Computergrafik 2: Übung 4

Transformation im Ortsraum,
Fourier Transformation

Quiz

- Fourier Transformation: Grundidee?
- Fouriers Theorem
- $F(u,v) = x + iy$
  - Bedeutung u,v?
  - Betrag?
  - Phase?
  - Eulersche Formel?
- Grundidee FFT?

Besprechung Übung 4

- Probleme?

DISKRETE FOURIER-TRANSFORMATION

Anschaulich: Basisvektoren eines Bildes

10	255	4	250	=	10 *	1	0	0	0
				+	255 *	0	1	0	0
				+	4 *	0	0	1	0
				+	250 *	0	0	0	1

bilden eine Basis des R^4
sind paarweise orthogonal
haben Länge 1

- Wahl anderer Basisvektoren ➔ Transformation mittels Basiswechsel
- Basiswechselmatrix vom Rang der Pixelanzahl

Basisfunktionen der Fourierbasis (N = 4)

- Basisfunktionen f_u $$f_u(n) = \exp\left(-\frac{i2\pi}{N}un\right)$$
- Wertetabelle:

$f_u(n)$	n=0	n=1	n=2	n=3
u=0	exp(0)=1	exp(0)=1	exp(0)=1	exp(0)=1
u=1	exp(0)=1	exp(-iπ/2)=-i	exp(-iπ)=-1	exp(-i3π/2)=i
u=2	exp(0)=1	exp(-iπ)=-1	exp(-i2π)=1	exp(-i3π)=-1
u=3	exp(0)=1	exp(-i3π/2)=i	exp(-i3π)=-1	exp(-i9π/2)=-i

- Basisvektoren: $f_{u=0} = (1,1,1,1)$, $f_{u=1} = (1,-i,-1,i)$, $f_{u=2} = (1,-1,1,-1)$, $f_{u=3} = (1,i,-1,-i)$
- Basiswechselmatrix:

$$B = \frac{1}{4}\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -i & -1 & i \\ 1 & -1 & 1 & -1 \\ 1 & i & -1 & -i \end{pmatrix}$$

Basisfunktionen der Fourierbasis (N = 4)

- Basisfunktionen f_u $$f_u(n) = \exp\left(\frac{i2\pi}{N}un\right)$$
- Wertetabelle:

$f_u(n)$	n=0	n=1	n=2	n=3
u=0	exp(0)=1	exp(0)=1	exp(0)=1	exp(0)=1
u=1	exp(0)=1	exp(iπ/2)=i	exp(iπ)=-1	exp(i3π/2)=-i
u=2	exp(0)=1	exp(iπ)=-1	exp(i2π)=1	exp(i3π)=-1
u=3	exp(0)=1	exp(i3π/2)=-i	exp(i3π)=-1	exp(i9π/2)=i

- Basisvektoren: $f_{u=0}$ = (1,1,1,1), $f_{u=1}$ = (1,i,-1,-i), $f_{u=2}$ = (1,-1,1,-1), $f_{u=3}$ = (1,-i,-1,i)
- Basiswechselmatrix: $$B = \frac{1}{4}\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & i & -1 & -i \\ 1 & -1 & 1 & -1 \\ 1 & -i & -1 & i \end{pmatrix}$$

1D-Basisfunktionen

Bildfunktion: $f(n)$, n=0,N-1,

1	0	1	0

also: N Basisfunktionen

$b_u(n) = \exp(i \cdot 2\pi / N \cdot n \cdot u)$, mit Frequenzen u=0,N-1

z.B. $b_0(n) = [(1,0),(1,0),\ldots,(1,0)]$

Transformation **FT** : **FT(f)** = **F** = **f** · **B** (Vektor-Matrix-Schreibweise)

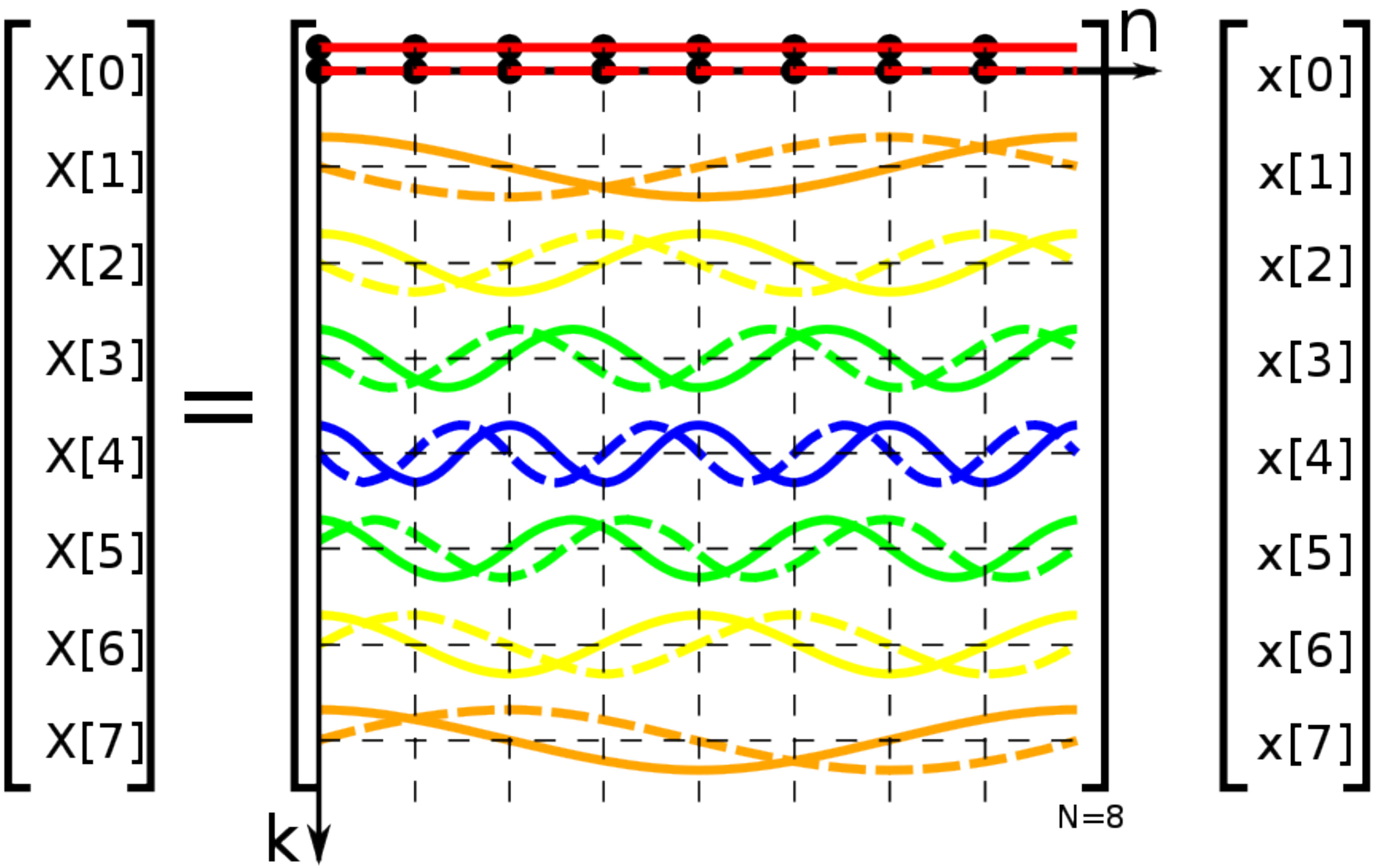
$F(u) = \sum_n f(n) \cdot \exp(-i \cdot 2\pi / N \cdot n \cdot u)$, für alle u=0,N-1

Rücktransformation $\mathbf{FT}^{-1}$: $\mathbf{FT}^{-1}(\mathbf{F}) = \mathbf{F} \cdot \mathbf{B}^T$ (Vektor-Matrix-Schreibweise)

$f(n) = 1/N \cdot \sum_u F(u) \cdot \exp(i \cdot 2\pi / N \cdot n \cdot u)$, für alle n=0,N-1

Skalierungsfaktor, weil die Basisfunktionen nicht normiert sind.

DFT als Matrixoperation



[Wikipedia.org, CC-BY-SA]

DFT als Matrixoperation

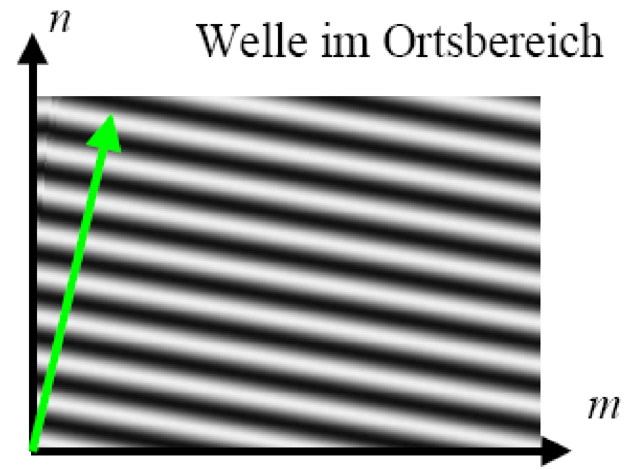
$$X = Wx$$

$$W \in N \times N$$

$$W = \frac{1}{\sqrt{N}} \begin{bmatrix} 1 & 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega & \omega^2 & \omega^3 & \cdots & \omega^{N-1} \\ 1 & \omega^2 & \omega^4 & \omega^6 & \cdots & \omega^{2(N-1)} \\ 1 & \omega^3 & \omega^6 & \omega^9 & \cdots & \omega^{3(N-1)} \\ \vdots & \vdots & \vdots & \vdots & & \vdots \\ 1 & \omega^{N-1} & \omega^{2(N-1)} & \omega^{3(N-1)} & \cdots & \omega^{(N-1)(N-1)} \end{bmatrix}$$

$$\omega = e^{-\frac{2\pi i}{N}}$$

2D-Basisfunktionen



Basisfunktionen sind Wellen (Frequenz, Richtung, Amplitude, Phase):

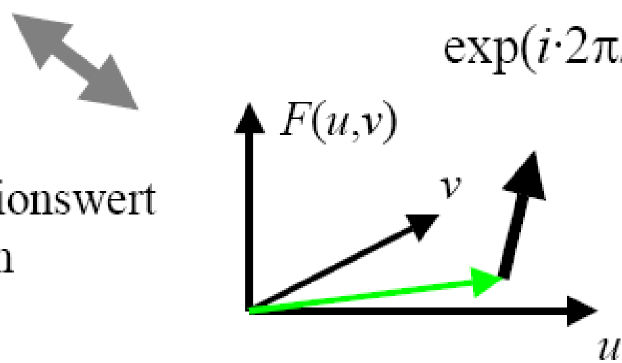
$$\exp(i \cdot 2\pi/N \cdot (mu+nv))$$

Richtung ist durch Vektor $(u\ v)$ gegeben.

Die Basisfunktionen der 2-D Fourier-transformation sind zerlegbar:

$$\exp(i \cdot 2\pi/N \cdot (mu+nv)) = \exp(i \cdot 2\pi/N \cdot m \cdot u) \cdot \exp(i \cdot 2\pi/N \cdot n \cdot v)$$

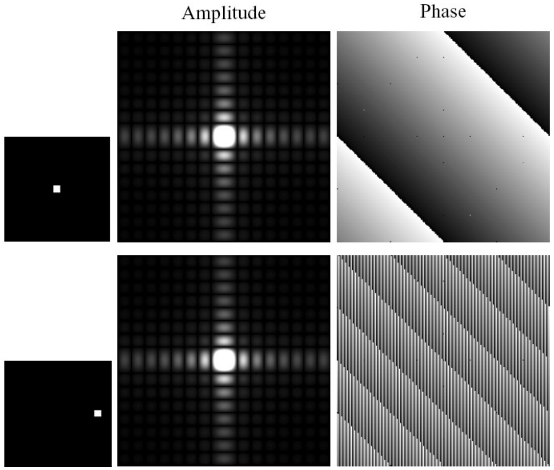
(komplexer) Funktionswert im Frequenzbereich



Frequenz: $f(u,v) = \sqrt{u^2 + v^2}$

Richtungsvektor: $r(u,v) = \dfrac{1}{f(u,v)} \begin{pmatrix} u \\ v \end{pmatrix}$

Eigenschaften der 2D-DFT: Translation



- 2D-DFT: $$F(u,v) = \sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x,y)\cdot e^{-i2\pi(ux/M+vy/N)}$$

$$f(x,y) = \frac{1}{MN}\sum_{u=0}^{M-1}\sum_{v=0}^{N-1} F(u,v)\cdot e^{i2\pi(ux/M+vy/N)}$$

- Verschiebung im Ortsraum führt zu Phasenverschiebung im Frequenzraum:

$$f(x-x_0, y-y_0) = \frac{1}{MN}\sum_{u=0}^{M-1}\sum_{v=0}^{N-1} F(u,v)\cdot e^{i2\pi(u(x-x_0)/M+v(y-y_0)/N)}$$

$$= \frac{1}{MN}\sum_{u=0}^{M-1}\sum_{v=0}^{N-1} F(u,v)\cdot e^{i2\pi(ux/M+vy/N)}\cdot e^{-i2\pi(ux_0/M+vy_0/N)}$$

$$= \frac{1}{MN}\sum_{u=0}^{M-1}\sum_{v=0}^{N-1} \left[F(u,v)\cdot e^{-i2\pi(ux_0/M+vy_0/N)}\right]\cdot e^{i2\pi(ux/M+vy/N)}$$

Separierbarkeit

$$F(u,v) = \frac{1}{N^2} \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} f(m,n) \exp\left[-i\frac{2\pi}{N}(um+vn)\right] =$$

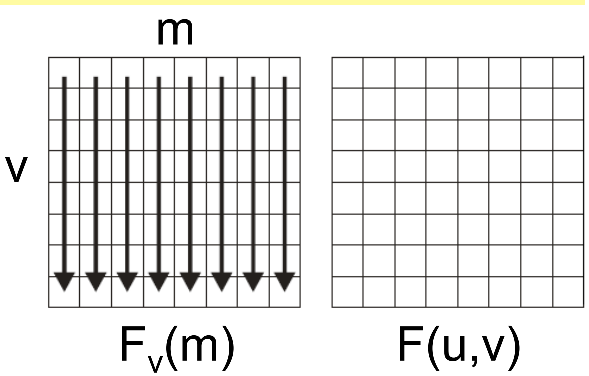
$$= \frac{1}{N^2} \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} f(m,n) \exp\left[-i\frac{2\pi}{N}um\right] \exp\left[-i\frac{2\pi}{N}vn\right] =$$

$$= \frac{1}{N} \sum_{m=0}^{N-1} \left(\frac{1}{N} \sum_{n=0}^{N-1} f(m,n) \exp\left[-i\frac{2\pi}{N}vn\right] \right) \exp\left[-i\frac{2\pi}{N}um\right] =$$

$$= \frac{1}{N} \sum_{m=0}^{N-1} \exp\left[-i\frac{2\pi}{N}um\right] \left(\frac{1}{N} \sum_{n=0}^{N-1} f(m,n) \exp\left[-i\frac{2\pi}{N}vn\right] \right) =$$

$$= \frac{1}{N} \sum_{m=0}^{N-1} \exp\left[-i\frac{2\pi}{N}um\right] F_v(m)$$

Vorgehensweise: zunächst $F_v(m)$ für alle (v,m) berechnen und dann verwenden.



FAST FOURIER TRANSFORM (FFT)

Vorgehensweise generell

- Vereinfachende Annahme: $N=2^k$, $k>1$
- Nutze Separierbarkeit, um 2D-FT auf 1D zurückzuführen ($O(N^4) \rightarrow O(N^3)$)
- Teile Summe in zwei Teilsummen auf
- Finde Gemeinsamkeiten in den Teilsummen und berechne beide Teilsummen miteinander
- Betrachte die Teilsumme und unterteile rekursiv bis $N=1$ ($O(N^3) \rightarrow O(N^2 \log N)$)

Divide Schritt

- Teile Summe in zwei Teilsummen auf

$$N = 2K, W_N = \exp\left(-i\frac{2\pi}{N}\right) = \exp\left(-i\frac{\pi}{K}\right), W_N^2 = \exp\left(-i\frac{2\pi}{K}\right) = W_K$$

$$F_N(u) = \frac{1}{N}\sum_{n=0}^{N-1} f(n)(W_N)^{un} = \frac{1}{2K}\sum_{n=0}^{2K-1} f(n)(W_{2K})^{un} =$$

$$= \frac{1}{2}\left(\frac{1}{K}\sum_{n=0}^{K-1} f(2n)(W_{2K})^{2nu} + \frac{1}{K}\sum_{n=0}^{K-1} f(2n+1)(W_{2K})^{(2n+1)u}\right)$$

- Finde Gemeinsamkeiten in den Teilsummen

$$F_{even,K}(u) = \frac{1}{K}\sum_{n=0}^{K-1} f(2n)(W_{2K})^{2nu} = \frac{1}{K}\sum_{n=0}^{K-1} f(2n)(W_K)^{nu}$$

$$F_{odd,K}(u) = \frac{1}{K}\sum_{n=0}^{K-1} f(2n+1)(W_{2K})^{2nu} = \frac{1}{K}\sum_{n=0}^{K-1} f(2n+1)(W_K)^{nu}$$

Ausnutzen der Periodizität

$$N = 2K, \quad W_N = \exp\left(-i\frac{2\pi}{N}\right) = \exp\left(-i\frac{\pi}{K}\right), \quad W_K = \exp\left(-i\frac{2\pi}{K}\right)$$

$$F_{even,K}(u) = \frac{1}{K}\sum_{n=0}^{K-1} f(2n)(W_K)^{nu}, \quad F_{odd,K}(u) = \frac{1}{K}\sum_{n=0}^{K-1} f(2n+1)(W_K)^{nu}$$

$$F_N(u) = \frac{1}{2}\left(F_{even,K}(u) + F_{odd,K}(u)(W_{2K})^u\right)$$

- Berechne F(u+K)

$$\left(W_K\right)^{u+N} = \left(W_K\right)^u, \quad \left(W_{2K}\right)^{u+K} = -\left(W_{2K}\right)^u$$

$$F_N\left(u+K\right) = \frac{1}{2}\left(F_{even,K}\left(u+K\right) + F_{odd,K}\left(u+K\right)\left(W_{2K}\right)^{u+K}\right)$$

$$F_N\left(u+K\right) = \frac{1}{2}\left(F_{even,K}\left(u\right) - F_{odd,K}\left(u\right)\left(W_{2K}\right)^u\right)$$

Ausnutzen der Periodizität

$$N = 2K, W_N = \exp\left(-i\frac{2\pi}{N}\right)$$

$$F(u) = \frac{1}{2}\left(F_{even}(u) + F_{odd}(u)(W_{2K})^u\right)$$

$$F(u+K) = \frac{1}{2}\left(F_{even}(u) - F_{odd}(u)(W_{2K})^u\right)$$

- Also kann man F(u+K) mithilfe F(u) berechnen (einmal $F_{even} + F_{odd}$, einmal $F_{even} - F_{odd}$)
- Betrachte die Teilsumme [0…K-1] und unterteile rekursiv bis K=1 ($O(n^3) \rightarrow O(n^2 \log n)$)

Zusammenfassung

$$N = 2K, W_N = \exp\left(-i\frac{2\pi}{N}\right)$$

$$F_N(u) = \frac{1}{N}\sum_{n=0}^{N-1} f(n)(W_N)^{nu} = \frac{1}{2}\left(F_{even,K}(u) + F_{odd,K}(u)(W_{2K})^u\right)$$

$$F_{even,K}(u) = \frac{1}{K}\sum_{n=0}^{K-1} f(2n)(W_K)^{nu}$$

$$F_{odd,K}(u) = \frac{1}{K}\sum_{n=0}^{K-1} f(2n+1)(W_K)^{nu}$$

$$F_N(u+K) = \frac{1}{2}\left(F_{even,K}(u) - F_{odd,K}(u)(W_{2K})^u\right)$$

Rekursiver Algorithmus

$$F_N(u) = \frac{1}{N}\sum_{n=0}^{N-1} f(n)(W_N)^{nu} = \frac{1}{2}\left(F_{even,K}(u) + F_{odd,K}(u)(W_{2K})^u\right)$$

$$F_{even,K}(u) = \frac{1}{K}\sum_{n=0}^{K-1} f(2n)(W_K)^{nu}$$

$$F_{odd,K}(u) = \frac{1}{K}\sum_{n=0}^{K-1} f(2n+1)(W_K)^{nu}$$

$$F_N(u+K) = \frac{1}{2}\left(F_{even,K}(u) - F_{odd,K}(u)(W_{2K})^u\right)$$

def $F_{rek}(N, (f_0, f_1, f_2, \ldots, f_{N-1}))$:

  if N == 1 then return (f_0)

  K = N/2

  $F_{even} = F_{rek}(K, (f_0, f_2, f_4, \ldots, f_{N-2}))$

  $F_{odd} = F_{rek}(K, (f_1, f_3, f_5, \ldots, f_{N-1}))$

  F = zeros(N)

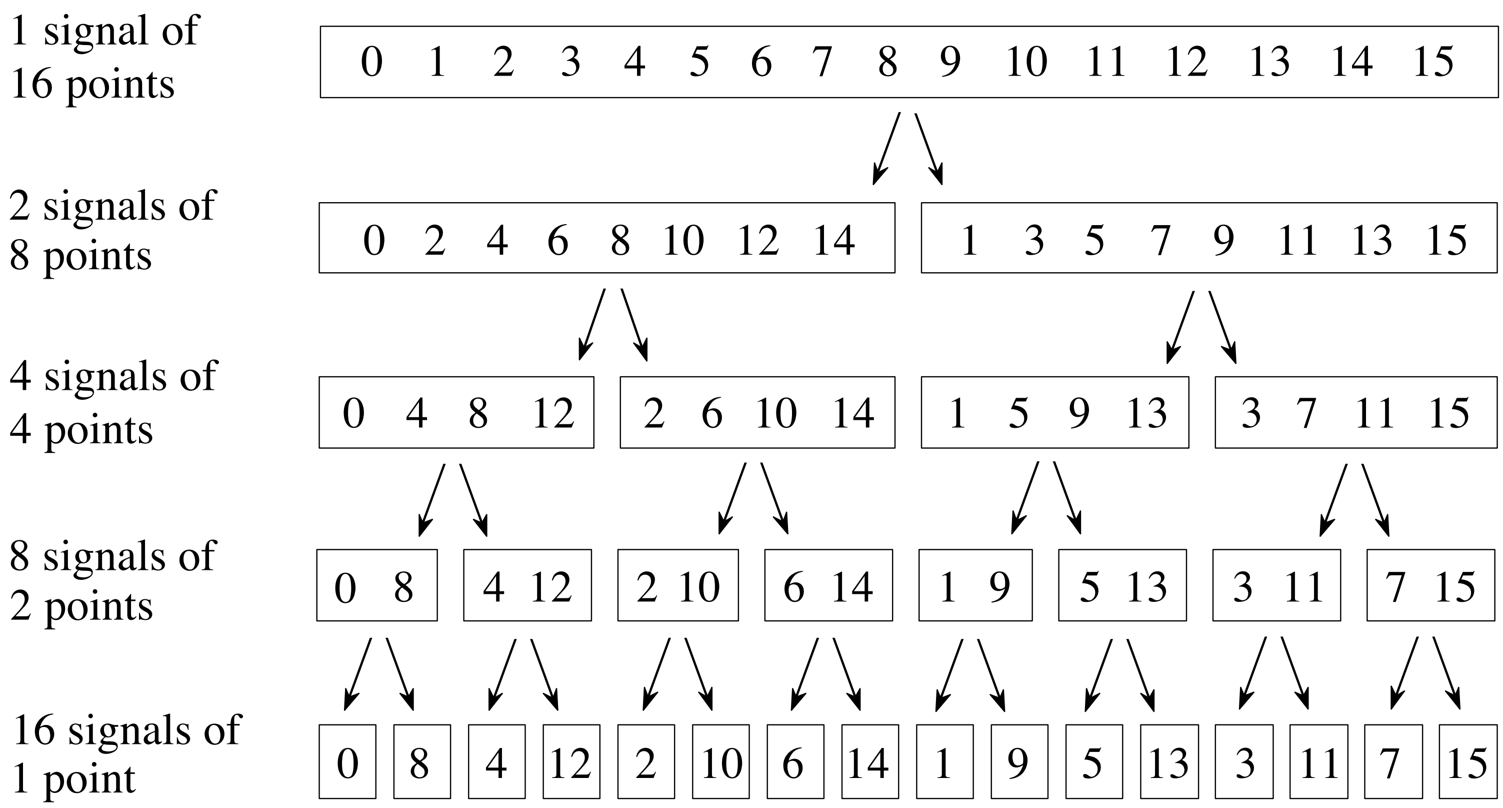
  for u = 0..K-1:

    $F[u] = 0.5 * (F_{even}[u] + F_{odd}[u] * W_N^u)$

    $F[u+K] = 0.5 * (F_{even}[u] – F_{odd}[u] * W_N^u)$

  return F

Rekursive Aufteilung in F_{odd} und F_{even}



- Signal aus N Datenpunkten in N Signale mit je einem Datenpunkt umwandeln
- Frequenzspektra der N Zeitraumsignale berechnen
- N Spektra zu einem einzelnen Spektrum zusammenfassen

[Abbildungen zu FFT aus: *The Scientist and Engineer's Guide to Digital Signal Processing*, http://www.dspguide.com/CH12.PDF]

Bit-Inverse Sortierung

→ iterativer Algorithmus bottom-up

Sample numbers in normal order	
Decimal	*Binary*
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
10	1010
11	1011
12	1100
13	1101
14	1110
15	1111

⇒

Sample numbers after bit reversal	
Decimal	*Binary*
0	0000
8	1000
4	0100
12	1100
2	0010
10	1010
6	0100
14	1110
1	0001
9	1001
5	0101
13	1101
3	0011
11	1011
7	0111
15	1111

[Abbildungen zu FFT aus: *The Scientist and Engineer's Guide to Digital Signal Processing*, http://www.dspguide.com/CH12.PDF]

Python-Hinweise

- Komplexe Zahlen

```
N = 256
B = np.zeros((N, N), dtype='complex')
for v in xrange(N):
    for n in xrange(N):
        B[v,n] = np.exp(-1j*2*np.pi*v*n/N)
```

Array komplexer Zahlen

imaginäre Einheit: 1j

3.1415...: np.pi

- Amplitudenbild

```
plt.subplot(232)
plt.imshow(np.log(np.abs(F)))
```

np.abs(F) = ||F||

- Skalarprodukt

```
s = np.dot(a, b)
```

Python-Hinweise

- Realteil, Imaginärteil

  ```
  re = np.real(f), im = np.imag(f)
  ```

- 2D-FFT in NumPy

  ```
  F = np.fft.fft2(f)
  ```

- Koordinatenursprung zentrieren

  ```
  F2 = np.fft.fftshift(F)
  ```

- Inverse 2D-FFT

  ```
  g = np.fft.ifft2(F)
  ```