

## Kinetische Datenstrukturen:

s. Folien!

Problem: Objekte in geom. DS sind in Bewegung  $\rightarrow$   
wie kann man DS geschicht updaten?

Beob.:

- 1) Wenn sich ein Obj bewegt, ist die Änderung in der DS beschränkt, d.h., wenn man DS zum alten Zustand und zum neuen Zustand von Scratch aufbaut, so ist Unterschied meist nur klein.
- 2) Wenn sich alle Obj nur um ein  $\epsilon$  bewegen, so bleibt DS oft gültig.

Frage:

- 1) Wie stellt man fest, ob DS geändert werden muß?
- 2) Wie macht man diese Änderung geschicht?

Wunsch: Beides zusammen sollte schneller sein, als einfach DS von Scratch neu aufzubauen?

Lsg.: KDS (ist allg. Algorithmentechnik)

Annahmen:

- alle Obj<sup>(Pkte, Linien, etc)</sup> bewegen sich auf bestimmten Flugbahnen (falls über)
- Flugbahnen sind algebraische Elten (rationale Polynome und daraus zus. gesetzte Elten durch  $+ - * / \sqrt{\quad}$ )
- Flugbahnen dürfen sich ändern zu diskreten Zeiten (also nicht kontinuierlich); Änderung bringt natürlich Kosten mit sich.



Begriffe (zunächst noch abstrakt): sind i.A. alles Teile der KDS

o Attribut:

Das, was man mit der KDS starten und berechnen will;

z.B.: konvexe Hülle, dichtest Paar von Ecken, etc.;

oft ist Attribut ein Teil der Gesamt-KDS.

[Besch et al. ;  
de Berg et al.  
"in Kinetic CSP"]

o Kombinatorische Struktur:

"alles mögliche Koord. und andere konkrete Werte";

z.B.: Graph, Baum, Verzweigungen zwischen Entitäten der DS, ...;

ist Teil der KDS.

o Zertifikat (certificate):

Prädikat über eine Teilmenge aller geom. Objekte und

Teile der komb. Struktur;

z.B.:  $p \cdot n < 0$ ,  $p$  Ekt der Eingabe,  $n$  konkreter Wert

einer Ebene der komb. Struktur

o Beweis (proof):

Menge von Zertifikaten mit Eigenschaft:

solange alle Prädikate wahr bleiben, ändert sich komb.

Struktur nicht. (Gesucht: minimale vollständige Menge von Z.)

o Fehlschlagzeit (failure time, death time):

Zeit eines Prädikats (in der Zukunft), ab der es zum ersten

Mal ungültig wird (kann man aus Pfad aller Objekte

berechnen; läuft im Prinzip auf Bestimmung der

Nullstellen einer, beliebig komplexen, algebraischen Ekt. hinaus).

KDS bleibt korrekt/gültig bis zur nächsten F.zeit

eines Prädikats.



o Antwortzeit (response time):

Zeit (worst-case) zum Update der KDS, wenn ein  
Zeit. fehlschlägt.

A.zeit  $\in O(\log^k n) \rightarrow$  KDS heißt "responsive" (schnell reagierend)

o KDS heißt "kompakt", falls huz. Zertifikate nahe linear  
in Größe der Eingabe.

o "local", falls jedes Obj aus Eingabe nur in  
kleiner (polylog) huz. Zertifikate beteiligt.

(wichtig, wenn Flugbahn sich ändert)

o Event:

- extern := Fehlschlag eines Zertifikates und Attribut ändert sich

- intern := Fehlschlag, aber Attribut ändert sich nicht

Abkürzung: Begriff "externer Event" ist nur sinnvoll,  
wenn Attribut eindeutig!

(Für konvexe Hülle ggf., für BSP nicht)

o "effizient", falls huz. externer Events / huz. interner Events  
 $\in O(\log^k n)$  in worst-case.



Main Loop einer KDS:

init DS (ist häufig gleich statische DS)

init Zertifikate

berechne Fehlschlagzeit aller Zertifikate

init Priority-Queue mit allen F.zeiten

Loop forever

hole Zertifikat aus P-Queue mit frühester Zeit

lause KDS um (falls externer Event, erstellt dabei neues Attribut)

best. neue Zertifikate <sup>und</sup> oder lösche einige

berechne neue Fehlschlagzeit für "betroffene" Zertifikate

[ Canba:  
"Kinetic vertical  
descent trees",  
PhD, 1993 ]