An Efficient Kinetic Range Query for One Dimensional Axis Parallel Segments

T. Hema, Anna University, Department of Computer Science and Engineering, Chennai, India
K. S. Easwarakumar, Anna University, Department of Computer Science and Engineering, Chennai, India

ABSTRACT

We present a kinetic data structure named Kinetic Interval Graph (KI-Graph) for performing efficient range search on moving one dimensional axis-parallel segments. This finds applications in Artificial Intelligence such as robotic motion. The structure requires O(n) storage. The time taken per update when a critical event occurs is O(1) thereby improving responsiveness when compared to the kinetic segment trees, while the overall updates across all segments at a time instance is at most n/2. Also, range query is performed efficiently in Θ(k) time, where k segments are reported.

KEYWORDS

Computational Geometry, Interval Graph, Kinetic Data Structures, Range Search, Segments

INTRODUCTION

Kinetic data structures (Basch, 1999; Basch, Guibas, & Hershberger, 1997; L. J. Guibas, 1998) are a class of geometric data structures used to represent moving objects, which are objects in continuous motion. Unsurprisingly, such data structures have many applications in robotics for collision detection and collision avoidance, a sub-discipline of Artificial Intelligence (Russell, Norvig, & Intelligence 1995). Robotics includes intelligent control as a sub-area of research. Some of the other applications of kinetic data structures are in motion planning, computer graphics such as games, animation, mesh generation, traffic control, transaction time databases etc. Computational geometry (de Berg, Cheong, Kreveld, & Overmars, 2008; Preparata & Shamos, 1985) offers solutions to such problems by considering the objects depending on their respective shapes such as points, lines, etc. Normally, a geometric data structure such as kd-trees are used to store point data and segment trees are used to store line segments in these problems. In this work, we consider line segments representing a robot or a vehicle. The primary goal is to design a kinetic data structure that requires fewer updates to the data structure during an event, and this actually determines the efficiency of the data structure. Kinetic segment trees (de Berg et al., 2001) have realized an O(log n) time and the running time has to be reduced further. The best-known application of a KDS is maintaining a convex hull on a set of moving points (Abam & de Berg, 2005; Basch, Guibas, & Zhang, 1997). Recently, nearest neighbours of points are found using kinetic triangulation as a KDS for mesh refinement (Acar, Hudson, & Turkoglu, 2011; Kaplan, Rubin, & Sharir, 2011).

Very recently, the kinetic based applications for well-known problems such as closest pairs, all nearest neighbours and reverse nearest neighbours have been shown to be local, responsive, compact and efficient in either amortized sense or average case analysis on these measures (Rahmati, 2014) for each problem. Most of the data structures such as range trees, segment trees, interval trees, kd-trees as well as the Voronoi diagram, Delaunay triangulation used in computational geometry are dealt for

DOI: 10.4018/IJIIT.2018010104

Copyright © 2018, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.
kinetic data sets (Abam & de Berg, 2005; Abam, de Berg, & Speckmann, 2007; Agarwal, Erickson, & Guibas, 1998; Agarwal, Guibas, Murali, & Vitter, 2000; Agarwal, Kaplan, & Sharir, 2008; Basch, Guibas, & Hershberger, 1997; Basch, Guibas, & Zhang, 1997; de Berg, Comba, & Guibas, 2001; Czumaj, Artur, & Sohle, 2007; Guibas, Mitchell, & Roos, 1991). Recent work includes proximity problems on moving points and kinetic closest pair (Rahmati et al., 2015; Rahmati et al., 2016; Chan, & Rahmati 2016.). A detailed study on the methods used for such structures are described in (Gotthardt, 2008). A kinetic binary space partitioning tree (BSP) was proposed to address ray shooting queries efficiently and maintain moving segments in a plane (Abam et al., 2007). They assume that segments are interior-disjoint, arbitrary and an event occurs when x-coordinates of segments coincide, and the BSP is updated in $O(\log n)$ time per event. Although kinetic BSP-trees are efficient and local, since $\Omega(n)$ time is taken for processing events, they are not responsive. However, the issue of whether there could be a BSP structure with fewer events and updates to the tree for a set of segments, remains open (de Berg et al., 2001). To our knowledge, the aforementioned problems are the only known kinetic segment related problems found in the literature. Applications: A typical example of a transaction range time-slice query that is common in transaction-time database applications (Salzberg & Tzotras, 1999) is as follows: Given a key range and a contiguous time interval $T$, find the objects with keys in the given range that are alive during interval $T$. Likewise, in the case of visibility problems, when two robots are in linear motion, collision can be detected when they are about to coincide. In collision detection, for instance, let us consider two robots moving towards each other. Here, one has to determine the next move at which they are expected to collide. We translate this into a geometric problem as follows: Given two axis parallel line segments, moving towards each other, one has to find the critical event, an event when either two segments collide or cross each other. Yet another application of kinetic segments is traffic control where the vehicles are considered as line segments.

In this work, we introduce the $KI$-Graph to reduce the update time for an event in order to achieve locality to address range query for a set of segments. We assume that the segments are one-dimensional, which are parallel to the x-axis, and move with uniform velocity following a back and forth movement. Also, some segments may be stationary but not necessarily, while other segments are moving continuously. Thus, a range query reports set of overlapping segments of a particular segment (query segment), for a given time instance. Since segments refer to intervals during a time instance, we make use of an interval graph representation. Also, the main reason for choosing an interval graph to represent segments is that, until endpoints cross each other, the combinatorial changes do not occur for every flight plan. Background: Recall that a graph $G$ is an interval graph $I(G)$ if its vertices can be assigned to intervals on the real line such that there is an edge between two vertices if and only if their corresponding intervals overlap. Also, an interval graph-based representation requires linear construction time and storage. Moreover, the interval graph representation is useful in addressing range query, as the answer lies only within the set of adjacent vertices of the vertex that represents the query segment.

**Definition 1 (Hajos, 1957)**

An undirected graph $G = (V, E)$ is said to be an interval graph for a set of closed intervals $I = \{I_i : [b_i, e_i], i = 1, 2,...n\}$ on a line, where $b_i$ and $e_i$ respectively refer to x-coordinates of the left and right points of the interval $I_i$. Also, each vertex $v_i \in V$ corresponds to an interval $I_i$ and for $i = j$, $(v_i, v_j) \in E$ if and only if $I_i \cap I_j = \phi$.

The best representation of an interval graph is to label each of the endpoints with a number within $[1, 2n]$ according to the order of its presence and store them in a sorted array. Thus, the adjacency of the nodes in an interval graph corresponding to intervals $I_i, [a,b]$ and $I_j, [c,d]$ are checked as if $a \leq c \leq b$ or $c \leq a \leq d$. Hence, this requires linear storage which is better than the other representations of the graph such as adjacency matrix or adjacency list.
Towards a New Multicriteria Decision Support Method Using Fuzzy Measures and the Choquet Integral
www.igi-global.com/article/towards-a-new-multicriteria-decision-support-method-using-fuzzy-measures-and-the-choquet-integral/144204?camid=4v1a

Symbolic Data Analysis: A Paradigm for Complex Data Mining?
www.igi-global.com/article/symbolic-data-analysis/104639?camid=4v1a