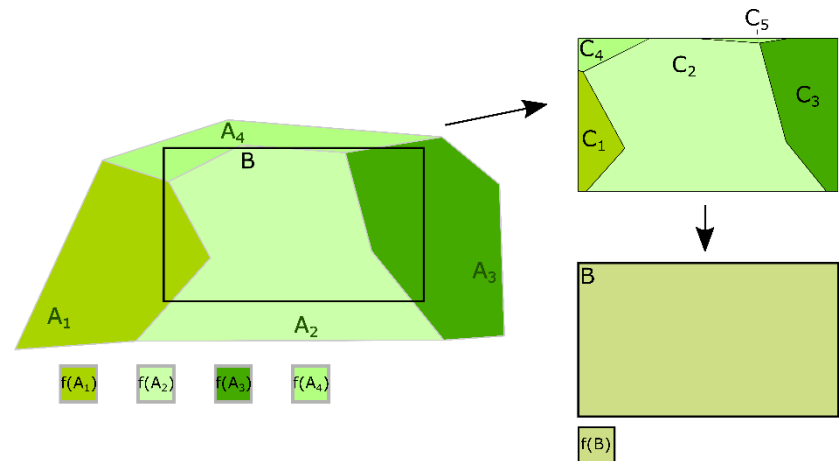


Sampling Transformationen

Probleme

- Gegeben sind ein oder mehrere Geoobjekte mit Attributen. Diese Attributwerte sollten jetzt für einen beliebigen Punkt oder eine Region abgefragt/ermittelt werden ...
 - Punkt → Punkt: **Interpolation!**
 - Linie → Punkt: **Interpolation!**
 - Fläche → Punkt ...
 - Fläche → Fläche ...

- Geoobjekte in einer gegebenen Auflösung sollen in eine andere Auflösung überführt werden (*Resampling*)



Problem

Gegeben sind ein oder mehrere Flächenobjekte mit Attributen. Diese sollten jetzt an gegebenen Punktpositionen ausgewertet werden.

- Geometrisches Problem: **Punkt-in-Polygon-Test**
Befindet sich ein Punkt innerhalb eines Polygons?
- Verschiedene Verfahren basierend auf verschiedenen Arten von Polygonen
 - Achs-paralleles Rechteck
 - Dreieck
 - Beliebiges Polygon

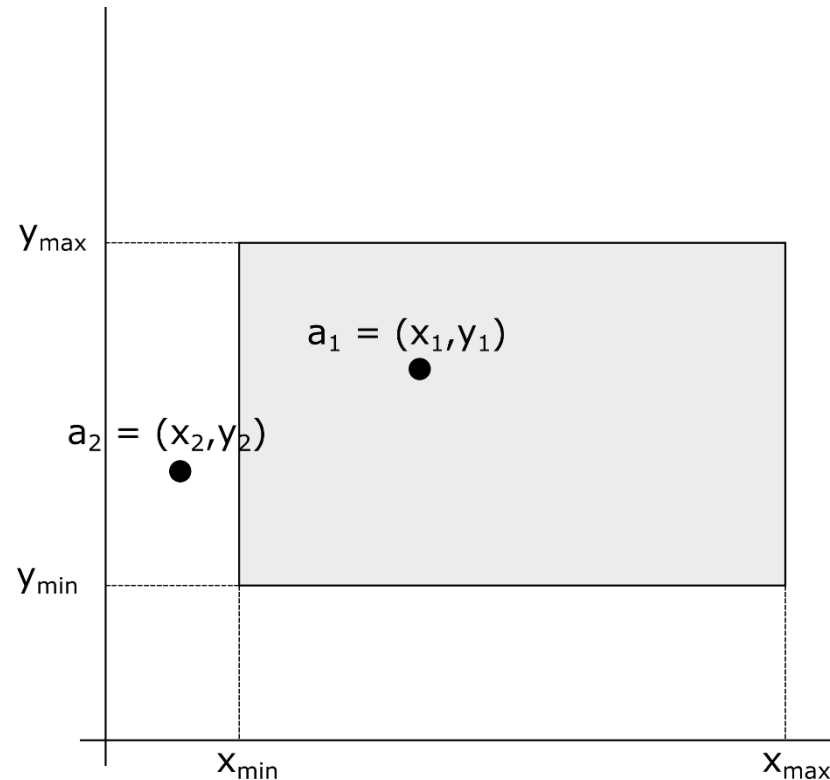
Bounding-Box-Test

- Gegeben:
Rechteck $R = (x_{min}, x_{max}, y_{min}, y_{max})$
Abfragepunkt (x, y)

- Bedingung:

$$x \in R \rightarrow \begin{cases} x_{min} \leq x \leq x_{max} \\ y_{min} \leq y \leq y_{max} \end{cases}$$

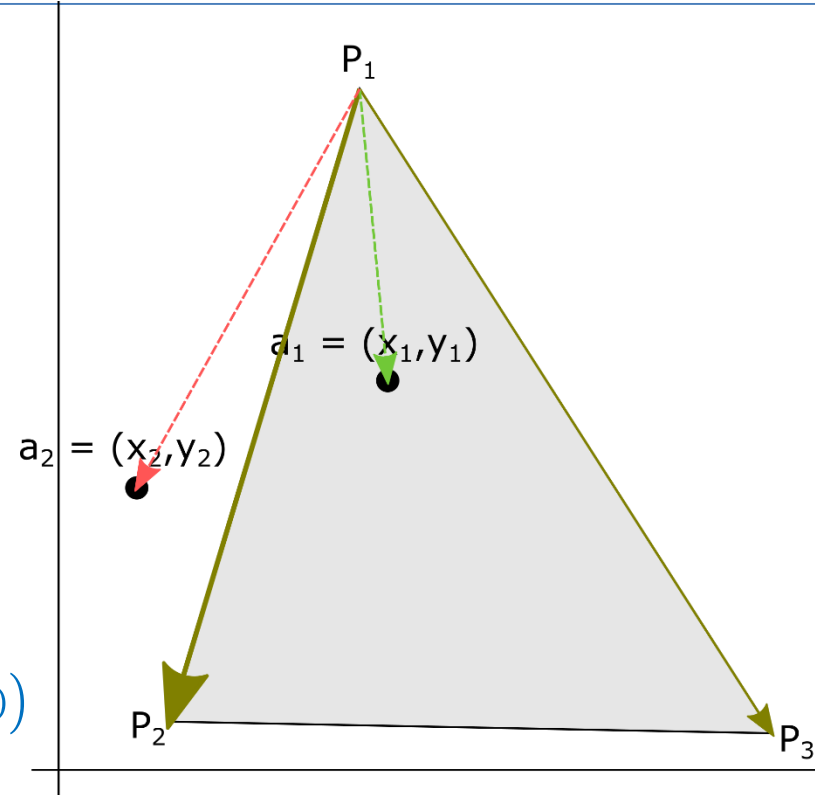
- Vorteil:
 - sehr effizient
 - leicht erweiterbar
in höhere Dimensionen
- Nachteil: begrenzt anwendbar
- Anwendung:
Identifizierung, ob ein Punkt in der AABB* eines Objektes liegt, wenn ja lohnen sich aufwändigere Verfahren für genauere Abfragen.



*axis-aligned bounding box

Punkt-in-Dreieck-Test

- Gegeben:
Dreieck $\Delta(\vec{p}_i, \vec{p}_j, \vec{p}_k)$
Abfragepunkt $\vec{x} = (x, y)$
 - Ansatz:
 - jede Dreiecksseite teilt den Raum in 2 Halbräume
 - liegt \vec{x} im gleichen Halbraum wie der der Kante (\vec{p}_i, \vec{p}_j) gegenüberliegende Punkt \vec{p}_k liegt \vec{x} potentiell im Dreieck Δ
 - $w_{ij}(\vec{x}) > 0$ mit
- $$w_{ij}(\vec{x}) = ((\vec{p}_j - \vec{p}_i) \times (\vec{p}_k - \vec{p}_i)) \cdot ((\vec{p}_j - \vec{p}_i) \times (\vec{x} - \vec{p}_i))$$
- $\vec{x} \in \Delta \rightarrow w_{ij}(\vec{x}) > 0, w_{jk}(\vec{x}) > 0, w_{ki}(\vec{x}) > 0$
- Test kann abgebrochen werden, wenn **ein** $w < 0$ gefunden wird
 - Vorteil: Berechnung leicht optimierbar
 - Nachteil: nur für Dreiecke geeignet



Punkt-in-Polygon-Test / Strahlmethode nach Jordan

- Gegeben:
 Polygon $P(\vec{p}_1, \vec{p}_2, \dots, \vec{p}_n)$ mit Sequenz aus n Randpunkten
 \Rightarrow Kante $k_i = \vec{p}_{i+1} - \vec{p}_i, 1 \leq i \leq n - 1; k_n = \vec{p}_1 - \vec{p}_n$

 Abfragepunkt $\vec{x} = (x, y)$
- Ansatz:
 1. Definiere einen Strahl mit beliebiger Richtung \vec{r} , ausgehend von \vec{x}
 2. Teste für jede Kante, ob der Strahl die Kante schneidet

$$\forall k_i: \vec{x} + \lambda_r \vec{r} = \vec{p}_i + \lambda_i (\vec{p}_{i+1} - \vec{p}_i)$$
 - Schnitt wenn $\lambda_i \geq 0$ und $\lambda_i < 1$
 3. Zähle alle Schnitte zwischen Strahl und Kanten: n_h
- n_h ist **ungerade**: $\vec{x} \in P$
- n_h ist **0 oder gerade**: $\vec{x} \notin P$

Punkt-in-Polygon-Test / Strahlmethode nach Jordan

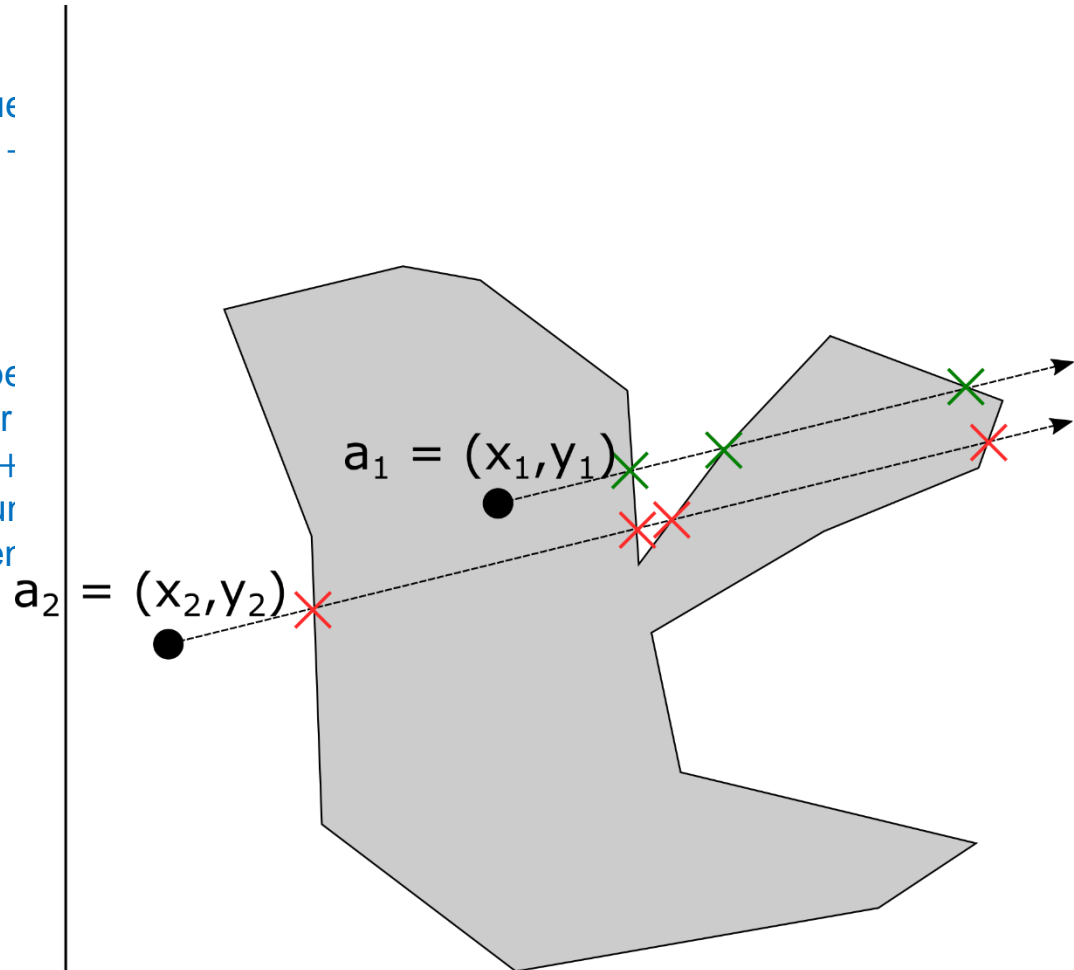
- Gegeben:
 Polygon $P(\vec{p}_1, \vec{p}_2, \dots, \vec{p}_n)$ mit Sequenz
 \Rightarrow Kante $k_i = \vec{p}_{i+1} - \vec{p}_i, 1 \leq i \leq n - 1$

Abfragepunkt $\vec{x} = (x, y)$

- Ansatz:
 1. Definiere einen Strahl mit beliebigem Richtungsvektor \vec{r}
 2. Teste für jede Kante, ob der Strahl die Kante schneidet
 $\forall k_i: \vec{x} + \lambda_r \vec{r} = \vec{p}_i + \lambda_i k_i$
 \Rightarrow Schnitt wenn $\lambda_i \geq 0$ und $\lambda_r < 0$
 3. Zähle alle Schnitte zwischen Strahl und Polygon

- n_h ist **ungerade**: $\vec{x} \in P$
- n_h ist **0 oder gerade**: $\vec{x} \notin P$

- Vorteil: funktioniert für **alle** Polygone
- Nachteil: aufwändig, es müssen immer **alle** Kanten getestet werden



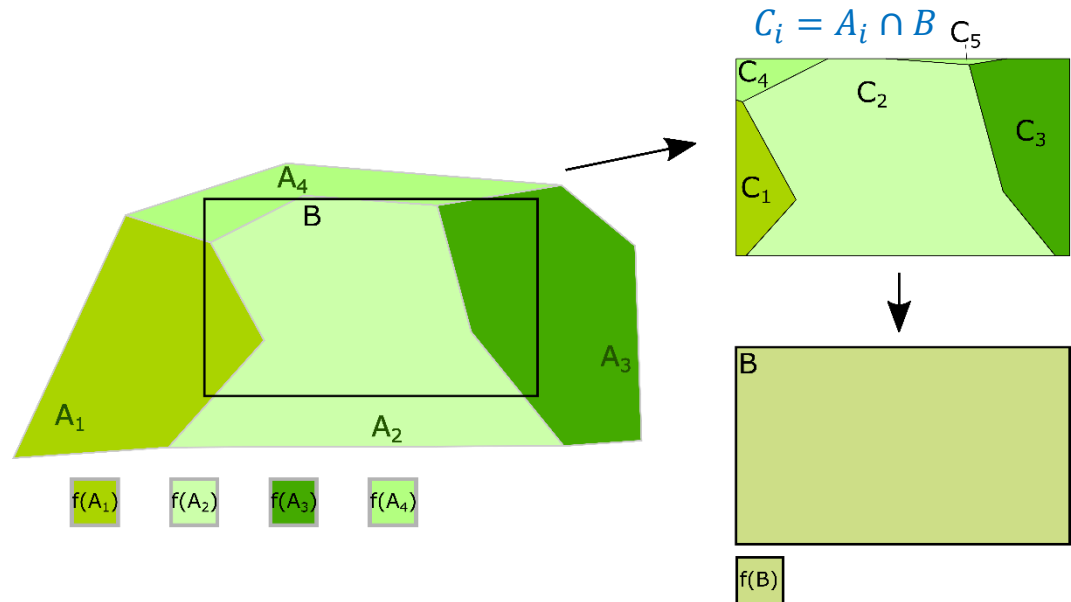
Fläche-zu-Fläche Transformation

Problem

Gegeben sind ein oder mehrere Flächenobjekte mit Attributen. Diese sollten jetzt für ein gegebenes anderes Flächenobjekt ausgewertet werden.


- Gegeben: Menge von Polygonen $(A_i, f(A_i))$ und ein Abfragepolygon B
- Gesucht: $f(B) = \sum_i \lambda_i f(A_i)$

$$\lambda_i = \frac{\text{Fläche}(A_i \cap B)}{\text{Fläche}(B)}$$



upsampling

➤ Erhöhung der Auflösung

- Hinzufügen von zusätzlichen unabhängigen Daten 
- Interpolation ...? → keine zusätzliche Information

downsampling

➤ Verringerung der Auflösung

- Interpolation ...? → Gefahr von Informationsverlust
- **weeding**

weeding

- Entfernen von Datenpunkten mit dem geringsten Beitrag zum Informationsgehalt
- basierend auf Geometrie und/oder Parameterverteilung
- Ziel: verbleibende Datenpunkte umfassen die relevante Information
- Beispiel: **Douglas-Peucker-Algorithmus zur Ausdünnung von Liniendaten**

Douglas-Peucker-Algorithmus zur Ausdünnung von Liniendaten

1. Verbinde Anfangs- und Endpunkt mit einer geraden Linie
 2. Füge den Datenpunkt ein, der den größten senkrechten Abstand zur Verbindungslinie
 - Zwei neue Segmente
 3. Wiederhole Punkt 2 für jedes neue Segment bis kein Datenpunkt ist weiter entfernt als gegebenes Minimum
- Datenpunkte, welche nicht eingefügt werden, tragen zu wenig Information zur Liniengeometrie bei

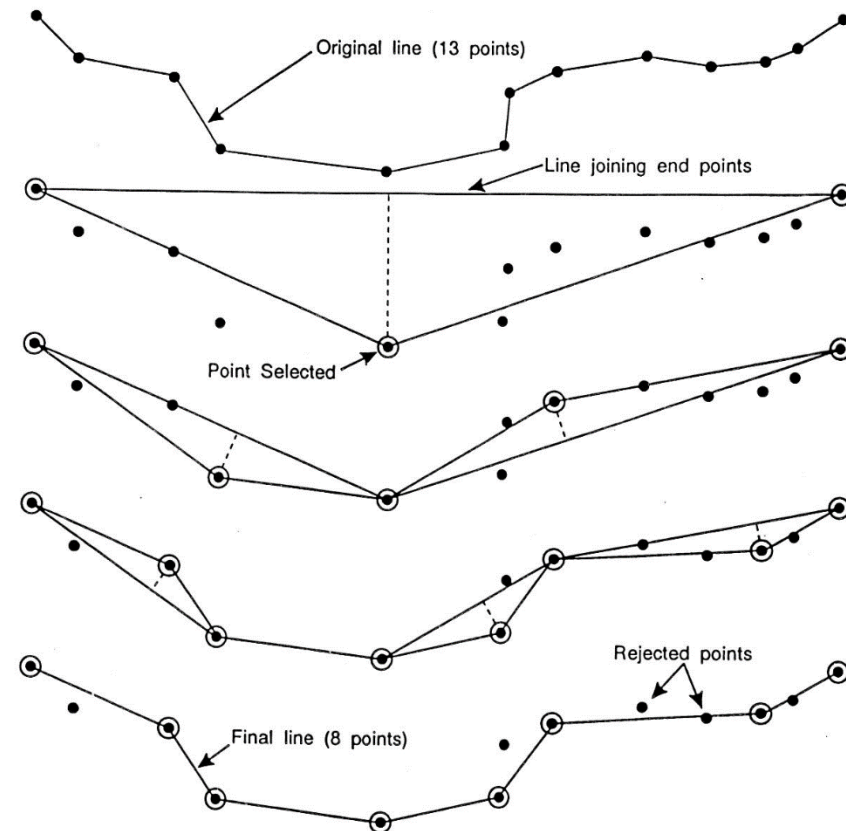


FIG. 6-10. Douglas-Peucker algorithm for line generalization. The original line contains 13 points, and the final line contains 8 points. The rejected points are said to be "weeded". A number of alternative algorithms for line weeding have been proposed.