**Motivation**

Motion is ubiquitous in the physical world and due to recent advances in sensing and tracking technology, motion data is becoming more and more available in a variety of areas: mobile communication, geographic information system, air-traffic control, and so on. It is not surprising, therefore, that it is necessary to store, analyze, and create or manipulate motion data. As a result, modeling moving objects has become an important area of study in many areas of computer science such as computation geometry, databases, graphics, wireless networks, ....

**Geometric study of moving objects**

Simulate system of continuously moving objects and efficiently maintain discrete geometric attributes of objects such as the convex hull of moving points.

**Two main approaches**

- Time sampling
- Kinetic data structures

---

**Kinetic Data Structures**

**Time Sampling Approach**

- Choose fixed time step.
- Update the positions of moving objects at each time step.
- Update the data structure with the new positions of objects.

**How to choose time step?**

- Oversampling:
  - Set of valid certificates
  - Certificate failure
  - Certificate update
  - Attribute update

- Undersampling

**Kinetic Data Structures**

- Combinatorial changes occur in irregular patterns.
- KDS consists of two parts
  - Combinatorial description of the attribute.
  - A set of certificates—elementary test on the input objects—with the property that as long as the outcome of the certificates do not change, the attribute does not change.

**Structures**

- Proof of correctness of attribute (certificates)
- Priority queue (event queue)

**Assumptions**

- A simple model for motion: each object follows a known flight plan with rational parameters.
- Certificates are algebraic; failure is next largest root.

---

**KDS Framework**

**KDS Properties**

- Compact: if it uses little space in addition to the input.
- Responsive: if data structure invariants can be restored quickly after the failure of a certificate.
- Local: if it can be updated easily if flight plan for an object changes.
- Efficient: if the worst-case number of events handled by the data structure is small compared to some worst case number of external events.

**Collision Detection**

Kinetic methods can be applied to collision detection problem which is a basic problem arising in all areas of geometric modeling involving objects in motion—motion planning, computer-simulated environments, ...

---

**Query Data Structures**

Kinetic method can be used to maintain a QDS in order to quickly answer queries involving objects in motion:

- What are the points currently inside a given region?
- What is currently nearest point to a given query point?

Rank-based kd-tree is efficient to answer above queries.