

# Moving Objects – Bewegliche Objekte

---

Martin Tritschler, 06.07.2007

**Seminar:  
Mobile and Context-aware Database Technologies  
and Applications**

---



Einleitung  
Anforderungen  
Modellierung der Position  
Anfragen  
Ungenauigkeit  
Indexstrukturen  
Ausblick

- 1 Einleitung**
- 2 Anforderungen**
- 3 Modellierung der Position**
- 4 Anfragen**
- 5 Ungenauigkeit**
- 6 Indexstrukturen**
  - 6.1 B<sup>x</sup>-Baum und BB<sup>x</sup>-Index**
  - 6.2 LUGrid**
- 7 Ausblick**

## Einleitung

Anforderungen  
Modellierung der Position  
Anfragen  
Ungenauigkeit  
Indexstrukturen  
Ausblick

- **Informationen mit Ortsbezug sind die am meisten genutzten Kontextinformationen**
- **Bekannte Beispiele:**
  - Navigationssystem
  - Ortsabhängige Mobilfunkdienste
- **Datenbasis ist statisch**
  - Objekte haben feste Position
  - Wenige und nur diskrete Änderungen



## Einleitung

Anforderungen

Modellierung der Position

Anfragen

Ungenauigkeit

Indexstrukturen

Ausblick

- **Bewegliche Objekte (Moving Objects)**
  - Objekte, die ständig ihre Position ändern
  - Beispiele: Fahrzeuge, Flugzeuge, Personen, Tiere, Eisschollen, Tiefdruckgebiete etc.
- **Moving-Objects-Datenbanksystem (MOD-System)**
  - Datenbanksystem zur Verwaltung der Positionen beweglicher Objekte

## Einleitung

Anforderungen

Modellierung der Position

Anfragen

Ungenauigkeit

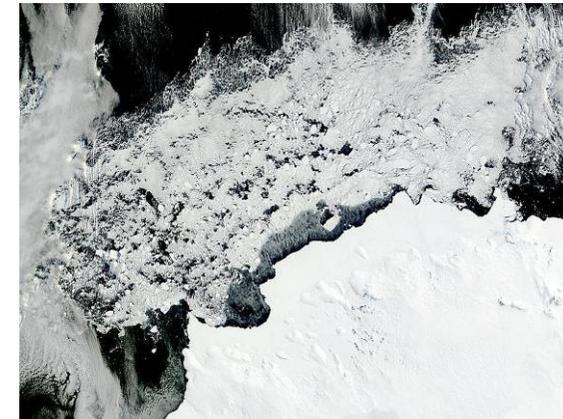
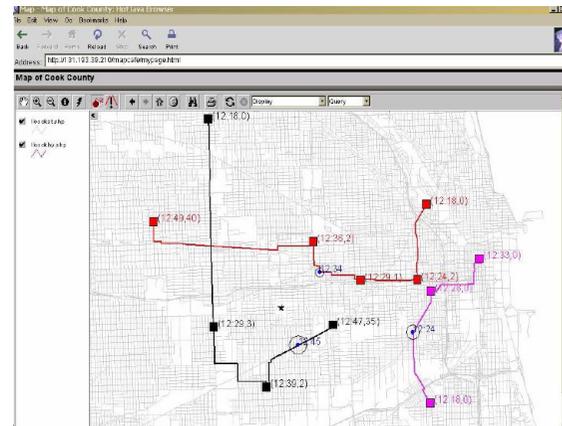
Indexstrukturen

Ausblick

- **Bewegliche Objekte ermitteln oft selbständig ihre Position (z.B. per GPS)**
- **Übermitteln diese über Funknetzwerk an MOD-System (z.B. per WLAN, GSM oder UMTS)**
- **Repräsentation als beweglicher Punkt**
- **Eisschollen? Tiefdruckgebiete?**

Einleitung  
Anforderungen  
Modellierung der Position  
Anfragen  
Ungenauigkeit  
Indexstrukturen  
Ausblick

- **„Location-Management-Perpektive“**  
Aktuelle und zukünftige Positionen



- **„Spatio-temporale Perspektive“**  
Vergangenheitsbezogener Verlauf von Positionen

# Anforderungen an MOD-Systeme

Einleitung  
**Anforderungen**  
Modellierung der Position  
Anfragen  
Ungenauigkeit  
Indexstrukturen  
Ausblick

- „Finde alle **Flugzeuge**, die in den **nächsten 20 Minuten** das **Bundesgebiet** überfliegen werden.“

## Zeitliche Komponente

- Zeitliche Anfrageprädikate
- Geeignete Sprachkonstrukte
- **Temporale Datenbanken**

## Räumliche Komponente

- Räumliche Datentypen
- Methoden für geometrische Anfragen (Abstände, Schnittpunkte etc)
- „Nearest-Neighbour“, „Range-Query“
- **Räumliche Datenbanken**

## Bewegliche Objekte

- Temporale und räumliche Datenbanken oft nicht für bewegliche Objekte geeignet

Einleitung  
**Anforderungen**  
Modellierung der Position  
Anfragen  
Ungenauigkeit  
Indexstrukturen  
Ausblick

- **Modellierung der Position**
  - Sich ständig ändernde Positionen fordern sehr häufige Aktualisierungen
  - Hohe Belastung der Datenbank und des Funknetzwerks
  
- **Anfragesprachen**
  - Anfragekonstrukte für zeitliche und räumliche Bedingungen benötigt

Einleitung  
**Anforderungen**  
Modellierung der Position  
Anfragen  
Ungenauigkeit  
Indexstrukturen  
Ausblick

- **Ungenauigkeit**
  - Quantifizierung der Ungenauigkeit
  - Kompromiss zwischen Ungenauigkeit und Anfragehäufigkeit
- **Indexstrukturen**
  - Effiziente Anfrageverarbeitung benötigt Indexstrukturen
  - Sehr häufige Aktualisierungen führen zu Problemen

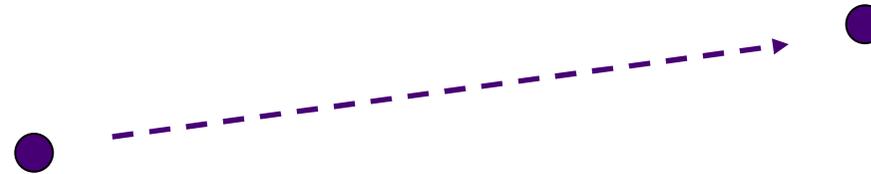
Einleitung  
Anforderungen  
**Modellierung der Position**  
Anfragen  
Ungenauigkeit  
Indexstrukturen  
Ausblick

- **„Database for Moving Objects“ (kurz DOMINO)**
  - Von Wolfson et al. 1998 vorgestellt
  - Beinhaltet das „**M**oving **O**bjects **S**patio-**T**emporal Model“ (kurz MOST)
- **Verzicht auf permanente Aktualisierungen**
  - Nicht die Position eines Objekts wird gespeichert, sondern dessen Bewegungsvektor
  - Kein eigener Datentyp, sondern „**dynamisches Attribut**“
  - Wert als lineare Funktion der Zeit

Einleitung  
Anforderungen  
**Modellierung der Position**  
Anfragen  
Ungenauigkeit  
Indexstrukturen  
Ausblick

- **Dynamisches Attribut hat drei Subattribute:**

- *updatevalue* hat Typ des Attributs
- *updatetime* ist Zeitpunkt der Aktualisierung
- *function(t)* ist lineare Funktion der Zeit



Zeitpunkt: *updatetime*  
Position: *x.updatevalue*  
*y.updatevalue*

Zeitpunkt: *updatetime + t*  
Position: *x.updatevalue + x.function(t)*  
*y.updatevalue + y.funktion(t)*

- Änderung des Werts im Zeitverlauf ohne explizite Aktualisierung

Einleitung  
Anforderungen  
**Modellierung der Position**  
Anfragen  
Ungenauigkeit  
Indexstrukturen  
Ausblick

- **Erweiterung des dyn. Attributs für Route**
  - Sechs Subattribute

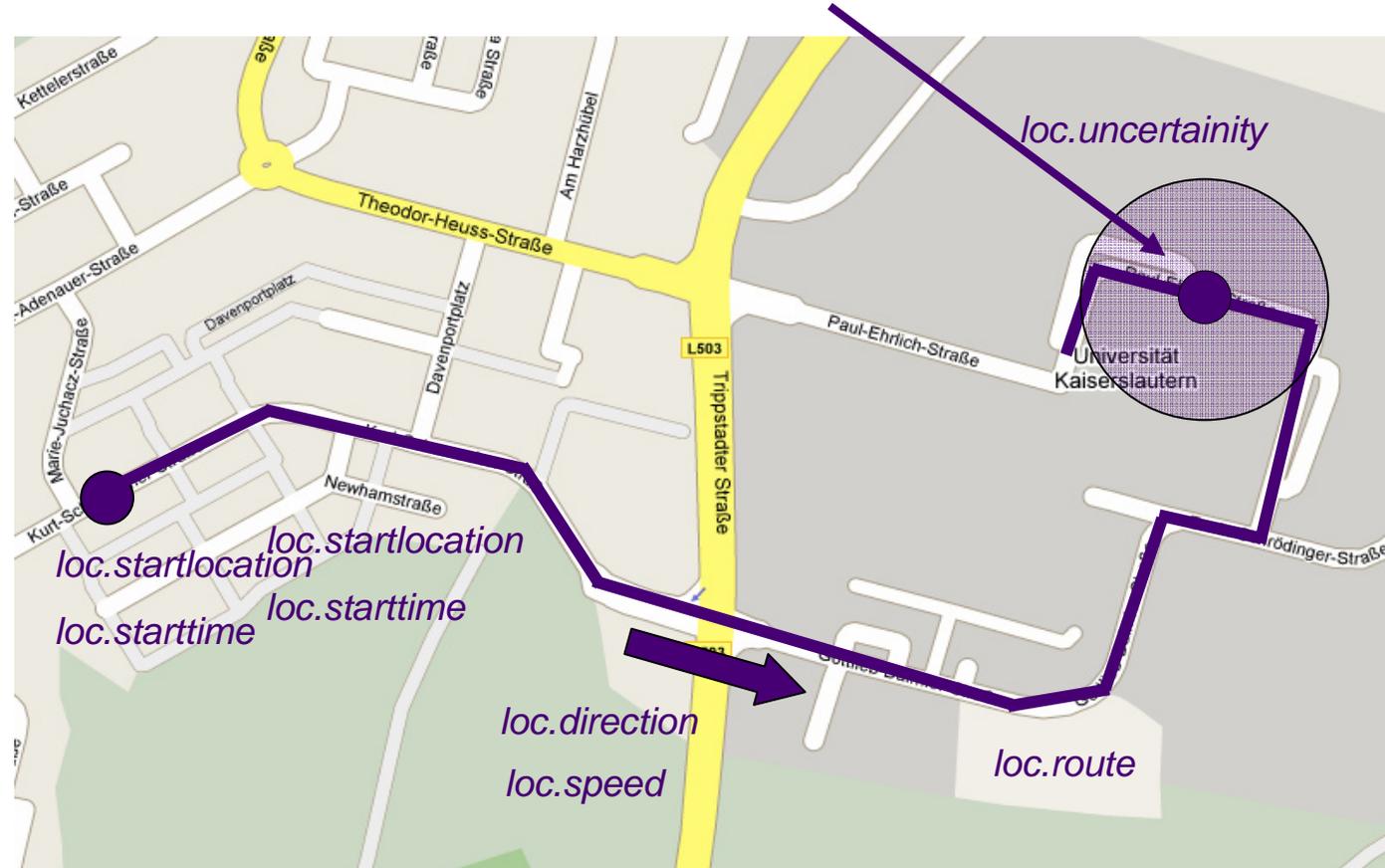
<i>loc.route:</i>	Route auf der sich Objekt bewegt
<i>loc.starttime:</i>	Zeitpunkt der Aktualisierung
<i>loc.startlocation:</i>	Position zum Zeitpunkt <i>starttime</i>
<i>loc.direction:</i>	Relative Richtung auf Route
<i>loc.speed:</i>	Bewegungsgeschwindigkeit
<i>loc.uncertainty:</i>	Unsicherheit

# Dynamisches Attribut

Einleitung  
Anforderungen  
**Modellierung der Position**  
Anfragen  
Ungenauigkeit  
Indexstrukturen  
Ausblick

Zeitpunkt:  $starttime + t$

Position auf Route:  $startlocation + speed * t$  (in Richtung *direction*)



Einleitung  
Anforderungen  
Modellierung der Position  
**Anfragen**  
Ungenauigkeit  
Indexstrukturen  
Ausblick

- **Anfragen im MOST-Modell**
  - Datenbankzustand und Datenbankhistorie
  - Anfrage ist Prädikat über Datenbankhistorie
  - Antwort ist Menge der erfüllenden Instanzen der Prädikatsvariablen
  
- **Anfragen lassen sich auf unterschiedliche Art stellen**

Einleitung  
Anforderungen  
Modellierung der Position  
**Anfragen**  
Ungenauigkeit  
Indexstrukturen  
Ausblick

- **Moment-Anfrage**
  - „Finde alle Hotels im Umkreis von 3km“
  - „Finde alle Hotels, die von meinem Standort in 20 Minuten weniger als 3km entfernt sind“
- **Kontinuierliche Anfragen**
  - Antwortmenge kann sich im Zeitverlauf dynamisch ändern
  - Einmalige Auswertung

Einleitung  
Anforderungen  
Modellierung der Position  
**Anfragen**  
Ungenauigkeit  
Indexstrukturen  
Ausblick

- **„Future Temporal Logic“ (kurz FTL)**
  - Temporale Anfragesprache
  - Im Rahmen des DOMINO-Projekts vorgestellt
  - Benötigt Datenbank-Management-System mit herkömmlicher nichttemporalen Anfragesprache (SQL), die um räumliche Prädikatsfunktionen erweitert ist
  - FTL-Sprachelemente als zeitliche Bedingungen in WHERE-Klausel

Einleitung  
Anforderungen  
Modellierung der Position  
**Anfragen**  
Ungenauigkeit  
Indexstrukturen  
Ausblick

- **FTL ist ähnlich der Prädikatenlogik erster Stufe**
- **Menge von Symbolen**
  - Variablen, n-stellige Funktionen, Konstanten, n-stellige Prädikate, Junktoren, Negation und Klammern.
  - Zuweisungsquantor  $\leftarrow$
  - Zeitbezogene Operatoren wie *until* und *nexttime*
- **Auf der Menge der Symbole werden induktiv Terme und Formeln definiert**

Einleitung  
Anforderungen  
Modellierung der Position  
**Anfragen**  
Ungenauigkeit  
Indexstrukturen  
Ausblick

- Seien  $s_a, s', s$  Datenbankzustände der Datenbankhistorie  $H$
- Sei dabei  $s_a$  der aktuelle Zustand der Datenbank
- Variablenbelegung  $\mu$  im Zustand  $s$  wird verkürzt mit  $(s, \mu)$  angegeben
- Seien  $F, G$  zwei Formeln,  $x$  eine Variable und  $t$  ein Term der FTL
- **$F$  until  $G$  ist erfüllt gdw.:**
  - Entweder ist  $G$  in  $(s_a, \mu)$  erfüllt oder es existiert ein zukünftiger Datenbankzustand  $s'$ , so dass gilt:  $G$  ist in  $(s', \mu)$  erfüllt und für alle Zustände  $s_i$  mit  $s_i < s'$  ist  $F$  in  $(s_i, \mu)$  erfüllt
- **$Nexttime F$  ist erfüllt gdw:**
  - $F$  ist in  $(s', \mu)$  erfüllt, wobei  $s'$  der Zustand ist, der sofort nach dem aktuellen Zustand  $s_a$  eintritt
- **$[x \leftarrow t](F)$  ist erfüllt gdw:**
  - $F$  ist in  $(s_a, \mu)$  erfüllt, wobei jedes Auftreten der Variable  $x$  durch den Term  $t$  ersetzt wird

Einleitung  
Anforderungen  
Modellierung der Position  
**Anfragen**  
Ungenauigkeit  
Indexstrukturen  
Ausblick

- Mit der gegebenen Semantik von *until* und *nexttime* lassen sich weitere zeitbezogene Operatoren definieren
- ***eventually F***
  - Bedeutet, dass *F* in einem beliebigen zukünftigen Zustand erfüllt sein wird. Wird *true* als das Boolesche „Wahr“ definiert, so gilt:
  - $\text{eventually } F \Leftrightarrow \text{true until } F$
- ***eventually\_within\_c F***
  - Bedeutet, dass *F* innerhalb von maximal *c* Zeiteinheiten erfüllt sein wird. Es gilt:
  - $\text{eventually\_within\_c } F \Leftrightarrow [t \leftarrow \text{Time}](\text{true until } (F \wedge (\text{Time} < t+c)))$

Einleitung  
Anforderungen  
Modellierung der Position  
**Anfragen**  
Ungenauigkeit  
Indexstrukturen  
Ausblick

- Funktion *route\_dist*: Länge des kürzesten Weges zwischen zwei Punkten in Routennetzwerk
- Prädikat *inside*: Überprüft, ob ein Punkt im Routennetzwerk in einem räumlich ausgedehnten Objekt ist.
- Tabelle *lastwagen*: Primärschlüssel *id*, dynamischem Attribut *loc*, räumlichem Attribut *ziel*
- Polygon *S*: Gebiet einer Stadt dar

- ```
SELECT id FROM lastwagen WHERE
    eventually_within_20 (dist(loc, ziel)=0)
```
- ```
SELECT id, loc FROM lastwagen WHERE
    eventually_within_20 (
        inside(loc, S)
        ^ dist(loc, ziel) < 2000
    )
```

Einleitung  
Anforderungen  
Modellierung der Position  
Anfragen  
**Ungenauigkeit**  
Indexstrukturen  
Ausblick

- **Aktualisierungen der Position**

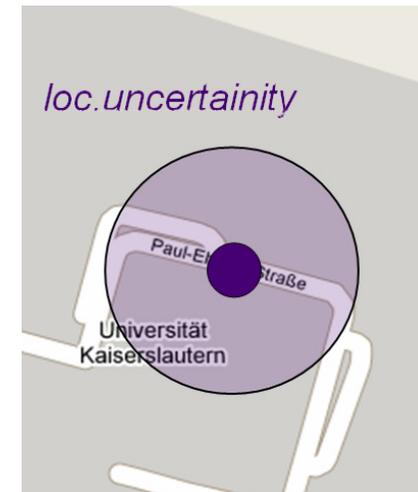
- Konstante Intervalle
- Wenn Ungenauigkeit bestimmtes Maß überschreitet

- **Zwei Konzepte**

- Abweichung
- Ungenauigkeit

- **Kostenmodell**

- Dritte Komponente ist Aktualisierungsrate
- Kostenminimaler Aktualisierungszeitpunkt bestimmbar

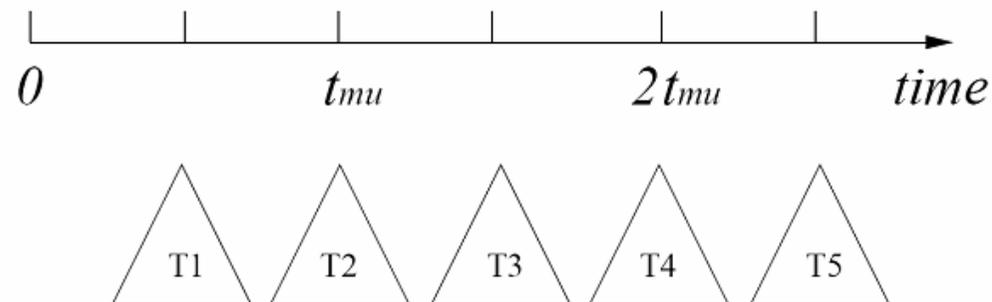


Einleitung  
Anforderungen  
Modellierung der Position  
Anfragen  
Ungenauigkeit  
**Indexstrukturen**  
Ausblick

- **Zur effizienten Anfrageausführung wird Indexierung der Position benötigt**
- **Indexstrukturen für bewegliche Objekte:**
  - R-Tree: HR-Tree, TPR-Tree, MV3R-Tree, R<sup>PPF</sup>-Tree
  - Quad-Tree: STRIPES
  - B-Tree: **B<sup>x</sup>-Tree** und **BB<sup>x</sup>-Index**
  - Grid-File: **LUGrid**

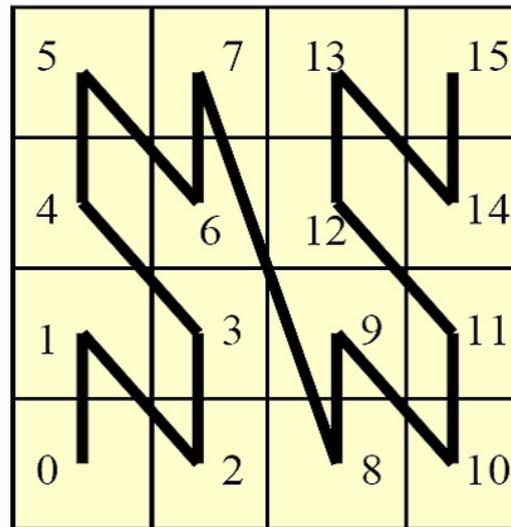
Einleitung  
Anforderungen  
Modellierung der Position  
Anfragen  
Ungenauigkeit  
**Indexstrukturen**  
Ausblick

- „**Broad B<sup>x</sup> Index Technology**“ (kurz **BB<sup>x</sup>-Index**)
  - Von Lin et al. 2005 vorgestellt
  - B-Baum-basierte Indexstruktur, die auf dem von Lin et al. 2004 entwickelten **B<sup>x</sup>-Baum** aufbaut
  - Ermöglicht vergangene, aktuelle und zukünftige Positionen beweglicher Objekte zu indexieren

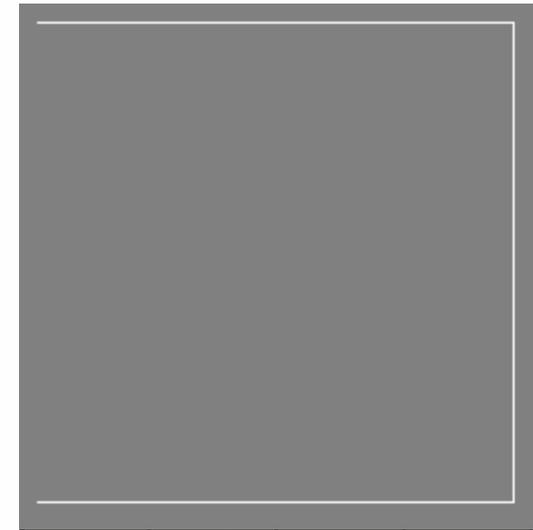


Einleitung  
Anforderungen  
Modellierung der Position  
Anfragen  
Ungenauigkeit  
**Indexstrukturen**  
Ausblick

- **Raumfüllende Kurve (FASS-Kurve)**
  - FASS: „space-filling, self-avoiding, simple und self-similar“



Peano-Kurve



Hilbert-Kurve

Einleitung  
Anforderungen  
Modellierung der Position  
Anfragen  
Ungenauigkeit  
**Indexstrukturen**  
Ausblick

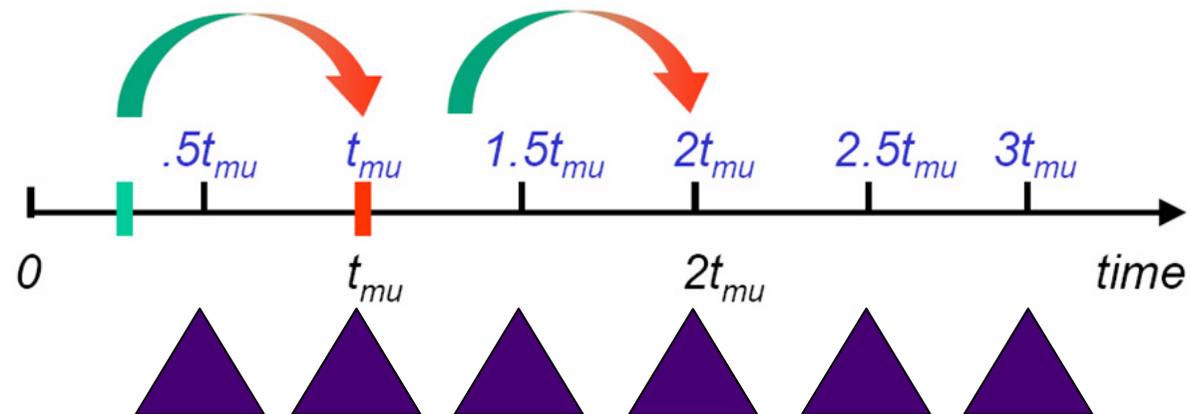
- **Problem der hohen Aktualisierungsrate**
  - Ähnlich wie beim MOST-Datenmodell wird lineare Bewegungsfunktion herangezogen, um Position zu bestimmen
- **Partitionierung der Zeitachse**
  - Länge der **Intervalle** ist maximale Zeit  $\Delta t_{mu}$  zwischen zwei Aktualisierungen einer Objektposition
  - Weitere Unterteilung der Intervalle in  $n$  gleich lange **Phasen**
  - Jeder Phase ist ein **Zeitstempel** zugeordnet, der den Endzeitpunkt angibt

Einleitung  
Anforderungen  
Modellierung der Position  
Anfragen  
Ungenauigkeit  
**Indexstrukturen**  
Ausblick

- **Einfügen einer Aktualisierung**

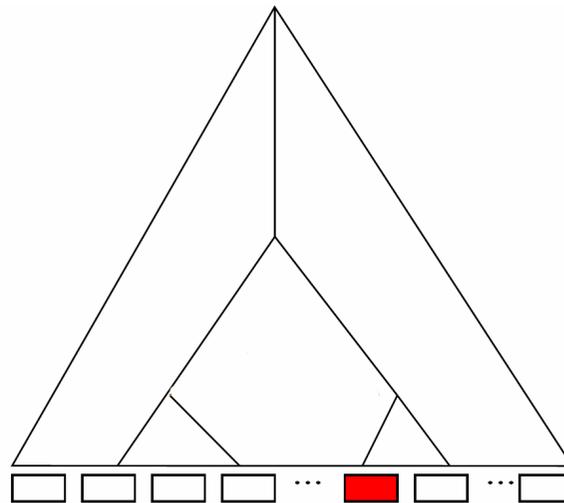
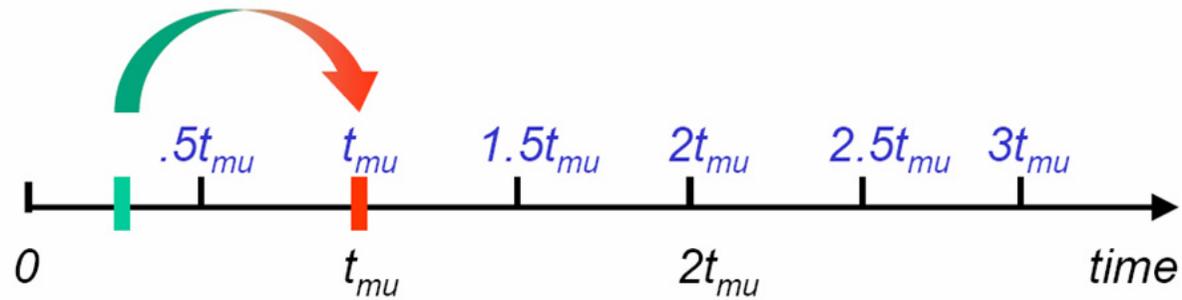
- Indexierung zum Phasenendzeitpunkt  $t_{ts}$ , der zeitlich nach der Aktualisierungszeit  $t_u$  liegt
- Berechnung:

$$t_{ts} = \lceil t_u + \Delta t_{mu} / n \rceil$$

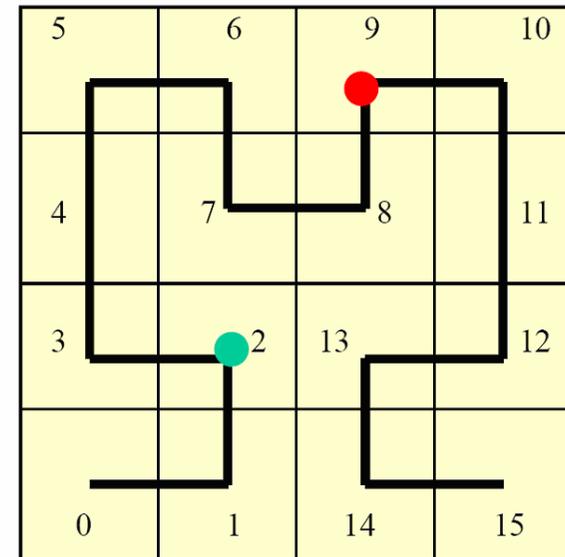


# BB<sup>x</sup>-Index

Einleitung  
Anforderungen  
Modellierung der Position  
Anfragen  
Ungenauigkeit  
**Indexstrukturen**  
Ausblick



9

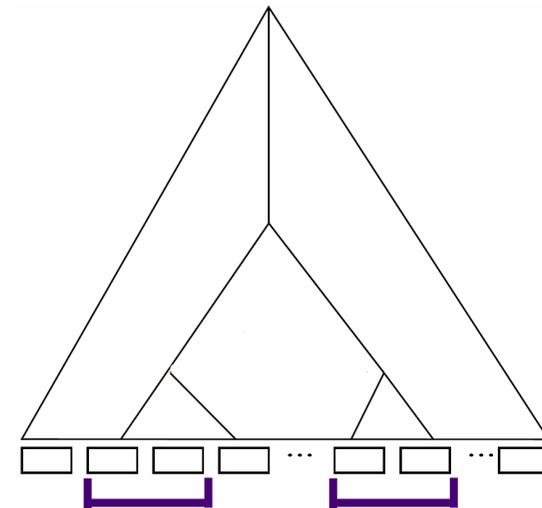
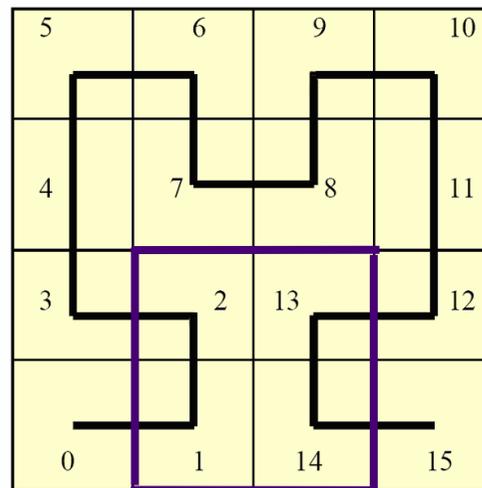


$$xrep = FASS(x_u + \vec{v}(t_{ts} - t_u))$$

Einleitung  
Anforderungen  
Modellierung der Position  
Anfragen  
Ungenauigkeit  
**Indexstrukturen**  
Ausblick

- **Anfragen**

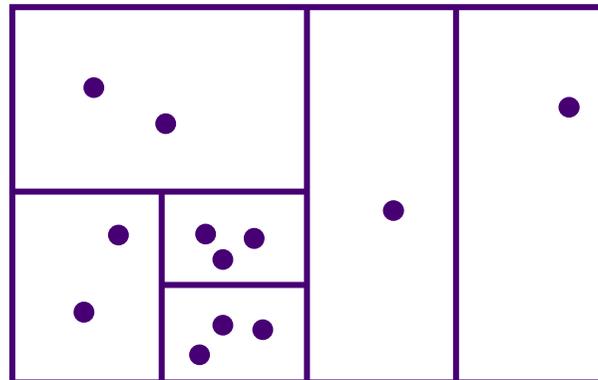
- Zeit bestimmt die Bäume
- Ort bestimmt Positionen im Baum



Intervalle [1,2], [13,14]

Einleitung  
Anforderungen  
Modellierung der Position  
Anfragen  
Ungenauigkeit  
**Indexstrukturen**  
Ausblick

- „**Lazy-Update Grid-based Index**“ (kurz LUGrid)
  - Von Xiong et al. 2006 vorgestellt
  - Nur aktuelle Positionen
  - Hohe Aktualisierungsraten bei stark verminderten Zugriffskosten im Datenbanksystem
  - Auf multidimensionaler Indexstruktur „**Grid File**“ basierend

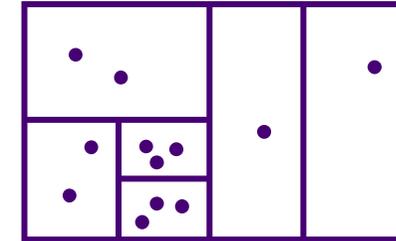


Einleitung  
Anforderungen  
Modellierung der Position  
Anfragen  
Ungenauigkeit  
**Indexstrukturen**  
Ausblick

- **„Lazy Insertion“**
  - Hauptspeicherbasierte Puffer-Schicht über eigentlicher Indexstruktur
  - An gleiche Seite in der Datenbank gerichtete Aktualisierungen werden zusammengefasst ausgeführt
- **„Lazy Deletion“**
  - Aktualisierte Werte werden eingefügt, ohne veraltete Werte zu löschen
  - Löschen wird hinausgezögert, bis betreffende Seite durch nachfolgende Operation ohnehin in den Hauptspeicher geladen wird

Einleitung  
Anforderungen  
Modellierung der Position  
Anfragen  
Ungenauigkeit  
**Indexstrukturen**  
Ausblick

- **„Disk Grid“ (DG)**
  - Im Permanentenspeicher
  - Nichtüberlappende DG-Zellen  
enthalten Objektpositionen

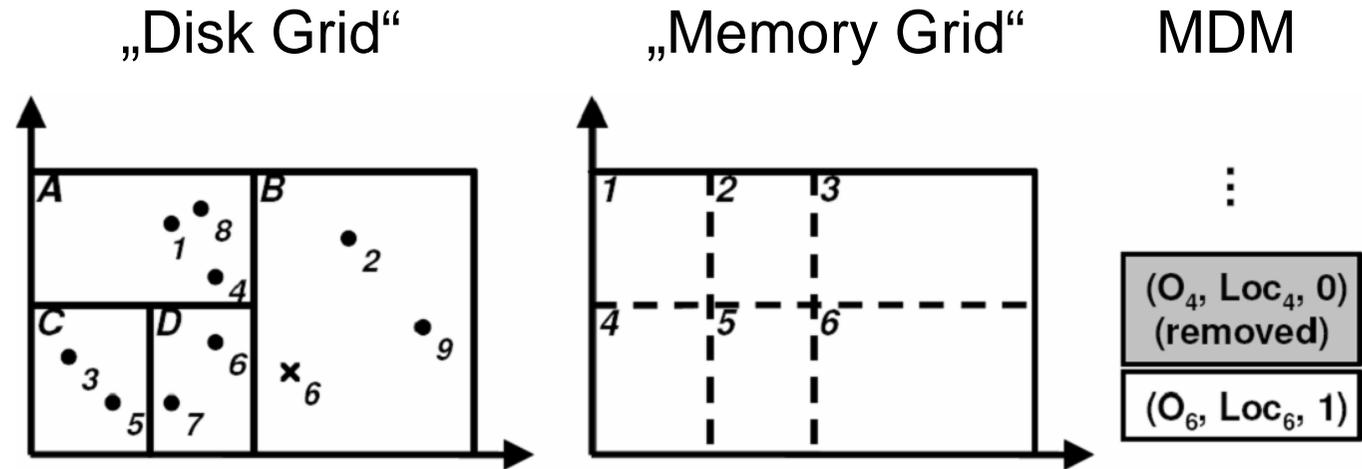


- **„Memory Grid“ (MG)**
  - Im Hauptspeicher liegender Puffer für DG
  - Zuordnung MG-Zelle : DG-Zelle = n : 1
- **„Miss-Deletion Memo“ (MDM)**
  - Hauptspeicherbasierte Hash-Struktur
  - Anzahl veralteter Versionen des Objekts



Einleitung  
 Anforderungen  
 Modellierung der Position  
 Anfragen  
 Ungenauigkeit  
**Indexstrukturen**  
 Ausblick

- **Beispiel**

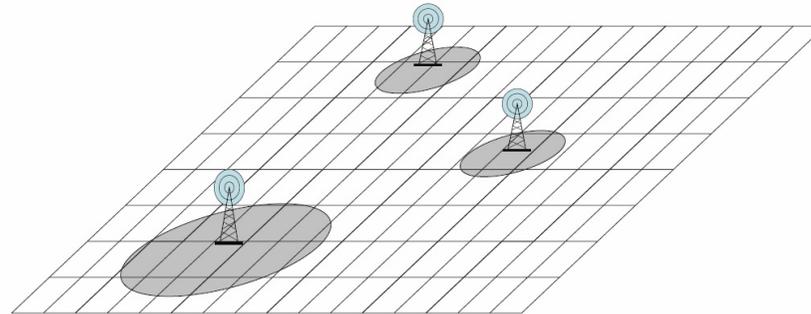


Einleitung  
Anforderungen  
Modellierung der Position  
Anfragen  
Ungenauigkeit  
**Indexstrukturen**  
Ausblick

- **Anfragen**
  1. Bestimmung der sich mit Anfragegebiet überlappenden Zellen
  2. Durchsuchen der MG-Zellen nach gepufferten Objekten
  3. Einlesen der DG-Zellen
  4. Durchsuchen der DG-Zellen nach aktuellen Werten unter Zuhilfenahme von MDM
- **Keine Anfragen mit Zeitbezug**

Einleitung  
Anforderungen  
Modellierung der Position  
Anfragen  
Ungenauigkeit  
Indexstrukturen  
**Ausblick**

- **Seite der beweglichen Objekte**
  - Bekommen oft kontextbezogene Informationen vom MOD-System gesendet
  - Bandbreite beschränkt oder Verlust der Verbindung



- **Hoarding**
  - Übermittlung von voraussichtlich in Zukunft genutzten Daten im Voraus

Einleitung  
Anforderungen  
Modellierung der Position  
Anfragen  
Ungenauigkeit  
Indexstrukturen  
**Ausblick**

- **Viele Veröffentlichungen in den letzten Jahren**
- **Technische Voraussetzungen gegeben**
- **Systeme werden teilweise schon eingesetzt**
  
- **Es bleibt abzuwarten, wann praxisbezogene Umsetzung im großen Maßstab eintritt und breite Masse an Endbenutzern das Potential der Technologie nutzen kann**

Vielen Dank



**Zeit für Fragen und  
Diskussion**