Algorithmen zur effizienten Schätzung der Kovarianzfunktion von Binärbildern

1 Definition der Kovarianz

- Ξ bezeichne eine zufällige abgeschlossene Menge (RACS) in $\mathbb{R}^2$, die stationär und isotrop sei.
- Die *Kovarianz* $C(h)\,, h \in \mathbb{R}^2$, von Ξ ist dann gegeben durch

$$C(h) = \mathbb{P}(o \in \Xi, h \in \Xi)\,. \tag{1}$$

- Bei gegebenem Beobachtungsfenster W folgt wegen der Stationarität von Ξ

$$C(h) \;=\; \frac{\mathbb{E}|\Xi \cap (\Xi + h) \cap W \cap (W+h)|}{|W \cap (W+h)|}\,. \tag{2}$$

- Per Definition ist durch den folgenden Ausdruck ein erwartungstreuer Schätzer für $C(h)$ gegeben:

$$\widehat{C}(h) \;=\; \frac{|\Xi \cap (\Xi + h) \cap W \cap (W+h)|}{|W \cap (W+h)|}\,. \tag{3}$$

- Formel (2) lässt sich mit Hilfe der Stationarität von Ξ nachweisen:

$$
\begin{aligned}
C(h) &= \mathbb{P}(o \in \Xi, h \in \Xi) \cdot \frac{|W \cap (W+h)|}{|W \cap (W+h)|} \\
&= \frac{\int_{W \cap (W+h)} \mathbb{P}(o \in \Xi, h \in \Xi)\, dx}{|W \cap (W+h)|} = \frac{\int_{W \cap (W+h)} \mathbb{P}(x \in \Xi, (x-h) \in \Xi)\, dx}{|W \cap (W+h)|} \\
&= \frac{\int_{W \cap (W+h)} \mathbb{P}(x \in \Xi, x \in (\Xi+h))\, dx}{|W \cap (W+h)|} = \frac{\int_{W \cap (W+h)} \mathbb{E}\, \mathbf{1}_{\Xi \cap (\Xi+h)}(x)\, dx}{|W \cap (W+h)|} \\
&= \frac{\mathbb{E}|\Xi \cap (\Xi+h) \cap W \cap (W+h)|}{|W \cap (W+h)|}
\end{aligned}
$$

$\square$

2 Beispiel für den Fall $d = 1$

- Gegeben sei ein Bildvektor mit Binärwerten der Länge n

  $0 \quad 1 \quad 1 \quad 0 \quad 1 \quad 1 \quad 1 \quad 0 \quad \dots$

- Für $n = 4$ beispielsweise werde der Bildvektor $\big(a_0, a_1, a_2, a_3\big)$ betrachtet.

  Der Berechnung der ,,diskretisierten" Kovarianz für einen bestimmten Vektor $h \in \mathbb{R}$ entspricht nun ein Skalarprodukt des Bildvektors mit dem um den Vektor h verschobenen Bildvektor (*Verschiebungsoperation*).

$$\begin{array}{c c c c c} & \boxed{\begin{array}{c} a_0 \\ a_1 \end{array}} & \boxed{\begin{array}{c} a_1 \\ a_2 \end{array}} & \boxed{\begin{array}{c} a_2 \\ a_3 \end{array}} & a_3 \\ a_0 & & & & \end{array}$$

  Beispielsweise ergibt sich für $h = 1$ der folgende Schätzer für die Kovarianz (vgl. Formel (3)):

$$\widehat{C}(1) = \frac{a_1 a_0 + a_2 a_1 + a_3 a_2}{3} \, .$$

- Alternativ kann zur Berechnung der Kovarianz auch die Multiplikation zweier Polynome $A(x)$ und $B(x)$ vom Grade $n-1$ betrachtet werden:

$$A(x) = \sum_{j=0}^{n-1} a_j x^j \,, \qquad B(x) = \sum_{j=0}^{n-1} b_j x^j \,,$$

  wobei für die Koeffizientenvektoren gilt:
  $b = \breve{a}$, d.h., $\big(b_0, \ldots, b_{n-1}\big) = \big(a_{n-1}, \ldots, a_0\big)$.

- Das Produktpolynom $C(x) = A(x) \cdot B(x)$ ergibt sich dann zu

$$C(x) = \sum_{i=0}^{2n-2} c_i x^i \,, \qquad \text{wobei} \quad c_i = \sum_{k=0}^{i} a_k b_{i-k} \,.$$

  Bemerkung: Diese letzte Operation, mittels der die Koeffizienten c_i bestimmt werden, nennt man auch oft *Faltung*. Der dazugehörige Koeffizientenvektor besitzt die Länge $2n-1$ und ist hier symmetrisch bzgl. der Stelle $n-1$.

- Für das obige Beispiel ($n = 4$) ergibt sich mit $\big(b_0, b_1, b_2, b_3\big) = \big(a_3, a_2, a_1, a_0\big)$:

$$\begin{aligned} C(x) &= a_3 b_3 x^6 + \big(a_3 b_2 + a_2 b_3\big) x^5 + \big(a_3 b_1 + a_2 b_2 + a_1 b_3\big) x^4 \\ &\quad + \big(a_3 b_0 + a_2 b_1 + a_1 b_2 + a_0 b_3\big) x^3 + \big(a_2 b_0 + a_1 b_1 + a_0 b_2\big) x^2 \\ &\quad + \big(a_1 b_0 + a_0 b_1\big) x + a_0 b_0 \end{aligned}$$

  Dann lässt sich die Kovarianz für $h = 1$ mit Hilfe des zum Term der Ordnung 4 gehörenden Koeffizienten von $C(x)$ berechnen (*Faltungsoperation*).

- **Allgemein gilt:** Bei einem Bildvektor der Länge n ist ein Schätzer für die ,,diskretisierte'' Kovarianz für $h = l$ gegeben durch

$$\widehat{C}(l) = \frac{c_{n-1+l}}{n-l}\,, \quad l = -n+1, \ldots, n-1,$$

  wobei mit c_{n-1+l} der zum Term der Ordnung $(n - 1 + l)$ gehörende Koeffizient des Produktpolynoms bezeichnet wird.

3 Beispiel für den Fall $d = 2$

- Es seien nun zwei $(m \times n)$-Bildmatrizen $\mathbf{A}$ und $\mathbf{B}$ gegeben, mit $\check{\mathbf{B}} = \mathbf{A}$, d.h., die Werte der Matrix $\mathbf{B}$ gleichen nach einer Spiegelung an der Hauptdiagonalen der Matrix denen der Matrix $\mathbf{A}$. Die Berechnung der ,,diskretisierten'' Kovarianz für einen Vektor $h \in \mathbb{R}^2$ kann analog zum Fall $d = 1$ mit Hilfe der *Verschiebungs*- oder der *Faltungsoperation* erfolgen.

- Zur Illustration der *Verschiebungsoperation* werden die folgenden (4×6) Matrizen $\mathbf{A}$ und $\check{\mathbf{B}}$ betrachtet:

$$\mathbf{A} = \begin{pmatrix} a_{30} & a_{31} & a_{32} & a_{33} & a_{34} & a_{35} \\ a_{20} & a_{21} & a_{22} & a_{23} & a_{24} & a_{25} \\ a_{10} & a_{11} & a_{12} & a_{13} & a_{14} & a_{15} \\ a_{00} & a_{01} & a_{02} & a_{03} & a_{04} & a_{05} \end{pmatrix} \qquad \check{\mathbf{B}} = \begin{pmatrix} b_{05} & b_{04} & b_{03} & b_{02} & b_{01} & b_{00} \\ b_{15} & b_{14} & b_{13} & b_{12} & b_{11} & b_{10} \\ b_{25} & b_{24} & b_{23} & b_{22} & b_{21} & b_{20} \\ b_{35} & b_{34} & b_{33} & b_{32} & b_{31} & b_{30} \end{pmatrix}$$

So ergibt sich z.B. für $h = \begin{pmatrix} 3 \\ 2 \end{pmatrix}$ der folgende Schätzer für die Kovarianz:

$$\widehat{C}(\begin{pmatrix} 3 \\ 2 \end{pmatrix}) = \frac{a_{23}b_{35} + a_{24}b_{34} + a_{25}b_{33} + a_{33}b_{25} + a_{34}b_{24} + a_{35}b_{23}}{6} .$$

- Zur Illustration der *Faltungsoperation* werden nun die Einträge der Matrizen $\mathbf{A}$ und $\mathbf{B}$ als Koeffizienten von Polynomen in x und y verstanden, d.h.,

$$A(x,y) = \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} a_{ij} x^i y^j \,, \qquad B(x,y) = \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} b_{ij} x^i y^j \,.$$

  So geht in die Berechnung der Kovarianz der entsprechende Koeffizient des Produktpolynoms von $A(x,y)$ und $B(x,y)$ ein. Für obiges Beispiel folgt deshalb

$$\widehat{C}(\begin{pmatrix} 3 \\ 2 \end{pmatrix}) = \frac{a_{23}b_{35} + a_{24}b_{34} + a_{25}b_{33} + a_{33}b_{25} + a_{34}b_{24} + a_{35}b_{23}}{6} = \frac{c_{58}}{6} \,,$$

  wobei mit c_{58} der zum Term $x^5 y^8$ gehörende Koeffizient bezeichnet wird.

- **Allgemein gilt:** Bei einer $(m \times n)$-Bildmatrix ist ein Schätzer für die ,,diskretisierte" Kovarianz für $h = \begin{pmatrix} l \\ k \end{pmatrix}$ gegeben durch

$$\widehat{C}(\begin{pmatrix} l \\ k \end{pmatrix}) = \frac{c_{m-1+k,n-1+l}}{(m-k)(n-l)} \,, \quad k = -m+1, \ldots, m-1, \; l = -n+1, \ldots, n-1,$$

  wobei mit $c_{m-1+k,n-1+l}$ der zum Term der Ordnung $(m-1+k, n-1+l)$ gehörende Koeffizient des Produktpolynoms bezeichnet wird.

4 Bestimmung der Kovarianz mittels Verschiebungs- bzw. Faltungsoperation

- Sei Ξ eine stationäre und isotrope RACS in $\mathbb{R}^2$. Dann lässt sich Ausdruck (2) mit Hilfe der *Set-Kovarianz* $c_W(h) = |W \cap (W+h)|$, $h \in \mathbb{R}^2$, umschreiben zu

$$\begin{aligned} C(h)\, c_W(h) &= \mathbb{E}|\Xi \cap (\Xi + h) \cap W \cap (W+h)| \\ &= \mathbb{E} \int_{W \cap (W+h)} \mathbf{1}_\Xi(x) \mathbf{1}_{\Xi+h}(x)\, dx \\ &= \mathbb{E} \int_{W \cap (W+h)} \mathbf{1}_\Xi(x) \mathbf{1}_\Xi(x-h)\, dx\,. \end{aligned}$$

Weiterhin ergibt sich mit $f(x) := \mathbf{1}_W(x)\mathbf{1}_\Xi(x)$ und $g(x) := f(-x)$

$$\begin{aligned} C(h)\, c_W(h) &= \mathbb{E} \int_{\mathbb{R}^2} f(x)\, f(x-h)\, dx && \textit{(Verschiebungsoperation)} \\ &= \mathbb{E} \int_{\mathbb{R}^2} f(x)\, g(h-x)\, dx && \textit{(Faltungsoperation)} \\ &= \mathbb{E}\, f * g(h)\,, \end{aligned}$$

wobei mit „$*$" der Faltungsoperator bezeichnet wird.

5 Schaubild zur Polynommultiplikation

$A(x), B(x)$ in
Koeffizientendarstellung:
$(a_0, \ldots, a_{n-1})$,
$(b_0, \ldots, b_{n-1})$

Multiplikation
mittels Faltung
Komplexität $O(n^2)$

$C(x)$ in
Koeffizientendarstellung:
$(c_0, \ldots, c_{2n-2})$

An $2n-1$ Punkten
$x_0, \ldots, x_{2n-2}$
evaluieren
(DFT)

Interpolation
(inverse DFT)

$(x_0, A(x_0), B(x_0))$
$\vdots$
$(x_{2n-2}, A(x_{2n-2}), B(x_{2n-2}))$
Punkt-Wert-Darstellung

$2n-1$ Werte
multiplizieren
Komplexität $O(n)$

$(x_0, A(x_0) \cdot B(x_0))$
$\vdots$
$(x_{2n-2}, A(x_{2n-2}) \cdot B(x_{2n-2}))$
Punkt-Wert-Darstellung

- Wenn bei der Durchführung der Polynommultiplikation die beiden Polynome an beliebigen $2n-1$ Stützstellen z.B. mittels *Hornerschema* evaluiert werden und die Interpolation nach *Newton* erfolgt, so ergibt jede dieser Operationen eine Komplexität von $O(n^2)$. Man erzielt also keinen Effizienzgewinn gegenüber der direkten Methode mittels Faltung.

- Durch geschickte Wahl der Stützstellen kann das Evaluieren (ebenso wie das Interpolieren) jedoch effizienter als mit Komplexität $O(n^2)$ erfolgen.

- Die *diskrete Fourier-Transformation* (DFT) besteht darin, ein Polynom, das durch n Koeffizienten $(a_0, \dots, a_{n-1})$ gegeben ist, an genau n Stützstellen, nämlich den sogenannten n-ten Einheitswurzeln, auszuwerten. In der obigen Situation hat das Polynom n Koeffizienten und soll an $2n-1$ Stellen evaluiert werden. Folglich muss die Koeffizientendarstellung des Polynoms mit $n-1$ Nullen erweitert werden.

- Die *Fast Fourier-Transformation* (FFT) stellt einen schnellen Algorithmus zur Berechnung der DFT dar, nämlich mit Komplexität $O(n \log n)$.

- Die Polynommultiplikation mit Hilfe der Fourier-Transformation wird im sogenannten *Faltungssatz* statuiert (siehe nächsten Abschnitt).

6 Fourier-Transformierte und Faltungssatz im stetigen Modell

- Die *Fourier-Transformierte* der Funktion $h(x) : \mathbb{R} \to \mathbb{C}$ ist gegeben durch

$$H(s) = \mathcal{F}\{h\}(s) = \int_{-\infty}^{\infty} h(x) \exp(2\pi i s x)\, dx\,,$$

bzw. die *inverse Fourier-Transformierte* der Funktion $H(s) : \mathbb{R} \to \mathbb{C}$ durch

$$h(x) = \mathcal{F}^{-1}\{H\}(x) = \int_{-\infty}^{\infty} H(s) \exp(-2\pi i x s)\, ds\,.$$

- Die *Fourier-Transformierte* der Funktion $h(x,y) : \mathbb{R}^2 \to \mathbb{C}$ ist gegeben durch

$$H(s,t) = \mathcal{F}\{h\}(s,t) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} h(x,y) \exp(2\pi i(sx+ty))\, dx\, dy\,,$$

bzw. die *inverse Fourier-Transformierte* der Funktion $H(s,t) : \mathbb{R}^2 \to \mathbb{C}$ durch

$$h(x,y) = \mathcal{F}^{-1}\{H\}(x,y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} H(s,t) \exp(-2\pi i(xs+yt))\, ds\, dt\,.$$

- **Bemerkung:** Die Fourier-Transformation $\mathcal{F}\{h\}(s,t)$ kann auch als Superposition von zwei eindimensionalen Fourier-Transformationen aufgefasst werden:

$$\mathcal{F}\{h\}(s,t) = \int_{-\infty}^{\infty} \underbrace{\left[\int_{-\infty}^{\infty} h(x,y) \exp(2\pi isx)\, dx\right]}_{H_y(s)=\mathcal{F}\{h_y\}(s)=H'_s(y)} \exp(2\pi ity)\, dy = \mathcal{F}\{\mathcal{F}\{h_y\}(s)\}(t)$$

- **Faltungssatz:** Seien G, $H : \mathbb{R} \to \mathbb{C}$ die Fourier-Transformierten der Funktionen g, $h : \mathbb{R} \to \mathbb{R}$. Dann gilt für die Faltung der Funktionen g und h

$$g * h(u) = \mathcal{F}^{-1}\{G \cdot H\}(u)\,.$$

Beweis:

$$\begin{aligned} \mathcal{F}\{g * h(u)\}(s) &= \mathcal{F}\{\int_{-\infty}^{\infty} g(x)h(u-x)\, dx\}(s) \\ &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} g(x)h(u-x)\, dx\ \exp(2\pi isu)\, du \\ &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} g(x)h(w)\ \exp(2\pi is(x+w))\, dx\, dw \\ &= \mathcal{F}\{g\}(s)\cdot\mathcal{F}\{h\}(s) = G(s)\cdot H(s)\,, \qquad \forall\, u \in \mathbb{R}\,. \end{aligned}$$

7 Diskrete Fourier-Transformierte

- **Nyquist-Frequenz Theorem:** Wird eine stetige Funktion $h(x) : \mathbb{R} \to \mathbb{C}$ an diskreten Punkten im Abstand Δ gemessen und gehen diese Punkte in die Bestimmung der zugehörigen Fourier-Transformierten $H(s)$ ein, so ist $H(s)$ nur im Intervall $\left[-\frac{1}{2\Delta}, \frac{1}{2\Delta}\right]$ berechenbar, d.h., die sogenannte *Nyquist-Frequenz* $\frac{1}{2\Delta}$ stellt eine Obergrenze der mittels Fourier-Transformation noch erfassbaren Frequenzen dar (ohne Beweis).

- Messung von $h(x)$ an n Punkten (n gerade) ergibt:

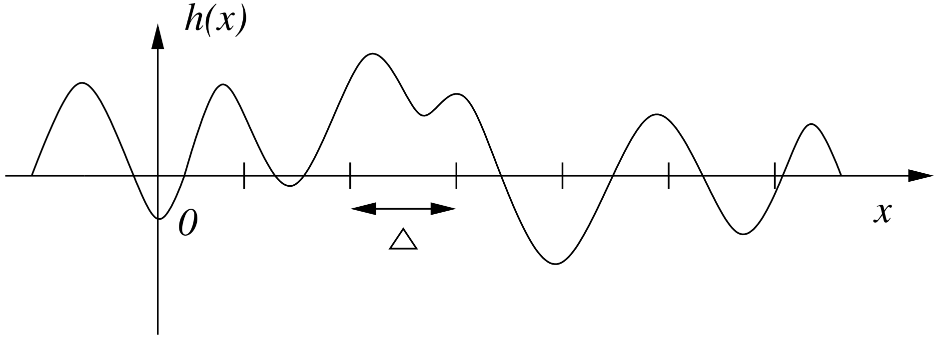
$$h_k = h(x_k)\,, \quad x_k = k\Delta\,, \quad k = 0, 1, \ldots, n-1\,.$$

- Schätzung der Funktion $H(s)$ erfolgt im Frequenzbereich mit Hilfe der folgenden Stützstellen s_u :

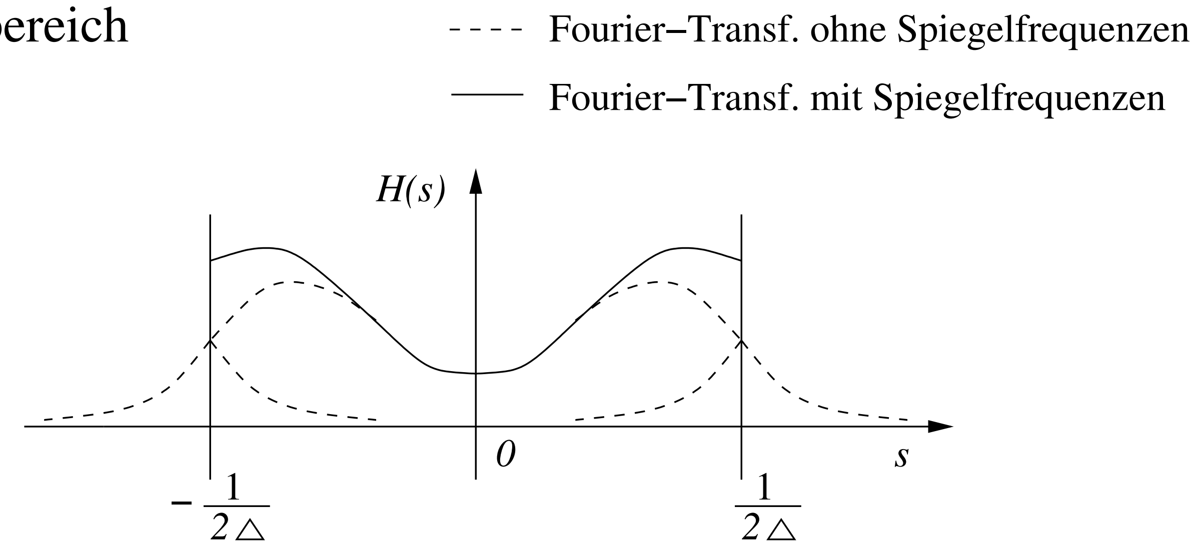
$$s_u = \frac{u}{n\Delta}\,, \quad u = -\frac{n}{2}, \ldots, +\frac{n}{2}\,.$$

  Bemerkung: Die n Messwerte der Funktion $h(x)$ dienen folglich der Schätzung der zugehörigen Fourier-Transformierten an n verschiedenen Stützstellen.

Zeitbereich



Frequenzbereich



- Die Schätzung der Fourier-Transformierten erfolgt mittels Integralapproximation:

$$
\begin{aligned}
H(s_u) &= \int_{-\infty}^{\infty} h(x) \exp(2\pi i\, x\, s_u)\, dx \\
&\approx \sum_{k=0}^{n-1} h_k \exp(2\pi i\, x_k\, s_u) \cdot \Delta \\
&= \sum_{k=0}^{n-1} h_k \exp(2\pi i\, k\, u/n) \cdot \Delta
\end{aligned}
$$

- **Definition:** Zu einem gegebenen Vektor $h = (h_0, \ldots, h_{n-1})$ bezeichnet der Vektor $H = (H_0, \ldots, H_{n-1})$ mit

$$
H_u = \sum_{k=0}^{n-1} h_k \exp(2\pi i\, k\, u/n)
$$

die *diskrete Fourier-Transformierte* (DFT) von h. Die Indizierung des Vektors H kann dabei von $u = 0, 1, \ldots, n-1$ laufen, da H_u periodisch in u mit Periode n ist, d.h., $H_u = H_{u+n}$.

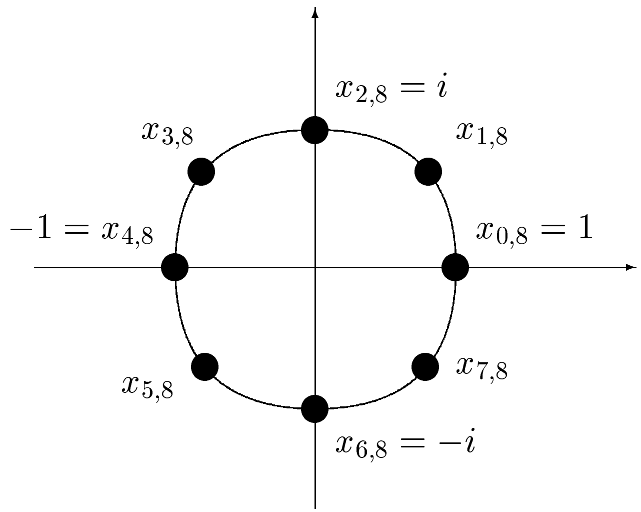
- **Bemerkung:** Die DFT bildet also einen n-dimensionalen (komplexen) Vektor auf einen n-dimensionalen komplexen Vektor ab.

- Sei nun ein Koeffizientenvektor $a = (a_0, \ldots, a_{n-1})$, welcher das Polynom $A(x)$ repräsentiert, gegeben. Dann bedeutet die Berechnung der DFT von a das Evaluieren von $A(x)$ an den sogenannten n-ten Einheitswurzeln.

- Ein komplexe Zahl x ist n-te Einheitswurzel, falls $x^n = 1$. Es gibt im Komplexen genau n Lösungen der Gleichung $x^n = 1$, nämlich $x_{0,n}, \ldots, x_{n-1,n}$ mit

$$x_{u,n} = e^{iu2\pi/n} \ .$$

Veranschaulichung der
n-ten Einheitswurzeln ($n = 8$):



- Mit der *Eulerschen Formel* kann überprüft werden, dass $x_{u,n}$ tatsächlich eine n-te Einheitswurzel ist:

$$(x_{u,n})^n = (e^{iu2\pi/n})^n = e^{iu2\pi} = \cos(u2\pi) + i\sin(u2\pi) = 1 + i \cdot 0 = 1$$

- Außerdem ergibt sich, dass man alle n-ten Einheitswurzeln als Potenzen von $x_{1,n}$ erhalten kann. Sei $z := x_{1,n} = e^{i2\pi/n}$, dann gilt:

$$(x_{0,n}, x_{1,n}, \ldots, x_{n-1,n}) = (z^0, z^1, \ldots, z^{n-1})$$

- Im Folgenden sei n eine Zweierpotenz. Dann erhält man durch Quadrieren der n-ten Einheitswurzeln gerade die $(n/2)$-ten Einheitswurzeln (und zwar zweimal). Sei im Folgenden $u = 0, \ldots, (n/2) - 1$:

$$\begin{aligned}
(x_{u,n})^2 &= e^{iu2\pi/(n/2)} = x_{u,n/2}\,, \\
(x_{(n/2)+u,n})^2 &= e^{i((n/2)+u)2\pi/(n/2)} \\
&= e^{i2\pi} \cdot e^{iu2\pi/(n/2)} = e^{iu2\pi/(n/2)} = x_{u,n/2}\,.
\end{aligned}$$

8 Fast Fourier-Transformation

- Im Folgenden werde das Polynom $A(x)$ vom Grade $n-1$ betrachtet, das durch den Koeffizientenvektor $a = (a_0, \ldots, a_{n-1})$ gegeben ist.
  Es wird sich als nützlich erweisen, das Polynom $A(x)$ in zwei kleinere Polynome mit jeweils $n/2$ vielen Koeffizienten aufzuspalten:

$$\begin{aligned} A^{\text{even}}(x) &:= \sum_{k=0}^{(n/2)-1} a_{2k} x^k \\ A^{\text{odd}}(x) &:= \sum_{k=0}^{(n/2)-1} a_{2k+1} x^k \end{aligned}$$

  D.h., die Polynome $A^{\text{even}}(x)$ bzw. $A^{\text{odd}}(x)$ werden repräsentiert durch die Koeffizientenvektoren
  $a^{\text{even}} = (a_0, a_2, a_4, \ldots, a_{n-2})$ bzw. $a^{\text{odd}} = (a_1, a_3, a_5, \ldots, a_{n-1})$.
  Es gilt die Beziehung:

$$A(x) = A^{\text{even}}(x^2) + x \cdot A^{\text{odd}}(x^2)\,.$$

- In dieser Formel und in der obigen Beobachtung, dass beim Quadrieren der n-ten Einheitswurzeln gerade die $(n/2)$-ten Einheitswurzeln entstehen, liegt der Schlüssel für den *FFT-Algorithmus* (Fast Fourier Transform). D.h., der FFT-Algorithmus ist ein schneller Algorithmus zur Berechnung der DFT.

- Gegeben sei also der Koeffizientenvektor a, der zerlegt die beiden Vektoren a^{even} und a^{odd} ergibt, jeweils der Länge $n/2$.

  Rekursive Anwendung des FFT-Algorithmus auf a^{even} und a^{odd} ergibt die diskreten Fourier-Transformierten

$$\begin{aligned} H^{\text{even}} &= (H_0^{\text{even}}, H_1^{\text{even}}, \ldots, H_{(n/2)-1}^{\text{even}}) \\ &= (A^{\text{even}}(x_{0,n/2}), \ldots, A^{\text{even}}(x_{(n/2)-1,n/2})) \end{aligned}$$

und

$$\begin{aligned} H^{\text{odd}} &= (H_0^{\text{odd}}, H_1^{\text{odd}}, \ldots, H_{(n/2)-1}^{\text{odd}}) \\ &= (A^{\text{odd}}(x_{0,n/2}), \ldots, A^{\text{odd}}(x_{(n/2)-1,n/2}))\,. \end{aligned}$$

- Aus der obigen Formel lassen sich nun die gesuchten Werte

$$(H_0, \ldots, H_{n-1}) = (A(x_{0,n}), \ldots, A(x_{n-1,n}))$$

rekonstruieren. Für $u = 0, 1, \ldots, (n/2) - 1$ gilt:

$$\begin{aligned}
H_u &= A(x_{u,n}) \\
&= A^{\mathsf{even}}((x_{u,n})^2) + x_{u,n} \cdot A^{\mathsf{odd}}((x_{u,n})^2) \\
&= A^{\mathsf{even}}(x_{u,n/2}) + x_{u,n} \cdot A^{\mathsf{odd}}(x_{u,n/2}) \\
&= H_u^{\mathsf{even}} + x_{u,n} \cdot H_u^{\mathsf{odd}}
\end{aligned}$$

und entsprechend

$$\begin{aligned}
H_{(n/2)+u} &= A(x_{(n/2)+u,n}) \\
&= A^{\mathsf{even}}((x_{(n/2)+u,n})^2) + x_{(n/2)+u,n} \cdot A^{\mathsf{odd}}((x_{(n/2)+u,n})^2) \\
&= A^{\mathsf{even}}(x_{u,n/2}) + x_{(n/2)+u,n} \cdot A^{\mathsf{odd}}(x_{u,n/2}) \\
&= H_u^{\mathsf{even}} - x_{u,n} \cdot H_u^{\mathsf{odd}} \,.
\end{aligned}$$

- **Programm zum FFT-Algorithmus:** Die Komplexität dieses Algorithmus ergibt sich aus der Rekursionsgleichung $T(n) = 2\ T(n/2) + O(n)$, deren Lösung nach dem Master-Theorem $T(n) = O(n \log n)$ ist.

PROCEDURE *FFT*$(n, (a_0, \ldots, a_{n-1}))$
IF $n = 1$ THEN RETURN (a_0)
  ELSE
    $a^{\text{even}} := (a_0, a_2, \ldots, a_{n-2})$
    $a^{\text{odd}} := (a_1, a_3, \ldots, a_{n-1})$
    H^{even} :=*FFT*$(n/2, a^{\text{even}})$
    H^{odd} :=*FFT*$(n/2, a^{\text{odd}})$
    $x := 1$
    $z := e^{i2\pi/n}$
    FOR $u := 0$ TO $(n/2) - 1$ DO
      $H_u := H_u^{\text{even}} + x * H_u^{\text{odd}}$
      $H_{(n/2)+u} := H_u^{\text{even}} - x * H_u^{\text{odd}}$
      $x := x * z$
    RETURN $(H_0, H_1, \ldots, H_{n-1})$

- Die *inverse DFT* unterscheidet sich von der DFT nur geringfügig: Um aus einem gegebenen Wertevektor $(H_0, \ldots, H_{n-1})$ den Koeffizientenvektor $(a_0, \ldots, a_{n-1})$ zurückzuerhalten, wende man die folgende Formel an:

$$a_k = \frac{1}{n} \cdot \sum_{u=0}^{n-1} H_u e^{-iuk2\pi/n}$$

- **Beweis:** Man betrachte hierzu die DFT als Vektor-Matrix-Multiplikation, wobei wiederum $z := x_{1,n} = e^{i2\pi/n}$:

$$\begin{pmatrix} H_0 \\ H_1 \\ H_2 \\ H_3 \\ \vdots \\ H_{n-1} \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & z & z^2 & \cdots & z^{n-1} \\ 1 & z^2 & z^4 & \cdots & z^{2(n-1)} \\ 1 & z^3 & z^6 & \cdots & z^{3(n-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & z^{n-1} & z^{2(n-1)} & \cdots & z^{(n-1)(n-1)} \end{pmatrix} \cdot \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ \vdots \\ a_{n-1} \end{pmatrix}$$

Diese Matrix wird *Vandermondesche Matrix* genannt. Obige Formel für die inverse DFT behauptet, dass die zur Vandermondeschen Matrix inverse Matrix gerade so aussieht, dass im (u,k)-ten Eintrag z^{-uk}/n steht. Dies wird überprüft, indem die Vandermondesche Matrix mit der (behaupteten) Inversen multipliziert wird. Der (u,k)-te Eintrag dieser Produktmatrix ergibt sich zu:

$$\sum_{l=0}^{n-1} z^{ul} z^{-lk}/n = \frac{1}{n} \cdot \sum_{l=0}^{n-1} z^{l(u-k)} = \begin{cases} 1, & u = k \\ 0, & u \neq k \end{cases}$$

D.h., das Ergebnis ist die Einheitsmatrix. □

9 Diskrete Fourier-Transformation im Fall $d = 2$

- Analog zum Fall $d = 1$ wird hier eine (komplexe) Matrix auf eine komplexe Matrix mit denselben Dimensionen abgebildet, d.h., zu einer gegebenen $(m \times n)$-Matrix $\mathbf{h}$ bezeichnet die $(m \times n)$-Matrix $\mathbf{H}$ mit

$$H_{uv} = \sum_{k=0}^{m-1} \sum_{l=0}^{n-1} h_{kl} \exp(2\pi i\, l\, v/n) \exp(2\pi i\, k\, u/m)$$

die DFT von $\mathbf{h}$, wobei $u = 0, \ldots, m-1$ und $v = 0, \ldots, n-1$.

- Desweiteren ist die inverse DFT gegeben durch

$$h_{kl} = \frac{1}{m \cdot n} \sum_{k=0}^{m-1} \sum_{l=0}^{n-1} H_{uv} \exp(-2\pi i\, l\, v/n) \exp(-2\pi i\, k\, u/m) ,$$

wobei $k = 0, \ldots, m-1$ und $l = 0, \ldots, n-1$.

10 Beispiel zur Implementierung im Fall $d = 2$

- Es sei die folgende (4×4)-Bildmatrix $\mathbf{A}$ gegeben.

$$\mathbf{A} = \begin{pmatrix} a_{30} & a_{31} & a_{32} & a_{33} \\ a_{20} & a_{21} & a_{22} & a_{23} \\ a_{10} & a_{11} & a_{12} & a_{13} \\ a_{00} & a_{01} & a_{02} & a_{03} \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

- Die Matrizen $\mathbf{A}'$ und $\mathbf{B}'$ ergeben sich, indem die Matrizen $\mathbf{A}$ und $\mathbf{B}$ mit entsprechend vielen Nullen aufgefüllt werden, wobei $\mathbf{B} = \check{\mathbf{A}}$.

$$\mathbf{A}' = \begin{pmatrix} * & 0 \\ \boxed{\begin{matrix} & * & \\ a_{00} & a_{01} & \cdots \end{matrix}} & * \end{pmatrix}_{8\times 8} \qquad \mathbf{B}' = \begin{pmatrix} * & 0 \\ \boxed{\begin{matrix} \cdots & a_{01} & a_{00} \\ & * & \\ b_{00} & b_{01} & \cdots \end{matrix}} & * \end{pmatrix}_{8\times 8}$$

- Mittels Faltung der Matrizen $\mathbf{A}'$ und $\mathbf{B}'$ ergibt sich die Matrix $\mathbf{C}$, deren Eintragungen den Werten des Zählers von Formel (2) für alle zulässigen Vektoren $h \in \mathbb{R}^2$ entsprechen und somit der Berechnung der Kovarianz dienen.

$$\mathbf{C} = \begin{pmatrix} 0 & \cdots & & & & & \cdots & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & \vdots \\ 0 & 1 & 1 & 2 & 2 & 1 & 1 & \\ 0 & 2 & 3 & 4 & 4 & 3 & 2 & \\ 1 & 3 & 4 & 9 & 4 & 3 & 1 & \\ 2 & 3 & 4 & 4 & 3 & 2 & 0 & \\ 1 & 1 & 2 & 2 & 1 & 1 & 0 & \vdots \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \end{pmatrix}_{8\times 8}$$

Literatur

[1] Daley, D.J. and Vere-Jones, D. (2001) *An Introduction to the Theory of Point Processes*. Springer-Verlag, New York (preprint).

[2] Jähne, B. (1995) *Digital Image Processing*. Springer-Verlag, New York.

[3] Molchanov, I. (1997) *Statistics of the Boolean Model for Practitioners and Mathematicians*. J. Wiley & Sons, Chichester.

[4] Ohser, J. and Mücklich, F. (2000) *Statistical Analysis of Microstructures in Material Science*. J. Wiley & Sons, Chichester.

[5] Press, W.H., Flannery, B.P., Teukolsky, S.A., and Vetterling, W.T. (1986) *Numerical Recipes*. Cambridge University Press, Cambridge.

[6] Schöning, U. (1996) *Vorlesungsskript zu Algorithmen*. Universität Ulm (2. Auflage).

[7] Stoyan, D., Kendall, W.S., and Mecke, J. (1995) *Stochastic Geometry and its Applications*. J. Wiley & Sons, Chichester (2nd ed.).