



## Morphologie auf Binärbildern

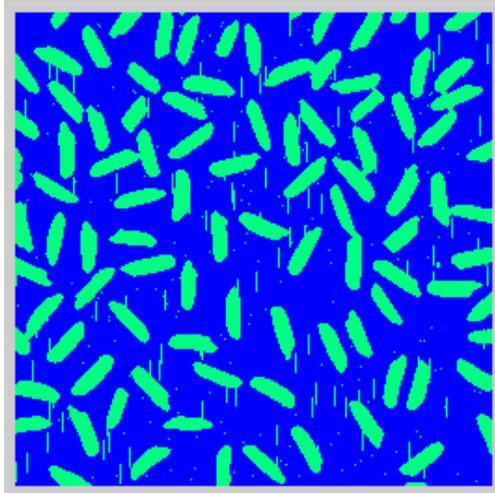
## Motivation

---

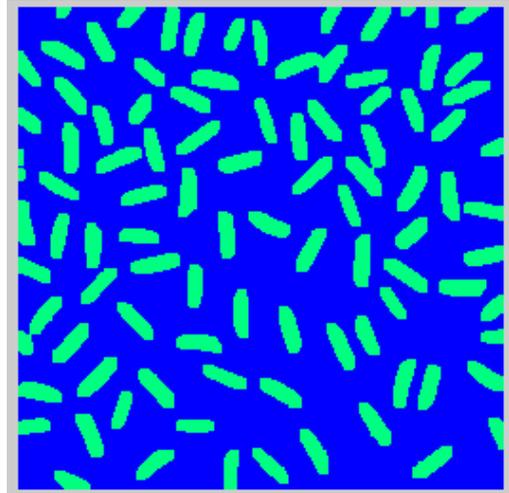
□ Aufgabe: Objekte zählen

□ Probleme:

- "Salt-&-Pepper"-Rauschen erzeugt falsche Objekte
- Verschmelzen richtiger Objekte durch falsche Linien
- Objekte verschmelzen, weil sie aneinanderstoßen



Lösung →



---

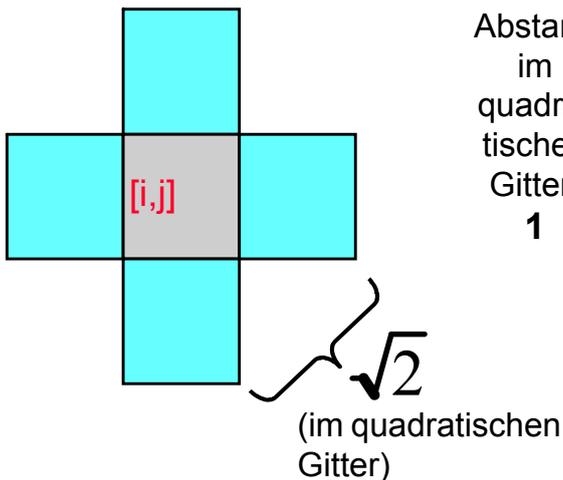
□ Wie kann man die Störungen wegbekommen?

- **isolierte Kleinstobjekte: minimale Fläche fordern**
- **Linien: Mindestdicke fordern >> aber wie effizient umsetzen, verschiedene Orientierungen**
- **Objektverschmelzung: wenn "Landzunge" zu klein, dann "Kanal" bauen >> wie setzen wir das um?**

# Nachbarschaften in digitalen Bildern (im quadratischen/rechteckigen Gitter)

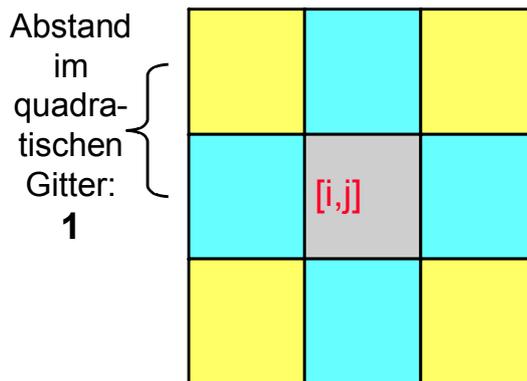
## □ 4-Nachbarschaft:

$[i+1, j]$ ,  $[i-1, j]$ ,  
 $[i, j+1]$ ,  $[i, j-1]$



## □ 8-Nachbarschaft:

$[i+1, j+1]$ ,  $[i+1, j-1]$ ,  
 $[i-1, j+1]$ ,  $[i-1, j-1]$   
und alle 4-Nachbarn



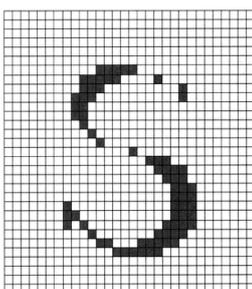
Bildverarbeitung und Algorithmen  
WS07 5.5 ©Konen, Zielke

## Dilatation und Erosion (1)

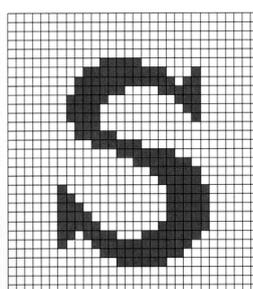
- Ein binäres Bildobjekt (connected component) kann systematisch ausgedehnt (dilatiert) oder geschrumpft (erodiert) werden.

Ein einfaches Verfahren ist:

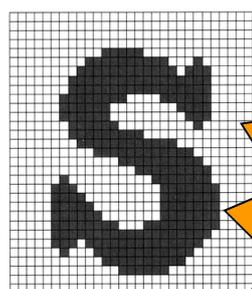
- **Dilatation (wirkt expandierend)**  
Ändere ein Pixel von '0' zu '1' wenn mindestens ein Nachbarpixel '1' ist.
- **Erosion (wirkt schrumpfend)**  
Ändere ein Pixel von '1' zu '0' wenn mindestens ein Nachbarpixel '0' ist.



erodiertes  
Bild



Original  
( '0': hell, '1': dunkel )



dilatiertes  
Bild

Aktivierung:  
Welche  
Nachbarschaft?

Bildverarbeitung und Algorithmen  
WS07 5.6 ©Konen, Zielke

## Dilatation und Erosion (2)

---

- Die Dilatation ist *nicht* die Umkehrung der Erosion!
- Aktivierung: Wieso???**
- Dilatation gefolgt von Erosion wird benutzt, um ungewollte "Löcher" in binären Bildobjekten zu füllen.
- Erosion gefolgt von Dilatation kann isolierte Stör-Pixel entfernen.
- Generell können mit Kombinationen von Dilatationen und Erosionen sehr nützliche und scheinbar komplexe Operationen auf binären Bildern realisiert werden. Die Klasse dieser Operationen bezeichnet man als **Morphologische Filter**.



Bildverarbeitung und Algorithmen  
WS07 5.7 ©Konen, Zielke

## Morphologische Filter

---

- In der Theorie der **Morphologischen\* Bildverarbeitung** wird dieses Prinzip von 4er- oder 8er-Nachbarschaft auf beliebige **Strukturelemente** übertragen

\*Morphologie ist die Wissenschaft von Gestalt, Form, und Struktur von Körpern

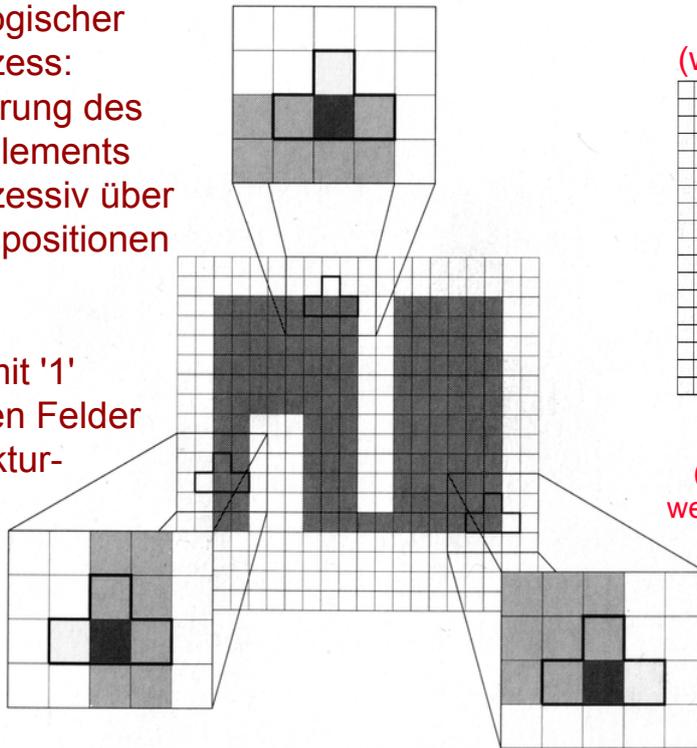
Lit.: [Jähne02, Kap. 18, S. 507-521]

Bildverarbeitung und Algorithmen  
WS07 5.8 ©Konen, Zielke

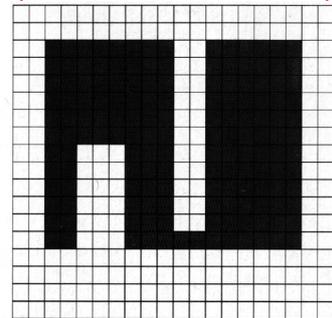
# Morphologische Filterung mit Strukturelement (structuring element)

Morphologischer Filterprozess:  
Der Ursprung des Strukturelements wird sukzessiv über allen Bildpositionen plaziert.

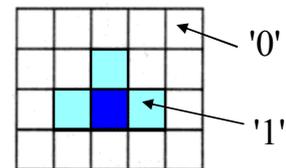
Nur die mit '1' markierten Felder des Strukturelements werden berücksichtigt.



**Binärbild**  
(weiß = 0, dunkel = 1)



**Strukturelement**  
(Ursprung dunkelblau, weiße Felder = don't care)



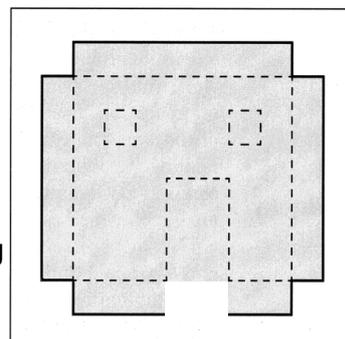
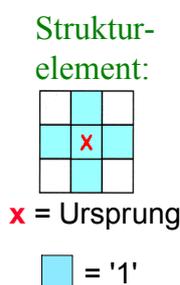
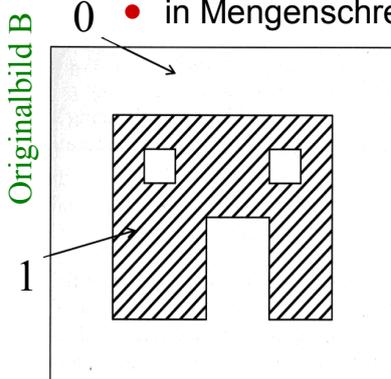
Bildverarbeitung und Algorithmen  
WS07 5.9 ©Konen, Zielke

## Dilatation (dilation)

□ Die Dilatation wird dadurch ausgeführt, dass ein **Strukturelement** (structuring element) als eine Art "Filtermaske" über alle Bildpositionen  $p$  "geschoben" wird:

- Das Strukturelement besteht aus '1'-Pixeln und einem Referenzpunkt (Ursprung).  $S_p$ : Referenzpunkt in  $p$ .
- Wenn an den '1'-Pixeln des Strukturelementes  $S_p$  mind. ein Bildpixel '1' ist, dann setze im Zielbild den Referenzpunkt  $p$  auf 1, sonst 0.

• in Mengenschreibweise:  $Z = \{p \mid S_p \cap B \neq \emptyset\}$

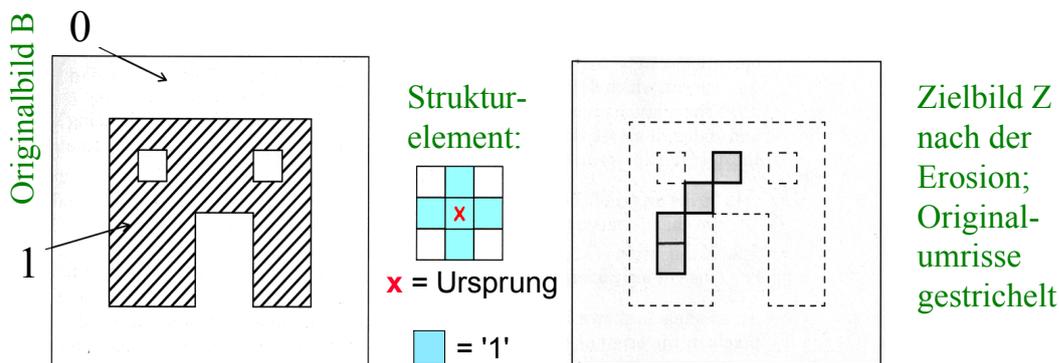


Zielbild Z  
nach der  
Dilatation;  
Original-  
umrisse  
gestrichelt

Bildverarbeitung und Algorithmen  
WS07 5.10 ©Konen, Zielke

## Erosion (erosion)

- Die Erosion wird dadurch ausgeführt, dass ein *Strukturelement* als "Filtermaske" über alle Bildpositionen "geschoben" wird:
  - Sei  $S_p$  das in den Referenzpunkt  $p$  verschobene Strukturelement
  - Wenn das Strukturelement  $S_p$  mit seinen '1'-Pixeln komplett in '1'-Pixeln des Bildes liegt, dann setze im Zielbild den Referenzpunkt  $p$  auf 1, sonst 0.
  - in Mengenschreibweise:  $Z = \{p \mid S_p \subseteq B\}$



Bildverarbeitung und Algorithmen  
WS07 5.11 ©Konen, Zielke

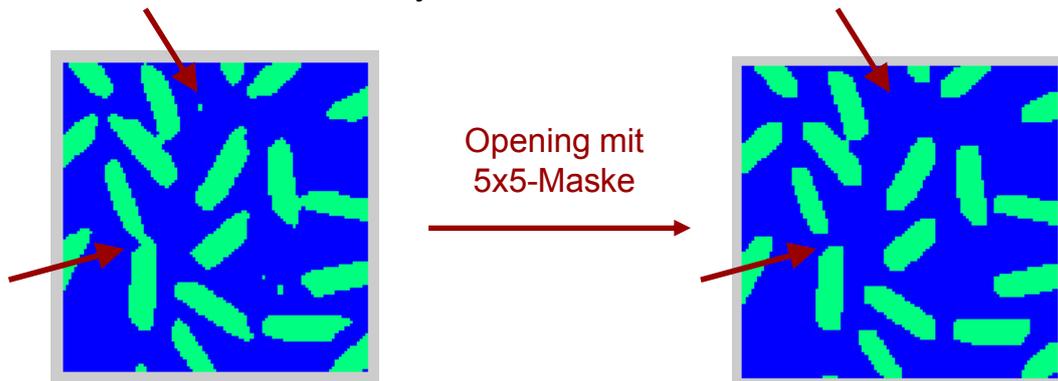
## Öffnung und Schließung (opening and closing)

- Eine Erosion gefolgt von einer Dilatation wird **Opening (Öffnung)** genannt.
- Eine Dilatation gefolgt von einer Erosion wird **Closing (Schließung)** genannt.

Bildverarbeitung und Algorithmen  
WS07 5.12 ©Konen, Zielke

## Öffnung (Opening)

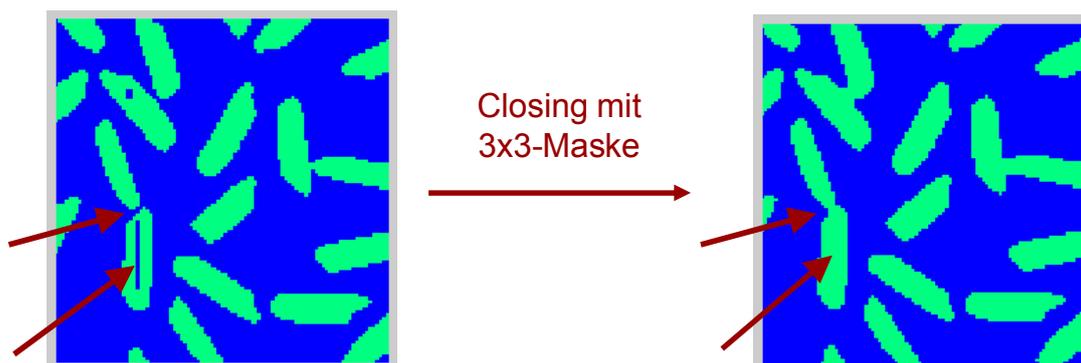
- Nutzen der Erosion: kleine Störobjekte werden entfernt
- Nachteil: alle Objekte werden kleiner
- Abhilfe: Opening
  - $\text{opening()} = \text{dilate}(\text{erode}())$
  - "öffnet" (trennt) dünn verbundene Objekte
  - eliminiert kleine Störobjekte und dünne Linien



Bildverarbeitung und Algorithmen  
WS07 5.13 ©Konen, Zielke

## Schließung (Closing)

- Nutzen der Dilatation: kleine Löcher werden geschlossen
- Nachteil: alle Objekte werden größer
- Abhilfe: Closing
  - $\text{closing()} = \text{erode}(\text{dilate}())$
  - "schließt" kleine Löcher und dünne Risse
  - nahe Objekte verschmelzen ("close the gap")



Bildverarbeitung und Algorithmen  
WS07 5.14 ©Konen, Zielke

## Aktivierung: Füllen Sie die Tabelle aus! Welche Wirkung haben die Operatoren?

	Erosion	Opening	Dilatation	Closing
kleine Obj. *, dünne Linien				
alle anderen Objekte				
kleine Löcher dünne Risse				
alle anderen Objekte				
nahe Objekte				
dünn verbund. Objekte				

\* Objekte, die Strukturelement nicht ganz aufnehmen können

Bildverarbeitung und Algorithmen  
WS07 5.15 ©Konen, Zielke

## Fazit: Wirkung morphologischer Operatoren

	Erosion	Opening	Dilatation	Closing
kleine Obj. *, dünne Linien	verschwinden		größer	gleich
alle anderen Objekte	werden kleiner	gleich groß	größer	gleich
kleine Löcher dünne Risse	größer	gleich	verschwinden	
alle anderen Objekte			werden größer	gleich groß
nahe Objekte	weiter entfernt	gleich	verschmelzen (wenn Riss länger als Strukturelement)	
dünn verbund. Objekte	werden getrennt		Verb. größer	gleich

\* Objekte, die das Strukturelement nicht ganz aufnehmen können

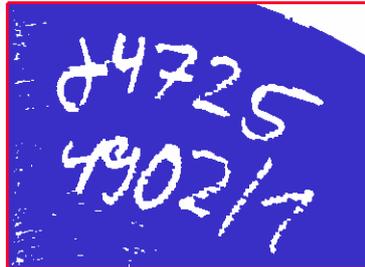
Bildverarbeitung und Algorithmen  
WS07 5.16 ©Konen, Zielke

## Morphologische Filterung weitere Beispiele

---

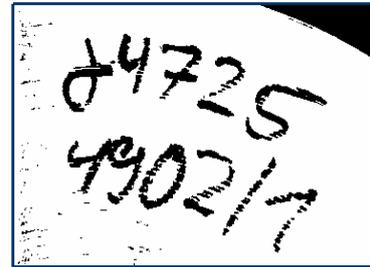


Opening

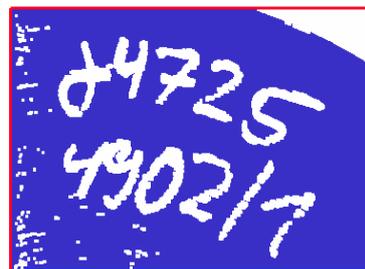


Closing

Original-Binärbild



Erosion



Dilatation



Bildverarbeitung und Algorithmen  
WS07 5.17 ©Konen, Zielke

## Bildverarbeitung und Algorithmen

---

Morphologie auf  
Grauwertbildern

## Morphologische Operatoren auf Grauwertbildern

- Bild: Grauwerte  $v$ ,
- Maske (Strukturelement)  $H$ : Grauwerte + Punktmenge.
- Dilatation:

$$(I \oplus H)(u, v) = \max_{(i,j) \in H} \{I(u+i, v+j) + H(i, j)\}$$

- Erosion:

$$(I \ominus H)(u, v) = \min_{(i,j) \in H} \{I(u+i, v+j) - H(i, j)\}$$

- Beachte: Für  $H$  ist 0 und  $x =$  „don't care“ verschieden!

$$\begin{array}{|c|c|c|} \hline 0 & 1 & 0 \\ \hline 1 & \mathbf{2} & 1 \\ \hline 0 & 1 & 0 \\ \hline \end{array} \neq \begin{array}{|c|c|c|} \hline \times & 1 & \times \\ \hline 1 & \mathbf{2} & 1 \\ \hline \times & 1 & \times \\ \hline \end{array} = \begin{array}{|c|c|c|} \hline & 1 & \\ \hline 1 & \mathbf{2} & 1 \\ \hline & 1 & \\ \hline \end{array}$$

arbeitung und Algorithmen  
WS07 5.19 ©Konen, Zielke

## Grauwert-Dilatation

$$\begin{array}{|c|c|c|c|} \hline I & & & \\ \hline 6 & 7 & 3 & 4 \\ \hline 5 & 6 & 6 & 8 \\ \hline 6 & 4 & 5 & 2 \\ \hline 6 & 4 & 2 & 3 \\ \hline \end{array} \oplus \begin{array}{|c|c|c|} \hline H & & \\ \hline 1 & 1 & 1 \\ \hline 1 & 2 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array} = ?$$

- Aktivierung: Welcher Grauwert gehört an die markierte Stelle?
- Welche Grauwerte gehören an die anderen möglichen Stellen?

## Grauwert-Erosion

$$\begin{array}{c} I \\ \begin{array}{|c|c|c|c|} \hline 6 & 7 & 3 & 4 \\ \hline 5 & 6 & 6 & 8 \\ \hline 6 & 4 & 5 & 2 \\ \hline 6 & 4 & 2 & 3 \\ \hline \end{array} \ominus \begin{array}{c} H \\ \begin{array}{|c|c|c|} \hline 1 & 1 & 1 \\ \hline 1 & 2 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array} = ? \end{array}$$

- Aktivierung: Welcher Grauwert gehört an die markierte Stelle?
- Welche Grauwerte gehören an die anderen möglichen Stellen?

Bildverarbeitung und Algorithmen  
WS07 5.21 ©Konen, Zielke

## Beispiele zur Grauwert-Morphologie

- [Bilder aus \[BurgerBurge05, Abb. 10.19ff\]](#)
- Wirkung: mit geeigneten Strukturelementen lassen sich einfach interessante Effekte erzielen („Pointilismus“, Pinselstriche)

Aktivierung: Welche Filter entstehen, wenn (a) bei Dilatation und (b) bei Erosion alle  $H(i,j)=0$  sind?

Bildverarbeitung und Algorithmen  
WS07 5.22 ©Konen, Zielke

## Spezialfall $H(i,j)=0$

- Bild: Grauwerte  $v$ ,
- Maske (Strukturelement)  $H: (0,0,\dots,0)$
- Es ergibt sich der **Minimum-** bzw. **Maximum-Filter**

Erosion	$(\text{erode}(I))(u,v) = \min_{(i,j) \in H} (I(u+i, v+j))$	
Dilatation	$(\text{dilate}(I))(u,v) = \max_{(i,j) \in H} (I(u+i, v+j))$	
Opening	$\text{dilate}(\text{erode}())$	setzt kleine, helle Objekte auf Hintergrund-Grauwert, lässt aber große Objekte gleich
Closing	$\text{erode}(\text{dilate}())$	entfernt kleine, dunkle Background-Einsprengsel, verbindet helle, knapp getrennte Komponenten

Bildverarbeitung und Algorithmen  
WS07 5.23 ©Konen, Zielke

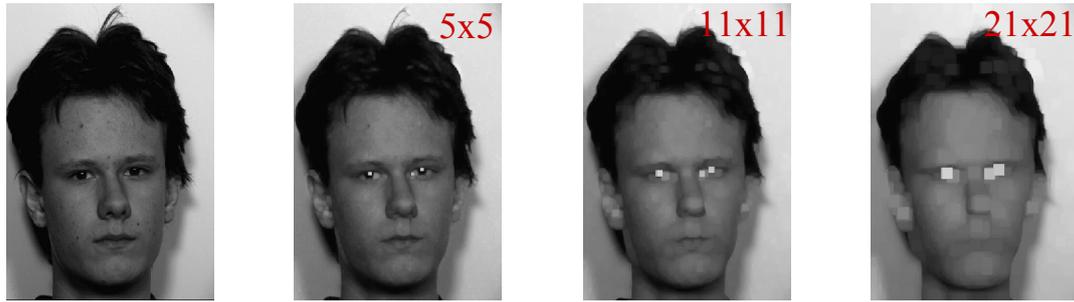
## Minimum-Filter



- Der Minimum-Filter wird so berechnet, dass der Ergebniswert für eine Pixel-Nachbarschaft das Minimum der Grauwerte in dieser Nachbarschaft ist.
- Der Minimum-Filter entspricht einer **Erosion** in der grauwert-basierten morphologischen Bildverarbeitung.
- Anwendungen:  
Hervorhebung (Kontrastverstärkung) von kleinen dunklen Strukturen, z.B. bei Strichzeichnungen.

Bildverarbeitung und Algorithmen  
WS07 5.24 ©Konen, Zielke

## Maximum-Filter

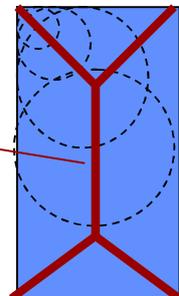


- Der Maximum-Filter wird so berechnet, dass der Ergebniswert für eine Pixel-Nachbarschaft das Maximum der Grauwerte in dieser Nachbarschaft ist.
- Der Maximum-Filter entspricht einer **Dilatation** in der grauwert-basierten morphologischen Bildverarbeitung.
- Anwendungen:  
Hervorhebung (Vergrößerung) von kleinen hellen Strukturen vor dunklem Hintergrund.

Bildverarbeitung und Algorithmen  
WS07 5.25 ©Konen, Zielke

## (Weitere) Anwendungen Morphologie

- Backgroundbild mit Helligkeits-Gradienten frei von Objekten bekommen
  - s. [Aufgabe Segmentierung](#)
- Skelettierung, Verdünnung (Thinning)
  - Skelett: Orte der Zentren aller maximalen einbeschriebenen Kreise.
  - Bsp.: Zusammenhangsstrukturen auf Leiterplatten
  - OCR, Erkennen von Buchstaben: gleiche Verzweigungsstruktur im Skelett



- Hit-or-Miss-Operator
  - [Tönnies05, S. 284]
- Distanztransformation
  - Wie weit ist Pixel vom Objekttrand weg?
  - wdh. Erode anwenden

Bildverarbeitung und Algorithmen  
WS07 5.26 ©Konen, Zielke

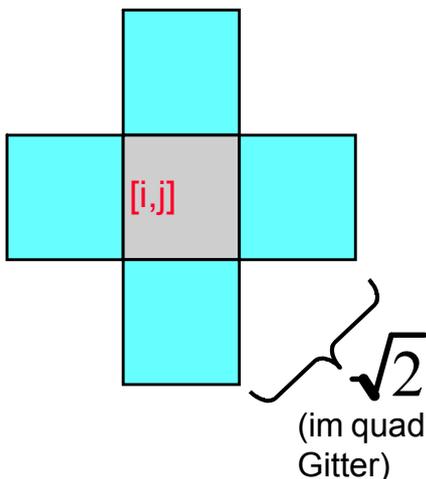
## Nachbarschaften

## Nachbarschaften in digitalen Bildern (im quadratischen/rechteckigen Gitter)

---

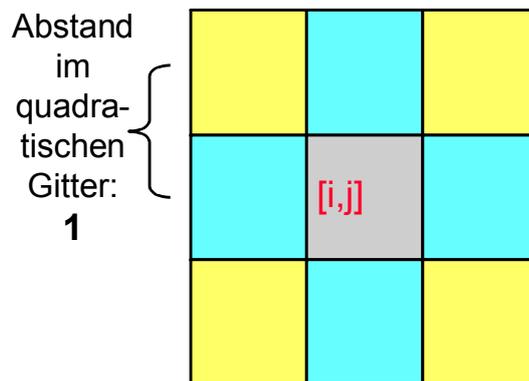
□ 4-Nachbarschaft:

$[i+1, j]$ ,  $[i-1, j]$ ,  
 $[i, j+1]$ ,  $[i, j-1]$



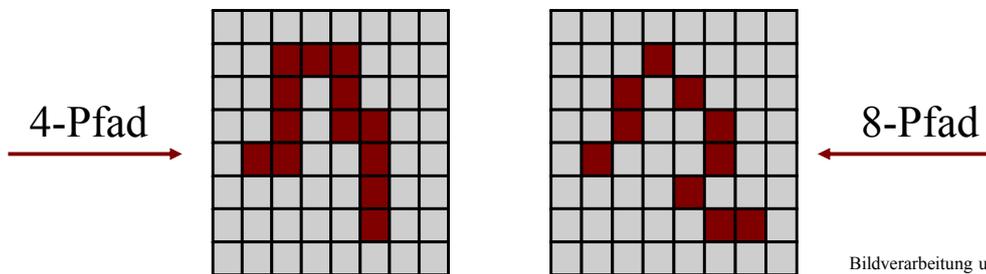
□ 8-Nachbarschaft:

$[i+1, j+1]$ ,  $[i+1, j-1]$ ,  
 $[i-1, j+1]$ ,  $[i-1, j-1]$   
und alle 4-Nachbarn



## Pfad und Konnektivität (path and connectivity)

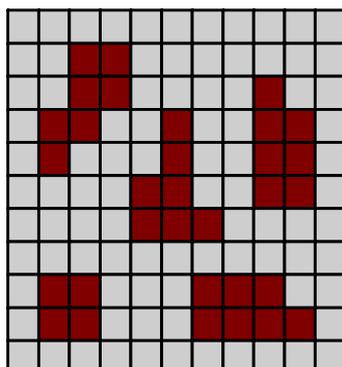
- Ein *Pfad* von dem Pixel an Position  $[i_0, j_0]$  zu dem Pixel an  $[i_n, j_n]$  ist eine Sequenz von Pixel-Positionen  $[i_0, j_0], [i_1, j_1], \dots, [i_n, j_n]$  so dass der Pixel an  $[i_k, j_k]$  ein Nachbar des Pixels an  $[i_{k+1}, j_{k+1}]$  ist, und zwar für alle  $k$  mit  $0 \leq k \leq n-1$ .
- Wenn die Nachbarschaftsrelation **4-verbunden** ist (4-Nachbarschaft), dann ist der Pfad ein **4-Pfad**; bei einer 8-Nachbarschaft ist der Pfad ein **8-Pfad**.
- Ein Pixel  $p \in S$  wird als **verbunden (connected)** mit  $q \in S$  bezeichnet, wenn es einen Pfad von  $p$  nach  $q$  gibt, der vollständig aus Pixeln der Menge  $S$  besteht.  $S$  kann z.B. die Menge der Pixel mit einem bestimmten Farb- bzw. Grauwert sein.



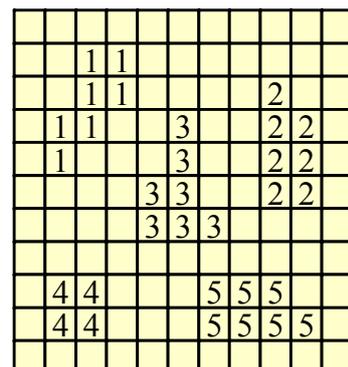
Bildverarbeitung und Algorithmen  
WS07 5.29 ©Konen, Zielke

## Connected Components (Zusammenhangskomponenten)

- Eine Menge von Pixeln, in der jedes Pixel mit allen anderen Pixeln *verbunden* ist, bezeichnet man als **connected component**.
- In der industriellen Bildverarbeitung (machine vision) ist das Finden von connected components in einem Bild eine häufig verwendete Operation → **component labeling**.  
In der Literatur werden etliche Connected Components Algorithmen beschrieben, die sich hauptsächlich im Rechenaufwand unterscheiden.

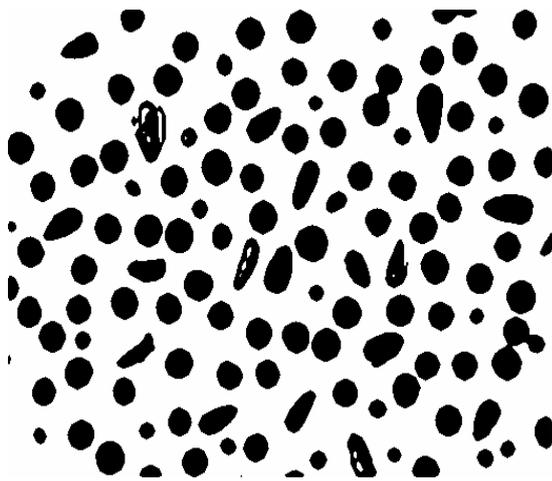


Connected Component  
Algorithmus

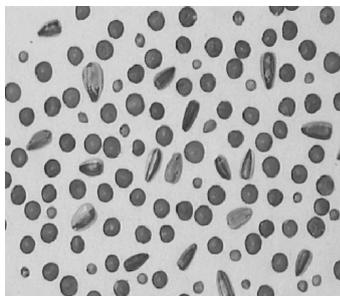
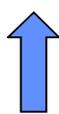


gorithmen  
WS07 5.29 ©Konen, Zielke

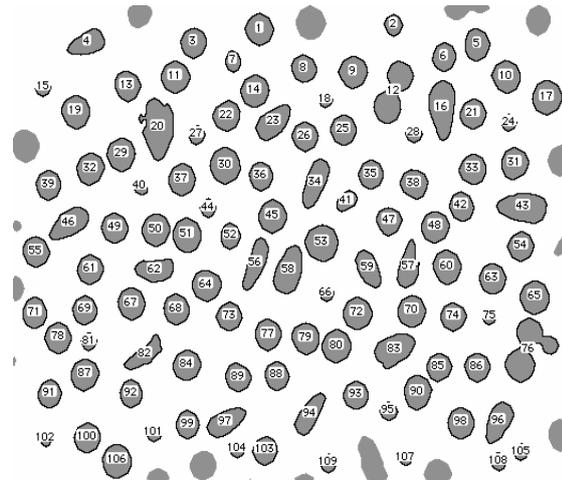
## Analyse von Partikeln mit Hilfe von Component Labeling (Beispiel)



Binarisierung



Labeling



LABEL	AREA	X	Y	LENGTH
1	581.00	229.69	392.36	91.40
2	258.00	354.79	396.11	61.25
3	514.00	168.23	380.61	84.91
4	577.00	69.08	380.53	97.15
...	...	...	...	...
108	197.00	452.76	17.29	52.53
109	204.00	294.15	15.66	53.60

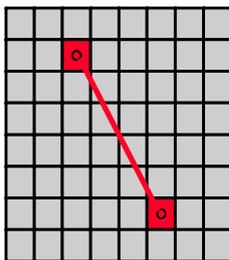
Auswertung



Bildverarbeitung und Algorithmen  
WS07 5.31 ©Konen, Zielke

## Distanzmaße (distance measures)

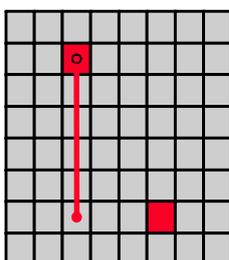
- In vielen Anwendungen muss der Abstand zwischen zwei Pixel-Positionen berechnet werden. In der digitalen Geometrie gibt es hierfür mehrere Distanzmaße. Die wichtigsten sind:



Euklidische Distanz zwischen zwei Pixeln p und q:

$$D_{Euclid}(p, q) = \sqrt{(i_p - i_q)^2 + (j_p - j_q)^2}$$

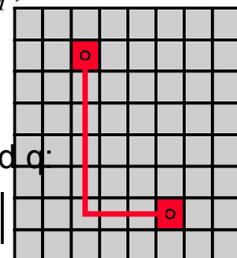
Beispiel:  $D_{Euclid} = 5.83$



Schachbrett Distanz zwischen zwei Pixeln p und q:

$$D_{Chess}(p, q) = \max(|i_p - i_q|, |j_p - j_q|)$$

Beispiel:  $D_{Chess} = \max(5, 3) = 5$



City-Block Distanz zwischen zwei Pixeln p und q:

$$D_{City}(p, q) = |i_p - i_q| + |j_p - j_q|$$

Beispiel:  
 $D_{City} = 5 + 3 = 8$

Bildverarbeitung und Algorithmen  
WS07 5.32 ©Konen, Zielke