

Merkzettel Bildverarbeitung

Sven Findeiß

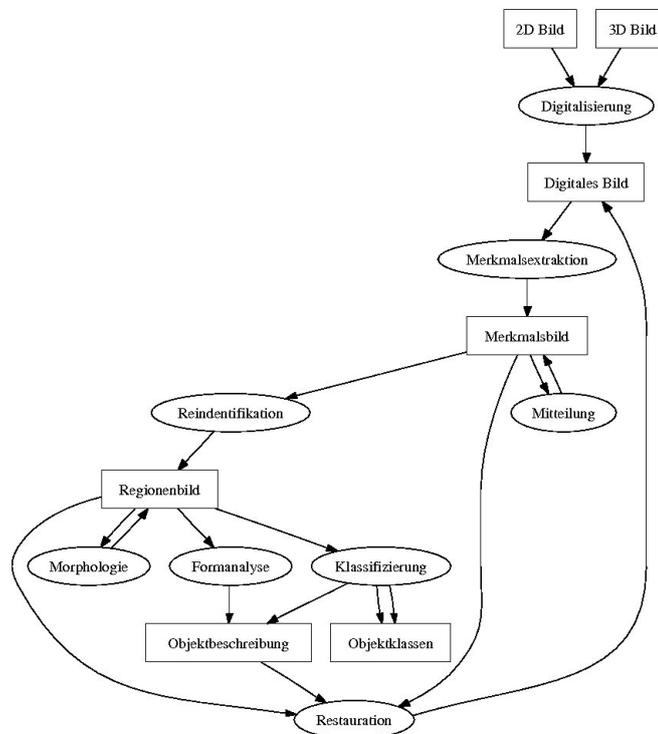
im
Sommersemester 2005

Inhaltsverzeichnis

1	Bildrepresentation	3
1.1	Nachbarschaften	4
1.2	Abstandsmaße	4
2	Fouriertransformation	4
2.1	Bilder als Vektorräume	4
2.2	Idee der Fouriertransformation	5
2.3	diskrete Fouriertransformation	5
2.4	Serie 1	6
2.5	Fast - Fouriertransformation	9
2.6	Serie 2	11
3	Theoretische Grundlagen	12
3.1	Lineare Filter	12
3.2	Skalenraum	14
4	Merkmalsextraktion und Mittelung	16
4.1	Merkmalsextraktion	16
4.2	Mittelung	17
4.3	Serie 3	21
4.4	Kantendetektion	24
4.5	Serie 4	28
4.6	Texturanalyse	29
4.7	Segmentierung	33
5	Hinweise	36
6	Anhang	36

Definition Bildverarbeitung

Bildverarbeitung ist ein mehrstufiger Prozess.



1 Bildrepresentation

1. Analoge Bilder

- analoges 2D Bild ist ein **zweikanaliges, kontinuierliches, reellwertiges** Signal über einem Rechteck.

$$\mathbb{R}^2 \supset R \rightarrow \mathbb{R}; x \rightarrow g(x)$$

- analoges 3D Bild ist ein **dreikanaliges, kontinuierliches, reellwertiges** Signal über einem Quader.

$$g : \mathbb{R}^3 \supset V \rightarrow \mathbb{R}; x \rightarrow g(x)$$

2. Digitale Bilder

- digitales 2D Bild ist ein **diskretes, skalares** Signal über einem Rechteckgitter mit den Abständen $\Delta x, \Delta y$ und Werten aus der Wertmenge W .

$$p : |0, 1\Delta x, \dots, (N_1 - 1)\Delta x| \times |0, 1, \Delta y, \dots, (N_2 - 1)\Delta y| \rightarrow W$$

$$r_{ij} = \begin{pmatrix} i\Delta x \\ j\Delta y \end{pmatrix}$$

$$p_{ij} = p_{ij}(r_{ij})$$

Ein 2D Bild ist also $P = (N_1, N_2, \Delta x, \Delta y, W, (p_{ij}))$

- digitales 3D Bild ist ein **diskretes, skalares** Signal über einem regelmäßigen 3D Gitter mit Abständen $\Delta x, \Delta y, \Delta z$ mit Werten in W .

$$p : |0, 1\Delta x, \dots, (N_1-1)\Delta x| \times |0, 1, \Delta y, \dots, (N_2-1)\Delta y| \times |0, 1, \Delta z, \dots, (N_3-1)\Delta z| \rightarrow W$$

$$r_{ijk} = \begin{pmatrix} i\Delta x \\ j\Delta y \\ k\Delta z \end{pmatrix}$$

$$v_{ijk} = v(r_{ijk})$$

Ein 3D Bild ist also $V = (N_1, N_2, \Delta x, \Delta y, \Delta z, W, (v_{ijk}))$

1.1 Nachbarschaften

1. Nachbarschaften in 2D

- (a) Pixel mit gemeinsamen Kanten als Nachbarn ansehen (4er Nachbarschaft)

	m-1,n	
m,n-1	m,n	m,n+1
	m+1,n	

- (b) Pixel mit gemeinsamen Endpunkten als Nachbarn ansehen (8er Nachbarschaft)

m-1,n-1	m-1,n	m-1,n+1
m,n-1	m,n	m,n+1
m+1,n-1	m+1,n	m+1,n+1

2. Nachbarschaften in 3D

- (a) Voxel mit gemeinsamer Seitenfläche (6er Nachbarschaft)
 (b) Voxel mit gemeinsamer Kante (18er Nachbarschaft)
 (c) Voxel mit gemeinsamen Eckpunkten (26er Nachbarschaft \rightarrow vollständiger Würfel)

1.2 Abstandsmaße

1. Euklidische Distanz zweier Punkte

- Entfernungsmessung im Gitter
- einziges für die Praxis bedeutendes Maß

2. Blockdistanz zweier Punkte

- gibt die Länge des Weges zwischen 2 Punkten an, wenn man nur horizontal und vertikal gehen kann
- entspricht 4er Nachbarschaft

3. Schachbrettdistanz zweier Punkte

- Maximum der Entfernung von horizontaler und vertikaler Richtung

2 Fouriertransformation

2.1 Bilder als Vektorräume

- 2D Bilder als $N \times N$ - dimensionale reelle Vektorräume (für $W = \mathbb{R}$)

- $N_1 \times N_2$ Basisbilder

$${}^{m,n}P_{ij} = \begin{cases} 1; i = m \wedge j = n \\ 0; \text{sonst} \end{cases}$$

- nur ein Bildpunkt mit Wert 1, sonst lauter Bildpunkte mit Wert 0
- analog für 3D Bilder $N_1 \times N_2 \times N_3$ mit Basisbilde

$${}^{m,n,o}v_{ijk} = \begin{cases} 1; i = m \wedge j = n \wedge k = o \\ 0; \text{sonst} \end{cases}$$

- jedes 3D Bild lässt sich als Linearkombination $V = \sum_{i=0}^{N_1-1} \sum_{j=0}^{N_2-1} \sum_{k=0}^{N_3-1} v_{ijk} {}^{ijk}V$ darstellen

2.2 Idee der Fouriertransformation

Die grundlegende Idee der Fouriertransformation ist es an Stelle der vorigen Basis periodische Muster als Basis zu verwenden \Rightarrow Basistransformation die eindeutig und umkehrbar ist. Die Verschiebung des Musters wird durch die Exponentialfunktion mit folgenden Bedingungen ermöglicht:

$$\begin{aligned} \exp(i\varphi) &= \cos\varphi + i\sin\varphi \\ \exp(-i\varphi) &= \cos\varphi - i\sin\varphi \\ \exp(i\varphi) + \exp(-i\varphi) &= 2\cos\varphi \end{aligned}$$

2.3 diskrete Fouriertransformation

- findet Anwendung im 1D Fall
- gegeben ist ein komplexer Vektor

$$g = (g_0, g_1, \dots, g_{N-1})^T$$

- der komplexe Vektor wird durch die 1D DFT auf den Vektor \hat{g} der gleichen Dimension N abgebildet

$$\hat{g}_v = \frac{1}{N} \sum_{n=0}^{N-1} g_n \exp\left(\frac{-2\pi i n v}{N}\right); v = 0, \dots, N-1$$

- Rücktransformation gegeben durch

$$g_n = \sum_{v=0}^{N-1} \hat{g}_v \exp\left(\frac{2\pi i n v}{N}\right); n = 0, \dots, N-1$$

- Kurzschreibweise mit $w_N = w^{\frac{1}{N}} = \exp\left(\frac{2\pi i}{N}\right)$

$$\hat{g}_n = \frac{1}{N} \sum_{v=0}^{N-1} w_N^{-nv} g_v = \langle {}^v b, g \rangle = {}^v b^T g$$

$${}^v b = \frac{1}{N} [w_N^0, w_N^{1v}, \dots, w_N^{(N-1)v}]^T$$

- **Eigenschaften der Fouriertransformation**

- ein Beispiel aus der Klasse der unitären Transformationen in einem Vektorraum
- komplexe Notation verursacht verdoppelten Speicherbedarf
- DFT einer reellen, diskreten Funktion ist hermitesch
- für beliebige gerade N_1, N_2, N_3 lässt sich die DFT $\hat{g}_{\mu\nu\zeta} \in \mathbb{C}$, auf Grund der Symmetrie, durch $N_1 \cdot N_2 \cdot N_3$ reelle Zahlen abbilden.

- **Separabilität**

$$\prod_{p=1}^r g(x) \Leftrightarrow \prod_{p=1}^r \hat{g}(x)$$

- **Ähnlichkeit**

dehnt man ein 2D oder 3D Bild um die Faktoren d_1, d_2 bzw. d_1, d_2, d_3 so erzeugt diese Aufwärtsabtastung um d_1, d_2, d_3 eine (d_1, d_2, d_3) -fache Wiederholung der DFT.

5	7	3
1	4	2

 \Rightarrow

5		7		3	
1		4		2	

- **Verschiebung**

Man nimmt vereinfacht an, dass sich das Bild in alle Richtungen wiederholt

2.4 Serie 1

1. Zeigen sie, dass die auf Folie 18 im Skript definierte Rücktransformation der 1D DFT wirklich ihre Funktion erfüllt.

zu zeigen:

$$\begin{aligned} g_n &= \sum_{v=0}^{N-1} \hat{g}_v \exp\left(\frac{2\pi i n v}{N}\right) \\ &= \sum_{v=0}^{N-1} \hat{g}_v \cdot w_N^{nv} \end{aligned}$$

Beweis:

$$\begin{aligned} \sum_{v=0}^{N-1} \hat{g}_v \cdot w_N^{nv} &= \sum_{v=0}^{N-1} \left(\frac{1}{N} \sum_{r=0}^{N-1} g_r \cdot w_N^{rv} \right) \cdot w_N^{nv} \\ &= \frac{1}{N} \sum_{r=0}^{N-1} g_r \cdot \sum_{v=0}^{N-1} w_N^{v(n-r)} \end{aligned}$$

Für $r = n$ hat die innere Summe

$$\sum_{v=0}^{N-1} w_N^{v0} = N$$

Für $r \neq n$ gilt nach der Formel für die Teilsumme der geometrischen Reihe:

$$\sum_{k=0}^{N-1} x^k = \frac{x^N - 1}{x - 1}, x \neq 1$$

$$\begin{aligned}
&\Rightarrow 1 + w_N^{n-r} + (w_N^{n-r})^2 + \dots + (w_N^{n-r})^{N-1} \\
&\frac{(w_N^{n-r})^N - 1}{w_N^{n-r}} \\
&\frac{(w_N^N)^{nr} - 1}{w_N^{n-r}} \\
&\frac{1^{n-r} - 1}{w_N^{n-r} - 1} \\
&\Rightarrow \frac{1}{N} \sum_{r=0}^{N-1} g_r \cdot \sum_{v=0}^{N-1} w_N^{v(n-r)} = \frac{1}{N} \cdot N \cdot g_n = g_n
\end{aligned}$$

q. e. d.

2. Berechnen sie die diskrete Fouriertransformierte des Vektors

$$\begin{aligned}
c &= \begin{pmatrix} 3 \\ -2 \\ 0 \\ 1 \end{pmatrix} = (3, -2, 0, 1)^T \\
\hat{g}_n &= \sum_{n=0}^{N-1} w_N^{-nv} \\
g_n &= \langle {}^v b, g \rangle = {}^v b^T g \\
{}^v b &= \frac{1}{N} [w_N^{0 \cdot v}, w_N^{1 \cdot v}, w_N^{2 \cdot v}, \dots, w_N^{(N-1)v}] \\
&\Rightarrow w_y^x = \exp\left(\frac{2\Pi i x}{y}\right) \\
&\Rightarrow w_4^x = \exp\left(\frac{2\Pi i x}{4}\right)
\end{aligned}$$

Nimmt man an dieser Stelle von $\exp\left(\frac{2\Pi v n}{N}\right)$ den Term $\exp\left(-\frac{2\Pi v n}{N}\right)$ spart man sich das komplex Konjugieren beim Einsetzen.

Zum umrechnen der komplexen Zahlen siehe Anhang Abb 1

$${}^0b = \frac{1}{4} [w_4^{0\cdot 0}, w_4^{1\cdot 0}, w_4^{2\cdot 0}, w_4^{3\cdot 0}]$$

$${}^0b = \frac{1}{4} [1, 1, 1, 1]$$

$${}^1b = \frac{1}{4} [w_4^{0\cdot 1}, w_4^{1\cdot 1}, w_4^{2\cdot 1}, w_4^{3\cdot 1}]$$

$${}^1b = \frac{1}{4} [1, i, -1, -i]$$

$${}^3b = \frac{1}{4} [w_4^{0\cdot 2}, w_4^{1\cdot 2}, w_4^{2\cdot 2}, w_4^{3\cdot 2}]$$

$${}^2b = \frac{1}{4} [1, -1, 1, -1]$$

$${}^3b = \frac{1}{4} [w_4^{0\cdot 3}, w_4^{1\cdot 3}, w_4^{2\cdot 3}, w_4^{3\cdot 3}]$$

$${}^3b = \frac{1}{4} [1, -i, -1, i]$$

Allgemein:

$$\frac{1}{4} \begin{pmatrix} w_4^0 & w_4^0 & w_4^0 & w_4^0 \\ w_4^0 & w_4^1 & w_4^2 & w_4^3 \\ w_4^0 & w_4^2 & w_4^4 & w_4^6 \\ w_4^0 & w_4^3 & w_4^6 & w_4^9 \end{pmatrix} \cdot c$$

Eingesetzt (ACHTUNG komplex konjugiert

→ Vorzeichen beim Imaginerteil tauschen):

$$\frac{1}{4} \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -i & -1 & i \\ 1 & -1 & 1 & -1 \\ 1 & i & -1 & -i \end{pmatrix} \cdot \begin{pmatrix} 3 \\ -2 \\ 0 \\ 1 \end{pmatrix}$$

Matrixmultiplikation (Summe aus Zeile mal Spalte):

$$\frac{3}{4} - \frac{1}{2} + \frac{1}{4} + 0 = \frac{1}{2}$$

$$\frac{3}{4} + \frac{2i}{4} + 0 + \frac{i}{4} = \frac{3}{4} + \frac{3i}{4}$$

$$\frac{3}{4} + \frac{2}{4} + 0 - \frac{1}{4} = 1$$

$$\frac{3}{4} - \frac{2i}{4} + 0 - \frac{i}{4} = \frac{3}{4} - \frac{3i}{4}$$

es ergibt sich folgende Fouriertransformierte:

$$\begin{pmatrix} \frac{1}{2} \\ \frac{3}{4} + \frac{3i}{4} \\ 1 \\ \frac{3}{4} - \frac{3i}{4} \end{pmatrix}$$

3. Berechnen sie die diskrete Fast-Fouriertransformation des Laplace Filters:

$$L = \begin{pmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$

$$\hat{l}_{\mu v} = \frac{1}{N_1 N_2} \sum_{m=0}^{N_1-1} \left(\sum_{n=0}^{N_2-1} l_{mn} w_{N_2}^{-nv} \right) w_{N_1}^{-m\mu}$$

Aufspaltung in 1 D DFT (2 mal)

$$l_{mv}^* = \frac{1}{N_2} \cdot \sum_{n=0}^{N_2-1} l_{mn} w_{N_2}^{-nv} = \frac{1}{3} \cdot \sum_{n=0}^2 l_{mn} w_3^{-nv}$$

$$\hat{l}_{\mu v} = \frac{1}{N_1} \cdot \sum_{m=0}^{N_1-1} l_{mv}^* w_{N_1}^{-\mu m}$$

$$l_{00}^* = \frac{1}{3} \cdot \sum_{n=0}^2 l_{04} = \frac{1}{3} (0 + 1 + 0) = \underline{\underline{\frac{1}{3}}}$$

$$l_{10}^* = -\frac{2}{3}$$

$$l_{20}^* = \frac{1}{3}$$

$$l_{01}^* = \frac{1}{3} \cdot \sum_{n=0}^2 l_{04} \exp\left(\frac{-2\Pi}{3}\right)$$

$[\frac{1}{3}(-\frac{1}{2}-\frac{1}{2}\sqrt{3})]$

$$= \frac{-1 - \sqrt{3}i}{6}$$

$$0; 1 + \sqrt{3}i; 1 - \sqrt{3}i$$

$$l = \begin{pmatrix} \frac{1+\sqrt{3}i}{6} & \frac{1-7\sqrt{3}i}{36} & \frac{-3-\sqrt{3}i}{9} \\ \frac{1-\sqrt{3}i}{6} & -\frac{7}{8} & \frac{1+\sqrt{3}i}{3} \end{pmatrix}$$

2.5 Fast - Fouriertransformation

Die "normale" Fouriertransformation ist zu komplex, um mit ihr in der Bildverarbeitung arbeiten zu können// \Rightarrow ein vereinfachter Algorithmus ist gesucht.

Algorithmus: (für $N = 2^k$)

- Die DFT berechnet sich aus der Summe

$$\hat{g}_v = \frac{1}{N} \sum_{n=0}^{N-1} g_n \exp\left(\frac{-2\Pi i n v}{N}\right)$$

$$v = 0, \dots, N-1$$

$$w_N^{nv} := \exp\left(\frac{-2\Pi i n v}{N}\right)$$

$$\Rightarrow \hat{g}_v = \frac{1}{N} \sum_{n=0}^{N-1} g_n w_N^{nv}$$

Wenn wir z.B. einen Vektor \vec{c} der Länge $N = 8$ betrachten so hat dieser folgende Indizes

$$\vec{c} = (c[0], c[1], c[2], c[3], c[4], c[5], c[6], c[7])$$

Den ersten Teilungsschritt könnte man als eine Aufteilung der oberen Summe in zwei Teilsommen erklären. Die eine Teilsomme enthält alle geraden Indizes des Vektors und die andere alle ungeraden.

$$N \cdot \hat{g}_v = \sum_{n=0}^{\frac{N}{2}-1} g_{2n} w_{\frac{N}{2}}^{vn} + w_N^v \sum_{n=0}^{\frac{N}{2}-1} g_{2n+1} w_{\frac{N}{2}}^n$$

$$v = 0, \dots, N - 1$$

Es ergeben sich zwei Teilvektoren:

$$\vec{c}' = (c[0], c[2], c[4], c[6])$$

$$\vec{c}'' = (c[1], c[3], c[5], c[7])$$

Diese Art der Erläuterung des ersten Teilungsschritts erschwert es aber die weiteren Teilungsschritte zu erklären.

Deshalb hier eine andere Variante:

Nimmt man die Indizes des Vektors und schreibt sie als Binärzahlen ergibt sich folgendes:

$$\vec{c} = (c[0], c[1], c[2], c[3], c[4], c[5], c[6], c[7])$$

$$= (c[000], c[001], c[010], c[011], c[100], c[101], c[110], c[111])$$

Sortiert man diese Binär - Indizes nach dem letzten Bit ergeben sich folgende zwei Vektoren:

$$\vec{c}'_2 = (c[000], c[010], c[100], c[110])$$

$$\vec{c}''_2 = (c[001], c[011], c[101], c[111])$$

Das diese Aufteilung die selbe ist wie bei der ersten Erläuterung wird durch einfaches umrechnen klar. Die beiden Vektoren \vec{c}'_2 und \vec{c}''_2 haben beide nun die Länge $\frac{N}{2} = 4$. Da diese auch durch eine 2er Potenz darstellbar ist kann man die beiden Vektoren nochmals teilen. Dies erfolgt durch die Sortierung des nächst höherwertigen Bits:

$$\vec{c}'_2^* = (c[000], c[100])$$

$$\vec{c}'_2^{**} = (c[010], c[110])$$

$$\vec{c}''_2^{***} = (c[001], c[101])$$

$$\vec{c}''_2^{****} = (c[011], c[111])$$

Es ergeben sich also nun 4 Vektoren der Länge $\frac{N}{4} = 2$.

Eine weitere Zerlegung nach diesem Schema würde uns acht Vektoren der Länge 1 liefern. Deren Fouriertransformierte ist trivial, nämlich der Vektor selbst.

Es schließt sich nun eine schrittweise Zusammenfügung der einzelnen Vektoren an. Diese Schritte sind in Serie 2 möglichst ausführlich dargestellt.

2.6 Serie 2

1. Berechnen sie die diskrete Fast - Fouriertransformation des Vektors

$$c = \begin{pmatrix} 3 \\ -5 \\ 0 \\ 1 \\ 2 \\ 7 \\ 3 \\ -2 \end{pmatrix}$$

Eine grafische Darstellung der Berechnung siehe Anhang Abbildung 2

Erläuterungen und Berechnungen:

- Bevor man die einzelnen sogenannten Butterfly - Operationen durchführt ist es ratsam alle Termgewichte w^x zu berechnen:

$$w^x = \exp\left(-\frac{2\Pi i x}{8}\right)$$

$$w^0 = \exp\left(-\frac{2\Pi i 0}{8}\right) = 1$$

$$w^1 = \exp\left(-\frac{2\Pi i 1}{8}\right) = \exp\left(-\frac{1\Pi i}{4}\right) = \frac{\sqrt{2}}{2} - \frac{\sqrt{2}}{2}i$$

$$w^2 = \exp\left(-\frac{2\Pi i 2}{8}\right) = \exp\left(-\frac{1\Pi i}{2}\right) = -i$$

$$w^3 = \exp\left(-\frac{2\Pi i 3}{8}\right) = \exp\left(-\frac{3\Pi i}{4}\right) = -\frac{\sqrt{2}}{2} - \frac{\sqrt{2}}{2}i$$

$$w^4 = \exp\left(-\frac{2\Pi i 4}{8}\right) = \exp(-\Pi i) = -1$$

$$w^5 = \exp\left(-\frac{2\Pi i 5}{8}\right) = \exp\left(-\frac{5\Pi i}{4}\right) = -\frac{\sqrt{2}}{2} + \frac{\sqrt{2}}{2}i$$

$$w^6 = \exp\left(-\frac{2\Pi i 6}{8}\right) = \exp\left(-\frac{3\Pi i}{2}\right) = i$$

$$w^7 = \exp\left(-\frac{2\Pi i 7}{8}\right) = \exp\left(-\frac{7\Pi i}{4}\right) = \frac{\sqrt{2}}{2} + \frac{\sqrt{2}}{2}i$$

Die Ergebnisse der letzten Spalten lassen sich aus der Abbildung 1 im Anhang ablesen.

- Die jeweiligen Termgewichte in Abbildung 2 an den Pfeilen sind in zwei Farben dargestellt. Blau sind Termgewichte für die Pfeile von links unten nach rechts oben und rot die waagerechten Pfeile. Es wird jeweils der am Startpunkt des Pfeils befindliche Eintrag mit dem Termgewicht multipliziert.

$$c[0] = 3 \rightarrow c[0] + c[4] \cdot w^0 = 3 + 2 \cdot 1 = 5$$

$$c[4] = 2 \rightarrow c[0] + c[4] \cdot w^4 = 3 + 2 \cdot -1 = 1$$

Diese Berechnungen setzen sich für alle weiteren Teilschritte fort.

- Für die letzte Spalte benötigt man teilweise die Berechnung mit Komplexenzahlen. Hier beispielhaft die Berechnung von $\hat{c}[1]$:

$$\begin{aligned}
& (1 + 3i) + (-12 - 3i) \cdot \left(\frac{\sqrt{2}}{2} - \frac{\sqrt{2}}{2}i \right) \\
&= (1 + 3i) + \left(-\frac{12\sqrt{2}}{2} + \frac{12\sqrt{2}}{2}i - \frac{3\sqrt{2}}{2}i + \frac{3\sqrt{2}}{2}i^2 \right) \\
& \quad i^2 \Rightarrow -1 \\
&= (1 + 3i) + \left(-\frac{12\sqrt{2}}{2} + \frac{12\sqrt{2}}{2}i - \frac{3\sqrt{2}}{2}i - \frac{3\sqrt{2}}{2} \right) \\
&= (1 + 3i) + \left(-\frac{15\sqrt{2}}{2} + \frac{9\sqrt{2}}{2}i \right) \\
&= 1 + \frac{9\sqrt{2}}{2}i - \frac{15\sqrt{2}}{2} + 3i
\end{aligned}$$

Es ergibt sich nach der Berechnung folgender Vektor \hat{c}

$$\begin{pmatrix} 9 \\ 1 - \frac{15\sqrt{2}}{2} + \frac{9\sqrt{2}}{2}i + 3i \\ 2 - 3i \\ 1 + \frac{15\sqrt{2}}{2} + \frac{9\sqrt{2}}{2}i - 3i \\ 7 \\ 1 + \frac{15\sqrt{2}}{2} - \frac{9\sqrt{2}}{2}i + 3i \\ 2 + 3i \\ 1 - \frac{15\sqrt{2}}{2} - \frac{9\sqrt{2}}{2}i - 3i \end{pmatrix}$$

3 Theoretische Grundlagen

3.1 Lineare Filter

- Ziel ist es, in 2-dimensionalen Grauwertbildern, mit uniformen grauen Objekten auf schwarzen Hintergrund, deren Kanten zu finden
- am Beispiel typischer Nachbarschaftsoperator Laplace - Filter

7	7	3	6	2	1	1
7	7	3	6	2	1	1
1	1	2	4	0	0	0
0	0	2	1	10	1	1
6	6	5	3	4	7	7
6	6	5	3	4	7	7

$$L = \begin{pmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{pmatrix} \rightarrow \boxed{2}$$

- Filtermaske wird auf das Bild gelegt und die jeweiligen Werte mit dem darunter liegenden (Pixel des Ausgangsbildes) multipliziert
- bilden der Summe der entstanden Produkte
 $7 \cdot 0 + 3 \cdot 1 + 6 \cdot 0 + 1 \cdot 1 + 2 \cdot (-4) + 4 \cdot 1 + 0 \cdot 0 + 2 \cdot 1 + 1 \cdot 0 = 2$
- Theorie geht von einem Torus aus
- Praxis verwendet am Bildrand einfache Pixelwiederholung (siehe oberes Beispiel)

- **Filterdefinition:**

Bilder werden transformiert → resultierende Merkmalsbilder um leichter Erkenntnisse über

- Kanten
- Texturgrenzen
- Textureigenschaften
- Regionen

zu erhalten

- **Linearität:**

gilt wenn:

$$H(G + G') = H(G) + H(G')$$

Gegenbsp. Medianfilter:

$$M([0100] + [0010]) = [0110] \neq M[0100] + M[0010] = [0000]$$

- **Verschiebungsinvarianz:**

Die Antwort des Operators ist nicht ausdrücklich von der Position im Bild abhängig

→ verschieben wir ein Bild ist das Ausgabebild identisch (mit Ausnahme der Verschiebung).

Bei Verschiebung $S_{r,s,t}$ um $(r, s, t) \in \mathbb{R}^3$ eines Bildes gilt für einen verschiebungsinvariante Filter H

$$S_{r,s,t}(H(V)) = H(S_{r,s,t}(V))$$

⇒ es ist egal ob erst verschoben wird und dann gefiltert oder umgedreht

- **Punktantwort:**

verschiebungsinvariante, lineare Filter (linear shift invariant filter - LSI - Filter) können durch ihre Punktantwort (point spread function - PSF) dargestellt werden

PSF zu H:

$$H = H^{(0,0,0)}V$$

PSF für Laplace Filter:

$$\begin{pmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{pmatrix} = \begin{pmatrix} l_{11} & l_{01} & l_{-11} \\ l_{10} & l_{00} & l_{-10} \\ l_{1-1} & l_{0-1} & l_{-1-1} \end{pmatrix}$$

bei allen LSI - Filtern ist die PSF in alle Richtungen um das Zentrum gespiegelt

allg. Eigenschaften von LSI - Filter:

$$G' = HG$$

H ist LSI - Operator bzw. LSI - Filter wenn:

- **Kommutativität**
Reihenfolge der Operanden tauschbar

$$HH' = H'H$$

Assoziativität

komplexer Operator aus einfache Operanden zusammen setzbar (in einfache zerlegbar)

$$H = H' H''$$
$$\begin{pmatrix} 1 & 4 & 6 & 4 & 1 \\ 4 & 16 & 24 & 16 & 4 \\ 6 & 24 & 32 & 24 & 6 \\ 4 & 16 & 24 & 16 & 4 \\ 1 & 4 & 6 & 4 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 4 & 6 & 4 & 1 \end{pmatrix} * \begin{pmatrix} 1 \\ 4 \\ 6 \\ 4 \\ 1 \end{pmatrix}$$

* der komplexe Filter benötigt 25 Multiplikationen und 24 Additionen

* die Zerlegung benötigt 10 Multiplikationen und 8 Additionen

• **Faltung:**

- jeder LSI - Filter lässt sich als Faltung mit der PSF beschreiben
- ein LSI - Operator ist immer ein Faltungsoperator im Ortsraum

Theorem:

Faltung zweier Bilder im Ortsraum entspricht ihrer komplexen Multiplikation in Fourierraum

• **Abtastung:**

- Gefahr, dass Einzelheiten der Bildauflösung verloren gehen
- Abtastung feiner Details kann zur Erzeugung größerer Strukturen führen, die vorher nicht im Ausgabebild waren (“Moire - Effekt“)

Theorem:

Ist die Fouriertransformierte $\hat{g}(k)$ durch eine kontinuierliche Funktion $g(x)$ bandbegrenzt

$$\hat{g}(k) = 0; \forall |k_j| \geq \frac{1}{K_j}; j = 1, 2, 3$$

so kann sie aus einem Bild abgetasteter Punkte mit Schrittweite

$$\Delta x_j = \frac{1}{K_j}$$

exakt rekonstruiert werden.

⇒ periodische Struktur kann rekonstruiert werden, falls mind. 2 Abtastpunkte pro Wellenlänge gesetzt werden

⇒ größte Wellenlänge die dies erfüllt heißt Nyquist - Wellenzahl

3.2 Skalenraum

- lokale Merkmale lassen sich durch LSI - Filter erkennen mit kleinen Masken
- größerzahlige Informationen mit größeren Filtermasken zu ermitteln benötigt achtfachen Rechenaufwand ∩ unpraktikabel
- Datenstruktur, die aus einer Folge von Bildern mit unterschiedlicher Auflösung besteht wird als Skalenraum bezeichnet
 $g(x, \xi)$ ist Skalenraum des Bildes $g(x)$

• **Diffusion:**

- Glättungsfilter muss vorher kleine Wellenlängen unterdrücken
 - Faltung mit Binomialfilter bestens geeignet
 - Filter B^2 in alle Richtungen → Gaußpyramide
- die ges. Pyramide benötigt nur wenig mehr Speicher als das Bild selbst
- weniger Rechenoperationen als original Bild
- Berechnung der $(q + 1)$ Ebene

$$G^{q+1} = B_{|2}G^q$$

$|2$ bezeichnet Abtastrate

$$G^0 = G \text{ Ausgangsbild}$$

- von Ebene zu Ebene Abnahme der Auflösung um die Hälfte
- in höheren Ebenen Abnahme und zu nehmend nur größere Details erkennbar

2. Laplacepyramide

- berechenbar aus der Gaußpyramide
- Sequenz von bandpassgefilterten Bildern
- führt zu grober Wellenzahlzerlegung ohne Richtungszerlegung
 - ↪ es bleibt eine gute räumliche Auflösung erhalten
- das letzte Bild der LP ist ein tiefpassgefiltertes Bild mit den größten Strukturen
- Subtraktion zweier Ebenen der Gaußpyramide als Annäherung
- das Bild höherer Ebene muss erst expandiert (auf gleiche Größe gebracht) werden
- Expansionsoperator $E_{|2}$ mit $|2$ als Grad der Expansion
- fehlende Infos müssen interpoliert werden

$$\Rightarrow L^p = G^p - E_{|2}G^{p+1}$$

- Wiederherstellung des original Bildes durch Teilbilder, rekursiv expandieren und aufsummieren

$$G^{p-1} = L^{p-1} + E_{|2}G^p$$

4 Merkmalsextraktion und Mittelung

4.1 Merkmalsextraktion

Pixelverarbeitung:

- bearbeiten der Ausgangsbilder
 - Korrektur von Verzerrungen
 - Kontrastverstärkung

1. Punktoperationen:

- modifiziert Bildpunkte nur in Abhängigkeit ihrer eigenen Werte

$$V'_{mno} = P_{mno}(V_{mno})$$

- homogene Punktoperationen sind von der Position unabhängig

$$V'_{mno} = O(V_{mno})$$

- **Anwendung:**

- Auflösungsverringering
herunterskalieren eines 14 Bit Bild auf 8 Bit
- Beleuchtungsdifferenzen
Löschen niedriger Bits zur visuellen Kontrolle inhomogener Beleuchtungszustände
- Kontrastverstärkung
liegen alle Intensitäten eines Bildes nur in einem kleinen Bereich, so liefert die gekonnte Multiplikation eine Kontrastverstärkung (siehe Serie 3)
- Fensterfunktion
Multiplikation eines Bildes mit einer Fensterfunktion führt zur Vermeidung von Problemen am Rand des Bildes

2. Geometrische Transformationen:

- verändert nur die Position des Bildpunktes und nicht den Wert

$$V_{G(mno)} = V_{mno}$$

- Vorwärts- und Rückwärtsabbildung als Möglichkeiten

- **Interpolation:**

- * Hauptproblem der schnellen und exakten Implementierung geometrischer Operatoren
- * Gitterpunkte werden im Allgemeinen nicht auf Gitterpunkte sondern auf Zwischengitterplätze abgebildet oder umgekehrt
- * allg. Verfahren
 - Rekonstruktion des kontinuierlichen Bildes aus dem Digitalen
 - erneute Abtastung auf neuen Gitterpunkten (funktioniert nur wenn neues Gitter nicht größer als das Alte → sonst Aliasing - Erscheinung)
- * sehr schwierig durchzuführen
→ Rekonstruktion sehr aufwendig und nur als Näherung zu sehen
- * Störung der (tri-)linearen Interpolation (meist akzeptiert)
 - kleine Wellenzahlen werden richtig interpoliert, große Wellenzahlen werden in ihrer Amplitude etwas reduziert ↪ leichte Glättung
 - $\hat{h}(\hat{k})$ für $\hat{k} > 1$ ist nicht 0
→ einige hohe Wellenzahlen werden erzeugt
→ erneutes Abtasten führt zu leichten Aliasing - Effekten

4.2 Mittelung

- Faltung bildet Basis für Mittelungs- und Glättungsfiler

- **Eigenschaften von Glättungsfiltern:**

- **Verschiebungsinvarianz (Verschiebungsfreiheit):**

- * Glättungsfiler darf die Position eines Objektes nicht verändern
- * um spätere Fehler zu vermeiden (durch Objektverschiebung) muss der Filter reell sein

* reelle Transferfunktion impliziert eine symmetrische Filtermasken

$$1D : h_{-n} = h_n$$

$$2D : H_{-m,n} = H_{m,n}, H_{m,-n} = H_{m,n}$$

$$3D : H_{-lmn} = H_{lmn}, H_{l-mn} = H_{lmn}, H_{lm-n} = H_{lmn}$$

– Mittelwerterhaltung:

* der Mittelwert soll erhalten bleiben

⇒ Transferfunktion für die 0 ist 1

⇒ die Summe aller Koeffizienten der Maske ist 1

$$1D : \hat{h}(0) = 1 \Leftrightarrow \sum_n h_n = 1$$

$$2D : \hat{H}(0) = 1 \Leftrightarrow \sum_m \sum_n H_{nm} = 1$$

$$3D : \hat{H}(0) = 1 \Leftrightarrow \sum_l \sum_m \sum_n H_{lmn} = 1$$

– Monoton fallende Transferfunktion (Stärkere Glättung feinerer Strukturen):

* ein Glättungsoperator sollte nicht eine bestimmte Struktur verschwinden lassen, während feinere Strukturen noch im Bild verbleiben

* Mathematisch: die Transferfunktion muss monoton mit der Wellenzahl abnehmen

$$\hat{h}(\bar{k}_2) \leq \hat{h}(\bar{k}_1) \\ \text{für } \bar{k}_2 > \bar{k}_1$$

– Isotropie:

* Unabhängigkeit der Glättung von der Richtung

* die Werte sollten nur radial vom zentralen Pixel abhängen

$$H(x) = H(|x|) \wedge \hat{H}(\bar{k}) = \hat{H}(|\bar{k}|)$$

* im diskreten Raum (→ diskrete Filter) nur annähernt erfüllbar

* Ziel ist die Konstruktion diskreter Masken mit geringer Anisotropie

• Rechteckfilter

– einfachster Glättungsfilter

– Bildpunkte innerhalb der Maske werden addiert und die Summe durch die Anzahl der Pixel geteilt

– Beispiel ist ein 1×3 -Filter

$${}^3R = \frac{1}{3}[111]$$

– Anwendung auf vertikale Kante:

$$\begin{array}{cccc} 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \end{array} * \frac{1}{3}[111] = \begin{array}{cccc} 0 & 1/3 & 2/3 & 1 \\ 0 & 1/3 & 2/3 & 1 \\ 0 & 1/3 & 2/3 & 1 \\ 0 & 1/3 & 2/3 & 1 \end{array}$$

es ergibt sich aus einer scharfen Kante eine Rampe

- Anwendung auf periodische Welle (Wellenzahl = 3)

$$\begin{array}{cccccccccccc} 1 & -2 & 1 & -1 & -2 & 1 & & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & -2 & 1 & -1 & -2 & 1 & * \frac{1}{3} & [111] = & 0 & 0 & 0 & 0 & 0 \\ 1 & -2 & 1 & -1 & -2 & 1 & & & 0 & 0 & 0 & 0 & 0 \end{array}$$

unser Rechteckfilter eine Struktur mit Wellenlänge 3 vollständig heraus

- Anwendung auf periodische (Welle Wellenzahl = 2)

$$\begin{array}{cccccccc} 1 & -1 & 1 & -1 & & -1/3 & 1/3 & -1/3 & 1/3 \\ 1 & -1 & 1 & -1 & * \frac{1}{3} & [111] = & -1/3 & 1/3 & -1/3 & 1/3 \\ 1 & -1 & 1 & -1 & & -1/3 & 1/3 & -1/3 & 1/3 \end{array}$$

die Amplitude wird gedämpft

- es ergibt sich, dass der Rechteckfilter ein guter Tiefpassfilter ist
- die Transferfunktion berechnet sich aus

$$\hat{R} = \frac{1}{3} + \frac{2}{3} \cos(\Pi \bar{k}_x)$$

- diese Funktion hat an $\bar{k} = \frac{2}{3}$ eine Nullstelle \rightarrow Signal mit dieser Wellenlänge wird vollständig eliminiert
- die Wellenlänge $\bar{k} = 1$ wird durch $\frac{1}{3}$ gedämpft und negiert \rightarrow Minimum und Maximum vertauscht
- Nachteile:
 - * keine monoton steigende Dämpfung hoher Wellenzahlen
 - * Dämpfung der hohen Wellenzahlen unzureichend
- ein großer Vorteil
unabhängig von der Filterlänge nur 3 Operationen zur Faltungsberechnung

• **Binomialfilter**

- effizient zu berechnen
- werden aus einfachen Komponenten zusammengesetzt

$$B^P = \frac{1}{2^P} \underbrace{[11] * [11] * \dots * [11]}_{P\text{-mal}}$$

es ergeben sich so folgende Filter:

$$\begin{aligned} B^1 &= \frac{1}{2} [11] \\ B^2 &= \frac{1}{4} [121] \\ B^2 &= \frac{1}{8} [1331] \\ B^3 &= \frac{1}{16} [14641] \end{aligned}$$

\Rightarrow Einträge des Pascalschen Dreieck

- wesentlich besserer Glättungsfilter als Rechteckfilter
- Transferfunktion nimmt monoton ab und geht bei den größten Wellenzahlen gegen 0

- Zweidimensionale Binomialfilter ergeben sich durch Faltung eines horizontalen und eines vertikalen 1D-Binomialfilter

$$B^2 = B_x^2 B_y^2 = \frac{1}{4} [121] * \frac{1}{4} \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} = \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

- Binomialfilter eliminieren Binärs Rauschen
- eine leichte Implementierung mit einfachen Netzwerkbausteinen (Addierer, Multiplizierer,...) möglich
- Hauptproblem ist es wenn die Filtermaske groß wird, dass ein Bildpunkt durch die Größe der Filtermaske verschmiert
- **Mehrschrittmittelung:**
 - * langsame großräumige Mittelungen (Hauptproblem der Binomialfilter) liegen an dem geringen Abstand der Pixel
 - * Lösung durch größere Entfernung der Bildpunkte in der Maske für die Diagonale:

$$B_{x+y} = \frac{1}{4} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$B_{x-y} = \frac{1}{4} \begin{bmatrix} 0 & 0 & 1 \\ 0 & 2 & 0 \\ 1 & 0 & 0 \end{bmatrix}$$

entlang der Achsen:

$$B_{2x} = \frac{1}{4} [1 \quad 0 \quad 2 \quad 0 \quad 1]$$

$$B_{2y} = \frac{1}{4} \begin{bmatrix} 1 \\ 0 \\ 2 \\ 0 \\ 1 \end{bmatrix}$$

für 3D zusätzlich B_{2z}

$$\frac{1}{4} [[1], [0], [2], [0], [1]]$$

- * diese Filter einzeln anzuwenden hilft nicht (für große Zahlen keine Glättungsfilter mehr)
- * man muss sie kaskadieren (Hintereinanderausführung verschiedener Binomialfilter)

• Zusammenfassung Lineare Filter

- dämpfen weißes Rauschen
- sind empfindlich gegenüber binärem Rauschen
- verwischen Objektkanten

• Rangordnungsfiler

- sortieren Werte innerhalb der Maske

2. diese Funktion bildet das umgekehrte Bild (in der Fotografie Negativ)
 → sie weist jedem Wert einen eindeutigen Grauwert zu \curvearrowright umkehrbar

$$P_3(P_2(b)) = 255 - (255 - b) = b$$

am Bsp.:

$$b = 10$$

$$P(b) = 255 - 10 = 245$$

$$P_3(P(b)_2) = 255 - (255 - 10) = \underline{\underline{10 = b}}$$

q. e. d.

3. Nehmen sie an, sie hätten ein unterbelichtetes 8-Bit Grauwert-Bild mit allen Werten unterhalb von 55. Geben sie eine passende Punktoperation zur Kontrastverstärkung an, damit eine möglichst gleichmäßige Grauwertverteilung entsteht.

- (a) • es steht uns ein Grauwertbereich von 8 Bit zur Verfügung $(0, \dots, 255)$
 • es wird aber nur der Bereich von $0, \dots, 54$ verwendet
 \curvearrowright weniger als $1/5$ des Grauwertbereichs wird genutzt
 \curvearrowright gesucht ist eine Punktoperation die den kleinen Grauwertbereich auf den vollen Kontrastbereich abbildet

$$q' = q \cdot k; k = \frac{255}{54}$$

$$q' = \begin{cases} 4q; & q < 54 \\ 255; & q \geq 54 \end{cases}$$

- (b) • wenn $g_{min} > 0$ (kleinster Grauwert) kann noch eine andere Funktion verwendet werden

$$g_{min} > 0 \qquad g_{max} \leq 54 \Rightarrow B = g_{max} - g_{min}$$

$$B = \text{Bereich in dem die Grauwerte liegen}$$

$$k = \frac{255}{B}$$

$$k' = 255 - g_{max} \cdot k$$

$$q' = q \cdot k + k'$$

4. Berechnen sie die Wirkung eines 3×3 Medianfilter mit der einer 3×3 Mittelwertfilters auf die folgenden Bilder. Was fällt auf? Worin unterscheiden sich die beiden Filter in ihrem Verhalten noch?

Medianfilter

Die Werte in der Filtermaske werden der Reihe nach sortiert und der mittlere (bei 9 Werten der 5.) ergibt den Wert für das Pixel.

Mittelwertfilter

Alle Einträge die in der Maske liegen werden aufsummiert und durch deren Anzahl dividiert.

(a) gestörter Bildpunkt

```

1  2  3  4  5
1  2  3  4  5
1  2 95  4  5
1  2  3  4  5
1  2  3  4  5

```

Medianfilter

```

1  2  3  4  5
1  2  3  4  5
1  2  3  4  5
1  2  3  4  5
1  2  3  4  5

```

Mittelwertfilter

```

1  2  3  4  5
1 12 13 14 5
1 12 13 14 5
1 12 13 14 5
1  2  3  4  5

```

(b) Achsenparallele Kante

```

0  0  0  20  20  20
0  0  0  20  20  20
0  0  0  20  20  20

```

Medianfilter

```

0  0  0  20  20  20
0  0  0  20  20  20
0  0  0  20  20  20

```

Mittelwertfilter

```

0  0   $\frac{60}{9} = 7$    $\frac{120}{9} = 13$   20  20
0  0   $\frac{60}{9} = 7$    $\frac{120}{9} = 13$   20  20
0  0   $\frac{60}{9} = 7$    $\frac{120}{9} = 13$   20  20

```

(c) periodisch wiederkehrende Struktur 1

```

1  -2  1  1  -2  1
1  -2  1  1  -2  1
1  -2  1  1  -2  1

```

Medianfilter

```

1  1  1  1  1  1
1  1  1  1  1  1
1  1  1  1  1  1

```

Mittelwertfilter

```

0  0  0  0  0  0
0  0  0  0  0  0
0  0  0  0  0  0

```

(d) periodisch wiederkehrende Struktur 2

```

3 -3 3 3 -3 3
3 -3 3 3 -3 3
3 -3 3 3 -3 3

```

Medianfilter

```

3 3 3 3 3 3
3 3 3 3 3 3
3 3 3 3 3 3

```

Mittelwertfilter

```

1 1 1 1 1 1
1 1 1 1 1 1
1 1 1 1 1 1

```

- **Beobachtung zum Medianfilter**

- der Medianfilter löscht “Ausreißer“ (erstes Bild “95“)
- Kanten bleiben unverändert erhalten (das gleiche gilt für Rampen)
- periodisch wiederkehrende Strukturen gehen die Informationen des Ausgangsbildes verloren

- **Beobachtung zum Mittelwertfilter**

- Ausreißer Pixel werden auf die benachbarten Pixel übertragen \Rightarrow Abhängigkeit von der Größe des Filters
- Kanten werden zu weichen Rampen verändert
- bei periodisch wiederkehrende Muster werden gelöscht, im ersten Beispiel wird sogar das gesammte Bild zu 0

- **Ergebnis**

Es ist sehr wichtig sich vorher klar zu machen was mit den Bildinformationen geschehen soll.

Will man einen Extremwert entfernen empfiehlt es sich den Medianfilter zu verwenden. Dies könnte z.B. der Fall sein, wenn man Fehler die bei der Übertragung des Bildes entstanden sind eliminieren will.

Um hingegen Kanten zu glätten empfiehlt es sich den Mittelwertfilter zu verwenden der Kanten bricht und daraus Rampen bildet.

Wenn man periodisch wiederkehrende Muster entfernen will, da sie Störungen sind, kann man beide Filter verwenden. Will man hingegen dieses Muster erhalten ist es nicht zu empfehlen einen der Filter anzuwenden

Mittelwertfilter

verwäscht Kanten
neue Grauwerte entstehen
mittlere Grauwert bleibt erhalten
kurze Rechenzeit
turen können beide nicht erhalten

Medianfilter

erhält Kanten
keine neuen Grauwerte
mittlere Grauwerte ändern sich
hoher Rechenaufwand

periodische Struk-

4.4 Kantendetektion

- Grundprinzip sind Ableitungen (meist erste und zweite) die diskretisiert werden

- Kanten beschreiben die Ränder von Objekten
- Anfang der Objekterkennung liefert Ausgangsdaten zur Ermittlung von Volumen/Flächen, Durchmesser oder Formen von Objekten
- basiert auf Filtern die konstante Bereiche unterdrücken und Änderungen verstärken
- **Eigenschaften:**

– **Verschiebungsinvarianz (Verschiebungsfreiheit):**

- * Kanten sollen dort im Bild markiert werden wo sie auftreten
- * erste Ableitung muss eine Phasenverschiebung von 90° haben
 - Extrema auf Null setzen
 - größte Steigungen auf Extremwerte abbilden
- ↪ antisymmetrische Filtermaske

$$\begin{aligned}
 h_{-n} &= -h_n \\
 -h_{-m,n} &= -h_{m,-n} = h_{mn} \\
 -h_{-m,n,o} &= -h_{m,-n,o} = -h_{m,n,-o} = h_{m,n,o}
 \end{aligned}$$

- * zentrale Nullen bei ungerader Filtermaskenlänge
- * zweite Ableitung bildet maximale Krümmungen im Grauwertbild auf Extrema im Merkmalsbild ab → Filtermaske symmetrisch

– **Unterdrückung des Mittelwertes:**

- * Kantenfilter sollte auf konstante Werte oder Veränderungen des durchschnittlichen Grauwertes nicht reagieren
- * die Summe der Einträge des Filters muss 0 sein
 - ⇒ $\frac{1}{2}[10 - 1] \curvearrowright \sum_{i=1}^3 x_i = 0$

– **Symmetrie (Transferfunktion (1D)):**

- * für einen Ableitungsoperator erster Ordnung ergibt sich eine Filtermaske ungerader Symmetrie $2r+1$ Koeffizienten
- Berechnung der Faltung:

$$g_n = \sum_{n'=1}^r h_{n'} \cdot (g_{n-n'} - g_{n+n'})$$

Berechnung der Transferfunktion:

$$\hat{h}(k) = 2i \sum_{v=1}^R h_v \sin(v\Pi(k))$$

- * für 2D muss gerade Symmetrie in Ableitungsrichtung gegeben sein

– **Isotropie:**

- * Operatorantwort sollte nicht von der Richtung der Kante abhängen ⇒ isotroper Kantendetektor

- **Gradientenfilter**

1. **Gradientenvektor**

- eindimensionale Differenzenoperatoren detektieren Kanten die senkrecht oder in Operatorrichtung verlaufen
- gesucht ist aber ein Filteroperator der Kanten unabhängig von der Orientierung nachweist → isotroper Kantendetektor

- Notation für Gradientenoperatoren

$$D = \begin{bmatrix} D_x \\ D_y \end{bmatrix}$$

$$D = \begin{bmatrix} D_x \\ D_y \\ D_z \end{bmatrix}$$

- Gradient ist ein Vektor dessen Betrag invariant ist gegenüber der Drehung des Koordinatensystems
- Berechnung des Betrag (2D)

$$|D| = [D_x \cdot D_x + D_y \cdot D_y]^{\frac{1}{2}}$$

· punktweise Multiplikation der Bildmatrize

- die Berechnung des Betrags ist rechenaufwendig → Annäherung

$$|D| \approx |D_x| + |D_y| = D'$$

- diese Approximation für kleine Wellenzahlen anisotrop (entlang der Diagonalen $\sqrt{2}$ mal empfindlicher)
- Grundproblem aller Kantendetektoren, dass sie nur auf diskreten Gittern approximiert werden können → falsche Kantenrichtung

2. Diskrete Differenzen erster Ordnung

- linearer Ansatz
- einfachster Ansatz für den Ableitungsoperator:

$$\frac{\varphi f(x_1, x_2)}{\varphi x_1} \approx \frac{f(x_1, x_2) - f(x_1 - \Delta x_1, x_2)}{\Delta x_1} \quad \text{rückwärts}$$

$$\approx \frac{f(x_1 + \Delta x_1, x_2) - f(x_1, x_2)}{\Delta x_1} \quad \text{vorwärts}$$

$$\approx \frac{f(x_1 + \Delta x_1, x_2) - f(x_1 - \Delta x_1, x_2)}{2\Delta x_1} \quad \text{symmetrisch}$$

- entspricht folgenden Filtermasken:

$$^-D_x = [1. \ -1]$$

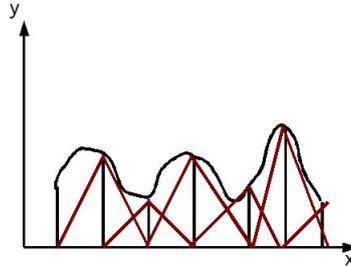
$$^+D_x = [1 \ -1.]$$

$$^-D_{2x} = \frac{1}{2}[10 \ -1]$$

- bezeichnet den zentralen Bildpunkt asymmetrischer Masken mit zwei Elementen
- Symmetrie zeigt nur die letzte Maske
- Lösung für die andern beiden → das Ergebnis nicht im rechten oder linken Bildpunkt speichern sondern zwischen den beiden (Verschiebung des Gitters um halbe Bildpunktdistanz)
- große Fehler für die Richtung des Gradientenfilter
- Fehler in Achsen- und Diagonalenrichtung sind Null dazwischen bis zu $\pm 10\hat{A}^\circ$

3. Kubischer Ableitungsoperator

- Darstellung durch kubische, interpolierende Splines (B-Splines)
- wenn man sich ein Grauwertbild im Querschnitt betrachtet, hat man für jeden Bildpunkt einen bestimmten Grauwert diesen kann man in einem Koordinatensystem wie folgt darstellen



um nicht die roten Linien als Funktion nehmen zu müssen approximiert man die Funktion durch eine weichere, um mehr Punkte für die Ableitung zu erhalten

4. Laplace Filter

- Kanten führen zu Nulldurchgängen der 2. Ableitung
→ Idee: die Summe der Quadrate der zweiten partiellen Ableitung als Kantendetektor
- kompakter Filter für die zweite Ableitung durch Verknüpfung von $-D_x$ und $+D_x$

$$D_x^2 = -D_x^+ D_x^-$$

$$[1 \ -2 \ 1] = [1 \ -1] * [1 \ -1]$$

2D

$$L = D_x^2 + D_y^2$$

$$L = [1 \ -2 \ 1] + \begin{bmatrix} 1 \\ -2 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

3D

$$L = D_x^2 + D_y^2 + D_z^2$$

$$L = [1 \ -2 \ 1] + \begin{bmatrix} 1 \\ -2 \\ 1 \end{bmatrix} + \begin{bmatrix} 1 \\ -2 \\ 1 \end{bmatrix}^z$$

3 Ebenen

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 1 & 0 \\ 1 & -6 & 1 \\ 0 & 1 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

5. regularisierte Kantendetektion

- bisherige Kantendetektoren reagieren empfindlich auf Rauschen
- zur Unterdrückung von Rauschen haben wir Mittelung kennengelernt
- Ziel Verbindung von Glättung parallel zu Kanten bzw. Rändern mit Ableitungsoperator zu kombinieren

- kleinste Operatoren dieser Idee

$$-D_x B_y^+ = \begin{bmatrix} 1. & -1 \end{bmatrix} * \frac{1}{2} \begin{bmatrix} 1. \\ 1 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 1. & -1 \\ 1 & -1 \end{bmatrix}$$

$$-D_y B_x = \begin{bmatrix} 1. \\ -1 \end{bmatrix} * \frac{1}{2} \begin{bmatrix} 1. & 1 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 1. & 1 \\ -1 & -1 \end{bmatrix}$$

- Probleme mit der Verschiebung bei geraden Filterlängen lassen sich mit Filterlänge umgehen → **Sobeloperator**

$$D_{2y} B_x^2 = \frac{1}{8} \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}$$

$$D_{2x} B_y^2 = \frac{1}{8} \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

- Verbesserung gegenüber der Ergebnisse ohne Glättung
- Kantendetektion mit der Gaußfunktion könnte auch genutzt werden bringt aber keine essentielle Verbesserung
- **Optimiertes Sobelverfahren**
folgende Filtermasken minimieren den Winkelfehler des Sobeloperators

$$\frac{1}{4} d_{2x}(3B_y^2 + I) = \frac{1}{32} \begin{bmatrix} 3 & 0 & -3 \\ 10 & 0 & -10 \\ 3 & 0 & -1 \end{bmatrix}$$

$$\frac{1}{4} d_{2y}(3B_x^2 + I) = \frac{1}{32} \begin{bmatrix} 3 & 10 & 3 \\ 0 & 0 & 0 \\ -3 & -10 & -3 \end{bmatrix}$$

- **LoG - Filter (Laplace of Gaussian)**
Laplace Filter verstärken sehr stark Rauschen → sinnvoll Gaußfilter vor anzustellen

$$L * B^2 = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} * \frac{1}{16} \begin{bmatrix} 0 & 1 & 2 & 1 & 0 \\ 1 & 0 & -2 & 0 & 1 \\ 2 & -2 & -8 & -2 & 2 \\ 1 & 0 & -2 & 0 & 1 \\ 0 & 1 & 2 & 1 & 0 \end{bmatrix}$$

4.5 Serie 4

1. Gegeben sei das Bild (5,5,4,3,2,1,0,0,0,6,0,0,0,0,1,3,1,0,0,0,0,7,7,7,...)
 - erstellen sie aus dem Bild ein Grauwertprofil
 - berechnen sie die erste und zweite Ableitung mittels:

$$\frac{\varphi f}{\varphi x} = f(x+1) - f(x)$$

und

$$\frac{\varphi^2 f}{\varphi^2 x} = f(x+1) + f(x-1) - 2f(x)$$

- machen sie sich anhand der berechneten Werte noch einmal die Eigenschaften der ersten und zweiten Ableitung bzgl. Kanten klar

2. Gegeben sei das folgende Bild:

0	1	2	0	4	5
0	0	1	1	4	5
0	2	0	4	5	4
0	0	5	4	6	6
0	0	6	6	5	6

Verwenden sie folgende Sobel - Filter um vertikale und horizontale Kanten im Bild zu ermitteln.

-1	0	1
-3	0	3
-1	0	1
-1	-3	-1
0	0	0
1	3	1

Aus Gründen fehlender Zeit habe ich meine Mitschrift aus dem Seminar als extra Datei auf meine Homepage gestellt (seminar4.tar.gz).

4.6 Texturanalyse

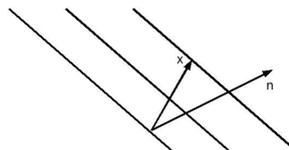
• Struktensoren

- man benötigt ein Maß für die “Gleichförmigkeit“ in der Nachbarschaft eines Pixels
- Idee das Bild mit einer lokalen Fensterfunktion zu multiplizieren um Texturen zu erkennen ...
- **einfache Nachbarschaften** sind Nachbarschaften bei denen sich die Grauwerte nur in eine Richtung verändern, die über eine Einheitsnormale n (Orientierungsvektor) definiert ist.

$$2D : \quad p(x) = \bar{p}(x^T n); x \in U \subset \mathbb{R}^2$$

$$3D : \quad v(x) = \bar{v}(x^T n); x \in U \subset \mathbb{R}^3$$

n bewirkt, dass die Grauwerte nur noch von einer skalaren Größe, der Koordinaten in Richtung n abhängt



- ein Gradient zeigt immer in die Richtung der größten Zunahme bzw. Veränderung
- die Suche nach nur einem gemittelten Vektor n in der Umgebung um ein Pixel (Voxel) reicht nicht
 - keine Unterscheidung von unkorreliertem Rauschen und konstanten Grauwerten
 - n immer 0
 - Verwendung einer tensoriellen Größe zur Beschreibung der Werte um das Pixel x

- betrachte die Umgebung um ein Pixel mit

$$\int w(x - x') (\nabla g(x') n)^2 d^3 x'$$

- $w(x - x')$ beschreibt die Fensterfunktion um ein Pixel x
- w steht für Window das auf 0 abfällt
- anders geschrieben

$$n^T J n \rightarrow \text{maximal}$$

$$\text{mit } J = \int w(x - x') (\nabla g(x') \nabla g(x')^T) d^3 x'$$

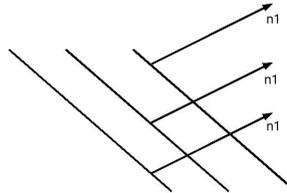
für die Komponenten gilt:

$$J_{pq} = \int_{\mathbb{R}^3} w(x - x') \left(\frac{\varphi g(x')}{\varphi x'_p} \frac{\varphi g(x')}{\varphi x'_p} \right) d^3 x'$$

- J ist ein symmetrischer Tensor = **Strukturtensor**

$$J = \begin{pmatrix} n_1' & n_2' & n_3' \end{pmatrix} \begin{pmatrix} J'_{11} & & \\ & J'_{22} & \\ & & J'_{33} \end{pmatrix} \begin{pmatrix} n_1^T \\ n_2^T \\ n_3^T \end{pmatrix}$$

- * entsteht durch Drehung der Diagonalform
- * n_1', n_2', n_3' stehen senkrecht zu einander
- * $J'_{11} > J'_{22} > J'_{33}$ ergibt n_1' die Normale des Eigenvektors von J zum größten Eigenwert J'_{11}



- Klassifizierung der Umgebung um ein Pixel (Voxel) Eigenwerte = 0

$$J \underbrace{k_w}_{\text{Eigenvektor von } J} = \underbrace{\lambda_w}_{\text{Eigenwert}} k_w$$

Wenn Eigenwert = 0 \curvearrowright Grauwert in Richtung des korrespondierenden Eigenvektors ändert sich nicht

- **Kohärenz**

- * Ziel ist die Unterscheidung von Bereichen mit konstanten Grauwerten und isotropen Grauwertstrukturen ohne Vorzugsrichtung
- * Definition Kohärenz

$$c = \frac{J'_{11} - J'_{22}}{J'_{11} + J'_{22}}$$

- * liegt einfache Nachbarschaft vor $c = 0$
- * isotrope Verhältnisse $c = 1$

– **Implementierung**

- * Diskretisierung von Ableitung und Integral nötig
- * pixelweise Multiplikation nötig
- * D_x, D_y, D_z sein Ableitungsoperatoren und B ein isotroper Glättungsoperator (z.B. Binomialfilter)

$$J_{pq} = B(D_p \cdot D_q)$$

$$p, q = 1, 2, 3$$

Die Komponente J_{pq} des Tensors wird durch separate Faltung des Bildes mit D_p und D_q pixelweiser Multiplikation (\cdot) der beiden gefalteten Bilder und Glättung des Ergebnisses gebildet.

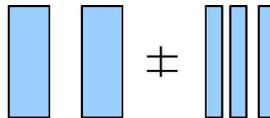
- * auf Grund der Symmetrie benötigt man $\frac{W(W+1)}{2}$ solche Berechnungen in der jeweiligen Dimension (2D = 3, 3D = 6)

– **Fehler**

- * Genauigkeit der Orientierungsbestimmung (Richtung von n) ist Hauptkriterium
- * Abwägung zwischen Rechenaufwand und Genauigkeit
- * Ziel ist ein möglichst robuster Algorithmus (Unempfindlichkeit gegenüber Rausch und anderen Störungen)

• **Lokale Wellenzahl und Phase**

- gesucht ist die Größe der lokalen Wellenzahl



- weitere Erklärung am eindimensionalen periodischen Signal:

$$g(x) = g_0 \cos(kx)$$

- die Phase des Signals

$$\Phi(x) = kx$$

k ist die Wellenzahl des Signals

– **Hilbertoperator**

- * verschiebt die Phase bzw. verzögert das Signal der ihm übergebenen Funktion

das Signal

$$H(g(x)) = -g_0 \sin(kx)$$

die Phase

$$\Phi(g(g)) = \arctan\left(\frac{-H(g(x))}{g(x)}\right)$$

- * durch die Verschiebung/Verzögerung wird die Amplitude nicht verändert

- * Fouriertransformierte (Transferfunktion)

$$\hat{h}(k) = \begin{cases} i & k > 0 \\ 0 & k = 0 \\ -i & k < 0 \end{cases}$$

$$h(x) = -\frac{1}{\Pi x}$$

es ergibt sich die Hilberttransformation mit

$$H(g) = h * g = \frac{1}{\Pi} \int_{-\infty}^{\infty} \frac{g(x')}{x - x'} dx'$$

- * Diskretisierung kann wegen der unendlich ausgedehnten Fouriertransformierten nicht exakt sein
- * Verschiebung um 90° → antisymmetrische Filtermaske

$$\hat{h}(k) = 2i \sum_{v=1}^R h_v \sin((2v-1)\Pi k)$$

ergibt folgende Filtermaske

$$[h_R, 0, \dots, h_1, 0, -h_1, \dots, 0, -h_R]$$

mit $R=4$

$$h_1 = 191, h_2 = 630, h_3 = 1683, h_4 = 6208$$

• Statistik erster Ordnung

- **Texturen** sind komplexe Muster, die in der Regel durch Wiederholung eng begrenzter Bereiche entstehen.
- menschliche Auge kann Objekte in einem Bild auf Grund von Texturen unterscheiden
- Ziel ist dies auch dem Bildverarbeitungsprogramm bei zu bringen
→ geeignetes Werkzeug zur Beschreibung der Eigenschaften von Texturen benötigt
- Zwei Klassen von Texturmerkmalen
 1. Rotations- und größeninvariante Merkmale
 - * die Merkmale werden durch Statistikmethoden geliefert
 - * Merkmale hängen vom Histogramm ab
 - * Merkmal z.B. Varianz

$$v_{m,n} = \frac{1}{P-1} \sum_{m'=-r}^r \sum_{n'=-s}^s (g_{m-m',n-n'} - \bar{g}_{m,n})^2$$

F ein Fenster mit Größe

$$(2r+1) \times (2s+1)$$

P Anzahl der Pixel im Fenst

$$P = (2r+1)(2s+1)$$

$\bar{g}_{m,n}$ ist der Mittelwert

$$\bar{g}_{m,n} = \frac{1}{P} \sum_{m'=-r}^r \sum_{n'=-s}^s g_{m-m',n-n'}$$

* es gilt also

$$V = R(I \cdot I) - (R \cdot R)$$

R ist Rechtecksoperator
 I ist Identität
 $\cdot, -$ Punktoperationen

2. Rotations- und größenabhängige Merkmale
 - * Ansätze über die Kohärenz, lokale Wellenzahl oder Laplacepyramide

4.7 Segmentierung

- Ziel ist die Extraktion der Abbilder von Objekten im Bild

1. Pixelorientierte Segmentierung

- betrachtet nur den Grauwert des Pixel/Voxel
- man betrachtet Histogramm der Grauwerte und ermittelt einen (mehrere) Schwellenwert(e) \Rightarrow Objekte und Hintergrund trennen
- Schwellenwertwahl ist kritisch bes. wenn Größe und Form der Objekte wichtig
- erster Schritt sollte immer Ausgleich der Helligkeitsunterschiede auszugleichen
- Probleme:

- * unterschiedliche Grauwerten für verschiedene Objekte
- * Größenänderungen durch schlechte Schwellenwerte und Lichtverhältnisse

2. Kantenbasierte Segmentierung

- Suche nach Maxima der ersten Ableitung oder Vorzeichenwechsel der Zweiten \Rightarrow zeigen Objektgrenzen
- Ansatz mit 2. Ableitung unabhängig von linearen Beleuchtungsänderungen und Intensität der Kanten nicht wichtig
- ist ein sequentieller Prozess
- das Bild wird zeilenweise durchlaufen
- wenn ein Maximum (Nulldurchgang) des Gradienten gefunden ist, wird diese Kontur solange verfolgt, bis man wieder am Ausgangspunkt ankommt
- man versucht also Nachbarvoxel(-pixel) zu finden, dass ebenfalls Teil der Kante ist.

3. Regionenorientierte Verfahren

(a) Regionenwachstum (Region Growing)

- aus einem "Saatvoxel" (seed voxels) wird sukzessive eine Region ausgedehnt
- zu jeder Region werden Randvoxel bestimmt und deren Nachbarn getestet
- wenn Voxel noch zu keiner Region gehört und sein Wert dem Mittelwert der Region entspricht wird es aufgenommen
- Problem der Anwender muss bestimmen wie sehr der Wert vom Mittelwert abweichen darf

(b) Regionentrennen (Region Splitting)

- gesamtes Bild als eine Region sehen
- dann in Oktanten zerlegen, sobald der Wert eines Voxels zu sehr vom Mittelwert abweicht
- fortsetzen bis Abweichung klein genug \Rightarrow max. isoliertes Voxel

- Vorteil die Regionen können durch einen Octree (Baum bei dem jeder Knoten max. 8 Kinder hat) beschrieben werden
- (c) **Regionenverschmelzung (Region Merging)**
 - jeder Voxel als eigene Region
 - Verschmelzen wenn die Abweichung vom Mittelwert unter einer bestimmten Grenze liegt
 - in der Regel wird Schicht - Zeile - Spalte vorgegangen
 - als erstes in positiver z-Richtung nach geeigneten Nachbarregionen für Fusionsprozess suchen
- (d) **Regionentrennen und -verschmelzen (Region Split - Merge)**
 - nach dem trennen erhält man meist ein übersegmentiertes Bild
 - durch Verschmelzen von “wenig“ abweichenden Regionen wird bessere Segmentierung erreicht.
 - wenn mehrere Regionen zur Auswahl stehen wird die mit der geringsten Abweichung gewählt
 - Nachteil Vereinigungsschritte sehr aufwendig
- (e) **Pyramid - Linkung**
 - alle anderen Verfahren nur gut wenn uniforme Objekte und Hintergründe vorhanden sind ohne fließende Übergänge
 - wenn dies nicht gegeben ist erzeugt man erst ein Merkmalsbild und segmentiert dieses
 - es sollen nur die Teile einem Objekt zugeordnet werden die ihm zweifelsfrei zu gehören \Rightarrow in den Übergangsbereichen keine Glättung
 - mehrstufiger Prozess nötig, da Objekte erst gefunden werden müssen bevor man die Maske verkleinern kann
 - **Algorithmus von Burt**
 - i. erzeugen einer Gaußpyramide durch Glättungsfilter (im 1D ein Rechteckfilter mit Länge 4 und halbierte Auflösung)
 - ii. jeder Bildpunkt einer Ebene gehört zu 2 Punkten der nächst Ebene mit geringerer Auflösung
Zuordnung erfolgt nun zu dem Punkt der nächsten Ebene mit näherem Grauwert \Rightarrow Baum entsteht
 - iii. die Werte der Vaterknoten werden als Mittelwerte der zugehörigen Blätter undefiniert
man läuft einmal von den Blättern bis zur Wurzel
wiederhole die letzten beiden Schritte bis zur stabilen Lösung
Verfahren an dem Beispiel aus der Vorlesung im Anhang Abbildung 3 (!eigentlich müsste man hier noch mind. einen Durchgang machen!)

4. Modellbasierte Segmentierung

- menschliche Auge als Vorbild
- wenn man Informationen über die im Bild vorhandenen Objekte hat kann man wesentlich besser segmentieren
- **Houghtransformation**
 - * Suche nach geraden Kanten im Bild
 - * jeder Punkt einer Kante liefert Informationen über die Geradengleichung der Kante
 - * alle Kantenpunkte benutzen um Parameter der Kante zu approximieren

- * sein (x_n, y_n) Punkte auf einer Kante so gilt:

$$y_n = a_0 + a_1 x_n$$

$$a_1 = \frac{y_n}{x_n} - \frac{1}{x_n} a_0$$

- * jeder Kantenpunkt liefert eine Gerade im Raum
- * Abbildung $(x_n, y_n) \rightarrow (a_0, a_1)$, heißt Houghtransformation mit Geradengleichung:

$$nx = d$$

bzw.

$$x \cos(\Theta) + y \sin(\Theta) = d$$

- * sehr rechenaufwendig
- * Beschleunigung der Houghtransformation ausgehend von Struktursoren \rightarrow neben Kanteneigenschaften auch Kantenrichtung ermittelbar
- Nachteil ist, dass das zu segmentierende Objekt bis auf einige Parameter bekannt sein muss \Rightarrow zu unflexibel

5. Globale Optimierungsansätze

- vom zu segmentierenden Objekt sollen Eigenschaften verlangt werden, die seine Form nicht direkt festlegen
- man fordert eine flexible Ähnlichkeitsbedingung der Bilddaten zu einem im Rahmen der Segmentierung gewonnenen Modell des Objektes und ferner in der Regel eine Glattheitsbedingung an das Modell
- globales Optimierungsproblem
- für die Rekonstruktion von Objekten gibt es Grenzen, da manche Objekte in einem Bild nicht vollständig oder verfälscht zu erkennen sind (Objekt hatte eine gewisse Höhe die Schatten wirft \rightarrow Objektkante nicht mehr von Schatten zu unterscheiden)
- Merkmale leisten nur begrenzt eine Unterscheidung der Bilddaten \rightarrow falsches Modell kann entstehen
- **Variationsansätze**
 - * innerhalb der Objekte ändern sich die Werte nur langsam
 - * Ränder eines Objektes bestehen aus glatten Kurven bzw. Kanten und sind geschlossen
 - * Objektgrenzen liegen nahe an den ermittelten Grenzen

6. Diffusionsmodelle

- die Erzeugung eines glatten Modells wird als Diffusionsprozess betrachtet
- der Ähnlichkeitsterm spielt dabei die Rolle einer zusätzlichen Quelle
- es ergibt sich die Diffusions - Reaktions - Gleichung (Euler - Lagrange - Gleichung)

$$\sum_{p=1}^3 \frac{\varphi}{\varphi x_p} L_{f_{x_p}} - L_f = 0$$

- **Inhomogene Diffusion**
 - * Glattheitstherm wird so modifiziert, dass die Diffusion von Kanten abhängt
 - * es kann passieren, dass Kanten verwaschen bleiben
- **Anisotrope Diffusion**
 - * es wird zugelassen, dass entlang der Kanten auch Diffusion stattfindet

5 Hinweise

Informationen zur Klausur Termin: Di, 19.07.2005, 7:30, H 22

- Hilfsmittel: nicht-programmierbarer Taschenrechner
- Mitbringen: Schreibzeug (Kugelschreiber und unbeschriftetes Papier)
- Stoff: Grundlage der Klausur sind sowohl die Übungen (ca 60%) als auch die Vorlesung (ca 40%)
- Stoff der Vorlesung: Kapitel 2 bis (einschließlich) Kapitel 4.2
- theoretische Herleitungen nicht Klausurrelevant
- Ergebnisse von Herleitungen sind wichtig
- Eigenschaften sind wichtig
- Anforderungen (an Filter) $f_{\tilde{A}}^{\frac{1}{4}}r$ bestimmte Aufgaben sind wichtig
- Überblick und Vergleiche zwischen verschiedenen Filtern wichtig
- Grundlegende Ideen hinter den Verfahren wichtig

6 Anhang

Darstellung
von \mathbb{C}

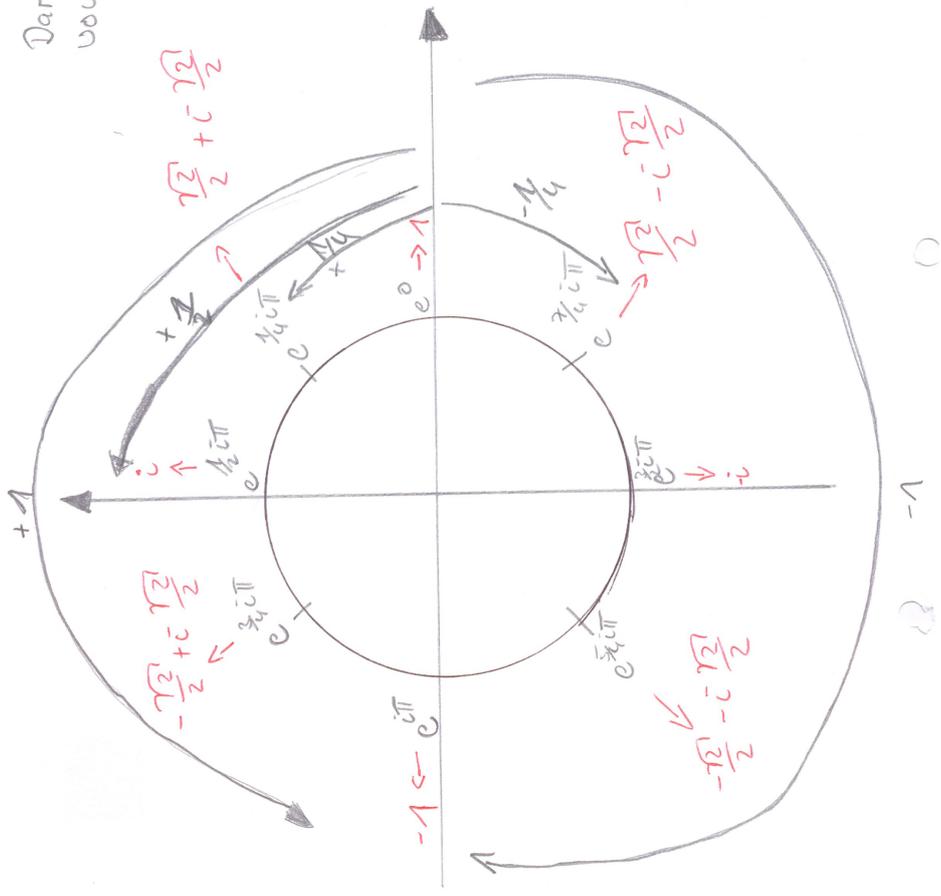


Abbildung 1: Darstellung der Komplexenzahlen

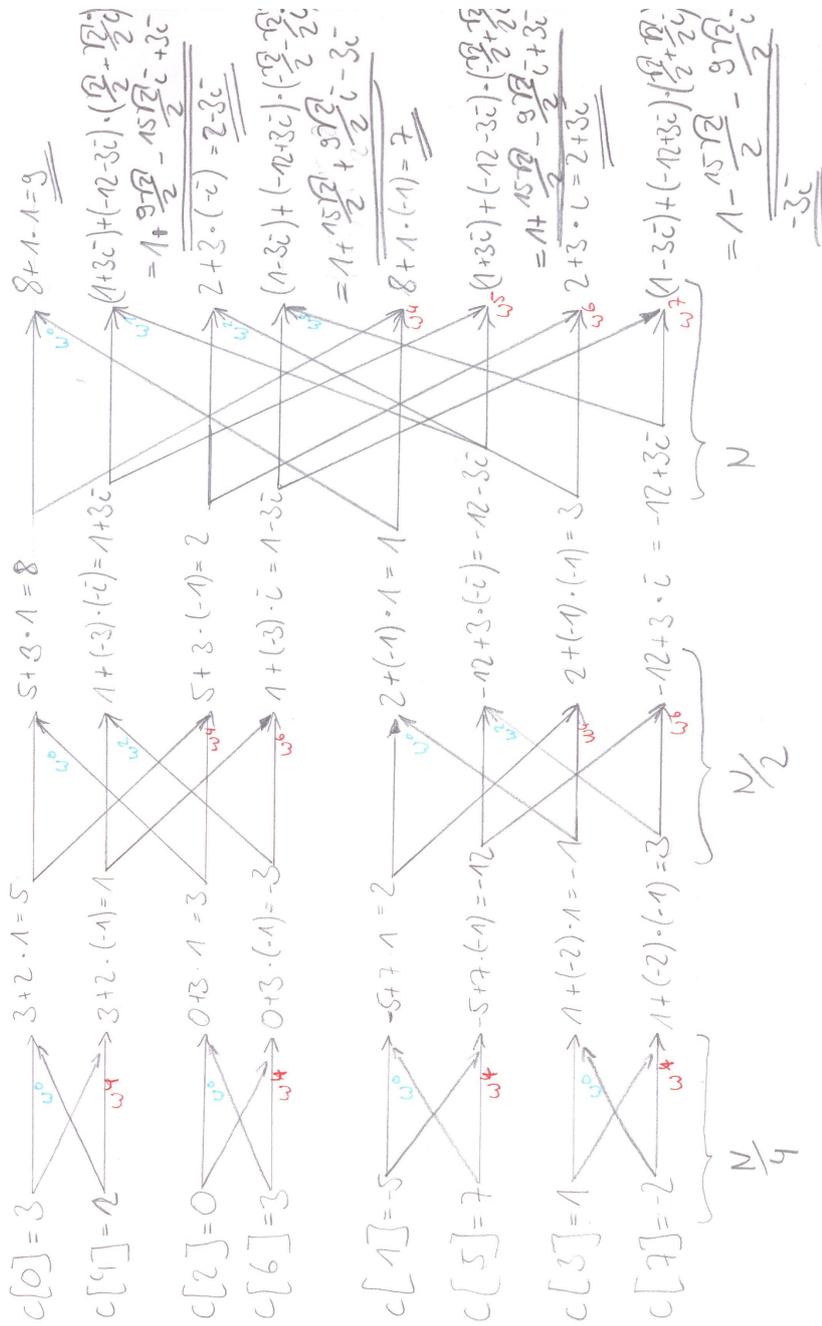


Abbildung 2: Serie 2

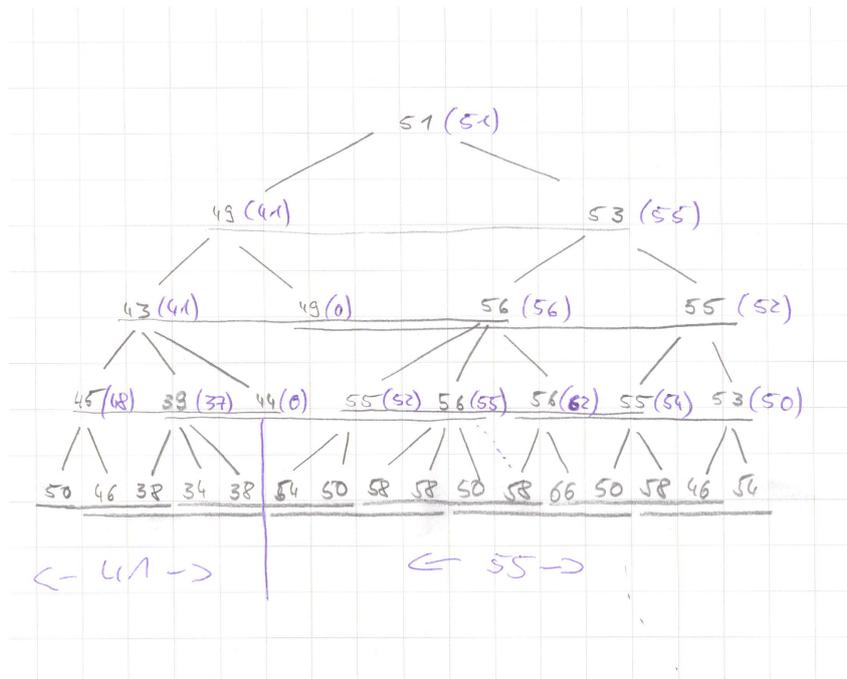


Abbildung 3: Beispiel aus der Vorlesung zu Segmentierung (Danke Alex!)