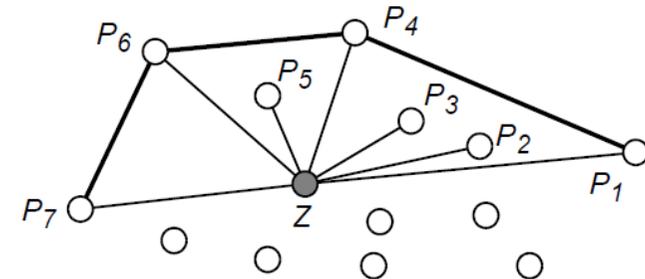


Fächermethode von Graham (1972)

- Bestimme einen Zentrumspunkt Z
- Sortiere sämtliche Punkte P_i nach aufsteigendem Winkel bezüglich Z
- Durchlaufe die Punkte P_i gegen den Uhrzeigersinn
- Betrachte dabei immer aufeinanderfolgende Punkte P_k, P_{k+1} und P_{k+2} :
 - 1. Fall: P_{k+2} erzwingt Linksdrehung: Weiterlaufen
 - 2. Fall: P_{k+2} erzwingt Rechtsdrehung:
 - Lösche P_{k+1}
 - Betrachte wiederholt P_{k-j}, P_{k+1-j} und P_{k+2} , bis kein Punkt mehr gelöscht wird



⇒ Die konvexe Hülle dient als Grundlage zur Berechnung des minimal umgebenden n-Ecks und des gedrehten MURs

- Das Rastermodell kann als flächiges Modell bezeichnet werden
 - Flächen werden durch flächige Basiselemente (explizit) dargestellt
 - Basisoperationen: einfach und unabhängig voneinander ausführbar
- Das Vektormodell kann als Linienmodell bezeichnet werden
 - Flächen werden durch Beschreibung des Randes (implizit) dargestellt
 - Geringer Speicherplatzbedarf
- *Strukturelle Zerlegungsrepräsentation* ist eine flächige Vektorrepräsentation
 - Zerlegung in Basiskomponenten
 - vollständig
 - disjunkt
 - Basiskomponenten sind im Vektormodell beschrieben
 - Explizite Repräsentation von Flächen
 - Basisoperationen unabhängig voneinander ausführbar
 - Geringer Speicherplatzbedarf

Motivation

- Beobachtung: komplexe und zeitaufwendige Algorithmen dominieren die Anfragebearbeitung
- Prinzip: Divide-and-Conquer
Zerlegung in einfache Komponenten
- Prinzip: Einsatz geeigneter geometrischer Datenstrukturen
- Prinzip: Vorverarbeitung
 - Investition von Zeit und Speicherplatz bei der Repräsentation
 - ⇒ höherer Aufwand zum Einfügezeitpunkt
 - ⇒ geringerer Aufwand bei der Anfrage

Vorteile

- + Vereinfachung der algorithmischen Komplexität von Anfragen und Operationen
- + Lokalität von Anfragen und Operationen kann ausgenutzt werden

Charakterisierung verschiedener Zerlegungen

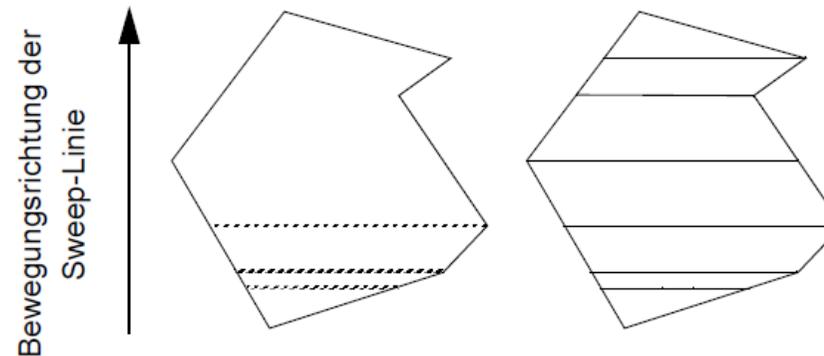
- qualitative Aspekte
 - Welche Typen von Komponenten werden erzeugt?
 - Ist die Menge der Komponenten homogen oder heterogen?
 - Ist die Zerlegung eindeutig?
 - Ist die Beschreibungslänge der Komponenten fest oder variabel?
 - Erfüllt die Zerlegung ein Gütekriterium?
 - Wie geeignet ist die Zerlegung für eine Verwaltung durch eine geometrische Datenstruktur?
- quantitative Aspekte
 - Anzahl der erzeugten Komponenten
 - Speicherplatzbedarf der Zerlegungsrepräsentation
 - Zeitkomplexität des Zerlegungsalgorithmus

5.2 Trapezzerlegung

- Zerlegung eines Polygons in (achsenparallele) Trapeze (Asano+Asano 1983)



- Berechnung durch Plane-Sweep-Algorithmus: $O(n \log n)$ für n Eckpunkte



- Speicherplatzaufwand
 - Anzahl der Komponenten = n bei n Eckpunkten
 - aber Vervielfachung des Speicherplatzes da nun Trapeze statt Punkten verwaltet, d.h. abgespeichert werden

Beobachtung

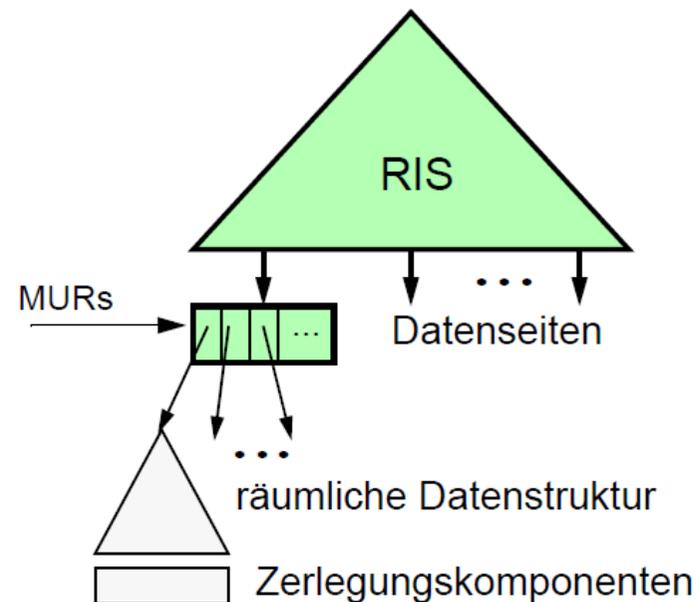
- Operationen auf Zerlegungskomponenten sind einfach
 - Anzahl der Zerlegungskomponenten: $O(n)$
- ⇒ Kein Gewinn, falls alle Komponenten getestet werden
- ⇒ Einsatz von Datenstrukturen zur Auswahl "relevanter" Komponenten
- ⇒ Einsatz von räumlichen Indexstrukturen (RIS)

1. Ansatz

- *Eine* RIS verwaltet Zerlegungskomponenten *aller* Objekte
 - Redundanz bei der Anfragebearbeitung (betrifft insbesondere größere Window Queries)

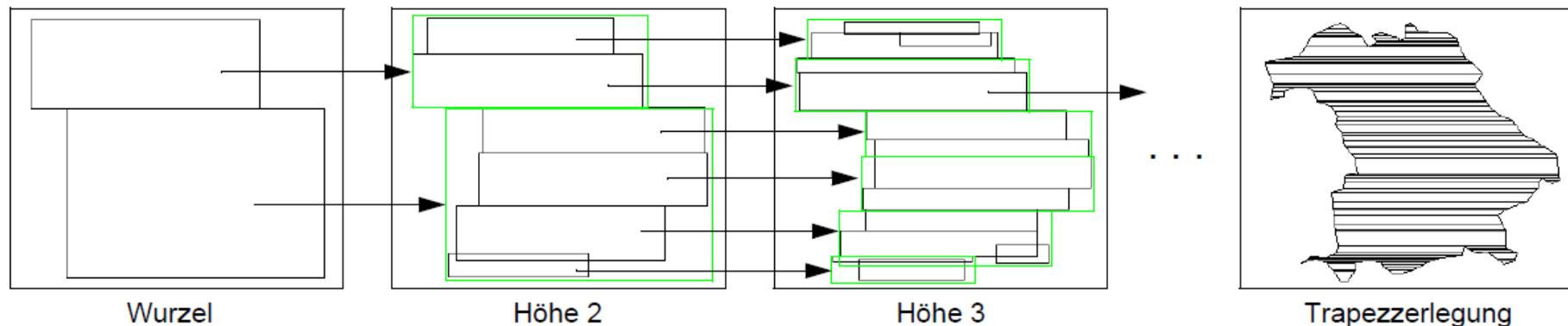
2. Ansatz

- Eine RIS verwaltet die Objektapproximationen (MUR) aller Objekte
- Für jedes Objekt verwaltet eine separate räumliche Datenstruktur die Zerlegungskomponenten dieses Objektes
- Wenn die exakte Objektgeometrie untersucht werden muß, werden die Zerlegungskomponenten einschließlich der zugehörigen räumlichen Datenstruktur vom Sekundärspeicher in den Hauptspeicher eingelesen



TR-Ansatz (Ausprägung des 2. Ansatzes)

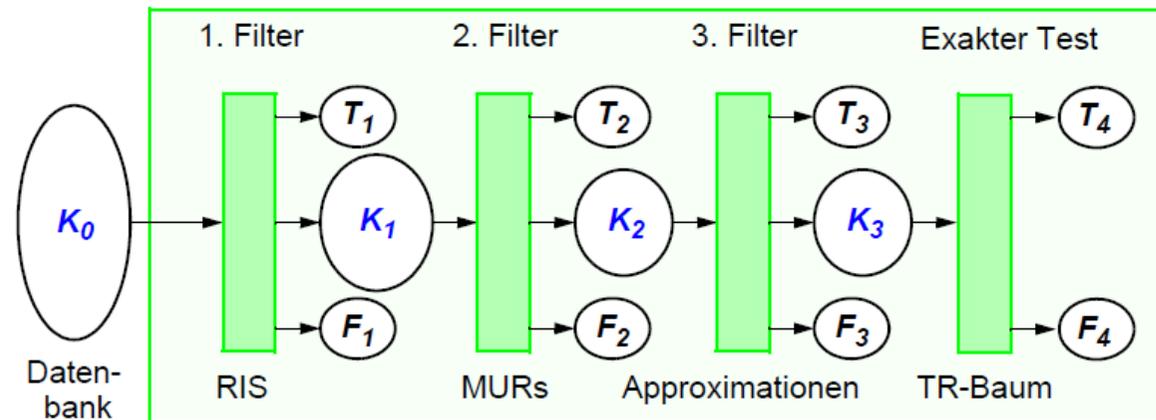
- Verwende R-Baum zur Verwaltung der Zerlegungskomponenten
- ⇒ Anpassung des R-Baums an die neuen Anforderungen (*TR-Baum*):
 - TR-Baum soll für den Hauptspeicher ausgelegt werden
 - möglichst kleine Knotengröße
 - TR-Baum soll möglichst schnell in den Hauptspeicher eingelesen werden
 - kompakte Speicherung auf dem Plattenspeicher
 - kein dynamischer Aufbau, keine Adressneuberechnungen
 - TR-Baum sollte möglichst kompakt gespeichert sein



Eigenschaften des TR-Ansatzes

- + sehr schnelle Bearbeitung geometrischer Operationen (z.B. Punkt-In-Polygon-Test)
- erheblich höherer Speicherplatzbedarf (und damit höhere Übertragungskosten beim Einlesen der exakten Geometrie)

Zusammenfassung



1. Bestimmung der Datenseiten, die Treffer und Kandidaten enthalten, durch RIS
 2. Bestimmung der Objekte auf den gefundenen Datenseiten, die aufgrund ihrer MURs als Treffer in Frage kommen
 3. Bestimmung weiterer Treffer oder Fehltreffer durch zusätzliche konservative und progressive Approximationen
 4. Einlesen der exakten Geometrie und Test der exakten Geometrie bezüglich der Anfragebedingung mit Hilfe des TR-Baum-Ansatzes
- ⇒ *GENESYS: Prototyp-System, das diese Anfragebearbeitung realisiert*