VIII International Conference on Computational Plasticity
COMPLAS VIII
E. Oñate and D.R.J. Owen (Eds)

SWEEP-PLANE APPROACH TO BOUNDING BOX INTERSECTION

Tomasz Koziara[⋆], **Nenad Bićanić**[†]

Department of Civil Engineering
University of Glasgow
Glasgow G12 8LT, Scotland, UK
[⋆]e-mail: t.koziara@civil.gla.ac.uk, web page: http://www.civil.gla.ac.uk/~koziara
[†]e-mail: bicanic@civil.gla.ac.uk, web page: http://www.civil.gla.ac.uk/~bicanic

Key words: Bounding Box Intersection, Collision Detection, Sweep-plane Approach, Spatial Hashing, Priority Search Tree

Summary. *This paper summarises an effort towards approximation of an optimal sweep-plane approach to axis aligned bounding box intersection problem. In particular a spatial hash table and priority search tree are combined in order to obtain a data structure suitable for solving two-dimensional dynamic rectangle intersection problem. Some variants of this structure allow logarithmic update and query times, although all of them suffer from repeated reports of intersections. This constrains an efficient application of presented algorithms only to sparse box distributions, where penalty of additional algorithmic effort for suppressing repeated reports is minor.*

1 BACKGROUND

Axis aligned bounding boxes are common heuristic, used whenever there is a need to reduce the amount of tests performed between interacting geometrical objects. Being only a special case of broader class of *bounding volumes*, bounding boxes are often chosen for the sake of their simplicity. Recently, Suri et al. [3] formally proved that as long as bounding boxes meet some reasonable assumptions about their shape with respect to shape of bounded objects, the number of intersections between boxes remains proportional to the number of actual objects intersections. This result reassures usefulness of bounding box heuristic.

Problem of intersection between three-dimensional boxes arises in a variety of areas. Motivation for this investigation comes from the need for an efficient contact detection framework in modelling of masonry structures. In [4], box intersection is one of the building blocks for more general algorithm searching for collisions between pairs of convex polyhedra. Employed there approach, called *sweep and prune*, exploits *temporal coherence* of a typical dynamic setting (similarity between the configurations of objects in adjacent time frames) to achieve $O(n+k)$run time complexity (n denotes number of boxes, while k number of intersections). Unfortunately the space requirement is $O(n^2)$, thus not feasible for large n. Another implemented and efficient approach [1], based on hybridisation of *multi-level segment tree*, achieves respectively $O(n \log^3 n + k)$ time and $O(n)$ space complexity. This algorithm, further called HYBRID, will serve as a reference one.

2 SWEEP-PLANE ALGORITHM

Sweeping is one of the classical techniques employed in computational geometry. Using a *sweep-plane* approach reduces three-dimensional box intersection problem to a *two-dimensional dynamic rectangle intersection* one. Assuming that one has a dynamic structure for rectangle intersection problem DR, and a sweep-plane is orthogonal to z-direction, whenever a box is about to enter the plane, DR is *queried* with its xy-rectangle r, and all the overlaps with rectangles already stored in DR are reported. Next r is *inserted* into DR. When a box is about to leave the sweep-plane, its xy-rectangle is *deleted* from DR. As one can sort z-coordinates of box endpoints in $O(n \log n)$ steps, the actual efficiency depends on the complexity of *query*, *insertion* and *deletion* operations. Optimal structure should enable logarithmic update and query times as well as have linear space complexity. There is no such structure to date, so here it is only approximated. Four versions of the sweep-plane algorithm are tested: SWEEP1 (DR is imitated by two-dimensional spatial hashing [2] and single linked lists for storing rectangles), SWEEP2 (DR is imitated by two-dimensional hashing and a *priority search tree* (PST [5]) along the x-direction), SWEEP3 (DR is imitated by PST along x-direction, for each reported x-intersection one needs to check the y-one), SWEEP4 (DR is imitated by one-dimensional hashing along x-direction and PST along y-direction). As the sweep-plane approach requires initial sorting of coordinates, it is can take advantage of *temporal coherence* and sort them in an expected $O(n)$ time, when executed for the consecutive time steps [4].

3 EXPERIMENTS

The testing suite comprises three different classes of sets. As a deficiency of DR structures results in a additional checks for repeated reports of intersections, testing asymptotic behaviour of the algorithms requires “over-packed” distributions of boxes. For this purpose random and sphere surface box distributions with $O(n^2)$ intersections are used. In most situations though, the number of box intersections will be of order $O(n)$. For example in case of masonry, each brick can contact only with a small number of surrounding ones, which results in the mentioned estimate. Here spherical surface and adjacent distributions are employed (Figure 1).

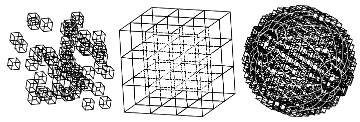


Figure 1: Examples of three classes of testing sets: random, adjacent and spherical surface distributions of bounding boxes.

All algorithms were implemented with approximately the same care and combined withing one testing software[1]. Results presented here were obtained on 1.7GHz CPU machine, with the code compiled with optimisation flags switched off.

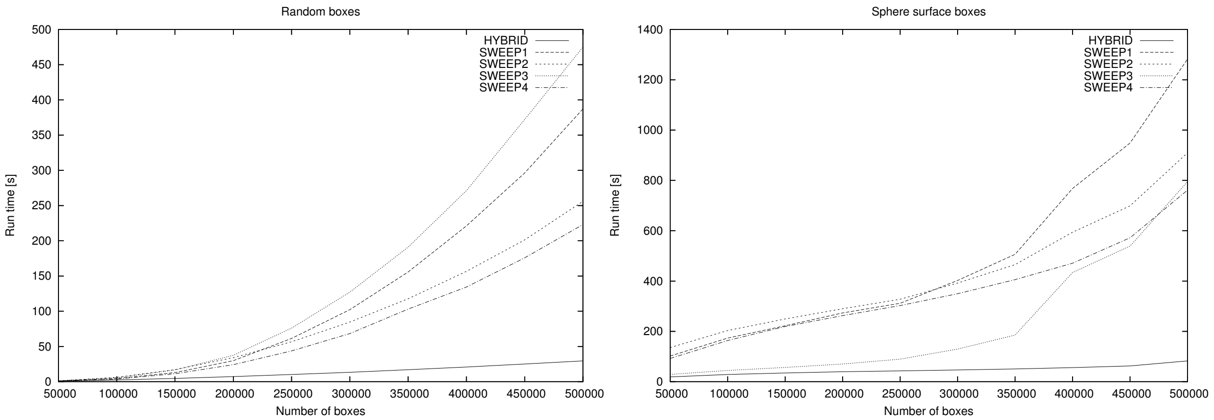


Figure 2: Timings for random and spherical surface sets with number of intersections $k = O\left(n^2\right)$.

Results for dense box distributions (Figure 2) show poor asymptotic behaviour of analysed sweep-plane algorithms. One can observe pretty symptomatic performance of DR structures applied for each version of sweep-plane approach. SWEEP2 and SWEEP4 act similarly, taking advantage of space-filtering properties of hashing and good indexing behaviour of PST. Hashing by itself is sensitive to box clustering and irregularities (SWEEP1). Priority search tree on the other hand - as it works with only along one dimension - appears to be sensitive to the particular distribution boxes (SWEEP3).

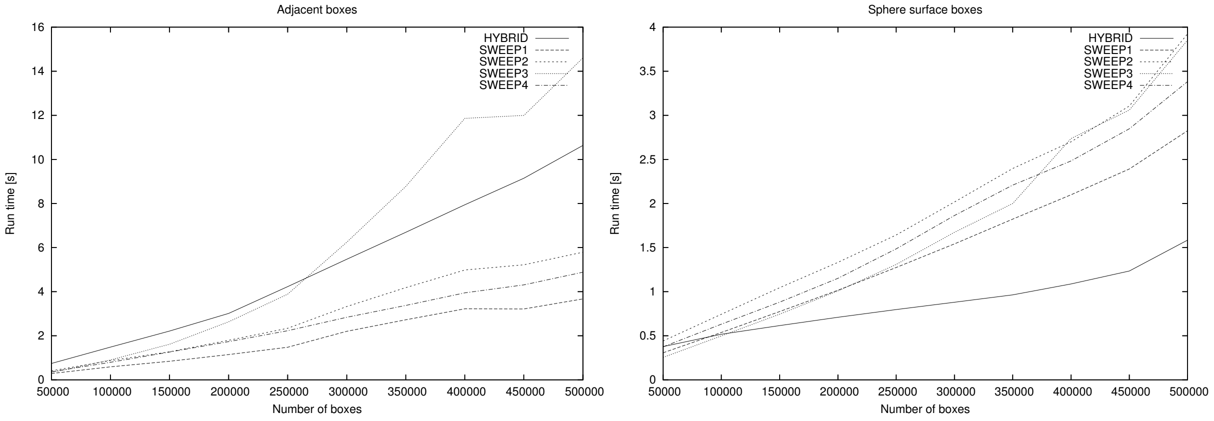
The case of adjacent distribution of boxes (Figure 3) confirms anticipated behaviour of SWEEP1. Naive application of spatial hashing in $DR1$ structure does not confine a good performance, as in this case all the operations do take expected constant time. For a sparse spherical distribution again, SWEEP1 outperforms the other sweep-plane approaches. It can be also added, that shaving $\log n$ factor from the initial sorting time with the *temporal coherence* assumption results in a speedup factor not bigger than two.

4 CONCLUSIONS

An obvious, although not trivial to accomplish improvement, would be to apply a correct data structure for the dynamic rectangle intersection problem. A correct, means here avoiding repeated reports of intersections. One could employ two-level dynamic segment tree [5], scarifying space usage and allowing $O(n \log^2 n)$ one in a worst case. Hopefully, for practical distributions of boxes, only small percentage of them would be

---

[1] One can download it from *http:/www.civil.gla.ac.uk/˜koziara/code.html*



Adjacent boxes

Sphere surface boxes

Figure 3: Timings for adjacent and spherical surface sets with number of intersections $k = O(n)$.

stored in a DR structure during the sweeping process. On the other hand en efficient implementation of such a structure is questionable.

An interesting direction for further investigation could be related with data structures commonly applied in a field of databases. Some variants of *R-Trees* [6] used there for answering window queries could eventually appear useful in the analysed setting.

REFERENCES

[1] A. Zomorodian, H. Edelsbrunner, *Fast software for box intersections*, 2002, *International Journal of Computational Geometry and Applications*, Vol. 12, Numbers 1-2, pp. 143-172.

[2] M. Teschner, B. Heidelberger, M. Müller, D. Pomeranets, M. Gross, *Optimized Spatial Hashing for Collision Detection of Deformable Objects*, 2003, *Proc. Vision, Modelling, Visualization VMV'03*, pp. 47-54.

[3] S. Suri, P. M. Hubbard, J. F. Hughes, *Analyzing Bounding Boxes for Object Intersection*, 1999, *ACM Trans. Graph 18(3)*, pp. 257-277.

[4] J. D. Cohen, M. C. Lin, D. Manocha, M. K. Ponamgi, *I-COLLIDE: An Interactive and Exact Collision Detection System for Large-Scale Environments*, 1995, *Proceedings 1995 Symposium on Interactive 3D Graphics*, pp. 189-196.

[5] H. Samet, *The Design and Analysis of Spatial Data Structures*, *Addison-Wesley, Reading, MA, 1990. ISBN 0-201-50255-0.*

[6] A. Guttman, *R-Trees: A Dynamic Index Structure For Spatial Searching*, 1984, Boston, In *Proc. ACM-SIGMOD International Conference on Management of Data*, pp. 47-57.