FAKULTÄT FÜR MATHEMATIK
UNIVERSITÄT ULM
ABT. STOCHASTIK
ABT. ANGEWANDTE
INFORMATIONSVERARBEITUNG

Seminar „Simulation und Bildanalyse mit Java“

Morphologische Bildverarbeitung II

BETREUER: JOHANNES MAYER
AUTOR: BARBARA PANZER SS03

Inhalt

- Grundlagen: Bilder
- Begriff der Morphologischen Transformation
- Lineare Distanztransformation
  - Lineare Distanztransformation entlang von Parallelen der Koordinatenachsen
  - Lineare Distanztransformation entlang von beliebigen Geraden
- Grundlagen: Erosion
- Sehnenlängenverteilung
- Star Volume
  - Exakte Definition
  - Schätzer für das Star Volume
- Sehnenlängentransformation

Grundlagen: Bilder (1)

Bilder sind Funktionen, die meist über einem rechteckigen Rahmen, dem sog. Definitionsbereich D ($D \subset \mathbb{Z}^n$) definiert werden:

- Hier ausschließlich Verwendung von 2D-Bildern
  ⇨ Der Definitionsbereich ist eine Ebene
- Binärbild:
  – Wertebereich: $W = \{0,1\}$
  – Definition Binärbild f:

$$\boxed{f : D_f \subset \mathbb{Z}^n \rightarrow \{0, 1\}}$$

  – Pixel gehört zum Objekt
    ⇨ Wert 1
    ⇨ Darstellung: ■
  – Pixel gehört zum Hintergrund
    ⇨ Wert 0
    ⇨ Darstellung: □

Grundlagen: Bilder (2)

- Grauwertbild:
  - Wertebereich: endliche Menge nichtnegativer ganzer Zahlen
  - Definition: Grauwertbild f

$$\boxed{f : D_f \subset \mathbb{Z}^n \to \{0, 1, \ldots, t_{\mathrm{max}}\}}$$

  - Bei einem Bild mit n Bit codierten Pixeln gilt: $t_{max}=2^n$-1

Morphologische Transformationen

Morphologische Transformationen

Mathematische Morphologie

- Entwickelt von **J. Serra** in Zusammenarbeit mit **G. Matheron** an der Ecole des Mines in Paris ab 1965
- Bilder werden als *Mengen* von Pixeln bzw. Pixelkoordinaten betrachtet
- Proben der Bilder mit *strukturierenden Elementen*, um bestimmte Merkmale in Bildern hervorzuheben oder zu entfernen

Morphologische Transformationen

Mathematische Morphologie

- Entwickelt von **J. Serra** in Zusammenarbeit mit **G. Matheron** an der Ecole des Mines in Paris ab 1965
- Bilder werden als *Mengen* von Pixeln bzw. Pixelkoordinaten betrachtet
- Proben der Bilder mit *strukturierenden Elementen*, um bestimmte Merkmale in Bildern hervorzuheben oder zu entfernen

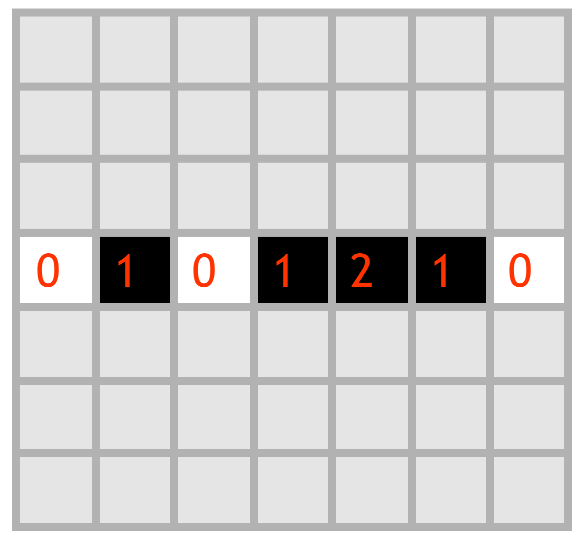
- Morphologische Transformationen sind Bild-zu-Bild-Transformationen
- Bild-zu-Bild-Transformation:
  - Das transformierte Bild und das Originalbild besitzen denselben Definitionsbereich
  - Das transformierte Bild ist immer noch eine Abbildung in die Menge nichtnegativer ganzer Zahlen

Lineare Distanztransformation (1)

- Die Distanztransformation D angewendet auf ein Binärbild f, verbindet jedes Pixel p des Definitionsbereichs D_f von f mit der Distanz zum nächsten Hintergrund-Pixel.

$$[D(f)](x) = min\{d(x,y) | f(y) = 0\}$$

- Bei der linearen Distanztransformation lässt sich d hierbei besonders einfach berechnen, denn es wird nur der Abstand zum nächsten nullwertigen Pixel entlang einer Gerade durch das Bild berücksichtigt.
- z.B. Gerade parallel zur x-Achse



Lineare Distanztransformation (2)

- Problem: Distanztransformation am Bildrand

?	2	1	0	1	2	1	0

  ➪ es fehlt die lokale Information für die „Pixel" außerhalb des Bildes

- Konsistente Lösung: „Pixel" außerhalb des Bildes werden als Hintergrund angesehen

0	0	0	0	0	0	0	0	0	0
0	1	2	1	0	1	2	1	0	0
0	0	0	0	0	0	0	0	0	0

Lineare Distanztransformation (3)

<u>Algorithmus zur Berechnung der linearen Distanztransformation</u>

(nach **Rosenfeld u. Pfaltz 1966**)

- Benötigt eine sequentielle Vorwärts- und eine sequentielle Rückwärts-Bildabtastung
- Rückwärtsnachbarn N^{-}_{G} werden beim Vorwärtsabtasten und Vorwärtsnachbarn N^{+}_{G} beim Rückwärtsabtasten berücksichtigt.
- Rückwärtsnachbar bei der linearen Distanztransformation: Bereits bearbeiteter Vorgänger entlang der betrachteten Gerade bei der Vorwärtsabtastung
- Vorwärtsnachbar:
  Bereits bearbeiteter Nachfolger entlang der betrachteten Gerade bei Rückwärtsabtastung

Lineare Distanztransformation (3)

<u>Algorithmus zur Berechnung der linearen Distanztransformation</u>

(nach **Rosenfeld u. Pfaltz 1966**)

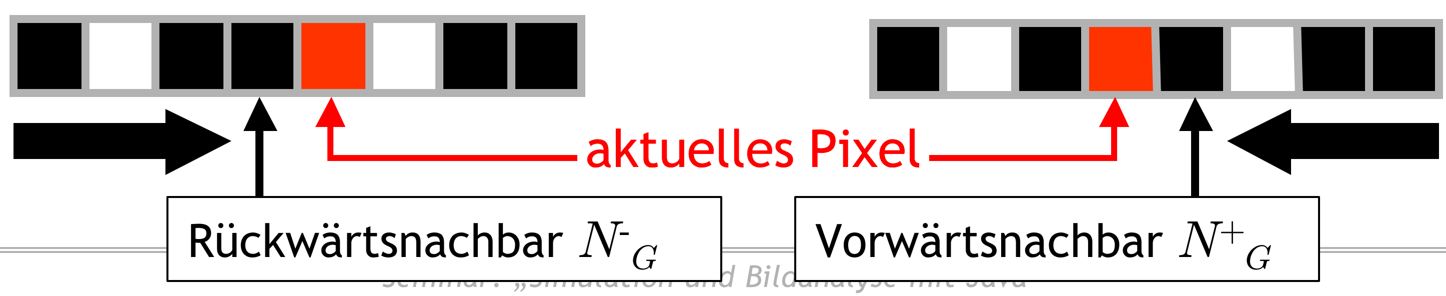
- Benötigt eine sequentielle Vorwärts- und eine sequentielle Rückwärts-Bildabtastung
- Rückwärtsnachbarn N^-_G werden beim Vorwärtsabtasten und Vorwärtsnachbarn N^+_G beim Rückwärtsabtasten berücksichtigt.
- Rückwärtsnachbar bei der linearen Distanztransformation: Bereits bearbeiteter Vorgänger entlang der betrachteten Gerade bei der Vorwärtsabtastung
- Vorwärtsnachbar:
  Bereits bearbeiteter Nachfolger entlang der betrachteten Gerade bei Rückwärtsabtastung

Lineare Distanztransformation (3)

<u>Algorithmus zur Berechnung der linearen Distanztransformation</u>

(nach **Rosenfeld u. Pfaltz 1966**)

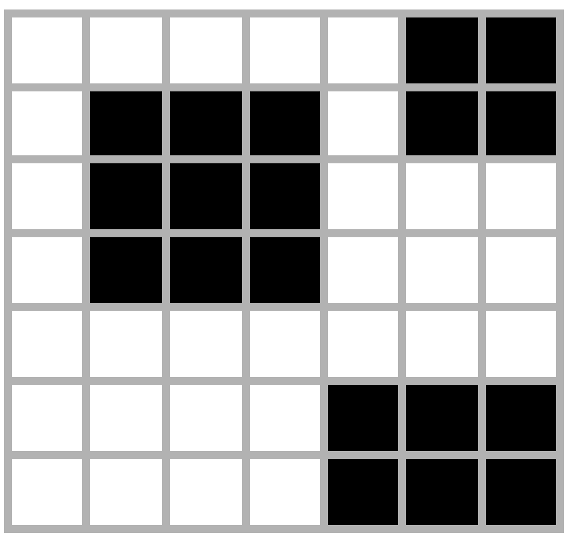
- Benötigt eine sequentielle Vorwärts- und eine sequentielle Rückwärts-Bildabtastung
- Rückwärtsnachbarn N^-_G werden beim Vorwärtsabtasten und Vorwärtsnachbarn N^+_G beim Rückwärtsabtasten berücksichtigt.
- Rückwärtsnachbar bei der linearen Distanztransformation: Bereits bearbeiteter Vorgänger entlang der betrachteten Gerade bei der Vorwärtsabtastung
- Vorwärtsnachbar:
  Bereits bearbeiteter Nachfolger entlang der betrachteten Gerade bei Rückwärtsabtastung



Lineare Distanztransformation (4)

<u>Algorithmus:</u>

1. Für alle Parallelen h_i der Geraden g, die sich im Bild befinden:
   Vorwärtsabtasten aller Pixel $p \in D_f$ entlang der Geraden h_i
   $$\textbf{IF} \quad f(p) = 1 \quad \textbf{THEN} \quad f(p) \leftarrow 1 + f(N_G^-(p))$$
2. Für alle Parallelen p_i der Geraden g, die sich im Bild befinden:
   Rückwärtsabtasten aller Pixel $p \in D_f$ entlang der Geraden h_i
   $$\textbf{IF} \quad f(p) \neq 0 \quad \textbf{THEN} \quad f(p) \leftarrow \min\{f(p), 1 + f(N_G^+(p))\}$$



ursprüngliches Bild

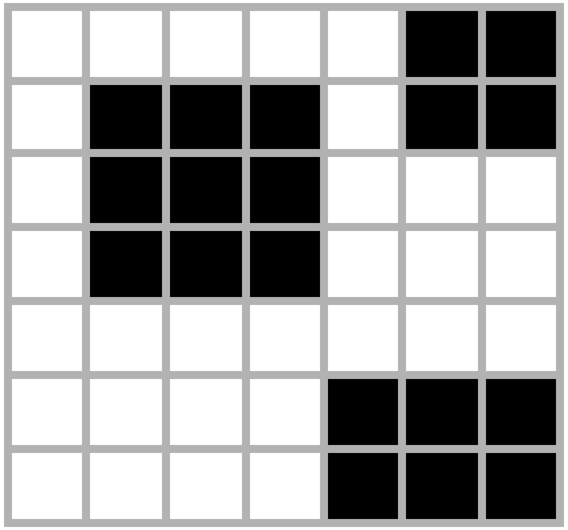
0	0	0	0	0	1	1
0	1	1	1	0	1	1
0	1	1	1	0	0	0
0	1	1	1	0	0	0
0	0	0	0	0	0	0
0	0	0	0	1	1	1
0	0	0	0	1	1	1

Lineare Distanztransformation (4)

Algorithmus:

1. Für alle Parallelen h_i der Geraden g, die sich im Bild befinden:
   Vorwärtsabtasten aller Pixel $p \in D_f$ entlang der Geraden h_i
   $\textbf{IF} \quad f(p) = 1 \quad \textbf{THEN} \quad f(p) \leftarrow 1 + f(N_G^-(p))$
2. Für alle Parallelen p_i der Geraden g, die sich im Bild befinden:
   Rückwärtsabtasten aller Pixel $p \in D_f$ entlang der Geraden h_i
   $\textbf{IF} \quad f(p) \neq 0 \quad \textbf{THEN} \quad f(p) \leftarrow \min\{f(p), 1 + f(N_G^+(p)\}$

Vorwärtsabtasten



ursprüngliches Bild

0	0	0	0	0	1	1
0	1	1	1	0	1	1
0	1	1	1	0	0	0
0	1	1	1	0	0	0
0	0	0	0	0	0	0
0	0	0	0	1	1	1
0	0	0	0	1	1	1

Lineare Distanztransformation (4)

<u>Algorithmus:</u>

1. Für alle Parallelen h_i der Geraden g, die sich im Bild befinden:
   Vorwärtsabtasten aller Pixel $p \in D_f$ entlang der Geraden h_i
   $\textbf{IF} \quad f(p) = 1 \quad \textbf{THEN} \quad f(p) \leftarrow 1 + f(N_G^-(p))$
2. Für alle Parallelen p_i der Geraden g, die sich im Bild befinden:
   Rückwärtsabtasten aller Pixel $p \in D_f$ entlang der Geraden h_i
   $\textbf{IF} \quad f(p) \neq 0 \quad \textbf{THEN} \quad f(p) \leftarrow \min\{f(p), 1 + f(N_G^+(p))\}$

Vorwärtsabtasten

0						

ursprüngliches Bild

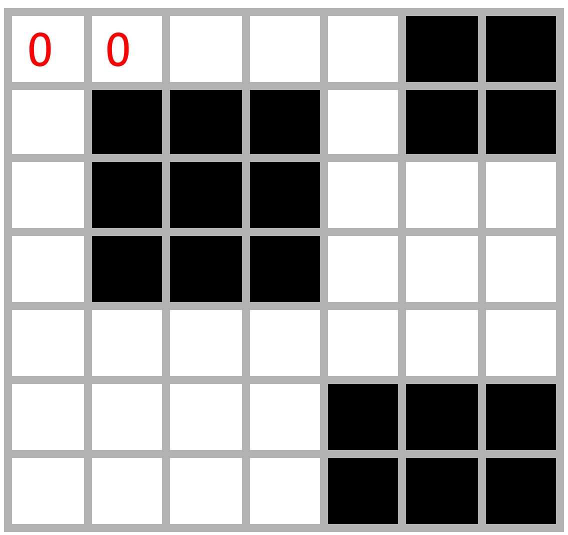
0	0	0	0	0	1	1
0	1	1	1	0	1	1
0	1	1	1	0	0	0
0	1	1	1	0	0	0
0	0	0	0	0	0	0
0	0	0	0	1	1	1
0	0	0	0	1	1	1

Lineare Distanztransformation (4)

Algorithmus:

1. Für alle Parallelen h_i der Geraden g, die sich im Bild befinden:
   Vorwärtsabtasten aller Pixel $p \in D_f$ entlang der Geraden h_i
   $\textbf{IF} \quad f(p) = 1 \quad \textbf{THEN} \quad f(p) \leftarrow 1 + f(N_G^-(p))$
2. Für alle Parallelen p_i der Geraden g, die sich im Bild befinden:
   Rückwärtsabtasten aller Pixel $p \in D_f$ entlang der Geraden h_i
   $\textbf{IF} \quad f(p) \neq 0 \quad \textbf{THEN} \quad f(p) \leftarrow \min\{f(p), 1 + f(N_G^+(p))\}$

Vorwärtsabtasten



ursprüngliches Bild

0	0	0	0	0	1	1
0	1	1	1	0	1	1
0	1	1	1	0	0	0
0	1	1	1	0	0	0
0	0	0	0	0	0	0
0	0	0	0	1	1	1
0	0	0	0	1	1	1

Lineare Distanztransformation (4)

Algorithmus:

1. Für alle Parallelen h_i der Geraden g, die sich im Bild befinden:
   Vorwärtsabtasten aller Pixel $p \in D_f$ entlang der Geraden h_i
   $\textbf{IF} \quad f(p) = 1 \quad \textbf{THEN} \quad f(p) \leftarrow 1 + f(N_G^-(p))$
2. Für alle Parallelen p_i der Geraden g, die sich im Bild befinden:
   Rückwärtsabtasten aller Pixel $p \in D_f$ entlang der Geraden h_i
   $\textbf{IF} \quad f(p) \neq 0 \quad \textbf{THEN} \quad f(p) \leftarrow \min\{f(p), 1 + f(N_G^+(p)\}$

Vorwärtsabtasten

0	0	0				

ursprüngliches Bild

0	0	0	0	0	1	1
0	1	1	1	0	1	1
0	1	1	1	0	0	0
0	1	1	1	0	0	0
0	0	0	0	0	0	0
0	0	0	0	1	1	1
0	0	0	0	1	1	1

Lineare Distanztransformation (4)

Algorithmus:

1. Für alle Parallelen h_i der Geraden g, die sich im Bild befinden:
   Vorwärtsabtasten aller Pixel $p \in D_f$ entlang der Geraden h_i
   $\textbf{IF} \quad f(p) = 1 \quad \textbf{THEN} \quad f(p) \leftarrow 1 + f(N_G^-(p))$
2. Für alle Parallelen p_i der Geraden g, die sich im Bild befinden:
   Rückwärtsabtasten aller Pixel $p \in D_f$ entlang der Geraden h_i
   $\textbf{IF} \quad f(p) \neq 0 \quad \textbf{THEN} \quad f(p) \leftarrow \min\{f(p), 1 + f(N_G^+(p)\}$

Vorwärtsabtasten

0	0	0	0			

ursprüngliches Bild

0	0	0	0	0	1	1
0	1	1	1	0	1	1
0	1	1	1	0	0	0
0	1	1	1	0	0	0
0	0	0	0	0	0	0
0	0	0	0	1	1	1
0	0	0	0	1	1	1

Lineare Distanztransformation (4)

<u>Algorithmus:</u>

1. Für alle Parallelen h_i der Geraden g, die sich im Bild befinden:
   Vorwärtsabtasten aller Pixel $p \in D_f$ entlang der Geraden h_i
   $\mathbf{IF} \quad f(p) = 1 \quad \mathbf{THEN} \quad f(p) \leftarrow 1 + f(N_G^-(p))$
2. Für alle Parallelen p_i der Geraden g, die sich im Bild befinden:
   Rückwärtsabtasten aller Pixel $p \in D_f$ entlang der Geraden h_i
   $\mathbf{IF} \quad f(p) \neq 0 \quad \mathbf{THEN} \quad f(p) \leftarrow \min\{f(p), 1 + f(N_G^+(p)\}$

Vorwärtsabtasten

0	0	0	0	0		

ursprüngliches Bild

0	0	0	0	0	1	1
0	1	1	1	0	1	1
0	1	1	1	0	0	0
0	1	1	1	0	0	0
0	0	0	0	0	0	0
0	0	0	0	1	1	1
0	0	0	0	1	1	1

Lineare Distanztransformation (4)

<u>Algorithmus:</u>

1. Für alle Parallelen h_i der Geraden g, die sich im Bild befinden:
   Vorwärtsabtasten aller Pixel $p \in D_f$ entlang der Geraden h_i
   $$\textbf{IF} \quad f(p) = 1 \quad \textbf{THEN} \quad f(p) \leftarrow 1 + f(N_G^-(p))$$
2. Für alle Parallelen p_i der Geraden g, die sich im Bild befinden:
   Rückwärtsabtasten aller Pixel $p \in D_f$ entlang der Geraden h_i
   $$\textbf{IF} \quad f(p) \neq 0 \quad \textbf{THEN} \quad f(p) \leftarrow \min\{f(p), 1 + f(N_G^+(p)\}$$

Vorwärtsabtasten

0	0	0	0	0	1	

ursprüngliches Bild

0	0	0	0	0	1	1
0	1	1	1	0	1	1
0	1	1	1	0	0	0
0	1	1	1	0	0	0
0	0	0	0	0	0	0
0	0	0	0	1	1	1
0	0	0	0	1	1	1

Lineare Distanztransformation (4)

<u>Algorithmus:</u>

1. Für alle Parallelen h_i der Geraden g, die sich im Bild befinden:
   Vorwärtsabtasten aller Pixel $p \in D_f$ entlang der Geraden h_i
   $$\textbf{IF} \quad f(p) = 1 \quad \textbf{THEN} \quad f(p) \leftarrow 1 + f(N_G^-(p))$$
2. Für alle Parallelen p_i der Geraden g, die sich im Bild befinden:
   Rückwärtsabtasten aller Pixel $p \in D_f$ entlang der Geraden h_i
   $$\textbf{IF} \quad f(p) \neq 0 \quad \textbf{THEN} \quad f(p) \leftarrow \min\{f(p), 1 + f(N_G^+(p))\}$$

Vorwärtsabtasten

0	0	0	0	0	1	2

ursprüngliches Bild

0	0	0	0	0	1	1
0	1	1	1	0	1	1
0	1	1	1	0	0	0
0	1	1	1	0	0	0
0	0	0	0	0	0	0
0	0	0	0	1	1	1
0	0	0	0	1	1	1

Lineare Distanztransformation (4)

<u>Algorithmus:</u>

1. Für alle Parallelen h_i der Geraden g, die sich im Bild befinden:
   Vorwärtsabtasten aller Pixel $p \in D_f$ entlang der Geraden h_i
   $\textbf{IF} \quad f(p) = 1 \quad \textbf{THEN} \quad f(p) \leftarrow 1 + f(N_G^-(p))$
2. Für alle Parallelen p_i der Geraden g, die sich im Bild befinden:
   Rückwärtsabtasten aller Pixel $p \in D_f$ entlang der Geraden h_i
   $\textbf{IF} \quad f(p) \neq 0 \quad \textbf{THEN} \quad f(p) \leftarrow \min\{f(p), 1 + f(N_G^+(p))\}$

Vorwärtsabtasten

0	0	0	0	0	1	2
0						

ursprüngliches Bild

0	0	0	0	0	1	1
0	1	1	1	0	1	1
0	1	1	1	0	0	0
0	1	1	1	0	0	0
0	0	0	0	0	0	0
0	0	0	0	1	1	1
0	0	0	0	1	1	1

Lineare Distanztransformation (4)

<u>Algorithmus:</u>

1. Für alle Parallelen h_i der Geraden g, die sich im Bild befinden:
   Vorwärtsabtasten aller Pixel $p \in D_f$ entlang der Geraden h_i
   $\textbf{IF} \quad f(p) = 1 \quad \textbf{THEN} \quad f(p) \leftarrow 1 + f(N_G^-(p))$
2. Für alle Parallelen p_i der Geraden g, die sich im Bild befinden:
   Rückwärtsabtasten aller Pixel $p \in D_f$ entlang der Geraden h_i
   $\textbf{IF} \quad f(p) \neq 0 \quad \textbf{THEN} \quad f(p) \leftarrow \min\{f(p), 1 + f(N_G^+(p))\}$

Vorwärtsabtasten

0	0	0	0	0	1	2
0	1					

ursprüngliches Bild

0	0	0	0	0	1	1
0	1	1	1	0	1	1
0	1	1	1	0	0	0
0	1	1	1	0	0	0
0	0	0	0	0	0	0
0	0	0	0	1	1	1
0	0	0	0	1	1	1

Lineare Distanztransformation (4)

<u>Algorithmus:</u>

1. Für alle Parallelen h_i der Geraden g, die sich im Bild befinden:
   Vorwärtsabtasten aller Pixel $p \in D_f$ entlang der Geraden h_i
   $\textbf{IF} \quad f(p) = 1 \quad \textbf{THEN} \quad f(p) \leftarrow 1 + f(N_G^-(p))$
2. Für alle Parallelen p_i der Geraden g, die sich im Bild befinden:
   Rückwärtsabtasten aller Pixel $p \in D_f$ entlang der Geraden h_i
   $\textbf{IF} \quad f(p) \neq 0 \quad \textbf{THEN} \quad f(p) \leftarrow \min\{f(p), 1 + f(N_G^+(p))\}$

Vorwärtsabtasten

0	0	0	0	0	1	2
0	1	2				

ursprüngliches Bild

0	0	0	0	0	1	1
0	1	1	1	0	1	1
0	1	1	1	0	0	0
0	1	1	1	0	0	0
0	0	0	0	0	0	0
0	0	0	0	1	1	1
0	0	0	0	1	1	1

Lineare Distanztransformation (4)

Algorithmus:

1. Für alle Parallelen h_i der Geraden g, die sich im Bild befinden:
   Vorwärtsabtasten aller Pixel $p \in D_f$ entlang der Geraden h_i
   $\textbf{IF} \quad f(p) = 1 \quad \textbf{THEN} \quad f(p) \leftarrow 1 + f(N_G^-(p))$
2. Für alle Parallelen p_i der Geraden g, die sich im Bild befinden:
   Rückwärtsabtasten aller Pixel $p \in D_f$ entlang der Geraden h_i
   $\textbf{IF} \quad f(p) \neq 0 \quad \textbf{THEN} \quad f(p) \leftarrow \min\{f(p), 1 + f(N_G^+(p)\}$

Vorwärtsabtasten

0	0	0	0	0	1	2
0	1	2	3			

ursprüngliches Bild

0	0	0	0	0	1	1
0	1	1	1	0	1	1
0	1	1	1	0	0	0
0	1	1	1	0	0	0
0	0	0	0	0	0	0
0	0	0	0	1	1	1
0	0	0	0	1	1	1

Lineare Distanztransformation (4)

Algorithmus:

1. Für alle Parallelen h_i der Geraden g, die sich im Bild befinden:
   Vorwärtsabtasten aller Pixel $p \in D_f$ entlang der Geraden h_i
   $$\textbf{IF} \quad f(p) = 1 \quad \textbf{THEN} \quad f(p) \leftarrow 1 + f(N_G^-(p))$$
2. Für alle Parallelen p_i der Geraden g, die sich im Bild befinden:
   Rückwärtsabtasten aller Pixel $p \in D_f$ entlang der Geraden h_i
   $$\textbf{IF} \quad f(p) \neq 0 \quad \textbf{THEN} \quad f(p) \leftarrow \min\{f(p), 1 + f(N_G^+(p))\}$$

Vorwärtsabtasten

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

ursprüngliches Bild

0	0	0	0	0	1	1
0	1	1	1	0	1	1
0	1	1	1	0	0	0
0	1	1	1	0	0	0
0	0	0	0	0	0	0
0	0	0	0	1	1	1
0	0	0	0	1	1	1

Lineare Distanztransformation (4)

Algorithmus:

1. Für alle Parallelen h_i der Geraden g, die sich im Bild befinden:
   Vorwärtsabtasten aller Pixel $p \in D_f$ entlang der Geraden h_i
   $\textbf{IF} \quad f(p) = 1 \quad \textbf{THEN} \quad f(p) \leftarrow 1 + f(N_G^-(p))$
2. Für alle Parallelen p_i der Geraden g, die sich im Bild befinden:
   Rückwärtsabtasten aller Pixel $p \in D_f$ entlang der Geraden h_i
   $\textbf{IF} \quad f(p) \neq 0 \quad \textbf{THEN} \quad f(p) \leftarrow \min\{f(p), 1 + f(N_G^+(p)\}$

Vorwärtsabtasten

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

ursprüngliches Bild

0	0	0	0	0	1	1
0	1	1	1	0	1	1
0	1	1	1	0	0	0
0	1	1	1	0	0	0
0	0	0	0	0	0	0
0	0	0	0	1	1	1
0	0	0	0	1	1	1

Lineare Distanztransformation (4)

Algorithmus:

1. Für alle Parallelen h_i der Geraden g, die sich im Bild befinden:
   Vorwärtsabtasten aller Pixel $p \in D_f$ entlang der Geraden h_i
   IF $f(p) = 1$ **THEN** $f(p) \leftarrow 1 + f(N_G^-(p))$
2. Für alle Parallelen p_i der Geraden g, die sich im Bild befinden:
   Rückwärtsabtasten aller Pixel $p \in D_f$ entlang der Geraden h_i
   IF $f(p) \neq 0$ **THEN** $f(p) \leftarrow \min\{f(p), 1 + f(N_G^+(p)\}$

Vorwärtsabtasten

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

ursprüngliches Bild

0	0	0	0	0	1	1
0	1	1	1	0	1	1
0	1	1	1	0	0	0
0	1	1	1	0	0	0
0	0	0	0	0	0	0
0	0	0	0	1	1	1
0	0	0	0	1	1	1

Lineare Distanztransformation (4)

Algorithmus:

1. Für alle Parallelen h_i der Geraden g, die sich im Bild befinden:
   Vorwärtsabtasten aller Pixel $p \in D_f$ entlang der Geraden h_i
   $\textbf{IF} \quad f(p) = 1 \quad \textbf{THEN} \quad f(p) \leftarrow 1 + f(N_G^-(p))$
2. Für alle Parallelen p_i der Geraden g, die sich im Bild befinden:
   Rückwärtsabtasten aller Pixel $p \in D_f$ entlang der Geraden h_i
   $\textbf{IF} \quad f(p) \neq 0 \quad \textbf{THEN} \quad f(p) \leftarrow \min\{f(p), 1 + f(N_G^+(p)\}$

Vorwärtsabtasten

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

ursprüngliches Bild

0	0	0	0	0	1	1
0	1	1	1	0	1	1
0	1	1	1	0	0	0
0	1	1	1	0	0	0
0	0	0	0	0	0	0
0	0	0	0	1	1	1
0	0	0	0	1	1	1

Lineare Distanztransformation (4)

<u>Algorithmus:</u>

1. Für alle Parallelen h_i der Geraden g, die sich im Bild befinden:
   Vorwärtsabtasten aller Pixel $p \in D_f$ entlang der Geraden h_i
   $\textbf{IF} \quad f(p) = 1 \quad \textbf{THEN} \quad f(p) \leftarrow 1 + f(N_G^-(p))$
2. Für alle Parallelen p_i der Geraden g, die sich im Bild befinden:
   Rückwärtsabtasten aller Pixel $p \in D_f$ entlang der Geraden h_i
   $\textbf{IF} \quad f(p) \neq 0 \quad \textbf{THEN} \quad f(p) \leftarrow \min\{f(p), 1 + f(N_G^+(p))\}$

Rückwärtsabtasten

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

ursprüngliches Bild

0	0	0	0	0	1	1
0	1	1	1	0	1	1
0	1	1	1	0	0	0
0	1	1	1	0	0	0
0	0	0	0	0	0	0
0	0	0	0	1	1	1
0	0	0	0	1	1	1

Lineare Distanztransformation (4)

<u>Algorithmus:</u>

1. Für alle Parallelen h_i der Geraden g, die sich im Bild befinden:
   Vorwärtsabtasten aller Pixel $p \in D_f$ entlang der Geraden h_i
   $\textbf{IF} \quad f(p) = 1 \quad \textbf{THEN} \quad f(p) \leftarrow 1 + f(N_G^-(p))$
2. Für alle Parallelen p_i der Geraden g, die sich im Bild befinden:
   Rückwärtsabtasten aller Pixel $p \in D_f$ entlang der Geraden h_i
   $\textbf{IF} \quad f(p) \neq 0 \quad \textbf{THEN} \quad f(p) \leftarrow \min\{f(p), 1 + f(N_G^+(p)\}$

Rückwärtsabtasten

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

ursprüngliches Bild

0	0	0	0	0	1	1
0	1	1	1	0	1	1
0	1	1	1	0	0	0
0	1	1	1	0	0	0
0	0	0	0	0	0	0
0	0	0	0	1	1	1
0	0	0	0	1	1	1

Lineare Distanztransformation (4)

<u>Algorithmus:</u>

1. Für alle Parallelen h_i der Geraden g, die sich im Bild befinden:
   Vorwärtsabtasten aller Pixel $p \in D_f$ entlang der Geraden h_i
   $\textbf{IF} \quad f(p) = 1 \quad \textbf{THEN} \quad f(p) \leftarrow 1 + f(N_G^-(p))$
2. Für alle Parallelen p_i der Geraden g, die sich im Bild befinden:
   Rückwärtsabtasten aller Pixel $p \in D_f$ entlang der Geraden h_i
   $\textbf{IF} \quad f(p) \neq 0 \quad \textbf{THEN} \quad f(p) \leftarrow \min\{f(p), 1 + f(N_G^+(p))\}$

Rückwärtsabtasten

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

ursprüngliches Bild

0	0	0	0	0	1	1
0	1	1	1	0	1	1
0	1	1	1	0	0	0
0	1	1	1	0	0	0
0	0	0	0	0	0	0
0	0	0	0	1	1	1
0	0	0	0	1	1	1

Lineare Distanztransformation (4)

Algorithmus:

1. Für alle Parallelen h_i der Geraden g, die sich im Bild befinden:
   Vorwärtsabtasten aller Pixel $p \in D_f$ entlang der Geraden h_i
   $\textbf{IF} \quad f(p) = 1 \quad \textbf{THEN} \quad f(p) \leftarrow 1 + f(N_G^-(p))$
2. Für alle Parallelen p_i der Geraden g, die sich im Bild befinden:
   Rückwärtsabtasten aller Pixel $p \in D_f$ entlang der Geraden h_i
   $\textbf{IF} \quad f(p) \neq 0 \quad \textbf{THEN} \quad f(p) \leftarrow \min\{f(p), 1 + f(N_G^+(p)\}$

Rückwärtsabtasten

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

ursprüngliches Bild

0	0	0	0	0	1	1
0	1	1	1	0	1	1
0	1	1	1	0	0	0
0	1	1	1	0	0	0
0	0	0	0	0	0	0
0	0	0	0	1	1	1
0	0	0	0	1	1	1

Lineare Distanztransformation (4)

<u>Algorithmus:</u>

1. Für alle Parallelen h_i der Geraden g, die sich im Bild befinden:
   Vorwärtsabtasten aller Pixel $p \in D_f$ entlang der Geraden h_i
   $\textbf{IF} \quad f(p) = 1 \quad \textbf{THEN} \quad f(p) \leftarrow 1 + f(N_G^-(p))$
2. Für alle Parallelen p_i der Geraden g, die sich im Bild befinden:
   Rückwärtsabtasten aller Pixel $p \in D_f$ entlang der Geraden h_i
   $\textbf{IF} \quad f(p) \neq 0 \quad \textbf{THEN} \quad f(p) \leftarrow \min\{f(p), 1 + f(N_G^+(p)\}$

Rückwärtsabtasten

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

ursprüngliches Bild

0	0	0	0	0	1	1
0	1	1	1	0	1	1
0	1	1	1	0	0	0
0	1	1	1	0	0	0
0	0	0	0	0	0	0
0	0	0	0	1	1	1
0	0	0	0	1	1	1

Lineare Distanztransformation (4)

Algorithmus:

1. Für alle Parallelen h_i der Geraden g, die sich im Bild befinden:
   Vorwärtsabtasten aller Pixel $p \in D_f$ entlang der Geraden h_i
   $\textbf{IF} \quad f(p) = 1 \quad \textbf{THEN} \quad f(p) \leftarrow 1 + f(N_G^-(p))$
2. Für alle Parallelen p_i der Geraden g, die sich im Bild befinden:
   Rückwärtsabtasten aller Pixel $p \in D_f$ entlang der Geraden h_i
   $\textbf{IF} \quad f(p) \neq 0 \quad \textbf{THEN} \quad f(p) \leftarrow \min\{f(p), 1 + f(N_G^+(p)\}$

Rückwärtsabtasten

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

ursprüngliches Bild

0	0	0	0	0	1	1
0	1	1	1	0	1	1
0	1	1	1	0	0	0
0	1	1	1	0	0	0
0	0	0	0	0	0	0
0	0	0	0	1	1	1
0	0	0	0	1	1	1

Lineare Distanztransformation (4)

Algorithmus:

1. Für alle Parallelen h_i der Geraden g, die sich im Bild befinden:
   Vorwärtsabtasten aller Pixel $p \in D_f$ entlang der Geraden h_i
   $\textbf{IF} \quad f(p) = 1 \quad \textbf{THEN} \quad f(p) \leftarrow 1 + f(N_G^-(p))$
2. Für alle Parallelen p_i der Geraden g, die sich im Bild befinden:
   Rückwärtsabtasten aller Pixel $p \in D_f$ entlang der Geraden h_i
   $\textbf{IF} \quad f(p) \neq 0 \quad \textbf{THEN} \quad f(p) \leftarrow \min\{f(p), 1 + f(N_G^+(p))\}$

Rückwärtsabtasten

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

ursprüngliches Bild

0	0	0	0	0	1	1
0	1	1	1	0	1	1
0	1	1	1	0	0	0
0	1	1	1	0	0	0
0	0	0	0	0	0	0
0	0	0	0	1	1	1
0	0	0	0	1	1	1

Lineare Distanztransformation (4)

<u>Algorithmus:</u>

1. Für alle Parallelen h_i der Geraden g, die sich im Bild befinden:
   Vorwärtsabtasten aller Pixel $p \in D_f$ entlang der Geraden h_i
   $\textbf{IF} \quad f(p) = 1 \quad \textbf{THEN} \quad f(p) \leftarrow 1 + f(N_G^-(p))$
2. Für alle Parallelen p_i der Geraden g, die sich im Bild befinden:
   Rückwärtsabtasten aller Pixel $p \in D_f$ entlang der Geraden h_i
   $\textbf{IF} \quad f(p) \neq 0 \quad \textbf{THEN} \quad f(p) \leftarrow \min\{f(p), 1 + f(N_G^+(p)\}$

Rückwärtsabtasten

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

ursprüngliches Bild

0	0	0	0	0	1	1
0	1	1	1	0	1	1
0	1	1	1	0	0	0
0	1	1	1	0	0	0
0	0	0	0	0	0	0
0	0	0	0	1	1	1
0	0	0	0	1	1	1

Lineare Distanztransformation (4)

Algorithmus:

1. Für alle Parallelen h_i der Geraden g, die sich im Bild befinden:
   Vorwärtsabtasten aller Pixel $p \in D_f$ entlang der Geraden h_i
   $\textbf{IF} \quad f(p) = 1 \quad \textbf{THEN} \quad f(p) \leftarrow 1 + f(N_G^-(p))$
2. Für alle Parallelen p_i der Geraden g, die sich im Bild befinden:
   Rückwärtsabtasten aller Pixel $p \in D_f$ entlang der Geraden h_i
   $\textbf{IF} \quad f(p) \neq 0 \quad \textbf{THEN} \quad f(p) \leftarrow \min\{f(p), 1 + f(N_G^+(p)\}$

Rückwärtsabtasten

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

ursprüngliches Bild

0	0	0	0	0	1	1
0	1	1	1	0	1	1
0	1	1	1	0	0	0
0	1	1	1	0	0	0
0	0	0	0	0	0	0
0	0	0	0	1	1	1
0	0	0	0	1	1	1

Lineare Distanztransformation (4)

Algorithmus:

1. Für alle Parallelen h_i der Geraden g, die sich im Bild befinden:
   Vorwärtsabtasten aller Pixel $p \in D_f$ entlang der Geraden h_i
   $\textbf{IF} \quad f(p) = 1 \quad \textbf{THEN} \quad f(p) \leftarrow 1 + f(N_G^-(p))$
2. Für alle Parallelen p_i der Geraden g, die sich im Bild befinden:
   Rückwärtsabtasten aller Pixel $p \in D_f$ entlang der Geraden h_i
   $\textbf{IF} \quad f(p) \neq 0 \quad \textbf{THEN} \quad f(p) \leftarrow \min\{f(p), 1 + f(N_G^+(p)\}$

Rückwärtsabtasten

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

ursprüngliches Bild

0	0	0	0	0	1	1
0	1	1	1	0	1	1
0	1	1	1	0	0	0
0	1	1	1	0	0	0
0	0	0	0	0	0	0
0	0	0	0	1	1	1
0	0	0	0	1	1	1

Lineare Distanztransformation (4)

<u>Algorithmus:</u>

1. Für alle Parallelen h_i der Geraden g, die sich im Bild befinden:
   Vorwärtsabtasten aller Pixel $p \in D_f$ entlang der Geraden h_i
   $\textbf{IF} \quad f(p) = 1 \quad \textbf{THEN} \quad f(p) \leftarrow 1 + f(N_G^-(p))$
2. Für alle Parallelen p_i der Geraden g, die sich im Bild befinden:
   Rückwärtsabtasten aller Pixel $p \in D_f$ entlang der Geraden h_i
   $\textbf{IF} \quad f(p) \neq 0 \quad \textbf{THEN} \quad f(p) \leftarrow \min\{f(p), 1 + f(N_G^+(p)\}$

Rückwärtsabtasten

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

ursprüngliches Bild

0	0	0	0	0	1	1
0	1	1	1	0	1	1
0	1	1	1	0	0	0
0	1	1	1	0	0	0
0	0	0	0	0	0	0
0	0	0	0	1	1	1
0	0	0	0	1	1	1

Lineare Distanztransformation (4)

Algorithmus:

1. Für alle Parallelen h_i der Geraden g, die sich im Bild befinden:
   Vorwärtsabtasten aller Pixel $p \in D_f$ entlang der Geraden h_i
   $\textbf{IF} \quad f(p) = 1 \quad \textbf{THEN} \quad f(p) \leftarrow 1 + f(N_G^-(p))$
2. Für alle Parallelen p_i der Geraden g, die sich im Bild befinden:
   Rückwärtsabtasten aller Pixel $p \in D_f$ entlang der Geraden h_i
   $\textbf{IF} \quad f(p) \neq 0 \quad \textbf{THEN} \quad f(p) \leftarrow \min\{f(p), 1 + f(N_G^+(p))\}$

Rückwärtsabtasten

0	0	0	0	0	1	1
0	1	2	1	0	1	1
0	1	2	1	0	0	0
0	1	2	1	0	0	0
0	0	0	0	0	0	0
0	0	0	0	1	2	1
0	0	0	0	1	2	1

ursprüngliches Bild

0	0	0	0	0	1	1
0	1	1	1	0	1	1
0	1	1	1	0	0	0
0	1	1	1	0	0	0
0	0	0	0	0	0	0
0	0	0	0	1	1	1
0	0	0	0	1	1	1

Lineare Distanztransformation (4)

Algorithmus:

1. Für alle Parallelen h_i der Geraden g, die sich im Bild befinden:
   Vorwärtsabtasten aller Pixel $p \in D_f$ entlang der Geraden h_i
   $\textbf{IF} \quad f(p) = 1 \quad \textbf{THEN} \quad f(p) \leftarrow 1 + f(N_G^-(p))$
2. Für alle Parallelen p_i der Geraden g, die sich im Bild befinden:
   Rückwärtsabtasten aller Pixel $p \in D_f$ entlang der Geraden h_i
   $\textbf{IF} \quad f(p) \neq 0 \quad \textbf{THEN} \quad f(p) \leftarrow \min\{f(p), 1 + f(N_G^+(p))\}$

Ergebnis (Grauwertbild)

0	0	0	0	0	1	1
0	1	2	1	0	1	1
0	1	2	1	0	0	0
0	1	2	1	0	0	0
0	0	0	0	0	0	0
0	0	0	0	1	2	1
0	0	0	0	1	2	1

ursprüngliches Bild

0	0	0	0	0	1	1
0	1	1	1	0	1	1
0	1	1	1	0	0	0
0	1	1	1	0	0	0
0	0	0	0	0	0	0
0	0	0	0	1	1	1
0	0	0	0	1	1	1

Lineare Distanztransformation (4)

Algorithmus:

1. Für alle Parallelen h_i der Geraden g, die sich im Bild befinden:
   Vorwärtsabtasten aller Pixel $p \in D_f$ entlang der Geraden h_i
   $\textbf{IF} \quad f(p) = 1 \quad \textbf{THEN} \quad f(p) \leftarrow 1 + f(N_G^-(p))$
2. Für alle Parallelen p_i der Geraden g, die sich im Bild befinden:
   Rückwärtsabtasten aller Pixel $p \in D_f$ entlang der Geraden h_i
   $\textbf{IF} \quad f(p) \neq 0 \quad \textbf{THEN} \quad f(p) \leftarrow \min\{f(p), 1 + f(N_G^+(p)\}$

Ergebnis (Grauwertbild)

0	0	0	0	0	1	1
0	1	2	1	0	1	1
0	1	2	1	0	0	0
0	1	2	1	0	0	0
0	0	0	0	0	0	0
0	0	0	0	1	2	1
0	0	0	0	1	2	1

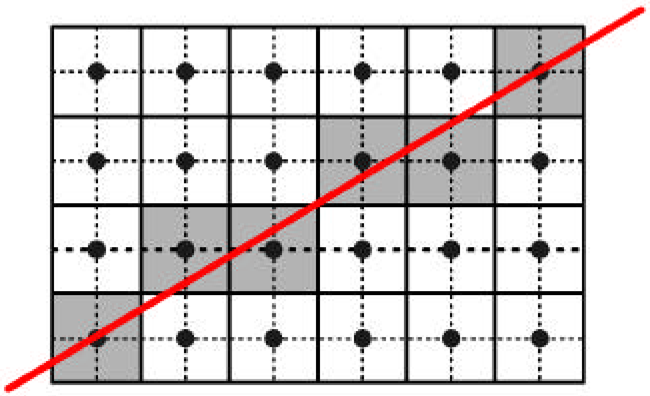
ursprüngliches Bild

0	0	0	0	0	1	1
0	1	1	1	0	1	1
0	1	1	1	0	0	0
0	1	1	1	0	0	0
0	0	0	0	0	0	0
0	0	0	0	1	1	1
0	0	0	0	1	1	1

Lineare Distanztransformation (5)

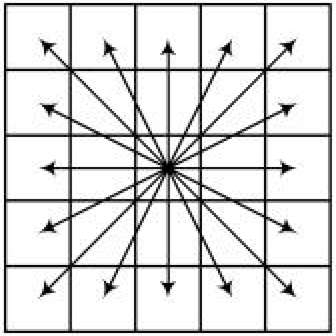
<u>Transformation entlang einer beliebigen Gerade</u>

- Problem: Wie kann man eine Gerade mit beliebiger Steigung diskretisieren?
- Mögliche Lösung (nach **Bresenham 1977**):
  - Auswahl der Pixel durch Distanzkriterium
  - Fiktive vertikale Linien durch die Pixelmittelpunkte ziehen
  - Für jede vertikale Linie gehört das Pixel entlang der kontinuierlichen Linie, dessen Mittelpunkt der vertikalen Linie am nächsten liegt, zu der entsprechenden diskreten Linie.
  - Bei Geraden mit Steigung $m > 1$ werden die Abstände zu den horizontalen Linien berücksichtigt.



Lineare Distanztransformation (6)

- Achtung: Die Winkelauflösung eines diskreten Liniensegmentes ist von der Länge des Segments abhängig.
- In einem quadratischen Gitter gibt es nur $2k$-2 mögliche Richtungen für ein Liniensegment, das eine ungerade Anzahl von k Pixeln enthält.
- Beispiel: 8 mögliche Orientierungen für k=5



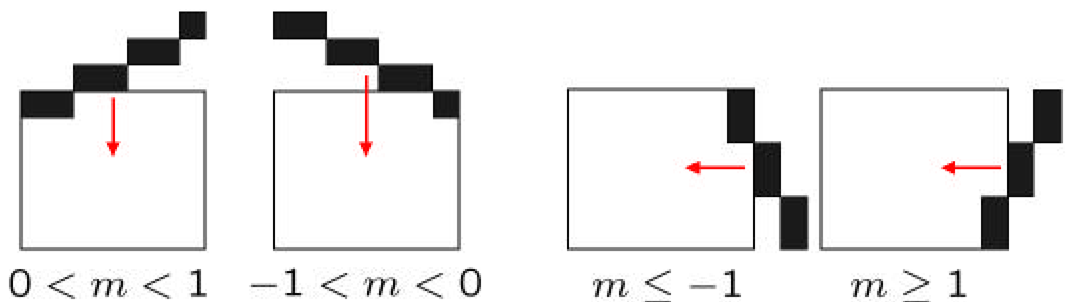
Lineare Distanztransformation (7)

<u>Vorgehen bei linearer Distanztransformation entlang einer beliebigen Linie:</u>

(analog zu Erosion mit linienförmigen strukturierenden Elementen nach **Breen & Soille (1993)** und **Soille et al. (1996)**)

- Linie diskretisieren
- Je nach Steigung der diskreten Linie wird diese von einer geeigneten Bildecke aus gezeichnet.

m: Steigung der Linie



➪ Auf diese Weise kann man durch Verschiebung der Linie in einer Richtung alle Bildpunkte abdecken, ohne dass Überlappungen entstehen.

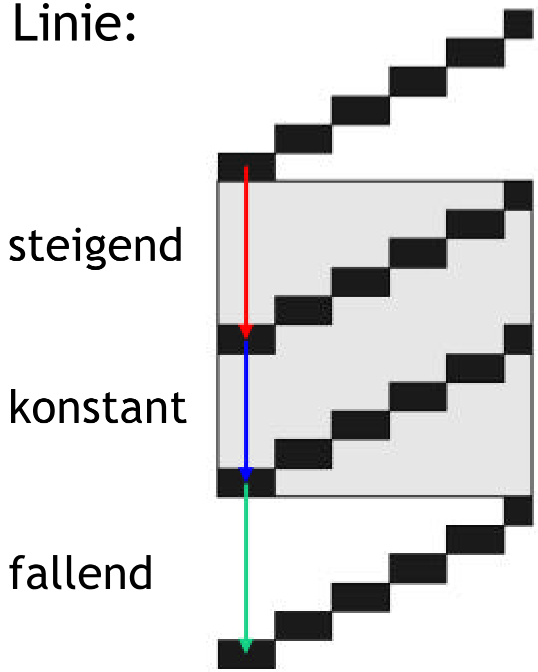
Lineare Distanztransformation (8)

Erweiterter Algorithmus für Linien mit beliebiger Richtung:

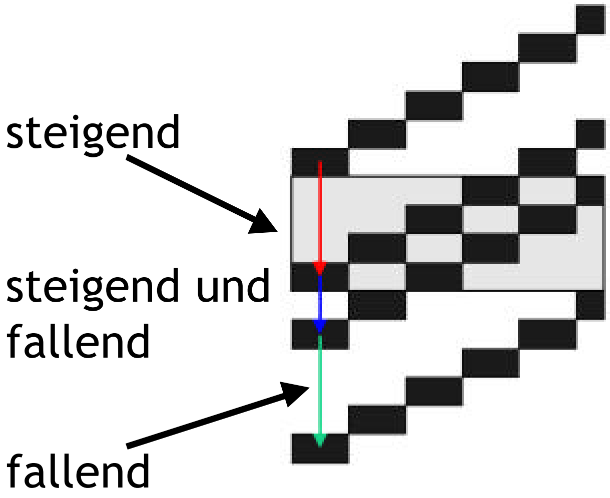
1. Zu Beginn des Algorithmus wird die Linie von der passenden Ecke aus gezeichnet
2. Anwendung der Transformation entlang der Linie
3. Verschiebung der Linie um ein Pixel in der geeigneten Richtung
4. Weiter bei 2.

Lineare Distanztransformation (9)

- Anzahl der von der Distanztransformation betroffenen Pixel auf der Linie:



(a) Es gibt eine vollständige Überdeckung der Linie mit dem Bild

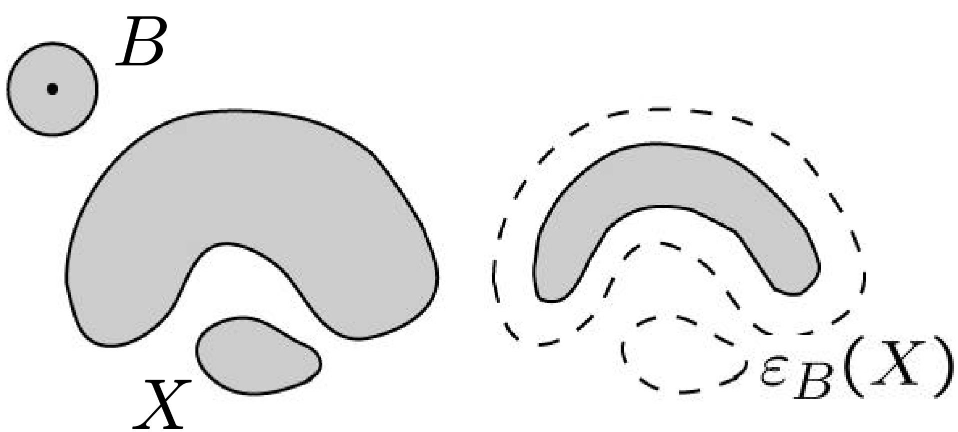


(b) Es gibt keine vollständige Überdeckung der Linie mit dem Bild

Grundlagen: Erosion (1)

Erosion mit einem linearen strukturierenden Element (SE):

- Ein SE ist eine kleine Menge, die zum Proben eines zu untersuchenden Bildes verwendet wird.
- Frage bei der Erosion an jedem Punkt des Bildes:
  „Passt das SE, wenn sein Bezugspunkt das aktuelle Pixel ist, vollständig in die Menge?"
- Erodierte Menge:
  Alle Punkte, an denen diese Frage mit JA beantwortet werden kann



Grundlagen: Erosion (2)

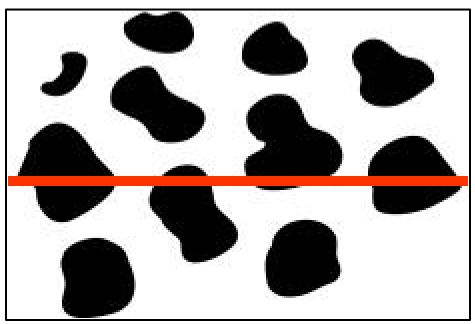
- Die Erosion einer Menge X durch ein strukturierendes Element B wird mit $\varepsilon_B(X)$ bezeichnet

$$\boxed{\varepsilon_B(X) = \{x | B_x \subseteq X\}}$$

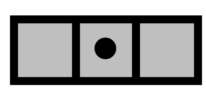
- Intuitiv:
  Menge aller Pixel x, so dass B vollständig in X enthalten ist, wenn sich sein Bezugspunkt am Pixel x befindet.

Sehnenlängenverteilung (1)

- Gegeben:
  - ein linearer Schnitt $\Phi := \Xi \cap e_\omega$ durch eine Mikrostruktur Ξ, wobei e_ω eine Linie mit Richtung ω ist. Dieser lineare Schnitt bildet nun ein System von Sehnen.

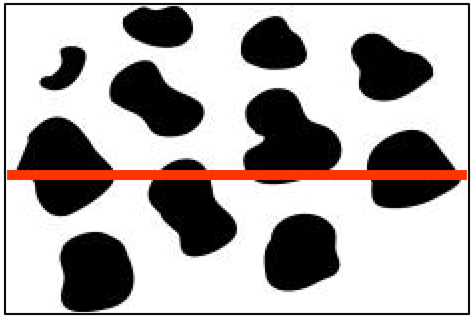


  - Ein Liniensegment $s_{r,\omega} = [o,(r,\omega)]$ der Länge r als SE, das eine Untermenge der Testlinie e_ω ist, also $s_{r,\omega} \subset e_\omega$.
    - ➪ $\Phi \ominus s_{r,\omega}$ stellt das System von Sehnen dar, aber erodiert um das lineare strukturierende Element $s_{r,\omega}$



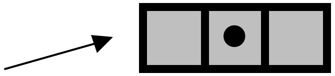
Sehnenlängenverteilung (1)

- Gegeben:
  - ein linearer Schnitt $\Phi := \Xi \cap e_\omega$ durch eine Mikrostruktur Ξ, wobei e_ω eine Linie mit Richtung ω ist. Dieser lineare Schnitt bildet nun ein System von Sehnen.



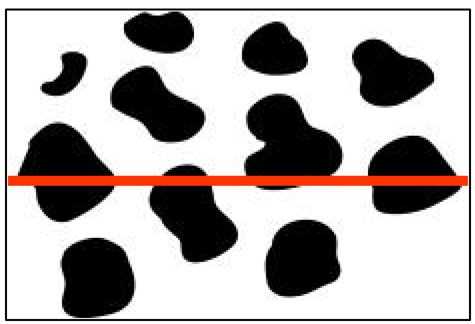
  - Ein Liniensegment $s_{r,\omega} = [o,(r,\omega)]$ der Länge r als SE, das eine Untermenge der Testlinie e_ω ist, also $s_{r,\omega} \subset e_\omega$.
    - ➪ $\Phi \ominus s_{r,\omega}$ stellt das System von Sehnen dar, aber erodiert um das lineare strukturierende Element $s_{r,\omega}$

verwendetes linienförmiges SE (vergrößert)



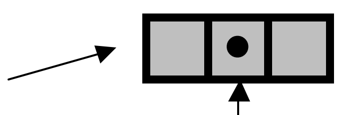
Sehnenlängenverteilung (1)

- Gegeben:
  - ein linearer Schnitt $\Phi := \Xi \cap e_\omega$ durch eine Mikrostruktur Ξ, wobei e_ω eine Linie mit Richtung ω ist. Dieser lineare Schnitt bildet nun ein System von Sehnen.



  - Ein Liniensegment $s_{r,\omega}$ =[o,(r,ω)] der Länge r als SE, das eine Untermenge der Testlinie e_ω ist, also $s_{r,\omega} \subset e_\omega$.
    ➪ $\Phi \ominus s_{r,\omega}$ stellt das System von Sehnen dar, aber erodiert um das lineare strukturierende Element $s_{r,\omega}$

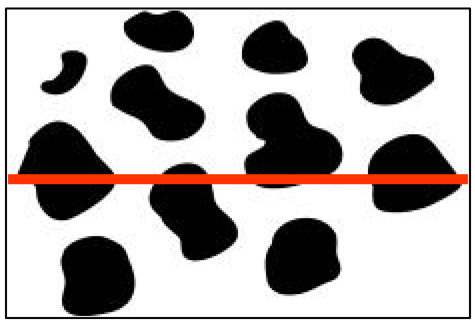
verwendetes linienförmiges SE (vergrößert)



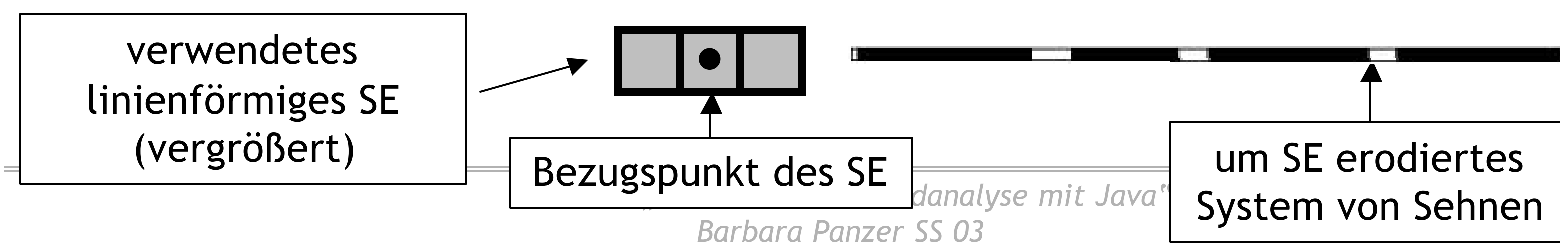
Bezugspunkt des SE

Sehnenlängenverteilung (1)

- Gegeben:
  - ein linearer Schnitt $\Phi := \Xi \cap e_\omega$ durch eine Mikrostruktur Ξ, wobei e_ω eine Linie mit Richtung ω ist. Dieser lineare Schnitt bildet nun ein System von Sehnen.



  - Ein Liniensegment $s_{r,\omega}$ =[o,(r,ω)] der Länge r als SE, das eine Untermenge der Testlinie e_ω ist, also $s_{r,\omega} \subset e_\omega$.
    ➪ $\Phi \ominus s_{r,\omega}$ stellt das System von Sehnen dar, aber erodiert um das lineare strukturierende Element $s_{r,\omega}$



Sehnenlängenverteilung (2)

- Definition:
  Verteilungsfunktion $F_L(r)$ der Sehnenlängen

$$F_L(r) = 1 - \frac{\chi_L(\Phi \ominus s_{r,w})}{\chi_L(\Phi)} \qquad r \geq 0$$

$\chi_L(x)$: Anzahl der Sehnen entlang der Linie x

Star Volume

- „Star" S_x am Pixel x:
  Menge aller Punkte in einem Bild, die man vom Punkt x aus direkt sehen kann, d.h. ohne durch ein Vordergrundobjekt hindurch sehen zu müssen.
- Definition:

$$\boxed{S_x = \{y \in \Xi : px + (1-p)y \in \Xi, \;\; 0 \leq p \leq 1\}}$$

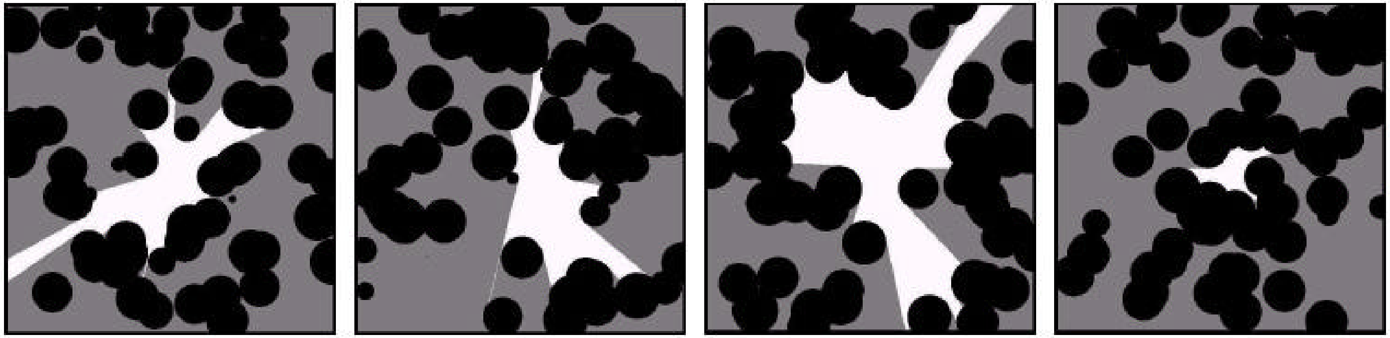
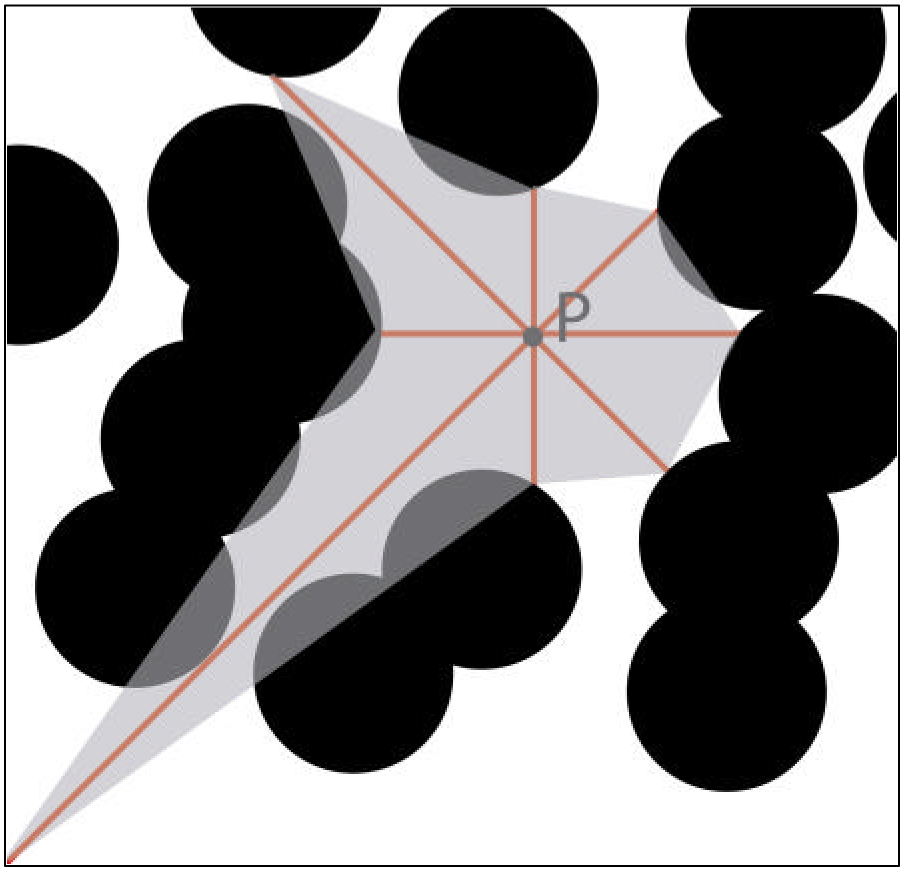
- Die Fläche von S_x ist das sog. Star Volume.
- 

Abb.: Beispiele für das Star Volume
(aus [Ohser, Mücklich] S.119)

Schätzer für das Star Volume (1)

- Das Star Volume an einem bestimmten Pixel eines Bildes kann mit Hilfe eines Polygons angenähert werden.



Fläche des Polygons „um“ P
=
Schätzer für das Star Volume am Punkt P

Schätzer für das Star Volume (2)

Schätzer für das Star-Volume wird an jedem Punkt des Bildes berechnet ➪ Ergebnis: Grauwertbild

Originalbild

Star Volume in 8 Richtungen
➪ Zwischenwinkel $\theta = 45^\circ$

Schätzer für das Star Volume (3)

Originalbild

Star Volume in 16 Richtungen
➪ Zwischenwinkel θ = 22,5°

Schätzer für das Star Volume (4)

Originalbild

Star Volume in 32 Richtungen
➪ Zwischenwinkel $\theta = 11{,}25^\circ$

Schätzer für das Star Volume (5)

Originalbild

Star Volume in 64 Richtungen
➪ Zwischenwinkel $\theta = 5{,}625^\circ$

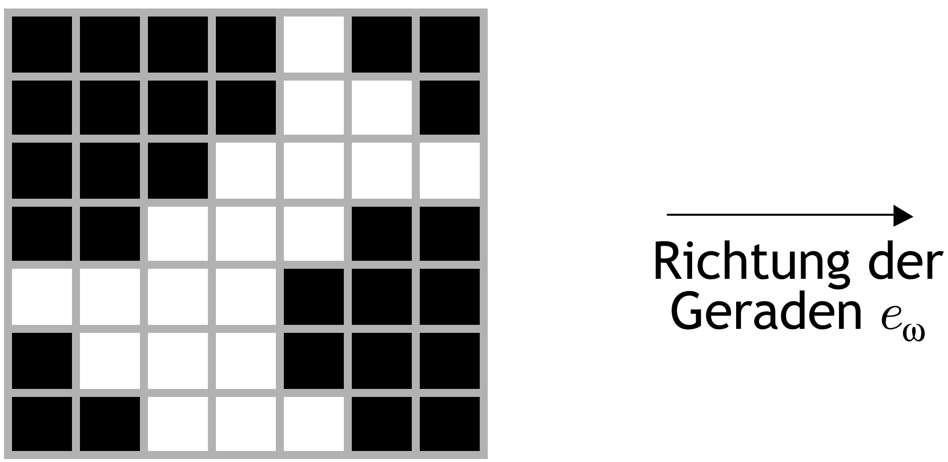
Sehnenlängentransformation

- Die Sehnenlängentransformation Ch_ω, angewendet auf ein Binärbild f, verbindet jedes Pixel p des Definitionsbereichs D_f von f mit der Länge der Sehne entlang der Geraden e_ω an diesem Pixel. Hierbei ist e_ω wiederum eine Gerade mit Richtung ω.

$$\boxed{[Ch_\omega(f)](x) = length(ch_\omega(x))}$$

$ch_w(x)$: Sehne am Pixel x mit Richtung ω

- Beispiel:



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

Literatur

- Soille, Pierre:
  Morphologische Bildverarbeitung
  Grundlagen, Methoden, Anwendungen
  Berlin: Springer 1998
- Ohser, J.; Mücklich, F.:
  Statistical Analysis of Microstuctures in Materials Science
  2001
  http://www.itwm.fhg.de/mab/employees/lecturews01/Book.ps.gz
- Torquato, S.; Lu, B.:
  Chord-length distribution of two-phase random media
  Phys. Rev. E 47 no.4 (1993) S.2950-3