The Pathtree, a Suitable Datastructure for Parallel Rayshooting in 3-Dimensional Space

W. Preilowski
Department of Mathematics and Computer Science
University of Paderborn

Abstract

A new parallel algorithm for the $3D$ rayshooting problem is presented on a CREW-PRAM. For this the pathtree, a suitable datastructure for representing the boundary of convex polytopes is introduced.

It will be shown, that for a convex polytope $\mathcal{P}$ of size $O(n)$ the pathtree is computable in $O(log(n))$ with $O(n)$ processors. Then, with the help of the pathtree the intersection segment of $\mathcal{P}$ and any line g in 3-dimensional space can be computed in time $O(log(n))$ with $O(n^\epsilon)$ processors. This is the first parallel algorithm with global running time $O(log(n))$ for the $3D$ rayshooting problem.

An application of the rayshooting algorithm is given with a new parallel algorithm for the computation of the intersection of two convex polytopes in $\mathbf{R}^3$. It needs time $O(log(n))$ with $O(n^{1+\epsilon})$ processors and is the first parallel algorithm for this problem using less than $O(log(n)^2)$ time with a reasonable number of processors.

Topics: Computational Geometry, Theory of Parallel and Distributed Computation

1 Introduction

The rayshooting problem is of vital importance in computational geometry. It is the main subroutine in a lot of geometric algorithms, e.g. intersection problems like *convex hull, intersection of halfspaces* and *polytopes* or *sweepline, hidden surface* and *visibility problems*, etc. .

Given a convex polytope $\mathcal{P}$ with n extreme points and a set of straight lines G in 3-dimensional space, the rayshooting problem is defined to decide for every line $g \in G$ whether it intersects $\mathcal{P}$. If g is intersecting $\mathcal{P}$, the intersetion segment is computed. Otherwise, the point on the boundary of $\mathcal{P}$ nearest to g is found.

In $\mathbf{R}^2$ the rayshooting problem can be solved optimally in $O(log(n))$ with $O(n)$ processors, whereas in 3-dimensional space this problem has not yet been optimally solved with the same time and processor bounds.

Dadoun and Kirkpatrick have presented the best result [DaKi 87] in $\mathbf{R}^3$. In a preprocessing-phase, they compute a data-structure for $\mathcal{P}$ to solve line-queries. Time $O(log(n)\cdot\log^*(n))$ with $O(n)$ processors is needed, achieving a query time of $O(log(n))$

for any line with one processor. This leads to a global running time of $O(log(n) \cdot \log^*(n))$ and $O(n)$ processors for all lines of G.

For the new parallel algorithm a datastructure called *pathtree* for the boundary $O_{\mathcal{P}}$ of $\mathcal{P}$ is computed. The pathtree is an interesting datastructure and should have many applications in a lot of other geometric problems.

A pathtree can be used to solve any point query which is given by a point and a polygone, the result of an intersection of a plane in $\mathbf{R}^3$ with a line and $\mathcal{P}$.

In this paper it is shown that the rayshooting problem for any line g can be reduced to solve n^ϵ point queries recursively.

The new algorithm is based on a result by Cole [Co 86]. In his work a datastructure is described, which is suitable for storing convex polytopes with optimal space $O(n)$. He has shown that it can be used to solve some geometric problems like *planar point location* or 3-*dimensional point location*.

For the preprocessing, say the computation of the pathtree, the new algorithm needs time $O(log(n))$ with $O(n)$ processors. Then a query time of $O(log(n))$ with $O(n^\epsilon)$ processors is achieved for any line $g \in G$.

Given two convex polytopes in $\mathbf{R}^3$, the new parallel algorithm given here needs time $O(log(n))$ with $O(n^{1+\epsilon})$ processors to compute their intersection.

If one is able to compute one point of the intersection-polytope in advance, a method described by Chazelle, namely a geometric dual point/flat transformation, can be used to reduce the polytope intersection problem to the problem of computing the transitive closure of the union of two convex polytopes [Cha 89].

One of these witnesses of the intersection can be computed with the help of the new procedure for rayshooting in time $O(log(n))$ with $O(n^{1+\epsilon})$ processors. Then the transformed problem can be solved by execution of the merging step of the parallel algorithm by Preilowski et.al. [PrDaWe 92] computing the convex hull of a set of points in $\mathbf{R}^3$ in $O(log(n))$ with $O(n^{1+\epsilon})$ processors.

The parallel *CREW-PRAM* model used here is a concurrent read, exclusive write random access machine. Such a machine is conceived as having a large number of processors with common access to a common memory. Any number of processors can read the same memory cell at the same time, but only one can access the same memory cell at the same time for writing.

References

[AtCoGo 89] M.J.Atallah, R.Cole, M.T.Goodrich. *Cascading Divide- and Conquer: A Technique for Designing Parallel Algorithms. SIAM Journal of Computing, vol. 18, no. 3, Seiten 499-532, 1989.*

[Cha 89] B. Chazelle. *An Optimal Algorithm for Intersecting Three-Dimensional Convex Polyhedra. Proceedings 30th Symp. on Foundations of Computer Science, Pages 586-591, 1989.*

[ChDo 87] B.Chazelle, D.P.Dobkin. *Intersection of Convex Objects in Two and Three Dimensions. Journal of the Association for Computing Machinery, vol. 34, no. 1, pages 1-27, 1987.*

[ChGu 86,a] B. Chazelle, L. L. Guibas. *Fractional Cascading: I. A Data Structure Rechnique. Algorithmica, vol 1, pages 133-162, 1986.*

[ChGu 86,b] B. Chazelle, L. L. Guibas. *Fractional Cascading: II. Applications. Algorithmica, vol 1, pages 163-191, 1986.*

[Co 86] R.Cole. *Searching and Storing Similar Lists. Journal of Algorithms, no. 7, Seiten 202-220, 1986.*

[Co86] R. Cole. *Parallel Merge Sort. Proceedings 27th IEEE Symp. on Foundations of Computer Science, Seiten 511-516, 1986.*

[DaKi 87] N. Dadoun, D.G. Kirkpatrick. *Parallel Construction of Subdivision Hierarchies Journal of Computer and System Science 39, Seiten 153-165, 1989.*

[Ki 53] J. Kiefer. *Sequential minimax search for a maximum. Proc. Am. Math. Soc., No. 4, Seiten 502-506, 1953*

[PrDaWe 92] W. Preilowski, E. Dahlhaus, G. Wechsung. *New parallel algorithms for convex hull and triangulation in 3-dimensional space. 17th. International Symposium of Mathematical Foundation of Computer Science, Seiten 442-450, 1992.*

[Ya 88] C.-K. Yap. *Parallel Triangulation of a Polygon in Two Calls to the Trapezoidal Map. Algorithmica, vol. 3, Seietn 279-288, 1988*