Parallel Halo Finding in N-body Cosmology Simulations

David W. Pfitzner
Mount Stromlo Observatory,
PB Weston Creek, Weston ACT 2611 Australia,
dwp@mso.anu.edu.au

John K. Salmon
Center for Advanced Computing Research,
Caltech 206-49, Pasadena, California 91125,
johns@cacr.caltech.edu

Abstract

Cosmological N-body simulations on parallel computers produce large datasets — about five hundred Megabytes at a single output time, or tens of Gigabytes over the course of a simulation. These large datasets require further analysis before they can be compared to astronomical observations. We have implemented two methods for performing *halo finding*, a key part of the knowledge discovery process, on parallel machines. One of these is a parallel implementation of the *friends of friends* (FOF) algorithm, widely used in the field of N-body cosmology. The new *isodensity* (ID) method has been developed to overcome some of the shortcomings of FOF. Both have been implemented on a variety of computer systems, and successfully used to extract halos from simulations with up to 256^3 (or about 16.8 million) particles, which are among the largest N-body cosmology simulations in existence.

Introduction

According to current cosmological theory, most of the mass in the universe (e.g., perhaps 90%) is in the form of so-called *dark matter*, whose only significant interaction is gravitational. During the evolution of the universe, this dark matter forms dense objects, called *halos*, due to gravitational instability. Within these halos, the dark matter is supported against further collapse by random motions. The normal matter in the universe collects at the centers of these halos, where star formation leads to the existence of luminous galaxies and other observable phenomena.

A detailed analytic understanding of the evolution of the dark matter is hampered by the highly non-linear nature of the problem, and the complexity of the structures formed. Hence numerical methods have become a very important tool for understanding this evolution, and for comparing cosmological theories with astronomical observations. In N-body simulations, the mass in the universe is represented by a set of N discrete particles, which can be interpreted as a Monte Carlo sampling of the (incredibly more numerous) dark matter particles. Simulations with larger N produce more accurate results, because the sampling is more complete. Larger N also gives larger dynamic range, in terms of the range between the smallest and largest scales which can be addressed in a single simulation. This is important, for example, for investigating small-scale structure in simulations with a volume large enough to sample a fair region of the universe (Zurek *et al.* 1994), or for better resolution of substructure in simulations of clusters of galaxies (Carlberg 1994).

Recently, systems with N as large as 256^3 have been simulated on massively parallel computers. These large simulations produce correspondingly large datasets, posing a challenge for analysis, which has traditionally been done on workstations. Practical considerations like available memory, reasonable turnaround time, and a desire to study time-dependent processes make it increasingly desirable for critical data analysis tasks to run on the same parallel machines that performed the simulations. One such task is *halo finding*: identifying all the isolated collections of gravitationally bound particles, i.e., the dark matter halos. Galaxies are believed to form at the centers of these halos, so once they have been found, their distributions and properties can be compared to astronomical observations of galaxies and galaxy clusters.

The halo finding methods discussed here use data at a single output time. Other methods (Couchman & Carlberg 1992; Carlberg 1994; Summers, Davis, & Evrard 1995) use data from several output times, which is expected to be useful since halos should persist as distinct objects over time, apart from processes such as halo formation, merging, and disruption. However finding halos independently at different individual times should be a useful check of the robustness of the method in finding persistent halos, as well as an objective tool to study the evolution of halos.

The methods have been applied to real data sets obtained from cosmological simulations. Table 1 lists the basic parameters of simulations which have been analyzed and will be referred to later. (Note 1 Mpc = 3.26×10^6 light years.) Models 2, 3 and 4 are subregions extracted from larger simulations.

In the next section we describe the friends of friends

Model	N	Volume
Model 1	16,777,216	(100 Mpc)^3 cube
Model 2	525,002	(20 Mpc)^3 cube
Model 3	8,599	$\frac{4}{3}\pi$ (10 Mpc)^3 sphere
Model 4	1,578,230	(10 Mpc)^3 cube

Table 1: Simulation Parameters

(FOF) halo finding method, and note some shortcomings. This motivates the subsequent description of the new isodensity (ID) method, based on kernel density estimation (McLachlan 1992). Finally we explain how the two methods have been implemented on parallel computers, and present some timing results.

The Friends of Friends Method

In the friends of friends (FOF) method, (Davis *et al.* 1985) one specifies a linking length, h_{link}, and identifies all pairs of particles with a separation of h_{link} or less. Such pairs are designated *friends*, and halos are defined as sets of particles that are connected by one or more friendship relations, i.e., *friends of friends*. The linking length is usefully parameterized as a density, and following (Summers, Davis, & Evrard 1995), we define δ_{min} as the density, divided by the background cosmology density ($\bar{\rho}$), defined by two particles, of average mass, inside a sphere of radius h_{link}.

A second parameter in FOF is the minimum number of particles (N_{min}) in a halo. This is necessary because the particles represent a Monte Carlo sampling of the underlying matter distribution. Because of the essential randomness in the particles' positions, there will be statistical fluctuations in the number of particles in any particular region. Hence it is possible to find groups of friends that do not form persistent objects in the simulation. Obviously, chance associations involving larger numbers of particles are less likely than those involving fewer, so by setting N_{min} sufficiently large one hopes to avoid most of these spurious halos.

Figure 1 show the results of FOF on Model 3, using $N_{min} = 10$, and the left hand panels of figure 2 show the effects of increasing N_{min} to 30. These figures demonstrate two problems with FOF: joining halos together, and poor distinction of small halos from noise. The first problem is that at the center of the cluster, FOF finds one large halo which is clearly composed of at least several distinct halos. This is due to the fact that using FOF, everything in a region where the density is above δ_{min} is joined into a single halo, whether or not the region includes objects which are distinct at some higher density. (This problem was previously noted by (Bertschinger & Gelb 1991).) From the density plots it is seen that there is no value of δ_{min} which will distinguish the halos in the high density region without missing some of the lower density halos.

The second problem is related to N_{min}: The value of $N_{min} = 10$ is seen to be too small, since many of the

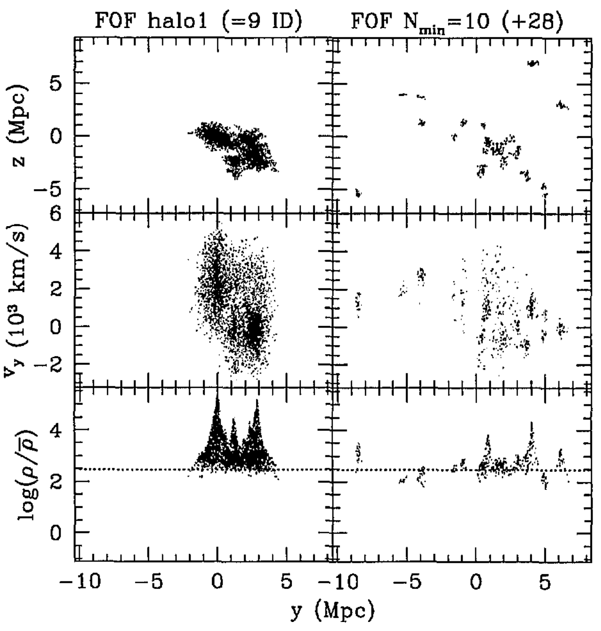

Figure 1: The FOF halo finding results for Model 3. (See text for various details in what follows. The complete model is shown in the left panels of Figure 3.) The upper panels show particle positions, projected into an arbitrary plane; the middle panels show one spatial coordinate and one velocity coordinate; the lower panels show the density (as calculated by ID) against one spatial coordinate. The horizontal line in the lower panels indicates a density of 350 $\bar{\rho}$, corresponding to the δ_{min} value used for the FOF identification. The panels on the left show the particles in the single most massive halo found by FOF; those on the right show the particles in the other FOF halos, using $N_{min} = 10$.

small halos found have high internal velocity scatter (and hence are not bound), in contrast to others which are clearly distinct, compact objects in velocity space. At a higher N_{min} of 30, most of the spurious halos are rejected, but one remains, and in addition several real (but small) halos have been rejected.

We suggest here a simple improvement to N_{min}: Take as a parameter some minimum number of friends (i.e., direct links), N_{fmin}, and only accept halos which have at least one particle with at least N_{fmin} friends. The advantage of N_{fmin} over N_{min} is that diffuse, relatively low-density linked groups (possibly with many particles), are rejected, while isolated tight clumps (still with at least N_{fmin} particles, but possibly less than N_{min}) are accepted as real halos. The effects of N_{fmin} in Model 3 are shown in the right hand panels of figure 2, using $N_{fmin} = 10$. In this case all of the spurious halos are rejected, and more real halos are accepted than for $N_{min} = 30$.

One could consider alternative criteria for rejecting

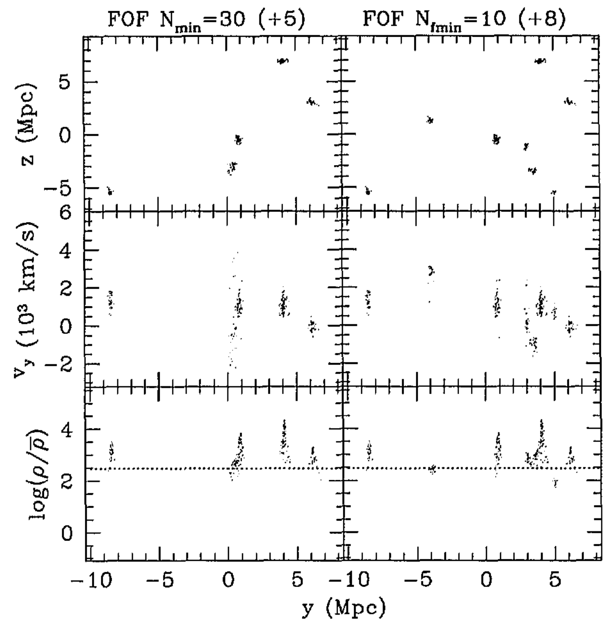

Figure 2: The same as the right hand panels of figure 1, but with alternatives to N_{min} = 10: on the left N_{min} = 30, and on the right N_{fmin} = 10.

spurious halos, such as using actual particle velocities to calculate whether putative halos are gravitationally bound. It is probably most productive to perform these procedures in a later stage of analysis when the halos are actually studied, cataloged, correlated, compared to observational data, etc.

The Isodensity (ID) Method

The main aim of the isodensity (ID) method is to improve on FOF by identifying halos over a wide range of densities, thereby exploiting the full dynamic range available in a simulation. In some respects the method corresponds to applying the FOF method at a range of densities or linking lengths, a course suggested by (Davis *et al.* 1985), but in an integrated and consistent way.

The idea of ID is to calculate a spatial density field defined by the particles, and identify halo centers as local peaks in this field. Isodensity surfaces are then grown around each center, to find those particles belonging to each peak. When the isodensity surfaces of different centers touch at some density, then one instead considers their common isodensity surface. As mentioned regarding FOF, in any local spatial region there will be statistical fluctuations in the number of particles in the region, which will lead to noise in the calculated density field (relative to the hypothetical underlying mass distribution being sampled by the N-body particles). The ID method rejects density peaks which are merely noise peaks by using an estimate of the uncertainty in the calculated density field.

The ID method is related to the DENMAX algorithm (Bertschinger & Gelb 1991; Gelb & Bertschinger 1994), especially in terms of motivation, but differs in most details. (Zurek *et al.* 1994) also use a similar method, but they only consider spherical isodensity surfaces.

The density field calculated from the particles should satisfy two conditions. First, the density should be in some sense spatially continuous, in that nearby particles should generally have similar densities. This condition is necessary for the interpretation of the method in terms of isodensity surfaces. Second, the statistical uncertainty of each estimate must be computable so that chance associations can be reliably rejected.

The density at each particle, ρ_i, is calculated as the sum of the masses, m_j, of the nearest N_{kern} particles, divided by the volume of the sphere enclosing those particles. This is a form of kernel density estimation,

$$\rho_i = \sum_j m_j \, k(r_{ij}, h_i) \qquad (1)$$

using the nearest neighbor method to set the kernel smoothing scale, h_i, and a kernel function, k, with uniform density. A variable smoothing scale is important because of the large range in densities present in the simulations. By using the nearest neighbor method with, e.g., N_{kern} = 24, the local resolution in the density is tailored to the actual resolution available, in terms of the local number of particles.

The uncertainty in the calculated density can be estimated by assuming that the underlying density distribution is roughly uniform on scales that contain N_{kern} bodies, and that the particle positions are sampled at random from this density field. Then the uncertainty in the density is just due to Poisson noise, and the dispersion, σ, is simply $1/\sqrt{N_{kern}}$ times the density. For non-uniform kernels the situation is more complicated. We have experimented with alternative kernel functions, but find that for the same level of uncertainty in the density field, they are more computationally expensive.

In principle the ID method could use alternative density measures, so long as the requirements of continuity and known uncertainty are satisfied. One possibility is the phase space density: the mass per spatial volume element per velocity volume element. This may be advantageous for cosmology simulations, because low density halos generally have small internal velocities, (e.g., see figure 3) and hence have similar phase space densities to halos with higher spatial density.

The isodensity surfaces are defined implicitly by calculating for each particle a linking length, h_{link}, such that each particle links to precisely N_{link} spatial neighbors. Then, taking all the particles above some density, each group of linked particles (cf FOF), is considered to be surrounded by a single isodensity surface. The value of N_{link} should be large enough that at zero density, all particles are linked together, but not unnecessarily large since this would compromise the spatial resolution of the method. In practice values of 12 to 24 have been used.

We first present a simplified version of the ID method: One first sorts the particles by their density, and then considers each particle in turn in order of density from highest to lowest. Each particle is assigned a halo number, to specify which halo it belongs to. The halo number for each particle is calculated based on the halo numbers of the higher density particles to which it is linked, as follows: If there are no linking particles, the particle is given a new, unused halo number. If the linking particles all have the same halo number, the particle gets that number. Otherwise, the linking particles have different halo numbers. In this case the halos corresponding to those halo numbers are said to overlap at the density of the particle being considered. A new halo number is generated to represent the overlap of those halos, and for future halo linking purposes, particles with those old numbers are taken to have the new halo number, so that the new halo number can in turn participate in overlaps. However the original numbers are recorded so that the previously distinct halos are still identified as such. To keep track of these overlaps, a tree of halo numbers is constructed, where the leaves correspond to the central regions of distinct halos, and the internal nodes correspond to regions, defined by isodensity surfaces, where various halos overlap.

The first modification which is made to this simplified method is to take account of the known uncertainty in the density field to reject noise peaks. If the halo central (peak) density (ρ_c), is less than $n\,\sigma$ above the overlap density (ρ_o), with σ calculated at ρ_o, then the smaller halo is rejected, and joined into the larger halo. The value of n is a parameter of the method; an appropriate value (typically 3 to 4) can be determined by examining test cases (such as Model 3) in detail. The motivation for the above condition is that if it fails, then there is a reasonable probability (although not rigorously defined here) that the peak is a noise peak in some local region with mean density of approximately ρ_o.

A second modification is made to the simple method to improve computational efficiency and facilitate parallelism. Instead of considering each particle in turn in order of density, one makes discrete density cuts, and the particles above each cut are worked on at the same time in a consistent way. The cuts are made so that they are small compared to the uncertainty in the density, so that the consequences of this modification on the results should be small.

Figure 3 show results obtained using ID. In this particular case *all* of the halos found by ID turn out to be real, in terms of their internal velocities (when examined individually). Also, ID distinguishes halos even in the high density central region of the cluster — the single large central halo found by FOF is split into 9 smaller halos by ID. One halo found by FOF (with $N_{fmin} = 10$) is not found by ID; this is because it has fewer particles than $N_{kern} = 24$ used by ID in this case.

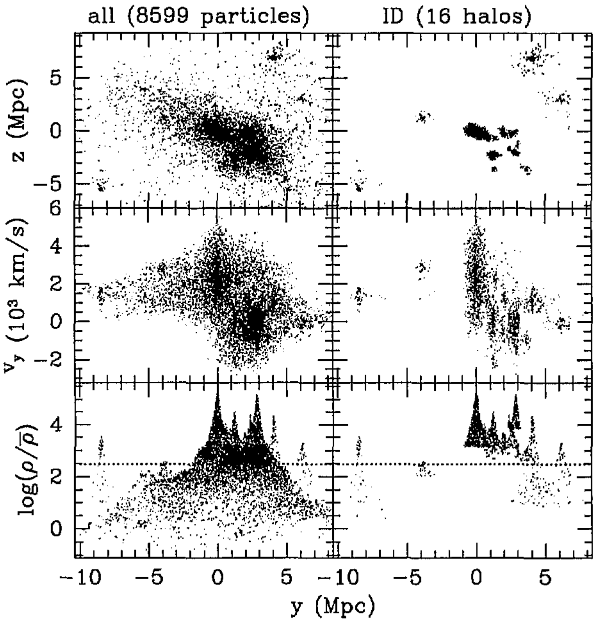

Figure 3: As in in figures 1 and 2. The panels on the left show all the particles; those on the right show the particles in the non-overlapping inner region of each halo found by ID.

In general the results of ID and FOF with N_{fmin} = 10 are rather similar, as shown in figure 4. The main noticeable systematic difference is in high density regions, where ID finds more halos. In this case FOF with N_{fmin} actually found more halos in total than ID, again because with the parameters used, FOF can detect halos which have fewer particles.

Implementation

We implement ID as a series of *linking passes* which will be defined shortly. Linking passes are used to calculate the particle kernel scale and linking lengths, then the particle densities, and finally the particle halo numbers. In the last step, successive density cuts are made in which particles without halo numbers are iteratively linked to those with halo numbers, and new halos are identified from local density peaks that pass the noise criterion. With many density cuts and several iterations on each cut, this step involves many individual linking passes, but executes quickly nonetheless because each pass involves relatively few particles.

A linking pass is defined as follows: Given two subsets of the particles, called sinks and sources, then for each sink, one finds all the sources within some distance of that sink, where the distance may depend on the sinks and sources. For each sink and its list of

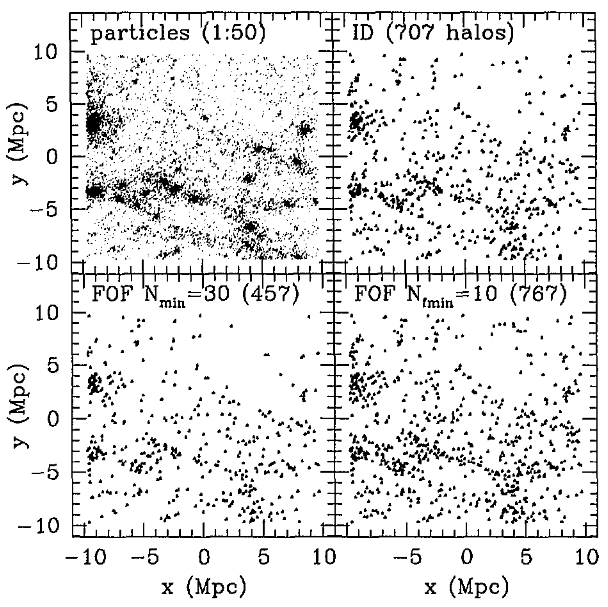

Figure 4: Comparison of particles and halos from ID and FOF (projected into an arbitrary plane). (From Model 2.)

linking sources, some calculations are performed, and information in the sink is updated. Because the density of particles is so irregular, an adaptive oct-tree is used to identify the candidate sources quickly. The neighbor-finding procedure is almost identical to that described in (Warren & Salmon 1995) in the context of smooth particle hydrodynamics (SPH). In fact, a large portion of the code is in a common library which handles all data structure manipulation and explicit communication on message passing parallel architectures. The library was originally designed to implement parallel "Fast" summation algorithms, e.g., (Barnes & Hut 1986) for the computation of gravitational interactions in $O(N \log N)$ time, but the data structures are far more general, as evidenced by this paper. The library distributes the data so that the sinks are uniquely assigned to processors, while read-only copies of the source data are transmitted and stored, on demand, on multiple processors. These two conditions completely eliminate any coherence problems associated with communication and storage and greatly simplify the programming. The libraries (and hence any applications that use them) have been ported to a wide variety of systems including message-passing and shared-memory machines, as well as networks of workstations and uniprocessor systems. In particular, the FOF and ID methods described here have been tested on single and multi-processor SPARC workstations, a 32-node CM-5 at the Australian National University and a 512-node Intel Paragon at Caltech.

Friends of friends can be implemented almost as a special case of ID, using the number of friends as a measure of density. Other simplifications include not keeping track of halo overlaps or noise estimations. Furthermore, density cuts are not restricted as they are in ID — they can instead be based on efficiency considerations. It is best to minimize the number of distinct density cuts, subject to the condition that for each cut a large fraction of the particles link on the first iteration.

Results

Figure 5 shows the halos found by ID in Model 1. The halo finder ran for approximately 75 minutes on a 512 node Paragon at Caltech, and required over 5 Gigabytes of memory. This computation would have been prohibitively time-consuming on a uniprocessor system – assuming we could have found one with sufficient memory!

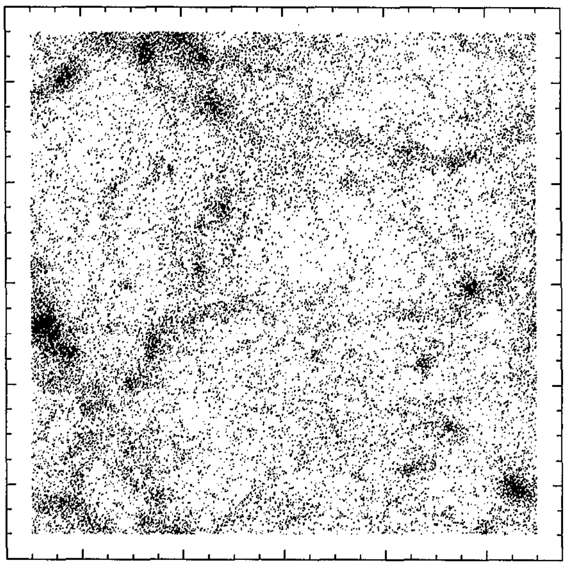

Figure 5: All 29,729 halos from Model 1, as found by ID, projected into an arbitrary plane; the box is 100 Mpc on a side.

Figure 6 shows some additional timing results from halo finding on a Paragon. The density calculation shows very good scaling, with the CPU time per processor, per number of particles on that processor, roughly constant. (The results for the density calculation for model 1 with 512 processors are not available, but are expected to be of the same order.)

The scaling is less good for the step involving density cuts and halo number calculation: when the number of processors is large, increasing the number of processors and keeping the total number of particles fixed does not reduce the time per processor. This is likely due to the

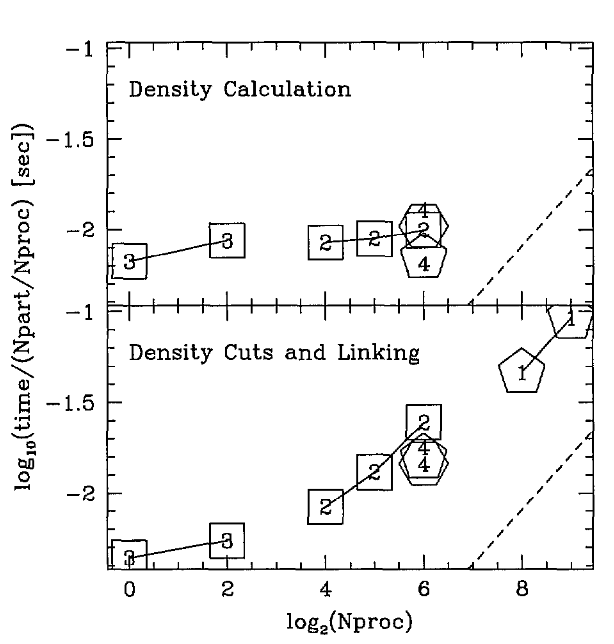

Figure 6: Timing results for the ID method. The *time* is the average CPU time on each processor, *Npart* is the total number of particles, and *Nproc* the number of processors. The different point shapes indicate somewhat different parameters, (for which the results may not be totally comparable), and the numbers correspond to model numbers from table 1 (in particular, different Npart). The dashed line indicates the slope for constant time independent of Nproc.

fact that this part of the method involves many linking passes with relatively few particles in each pass, and the small linking passes parallelize less well than large ones. Hence for this step one should use as few processors as possible, within limits set by memory constraints. In general, halo finding is memory limited rather than CPU-time limited, as are the simulations themselves. Fortunately, the linking passes with the poorest parallel efficiencies require less memory so fewer processors can be used, compared to the density calculation step.

Conclusions

We have presented a new isodensity (ID) algorithm for finding halos in N-body cosmology simulations, and described an implementation on parallel computers. This new method has advantages compared to the friends of friends (FOF) algorithm, which has also been implemented in parallel. In particular the ID method robustly finds density peaks even in the high density central regions of clusters. Our tests indicate that these halos are "real" in the sense of being gravitationally bound, persistent objects in the simulation, so the ID method is a genuine knowledge discovery process. The use of a statistical estimate of the uncertainty in density estimation to distinguish real peaks from chance associations is novel and effective, but lacks a firm theoretical foundation.

By implementing these methods on parallel machines we are able to use them to begin the analysis of the massive datasets produced by modern high resolution N-body cosmology simulations. This will allow us to address the task of accurately interpreting these simulations, to understand the physical processes involved in the formation and evolution of dark matter halos, and to compare the simulations to astronomical observations.

Acknowledgments

This research was performed in part using a Paragon supercomputer operated by the Center for Advanced Computing Research at Caltech; access to this facility was provided by Caltech. We also acknowledge computing resources provided by the Australian National University Supercomputer Facility. DWP acknowledges support from an Australian Postgraduate Award, and a Caltech Graduate Research Assistantship.

References

Barnes, J. E., and Hut, P. 1986. A hierarchical O(NlogN) force-calculation algorithm. *Nature* 324:446–449.

Bertschinger, E., and Gelb, J. M. 1991. Cosmological N-body simulations. *Computers in Physics* 5:164–179.

Carlberg, R. G. 1994. Velocity bias in clusters. *Astrophysical Journal* 433:468–478.

Couchman, H. M. P., and Carlberg, R. G. 1992. Large-scale structure in a low-bias universe. *Astrophysical Journal* 389:453–463.

Davis, M.; Efstathiou, G.; Frenk, C. S.; and White, S. D. M. 1985. The evolution of large-scale structure in a universe dominated by cold dark matter. *Astrophysical Journal* 292:371–394.

Gelb, J. M., and Bertschinger, E. 1994. Cold dark matter 1: The formation of dark halos. *Astrophysical Journal* 436:467–490.

McLachlan, G. J. 1992. *Discriminant Analysis and Statistical Pattern Recognition.* Wiley series in probability and statistics. Applied probability and statistics. John Wiley & Sons, Inc.

Summers, F. J.; Davis, M.; and Evrard, A. E. 1995. Galaxy tracers and velocity bias. *Astrophysical Journal* 454:1–14.

Warren, M. S., and Salmon, J. K. 1995. A portable, parallel particle program. *Computer Physics Communications* 87:266–290.

Zurek, W. H.; Quinn, P. J.; Salmon, J. K.; and Warren, M. S. 1994. Large scale structure after COBE: Peculiar velocities and correlations of cold dark matter halos. *Astrophysical Journal* 431:559–568.