Kinetic Data Structures
Literature Review*

Alex Flores, Parker Given, Emmett Tran

May 3, 2017

References

[1] Julien Basch. *Kinetic Data Structures*. PhD thesis, Stanford, CA, USA, 1999. AAI9943622.

> This article describes in much deeper mathematical detail the main points of [2].

[2] Leonidas J. Guibas. Kinetic data structures: A state of the art report. In *Proceedings of the Third Workshop on the Algorithmic Foundations of Robotics on Robotics : The Algorithmic Perspective: The Algorithmic Perspective*, WAFR '98, pages 191–209, Natick, MA, USA, 1998. A. K. Peters, Ltd.

> This article introduces *kinetic data structures* (KDSs) as a theoretical framework for designing and implementing algorithms to efficiently track attributes of a geometric system that varies continuously over time It explains the why such simulations of geometric systems are useful, and then points out that representing those systems with continuous functions (rather than updating each component of the system discretely) means that the continuity of movement for the components of the system can be exploited for more efficient algorithms. The article then describes the different aspects of the KDS framework: *certificates*, *events*, and *motion models*. *Certificates* are conditions which must be maintained for a described attribute of a system to remain true, and the total set of certificates for a system are what form the proof of correctness for the attribute overall. *Events* are moments in time at which one or more certificates become invalid, meaning that the proof of correctness must be "repaired" (by modifying

*I couldn't get LaTeXto order the entries properly and also have annotations so unfortunately you will have to jump around the document a bit. Sorry :(

certificates). Since these events happen due to individual certificates becoming invalid, repairing the proof can be done efficiently by only computing what is necessary to update the invalid certificates. The basic *motion model* for a KDS, according to the article, is one which maintains a "flight plan" of each component of the geometric system (typically a polynomial function) and provides a mechanism for updating flight plans at any point in the simulation of the system. Finally, the article introduces metrics for calculating the efficiency of a KDS, and discusses the applications of KDSs for several computational geometry problems.

[3] Dinesh P. Mehta and Sartaj Sahni. *Handbook Of Data Structures And Applications (Chapman & Hall/Crc Computer and Information Science Series.)*. Chapman & Hall/CRC, 2004.

Chapter 23 of this work is written by Guibas and simply re-states much of what is described in [2] and [1].

[4] Daniel Russel, Menelaos I. Karavelas, and Leonidas J. Guibas. A package for exact kinetic data structures and sweepline algorithms. *Computational Geometry: Theory and Applications*, 38:111 – 127, 2007.

This paper was published alongside a C++ library which provides useful features for implementing arbitrary kinetic data structures, and explains the details behind the implementation of that library. We hoped to use this paper to help us implement our own KDSs but were unable to do so by the end of the project.