Cost-Optimal Quadtrees for Ray Shooting*

Hervé Brönnimann† Marc Glisse† David R. Wood‡

1 Introduction

Given a set S of objects, called a *scene*, the ray-shooting problem asks, given a ray, what is the first object in S intersected by this ray. A popular approach to speed up ray-shooting queries is to construct a space decomposition such as a quadtree in 2D and an octree in 3D. The query is then answered by traversing the leaves of the tree as they are intersected by the ray, and for each cell in turn, testing for an intersection between the ray and the subset of objects intersecting that cell. In this paper, the only objects we consider are simplices (points and segments inside the unit square $[0,1]^2$ in $\mathbb{R}^2$, or points, segments and triangles inside the unit cube $[0,1]^3$ in $\mathbb{R}^3$). We denote the tree by $\mathcal{T}$ in either 2D or 3D, and use the notation $\mathcal{Q}$ for quadtrees specifically, and $\mathcal{O}$ for octrees.

The following cost measure was introduced by Aronov *et al.* [1] for the purpose of predicting the traversal cost of shooting a ray in $\mathbb{R}^d$, while using a quadtree (for $d = 2$) or an octree (for $d = 3$) to store S:

$$c_S(\mathcal{T}) = \sum_{\sigma \in \mathcal{L}(\mathcal{T})} (\gamma + |S \cap \sigma|) \times \lambda_{d-1}(\sigma), \qquad (1)$$

where $\mathcal{L}(\mathcal{T})$ is the set of leaves of the quadtree, $S \cap \sigma$ is the set of scene objects meeting a leaf σ, and $\lambda_{d-1}(\sigma)$ is the perimeter (if $d = 2$) or surface area (if $d = 3$) of σ. The coefficient γ depends on the implementation, and models the ratio of the cost of the tree traversal (per cell) to that of a ray-object intersection test (per test).

When S is fixed, we simply denote the cost by $c(\mathcal{T})$. If we assume the cost of finding the neighboring cell meet by a ray is bounded by γ, then this cost function provably models the cost of finding all the objects intersected by a random line, with respect to a rigid-motion invariant distribution of lines [1].

We are interested in estimating the quality of a tree with respect to its cost. In particular, we wish to construct trees with as low a cost as possible. We first give (Lemma 1) a lower bound on the infimum $M(S)$ of the cost of any quadtree, for any given S. We then examine several algorithms which do, or don't, give guaranteed optimal cost, or bounded approximation ratio. Our main result is that the 3-lookahead greedy algorithm (introduced below) produces quadtrees with cost $O(M)$ both for points (Lemma 2) and for segments (Lemma 4).

2 Quadtree construction schemes

Terminology. The *square* is a quadtree that has a single leaf (no subdivision). We denote it by $\mathcal{Q}^{(\text{empty})}$. If we subdivide this square recursively until depth k, we get a *complete tree (of depth k)*, denoted by $\mathcal{Q}_k^{(\text{complete})}$, and its leaves form a $2^k \times 2^k$ grid. If only the cells incident to one edge (resp. two and four adjacent edges) of a cell are recursively subdivided until depth k, the subtree rooted at that cell is called a *k-side* (resp. *k-corner* and *k-border*) quadtree, and denoted by $\mathcal{Q}_k^{(\text{side})}$ (resp. $\mathcal{Q}_k^{(\text{corner})}$ and $\mathcal{Q}_k^{(\text{border})}$); see Figure 1. This notation is extended to starting from a cell σ instead of a unit square, by substituting σ for $\mathcal{Q}$: for instance, the complete subtree of depth k subdividing σ is denoted by $\sigma_k^{(\text{complete})}$.

The subdivision operation induces a partial ordering $\prec$ on quadtrees, whose minimum is the square. Again, this partial ordering is extended to subtrees of a single cell σ.

We consider recursive algorithms for computing a quadtree of a given set S of objects, which subdivide each cell until some given termination criterion is satisfied. In particular, for points and segments, we may recursively subdivide the square until each leaf contains at most one object, or any number of identical points, or one segment intersection point (and all the segments intersecting at that point but no other object or endpoint of those segments). We call this the *separation criterion*, and the resulting quadtree the *minimum separating quadtree*, denoted $\mathcal{Q}^{(\text{sep})}(S)$,

*Work on this paper has been supported by NSF ITR Grant CCR-0081964. Research of Hervé Brönnimann has also been supported in part by NSF CAREER Grant CCR-0133599. Marc Glisse is supported by École Normale Supérieure during his stay at Polytechnic University.

†CIS, Polytechnic University, Six Metrotech, Brooklyn, NY 11201, USA. Email: hbr@poly.edu, marc.glisse@ens.fr.

‡School of Computer Science, Carleton University, Ottawa, Ontario K1S 5B6, Canada. Email: davidw@scs.carleton.ca.

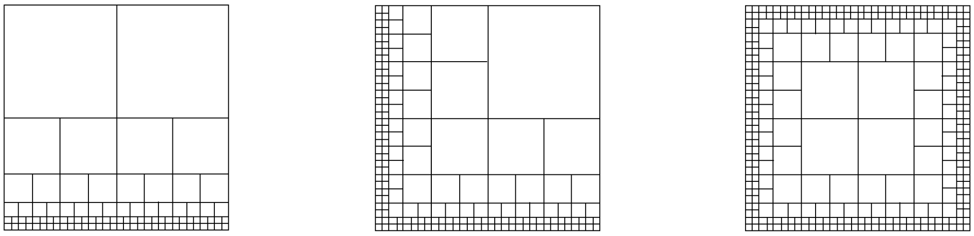

Figure 1: The k-side quadtree $\mathcal{Q}_k^{(\mathrm{side})}$ (left), a corner $\mathcal{Q}_k^{(\mathrm{corner})}$ (center), and a border $\mathcal{Q}_k^{(\mathrm{border})}$ (right).

with variants where the recursion stops at depth k, denoted $\mathcal{Q}_k^{(\mathrm{sep})}(S)$. (Note that the depth of $\mathcal{Q}^{(\mathrm{sep})}$ is always finite with our definition.)

The "optimal" quadtree may not have finite depth (it is conceptually possible to decrease the cost by subdividing at infinity), so we let M denote the infimum of $c(\mathcal{Q})$ over all quadtrees $\mathcal{Q}$ for S. (As a consequence of Lemma 1 below, $M \geq 4\gamma$.) All the subdivision schemes have variants where the recursion stops at depth k. When we say that a subdivision scheme is *optimal*, we mean that its cost when restricted to depth k tends to M as k tends to ∞. The expression *near-optimal* will be used when the cost is $O(M)$ instead of M. In order to have an algorithm that terminates, we can add an extra termination criterion such as a depth limit of $\log_2 n$. Although we will not prove it here, this increases the cost at most by a constant factor.

Cost measure under subdivision. First, we observe that the cost measure can be split into two terms: $c_q(\mathcal{Q}) = \gamma\lambda_1(\mathcal{Q}) = \gamma\sum_\sigma \lambda_1(\sigma)$ (the *quadtree cost*), and $c_o(\mathcal{Q}) = \sum_{s\in S}\lambda_1(s\cap\mathcal{Q})$ (the *object cost*), where $s\cap\mathcal{Q}$ denote the set of leaves of $\mathcal{Q}$ crossed by s and λ_1 is extended to sets of leaves by summation. Since the quadtree cost cannot be less than 4, and the object cost cannot be less that the total Euclidean length of S, we have:

Lemma 1 $c(\mathcal{Q}) \geq 4\gamma + 2\sqrt{2}\sum_{s\in S}\lambda_1(s)$.

In the analysis below, it is useful to keep in mind the following simple observations: when subdividing a cell, its quadtree cost doubles, and the object cost of an object becomes $m/2$ where $m \in [1\ldots4]$ is the number of children intersected by the object. Note that $m \leq 3$ for a segment (unless it passes through the center of the cell). As the quadtree grows finer, the quadtree cost increases while the object cost presumably decreases. Thus a certain degree of subdivision may lower the overall cost.

As examples, we compute the cost of several configurations.

0. With no subdivision, the cost of the unit square $\mathcal{Q}^{(\mathrm{empty})}$ is $c(\mathcal{Q}^{(\mathrm{empty})}) = 4(\gamma+n)$.

1. The cost of a full subdivision at depth k, $\mathcal{Q}_k^{(\mathrm{complete})}$, is at least $4(2^k + \frac{n}{2^k})$ (this is the exact value if all the points fall in a single cell of $\mathcal{Q}_k^{(\mathrm{complete})}$), and at most $4(2^k + \frac{4n}{2^k})$ (if all the points belong to four cells).

2. It is useful to know the quadtree costs of the k-side, k-corner, and k-border:

$$\begin{aligned} c_q(\mathcal{Q}_k^{(\mathrm{side})}) &= 4\gamma(k+1), \\ c_q(\mathcal{Q}_k^{(\mathrm{corner})}) &= 4\gamma\left(2k-1+\frac{1}{2^{k-1}}\right), \\ c_q(\mathcal{Q}_k^{(\mathrm{border})}) &= 8\gamma\left(2k-3+\frac{1}{2^{k-2}}\right). \end{aligned}$$

3. Let S_n denote n distinct points very close to one corner of the unit square, let's say the origin. After k levels of recursively subdividing the lower left cell,[1] we obtain the quadtree $\mathcal{Q}_k^{(\mathrm{sep})}(S_n)$ whose cost is $c(\mathcal{Q}_k^{(\mathrm{sep})}(S_n)) = 12\gamma + (n-2\gamma)2^{2-k}$. Whether this is an improvement for any value of k depends on n and γ. In particular, $\mathcal{Q}_k^{(\mathrm{sep})}(S_n)$ has cost lower than the unit square for large values of k, only if $n > 2\gamma$.

This example tells us that, to produce optimal or near-optimal quadtrees, subdivision strategies based on the number of objects in a cell—like the separation criterion—must depend on the value of γ.

Dynamic programming and greedy strategies. Since the previous examples show that the cost measure can be non-monotonous, there is no reason that simple termination criteria produce optimal or even near-optimal quadtrees. The *dynamic programming algorithm* finds the quadtree with depth at most k that minimizes the cost, which we denote by $\mathcal{Q}_k^{(\mathrm{opt})}(S)$ (or $\sigma_k^{(\mathrm{opt})}(S)$ if we start from a cell σ instead of the unit square). Indeed, either the optimal tree is the square itself, or the root is subdivided and

[1] Here, 'very close' means always within the lower left cell. It's then pointless to subdivide the other cells since they do not contain points of S, their cost would be doubled.

its four subtrees are also optimal (the *optimal substructure property*). The algorithm starts with the complete tree of depth k, $\mathcal{Q}_k^{(\text{complete})}$, and performs a bottom-up traversal of all the nodes, while maintaining the optimum cost of a quadtree rooted at that node. The decision whether to subdvide is made by comparing the cost of the cell vs. the optimum costs of subtrees rooted at its four children.

Unfortunately, the memory requirements of this algorithm are huge for large values of k. We therefore resort to a greedy strategy with bounded lookahead. The algorithm proceeds by recursively subdiving the nodes with a greedy termination criterion: when examining a cell σ, we run the dynamic programming within σ with depth k (k is a parameter called *lookahead*). If the best subtree $\sigma_k^{(\text{opt})}(S)$ does not improve the cost of the unsubdivided node σ, then the recursion terminates. Otherwise, we replace σ by it optimal subtree $\sigma_k^{(\text{opt})}(S)$ and recursively evaluate the criterion for the leaves of $\sigma_k^{(\text{opt})}(S)$. We denote the resulting quadtree by $\mathcal{Q}_k^{(\text{greedy})}(S)$ (or $\sigma_k^{(\text{greedy})}(S)$ if we start from a cell σ instead of the unit square).

With no lookahead ($k = 1$), the greedy strategy simply examines whether one subdivision decreases the cost measure. Below we analyze the greedy strategies without and with lookahead, first for points, then for segments.

3 Cost-optimal quadtrees for points

In the sequel, we will use the fact that a point can belong to at most 4 cells. In general a point in S can belong to 1, 2, or 4 leaves. We also allow a point to be assigned to a choice of 3 out of 4 adjacent leaves, when it sits at the center of a cell of the quadtree (the intent is to model infinitesimal segments). Arguably, the case of points is of theoretical interest only, but has relevance since simplices are usually very small in a graphics scene, and can be assimilated to points.

We first examine the simple termination criteria. Take the example of n points in a corner, S_n, described in our configuration **3**. With no subdivision, the cost is $4(\gamma + n)$. With full subdivision to depth k, the cost is $2^{k+2}\gamma + n2^{2-k}$ which is at least $8\sqrt{\gamma n}$ for any value of k. Now the minimum separation criterion gives a cost of $12\gamma + (4n - 3\gamma)2^{-k}$ (this is also what the greedy algorithm leads to). For large values of n, these are respectively $\Theta(n)$, $\Omega(\sqrt{n})$ and $O(1)$.

We now argue that the 3-lookahead greedy strategy produces a near-optimal quadtree for points. Note that this is not true of 2-lookahead greedy (simply consider sufficiently many points in the center of the square).

Lemma 2 *Let S be a set of n points in the unit square, and let M be the infimum of $c(\mathcal{Q})$ over all quadtrees $\mathcal{Q}$. Then $\mathcal{Q}_3^{(\text{greedy})}(S)$ has near optimal cost, namely $c(\mathcal{Q}_3^{(\text{greedy})}) = O(M)$.*

Proof. The cost of a cell σ is $(\gamma + |S \cap \sigma|)\lambda_1(\sigma)$ The cost of a complete subdivision $\sigma_3^{(\text{complete})}$ of σ to depth 3 is $(8\gamma + \frac{1}{8}\sum_{p \in S \cap \sigma} n_p)\lambda_1(\sigma)$, where n_p is the number of leaves of that subdivision that contain p. Since n_p is at most 4, the latter cost is at most $8\gamma + |S \cap \sigma|/2$. If σ is a leaf of $\mathcal{Q}_3^{(\text{greedy})}$, this means that $\sigma_3^{(\text{opt})}$ is just the leaf σ, then certainly $c(\sigma_3^{(\text{complete})})$ is greater than $c(\sigma)$. Hence $8\gamma + |S \cap \sigma|/2 \geq \gamma + n$, i.e. $|S \cap \sigma| \leq 14\gamma$. The only case when the 3-lookahead greedy strategy fails, is when the lookahead says it is wrong to subdivide whereas it is not. In that case, the number of points is at most 14γ. Then the cost is at most $15\gamma\lambda_1(\sigma)$, and because of Lemma 1 this is $O(M_\sigma)$, where M_σ is the infimum cost of all the possible quadtree subdivisions of σ. Since lookahead only subdivides a cell if it has a subtree that actually improves the cost, necessarily $\mathcal{Q}_3^{(\text{greedy})} \prec \mathcal{Q}^{(\text{opt})}$ (where $\mathcal{Q}^{(\text{opt})}$ is any optimal[2] quadtree). Thus every leaf σ of $\mathcal{Q}_3^{(\text{greedy})}$ is a cell of $\mathcal{Q}^{(\text{opt})}$, and since $c(\sigma) = O(M_\sigma)$ then $c(\mathcal{Q}_3^{(\text{greedy})}) = O(\mathcal{Q}^{(\text{opt})}) = O(M)$ as well. □

As for finding the optimal quadtree, the question is still open to know whether for given values of γ (and maybe of n) there exists a k such that the lookahead strategy yields to the optimal result. All we know is that if $n = 5$ and $\gamma < 1$ tends to 1, the required k tends to infinity. We can also mention that if any point belongs to at most one cell, then $k = 1$ is optimal.

Note that Aronov and Schiffenbauer [2] have also and independently proven that $\mathcal{Q}^{(\text{AS-sep})}(S)$ is near-optimal, where $\mathcal{Q}^{(\text{AS-sep})}$ uses a slightly different separation criterion: subdivide a cell whenever it covers more than one point of S, and also if any of its orthogonal (resp. diagonal) neighbors has depth which differs by more than 1 (resp. 2). Their construction is valid only if all the points in S are distinct.

4 Cost-optimal quadtrees for segments

The situation for line segments is somewhat different. Whereas the k-lookahead greedy strategy may

[2] The cost M is an infimum and may not be achieved by any finite-depth quadtree. But for this proof, $\mathcal{Q}^{(\text{opt})}$ can be allowed to have infinite depth. Alternately, one may consider an increasing sequence of quadtrees $\mathcal{Q}_k^{(\text{opt})}$ whose costs converge towards M.

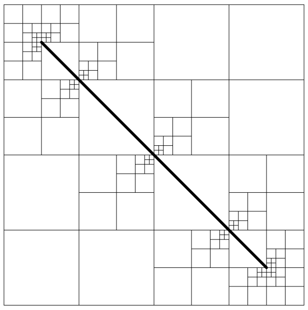

Figure 2: An example of quadtree $\mathcal{Q}^{(2)}_{k,m}$, consisting of n copies of the segment pq, with $p = (1 - 2^{-m-1}, 2^{-m-1})$, and q its reflexion about the main diagonal; here $m = 2$ and $d = 4$.

yield the optimal quadtree for points for some (large enough) k, for segments it fails to do so for any k.

Lemma 3 *The lookahead greedy strategy does not always give a cost-optimal quadtree. Specifically, for any k, there is a set S of n segments such that no quadtree of depth at most k has cost less than $4(\gamma+n)$, but some quadtree of depth at least $k+1$ has cost less than $4(\gamma+n)$.*

Proof. Let $p = (1 - 2^{-m-1}, 2^{-m-1})$, and $q = (2^{-m-1}, 1 - 2^{-m-1})$ be its reflexion about the main diagonal. Take n copies of the segment pq, and consider the cost-optimal quadtree $\mathcal{Q}^{(2)}_{k,m}$ of depth at most k. As long as $k \le m+1$, the situation is similar to the case where pq is the whole diagonal and the cost-optimal quadtree is the square, as can easily be verified. When k becomes larger, however, it becomes interesting to subdivide the corners, as shown in Figure 2. A recurrence shows that the cost of that quadtree is exactly

$$\begin{aligned} c(\mathcal{Q}^{(2)}_{k,m}) &= 4\gamma\left(10 - \frac{1}{2^{m-2}} - \frac{m-3}{2^{k-3}}\right) \\ &\quad +4n\left(1 - \frac{1}{2^m} + \frac{m+1}{2^{k-2}}\right). \end{aligned}$$

For large enough values of n and k, this is an improvement over the square, whose cost is $4(\gamma+n)$. Thus fixing m, one may choose n and compute $k > m$ such that $c(\mathcal{Q}^{(2)}_{k+1,m}) < 4(\gamma+n)$ but $c(\mathcal{Q}^{(2)}_{i,m}) \ge 4(\gamma+n)$ for all $i \le k$. □

In this counter-example, the ratio between the cost of the optimal quadtree and the cost of the initial quadtree is bounded below by a constant. Thus in that case the quadtree we obtained is a good approximation of the optimal solution. In fact, this can be proven for all sets of segments.

Lemma 4 *Given a set S of points or segments in the unit square, and let M be the infimum of $c(\mathcal{Q})$ over all quadtrees $\mathcal{Q}$. The quadtree $\mathcal{Q}_3^{(\mathrm{greedy})}$ constructed by the greedy 3-lookahead strategy has cost $c(\mathcal{Q}_3^{(\mathrm{greedy})}) = O(M)$.*

Proof. The intuition is that points or small objects behave well, and the cost of a big object is bounded below by a constant times its size so it cannot be reduced by very much. Let us look at a leaf σ of the quadtree $\mathcal{Q}_3^{(\mathrm{greedy})}$. Assume that among the objects in $S \cap \sigma$, there are a objects belonging to at most 4 leaves of $\sigma_3^{(\mathrm{greedy})}$, and b other objects. The cost of σ is $(\gamma + a + b)\lambda_1(\sigma)$. Since $\sigma_3^{(\mathrm{complete})}$ has cost $(8\gamma + a/2 + 64b/8)\lambda_1(\sigma)$, which must be at least $c(\sigma_3^{(\mathrm{opt})})$, if σ is a leaf of $\mathcal{Q}_3^{(\mathrm{greedy})}$ we have $c(\sigma) = (\gamma + a + b)\,\lambda_1(\sigma) \le c(\sigma_3^{(\mathrm{opt})}) \le (8\gamma + 4a/8 + 64b/8)\,\lambda_1(\sigma)$, which implies that $a \le 14\,(\gamma + b)$. A segment which belongs to at least 5 cells has length at least $\frac{1}{8}$ so its contribution to the cost is at least $\sqrt{2}/16$. The optimal cost M_σ is then greater than $\gamma + b\sqrt{2}/16$. Since $\gamma + a + b \le 15\gamma + 15b \le 120\sqrt{2}(\gamma + b\sqrt{2}/16)$, we have proved that M_σ is at least a fixed fraction of the cost of σ, and the lemma follows in a manner similar to the proof of Lemma 2. □

The separating quadtree strategy does not work as well for segments as for points, especially since it is not able to distinguish between a segment that barely intersects the corner of the square and the diagonal (in the first case it is usually good to subdivide, and in the second case it is not). The lookahead strategy is then a true improvement.

Acknowledgments

This research was initiated at the McGill-INRIA Workshop on Computational Geometry in Computer Graphics, February 9-15, 2002, co-organized by H. Everett, S. Lazard, and S. Whitesides, and held at the Bellairs Research Institute of McGill University. We thank the other workshop participants for stimulating conversations and for creating a productive atmosphere. Thanks also go to Boris Aronov and Robert Schiffenbauer for productive discussions.

References

[1] B. Aronov, H. Brönnimann, A.Y. Chang, Y.-J. Chiang. *Cost prediction for ray shooting.* To appear in *Proc. of Eighteenth ACM Symp. on Geom. Comput.*, 2002, Barcelona, Spain.

[2] B. Aronov and R. Schiffenbauer. Manuscript, 2001.

[3] H. Samet. *Applications of Spatial Data Structures.* Addison-Wesley, 1990.