

8. Operationen auf Rasterdaten

1. **Punktautonome Grauwertoperationen**
2. **Lineare Ortsfilter**
3. **Abstandstransformationen**
4. **Kombination von Bildern**

8.1 Punktautonome Grauwertoperationen (I)

Punktautonome Grauwertoperationen

- Eine *punktautonome Grauwertoperation* formt die Grauwerte $d(x,y)$ eines Eingangsbildes durch Anwendung einer Funktion $f(d)$ in die Grauwerte des Ausgangsbildes um.
- Die Funktion $f(d)$ wird als *Transfercharakteristik* (TC) bezeichnet. TC lässt sich als Tabelle (d_i, g_j) , $i = 0, \dots, d_{max}$ repräsentieren.
- Das Histogramm $h(d)$ des Eingangsbildes wird durch die Funktion $f(d)$ in das Histogramm $h(g)$ transformiert:

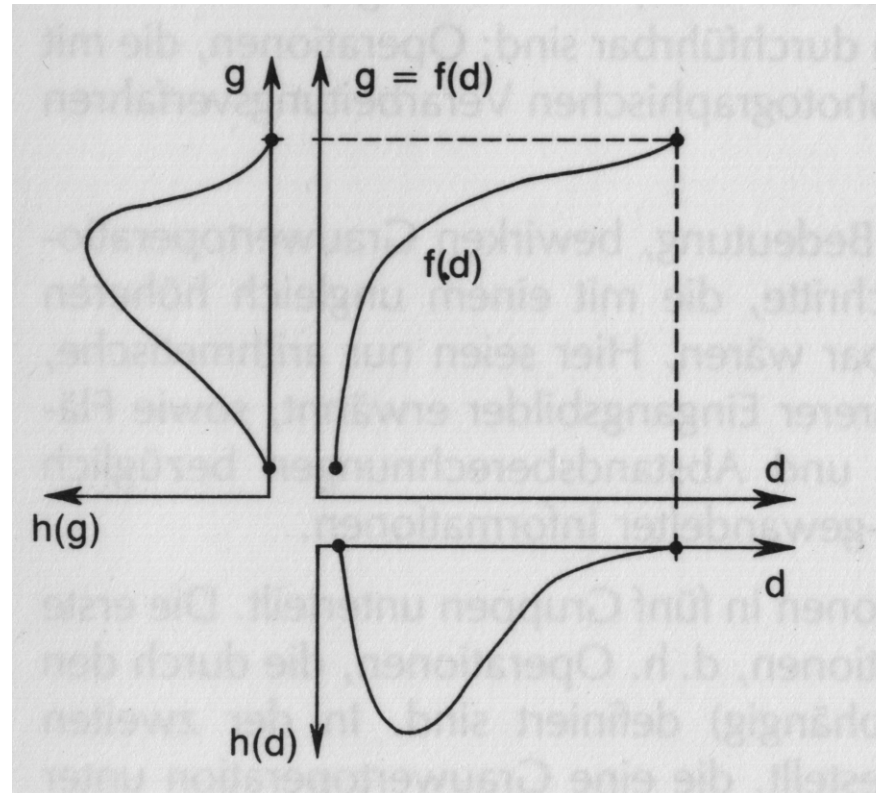
$$h(g) = \sum_{d=0}^{d_{max}} a(d, g) \cdot h(d)$$

mit

$$a(d, g) = \begin{cases} 1 & \text{für } g = f(d) \\ 0 & \text{sonst} \end{cases}$$

8.1 Punktautonome Grauwertoperationen (II)

Einfluss der TC $f(d)$ auf das Histogramm $h(d)$



Zusammenhang von $h(d)$, $f(d)$ und $h(g)$

8.1 TC Generierung (I)

Frage

Wie wird die Tabelle der TC mit Werten gefüllt?

Lösungsansatz

- Vorgabe einiger weniger Grauwertpaare (d_i, g_i)
- Berechnung der übrigen Grauwertpaare der TC durch Interpolation

Lineare Transfercharakteristika

- Für jedes lineare Segment der TC Eingabe zweier Wertepaare

$$TC(d_i, g_i, d_{i+1}, g_{i+1})$$

- Aufstellen der Geradengleichung

$$f(d) = g_i + \frac{(g_{i+1} - g_i)}{(d_{i+1} - d_i)} \cdot (d - d_i)$$

- Einsetzen aller Werte d , $d_i \leq d \leq d_{i+1}$ liefert die fehlenden g -Werte
- Die Steigung der Geraden steuert den Grad der Grauwertdehnung bzw. -stauchung und damit den Kontrast des Ausgangsbildes.

8.1 TC Generierung (II)

Beispiel

- Drei lineare TC Segmente:
 - $TC(0,0,100,20)$: Steigung = 0.2
 - $TC(100,20,200,100)$: Steigung = 0.8
 - $TC(200,100,255,100)$: Steigung = 0
- TC Tabelle

0	0
1	0
...	...
100	20
101	21
...	...
200	100
...	...
255	100

8.1 Äquidensitenbildung (I)

Äquidensiten

- Eine *Äquidensite* ist eine Menge benachbarter Pixel eines Bildes, die denselben Grauwert besitzen.
- Äquidensitenbildung ist z.B. nützlich für die Elimination von Rauschen in Rasterbildern oder für die Reduktion der Anzahl ihrer Grauwertstufen.

TC zur Äquidensitenbildung

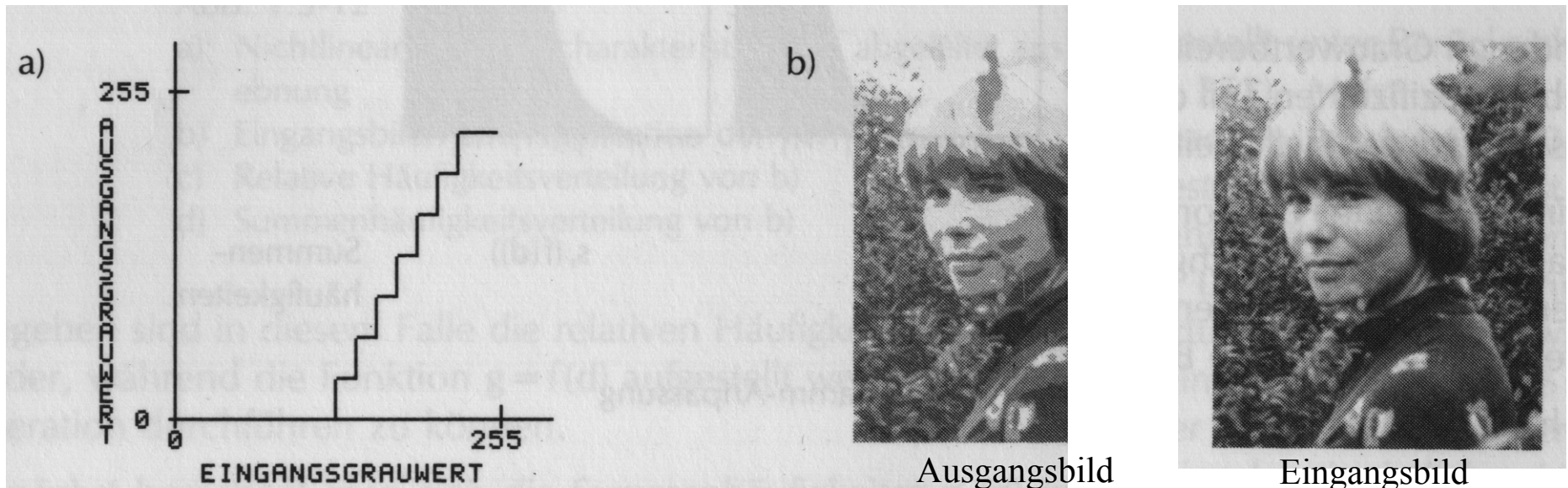
- Man benutzt eine TC der Form (d_i, g_i, d_{i+1}, g_i) , d. h. man setzt $g_{i+1} = g_i$.
⇒ bildet den Grauwertbereich $[d_i, d_{i+1}]$ auf den Grauwert g_i ab
- Annahme: benachbarte Pixel des Eingangsbildes besitzen ähnliche Grauwerte
- Dann wird benachbarten Pixeln des Ausgangsbildes i. A. derselbe Grauwert zugeordnet (Äquidensiten).

8.1 Äquidensitenbildung (II)

Beispiel 1

- Gegeben sei ein Eingangsbild mit den Grauwerten $0, \dots, 255$.
- Gesucht ist ein Ausgangsbild mit einer reduzierten Anzahl von Grauwertstufen:

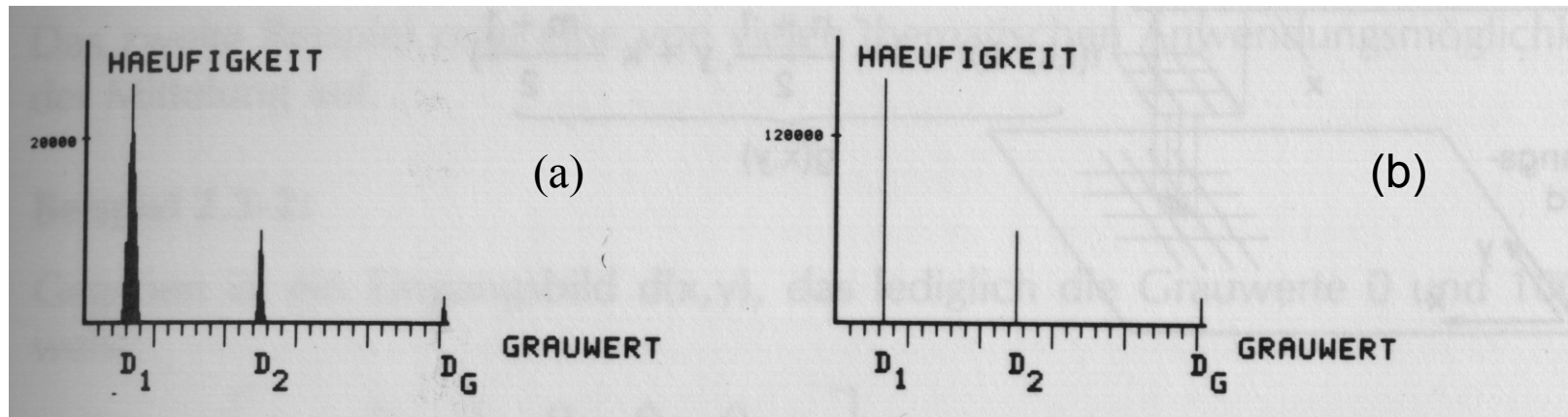
TC(0, 0, 126, 0)	--> 0
TC(127,12,127,12)	--> 12
TC(128,32,143,32)	--> 32
...	
TC(208,192,223,192)	--> 192
TC(224,224,255,224)	--> 224



8.1 Äquidensitenbildung (III)

Beispiel 2

- Gegeben sei eine Vorlage (z.B. Karte) mit den Grauwertstufen D_i , $i = 1, \dots, G$, und dem Histogramm (b).
- Aufgrund von Rauscheffekten bei der Datenerfassung erhält man aber ein Eingangsbild mit dem Histogramm (a).



- Mit Hilfe einer geeigneten TC lässt sich das Rauschen entfernen, d.h. ein Ausgangsbild mit dem erwarteten Histogramm erhalten:

$$\begin{aligned} &TC(0, D_1, (D_1 + D_2)/2, D_1) \quad \dots \\ &TC((D_{i-1} + D_i)/2, D_i, (D_i + D_{i+1})/2, D_i) \quad \dots \\ &TC((D_{G-1} + D_G)/2, D_G, 255, D_G) \end{aligned}$$

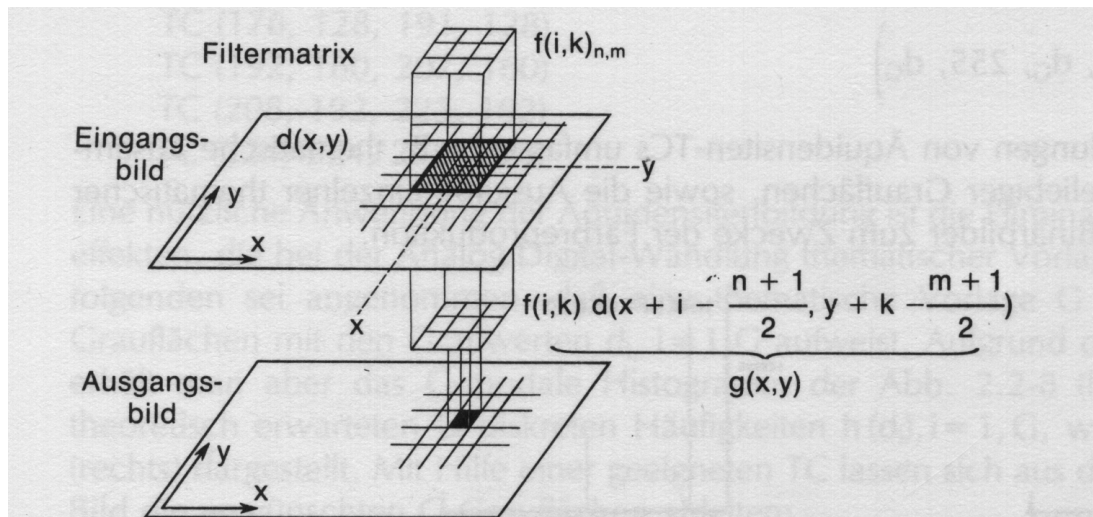
8.2 Lineare Ortsfilterungen (I)

Definition

- Eine *lineare Ortsfilterung (Faltung)* ist eine Funktion $g(x,y)$, die ein Eingangsbild $d(x,y)$ folgendermassen transformiert:

$$g(x,y) = \sum_{i=1}^n \sum_{k=1}^m f(i,k) \cdot d\left(x + i - \frac{n+1}{2}, y + k - \frac{m+1}{2}\right) + konst$$

- Die Koeffizienten $f(i,k)$ werden als *Filterkoeffizienten* bezeichnet und bestimmen den Typ der Filterung. Die Matrix der $f(i,k)$, $1 \leq i \leq n$, $1 \leq k \leq m$, ist die *Filtermatrix*.
- Die Werte n und m sind meist ungerade, sodass die Filtermatrix über (x,y) zentriert ist.



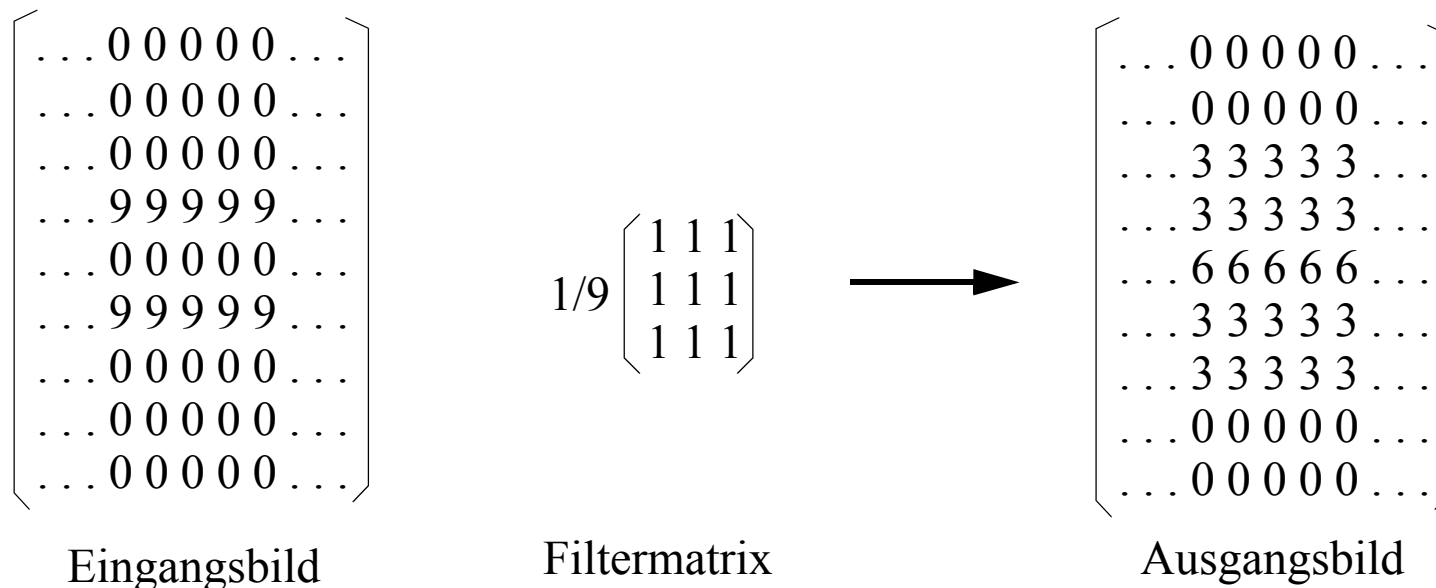
Lineare Ortsfilterung
mit $n = m = 3$

8.2 Lineare Ortsfilterungen (II) Mittelungen

Mittelungen

- Mit den Filterkoeffizienten $f(i,k) = 1/nm$ weist man jedem Pixel den Mittelwert der Grauwerte in seiner nm -Umgebung zu.
- Bei Mittelungen gilt $konst = 0$.

Beispiel 1



8.2 Lineare Ortsfilterungen (III)

Beispiel 2

$$\begin{bmatrix} \dots & 0 & 0 & 0 & 0 & 0 & \dots \\ \dots & 0 & 0 & 0 & 100 & 0 & \dots \\ \dots & 0 & 100 & 100 & 100 & 0 & \dots \\ \dots & 0 & 100 & 100 & 100 & 0 & \dots \\ \dots & 0 & 0 & 0 & 0 & 0 & \dots \end{bmatrix}$$

Eingangsbild
(100 = besiedelt)

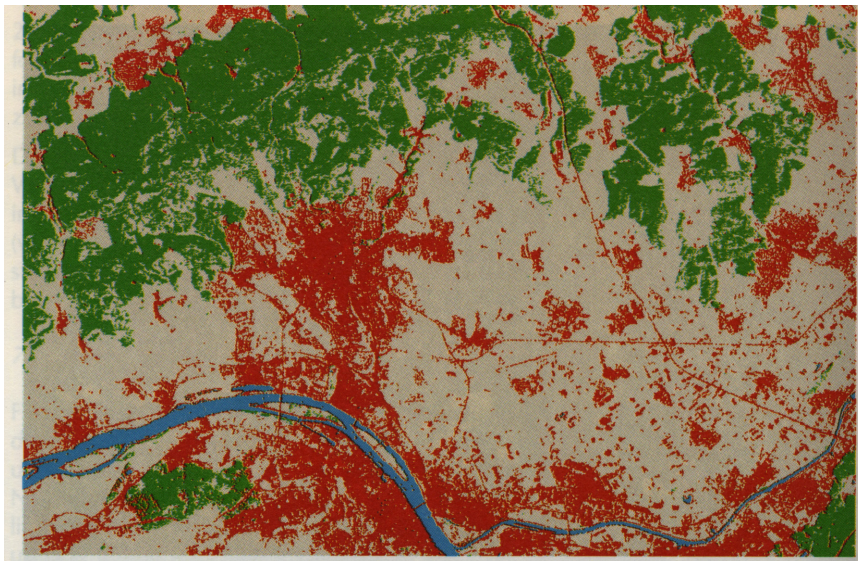
$$\frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Filtermatrix

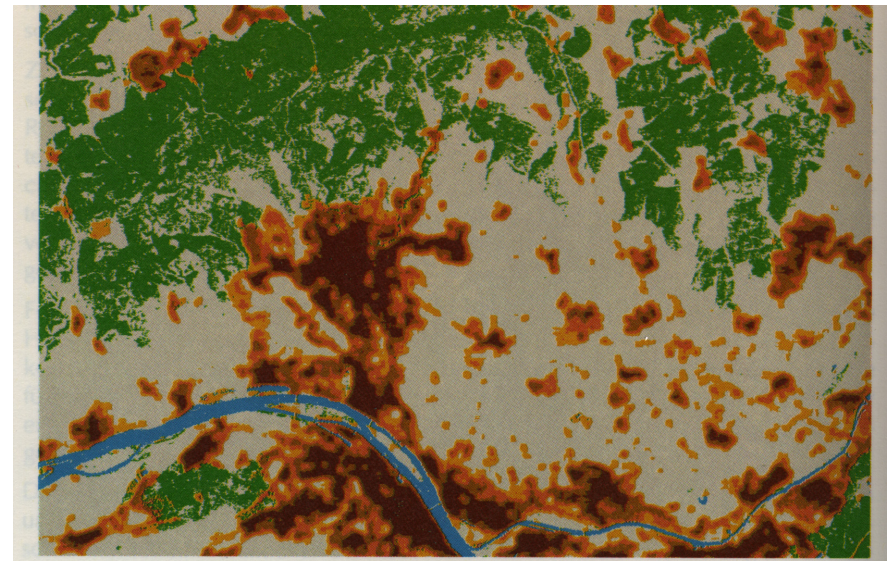


$$\begin{bmatrix} \dots & 0 & 0 & 11 & 11 & 11 & \dots \\ \dots & 11 & 22 & 44 & 33 & 22 & \dots \\ \dots & 22 & 44 & 77 & 55 & 33 & \dots \\ \dots & 22 & 44 & 66 & 44 & 22 & \dots \\ \dots & 11 & 22 & 33 & 22 & 11 & \dots \end{bmatrix}$$

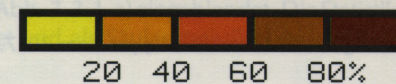
Ausgangsbild
(Besiedlungsdichte
in 3x3 Agglomerationen)



Eingangsbild (rot = besiedelt)



Ausgangs-
bild



Besied-
lungsdichte

8.2 Lineare Ortsfilterungen (IV)

Medianfilterung

- Die $n \times m$ -Umgebung des Eingangspixels $d(x,y)$ wird nicht gemittelt, sondern es wird der Median berechnet und als $g(x,y)$ gewählt.
- Das Ausgangsbild ist nicht so unscharf wie das bei der Mittelung entstehende, da "Ausreisser" bei den $d(x,y)$ nicht so stark ins Gewicht fallen.
- Die Medianfilterung wird verwendet, um "Störpixel" in thematischen Karten zu beseitigen und sie damit übersichtlicher zu machen.

Gradientenfilterungen

- Eine *Gradientenfilterung* ist eine Filterung, deren Filterkoeffizienten $f(i,k)$ so gewählt werden, dass homogene Grauwertbereiche einen Wert ≈ 0 erhalten, während an Grauwertsprüngen Werte $\gg 0$ oder $\ll 0$ auftreten.
- Bei Gradientenfilterungen setzt man (falls $d_{\max} = 255$) $konst = 127$.
- Gradientenfilterungen können dazu verwendet werden, ein Bild in Segmente mit homogenen Grauwerten zu zerlegen. Dies ist ein Schritt bei der *Vektorisierung* von Rasterbildern, d.h. der Umwandlung eines Rasterbilds in ein Vektorbild.

8.2 Lineare Ortsfilterungen (V)

Verschiedene Gradientenfilterungen

- Die Filtermatrix $f(i,k)_{1,2} = (-1 \ +1)$ liefert eine Approximation der *ersten partiellen Ableitung in x*, $\delta d(x,y)/\delta x$.
- Eine Approximation der Summe der zweiten partiellen Ableitungen liefert der *Laplace-Gradient*

$$f(i,k)_{3,3} = \begin{pmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{pmatrix}$$

$$\begin{pmatrix} \dots & 0 & 0 & 0 & 0 & 0 & \dots \\ \dots & 0 & 0 & 0 & 0 & 0 & \dots \\ \dots & 3 & 3 & 3 & 3 & 3 & \dots \\ \dots & 3 & 3 & 3 & 3 & 3 & \dots \\ \dots & 6 & 6 & 6 & 6 & 6 & \dots \\ \dots & 3 & 3 & 3 & 3 & 3 & \dots \\ \dots & 3 & 3 & 3 & 3 & 3 & \dots \\ \dots & 0 & 0 & 0 & 0 & 0 & \dots \\ \dots & 0 & 0 & 0 & 0 & 0 & \dots \end{pmatrix}$$

Eingangsbild

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

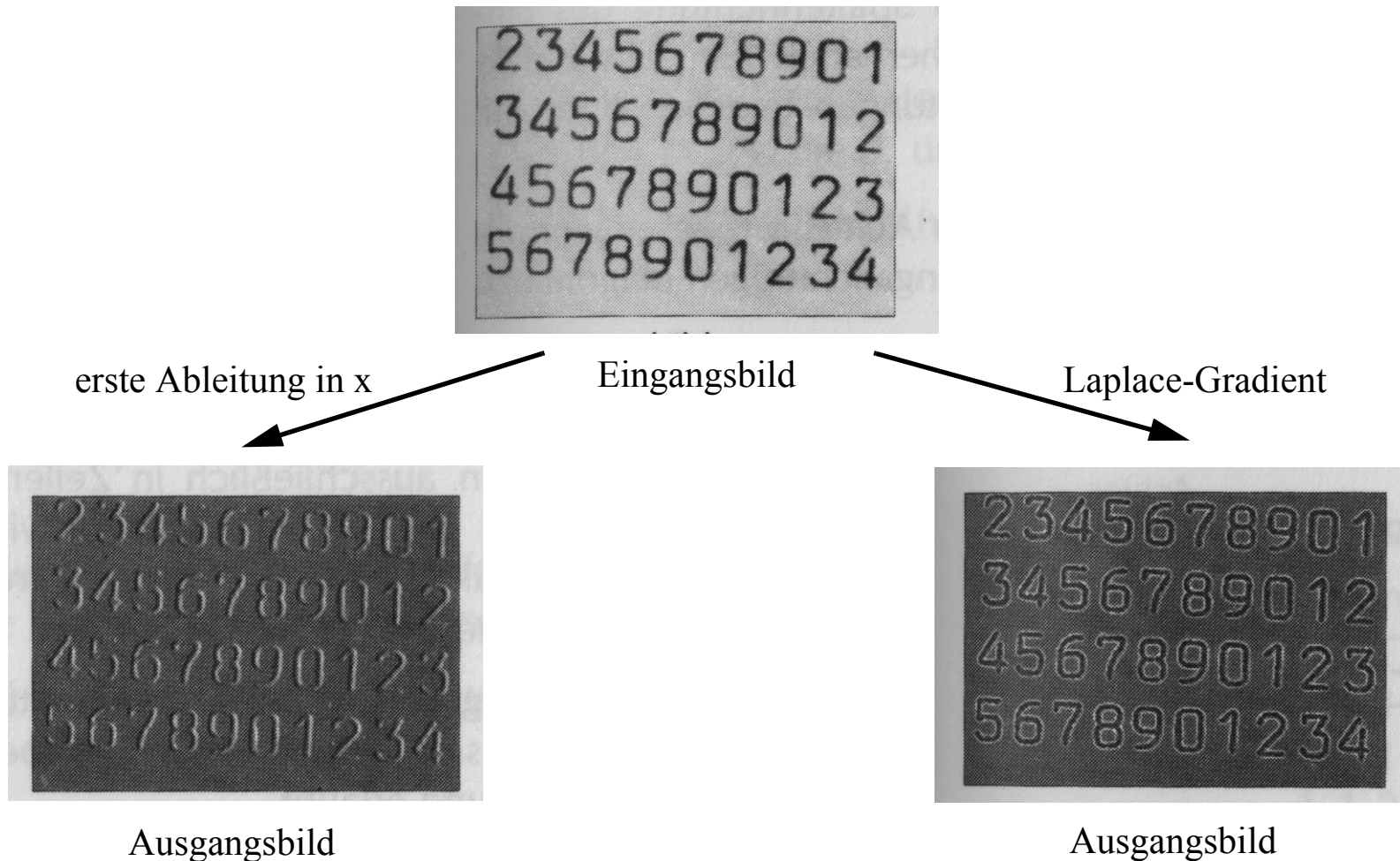


$$\begin{pmatrix} \dots & 0 & 0 & 0 & 0 & 0 & \dots \\ \dots & -3 & -3 & -3 & -3 & -3 & \dots \\ \dots & 6 & 3 & 3 & 3 & 6 & \dots \\ \dots & 0 & -3 & -3 & -3 & 0 & \dots \\ \dots & 12 & 6 & 6 & 6 & 12 & \dots \\ \dots & 0 & -3 & -3 & -3 & 0 & \dots \\ \dots & 6 & 3 & 3 & 3 & 6 & \dots \\ \dots & -3 & -3 & -3 & -3 & -3 & \dots \\ \dots & 0 & 0 & 0 & 0 & 0 & \dots \end{pmatrix}$$

Ausgangsbild

8.2 Lineare Ortsfilterungen (VI)

Beispiele



8.3 Abstandstransformationen (I)

Motivation

- “Wie gut ist die Anbindung eines Gebiets an den öffentlichen Verkehr?”
- Erstelle eine Karte, die für jeden Ort den Abstand zum nächstgelegenen Bahnhof darstellt.

Gegeben

- Ein Eingangsbild, das ein Zielobjekt darstellt (nicht unbedingt zusammenhängend)
- Eine Distanzfunktion für ein Paar von Pixeln

Gesucht

- Das Ausgangsbild, das jedem Pixel den Grauwert zuordnet, der seiner Distanz zum nächsten Pixel des Zielobjekts entspricht

8.3 Algorithmen (I)

Naiver Algorithmus

- Durchlaufe alle Pixel (x,y) des Rasterbildes und tue das folgende:
 - Bestimme für jedes andere Pixel (a,b) die Distanz $\text{dist}((x,y),(a,b))$.
 - Falls (a,b) zum Zielobjekt gehört und dist bisher minimal ist, setze den Grauwert von (x,y) auf dist .
- Dieser Algorithmus besitzt eine Laufzeit von $O(NM)^2$.

Idee zur Verbesserung

- Alle Pixel des Zielobjekts besitzen die Distanz 0.
- Wenn die minimale Distanz aller Pixel, die von einem Pixel p die Distanz 1 besitzen d ist, dann besitzt p die Distanz $d + 1$.

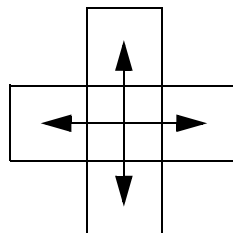
Ablauf des Algorithmus

- Setze für alle Pixel des Zielobjekts den Grauwert auf 0.
- Durchlaufe alle Pixel (x,y) des Rasterbildes und tue das folgende:
 - Sammele die Grauwerte d aller k Pixel, die von (x,y) die Distanz 1 besitzen;
 - Bestimme das Minimum Min von $\{d(x,y), d(x_1,y_1)+1, \dots, d(x_k,y_k)+1\}$;
 - Setze den Grauwert von (x,y) auf Min ;

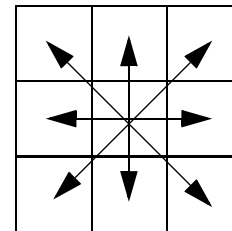
8.3 Umgebungen

Umgebungen

- Zu definieren ist die *Umgebung* eines Pixels p , d.h. die Menge aller Pixel, die eine Distanz von 1 zu p besitzen.
- Die *Viererumgebung* eines Pixels (x,y) besteht aus den Pixeln $(x,y+1)$, $(x,y-1)$, $(x-1,y)$ und $(x+1,y)$.
- Die *Achterumgebung* eines Pixels (x,y) besteht aus den vier Pixeln der Viererumgebung und zusätzlich den vier Pixeln $(x+1,y+1)$, $(x+1,y-1)$, $(x-1,y-1)$ und $(x-1,y+1)$.



Viererumgebung



Achterumgebung

Notation

Die Pixel p , p_1 bzw. p_2 besitzen die Koordinaten (x,y) , (x_1,y_1) bzw. (x_2,y_2) .

8.3 Distanzfunktionen

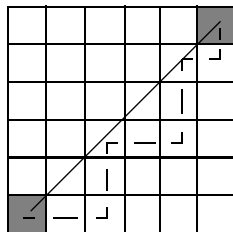
Distanzfunktionen

- Die gebräuchlichste Distanzfunktion ist die *Euklidische Distanz* D_e :

$$D_e(p_1, p_2) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

- Die *Viererdistanz* ist die Distanz, die durch Viererumgebungen induziert wird:

$$D_4(p_1, p_2) = |x_1 - x_2| + |y_1 - y_2|$$



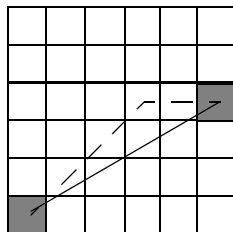
----- $D_4 = 10$

———— $D_e = 50^{1/2}$

$$D_e \leq D_4 \leq \sqrt{2} \cdot D_e$$

- Die *Achterdistanz* ist die Distanz, die durch Achterumgebungen induziert wird:

$$D_8(p_1, p_2) = \max\{|x_1 - x_2|, |y_1 - y_2|\}$$



----- $D_8 = 5$

———— $D_e = 34^{1/2}$

$$\frac{D_e}{\sqrt{2}} \leq D_8 \leq D_e$$

8.3 Algorithmen (II)

PROCEDURE Abstandstransformation (Eingangsbild, Distanz)

- (1) FOR ALL Pixel (x,y) aus Eingangsbild DO
 IF (x,y) gehört zum Zielobjekt THEN
 $d(x,y) := 0$
 ELSE $d(x,y) := \text{MAXDIST}$
END FOR;
- (2) FOR ALL Pixel (x,y) aus Eingangsbild *von links oben nach rechts unten* DO
 Sammele die Grauwerte d aller k Pixel, die von (x,y) die Distanz 1 besitzen;
 Bestimme das Minimum Min von $\{d(x,y), d(x_1,y_1)+1, \dots, d(x_k,y_k)+1\}$;
 $d(x,y) := \text{Min}$;
- (3) FOR ALL Pixel (x,y) aus Eingangsbild *von rechts unten nach links oben* DO
 Sammele die Grauwerte d aller k Pixel, die von (x,y) die Distanz 1 besitzen;
 Bestimme das Minimum Min von $\{d(x,y), d(x_1,y_1)+1, \dots, d(x_k,y_k)+1\}$;
 $d(x,y) := \text{Min}$;

8.3 Algorithmen (III)

Beispiel

2	3	5	2	4
2	4	7	1	1
6	7	1	2	3
3	1	3	6	5
1	2	7	4	5

Eingangsbild

8	8	8	8	8
8	8	8	0	0
8	8	0	8	8
8	0	8	8	8
0	8	8	8	8

Zwischenergebnis nach (1)

8	8	8	1	1
8	8	1	0	0
8	1	0	1	1
1	0	1	2	2
0	1	2	3	3

Zwischenergebnis nach (2)

4	3	2	1	1
3	2	1	0	0
2	1	0	1	1
1	0	1	2	2
0	1	2	3	3

Endergebnis nach (3)

Distanz =
Viererdistanz

8.4 Kombination von Bildern (I)

Definition

- Eine *Kombination von k Eingangsbildern* ist eine Funktion

$$g(x, y) = f(d_1(x, y), d_2(x, y), \dots, d_k(x, y))$$

- Von praktischer Bedeutung sind arithmetische und logische Operationen f .

Summenbildung

- Zur Summenbildung von k thematischen Rasterbildern werden die Grauwerte mit Hilfe von Funktionen TC_1, \dots, TC_k geeignet gewichtet.
- Die *Summenbildung* von k Bildern mit Pixeln $d_i(x, y)$, $1 \leq i \leq k$, ist also eine Funktion

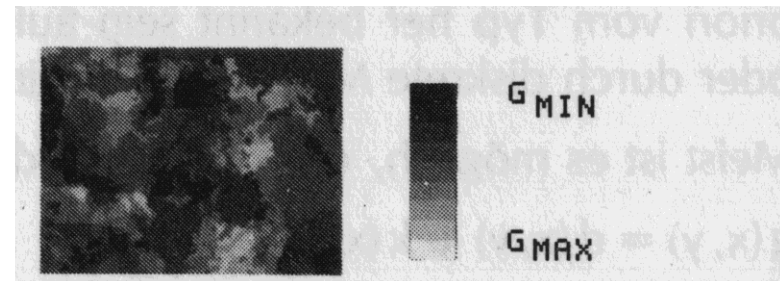
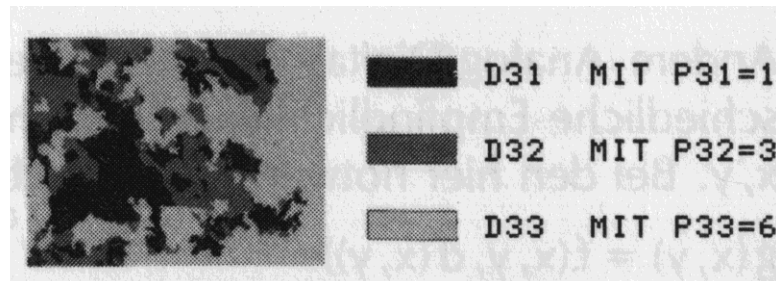
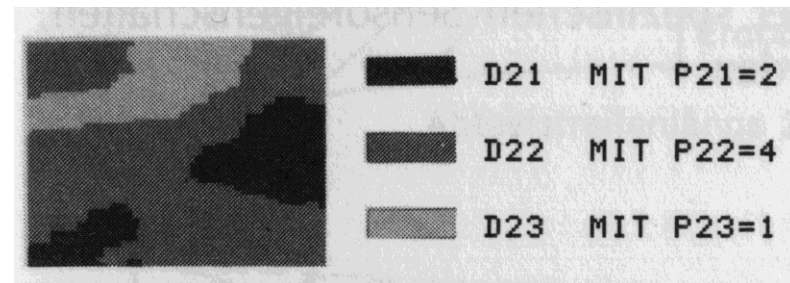
$$g(x, y) = \sum_{i=1}^k TC_i(d_i(x, y))$$

- Liegen nach Anwendung dieser Funktion der minimale (g_{min}) oder der maximale (g_{max}) Grauwert ausserhalb des Intervalls $[0, \dots, d_{max}]$, so muss $g(x, y)$ mittels der $TC(g_{min}, 0, g_{max}, d_{max})$ transformiert werden.

8.4 Kombination von Bildern (II)

Beispiel

- Gegeben sind $k = 3$ Eingangsbilder mit Grauwerten im Intervall $[1, 2, 3]$, z.B. derzeitige Flächennutzung, Bodenqualität, Topographie.
- Ausgangsbild: Eignung des Gebiets für einen gegebenen Planungszweck
- Die Funktionen TC_i (thematische Gewichte) sind folgendermassen definiert:
 $TC_1 (1,1,2,5,3,2)$, $TC_2 (1,2,2,4,3,1)$, $TC_3 (1,1,2,3,3,6)$
- $g_{min} = 3$ (sehr schlechte Eignung), $g_{max} = 15$ (sehr gute Eignung)



8.4 Kombination von Bildern (III)

Differenzbildung

- Die Differenzbildung zweier Bilder dient der Erkennung von Änderungen des Bildinhalts zeitlich versetzter, aber lageidentischer Aufnahmen.
- Die *Differenzbildung* zweier Eingangsbilder ist definiert als

$$g(x, y) = d_1(x, y) - d_2(x, y) + \frac{d_{max}}{2}$$

- Nulldifferenzen erhalten den Wert $d_{max}/2$, positive bzw. negative Differenzen erhalten Grauwerte $> d_{max}/2$ bzw. $< d_{max}/2$.
- $g_{min} = -d_{max}/2$, $g_{max} = 3 d_{max}/2$
- Durch Anwendung der folgenden TC (bei $d_{max} = 255$) erhält man ein Ausgangsbild im Grauwertbereich $[0, \dots, 255]$:

TC(-128,0,0,0)

TC(0,0,255,255)

TC(255,255,382,255)

8.4 Kombination von Bildern (IV)

Logische Operationen

- Logische Operationen sind nur für *binäre Eingangsbilder* möglich, d.h. für Rasterbilder mit den beiden Grauwerten 0 (= FALSE) und 1 (= TRUE):

$$g(x, y) = d_1(x, y) \text{ OP } d_2(x, y)$$

- *OP* ist eine der logischen Operationen
 - AND ($a \text{ AND } b = 1$ nur wenn $a=1$ und $b=1$)
 - OR ($a \text{ OR } b = 1$ wenn *mindestens* einer der Eingangswerte =1)
 - XOR ($a \text{ XOR } b = 1$ wenn *genau* einer der Eingangswerte =1)

Einblendungen

- Die Einblendung von Texturen o.ä. in ein Rasterbild erfolgt mit Hilfe der Operation OR:

$$g(x, y) = d(x, y) \text{ OR } \textit{Textur}(x, y)$$

8.4 Kombination von Bildern (V)

Beispiel

- Gegeben sind $k = 3$ Eingangsbilder mit Grauwerten im Intervall $[1, 2, 3]$, z.B. Flächennutzung, Bodenqualität, Topographie.
- Es sollen nur die jeweiligen Grauwerte mit dem höchsten Gewicht betrachtet werden (optimale Werte). Gesucht sind die Regionen, die in jedem Bild (d.h. nach jedem Thema) den optimalen Wert besitzen.
- Die Eingangsbilder werden mit Hilfe der TC_i in binäre Bilder transformiert:
 $TC_1/TC_2(1,0,2,1,3,0)$ für das erste und für das zweite Bild
 $TC_3(1,0,2,0,3,1)$ für das dritte Bild
- Die drei binären Bilder $d_i(x,y)$ werden folgendermassen kombiniert:

$$g(x, y) = d_1(x, y) \text{ AND } d_2(x, y) \text{ AND } d_3(x, y)$$

