

1. Definitionen, Funktionen, Anwendungen
2. Koordinatensysteme und -transformationen
3. Räumliche Datenmodellierung
4. Vermaschungen
5. Räumliche Interpolation
- 6. Transformationen, Filtermethoden**

Filter Methoden

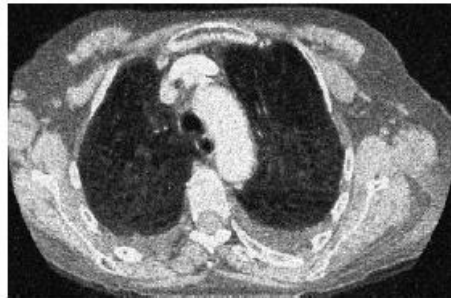
Anstelle eines „wahren“ Rasterbildes liegt typischerweise ein **verraushtes Bild** vor, mit

$$f = f_{\text{true}} + n,$$

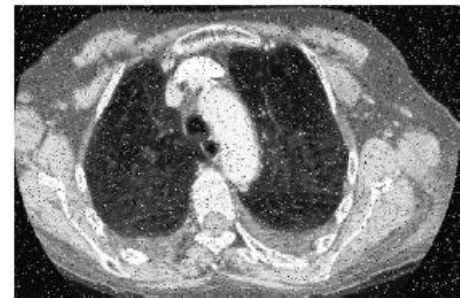
wobei der Fehler n mit verschiedenen Modellen beschrieben werden kann (z.B. Gauß-verteilt, „*gaussian noise*“). **Rauschentfernung / „Denoising“** beschreibt das Problem, eine Schätzung $g \approx f_{\text{true}}$ aus dem Wissen von f zu erhalten.



Original



Original mit „*Gaussian noise*“



Original mit „*Salt and Pepper noise*“

Filter Methoden

Anstelle eines „wahren“ Rasterbildes liegt typischerweise ein **verraushtes Bild** vor, mit

$$f = f_{\text{true}} + n,$$

wobei der Fehler n mit verschiedenen Modellen beschrieben werden kann (z.B. Gauß-verteilt, „*gaussian noise*“). **Rauschentfernung** / „**Denoising**“ beschreibt das Problem, eine Schätzung $g \approx f_{\text{true}}$ aus dem Wissen von f zu erhalten.

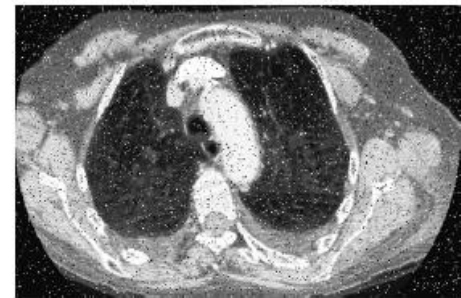
Rauschentfernung ist oft eine initialer Operation für weiter Bildbearbeitungsschritte.



Original



Original mit „*Gaussian noise*“



Original mit „*Salt and Pepper noise*“

Filter Methoden

Konvolution: Gegeben sein ein *Kernel* $w = [0,1]^2 \rightarrow \mathbb{R}$, die Konvolution einer Funktion f mit w ist definiert über

$$g(x, y) = (w * f)(x, y) = \int_{[0,1]^2} w(x - u, y - v) f(u, v) du, dv.$$

Fourier-Transformation: Die Fourier-Transformation einer Funktion f ist definiert über

$$F(u, v) = \hat{f}(u, v) = \int_{[0,1]^2} e^{-2\pi i(x,y) \cdot (u,v)} f(x, y) dx dy.$$

Konvolution und Fourier-Transformation: Die Fourier-Transformation einer Konvolution hat folgende Eigenschaft:

$$G(u, v) = W(u, v)F(u, v)$$

Filter Methoden

Konvolution: Gegeben sein ein *Kernel* $w = [0,1]^2 \rightarrow \mathbb{R}$, die Konvolution einer Funktion f mit w ist definiert über

$$g(x, y) = (w * f)(x, y) = \int_{[0,1]^2} w(x - u, y - v) f(u, v) du, dv.$$

Fourier-Transformation: Die Fourier-Transformation einer Funktion f ist definiert über

$$F(u, v) = \hat{f}(u, v) = \int_{[0,1]^2} e^{-2\pi i(x,y) \cdot (u,v)} f(x, y) dx dy.$$

Konvolution und Fourier-Transformation: Die Fourier-Transformation einer Konvolution hat folgende Eigenschaft:

$$G(u, v) = W(u, v)F(u, v)$$

Die Wahl der Kernelfunktion w oder W , entweder im Orts- oder Spektralbereich, bestimmt die Eigenschaften der Konvolution (Filter).

Filter Methoden

Diskrete Repräsentation / Pixel-Setup:

Konvolution: Gegeben sein ein Kernel $w \in \mathbb{R}^{(2K+1) \times (2L+1)}$, die diskrete Konvolution von f mit w ist definiert über

$$g(m, n) = (w * f)(m, n) = \sum_{k=-K}^K \sum_{l=-L}^L w(k, l) f(m - k, n - l)$$

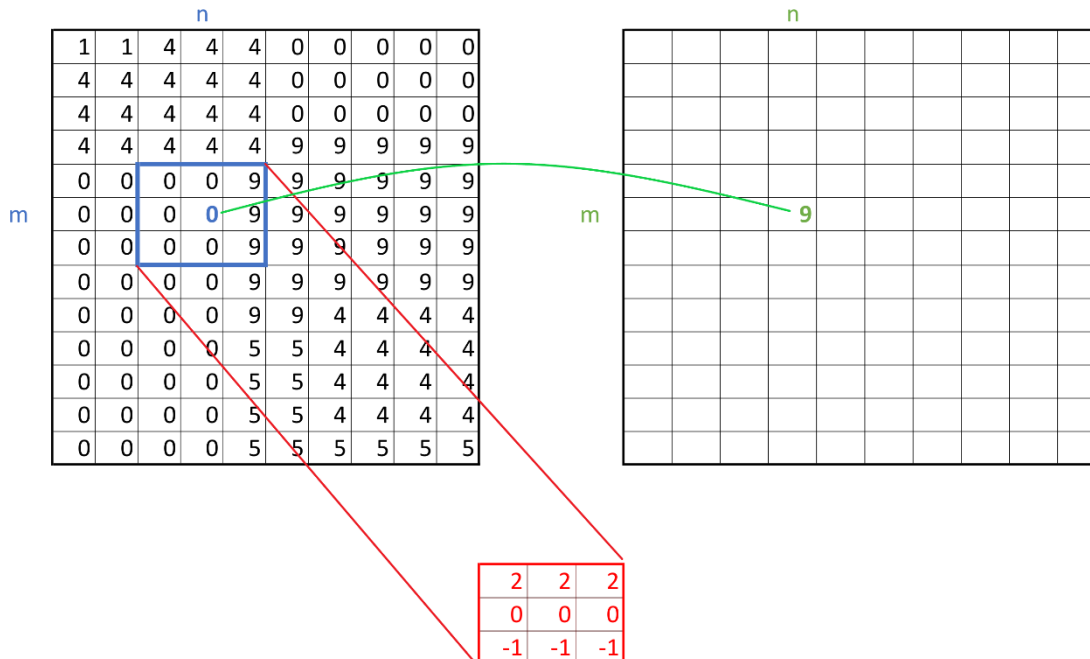
1	1	4	4	4	0	0	0	0	0
4	4	4	4	4	0	0	0	0	0
4	4	4	4	4	0	0	0	0	0
4	4	4	4	4	9	9	9	9	9
0	0	0	0	9	9	9	9	9	9
0	0	0	0	9	9	9	9	9	9
0	0	0	0	9	9	9	9	9	9
0	0	0	0	9	9	9	9	9	9
0	0	0	0	9	9	4	4	4	4
0	0	0	0	5	5	4	4	4	4
0	0	0	0	5	5	4	4	4	4
0	0	0	0	5	5	4	4	4	4
0	0	0	0	5	5	5	5	5	5

Filter Methoden

Diskrete Repräsentation / Pixel-Setup:

Konvolution: Gegeben sein ein Kernel $w \in \mathbb{R}^{(2K+1) \times (2L+1)}$, die diskrete Konvolution von f mit w ist definiert über

$$g(m, n) = (w * f)(m, n) = \sum_{k=-K}^K \sum_{l=-L}^L w(k, l) f(m - k, n - l)$$



Beispiele für diskrete Kernelfunktionen w :

Mittelwertkernel / Averaging Kernel in einer $(2K + 1) \times (2L + 1)$ -Nachbarschaft.

Für $L = K = 1$:

$$w = \frac{1}{9} \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}.$$

Gauß-Kernel mit Gewichtsfunktion $w(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{\sigma^2}}$. Für $L = K = 2$ und $\sigma = 1$:

$$w = \frac{1}{271} \begin{pmatrix} 1 & 4 & 7 & 4 & 1 \\ 4 & 16 & 26 & 16 & 4 \\ 7 & 26 & 41 & 26 & 7 \\ 4 & 16 & 26 & 16 & 4 \\ 1 & 4 & 7 & 4 & 1 \end{pmatrix}.$$

Der Gaußfilter ist ein Spezialfall eines Diffusionsfilters.

0	0.2	0
0.2	0.2	0.2
0	0.2	0

5	3	6	1	5
4	4	7	4	5
6	5	2	1	2
7	6	1	0	2
8	6	2	1	3

[Animation siehe OPAL!](https://bildungsportal.sachsen.de/opal/auth/RepositoryEntry/19758415873/CourseNode/102027240864975) (https://bildungsportal.sachsen.de/opal/auth/RepositoryEntry/19758415873/CourseNode/102027240864975)

Filter Methoden

Original



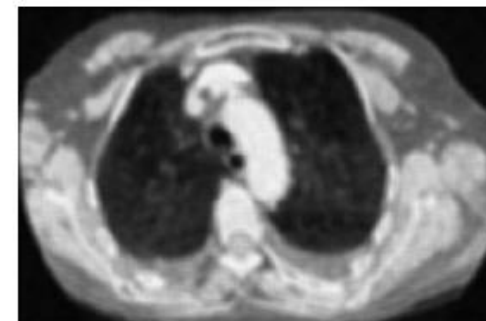
Gauß-verteiltes rauschen



Ergebnis Gaußfilter



Ergebnis Mittelwertfilter



Beispiele für nicht-lineare Filter (basieren nicht auf Konvolution):

Median Filter: Für jeden Pixel (m, n) wird eine lokale Umgebung $LA(m, n) = \{m - K, \dots, m + K\} \times \{n - L, \dots, n + L\}$ definiert und es wird der Median aller Werte in der Umgebung zugeordnet:

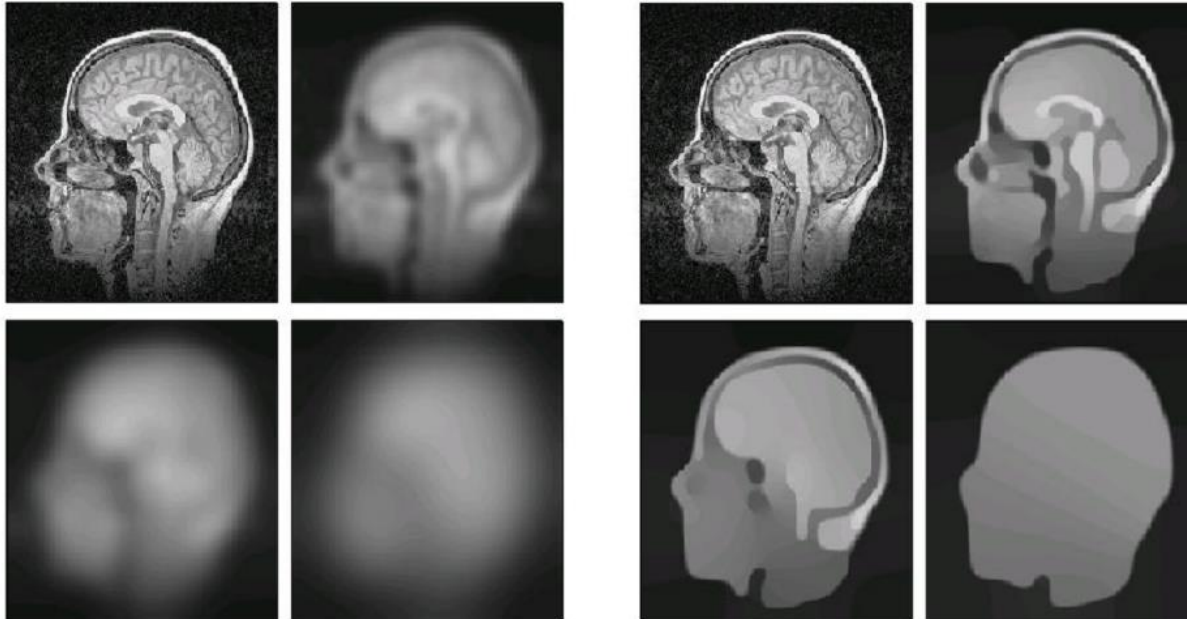
$$g(m, n) = \text{median}\{f(k, l) : (k, l) \in LA(m, n)\}.$$

Nicht-lineare Diffusion: Sei D der Diffusions-Koeffizient, der von g oder ∇g (letztere Ausdruck ist mit Kanten assoziiert) abhängen kann. Dann ist die Diffusionsgleich gegeben mit

$$\begin{aligned} \frac{d}{dt} g(t, x, y) &= \nabla \cdot (D(g, \nabla g)) \nabla g(t, x, y) \\ g(0, x, y) &= f(x, y). \end{aligned}$$

Oft wird $D = D(|\nabla g|) = e^{-\frac{|\nabla g|^2}{2\sigma^2}}$ gewählt (Perona-Malik Diffusion).

Filter Methoden



Scale space behaviour of **linear** Diffusion filtering

Scale space behaviour of **Nonlinear** Diffusion filtering

'Anisotropic Diffusion in Image Processing', J. Weickert, Teubner, 1998

Für welche Anwendung könnte man nicht-lineare Diffusionsfilterung noch verwenden?

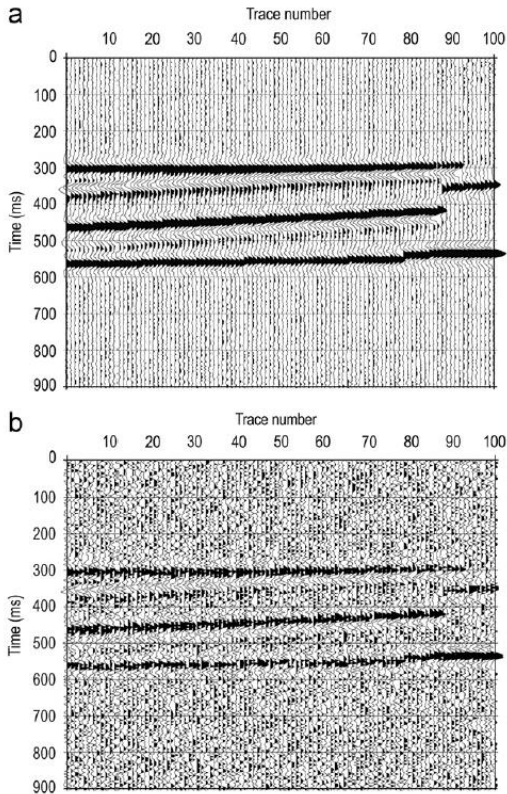


Fig. 3. Synthetic section contaminated with 10% (a) and 50% (b) random noise.

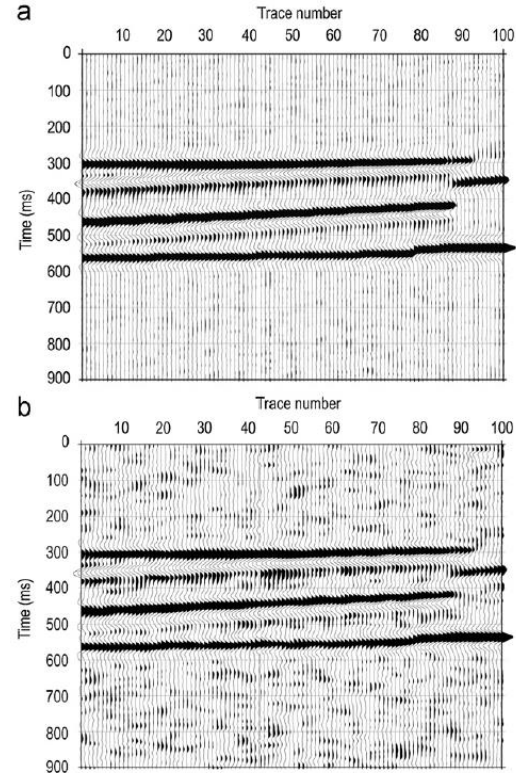
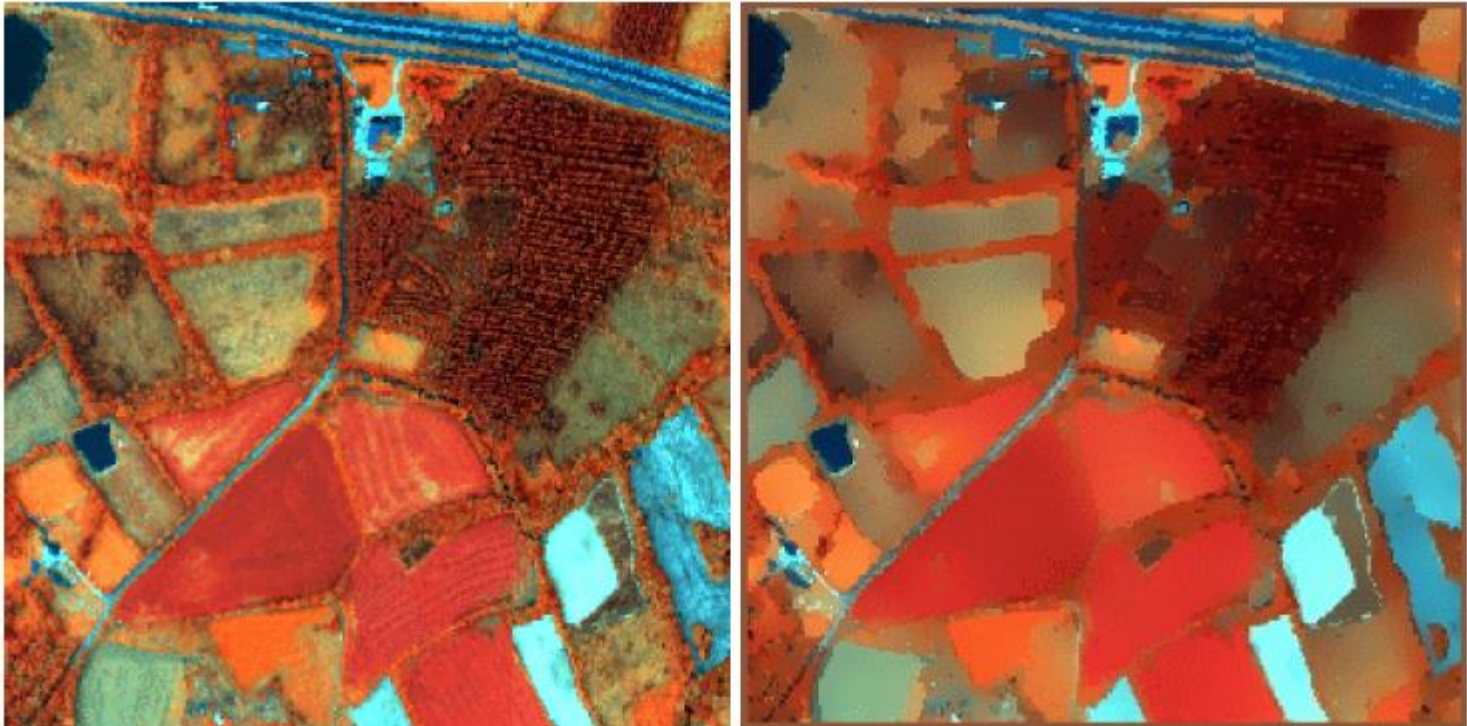


Fig. 4. Anisotropic non-linear diffusion filtering results: (a) section contaminated with 10% random noise and (b) section contaminated with 50% noise.

'Seismic noise attenuation by means of an anisotropic non-linear diffusion filter', Baddari et al., Computers & Geoscience, 2011

Filter Methoden

Für welche Anwendung könnte man nicht-lineare Diffusionsfilterung noch verwenden?



'Nonlinear filtering of hyperspectral images with anisotropic diffusion', Lennon et al., IEEE Int. Geosc. Remote Sensing Symp., 2002

Image Segmentation

„*Image segmentation*“ bezieht sich auf eine große Gruppe von Anwendungen. Dabei kann es um die Unterteilung von Bildern in Vorder- und Hintergrund, die Unterteilung eines Bildes in Regionen ähnlicher Eigenschaften (z.B. Klassifikation) oder die Unterteilung in Regionen anhand sicherer Strukturen (z.B. Kanten).

Zwei Hauptkriterien zur Unterteilung sind:

1. **Diskontinuitäten:** Unterteilung anhand von Kanten
2. **Ähnlichkeit:** Unterteilung von ähnlichen Eigenschaften

Grundsätzlich gibt es keine universelle Unterteilungsmethode für alle Problemstellungen. Manuelle Unterteilung führt zwar häufig zu den „besten“ Ergebnissen, ist aber nur für sehr kleine Datensätze sinnvoll ist.

Kantendetektion / *Edge Detection*

Die Visualisierung / Detektion von kanten in Rasterbildern erfolgt typischerweise basierend auf Gradienten. Häufig werden dafür konvolutionelle Filter verwendet, wie z.B.:

Prewitt-Filter: Die folgenden Filterkernels werden für die Ableitung in x - und y -Richtung gewählt

$$w_x = \begin{pmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix}, \quad w_y = \begin{pmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{pmatrix}$$

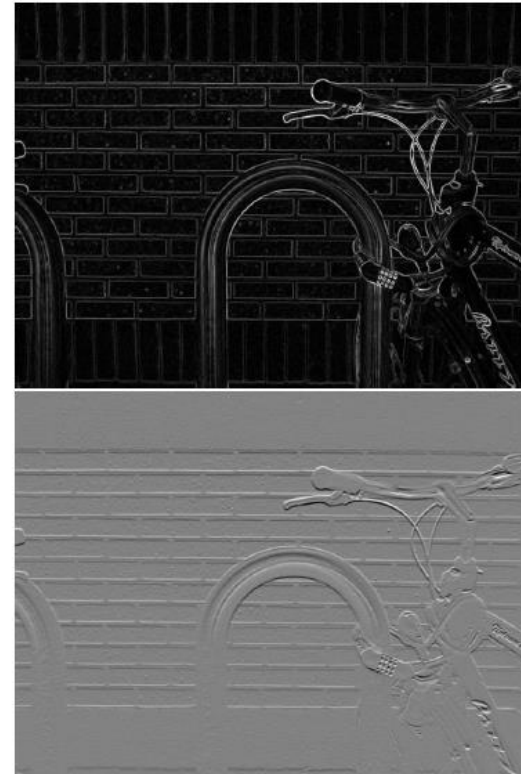
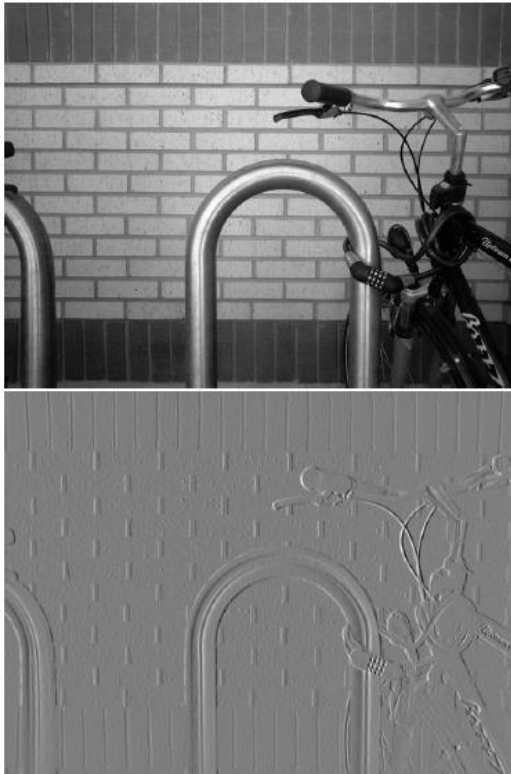
Sobel-Filter: Die folgenden Filterkernels werden für die Ableitung in x - und y -Richtung gewählt

$$w_x = \begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix}, \quad w_y = \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix}$$

Kanten werden durch Pixel mit sehr hohem Gradienten repräsentiert. Alternativ kann man auch mit 2. Ableitung arbeiten. Diese ist genau dort Null, wo der Gradient hoch ist. Ein Verfahren, welches die 2. Ableitung nutzt ist, z.b. *Laplace zero crossing*.

Kantendetektion / *Edge Detection*

Oben links: Originalbild, unten links: Sobel-Filter in x -Richtung, unten rechts: Sobel-Filter in y -Richtung, oben rechts: Sobel-Filter kombiniert in x - und y -Richtung.



Ähnlichkeitssegmentierung über *Thresholding*

„*Thresholding*“ basiert auf der Ähnlichkeit von Pixelwerten eines Grauwert-Bildes. Besonders für Rasterdaten mit **bimodalem Histogramm** erreicht *Thresholding* gute Ergebnisse. Man wählt einen Intensitäts-Schwelldwert (*threshold*) $T > 0$, das segmentierte Bild g ist definiert über

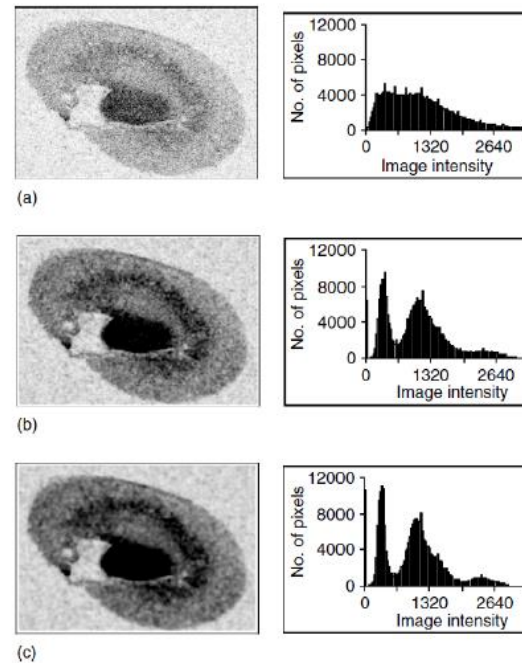
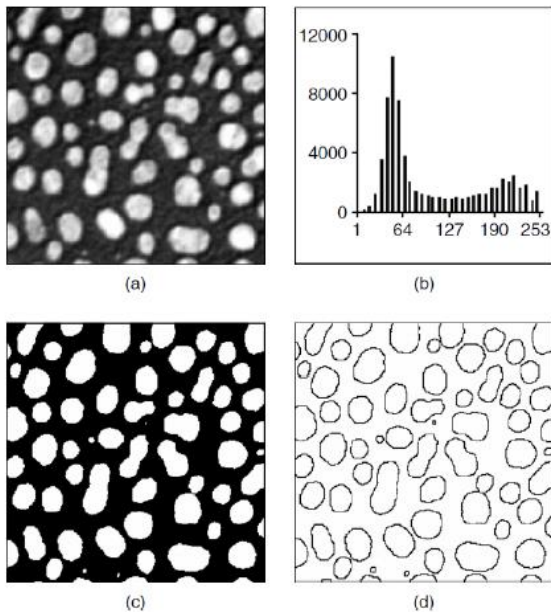
$$g(n, m) = \begin{cases} 1, & \text{if } f(n, m) > T, \\ 0, & \text{if } f(n, m) \leq T. \end{cases}$$

Alle Regionen mit $g(n, m) = 1$ können als „Vordergrund“ und alle anderen als „Hintergrund“ angesehen werden.

Ähnlichkeitssegmentierung über *Thresholding*

Verschiedene Segmentierungsverfahren lassen sich kombinieren.

Block links: (a) Original, (b) zugehöriges Histogramm, (c) Thresholding, (d) Laplace Zero Crossing; **Block rechts:** (a) Original + Histogramm, (b+c) Ergebnis + Histogramm nach Median-Filterung, (c) Filtern kann das Histogramm in ein bimodales Histogramm überführen, wodurch dann Thresholding angewendet werden kann.



Ähnlichkeitssegmentierung über *Thresholding*

Oft ist bei *Thresholding* initial nicht klar, welcher Schwellwert gewählt werden sollte. Eine automatisierte Methodik für dies ist z.B. die **Methode nach Otsu**:

1. Wähle initialen Schwellwert $T = 0$.
2. Unterteile f in Vordergrund g_1 and Hintergrund g_0
3. Berechne die Intra-Klassen-Varianz

$$\sigma(T)^2 = w_0(T)\sigma_0(T)^2 + w_1(T)\sigma_1(T)^2$$

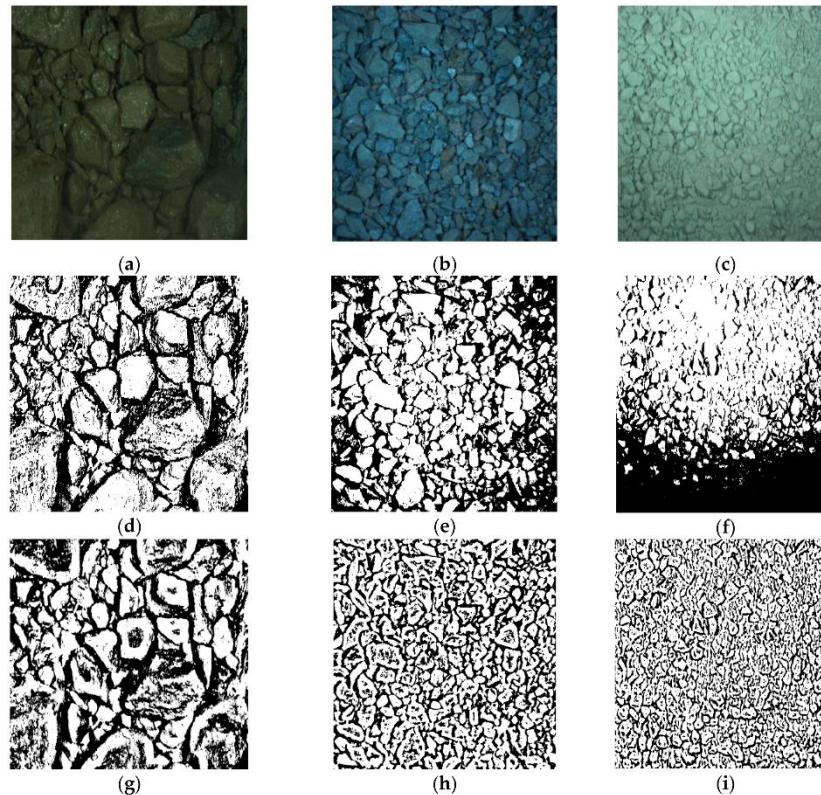
mit w_0 ist der relative Anteil der Hintergrundpixel und w_1 ist der relative Anteil der Vordergrundpixel, $\sigma_0(T)^2$ ist die Varianz des Hintergrunds und $\sigma_1(T)^2$ ist die Varianz des Vordergrunds.

4. Iteriere über alle möglichen Schwellwerte T .
5. Wähle T mit der minimalen Intra-Klassen-Varianz.

Ähnlichkeitssegmentierung über *Thresholding*

Beispiel für Verfahren nach Otsu:

Oben: Original, Mitte: klassisches Verfahren nach Otsu, unten: verbesserte Variante von Otsu's Verfahren

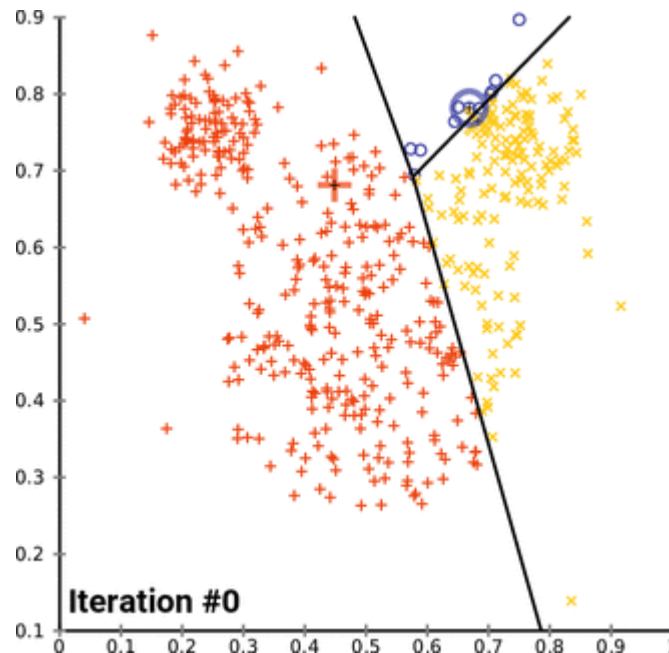


An Improved OTSU Algorithm using histogram accumulation moment for ore segmentation, Zhan and Zhang, Symmetry 11 (2019)

Ähnlichkeitssegmentierung über *kMeans*-Klassifikation

Gegen sein Datenpunkte $\mathbf{x}_i, i = 1, \dots, N$. Klassifikation diesen Datenpunkte basierend auf dem Abstand im Parameterraum zu Klassenzentren $\mu_k, k = 1, \dots, K$ für die Klassen S_k . Es gilt, folgende Zielfunktion zu minimieren:

$$J = \sum_{k=1}^K \sum_{\mathbf{x}_j \in S_k} \|\mathbf{x}_j - \mu_k\|^2$$



Ähnlichkeitssegmentierung über *kMeans-Klassifikation*

Gegen sein Datenpunkte $\mathbf{x}_i, i = 1, \dots, N$. Klassifikation diesen Datenpunkte basierend auf dem Abstand im Parameterraum zu Klassenzentren $\mu_k, k = 1, \dots, K$ für die Klassen S_k . Es gilt, folgende Zielfunktion zu minimieren:

$$J = \sum_{k=1}^K \sum_{\mathbf{x}_j \in S_k} \|\mathbf{x}_j - \mu_k\|^2.$$

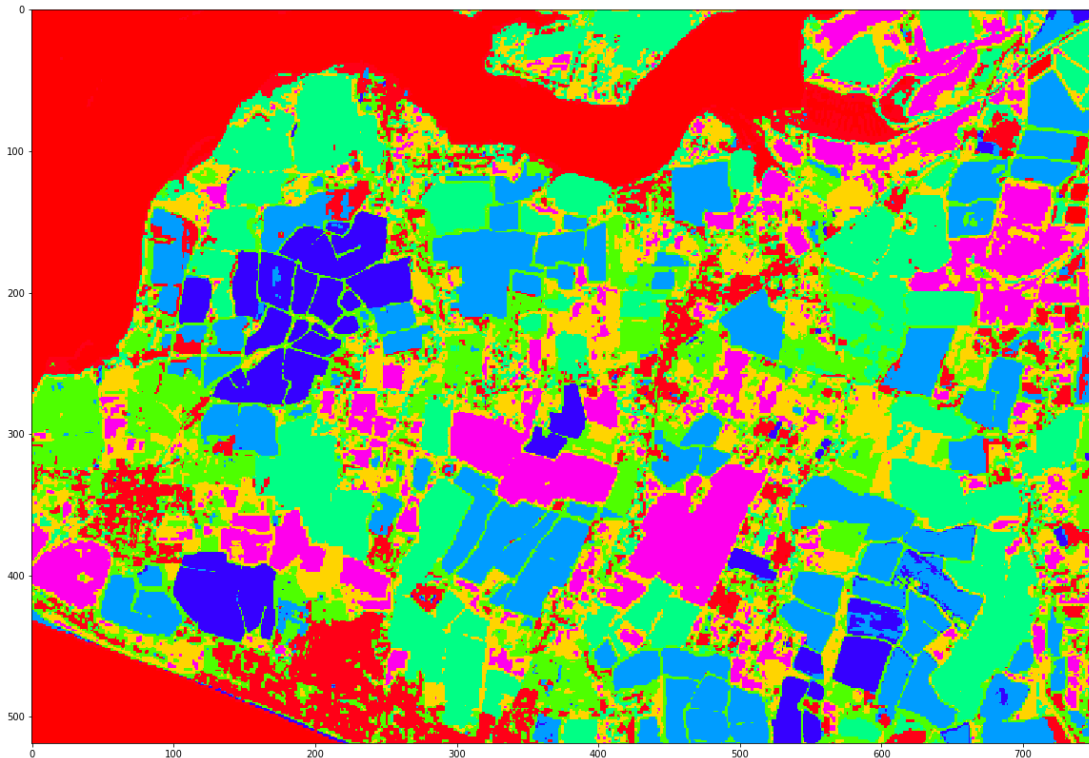
Im Allgemeinen wird versucht, jedem Datenpunkt \mathbf{x}_i der Klasse zuzuordnen, deren Varianz sich durch das Hinzufügen von \mathbf{x}_j am geringsten erhöht.

Allg. iteratives Vorgehen:

1. Initialisiere K Klassen mit ihren Zentren $\mu_k^i, i = 0$.
2. $\forall \mathbf{x}$: Ordne \mathbf{x}_i der Klasse zu, für die $\|\mathbf{x}_i - \mu_k^i\|^2$, minimal ist.
3. $\forall \mu_k^i$: Update die Position von μ_k : $\mu_k^{i+1} = \frac{\sum_{\mathbf{x}_j \in S_k^i} \mathbf{x}_j}{|S_k^i|}$
4. Wiederhole 2 – 4 mit $i = i + 1$ bis Abbruchbedingung erfüllt, z.B. $\mu_k^i = \mu_k^{i+1}, \forall \mu_k^i$.

Ähnlichkeitssegmentierung über *kMeans*-Klassifikation

1. Nicht-überwachtes Klassifikationsverfahren (un-supervised).
2. Anzahl der Klassen steht fest.
3. Optimale Wahl der Metrik für $\|\mathbf{x}_i - \mu_k^i\|^2$ und der Abbruchbedingung für Ergebnis entscheidend, abhängig vom Problemstellung.
4. Sehr einfache Implementierung.



<https://www.acgeospatial.co.uk/k-means-sentinel-2-python/>

Morphologische Operationen

Morphologische Operationen sind auf den ersten Blick mit Filtern verwandt, aber eher mengentheoretisch orientiert. Sie dienen u.a. dazu geometrische Informationen in Bildern zu analysieren und transformieren. Sie können mit der Anwendung von Filtern kombiniert werden.

Die Grundoperationen sind **Erosion** und **Dilatation**. Wir betrachten Sie hier nur für binäre Bilder (mit Bild meinen wir hier eine Funktion $f : \{1, \dots, N\} \times \{1, \dots, M\} \rightarrow \{0, 1\}$).

Morphologische Operationen

Morphologische Operationen sind auf den ersten Blick mit Filtern verwandt, aber eher mengentheoretisch orientiert. Sie dienen u.a. dazu geometrische Informationen in Bildern zu analysieren und transformieren. Sie können mit der Anwendung von Filtern kombiniert werden.

Die Grundoperationen sind **Erosion** und **Dilatation**. Wir betrachten Sie hier nur für binäre Bilder (mit Bild meinen wir hier eine Funktion $f : \{1, \dots, N\} \times \{1, \dots, M\} \rightarrow \{0, 1\}$).

Die Translation einer Menge $X \subset \{1, \dots, N\} \times \{1, \dots, M\}$ ist definiert durch

$$X_q = \{p \in \{1, \dots, N\} \times \{1, \dots, M\} : p - q \in X\}$$

Ein translatiertes Bild ist demnach gegeben durch $f_q(p) = f(p - q)$.
Als Punktinversion bezeichnen wir

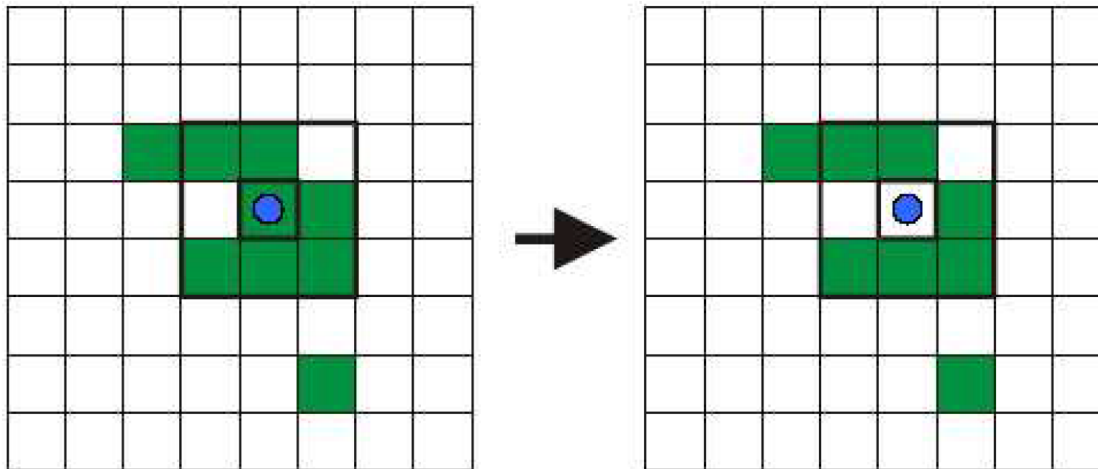
$$\check{X} = \{-p : p \in X\}$$

Morphologische Operationen

Die **Erosion** einer Menge X mit einem Strukturelement B ist nun definiert als

$$\varepsilon_B(X) = \{p \in \{1, \dots, N\} \times \{1, \dots, M\} : B_p \subset X\} = \bigcap_{p \in B} X_{-p}.$$

In Worten: Erosion beschreibt alle Pixel p , für die das Strukturelement B mit Zentrum p in X liegt.

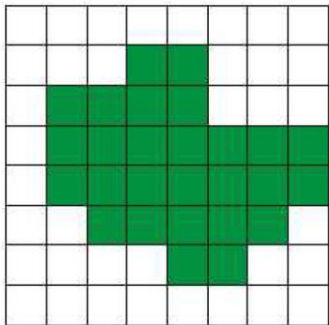


Morphologische Operationen

Die **Erosion** einer Menge X mit einem Strukturelement B ist nun definiert als

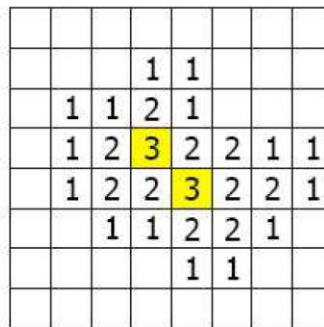
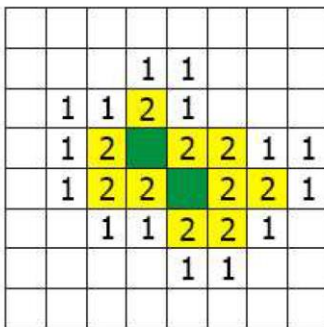
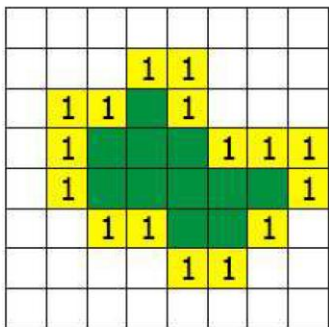
$$\varepsilon_B(X) = \{p \in \{1, \dots, N\} \times \{1, \dots, M\} : B_p \subset X\} = \bigcap_{p \in B} X_{-p}$$

In Worten: Erosion beschreibt alle Pixel p , für die das Strukturelement B mit Zentrum p in X liegt.

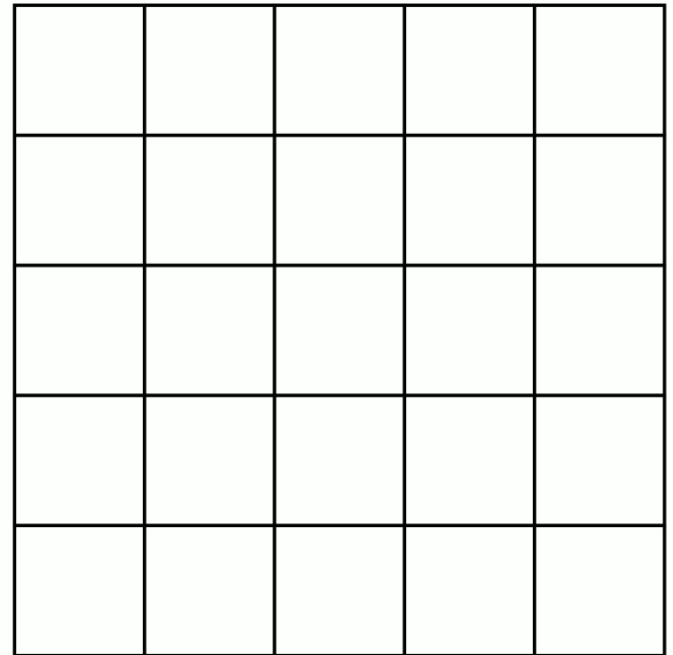
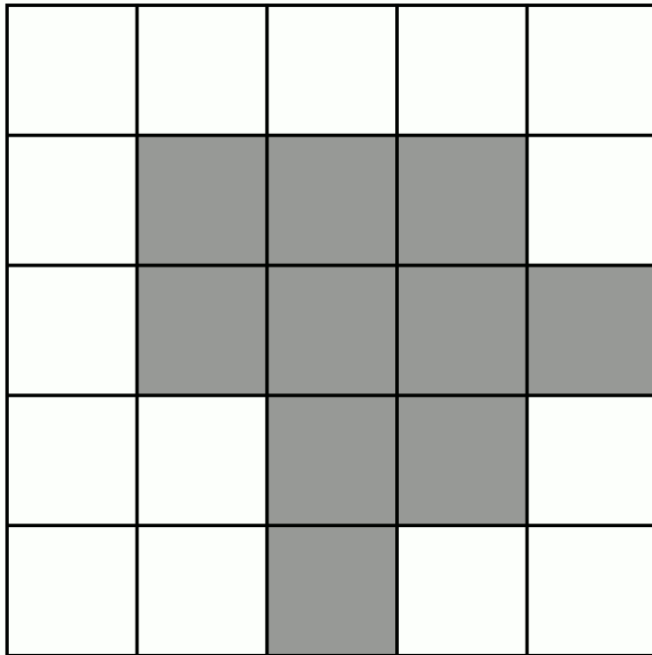
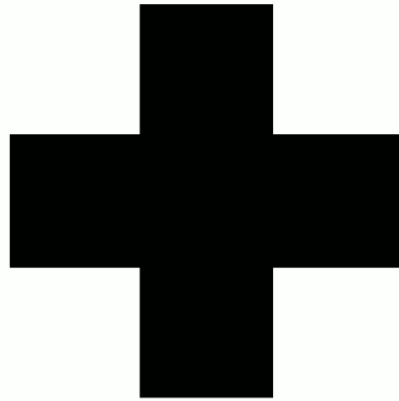


Originalbild

- Objektinneres (nach fortgesetzter Erosion)
- Randpixel nach der n-ten Erosion einschließlich Distanz



Welches Strukturelement wurde hier gewählt?



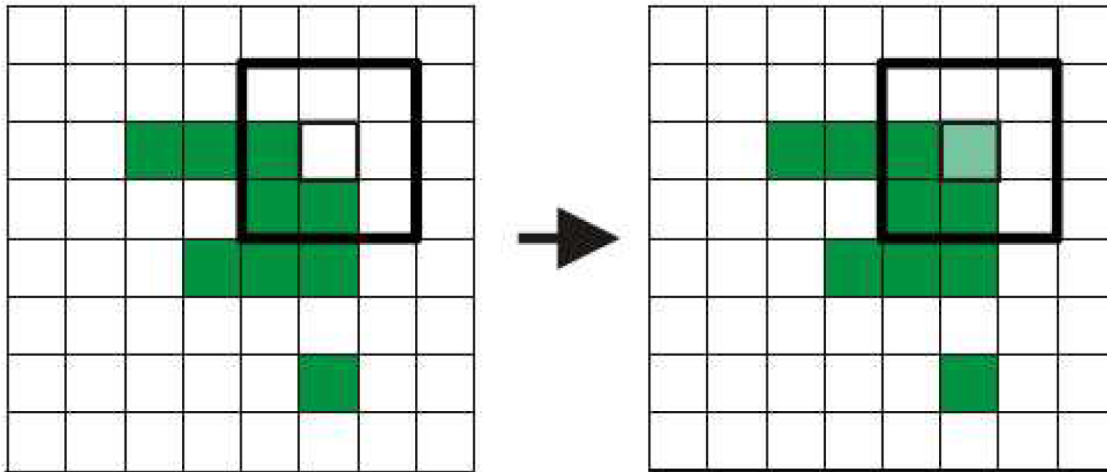
[Animation siehe OPAL!](https://bildungsportal.sachsen.de/opal/auth/RepositoryEntry/19758415873/CourseNode/102027240864975) (https://bildungsportal.sachsen.de/opal/auth/RepositoryEntry/19758415873/CourseNode/102027240864975)

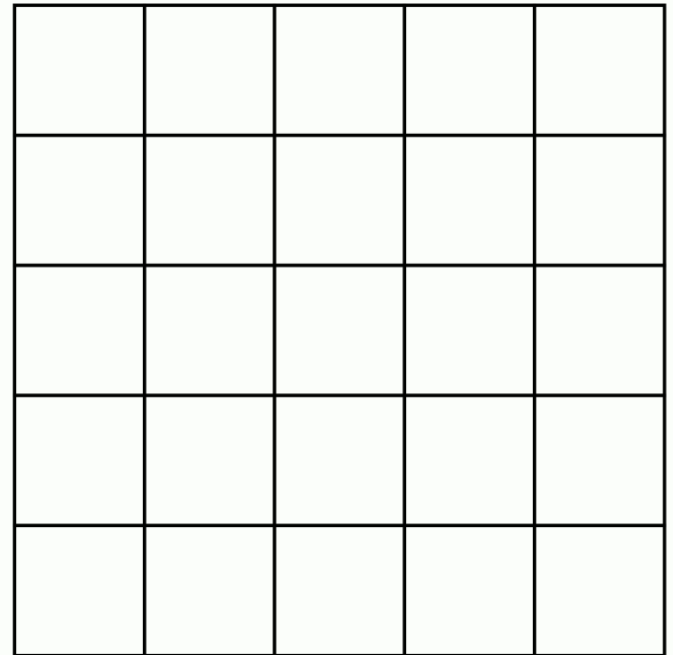
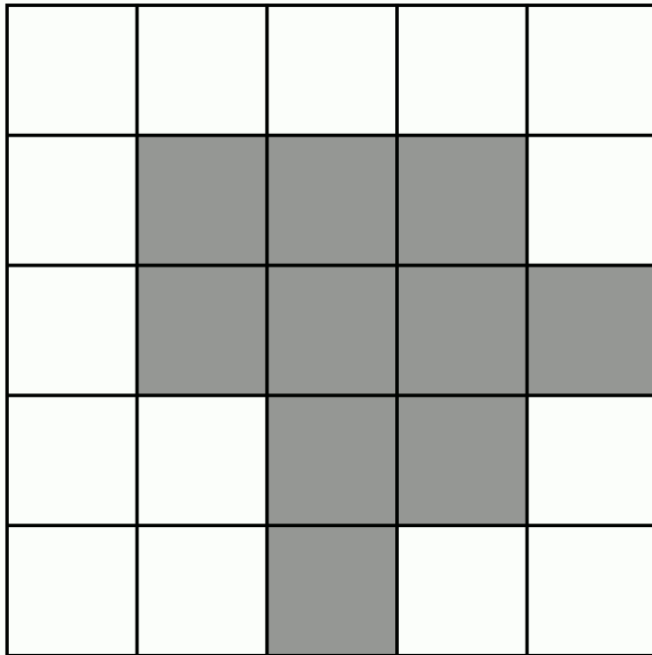
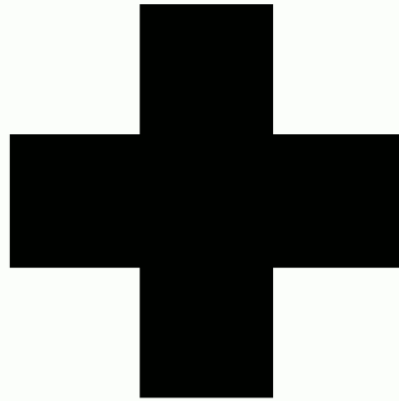
Morphologische Operationen

Die **Dilatation** einer Menge X mit einem Strukturelement B ist nun definiert als

$$\delta_B(X) = \{p \in \{1, \dots, N\} \times \{1, \dots, M\} : B_p \cap X \neq \emptyset\} = \bigcup_{p \in B} X_{-p}.$$

In Worten: Dilatation beschreibt alle Pixel p , für die das Strukturelement B mit Zentrum p mindestens ein Element der Menge X enthält.





[Animation siehe OPAL!](https://bildungsportal.sachsen.de/opal/auth/RepositoryEntry/19758415873/CourseNode/102027240864975) (https://bildungsportal.sachsen.de/opal/auth/RepositoryEntry/19758415873/CourseNode/102027240864975)

Morphologische Operationen

Die **Öffnung** einer Menge X mit einem Strukturelement B ist definiert als

$$\gamma_B(X) = \{B_p : p \in \{1, \dots, N\} \times \{1, \dots, M\}, B_p \subset X\} = \delta_{\check{B}}(\varepsilon_B(X)).$$

In Worten: Öffnung beschreibt die Vereinigung aller Strukturelemente B_p , die in X enthalten sind, nicht nur der Referenzpunkte p (letzteres ist Erosion).

Morphologische Operationen

Die **Öffnung** einer Menge X mit einem Strukturelement B ist definiert als

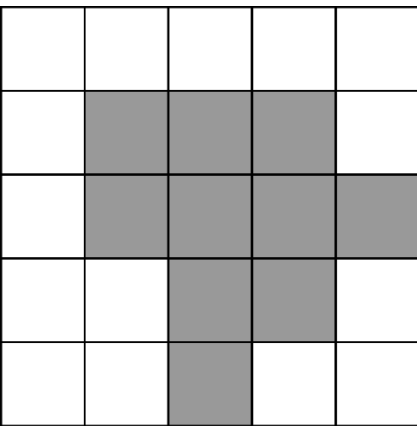
$$\gamma_B(X) = \{B_p : p \in \{1, \dots, N\} \times \{1, \dots, M\}, B_p \subset X\} = \delta_{\check{B}}(\varepsilon_B(X)).$$

In Worten: Öffnung beschreibt die Vereinigung aller Strukturelemente B_p , die in X enthalten sind, nicht nur der Referenzpunkte p (letzteres ist Erosion).

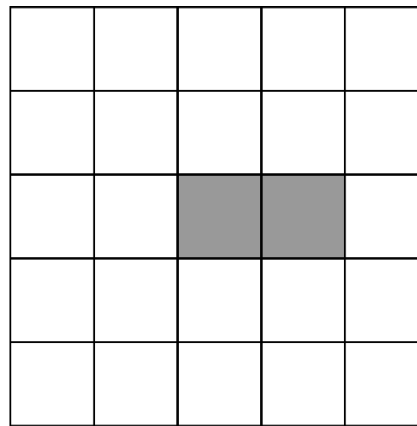
Die **Schließung** einer Menge X mit einem Strukturelement B ist definiert als

$$\begin{aligned} \phi_B(X) &= (\{B_p : p \in \{1, \dots, N\} \times \{1, \dots, M\}, B_p \subset X^c\})^c \\ &= \bigcap_{p \in \{1, \dots, N\} \times \{1, \dots, M\}} \{B_p^c : X \in B_p^c\} = \varepsilon_{\check{B}}(\delta_B(X)). \end{aligned}$$

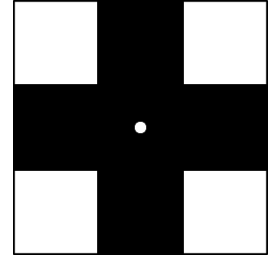
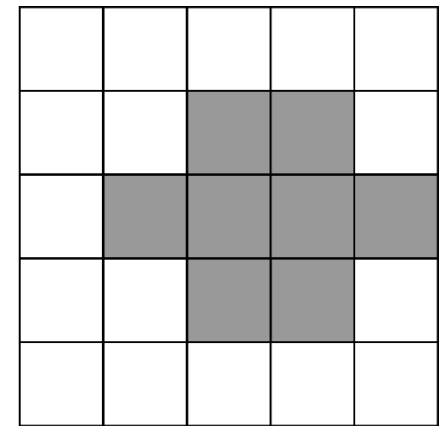
Öffnung / opening



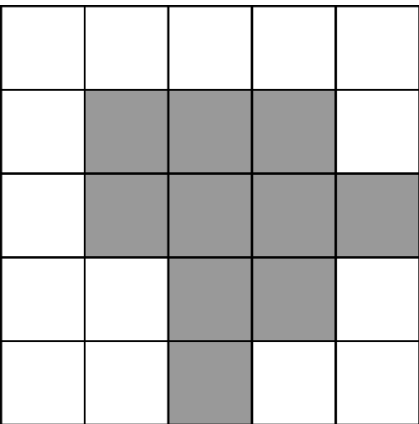
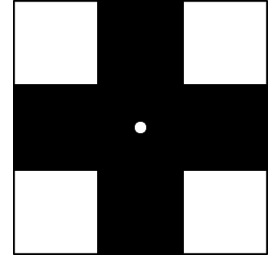
Erosion



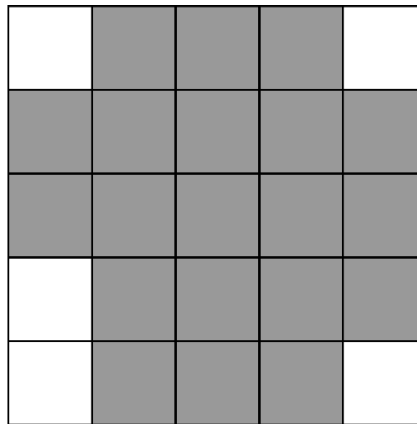
Dilation



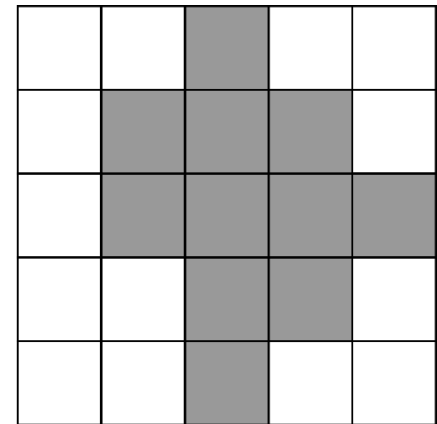
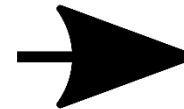
Schließung / Abschluss / closure



Dilation



Erosion



Morphologische Operationen

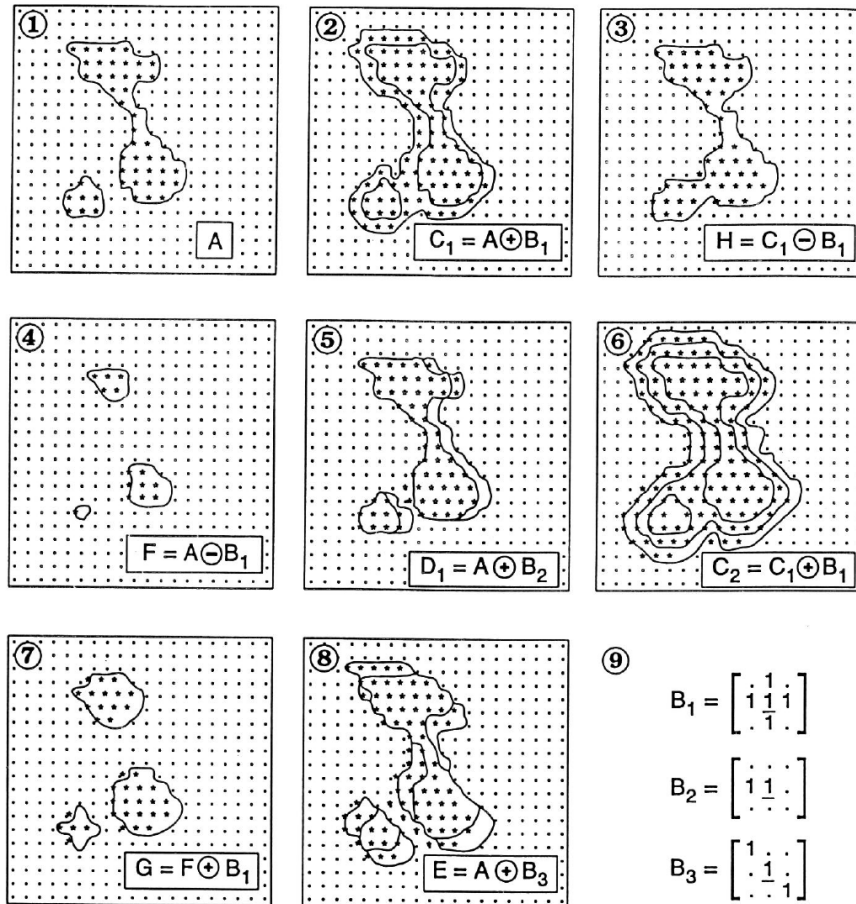


FIG. 6-9. Mathematical morphology operations on a small raster image. (1). Original image A. (2). Dilation of A using structuring element B_1 . (3). Closing of image A by B_1 . (4). Erosion of A by B_1 . (5). Dilation of A by structuring element B_2 . (6). Two successive dilations of A by B_1 . (7). Opening of A by B_1 . (8). Dilation of A by B_3 . (9). Three (3*3) structuring elements.

Arithmetische Differenzen

$$\delta_B(f) - \varepsilon_B(f)$$

$$\delta_B(f) - f$$

$$f - \varepsilon_B(f)$$

können genutzt werden, um Ränder in Bildern hervorzuheben (ähnlich wie Sobel- oder Prewittfilter).

Arithmetische Differenzen

$$\delta_B(f) - \varepsilon_B(f)$$

$$\delta_B(f) - f$$

$$f - \varepsilon_B(f)$$

können genutzt werden, um Ränder in Bildern hervorzuheben (ähnlich wie Sobel- oder Prewittfilter).

Öffnungen und Schliessungen können genutzt werden, um bestimmte geometrische Objekte in Bildern herauszufiltern oder zu erzwingen.

Morphologische Operationen

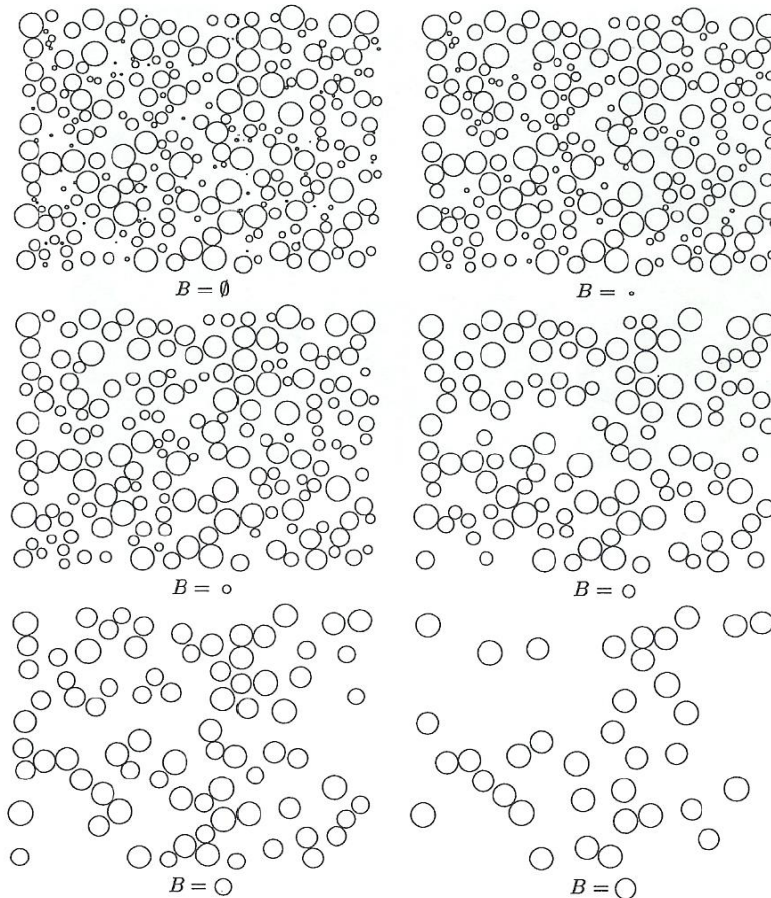


Figure 2.3 'Sieving' of disks of uniformly distributed diameter. The structuring element B is a disk of increasing diameter. If the diameter of the disk is smaller than the diameter of the current structuring element, this disk is eliminated from the image. For $B = \emptyset$ we have the original image.

Morphologische Operationen

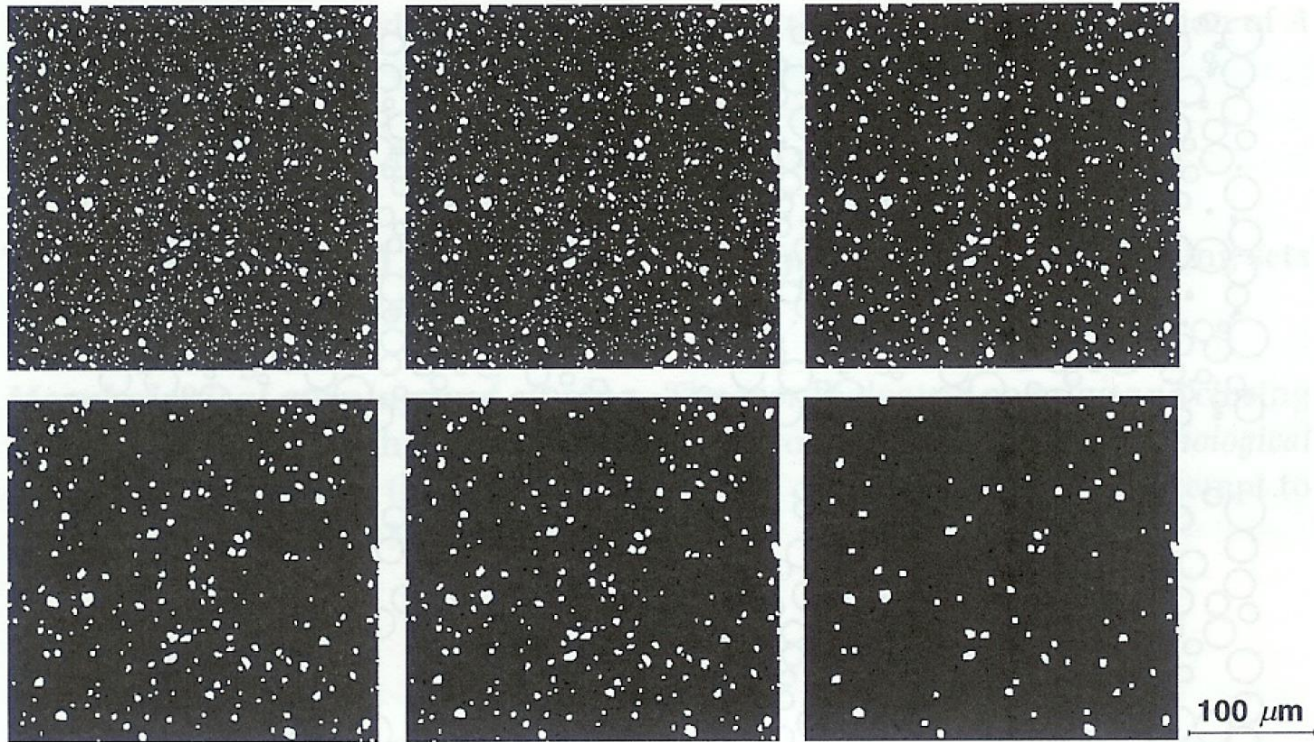
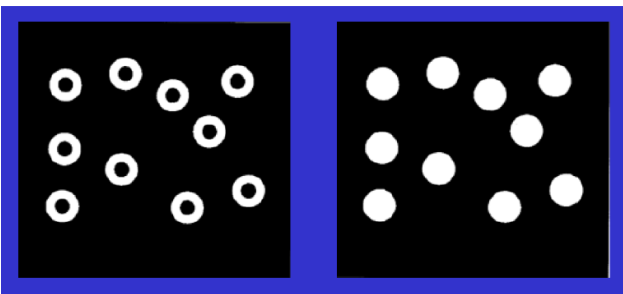
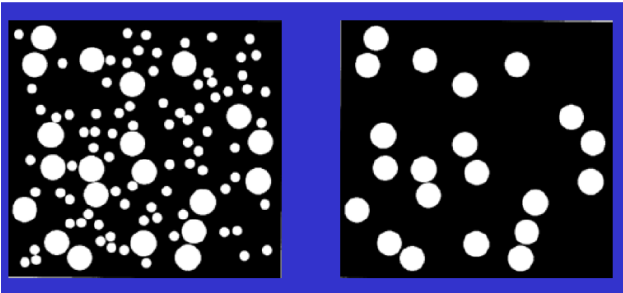
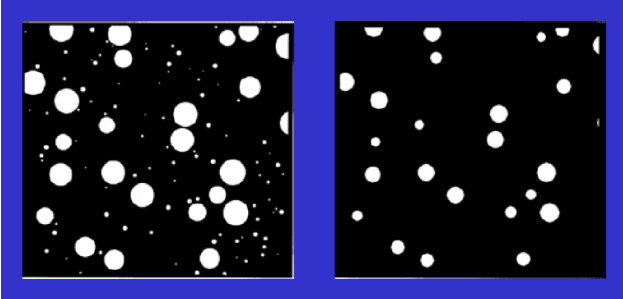


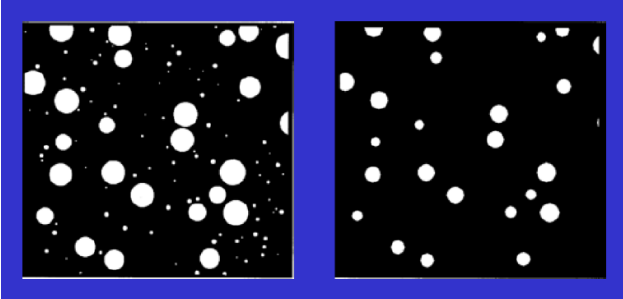
Figure 2.4 The use of morphological opening for the classification of the particle sizes of a spheroidized pearlite containing small granular carbide particles. The structuring element is a square of increasing edge length. This example shows that the shape of the objects is changed during the opening process.

Morphologische Operationen

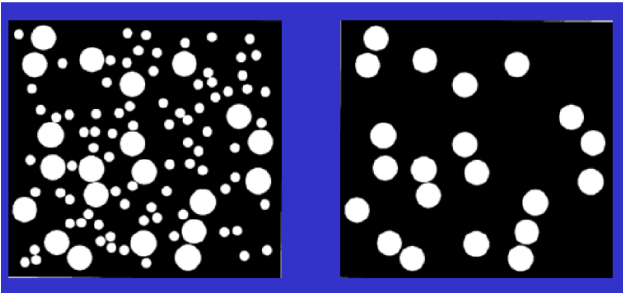


Welche Operationen wurden durchgeführt?

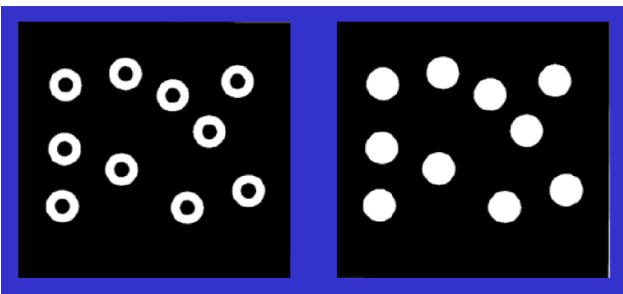
Morphologische Operationen



Erosion

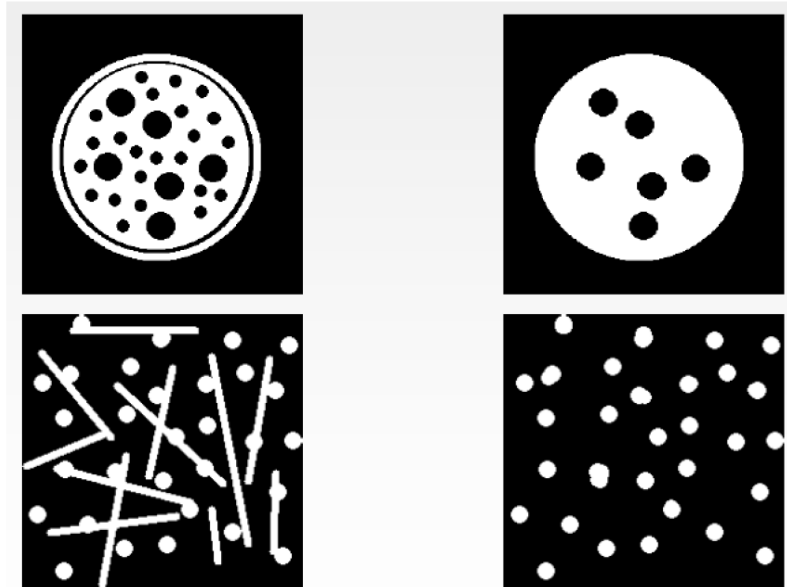


Öffnung



Schliessung

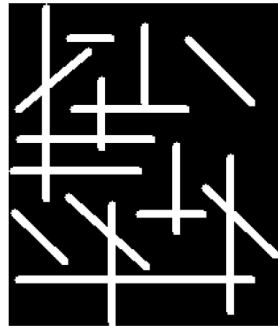
Morphologische Operationen



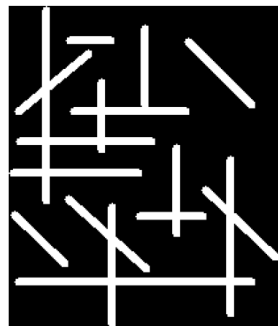
top: Closing with a disc shaped structuring element

bottom: Opening with a disc shaped structuring element

Morphologische Operationen



Opening with a 9×3 vertically oriented structuring element.



Opening with a 3×9 horizontally oriented structuring element.