
Kapitel 3

Anfragemethoden für Räumlich-Zeitliche Daten

Skript zur Vorlesung: Spatial, Temporal, and Multimedia Databases II
Wintersemester 2011/12, LMU München

© 2011 PD Dr. Matthias Renz, Johannes Niedermayer

3.1 Einleitung

3.1.1 Begriffe und Definitionen

- Was sind Räumlich-Zeitliche (Spatio-Temporal) Daten?
 - **Definition: Räumliches Objekt**
Objekt mit Attributen, die sich auf räumliche Informationen (Position, Lage, Geometrie) beziehen.
Position bezieht sich auch auf allgemeine Merkmalsräume.
 - **Definition: Räumlich-Zeitliches (Spatio-Temporal) Objekt**
Ein räumliches Objekt, dessen räumlicher Zustand (z.B.: Geometrie oder Position) sich mit der Zeit verändert. [EGSV99]
- Hauptanwendung: Moving Object Data (MOD)
 - Verwaltung von und Anfragen auf bewegliche Objekte (MODB: Moving Object Database)

- Zwei unterschiedliche Blickwinkel auf MODB:
 - Location Management Database: Verwaltung von räumlichen Zuständen (snapshots) über einen Zeitraum hinweg
 - Spatio-Temporal Database: Verwaltung von räumlich-zeitlichen Daten.

- Location Management Database (LMDB)
 - := (Snapshot) Datenbank die benutzt wird um
 - (aktuelle) Positionen einer Menge von sich bewegendenden Objekten dynamisch zu verwalten, z.B. eine Menge von Taxis.
 - (snapshot) Anfragen in der Gegenwart oder (sehr) nahen Zukunft zu unterstützen. (Monitoring)
 - Hauptanwendung:
 - *Monitoring*-Anfragen: kontinuierliche Ausgabe des gegenwärtigen Zustandes eines Anfrageergebnisses, das sich auf die Gegenwart bezieht.
 - Szenario: Client-Server Modell
 - Objekte (Clients) senden ihre jeweilige neue Position zum Server
 - Server führt einen Update auf die LM-Datenbank aus.

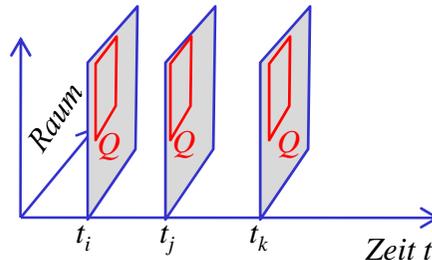
– Spatio-Temporal Database (STDB)

- Definition

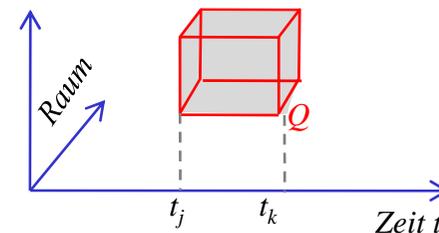
(Räumliche) Datenbank die über einen Zeitraum (häufig eine Historie von) räumliche(n) Zustandsänderungen verwaltet, d.h. räumlich-zeitliche Objekte (ST-Objekte) verwaltet.

- Hauptanwendung

- Verwaltung der Änderung des räumlichen Zustandes von (räumlichen) Objekten dessen räumlicher Zustand sich mit der Zeit verändert, z.B. Bewegungen (Trajektorien) von Tieren die über einen Tag beobachtet wurden.
- Unterstützung von räumlichen Anfragen zu einem oder mehreren bestimmten Zeitpunkten := **Snapshot-Anfragen (Snapshot Queries)**
- Unterstützung von räumlichen Anfragen über einen oder mehreren Zeitabschnitten := **Räumlich-zeitliche Anfragen (Spatio-Temporal Queries)**



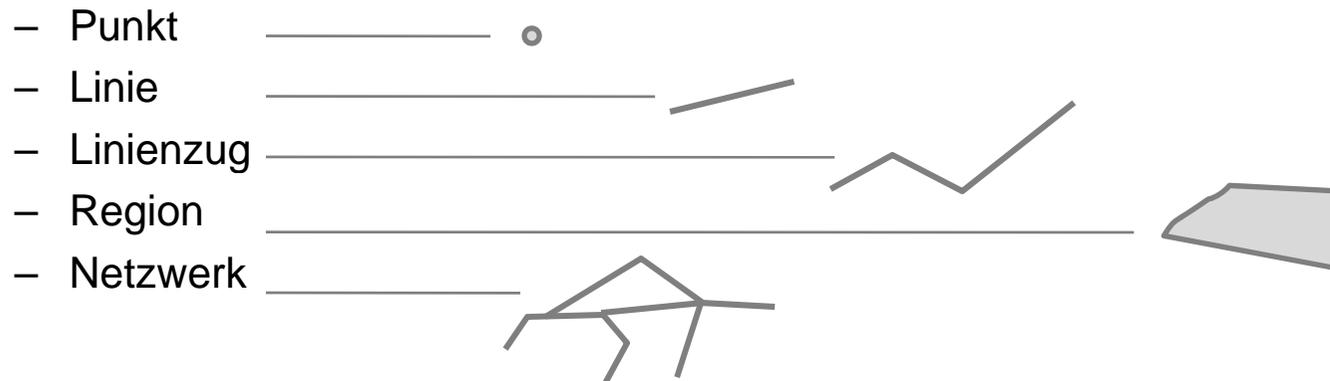
Snapshot-Anfragen



Räumlich-Zeitliche Anfragen

– Typen von räumlich-zeitlichen Objekten (ST-Objekten):

- Welche Arten von ST-Objekten werden in einer STDB verwaltet?



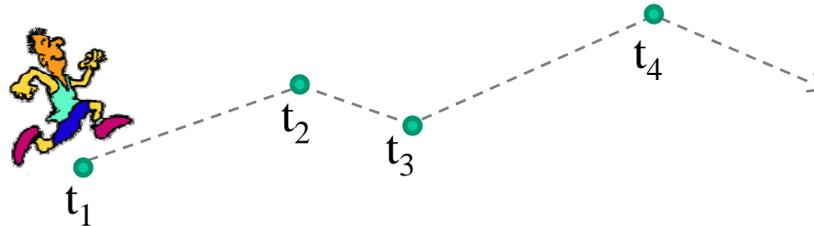
- Welche Arten von räumlichen Zustandsänderungen unterscheidet man?

- **Diskrete** Veränderung des räumlichen Zustands:
Beschreibung von Handlungen einer Person die an unterschiedlichen Orten statt finden, wie z.B. Geld abheben, einkaufen, Essen, etc.
- **Kontinuierliche** Veränderung des räumlichen Zustands:
Beschreibung der Bewegung einer Person, wie z.B. die Autofahrt von der Arbeit nach Hause.

- Wichtigste Abstraktionen von ST-Objekten

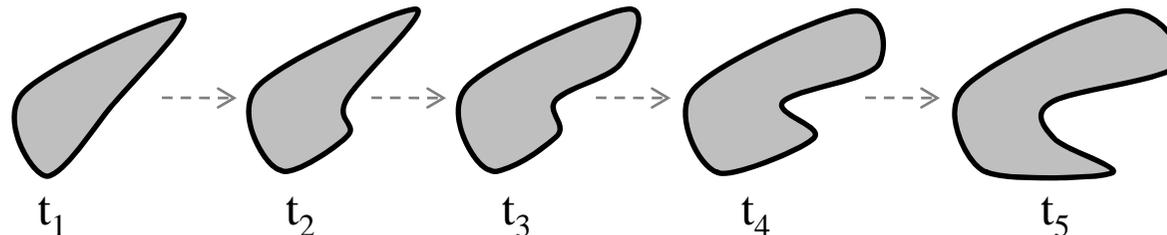
- **Verändernde Position** (z.B. Auto)

Abstrahierung eines „real“ existierenden Objektes (können auch Merkmale eines realen Vorgangs sein) das sich in einem mehr-dimensionalen Raum bewegt und lediglich dessen **Position** im Raum für Anfragen relevant ist.



- **Verändernde Geometrie** (z.B. Fläche eines Waldgebietes)

Abstraktion eines räumlich ausgedehnten Objektes in einem mehr-dimensionalen Raum dessen **Position**, **Ausdehnung** und **Form** zeitlich variiert.



3.1.2 Anwendungen

– Anfragen von ST-Objekten in modernen Informationssysteme [Quelle: S99]

- Geographic Information Systems (GIS)
 - “Welche Tankstellen befinden sich in der Nähe meines Autos?”
 - “Welches Taxi befindet sich am nächsten am Kunden?”
- Environmental Information Systems
 - “Wie haben sich die polaren Eiskappen in den letzten Jahren verändert?”
 - “In welchem Gebiet hält sich der Luchs normalerweise auf?”
- Multimedia Information Systems
 - Welche Schauspieler treten in mehr als einer Szene gemeinsam auf?
 - Wie bewegt sich Schauspieler X zur Zeit t in Video V?

– Weitere Anwendungen für ST-Objekte [Quelle: EGSV99]

Anwendungen für bewegliche Punkte	Anwendungen für veränderliche Flächen
<p>Personen:</p> <ul style="list-style-type: none"> • Bewegung von Terroristen/Kriminellen <p>Tiere:</p> <ul style="list-style-type: none"> • Trajektorien von Walen/Luchsen ermitteln • Welche Wege legen sie zurück? Wie schnell? Wie oft pausieren sie? • Wo befinden sich die Wale gerade? • Hat sich ihr Aufenthaltsbereich in den letzten 20 Jahren verändert? <p>Satelliten, Raumschiffe, Planeten:</p> <ul style="list-style-type: none"> • Welche Satelliten werden sich dem Raumschiff in den nächsten 4 Stunden nähern? <p>Autos:</p> <ul style="list-style-type: none"> • Taxis: Welches ist am nächsten zum Kunden? • Lastwagen: welche Routen werden regulär befahren? <p>Flugzeuge:</p> <ul style="list-style-type: none"> • Gab es beinahe-Kollisionen zwischen Flugzeugen? • Fliegen zwei Flugzeuge gerade aufeinander zu? • Sind Flugzeuge über ein bestimmtes Staatsgebiet geflogen? • Wie schnell fliegt ein Flugzeug? Maximalgeschwindigkeit? <p>Schiffe:</p> <ul style="list-style-type: none"> • Drohen Schiffe auf Grund zu laufen? • Sonderbare Routen von Schiffen entdecken (Illegale Machenschaften) • Militärische Analysen 	<p>Länder</p> <ul style="list-style-type: none"> • Was war die maximale Fläche des römischen Reiches? • Wann wurden Staaten vereinigt? • Welche Staaten wurden geteilt? <p>Wälder, Seen</p> <ul style="list-style-type: none"> • Wie schnell schrumpft der Regenwald am Amazonas? • Schrumpft das tote Meer? • Minimale und maximale Ausdehnung von Flüssen über das Jahr? <p>Gletscher</p> <ul style="list-style-type: none"> • Schrumpft der polare Eisschild? Verändert er seine Position? • Rückprojektion: Wie groß war Gletscher X zur Zeit t? <p>Hoch- und Tiefdruckgebiete</p> <ul style="list-style-type: none"> • Wohin bewegen sie sich? • Veränderung von Äquidensiten über die Zeit? <p>Armeen, Truppen:</p> <ul style="list-style-type: none"> • Trajektorie von Hannibals Truppen? <p>Krebs:</p> <ul style="list-style-type: none"> • Wie hat sich der Tumor über die letzten Jahre ausgebreitet? <p>Kontinente:</p> <ul style="list-style-type: none"> • Wie haben sich die Kontinente im letzten Jahr verschoben?

3.1.3 Wissenschaftliche Herausforderungen im Kontext von STDBMS [S99]

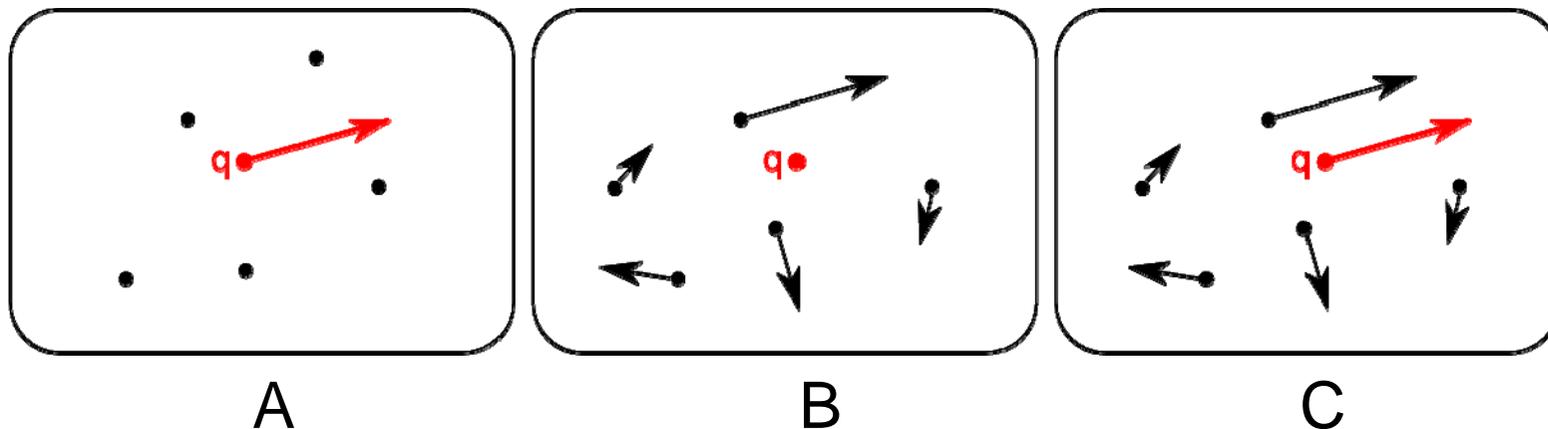
1. Ontologie, Struktur und Darstellung von Raum und Zeit
 - z.B. Unterscheidung zwischen Lebensdauer und Bewegung von Objekten
 2. Modelle und Sprachen für STDBMS
 - z.B. feste Integration des räumlich-zeitlichen Aspekts in das STDBMS
 3. GUIs für Spatio-Temporal Information
 4. Architektur eines STDMBS
 5. Speicherstrukturen und Indexierung
 - Erweiterung von Speicherstrukturen auf bewegliche Objekte, Benchmarks
 6. Anfragebearbeitung in Spatio-Temporal Databases
 - Anfragealgorithmen, Anfrageoptimierung
- In dieser Vorlesung beschäftigen wir uns primär mit den Problematiken Anfragebearbeitung und Indexierung (siehe 5. u. 6. oben)

3.1.4 Grundsätzliches zu Verwaltungs- und Anfragemethodiken

- Prinzipiell viele unterschiedliche Methoden, ST-Objekte effizient zu verwalten und anzufragen
- Wahl der Methode abhängig von der jeweiligen Anwendung
- Wichtigste Unterscheidungsmerkmale:
 - **Beweglichkeit** der Objekte:
Bewegen sich alle Objekte oder nur das Anfrageobjekt?
 - **Anfragezeitraum**:
Vergangenheit, Gegenwart oder Zukunft?
 - **Anfragetyp**:
Anfrage über einen Zeitpunkt, Zeitraum oder kontinuierliche Ausgabe des Resultats?

– Klassifikation nach Bewegung der Objekte

- **Query veränderlich (A):** „Alle Tankstellen, die sich in den nächsten 5 Minuten in der Nähe meines Autos befinden werden.“
- **DB-Objekte veränderlich (B):** „Alle Autos, die näher an meiner Tankstelle sind als an einer anderen Tankstelle.“
- **DB-Objekte und Query beweglich (C):** „Alle meinem Flugzeug benachbarten Flugzeuge.“



– Klassifikation nach Anfragezeitraum

- Querying the Past

- „*Welches Auto war zwischen t_1 und t_2 im Gebiet A?*“
- „*Welches Auto war zur Zeit t_1 nächster Nachbar von X?*“
- Historische Daten sind statisch, Updates nur im Sinne von Einfügungen
- Ggf. Approximation der Bewegung, z.B. linear oder polynomiell
- Beispiele: STR-Tree, TB-Tree, 3D R-Tree [MGA03]

- Querying the Presence

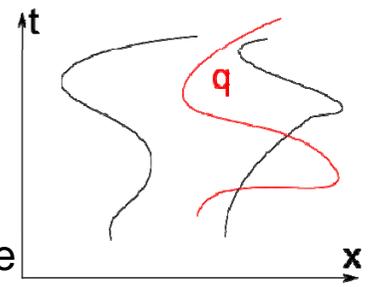
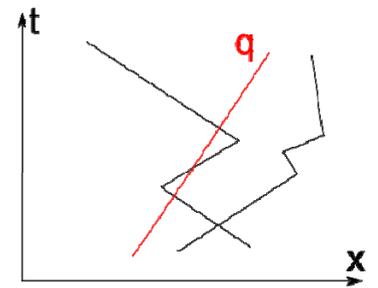
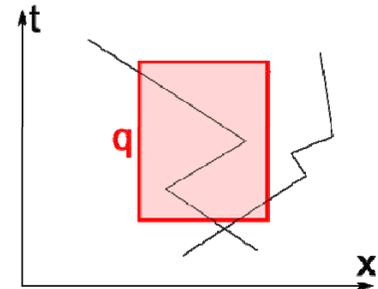
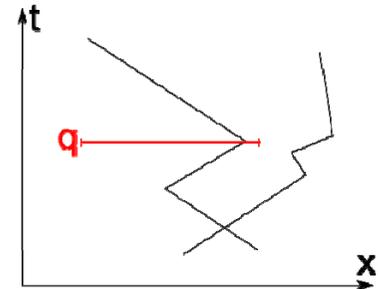
- „*Welches Auto ist aktuell im Einflussbereich meiner Tankstelle?*“
- „*Welche Autos sind aktuell im Gebiet A?*“
- Stark dynamische Daten
- Grid Indexing [YPK05], Hashing, LUR-Tree [MGA03]

- Querying the Future

- „*Welcher Tankstelle sind in 5 Minuten meiner Tankstelle am nächsten?*“
- Prädiktion notwendig, z. B. linear oder polynomiell
- TPR-Tree, TPR*-Tree, Future Quadtree [MGA03]

– Klassifikation nach Anfragetyp

- Snapshot- und Spatio-temporal Queries
 - Snapshot Queries beziehen sich auf einen Zeitpunkt
Beispiel: Range-Query zur Zeit t
 - Spatio-Temporal Queries beziehen sich auf Zeiträume
Beispiel: Range-Query für den Zeitraum $[t_0, t_1]$
- Continuous Queries
 - Query ist beweglich und kann deshalb durch Liniensegmente approximiert werden
 - Ergebnis wird in Zeitbereiche eingeteilt, während derer ein Ergebnis aktuell ist
 - Querying the Past sowie Querying the Future möglich
 - Modelle für die Approximation/Prädiktion von Trajektorien notwendig
- Monitoring Queries
 - Query wird initialisiert und soll dann über einen längeren Zeitraum eine Ergebnismenge aktuell halten
 - Oft Client/Server-basiert
 - Oft keine Annahmen bzgl. der Beweglichkeit der Objekte



3.2 Methoden für Snapshot- und ST-Anfragen

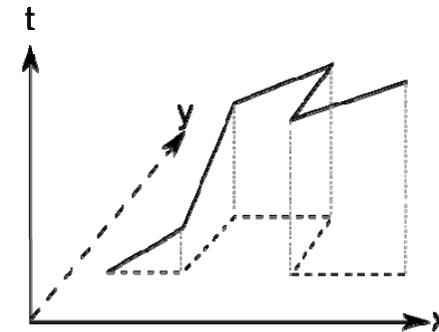
3.2.1 Allgemeines

- Snapshot Queries beziehen sich auf einen Zeitpunkt
 - Beispiel: „Alle Autos, die sich aktuell in München befinden“
- Spatio-Temporal Queries beziehen sich auf Zeiträume
 - Beispiel: „Alle Autos, die sich gestern irgendwann in München befanden“
- Hauptsächlich unterscheidet man zwischen
 - Anfragen auf die Vergangenheit
 - Anfragen auf die Gegenwart und (nahen) Zukunft

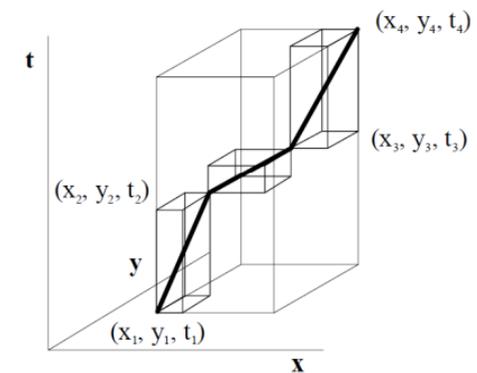
3.2.2 Effiziente Verwaltung von ST-Objekten (Überblick)

– Indexierung der Vergangenheit

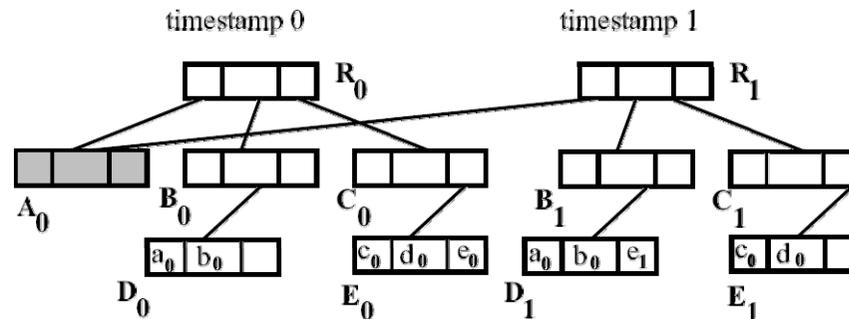
- Sei d die Anzahl räumlicher Dimensionen
- Darstellung der Vergangenheit
 - Zeit kann als zusätzliche Dimension betrachtet werden
 - Trajektorien als Linienzüge im $(d+1)$ -dimensionalen Raum



- Einfache Erweiterung des R-Baums
 - Zeitkomponente stellt zusätzliche Dimension dar, d.h. Verwaltung von $(d+1)$ -dim. Liniensegmenten.
 - z.B. 3D-R-Tree [MGA03], STR-Tree und TB-tree [PJT00]

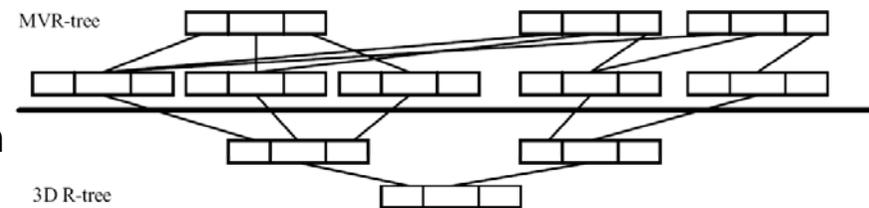


- Multi-Version Index [MGA03]
 - Separater Baum (z.B. R-tree) für jeden Zeitpunkt t.
 - Zusätzliche Verlinkung von Indexknoten die sich über zwei oder mehrere Zeitpunkte nicht verändern.
 - z.B. MR-tree [XHL90], HR-tree [NS98] und MVR-tree [TP01]



- Kombinierte Ansätze

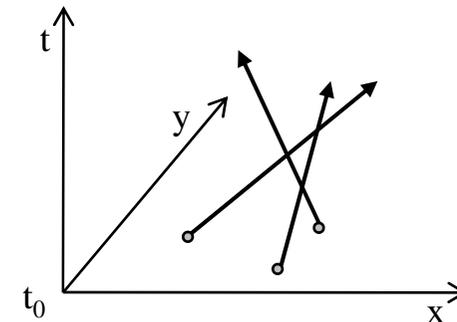
- Kombination aus 3D-R-Tree-Varianten und MVR-tree
- MVR-tree für Snapshot-Anfragen
- 3D-R-tree für ST-Anfragen
- Beide Bäume haben gemeinsame Blätter
- z.B. MV3R-tree [TP01]



– Indexierung der Gegenwart und nahen Zukunft

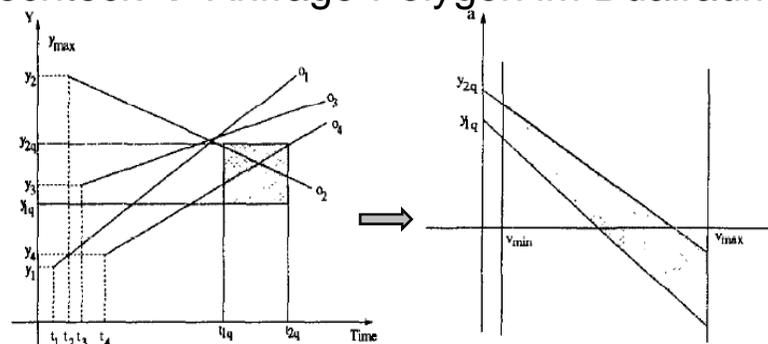
- Darstellung der Gegenwart und nahen Zukunft

- Zeit kann als zusätzliche Dimension betrachtet werden
- Zukünftige Bewegung über aktuelle Position (Position zum Zeitpunkt t_0) und (linearer) Bewegungsrichtung (gültig bis zu einem bestimmten Zeithorizont oder nächsten Pos.-Update)

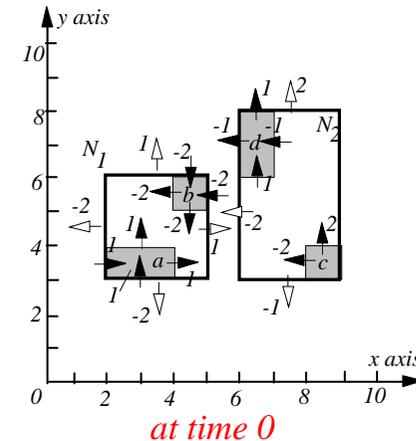


- Dual-Raum Transformation

- Transformation aller Trajektorien der Form $y(t) = vt + a$ in den Dual-Raum \Rightarrow Punkt (v, a) .
- Verwaltung über Punktzugriffsstruktur (z.B: R-Baum)
- Anfragerechteck \rightarrow Anfrage-Polygon im Dualraum



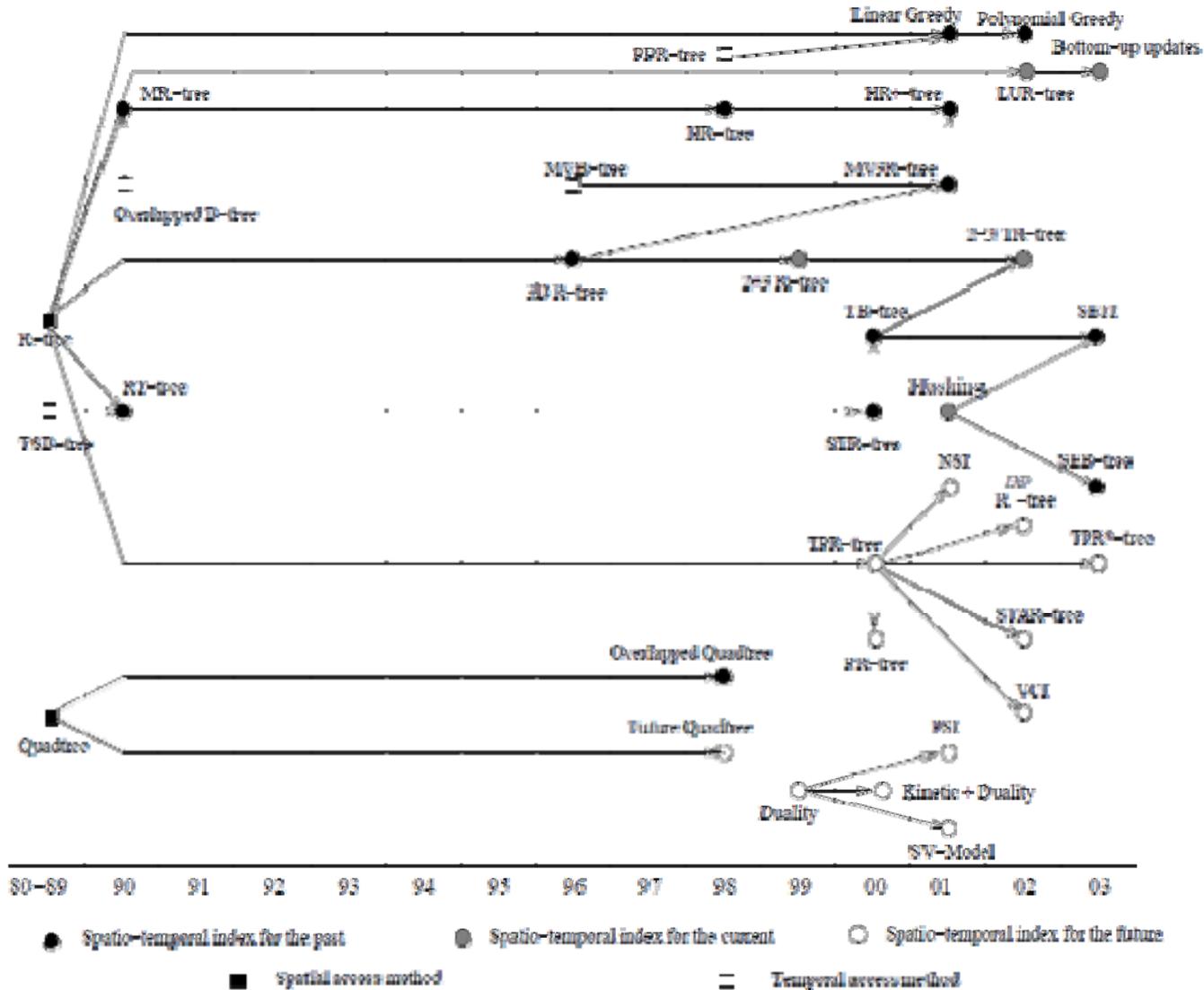
- Time-Parametrized R-Tree (TPR-tree)
 - Erweiterung der räumlichen Schlüssel eines R-Baums
 - Prinzip der zeit-parametrisierten Seitenregionen
 - z.B. TPR-tree [TPR00]



– Bemerkung:

- Überblick über weitere Varianten (siehe nächste Folie) in [MGA03] M. F. Mokbel, T. M. Ghanem, W. G. Aref. Spatio-temporal Access Methods. IEEE Data Engineering Bulletin, 2003. nachzulesen.
- Im Folgenden werden wir auf eine kleine Auswahl von Indexstrukturen etwas näher eingehen, insb. auf
 - Indexierung der Vergangenheit: Spatio-Temporal-R-Baum (STR-Tree) [PJT00]
 - Indexierung der Gegenwart und (nahen) Zukunft: Time-Parametrized R-tree (TPR-tree) [TPR00]

- Gesamt-Überblick über S-, T- und ST-Indexstrukturen



[Quelle: MGA03]

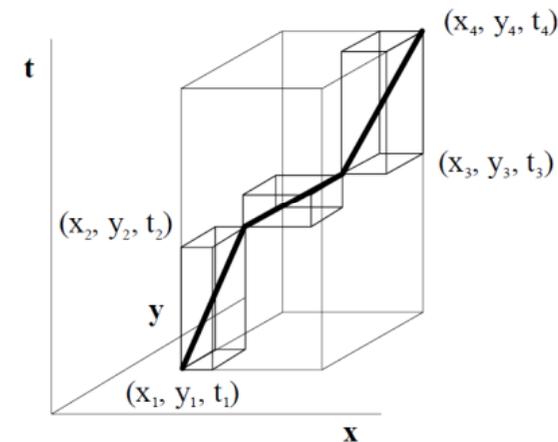
3.2.3 Verwaltung der Vergangenheit

– Idee:

- Erweiterung des R-Baums um eine weitere Dimension, die Zeit t
- Die Trajektorie eines Objektes kann als Sequenz von $(d+1)$ -dimensionalen Liniensegmenten repräsentiert werden
- Die einzelnen Liniensegmente werden in einem R-Baum abgelegt

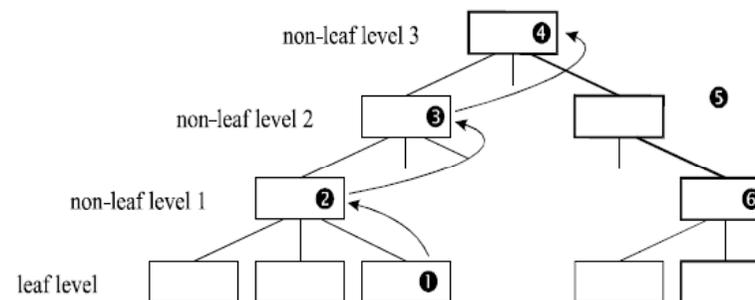
– Naive Indexierungs-Methode:

- Verwaltung der $(d+1)$ -dimensionalen Liniensegmente in einem herkömmlichen R-Baum (R^* -Baum).
- Prinzip des 3D-R-Baums



- Probleme:
 - Segmente der Trajektorie eines Objektes werden willkürlich über mehrere Seiten verstreut
 - Problem für ST-Anfragen über einen größeren Zeitbereich
 - Degenerierung der Seitenregion in der Zeitachse → zu wenig Information für einen Zeitpunkt
 - => Problem bei Snapshot-Anfragen
- Der Spatio-Temporal-R-Baum (STR-Baum) [PJT00]
 - Struktur analog zum R-Baum (Erweiterung um die Zeit-Dimension)
 - Unterschied zum herkömmlichen R-Baum (Naive Methode): Einfüge und Split-Heuristik
 - Ziele:
 - Speichere zusammenhängende Trajektoriensegmente möglichst nahe beieinander ab
 - Anfragen über längere Zeiträume
 - Trajektorien-basiert Anfragen, wie z.B. "Wie hat sich Klaus in der letzten Stunde bewegt?"

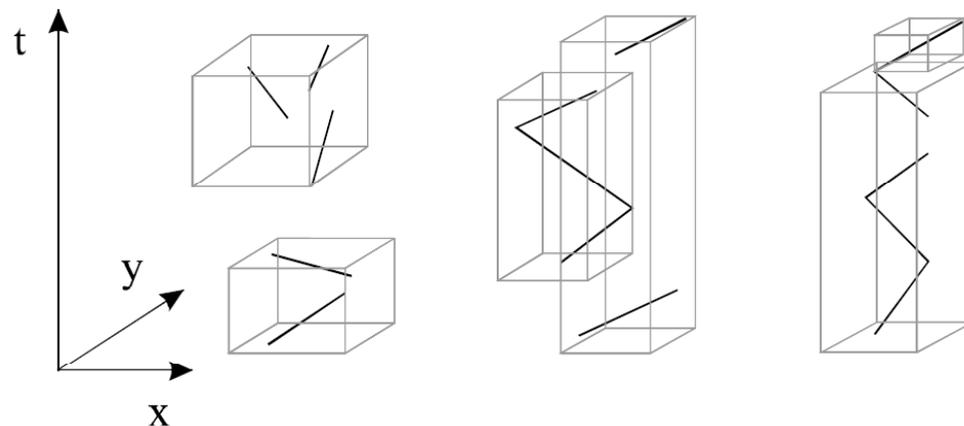
- Gruppierung von räumlich nahen Segmenten unterschiedlicher Trajektorien soll nicht vernachlässigt werden → Anfragen über kurze Zeiträume
=> Trade-Off zwischen beiden Zielrichtungen
- Einfüge-Strategie
 - Objekte werden nicht nur aufgrund räumlicher Nähe, sondern anhand ihrer Zugehörigkeit zu einer Trajektorie in den R-Baum eingefügt
 - Suche das Vorgängersegment des einzufügenden Segmentes
 - Knoten ist nicht voll -> Füge neues Segment in den Knoten ein
 - Knoten ist bereits voll:
 - » Ist einer der Eltern des Knotens in den $p-1$ darüber liegenden Hierarchieebenen nicht voll, dann splitte das Blatt (Parameter p kann frei gewählt werden und bestimmt in welchem Nachbarschaftsradius Segmente der gleichen Trajektorie abgespeichert werden)



- » Sonst: Wähle das Einfüge-Blatt anhand der R-Baum-Einfüge-Strategie

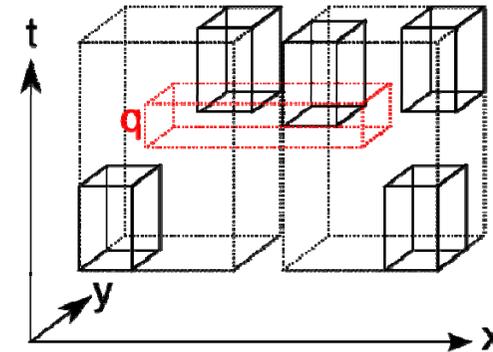
- Split-Strategie

- Idee: Füge immer die neuesten Segmente in den neuen Knoten ein, denn für sie ist es am wahrscheinlichsten, dass ein neues Segment eingefügt wird.
- Leaf Node:
 - » Alle Segmente im Blatt sind nicht miteinander verbunden: Verwende Quadratischen Algorithmus des R-Baumes
 - » Nicht alle aber mindestens ein Segment sind mit keinem anderen Segment verbunden: Nicht verbundene Segmente werden in den neuen Knoten eingefügt
 - » Alle Segmente sind mit einem anderen verbunden: Suche das Segment, das einen zeitlich älteren Nachbarn im gleichen Knoten hat, und füge es in den neuen Knoten ein.
- Non-Leaf Node:
 - » Erstelle neuen Eintrag für den neuen Knoten



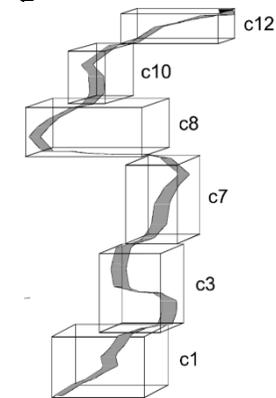
- Window-Query auf den STR-Baum

- Anfrage „Liefere mir alle Objekte, die sich innerhalb des Zeitraums (t_1, t_2) im Fenster (x_1, y_1, x_2, y_2) befanden
- Prinzipiell wie eine Fensteranfrage auf den R-Baum
- Anfrage liefert alle Segmente, die den Anfragequader schneiden.
- Segmente werden auf Objekte abgebildet
- Falls notwendig liefert der Schnitt zwischen Segmenten und Anfragequader die Trajektorien zwischen t_1 und t_2 .



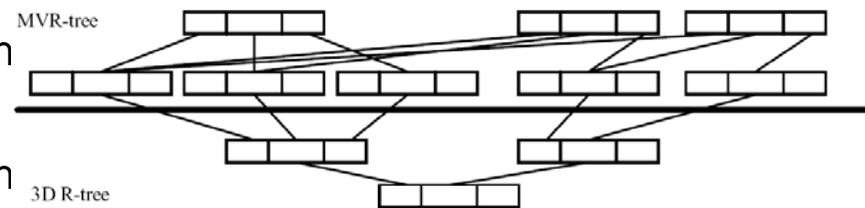
- Variante: TB-Baum [PJT00]

- Jedes Blatt enthält nur Segmente, die zur gleichen Trajektorie gehören
- Neue Segmente werden immer in den Blattknoten eingefügt, der das letzte Segment der Trajektorie enthält
- Angepasste Insert- und Splitstrategien
- **Nachteil:** Nahe Segmente, die zu unterschiedlichen Trajektorien gehören, werden in unterschiedlichen Knoten abgelegt => Spatial Range Queries (Snapshot) teuer



– Diskussion

- 3D-R-Baum und Varianten (STR-tree, TB-tree)
 - Funktionieren gut für Anfragen über einen größeren Zeitraum
 - Schlechtes Anfrageverhalten bei Snapshot-Anfragen
- Multi-Versions R-Bäume (HR und MVR)
 - Funktionieren gut für Anfragen zu bestimmten Zeitpunkten (Snapshot-Anfragen)
 - Anfragebearbeitung degeneriert bei Anfragen über einen größeren Zeitraum (ST-Anfragen)
- Hybride Ansätze wie der MV3R-Baum [TP01]
 - Koppelt zwei Indexstrukturen, den 3D-R-Baum und den MVR-Baum miteinander
 - Vereint die Stärken der beiden oben genannten Ansätze
 - Guter Kompromiss aus beiden Welten
 - Snapshot-Anfragen: Suche im jeweiligen MVR-Baum
 - ST-Anfragen: Suche im 3D-R-Baum falls Anfragezeitraum einen Benutzerdefinierten Grenzwert überschreitet.



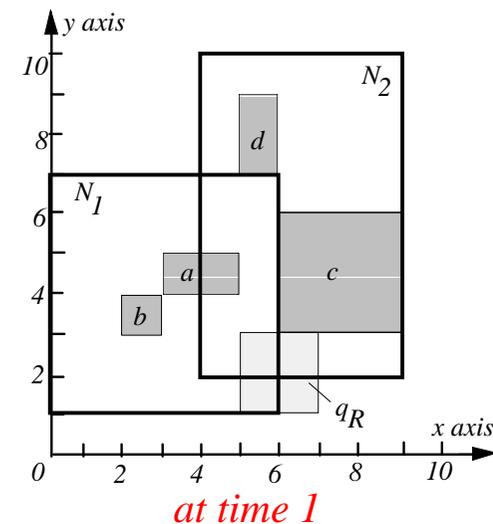
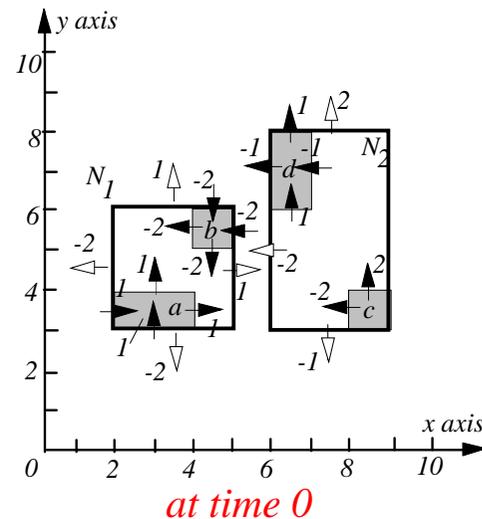
3.2.4 Verwaltung der Gegenwart und Zukunft

– Der Time-Parametrized R-Baum (TPR-Baum) [TPR00]

- Verwaltet Objektbewegungen der nahen Zukunft
- Konstruiert für Anfragen in der nahen Zukunft
- Erlaubt keine Anfragen in die Vergangenheit !
- Annahme: Bewegungen sind linear (bis zu einem best. Zeithorizont)
- Varianten:
 - STP-Baum (TPR-Baum-Variante für nichtlineare Bewegungen) [TFPL04]
- Struktur:
 - R-Baum-Variante, d.h. Prinzip der überlappenden Seitenregionen
 - Dimensionalität des indexierten Raums äquivalent zur Dimensionalität des räumlichen Attributs
 - Prinzip der zeitparametrisierten Seitenregionen
 - » Seitenregionen speichern neben der Ausdehnung zur Zeit t_0 einen Vektor über die (lineare) Veränderung über die Zeit
 - » Koordinaten der Seitenregionen sind Funktionen der Zeit

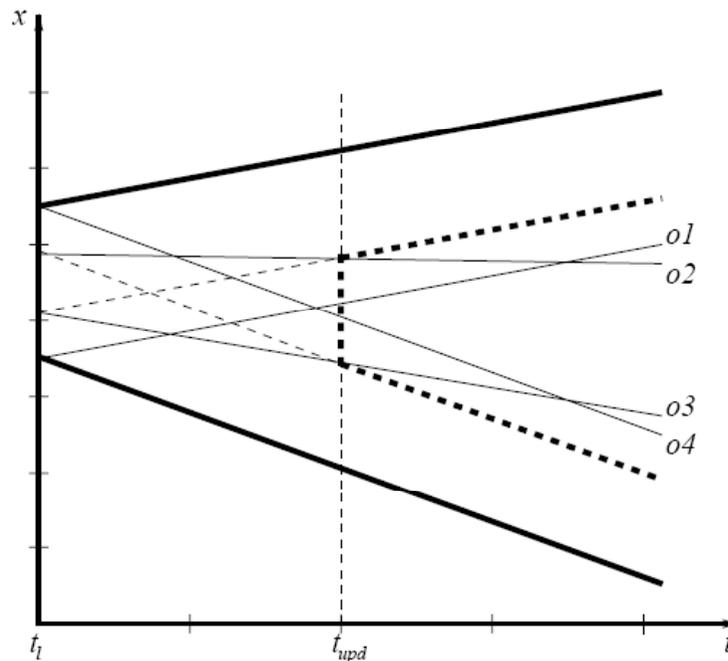
– Seitenregion eines Blattknotens

- » Achsenparalleles Rechteck, mit zeitparametrisierten Koordinaten $[(x^-_1(t), x^+_1(t)), (x^-_2(t), x^+_2(t)), \dots, (x^-_d(t), x^+_d(t))]$
wobei $(x^-_i(t), x^+_i(t)) = (x^-_i(t_0) + v^-_i(t-t_0), x^+_i(t_0) + v^+_i(t-t_0))$
- » v^-_i (v^+_i) ist die minimale (maximale) Geschwindigkeit der enthaltenen Objekte in der Dimension $1 \leq i \leq d$ (Vorzeichen der Geschwindigkeit bestimmt die Bewegungsrichtung, z.B. negative Geschwindigkeit -5m/s bedeutet Objekt bewegt sich nach links mit der Geschwindigkeit 5m/s)
- » Zum **Zeitpunkt** t_0 (Gegenwart) werden alle enthaltenen Objekte **minimal** umfasst.
- » Zum **Zeitpunkt** $t > t_0$ werden die Objekte zwar **konservativ** aber **nicht** mehr unbedingt **minimal** umfasst

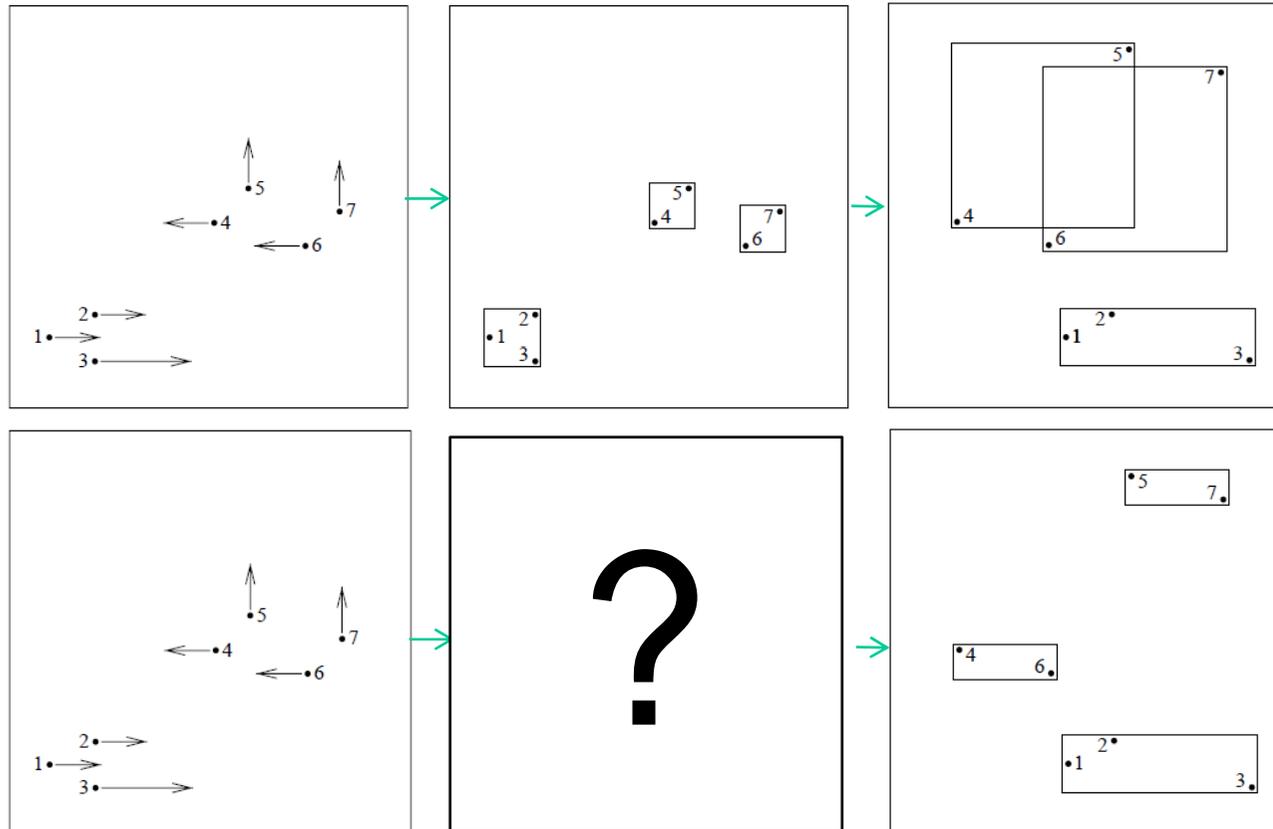


- Update-Strategie

- Seitenregionen wachsen kontinuierlich
=> kontinuierliche Degenerierung der Approximationsgüte
- Seitenregion ist solange gültig bis zu einem Zeitpunkt t_{upd} (Zeithorizont) zu dem es sich lohnt diese zu erneuern
=> Steigerung der Approximationsgüte
- Update der Seitenregionen zur Laufzeit möglich, z.B. bei Update eines Kindes



- Einfügestrategie
 - Welche Objekte sollen zusammen in einen Knoten eingefügt werden?
 - Räumliche Nähe nicht unbedingt optimal:



- Performanz des Baumes variiert mit dem Zeitraum H , für den Queries auf den Index gestellt werden
 - » $H=0$ -> Queryzeitpunkt entspricht Referenzzeitpunkt t_0 , damit kein negativer Einfluss veränderter Bounding Rectangles durch hohe Überschneidung
 - » $H \gg 0$ -> Gruppieren Punkte so, dass die Größe der Bounding Boxes zu allen Zeiten $[t_0, t_0+H]$ gering ist
 - Gute Performanz, wenn $v_{\min}-v_{\max}$ für jede Dimension gering ist (Bounding Box verändert dann ihre Größe nur geringfügig)
 - Minimiere Überschneidung, Umfang und Volumen wie bei der R^* -Baum Einfügestrategie, beachte aber dabei auch die zeitliche Komponente in $[t_0, t_0+H]$
-
- Unterstützte Anfragen:
 - Timeslice Query $q=(R,t)$ für einen festen Zeitpunkt t
 - Window Query $q=(R, (t_0,t_1))$ für ein Zeitintervall
 - Bewegliche Query $q=(R_1,R_2, (t_0, t_1))$ für das Trapezoid das durch die beiden Rechtecke R_1, R_2 sowie das Zeitintervall (t_0, t_1) aufgespannt wird

- Diskussion:
 - Nachdem der TPR-Baum Bewegungen vorhersagt sind die Ergebnisse einer Query immer approximativ.
 - Objekte im TPR-Baum können zur Laufzeit korrigiert werden
 - Eine Query kann also zu unterschiedlichen Zeiten unterschiedliche Ergebnisse liefern!
 - Queries weit in der Zukunft also wenig sinnvoll