Astronomy and Computing 22 (2018) 1–8

Contents lists available at ScienceDirect

Astronomy and Computing

journal homepage: www.elsevier.com/locate/ascom

Full length article

Constructing high-quality bounding volume hierarchies for *N*-body computation using the acceptance volume heuristic

O. Olsson

School of Information Technology and Electrical Engineering, The University of Queensland, Brisbane, QLD, Australia

ARTICLE INFO

Article history:
Received 25 September 2017
Accepted 6 November 2017
Available online 15 November 2017

Keywords:
N-body
BVH
GPU
Heuristic

ABSTRACT

We present a novel heuristic derived from a probabilistic cost model for approximate *N*-body simulations. We show that this new heuristic can be used to guide tree construction towards higher quality trees with improved performance over current *N*-body codes. This represents an important step beyond the current practice of using spatial partitioning for *N*-body simulations, and enables adoption of a range of state-of-the-art algorithms developed for computer graphics applications to yield further improvements in *N*-body simulation performance. We outline directions for further developments and review the most promising such algorithms.

1. Introduction

Direct *N*-body simulations are the most straightforward way of simulating stellar systems, requiring the least amount of assumptions about the underlying physics (Aarseth, 1999). Unfortunately, direct calculation of gravitational *N*-body interactions has a time complexity of $\mathcal{O}\left(N^2\right)$ making the approach intractable for large systems (Bédorf et al., 2015). A key approach, commonly referred to as tree-codes, to rapidly calculate approximate *N*-body gravitational interactions is to partition the domain using a hierarchy. To date, octrees, which splits the domain into eight equal children at each node, is by far the most widely used (Barnes and Hut, 1986; Barnes, 1990; Bédorf et al., 2012; Warren, 2013). Algorithms used to construct these trees typically follow a spatial subdivision approach, which is simple to implement and fast to execute. However, very little effort has been invested into evaluating the quality of the trees thus produced, or to improve their structure to adapt to the particle distribution. Current practice with respect to leaf-forming is also based on hand tuning constants sizes and lacks a robust approach.

A related computational problem with many similarities to *N*-body simulations exists in the field of computer graphics, where hierarchies of scene primitives are built with the primary purpose of accelerating ray tracing, which is used to produce high quality images for film and other non-interactive media. In this field, the construction algorithms have moved beyond spatial simple partitioning and are designed to optimize the *expected traversal performance* by exploiting knowledge about the distribution of primitives. The dominating structure used in modern applications is the binary *Bounding Volume Hierarchy* (BVH), where each node is represented by an axis aligned bounding box that contains the bounding boxes of all child nodes. This structure differs from spatial splitting by allowing nodes to overlap, which allows more flexibility for the construction algorithm. Practically all high-quality construction algorithms are based on the *Surface Area Heuristic* (SAH), which is a metric that is proportional to the expected cost of traversing the hierarchy (Goldsmith and Salmon, 1987; MacDonald and Booth, 1990). Today the SAH is used to guide a wide range of greedy construction algorithms that take a top-down, bottom-up, tree-rotation, insertion, or local optimization approach (Kensler, 2008; Karras and Aila, 2013; Bittner et al., 2013; Gu et al., 2013; Ganestam et al., 2015; Domingues and Pedrini, 2015; Meister and Bittner, 2016). It is also used to estimate and compare the quality of different trees and construction algorithms. A high-quality acceleration structure constructed using the SAH can offer much higher ray-tracing performance than one based on spatial median splits, with typical improvements of 30% or more (Aila et al., 2013; Karras and Aila, 2013; Lauterbach et al., 2009). While high-quality builders used to be very slow, and therefore unsuited to dynamic data sets such as *N*-body simulations, in recent years many new algorithms have been introduced that provide a range of quality and performance options, including iterative approaches that appear highly suited to *N*-body applications. A key insight from modern ray tracing research is that for dynamic scenes the highest quality tree is not always the most useful, as there is a minimum number of queries needed to justify any time spent improving the hierarchy. Karras and Aila (2013) explore this trade-off for a large set of algorithms and scenes. We observe that this trade-off exists for *N*-body algorithms as well, although it is simpler since the number of queries is known. State-of-the art *N*-body implementations spend

E-mail address: o.olsson@uq.edu.au.

https://doi.org/10.1016/j.ascom.2017.11.001
2213-1337/

10 times as much time calculating interactions as on construction of the hierarchy (Bédorf et al., 2014). Thus, we conjecture that there is a large incentive to explore construction algorithms that produce higher quality trees as even just a 10% improvement in traversal performance would justify almost doubling the construction time. Applying these algorithms to N-body simulations could enable a leap in performance and robustness by reducing reliance on hand-tuning parameters. However, to apply these techniques to N-body simulations requires a heuristic that models the cost of a tree with sufficient accuracy. The literature on N-body simulations lacks a well-studied and established heuristic derived to model the cost of N-body evaluations, which currently prevents exploration of high-quality algorithms for ray tracing.

In this paper we introduce a novel heuristic designed to model the cost function of tree traversal using a multipole acceptance criterion. This aims to lay the foundation to open up the large body of research from the computer graphics community for adoption in N-body simulations. We provide a detailed evaluation of the heuristic comparing to state of the art N-body simulations. To date, only one heuristic has previously been proposed for N-body simulations (Kofler et al., 2014) but it was introduced with little analysis or derivation, and we include measurements and comparison to this heuristic in our results. We call our novel heuristic the *Acceptance Volume Heuristic* (AVH) and derive it using a probabilistic cost model based on the properties of the acceptance criteria. To evaluate the effectiveness, we implemented a sweep-based construction algorithm similar to those known to produce very high-quality trees for ray tracing and tested this on a number of data-sets.

To evaluate traversal performance we use the GPU-based traversal implementation from Bonsai (Bédorf et al., 2014, 2012), which offers state-of-the art performance. We show that our most successful metric produces binary trees with a traversal performance that consistently exceeds that of Bonsai, despite using their unmodified traversal kernel. The average speed improvements are around 5% – 10%, depending on distribution and number of particles. While this is a relatively modest improvement, it demonstrates that the approach has potential and that greater improvements are probable with a customized traversal kernel.

1.1. Contributions

We derive and analyse a novel cost metric (heuristic) for constructing and evaluating bounding volume hierarchies for N-body simulations. Our heuristic enables:

- adoption of modern hierarchy construction techniques from computer graphics, opening up access to a large body of high-performance algorithms;
- rough estimation of tree quality, which enables comparison of tree construction algorithms independent of traversal algorithm implementation details;
- a cost based leaf-forming criterion – offering a replacement for current ad-hoc methods of determining the number of particles in each leaf; and
- trees offering improved performance over the current state of the art.

We also present the first evaluation of using binary BVHs for N-body simulations. In a BVH each node is not based on spatial sub-divisions but defines its own bounding volume. Consequently, they are trivial to update without changing the topology of the hierarchy, which makes them highly suitable for evolving data such as simulations, in particular when used with individual time steps where only a small sub-set of particles may change location in each simulation step. Our results show higher performance for a binary tree compared to the octree produced by Bonsai, which indicates that the current practice of building 8-way trees does not yield the best performance.

Additionally we contribute a more thorough investigation of the only previously presented heuristic for N-body simulations, the *Volume–Mass Heuristic* (VMH) (Kofler et al., 2014). Our results show that when used to guide our BVH construction algorithm, this heuristic consistently produces trees with a lower quality and performance than even a simple spatial median split.

2. The acceptance volume heuristic

The broad goal is to derive a function that can provide a metric for the cost of a given tree and, consequently, sub-tree. Such a metric, which is commonly referred to as a heuristic, can then be used to guide tree construction algorithms and to assess and compare the quality of different trees independently of traversal implementation details. Following the approach for the SAH (Surface Are Heuristic) that is used for ray tracing, we want to construct a probabilistic metric such that the cost is the expected value of a random query (Goldsmith and Salmon, 1987; MacDonald and Booth, 1990).

The inspiration and role model for this work, the SAH, is founded on the observation that a given straight line with start and end points outside the domain (scene) has a probability of intersecting any bounding volume, proportional to the surface area of the volume. This is a crude approximation of the true probability of intersection, since the rays typically originate within the scene, have a non-uniform distribution, and terminate at the first intersection. Nevertheless, it has proven very very successful, and though many improved models have been proposed, few have yielded consistent improvements.

N-body simulations differ from ray-tracing in a few important ways. The perhaps most important difference being that in ray-tracing the hierarchy is queried using rays, which interact with the scene only along a line, and thus the efficiency gain from using a hierarchy comes from quickly pruning away the portions of the scene that are not intersected. Crucially, there is no approximation involved. In contrast, a force query for a point in an N-body simulation is omni-directional and of infinite extent. Thus, the entire domain will be covered during the each query, and the efficiency improvements over direct summation is due to *approximation*. An important consequence of this is that for ray-tracing it has been acceptable to ignore the stopping criterion in the cost model, whereas for force calculations this is probably the most important aspect to model.

The model we propose uses three parameters: the relative cost of traversing an internal node, K_t, the cost of a direct force calculation, K_d, and the cost of an approximate force calculation, K_a. These parameters should be set to approximate the performance of the actual traversal code that will be used.

This enables modelling the cost of an internal node, n, conditional on it being traversed, with children $\mathrm{ch}(n)$, as follows:

$$C^N(n) = K_t + \Pr(A_n|T_n) \times K_a + \Pr(\neg A_n|T_n) \sum_{c \in \mathrm{ch}(n)} C^N(c), \tag{1}$$

where $\Pr(A_n|T_n)$ is the probability that node n is approximated given that it has been visited, and $\Pr(\neg A_n|T_n) = 1 - \Pr(A_n|T_n)$ is the inverse. T_n is the event that the node is visited. Similarly, the cost of a leaf node, l, is

$$C^L(l) = \Pr(A_l|T_l) \times K_a + \Pr(\neg A_l|T_l) \times K_d \times N_l, \tag{2}$$

where N_l is the number of particles contained in the leaf.

From the recursive nature of Eq. (1), it follows that the contribution of an arbitrary node to the total cost is can be expressed as follows:

$$G^N(n) = \prod_{a \in \Phi} \Pr(\neg A_a|T_a) \times C^N(n), \tag{3}$$