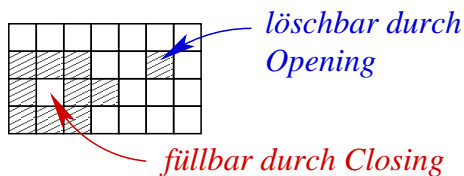


3.1.2 Morphologie (1)

Die Morphologischen Operationen

- elementare Operationen: *dilate, erode*
- zusammengesetzte Operationen: *open, close*

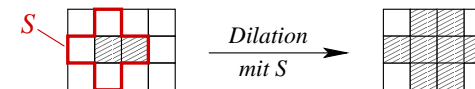
Open und *Close* können zur Noise Reduction eingesetzt werden:



1

3.1.2 Morphologie (3)

Beispiel für Dilation:



Dilation und Erosion sind nicht invers

- im allg. ist $B \neq (B \oplus S) \ominus S \neq (B \ominus S) \oplus S$
- *Closing*: $B \bullet S := (B \oplus S) \ominus S$
 - ▷ kann Löcher füllen
 - ▷ verbindet ggf. zuvor unverbundene CCs
- *Opening*: $B \circ S := (B \ominus S) \oplus S$
 - ▷ kann kleine Objekte und dünne Linien entfernen
 - ▷ CCs werden ggf. aufgetrennt and dünnen Stellen

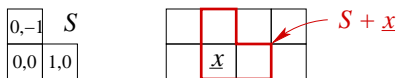
3

3.1.2 Morphologie (2)

Definition Dilation und Erosion

- ein *Strukturelement* S wandert über das Bild
 - ▷ das Strukturelement ist eine Menge von Punkten
- neuer Bildwert an \underline{x} hängt ab von Bildwerten an Stellen

$$S + \underline{x} := \{ \underline{z} + \underline{x} \mid \underline{z} \in S \}$$



- Dilation: $(B \oplus S)(\underline{x}) = \max \{ B(\underline{z}) \mid \underline{z} \in S + \underline{x} \}$
- Erosion: $(B \ominus S)(\underline{x}) = \min \{ B(\underline{z}) \mid \underline{z} \in S + \underline{x} \}$
- primär für Onebit-Bilder gedacht;
Definitionen gelten so aber auch für Greyscale-Bilder

2

3.1.2 Morphologie (4)

Eigenschaften der Dilation von Onebit-Bildern

- Dilation eines Bildes mit nur einem schwarzen Punkt ergibt das im Ursprung gespiegelte Strukturelement S
- Dilation ist assoziativ: $(B \oplus S) \oplus S = B \oplus (S \oplus S)$
 - ▷ kann Mehrfachdilation beschleunigen, da durch einzige Dilation ersetzbar
- Wenn das Strukturelement verbunden ist und den Ursprung (0,0) enthält, dann ist die Dilation des ganzen Bildes äquivalent zur Addition des Bildes zur Dilation der Randpunkte
 - ▷ kann manchmal die Dilation beschleunigen, aber nicht immer:
 - Überprüfung ob ein Punkt Randpixel ist, kostet auch Zeit

4

3.1.2 Morphologie (5)

Morphologische Operationen in Gamera

- `Image.dilate()`, `Image.erode()`
einmalige Operation mit 3x3 Strukturelement
- `Image.erode_dilate(int n,int what,int elem)`
mehrfache Operation mit zwei wählbaren Strukturelementen
- `Image.dilate_with_structure(Image S,Point origin)`
Dilation mit beliebigem Strukturelement;
analog `erode_with_structure`. Beispiel:

```
# erodiere mit 3x3 Kreuz
S = Image(Point(0,0), Point(2,2), ONEBIT)
S.draw_line(Point(0,1), Point(2,1), 1,1)
S.set(Point(1,0),1); S.set(Point(1,2),1)
image = image.erode_with_structure(S, Point(1,1))
```

5

3.1.3 Faltung (2)

Faltung zum Preprocessing

- **Tiefpassfilter**
 - gleicht benachbarte Pixel an, d.h. entfernt starke Fluktuationen ("hohe Frequenzen")
 - einfachster Kern: Averaging Kernel, d.h. Mittelwertbildung über Nachbarschaft
- **Hochpassfilter**
 - entfernt schwache Fluktuationen (niedrige Frequenzen) z.B. verrierende Illumination
 - verstärkt allerdings kleine Flecken (salt & pepper noise)
 - Beispielkern:

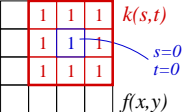
$$\begin{pmatrix} 1 & -2 & 1 \\ -2 & 5 & -2 \\ 1 & -2 & 1 \end{pmatrix}$$

7

3.1.3 Faltung (1)

Die Faltung (engl. *Convolution*)

- gewichteter Mittelwert über Werte der Nachbarpixel
Matrix der Gewichte heißt *Faltungskern*

$$g(x,y) = \sum_{s=-\infty}^{\infty} \sum_{t=-\infty}^{\infty} k(s,t) \cdot f(x-s,y-t)$$


- Achtung: Minuszeichen bewirkt Spiegelung des Faltungskerns bei der Gewichtung (egal bei symmetrischen Kernen)
- wenn Faltungskern normiert $\sum_s \sum_t k(s,t) = 1$
dann ändert die Faltung die mittlere Helligkeit nicht

6

3.1.3 Faltung (3)

Faltung in Gamera

- **vordefinierte Kernel**
 - `gaussian_smoothing(float stddev)` oder `convolve(GaussianKernel(stddev))`
Glättung mit Gaussglocke als Kernel (Tiefpassfilter)
 - `sobel_edge_detection()`
Filter zum Kanten finden
- **beliebige Kernel**
 - `convolve(kernel, border)`
kernel ist zeilenweise Liste der Gewichte
border spezifiziert Behandlung Randpixel, z.B. 3 für "reflect"

```
# Averaging Filter (auf 1 normiert)
image.convolve([[1.0/9, 1.0/9, 1.0/9],
               [1.0/9, 1.0/9, 1.0/9],
               [1.0/9, 1.0/9, 1.0/9]])
```

8