

Labor Matlab für die industrielle und medizinische Bildverarbeitung

Prof. Dr.-Ing. Bodo Rosenhahn

Institut für Informationsverarbeitung



Einleitung

- 19.10. Introduction (1h VL, 3 L), Accountvergabe (Präsenz)
- 26.10. Local operators (Harris, etc.) (1h VL, 3L)
- 02.11. Global Operators (Hough Transform) (1h VL, 3L)
- 09.11. Region Growing / Watershed Segmentation (1h VL, 3L)
- 16.11. Bayes Classifier (1h VL, 3L)
- 23.11. K-Means / Mean shift (1h VL, 3L)
- 30.11. Shape Context (1h VL, 3L)
- 07.12. Morphological Operators (1h VL, 3L)
- 14.12. Disparity estimation (DTW) (1h VL, 3L)
- 21.12. Restarbeiten vor Weihnachten (4L)
- 11.01. Calibration and Triangulation (1h VL, 3L)
- 18.01. PCA (1h VL, 3L)
- 25.01. Tracking (1h VL, 3L)

Klassen an Segmentierungsverfahren

- Pixelorientierte Verfahren
- Kantenorientierte Verfahren
- Regionengetrieben Verfahren

Pixelorientierte Verfahren

Pixelorientierte Verfahren treffen für jeden einzelnen Bildpunkt die Entscheidung, ob er zu einem bestimmten Segment gehört oder nicht. Diese Entscheidung kann – muss aber nicht – durch die Umgebung beeinflusst sein. Punktorientierte Verfahren sind meist einfach zu berechnen, liefern per se aber erstmal keine zusammenhängenden Segmente.

Beispiel: Schwellwertverfahren

$$u(x, y) = \begin{cases} 255 & I(x, y) > \theta \\ 0 & I(x, y) \leq \theta \end{cases}$$

Pixelorientierte Verfahren

Eine Verallgemeinerung der Pixelorientierten Verfahren stellen Klassifikatoren dar.

Einfaches Beispiel: K-means Clustering

Ziel: Optimierte:

$$J = \sum_{n=1}^N \sum_{k=1}^K \|x_n - \mu_k\|^2$$

K - means Clustering

Beim k-means Algorithmus wird die Anzahl k von Clustern vor dem Start festgelegt. Eine Funktion zur Bestimmung des Zentrums eines Clusters muss bekannt sein. Algorithmus:

- **Initialisierung:** (zufällige) Auswahl von k Clusterzentren $\mathbf{m}_1^{(1)}, \dots, \mathbf{m}_k^{(1)}$
- **Zuordnung:** Jedes Objekt wird dem ihm am nächsten liegenden Clusterzentrum zugeordnet

$$S_i^{(t)} = \left\{ x_j : \|x_j - m_i^{(t)}\|^2 \leq \|x_j - m_{i^*}^{(t)}\|^2 \text{ für alle } i^* = 1, \dots, k \right\}$$

- **Neuberechnung:** Es werden für jeden Cluster die Clusterzentren neu berechnet.

$$m_i^{(t+1)} = \frac{1}{|S_i^{(t)}|} \sum_{x_j \in S_i^{(t)}} x_j$$

- **Wiederholung:** Falls sich nun die Zuordnung der Objekte ändert, weiter mit Schritt 2, ansonsten Abbruch

Beispiel



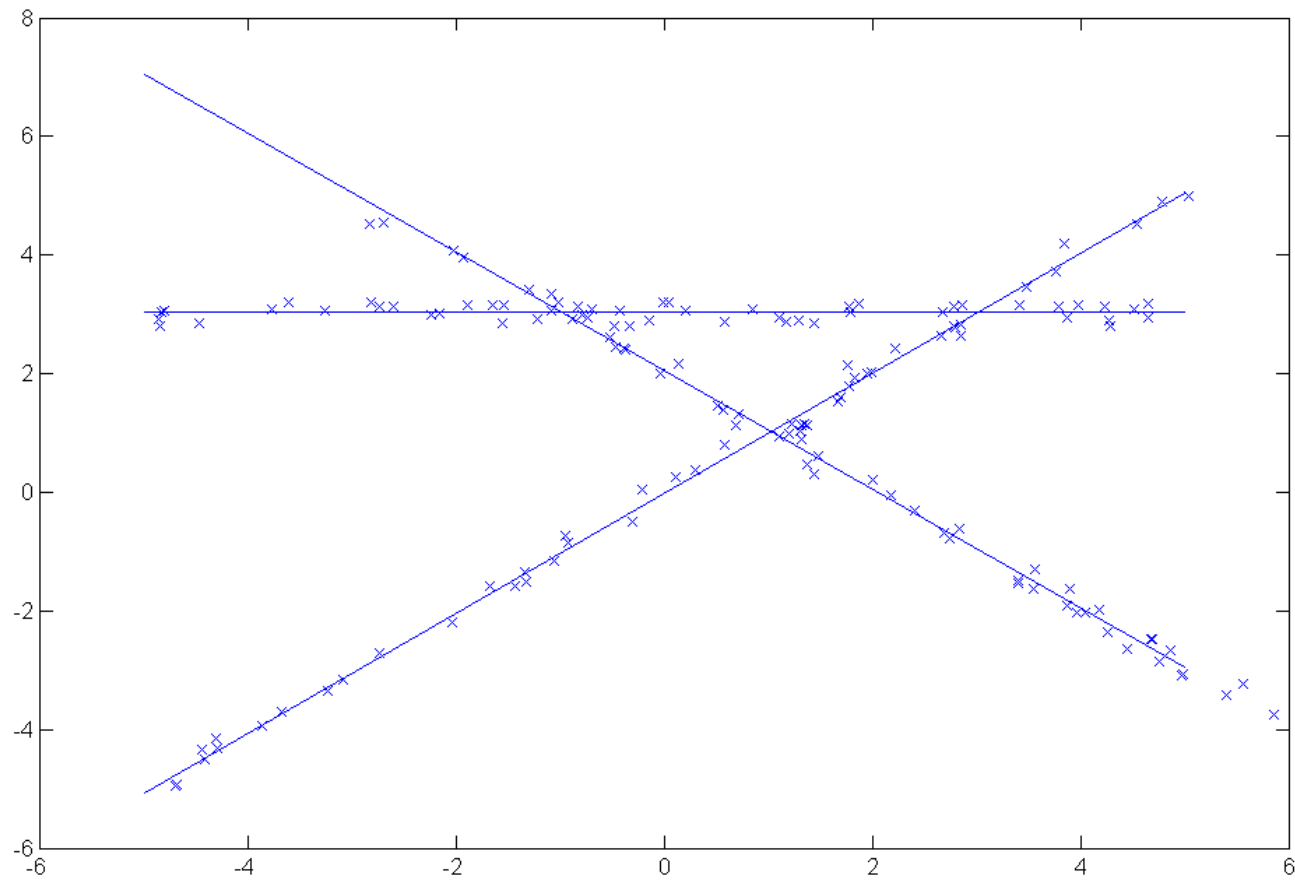
Matlab Example

Author: Christopher Bishop

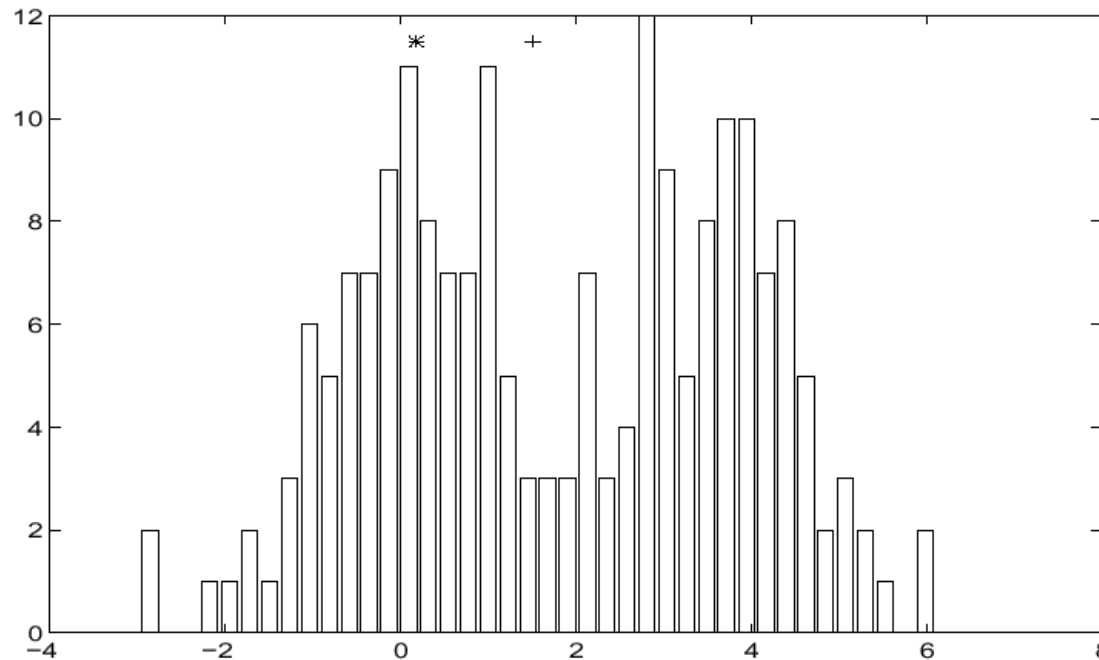
Matlab Beispiel

- Basic implementation
- kmeansN
- Alveola

KLines



Der Mean-Shift Algorithm



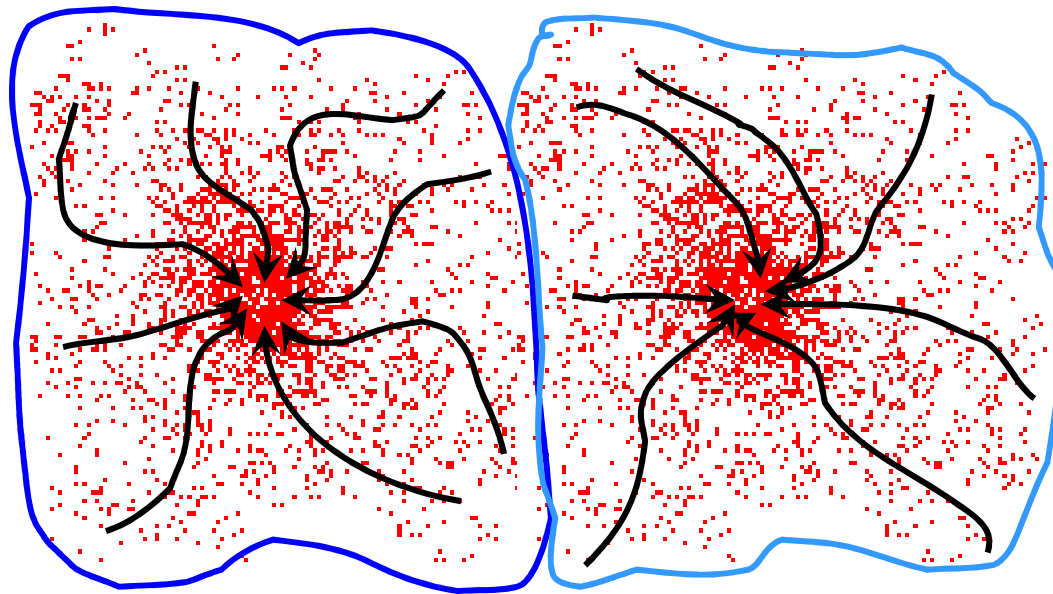
- **Iterative Mode Search**

1. Initialize random seed, and window W
2. Calculate center of gravity (the “mean”) of W : $\sum_{x \in W} xH(x)$
3. Shift the search window to the mean
4. Repeat Step 2 until convergence

Slide credit: Steve Seitz

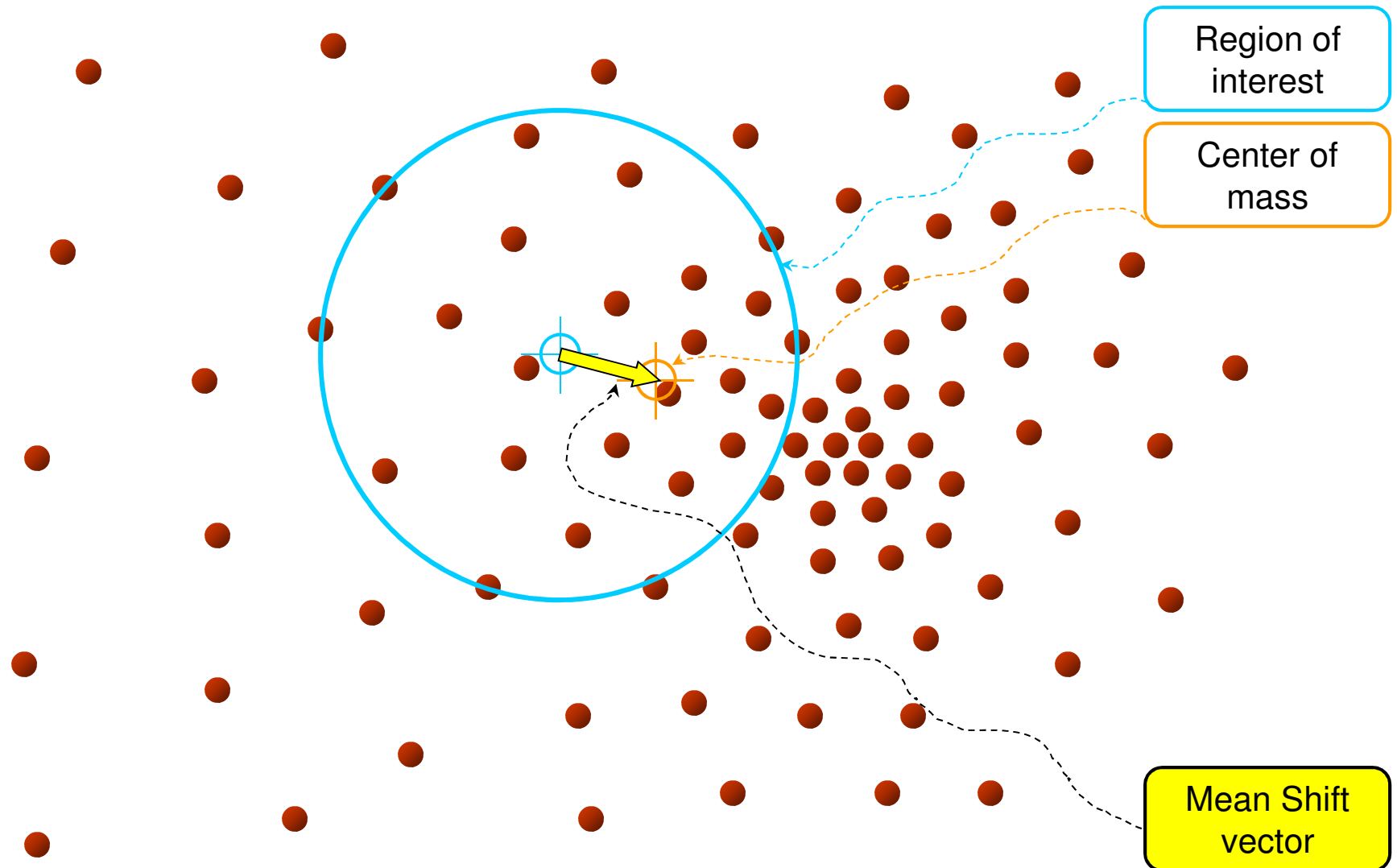
Mean-Shift Clustering

- Cluster: Alle Punkte werden zu einem Extremum hingezogen



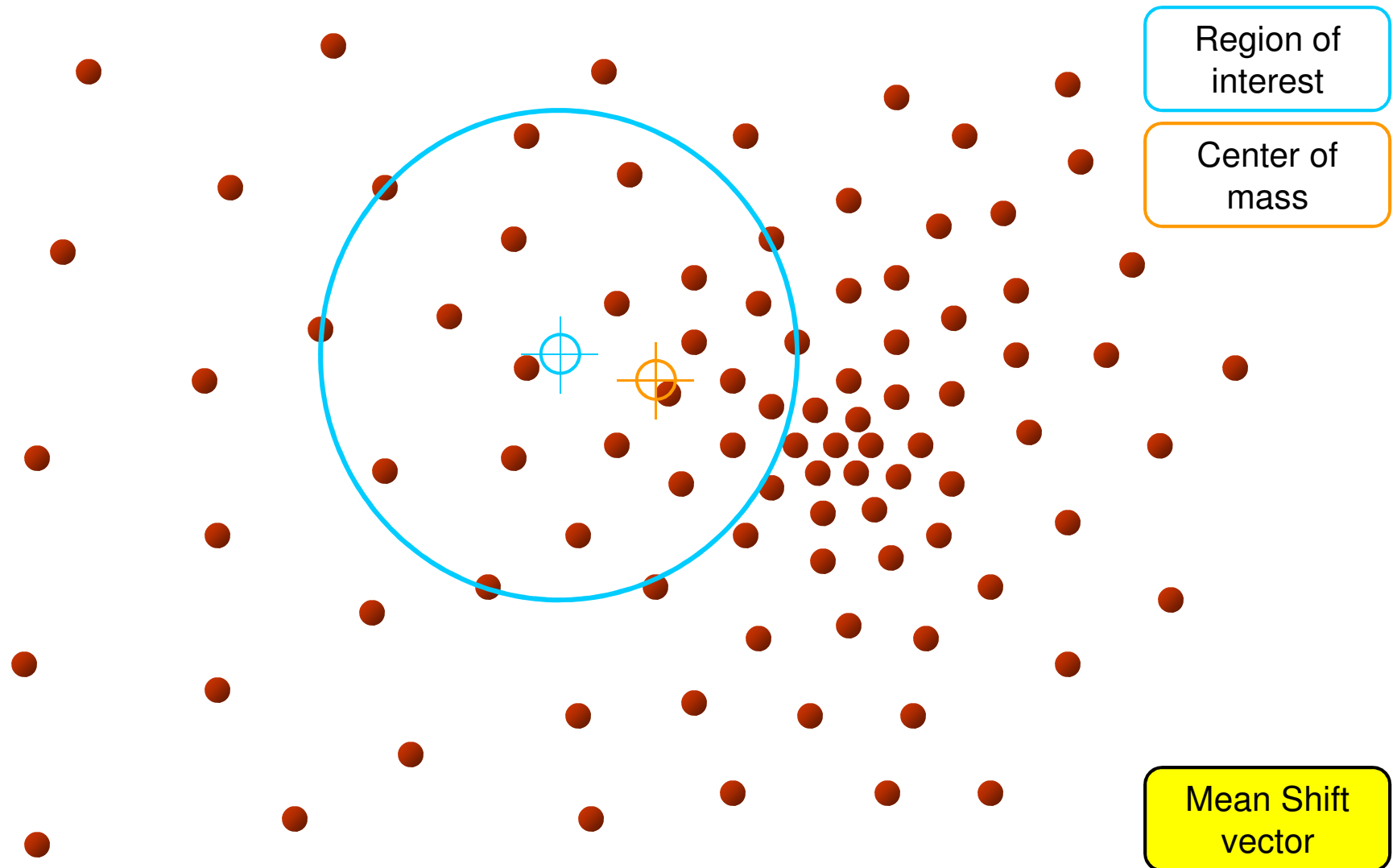
Slide by Y. Ukrainitz & B. Sarel

Mean shift



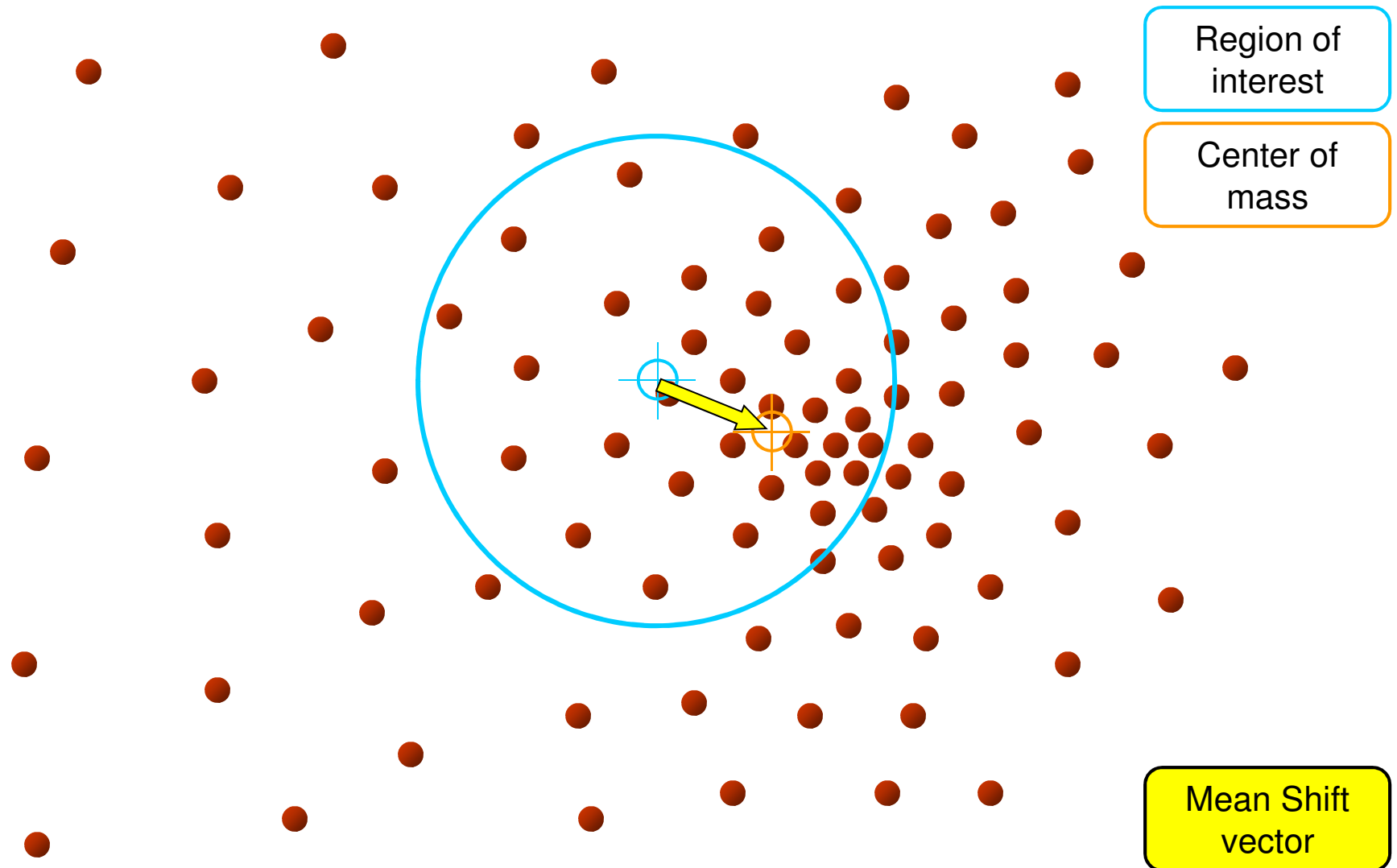
Slide by Y. Ukrainitz & B. Sarel

Mean shift



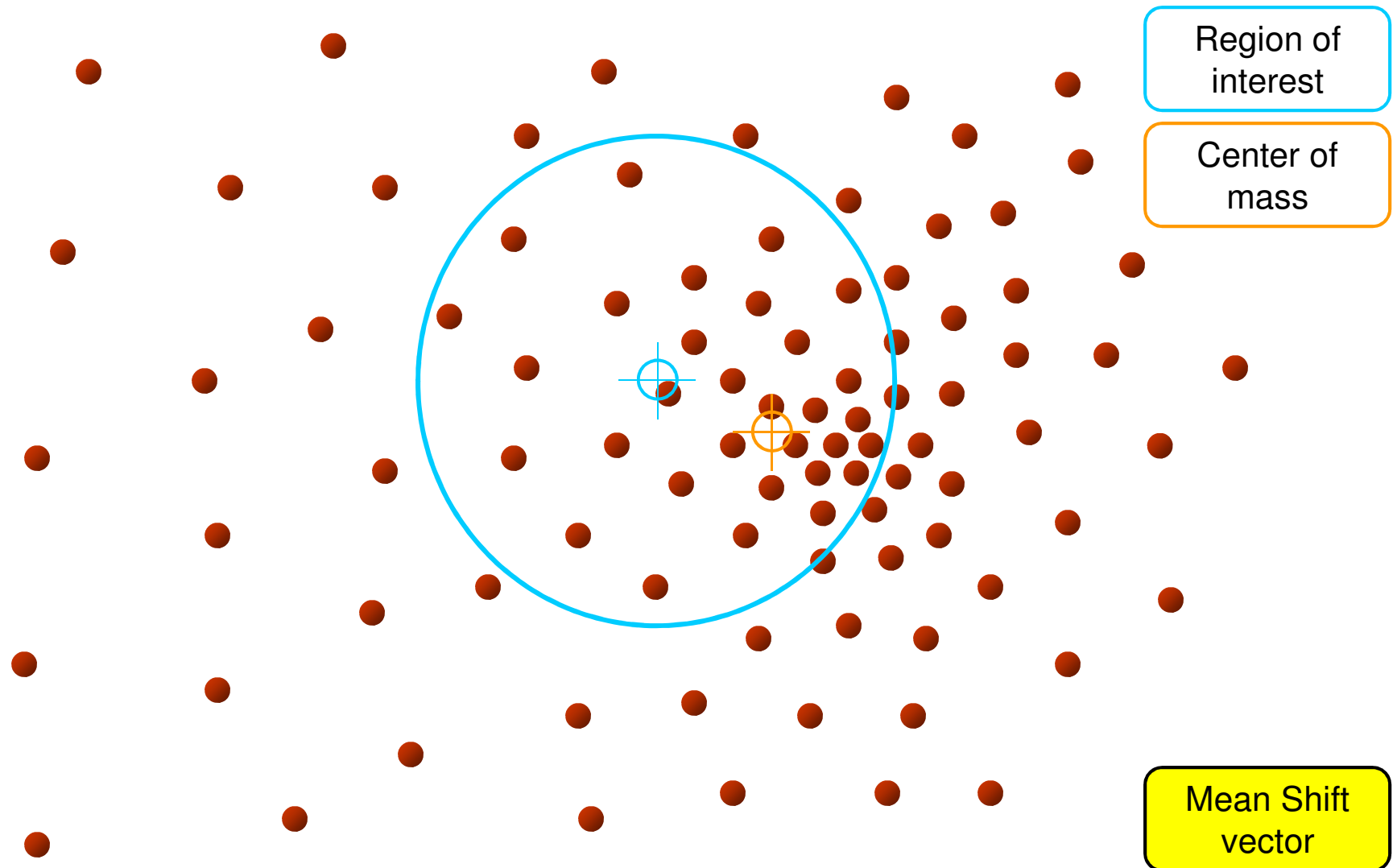
Slide by Y. Ukrainitz & B. Sarel

Mean shift



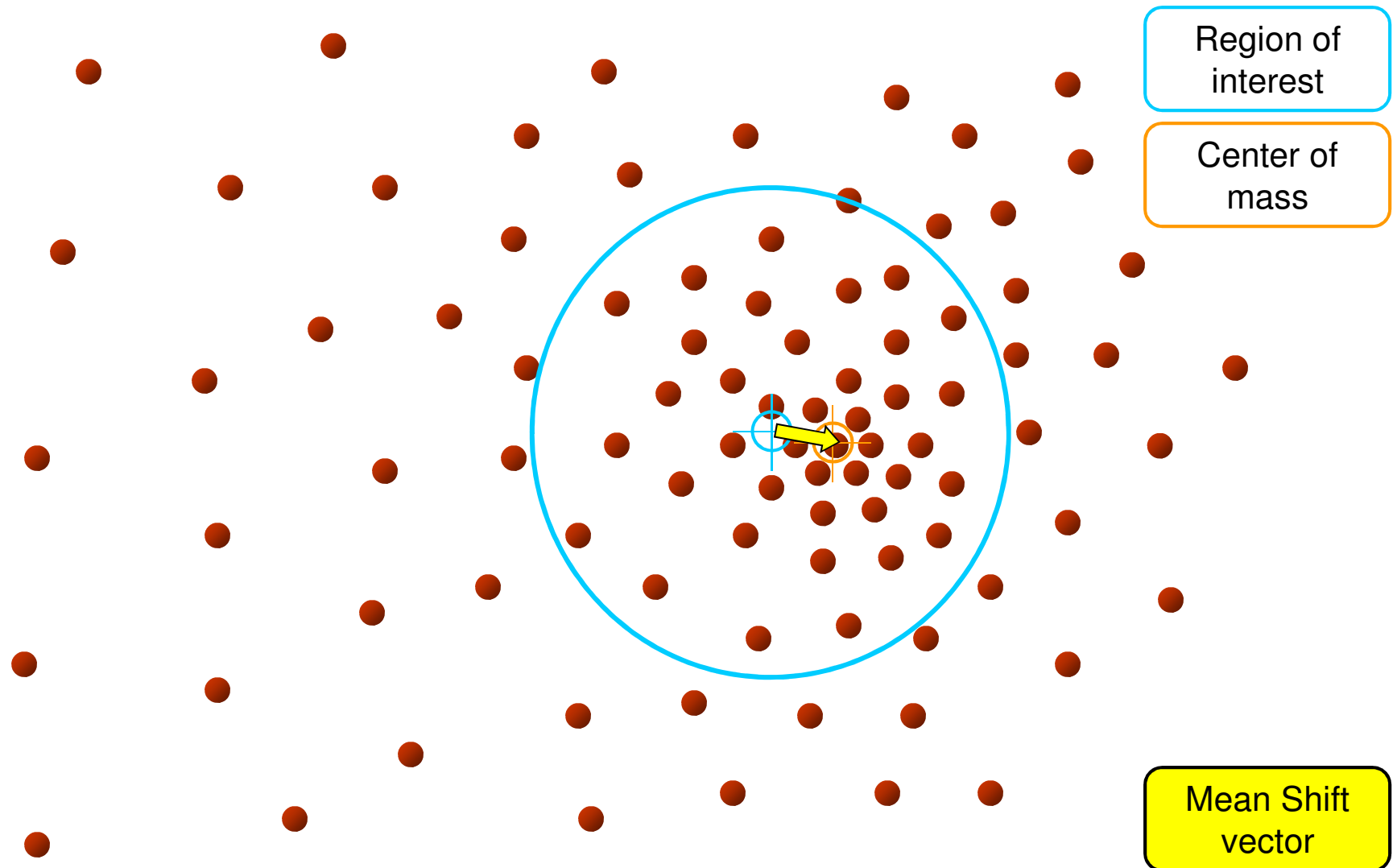
Slide by Y. Ukrainitz & B. Sarel

Mean shift



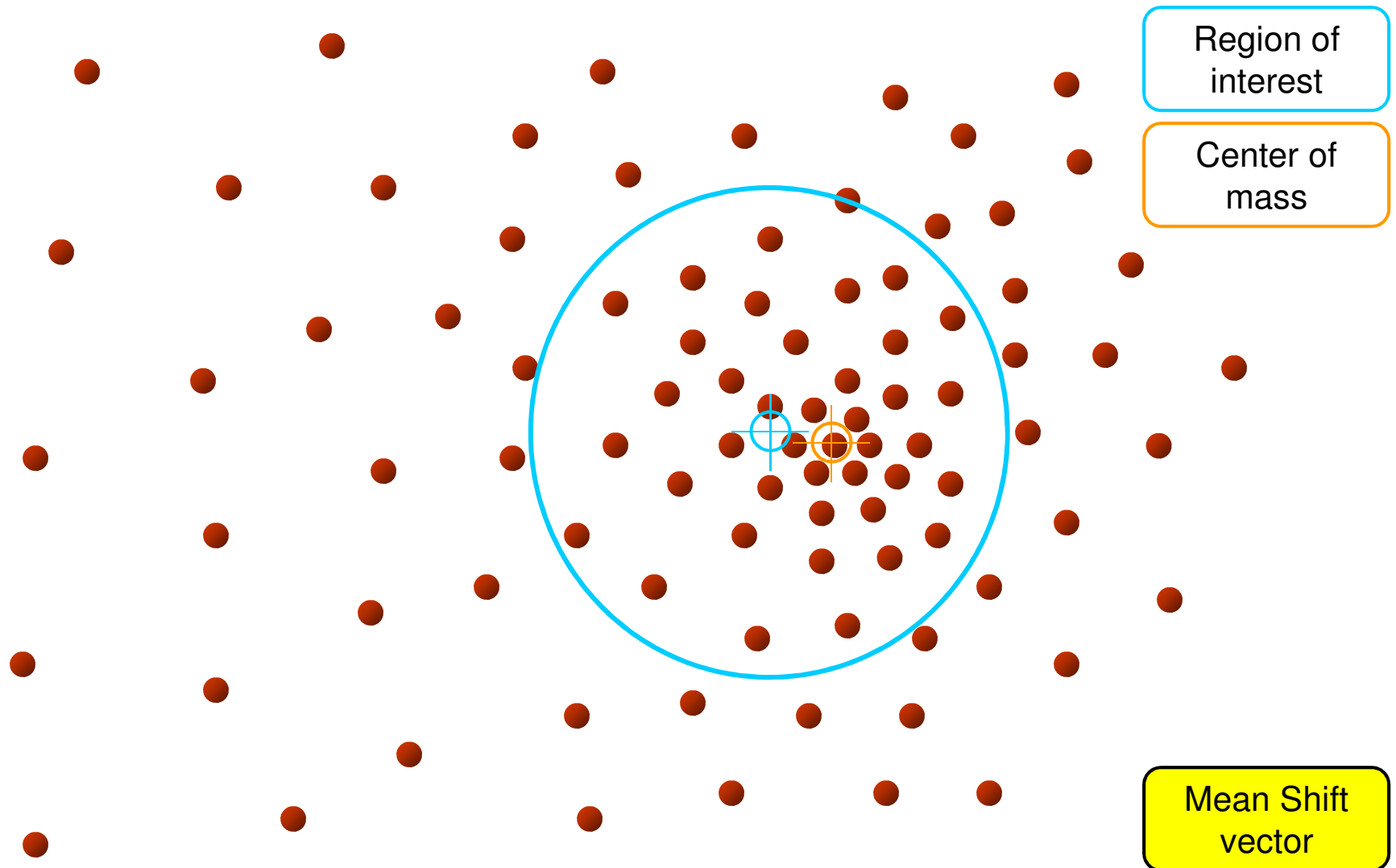
Slide by Y. Ukrainitz & B. Sarel

Mean shift



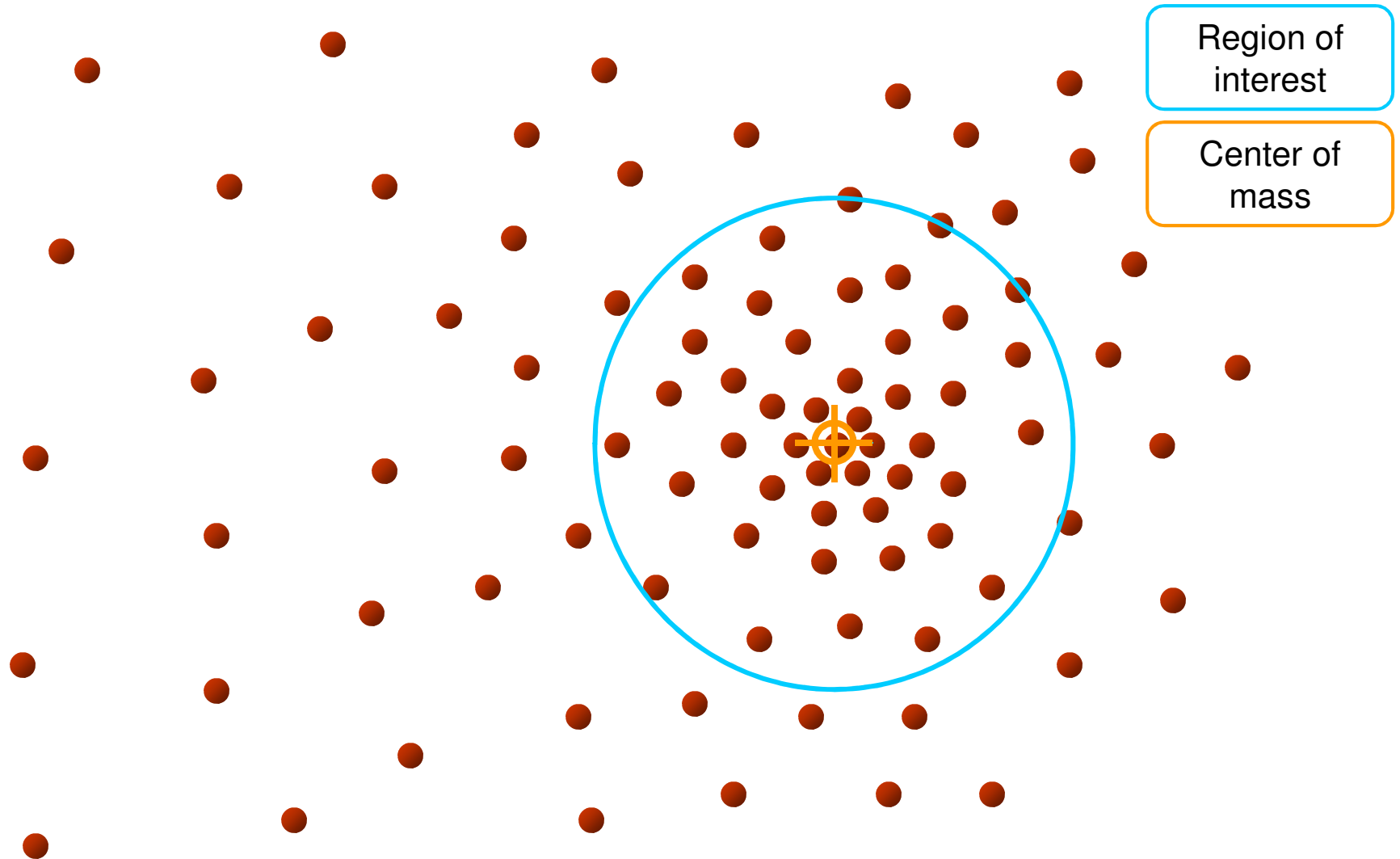
Slide by Y. Ukrainitz & B. Sarel

Mean shift



Slide by Y. Ukrainitz & B. Sarel

Mean shift



Slide by Y. Ukrainitz & B. Sarel

Beispiel (Bilder)

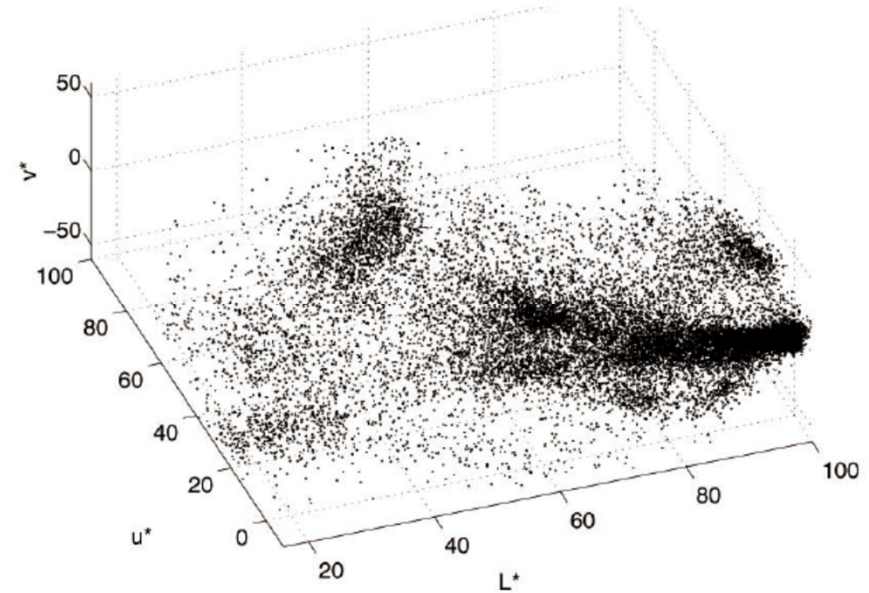
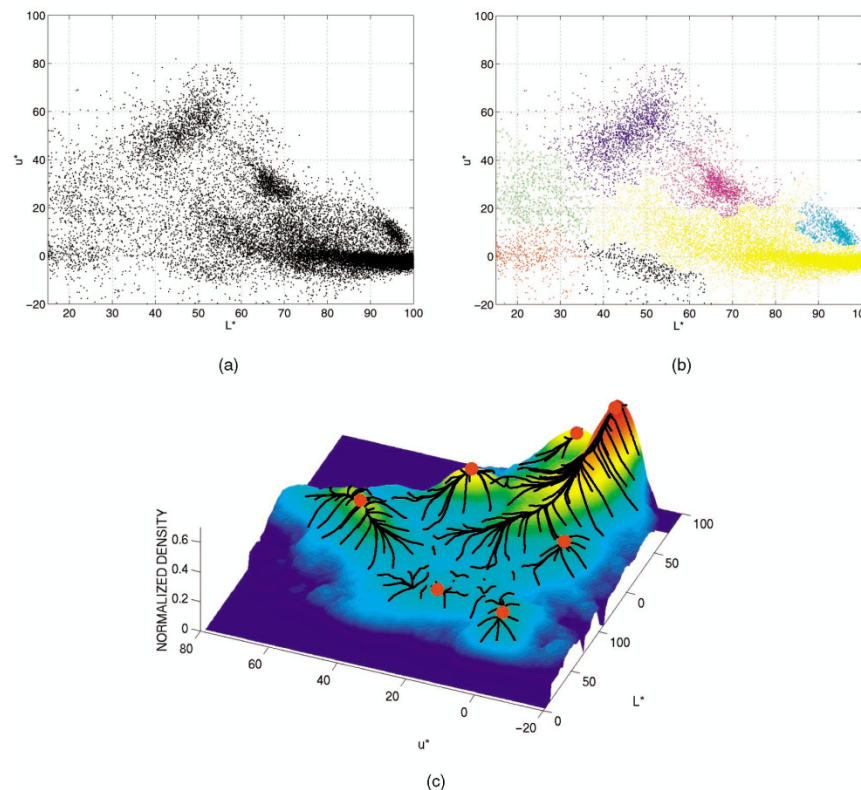


Bild + Farbwerte im LUV-Farbraum

Mean shift clustering/segmentation

- Finde Features (Farbe, Gradienten, Textur ...)
- Initialisiere Fenster an Pixel-Positionen
- Führe Mean-shift durch bis Verfahren konvergiert
- Verbinde Fenster die in einem *ähnlichen* Peak enden



Bildsegmentierung mit Mean-Shift



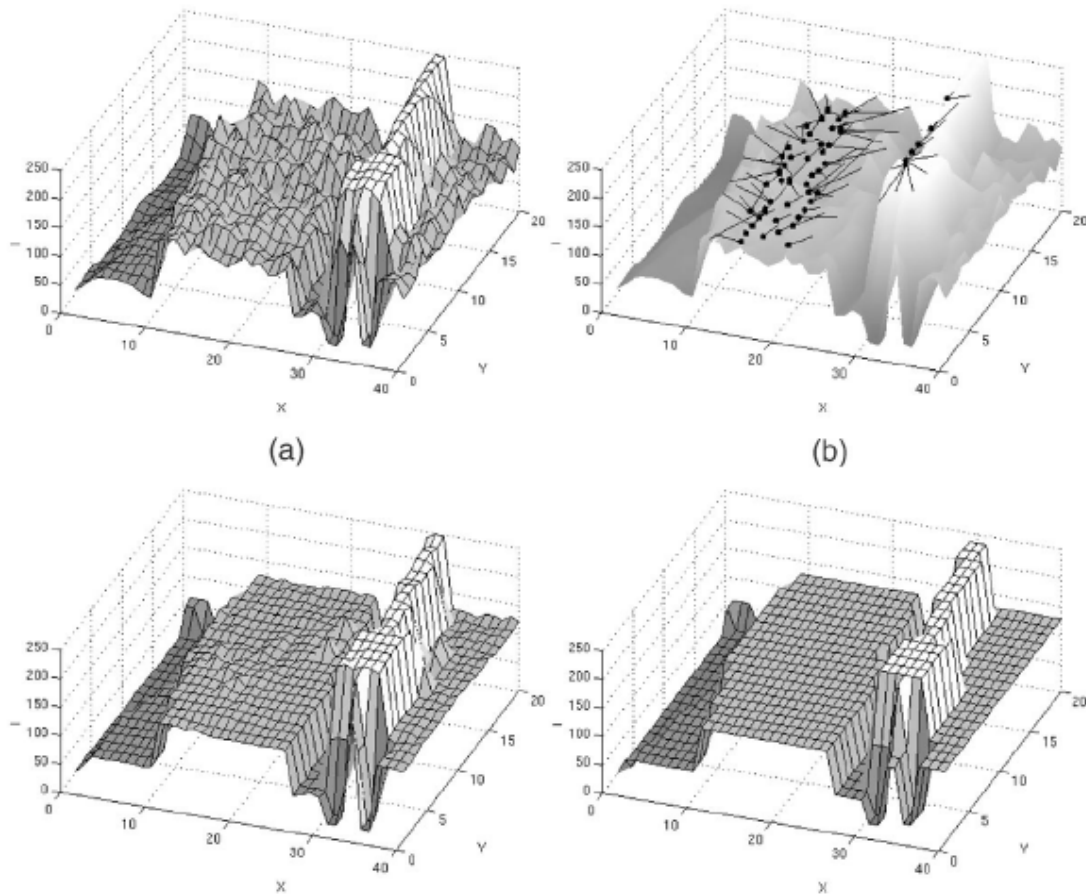
(a)



(b)

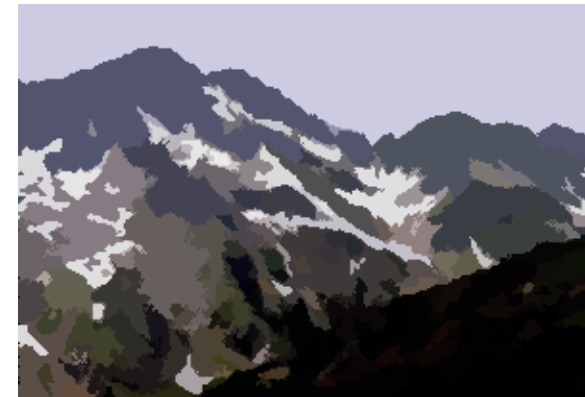
Im Gegensatz zu k-means muss man nicht die Anzahl der Cluster vorgeben

Mean-Shift



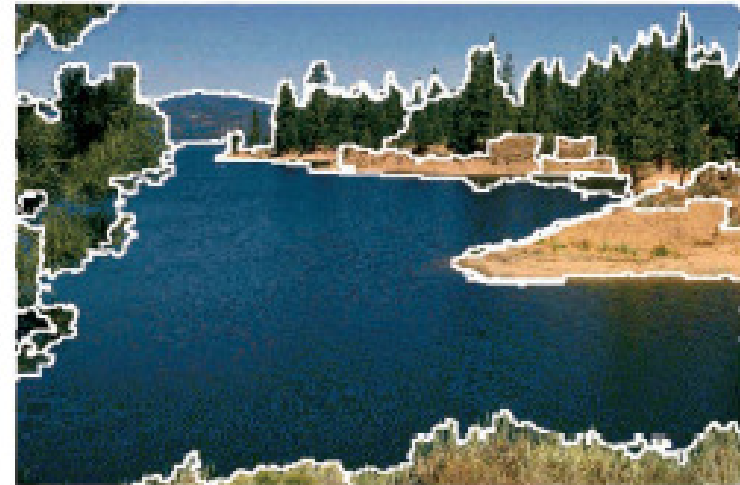
Grauwert-Bild als Oberfläche

Beispiele

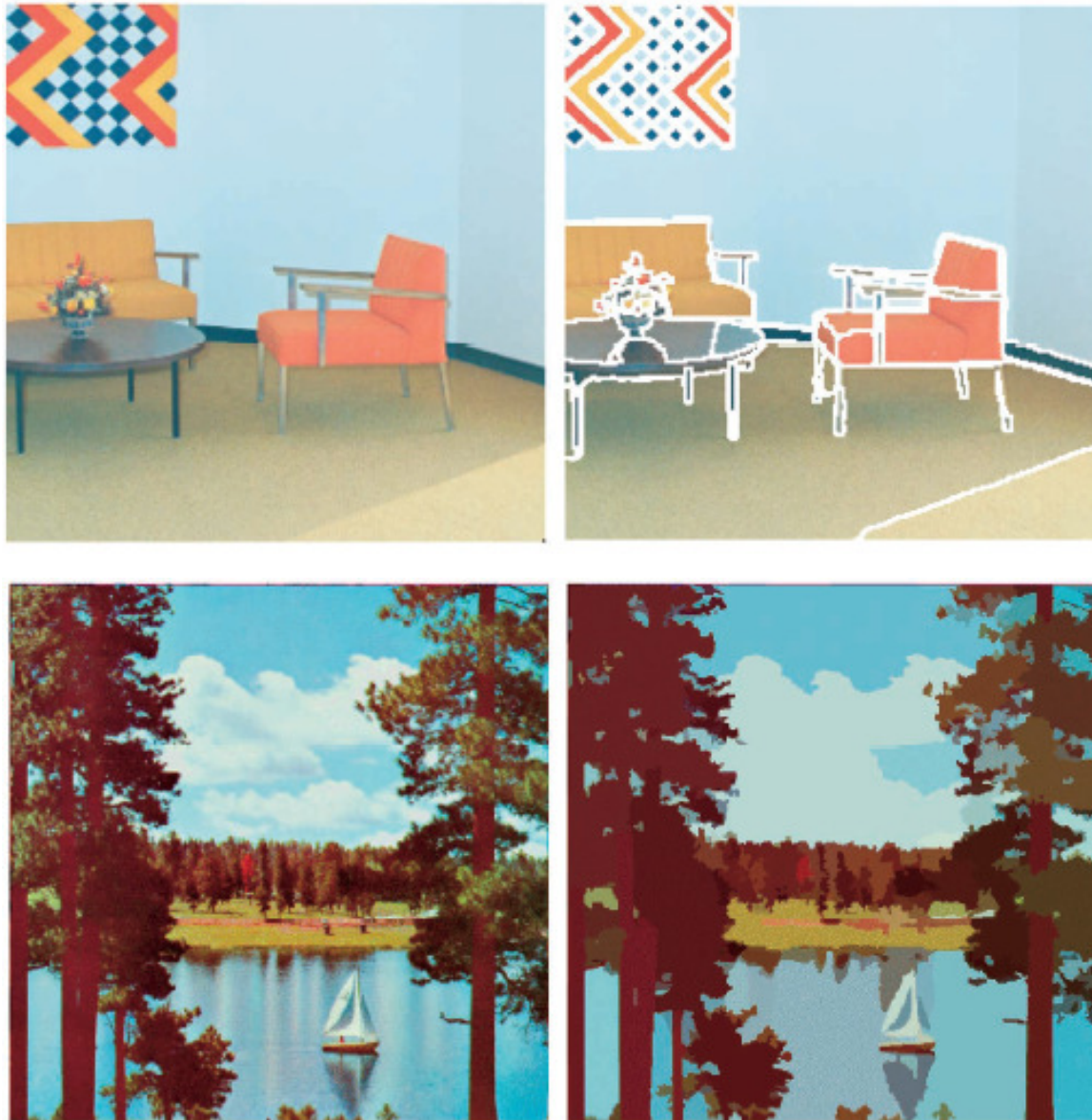


<http://www.caip.rutgers.edu/~comanici/MSPAMI/msPamiResults.html>

Beispiele



Beispiele



Topics

- K-Means
- Mean-Shift