



TECHNISCHE UNIVERSITÄT
BERGAKADEMIE FREIBERG

Die Ressourcenuniversität. Seit 1765.

Grundlagen GIS

Sommersemester 2024

Prof. Christian Gerhards
Arbeitsgruppe Geomathematik und Geoinformatik

OPAL:

<https://bildungsportal.sachsen.de/opal/auth/RepositoryEntry/19758415873?6>

Institut für Geophysik und Geoinformatik - TU Bergakademie Freiberg

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:

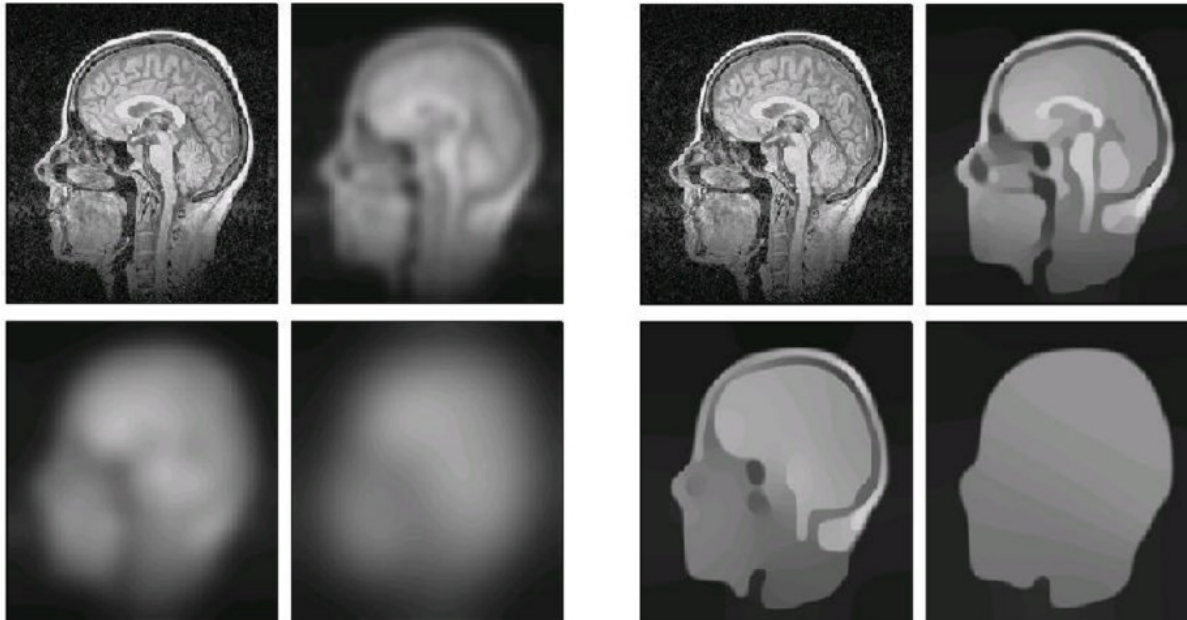
$$g(m, n) = \text{median}\{f(k, l) : (k, l) \in LA(m, n)\}.$$

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?

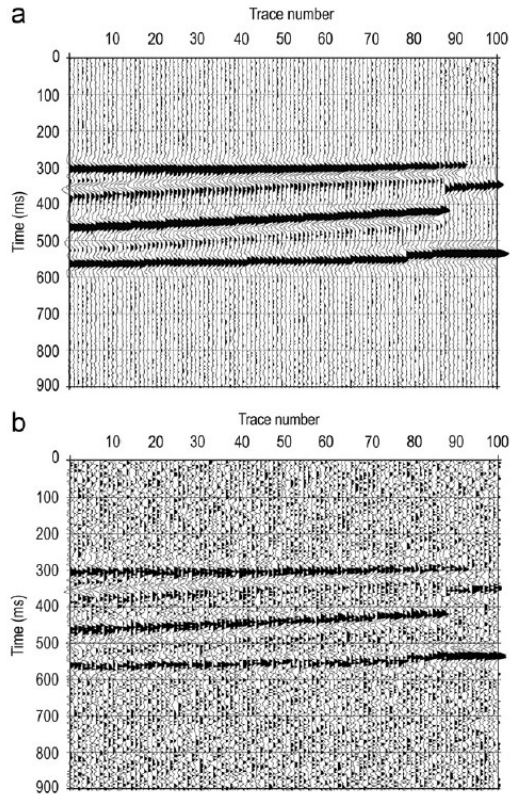


Fig. 3. Synthetic section contaminated with 10% (a) and 50% (b) random noise.

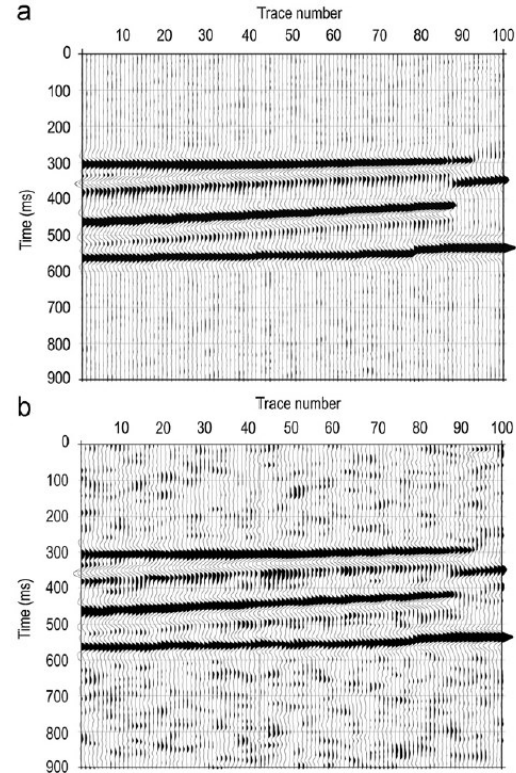
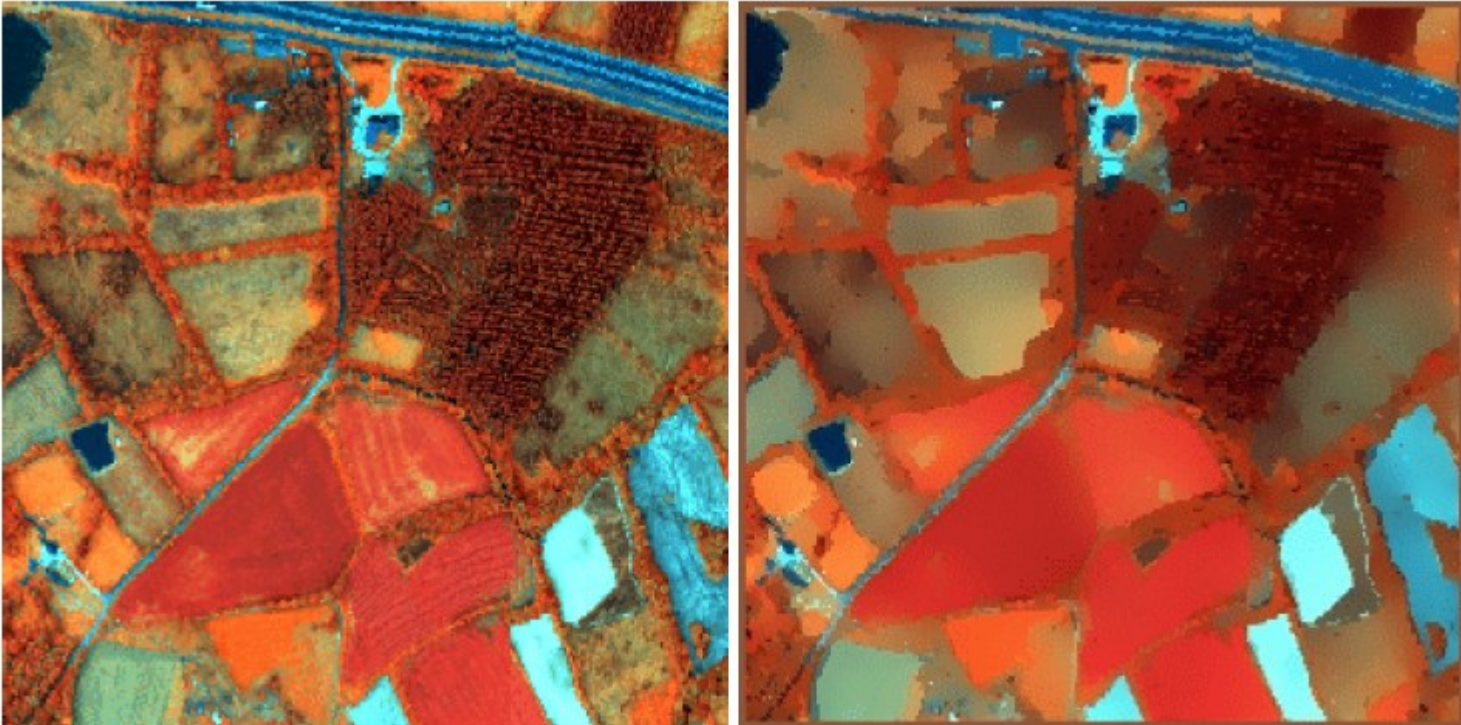


Fig. 4. Anisotropic non-linear diffusion filtering results: (a) section contaminated with 10% random noise and (b) section contaminated with 50% noise.

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

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

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

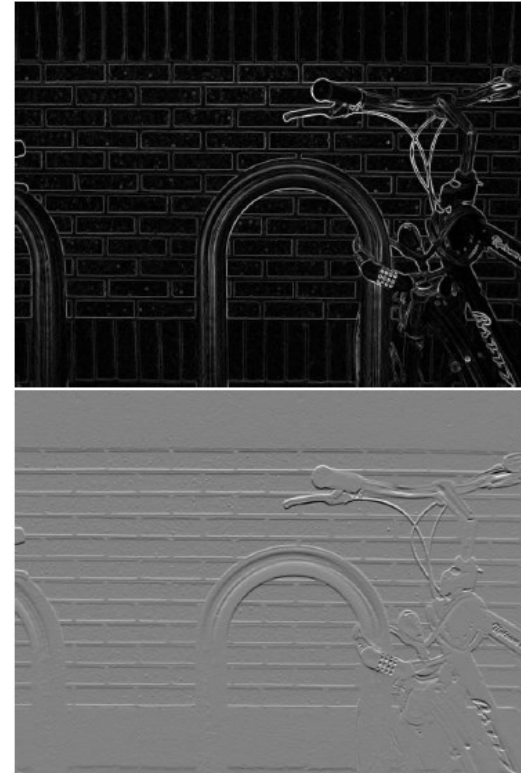
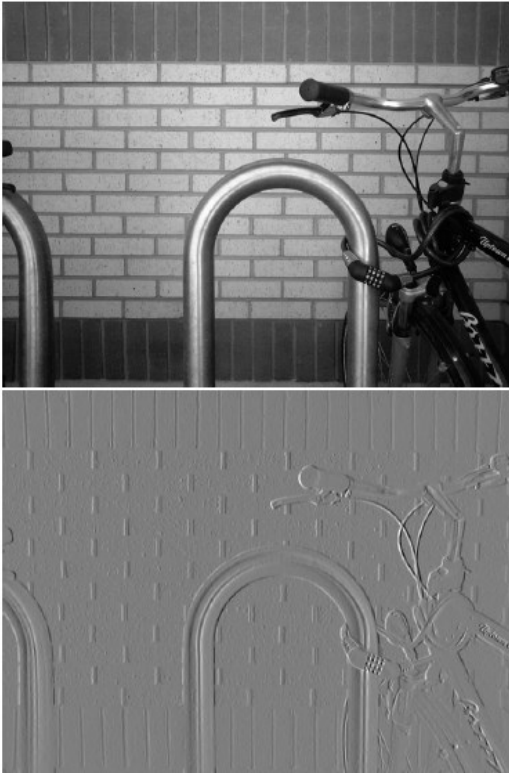
Sobel-Filter: Die folgenden Filterkernels werden für die Ableitung in x - und y -Richtung gewählt

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

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-Schwellwert (*threshold*) $T > 0$, das segmentierte Bild g ist definiert über

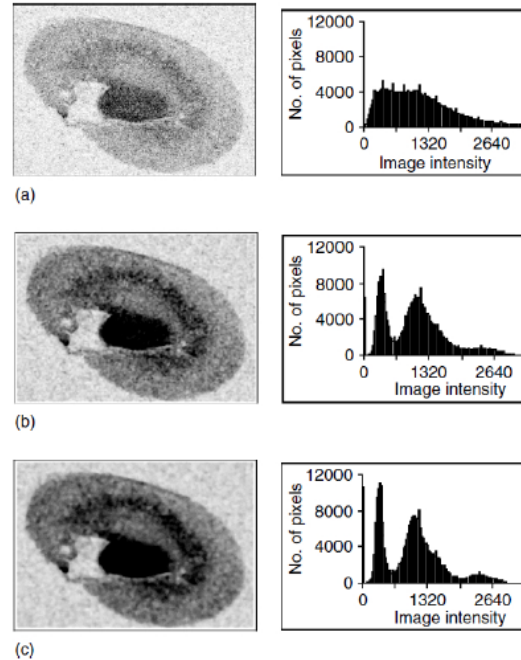
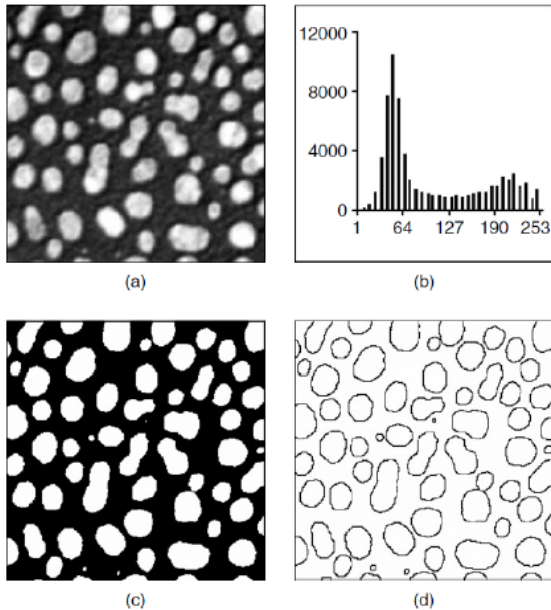
$$g(n, m) = \begin{cases} 1, & \text{if } f(n, m) > T, \\ 0, & \text{if } f(n, m) \leq T. \end{cases}$$

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

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

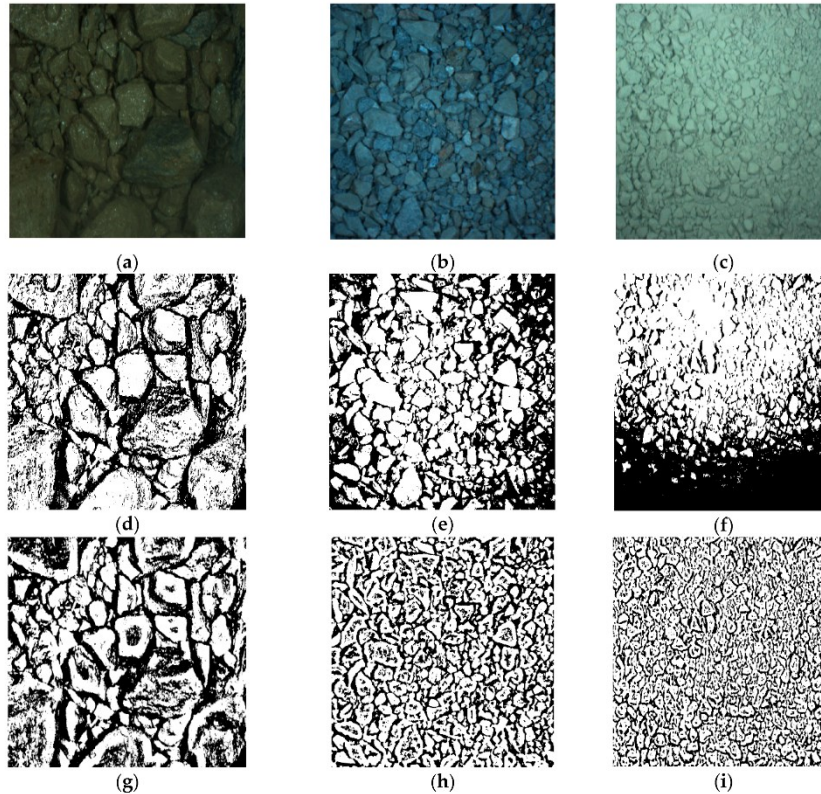
mit w_0 ist der relative Anteil der Hintergrundpixel und w_1 ist der relative Anteil der Vordergrundpixel, $\sigma_0(T)^2$ ist die Varianz des Hintergrunds und $\sigma_1(T)^2$ ist die Varianz des Vordergrunds.

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

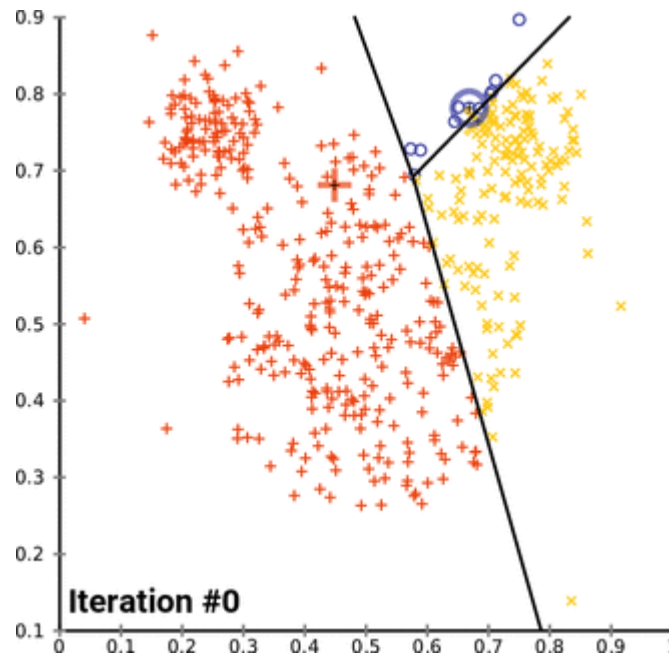


An Improved OTSU Algorithm using histogram accumulation moment for ore segmentation, Zhan and Zhang, Symmetry 11 (2019)

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:

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

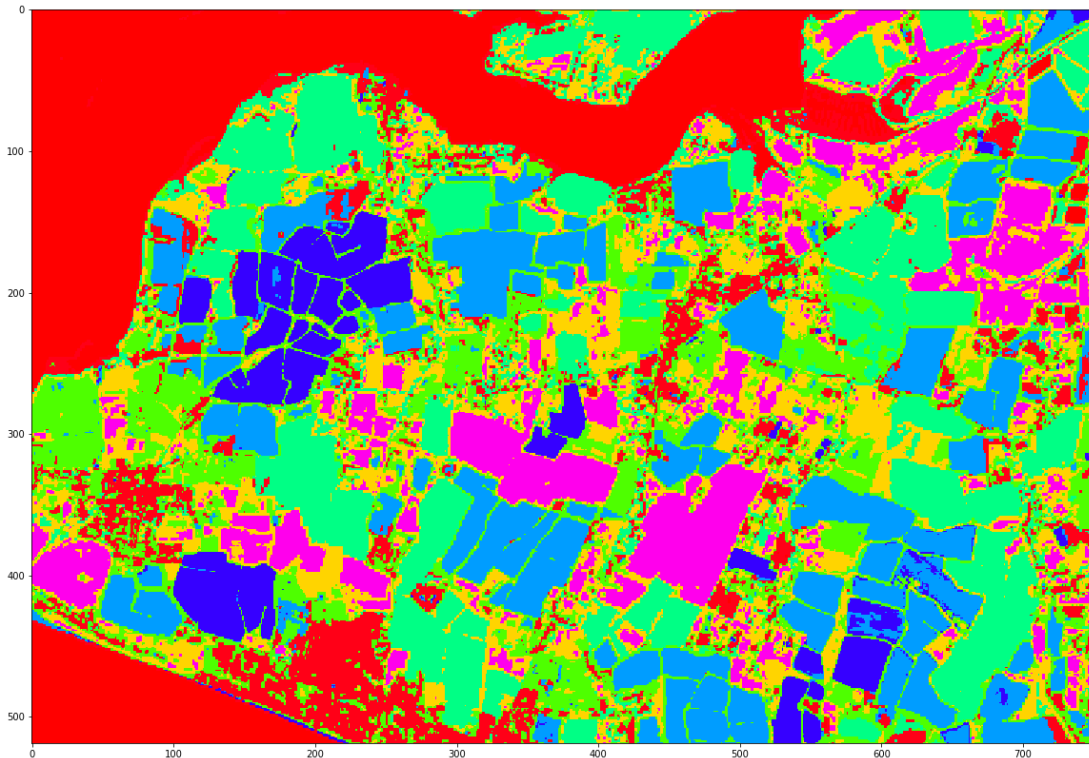
Im Allgemeinen wird versucht, jedem Datenpunkt \mathbf{x}_i der Klasse zuzuordnen, deren Varianz sich durch das Hinzufügen von \mathbf{x}_j am geringsten erhöht.

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/>