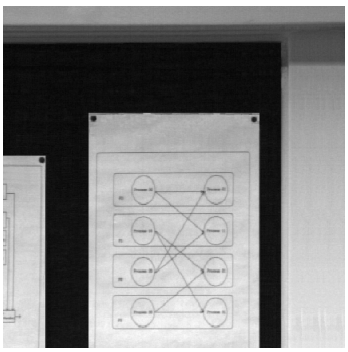


(g) 256^2 größte Koeffizienten

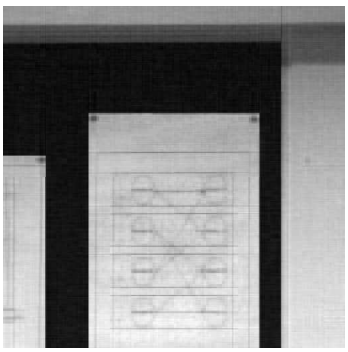


(h) 64^2 größte Koeffizienten

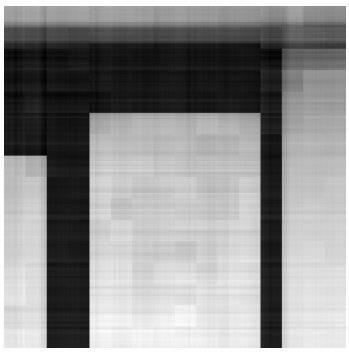
(i) 16^2 größte Koeffizienten



(j) 256^2 größte Koeffizienten; RMSE=2,09



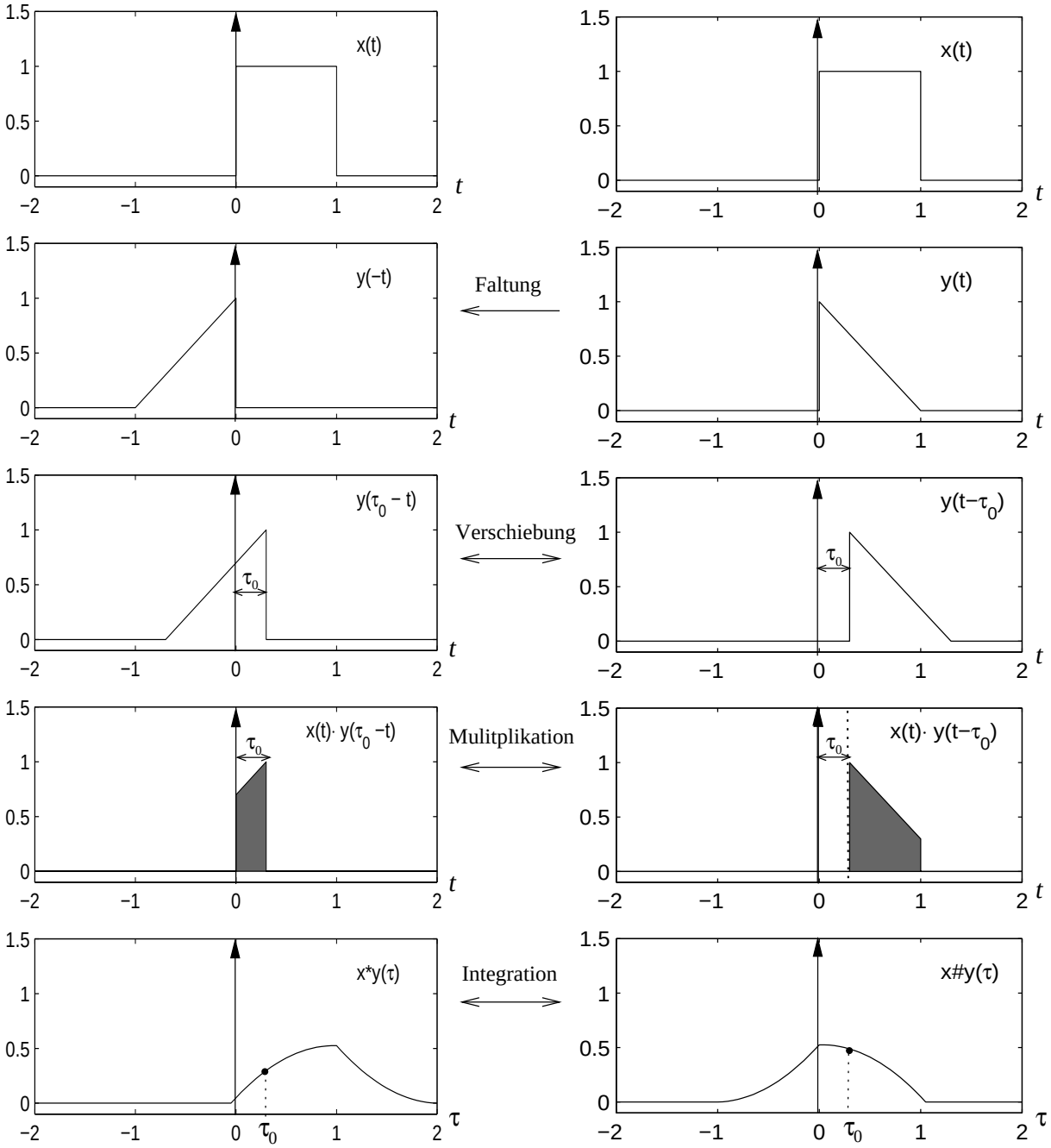
(k) 64^2 größte Koeffizienten; RMSE=5,97



(l) 16^2 größte Koeffizienten; RMSE=10,7

Graphische Interpretation von Faltung und Korrelation

Faltung $x * y(\tau)$ Korrelation $x\#y(\tau)$



Diskrete Faltung und Korrelation

Faltung $(x(n) * y(n))(r)$ Korrelation $(x(n)\#y(n))(r)$