THE ASTROPHYSICAL JOURNAL, 497:534–540, 1998 April 20

A SIMPLE VOID-SEARCHING ALGORITHM

J. AIKIO AND P. MÄHÖNEN
Department of Theoretical Physics, University of Oulu, Linnanmaa, Finland; and Technical Research Center of Finland, P. O. Box 1100, 90571 Oulu, Finland
Received 1997 June 26; accepted 1997 December 1

ABSTRACT

We have developed a new automatic void-searching algorithm for three-dimensional redshift surveys and N-body simulations. We strictly define the condition that is called a void. This definition is used to make a new and automatic algorithm that finds voids from particle distributions. We introduce our algorithm and show with test simulations that it is a robust tool for automatic void searching and void statistics calculations.

Subject headings: galaxies: statistics — large-scale structure of universe — methods: numerical

1. INTRODUCTION

The discovery of large voids from galaxy and cluster redshift surveys is an important discovery for theories about the origin of the structure of the universe (Zeldovich, Einasto, & Shandarin 1982; Oort 1983; Geller & Huchra 1989; da Costa et al. 1994; Einasto et al. 1994, 1997). In particular, recent claims about the 120 Mpc periodicity in the distribution of galaxy superclusters have alleviated the need to study void-supercluster statistics more closely and carefully than before (Broadhurst et al. 1990; Kaiser & Peacock 1991; Einasto & Gramann 1993; Einasto et al. 1997).

The major goal of the automatization of void searching is to derive a uniform and objective picture of the void structure of different particle distributions. One of the major difficulties is that we need to know exactly what we are looking for. There are several accepted definitions for a void and various methods for void searching; Einasto, Einasto, & Gramann (1989) use the "empty sphere method" to approximate the minimum diameters of voids. Ryden (1995) and Ryden & Melott (1996) search elliptical empty regions in two-dimensional particle distributions varying the shape and direction of the ellipses. Kaufmann & Fairall (1991) used an algorithm whereby one puts the largest possible cube into the galaxy distribution so that no galaxies are inside it and then adds smaller rectangular volumes outside the parent cube until the void is completely mapped. El-Ad, Piran, & da Costa (1996) and El-Ad & Piran (1997) used an algorithm whereby the voids consist of connected spheres with variable radii $r < r_{\min}$. The latter two algorithms basically can find all arbitrarily shaped voids, but in order to prevent the voids from joining together by thin connecting tunnels (fingers), one has to (arbitrarily) limit the minimum radius of the sphere or the cube used for finding.

In this paper we will be mathematically defining voids and a new void-searching algorithm more precisely than before. In § 2 we give a formal definition of voids, and in § 3 the new void-searching algorithm is described. In §§ 4 and 5 we discuss the resulting data and possible variations of the algorithm, respectively. We describe the simulated data and analysis of it in §§ 6 and 7. Finally, in § 8 we give our conclusions.

2. THE DEFINITION OF A VOID

Let the distribution of particles be placed for simplicity in a cubic region $L^3 \in \Re^3$ with side length L. First, following closely the "empty sphere method" of Einasto et al. (1989), we define a scalar field $D: L^3 \rightarrow R$ as the distance of a given point $\boldsymbol{x}$ to the *nearest* galaxy, thus

$$D(\boldsymbol{x}) = \min_n \{ |\boldsymbol{x} - \boldsymbol{X}_n| \} , \quad (1)$$

where $\boldsymbol{X}_n$, $n = 1, \ldots, N$ are the locations of the particles. We call this field a *distance field* (DF). The local maxima of DFs are the points in L^3 with the longest distance to the nearest particle. They are the "centers" of empty regions (cf. Einasto et al. 1989).

Let $M \in L^3$ be a local maximum of the DF. We define a *subvoid* $v_M \subset L^3$ as a region around M as follows: Point $\boldsymbol{x} \in L^3$ belongs to the subvoid v_M if and only if when starting from $\boldsymbol{x}$ and moving continuously in the direction of the highest ascent of the DF (i.e., in the direction of the derivative ∇D when defined) we end up at the maximum M. This is a good definition because for almost every point $\boldsymbol{x} \in L^3$ there is a unique maximum M such that $\boldsymbol{x} \in v_M$. As an extra condition we set a small distance $D_{\min}$, which defines the minimum distance between the point $\boldsymbol{x}$ and a particle, so that the point can be a member of a subvoid. The latter is merely a computational need to avoid particles belonging to a void. Parameter $D_{\min}$ can be arbitrarily small.

The upper definition of a subvoid leads to the interpretation that it is a simple, connected, compact region around a maximum of a DF. It does not have thinner parts between thicker ones; it does not form complex coral- or spongelike structures. The galaxies (particles) exist only on the outer boundaries of subvoids. The shape of a subvoid is completely determined by the distribution of particles and is not predefined.

Although a subvoid clearly belongs to one intuitively determined void, the subvoid does not necessarily cover the whole void volume. There may be more than one maximum of a DF inside a void, with approximately the same values for the DF (cf. Linder et al. 1995). We have to continue the definition process to get results more like the intuitive ones.

Let the distance between maxima M_1 and M_2 be $d_{1,2} =$

$|M_2 - M_1|$. We have successfully used the following unification of subvoids as a definition for a void: Subvoids v_{M_i} and v_{M_j} belong to the same void V if (1) the maxima M_i and M_j are nearer to each other than the nearest particle, i.e.,

$$d_{i,j} \leq \min \{D(M_i), D(M_j)\} , \quad (2)$$

where $D(M_k)$ is the value of the DF at point M_k, or (2) if there exists a chain of subvoids $\{v_{M_k}\}_{k=1}^{n}$ such that $d_{i,1} \leq \min \{D(M_i), D(M_1)\}$, $d_{k,k+1} \leq \min \{D(M_k), D(M_{k+1})\}$ for all $k = 1, \ldots, n-1$ and $d_{n,j} \leq \min \{D(M_n), D(M_j)\}$ (i.e., there exists a chain of subvoids obeying the condition in eq. [2] between the two subvoids). Hence, the voids are defined here as a kind of "friends-of-friends" percolation of subvoids.

The definition of a void is illustrated in the left panel of Figure 1; the (two-dimensional) particle distribution (*big dots*) forms four voids (bounded by lines). The local maxima of the DFs ("centers" of subvoids) are marked with small dots and the "main maxima" of each void—the maxima of the DFs inside a void—are marked with crosses. The topmost void consists of four subvoids. Circles with radius $D(M_i)$ are drawn around the two maxima M_1 and M_2. Because the maxima are inside each other's circles, the apparent subvoids belong to the same void. The boundaries of voids in the picture are not exact. They are found by the algorithm, which will be described next.

3. THE ALGORITHM

Numerically, the DF is calculated by defining a cubical mesh over the survey volume (cf. Einasto et al. 1989). We divide the cube L^3 into k^3 *elementary cells*, where $k = L/s$ and s is called a *resolution parameter*. It tells the spatial resolution of the void analysis. For each elementary cell center we calculate the minimum of distances to the particles. We get a discrete DF $D(\boldsymbol{x})$, from which it is easy to find the local maxima. Let us call the apparent maximum cells M_i. (In fact, the discrete DF should be smoothed before the search for maxima to avoid some numerical artifacts.)

In practice it is easier to organize the DF maxima into single-void groups before the search for subvoids and hence go straight to searching for voids. The maximum groups can be found, for example, by using the multiple labeling technique of Hochen & Kopelman (1976) with equation (2) as the joining condition. After this we know which voids each of the elementary cells M_i belongs to. Now all we have to do to map the voids is to divide the rest of the elementary cells x among the voids. This is done by using the "climber algorithm." The void to which an elementary cell x belongs is found by "climbing" from x to the neighboring cell of x, say x', that has the highest value in the DF. The climbing continues from x' to the neighboring cell x'' that again has the highest value in the DF. The stopping condition for climbing is satisfied when a maximum cell M_i is achieved. The cell x now belongs to the same void as M_i. Because of the way the climbing is done, all the elementary cells along the climbing path also belong to the same void as x. So we have concurrently determined the void for them too.

It is not always necessary to climb all the way up to the maximum cell. If the climbing ends up in a cell that has already joined into a void, we can just join the cells along the climbing path to the void. That is because we know immediately where the climbing will lead us. Also there is a possibility of ending up in the boundary of the volume L^3. If we are interested only in the interior voids of L^3 we ignore

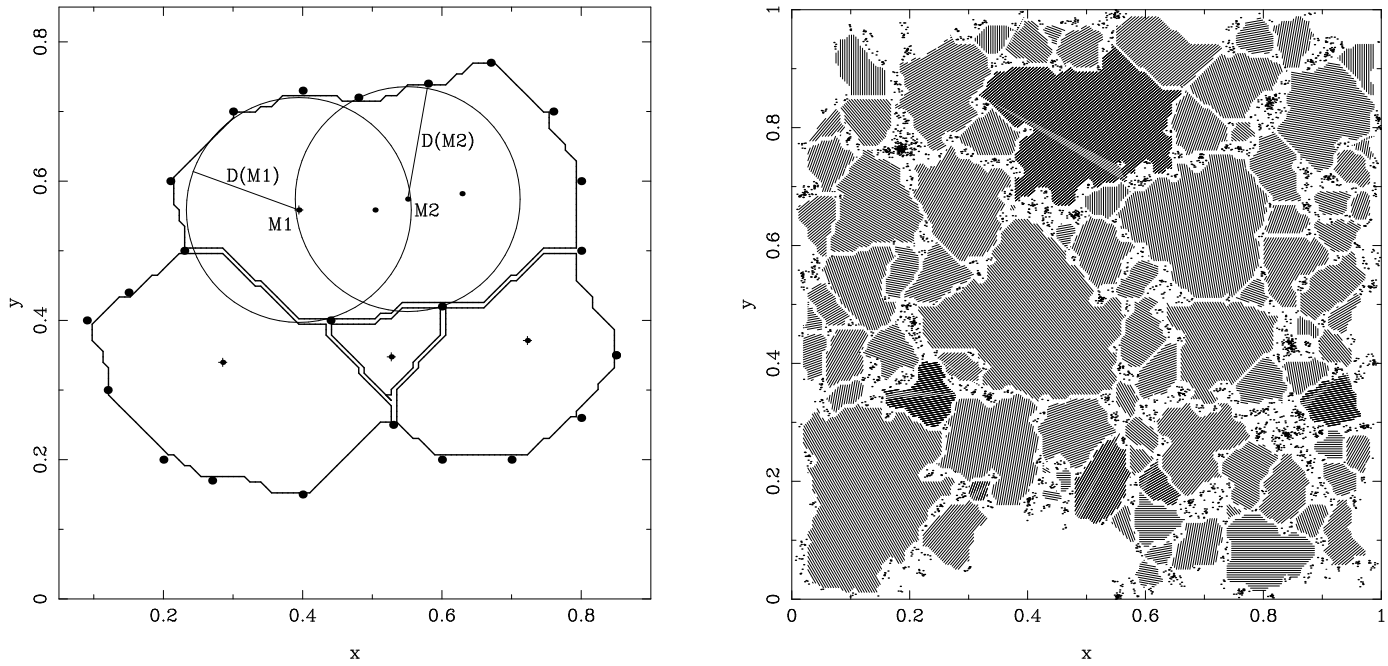

FIG. 1.—(*Left*) Example of the two-dimensional particle distribution with four voids. The maxima of the DF are shown by dots. Crosses denote the main maxima of voids. The condition for joining two subvoids with centers in M_1 and M_2 into one void is illustrated by drawing the circles with radius $D(M_i)$, $i = 1, 2$ around the apparent maxima. The two subvoids belong to the same void, because the distance between the maxima M_i is smaller than the distances $D(M_i)$. Since M_1 is the main maximum of DF inside the void, the radius $D(M_1)$ is (following the definition in § 5) the main radius of the void. In § 7 we approximate the size of the void by a sphere with radius equal to the main radius of the void. It can be seen that here the $D(M_1)$ circle clearly underestimates the size of the uppermost void. However, the size of the small triangular void in the middle would be overestimated with the method, as can be seen by drawing the apparent R_V circle to the DF maximum of this void. (*Right*) Voids found from the two-dimensional mock galaxy catalog.

this path, or in practice join the elementary cells along the climbing path into a "null void." We may also want to ignore the elementary cells that have particles inside them. This is done by checking the value of the DF. If $D(x) < s/2$ [or $s/(2)^{1/2}$] where s is the resolution parameter, we join x to the null void. So we choose $D_{\min} = s/2$.

In order to find all the voids, we have to start from every elementary cell x one at a time, climb, and see where we end up. If the starting cell is already a part of a void we just start from the next free elementary cell. Hence, the voids we find are collections of elementary cells.

4. THE RESULTING DATA

The results that the algorithm directly gives are the number and locations of the elementary cells in each void. The volume V of the void V is the number of cells in the void, n_V, multiplied by the volume of elementary cells, s^3. The main radius of the void, R_V, is defined as the maximum of the DF inside the void, i.e., it is the maximum of the DF maxima $D(M_k)$ inside the void:

$$R_V = \max_{M_k \in V} \{D(M_k)\} . \tag{3}$$

The corresponding elementary cell is called the *main maximum* of the void M_V. The main radius defines the radius of the *central* (or *main*) *sphere* of voids.

Another useful statistical measure of size is the (largest) diameter d_V of voids. We define the diameter of a void as the maximum of distances between two cell centers in the void added with the resolution parameter s. The adding of s corrects an error caused by using cell centers as the ends when measuring distances. The correction is especially necessary in the case of small voids. The diameter also defines the largest distance between two galaxies bounding the void. The locations of voids can be given by M_V or by the "center of mass" $X_V = 1/n_V \sum_{x \in V} x$, where the x are the locations of elementary cells.

The difference between d_V and r_V can be seen in Figure 1. The d_V as the maximum distance between two cell centers in the void is a kind of extreme size of the void. On the other hand, the main radius r_V is a more conventional measure of the void size. However, one should note that the radius R_V is not the radius of the largest sphere inside the void (see Fig. 1), although conceptually this is a good way to think about it. Both radii can be used to approximate the volume of void, but both give only an estimate, as will be discussed later.

The shapes of voids can be approximated as ellipsoids by using the method of inertia tensors. The inertia tensor $M_{ij} = \sum_{x \in V} x_i x_j$ is diagonalized, where x_i, x_j $(i, j = x, y, z)$ are the coordinates of an elementary cell x in the "center of mass" coordinates of the void V. The axial ratios of the ellipsoid are given by $b/a = (M'_{yy}/M'_{xx})^{1/2}$ and $c/a = (M'_{zz}/M'_{xx})^{1/2}$, where $a > b > c$ are the lengths of the axles and M' is the diagonalized inertia tensor.

The void probability function $\phi(R)$ is directly determined by checking the relative number of elementary cells having a value in the DF smaller than R. So $\phi(R) =$ [number of elementary cells x with $D(x) < R$]$/k^3$.

5. VARIATIONS OF THE METHOD

When dealing with dense particle distributions in particular, it is sometimes convenient to define a void as an underdense region in space. In such cases the algorithm can be boosted with a wall-finder algorithm (cf. El-Ad et al. 1996; El-Ad & Piran 1997). Before using the void-searching algorithm we can identify so-called wall galaxies—the galaxies that mainly form the walls and filaments in the basic wall-void structure. The remaining field galaxies are removed from the distribution so that the underdense regions remain totally empty and are recognized as voids by our algorithm. A wall galaxy is defined as (El-Ad et al. 1996; El-Ad & Piran 1997) a galaxy that has n other wall galaxies within a sphere of radius l around them. The adding of the wall finder brings two extra parameters to the void-finding system, that is the "number of neighbors" n and the "linking length" l. The values of the parameters will lead to some qualitative differences, e.g., in the distribution of the sizes of voids.[1]

A similar kind of result, not taking lonely particles into account, can also be achieved by choosing another scalar field to climb. If we replace the distance field $D(x)$ by the field $D_n(x)$ that tells the distance to the nth nearest particle, where $n = 2, 3, \ldots$, then the algorithm does not notice lonely particles or small enough groups. The bigger the n the denser are the structures that count in the void-finding process. The DF can also be replaced by some other field that depends on the mass distribution of particles. The voids can be found using, for example, the gravitational field, i.e., the less particles there are the less they gravitate. The density field smoothed by a Gaussian function could also work. If the "climbing" direction is turned toward the mass concentrations, for example, in the density field case, the climbing algorithm may also be used for group finding. Some promising preliminary analyses with these scalar fields other than the DF was done, and they will be treated in depth in forthcoming papers.

6. APPLICATION TO SIMULATED DISTRIBUTIONS

We have tested our algorithm with a two-dimensional test distribution of particles and with simulated three-dimensional "galaxy" distributions. Our goal is simply to show that our algorithm works robustly. We leave the proper treatment of a large number of simulations and true redshift catalogs (Abell-ACO, CfA) for a forthcoming paper.

The two-dimensional case helps the visual judgement of the results of the algorithm. The two-dimensional distributions were made by projecting parts of the following three-dimensional simulations and using the wall-finder algorithm to create large empty regions. The right panel of Figure 1 shows the voids found by the algorithm in one of the two-dimensional test distributions.

The three-dimensional N-body simulations have been done with the AP^3M code (Couchmann, Thomas, & Pearce 1995), which was used as a simple P^3M code with 100^3 particles in a 256^3 mesh. We have studied three different realizations. The physical size of the simulation box was 300 h^{-1} Mpc. We used the $\Omega = 1$ standard cold dark matter (CDM) model power spectrum for simulations using the primordial density fluctuation (PDF) function of Bond & Efstathiou (1984):

$$P(k) = \frac{k}{\{1 + [ak + (bk)^{3/2} + (ck)^3]\}} , \tag{4}$$

[1] We would like to point out here that this kind of classificational division of galaxies into walls and field galaxies is basically a technical one, which is done to empty the underdense parts of the space and to make the void finding easier. It should not be confused, for example, with the division made by Turner & Gott 1975, which is more physical in nature.

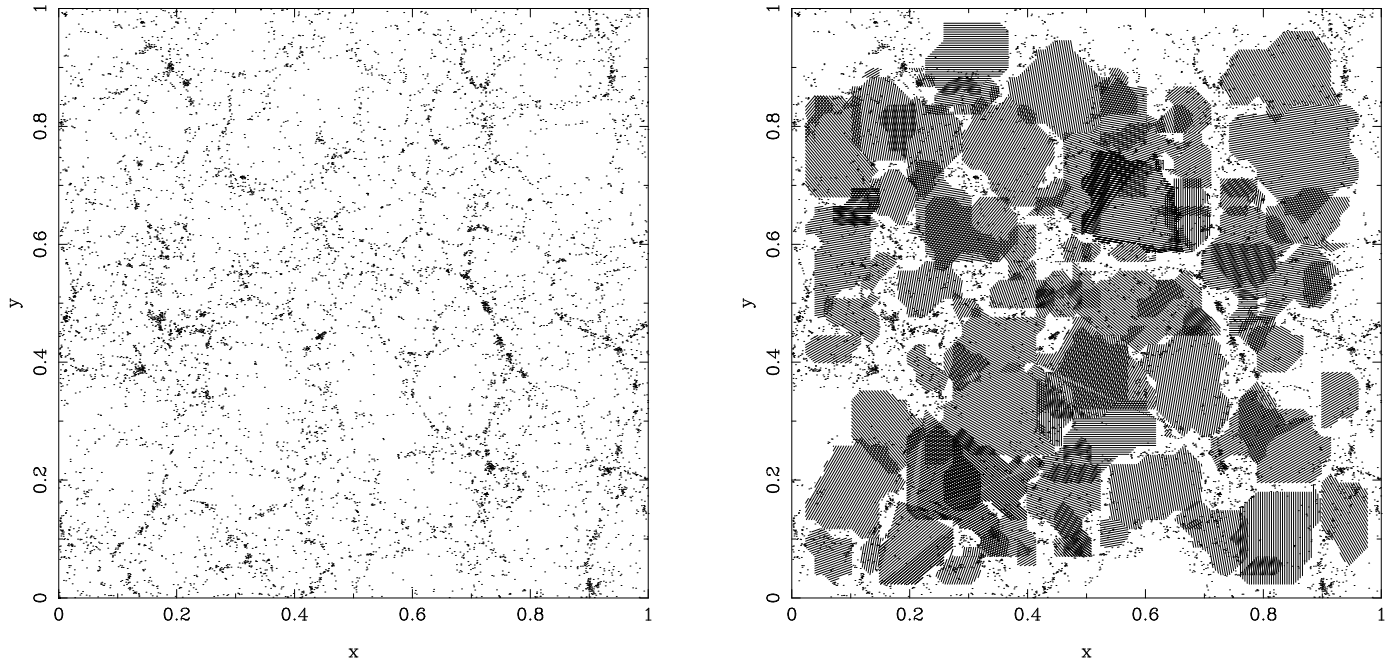

FIG. 2.—(*Left*) Dense particle clumps ("galaxies") found from the typical real-space three-dimensional simulation. Only the 30 h^{-1} Mpc thick slice is shown. (*Right*) Voids and wall galaxies found from the same simulation.

where $a = 6.4(\Omega\ h^2)^{-1}$, $b = 3.0(\Omega\ h^2)^{-1}$, $c = 1.7(\Omega\ h^2)^{-1}$, and $v = 1.13$. Two different epochs were studied, one with a density fluctuation variance within the 8 h^{-1} Mpc sphere σ_8 of 1.0 and another of 0.69.

The galaxylike clumps were found by the friends-of-friends algorithm (Einasto et al. 1984), and that algorithm gave us approximately 100,000 "galaxies" within a box. The wall galaxies were searched with parameters $n = 3$ and $l = \bar{l}_n + 1.5\sigma_n$, where $\bar{l}_n$ is the mean distance from a galaxy to its third nearest neighbor and σ_n is the variance of the l_n. The number of wall galaxies was about 10% less than the original number of galaxies.

The voids were found in real space using 64^3 elementary cells. The number of voids in the simulation box was about 550 ± 20 for $\sigma_8 = 1.0$ and 680 ± 20 for $\sigma_8 = 0.69$. The sizes of the voids were $(24 \pm 24) \times 10^3 (h^{-1}\ \mathrm{Mpc})^3$ for $\sigma_8 = 1.0$ and $(18 \pm 17) \times 10^3 (h^{-1}\ \mathrm{Mpc})^3$ for $\sigma_8 = 0.69$.

In the left panel of Figure 2 we show the galaxies in a 30 h^{-1} Mpc thick slice of one of the simulations at $\sigma_8 = 1.0$. In the right panel are the voids found from the same slice

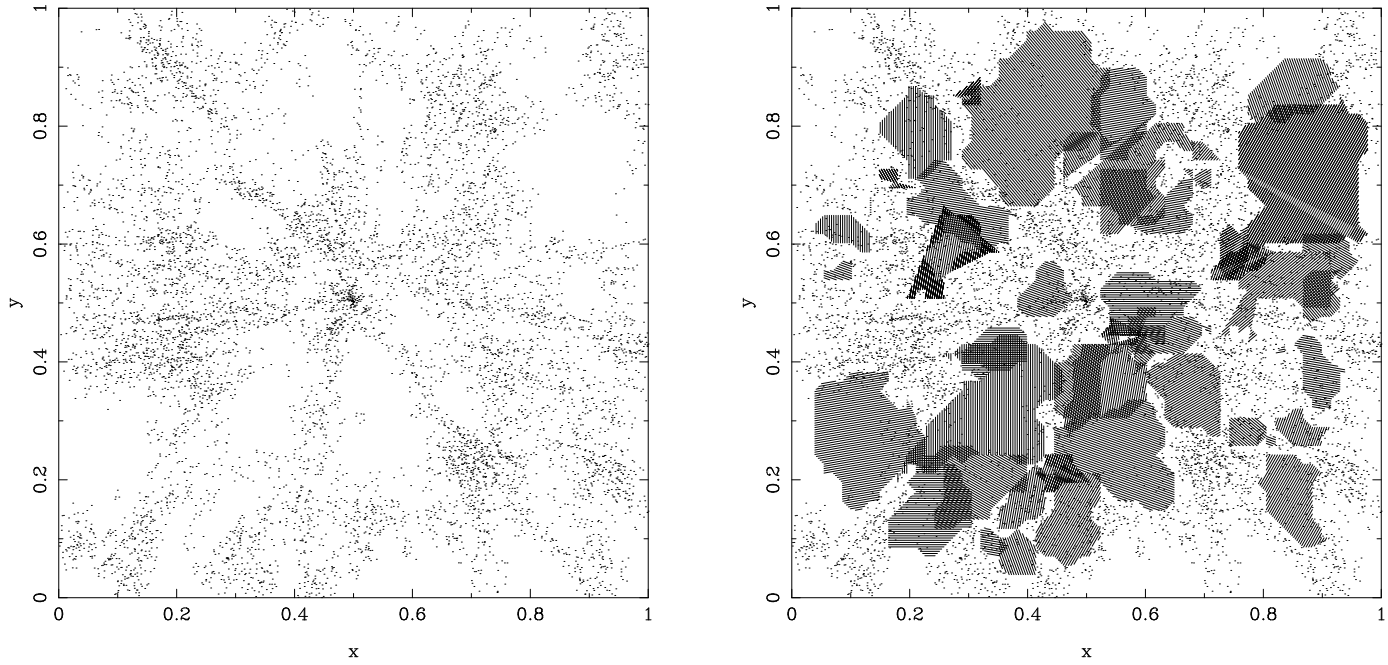

FIG. 3.—(*Left*) Galaxies found from the typical redshift-space three-dimensional simulation. Only the 30 h^{-1} Mpc thick slice is shown. (*Right*) Voids and wall galaxies found from the same simulation.

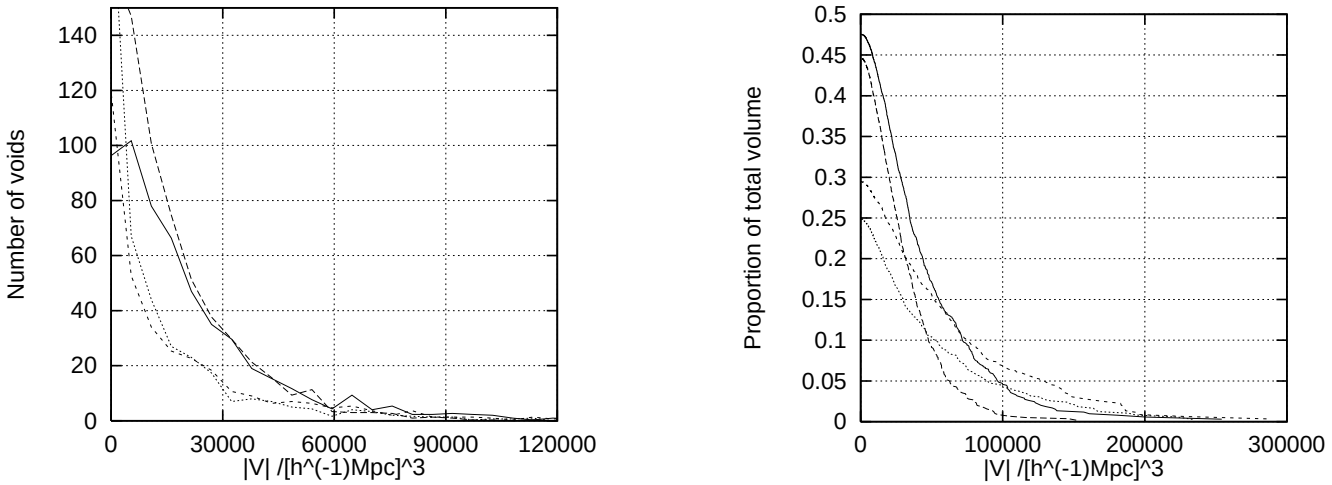

FIG. 4.—(*Left*) Distribution of the void volumes V for different simulations. The volumes are determined by counting the elementary cells belonging to the void. The solid line represents the real space with $\sigma_8 = 1.0$, and the long-dashed line represents the real space with $\sigma_8 = 0.69$. The redshift-space simulations are represented with the short-dashed and dotted lines for $\sigma_8 = 1.0$ and $\sigma_8 = 0.69$, respectively. (*Right*) Integrated proportion of the simulation box volume belonging to the voids larger than V. The line types are same as in the left panel.

together with the wall galaxies. Because of the three dimensionality, the voids are partly one on top of another.

We also tested the algorithm in the redshift space. The real-space galaxy coordinates were mapped into the redshift space using the linear Hubble law and known peculiar velocities. The number of voids decreased to 340 ± 10 for $\sigma_8 = 1.0$ and to 430 ± 15 for $\sigma_8 = 0.69$. The sizes of voids were $(23 \pm 33) \times 10^3 (h^{-1}$ Mpc$)^3$ for $\sigma_8 = 1.0$ and $(16 \pm 25) \times 10^3 (h^{-1}$ Mpc$)^3$ for $\sigma_8 = 0.69$. The galaxies and voids in the redshift space are shown in Figure 3.

In the case of real-space void finding, the program needs about 20 minutes to find voids from a distribution of 100,000 galaxies. In fact, the generation of the DF takes about 20 minutes and the rest of the algorithm needs only $\sim$15 s. All the timings were done in AlphaStation 500/266 running Digital Unix.

7. ANALYSIS

In this section we show some statistical properties of voids found by our algorithm. We also compare our results with the results of the empty sphere method (Einasto et al. 1989) by an analysis of the main radii of the voids. In a forthcoming paper a more detailed comparison with the method of El-Ad et al. (1996) and El-Ad & Piran (1997) and the use of real redshift catalogs is introduced.

In the left panel of Figure 4 we show the average distribution of the volumes of voids V over the three realizations of the different types of simulations. The right panel of Figure 4 shows how much of the simulation-box volume belongs to the voids whose volume is larger than V. As the volume of voids V reaches zero, we get the total volume of voids found by the algorithm. The rest of the volume belongs to the elementary cells, which are ignored during the search (i.e., cells that are not empty or are on the boundaries of the simulation box). This value is correlated to the resolution parameter s, but also to the distribution of galaxies as can be seen from the graph.

The left panel of Figure 5 shows the distribution of main radii R_V of the voids. The radii R_V—the maximum distances to the nearest galaxies—have been used in earlier works to

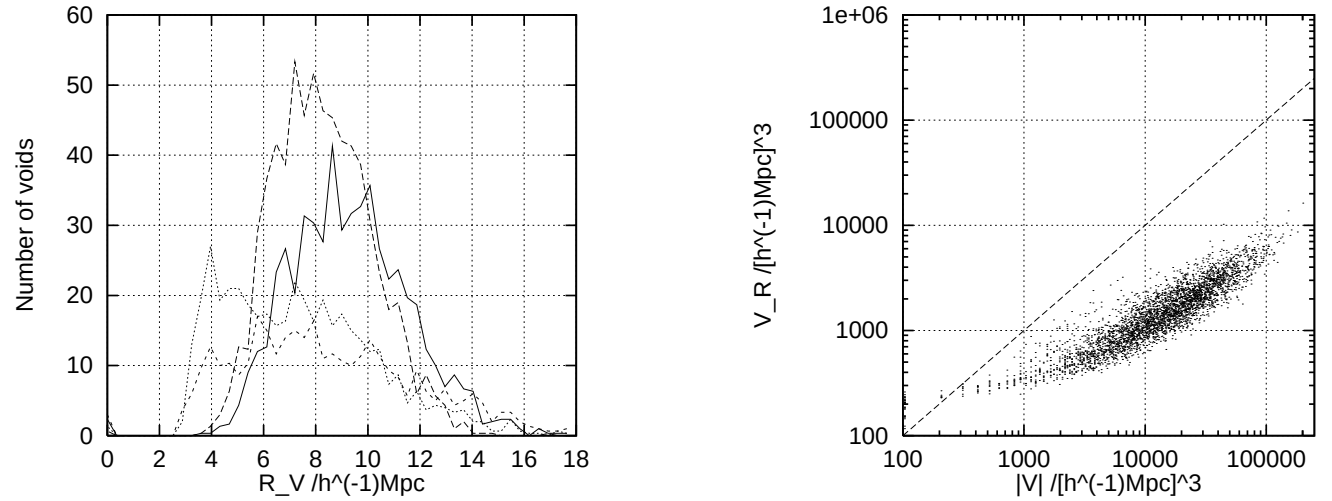

FIG. 5.—(*Left*) Distribution of the main radii R_V (the largest value of the DF inside the void) for different simulations. The line types are as in Fig. 4. (*Right*) Volumes of voids approximated from the size of the "central sphere" with radius equal to the main radius of the void: $V_R = (4/3)\pi R_V^3$ vs. the "real volumes" V. All voids from all of the real-space simulations are included in the graph. The behavior of the volumes was very similar for all types of simulations.

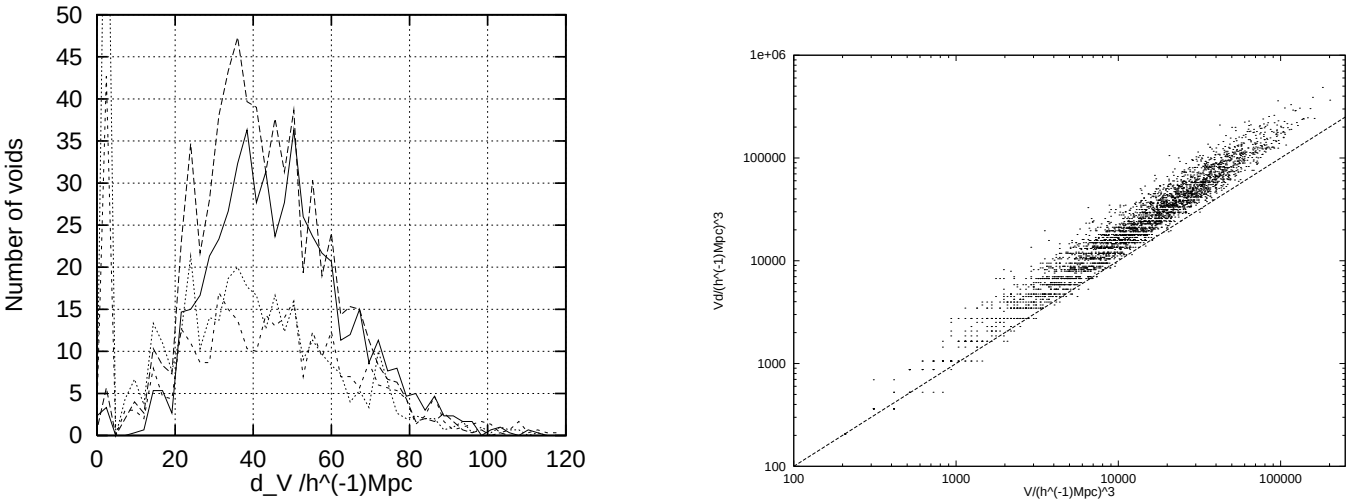

FIG. 6.—(*Left*) Distribution of the void diameters d_V (the largest distance between a pair of elementary cells in the void) for different simulations. The line types are as in Fig. 4. (*Right*) Volumes of voids approximated from the diameters of the voids $V_R = (4/3)\pi(d_V/2)^3$ vs. the "real volumes" V. All voids from all of the real-space simulations are included in the graph.

approximate the minimum radii of voids. However, if we adopt the void definition above, this approximation fails. The radius R_V is not the radius r_V of the largest sphere that can be put inside the void. In fact, $r_V \leq R_V$, since when centered in M_R, the R_V sphere almost always reaches outside the void. This behavior is especially significant with smaller voids because of the relative sparseness of the particle walls bounding them (cf. the circles drawn in the left panel of Fig. 1; also note the small triangular void in the middle of the same figure). This phenomenon is illustrated in the right panel of Figure 5. It shows the behavior of the volume of a void V_R approximated by the size of the central sphere of voids $[V_R = (4/3)\pi R_V^3]$ as a function of the "real volume" V of the void. The equality line $V_R = |V|$ is also drawn.

The right panel of Figure 5 can be used in the comparison of the present method with the empty sphere method of Einasto et al. (1988). This is because the sizes approximated by the main radii of voids are, in fact, the same as the sizes of voids found by the empty sphere method. The figure shows that the empty sphere method estimates the large voids to be much smaller than the present method. (That is because the empty sphere method only tries to approximate the minimum diameters of voids.) Also, the empty sphere method clearly overestimates the sizes of the smallest voids when compared with the results of the present method. Notice that there are dots above the equality line in the small end of the voids.

The left panel of Figure 6 shows the distribution of the diameters of the voids. In the right panel of Figure 6 an approximation of volumes similar to the one above is done, this time with the void diameters. The diameter volumes $V_d = (4/3)\pi(d/2)^3$ are compared with the detected volumes of voids. If the right panels of Figures 5 and 6 are compared, one can see that the diameter volumes correlate more linearly with the volumes of voids V than the main radius volumes V_R. In a forthcoming paper we shall give a more detailed comparison of our results with Frisch et al. (1995).

Finally, an example of the shape distribution determined with the inertia tensor method is shown in Figure 7. This figure presents the shapes of voids found from the $\sigma_8 = 1.0$ real-space simulations. The coordinates b/a and c/a are the axial ratios of the ellipsoids approximating the voids.

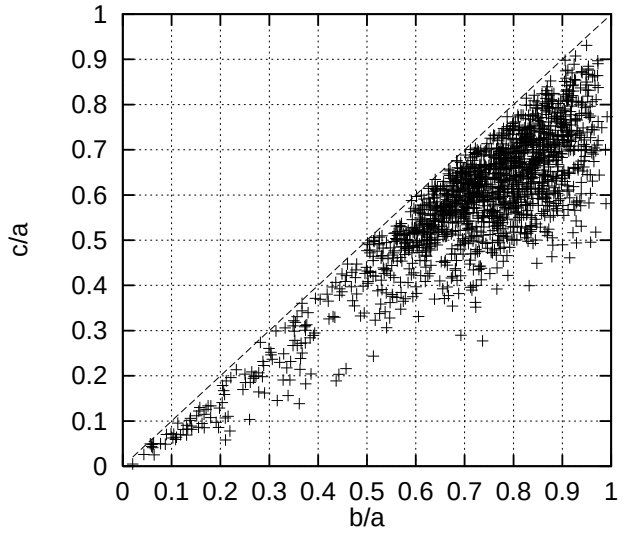

FIG. 7.—Shapes of voids of the real-space simulation with $\sigma_8 = 1.0$ in the axial ratio (b/a-c/a) coordinates. The shapes have been determined using the inertia tensor method described in § 5.

8. CONCLUSIONS

We have shown that our simple algorithm is a robust tool for automated void searching. This method does not predefine the shape of a void, and it has no other adjustable parameter than the resolution parameter of the mesh. The value of the resolution parameter is bounded only by the available memory of the computer used, and, in principle, the algorithm can be used in the continuum limit.

The definitions we have used are (1) the natural mathematical definition of a subvoid as that which surrounds a DF maximum and (2) the definition of a void as that which consists of a certain set of subvoids. The algorithm gives simple quantitative parameters to describe voids, e.g., main radius, outer diameter, and volume. The ellipsoid approximation can be used to approximate the shape. The algorithm is very versatile. Analyses can also be done with scalar fields other than the DF, and the algorithm can be boosted with the wall-finder algorithm. Further work is needed to compare the results with other methods, such as the algo-

rithm of El-Ad & Piran, and to use it for a real redshift catalog and different cosmological model simulations.

This work was partly supported by the Academy of Finland, and we gratefully acknowledge the computer resources provided by the Center of Scientific Computing and the University of Oulu. We thank the referee, who provided a number of suggestions that greatly improved the quality of the paper.

REFERENCES

Bond, J. R., & Efstathiou, G. B. 1984, ApJ, 285, L45
Broadhurst, T. J., Ellis, R. S., Koo, D. C., & Szalay, A. S. 1990, Nature, 343, 726
Couchman, H. M. P., Thomas, P. A., & Pearce, F. R. 1995, ApJ, 452, 797
da Costa, L. N., et al. 1994, ApJ, 424, L1
Einasto, J., Einasto, M., & Gramann, M. 1989, MNRAS, 238, 155
Einasto, J., & Gramann, M. 1993, ApJ, 407, 443
Einasto, J., Klypin, A. A., Saar, E., & Shandarin, S. F. 1984, MNRAS, 206, 529
Einasto, M., Einasto, J., Tago, E., Dalton, G. B., & Andernach, H. 1994, MNRAS, 269, 301
Einasto, M., Tago, E., Jaaniste, J., Einasto, J., & Andernach, H. 1997, A&AS, 123, 119
El-Ad, H., & Piran, T. 1997, ApJ, 491, 421
El-Ad, H., Piran, T., & da Costa, L. N. 1996, ApJ, 462, L13
Frisch, P., Einasto, J., Einasto, M., Freudling, W., Fricke, K. J., Gramann, M., Saar, V., & Toomet, O. 1995, A&A, 296, 611
Geller, M. J., & Huchra, J. P. 1989, Science, 246, 897
Hochen, J., & Kopelman, R. 1976, Phys. Rev. B, 14, 3438
Kaiser, N., & Peacock J. A. 1991, ApJ, 379, 482
Kauffmann, G., & Fairall, A. P. 1991, MNRAS, 248, 313
Linder, U., Einasto, J., Einasto, M., Freudling, W., Fricke, K., & Tago, E. 1995, A&A, 301, 329
Oort, J. H. 1983, ARA&A, 21, 373
Ryden, B. S. 1995, ApJ, 452, 25
Ryden, B. S., & Melott, A. L. 1996, ApJ, 470, 160
Turner, E. L., & Gott, J. R. III 1975, ApJ 197, L89
Zeldovich, Y. B., Einasto, J., & Shandarin, S. F. 1982, Nature, 300, 407