Multigrid contact detection method

Kejing He* and Shoubin Dong
Department of Computer Science, South China University of Technology, Guangzhou 510641, China

Zhaoyao Zhou
Guangdong Key Laboratory for Advanced Metallic Materials Processing, South China University of Technology, Guangzhou 510641, China

(Received 9 January 2007; published 28 March 2007)

Contact detection is a general problem of many physical simulations. This work presents a $O(N)$ multigrid method for general contact detection problems (MGCD). The multigrid idea is integrated with contact detection problems. Both the time complexity and memory consumption of the MGCD are $O(N)$. Unlike other methods, whose efficiencies are influenced strongly by the object size distribution, the performance of MGCD is insensitive to the object size distribution. We compare the MGCD with the no binary search (NBS) method and the multilevel boxing method in three dimensions for both time complexity and memory consumption. For objects with similar size, the MGCD is as good as the NBS method, both of which outperform the multilevel boxing method regarding memory consumption. For objects with diverse size, the MGCD outperform both the NBS method and the multilevel boxing method. We use the MGCD to solve the contact detection problem for a granular simulation system based on the discrete element method. From this granular simulation, we get the density property of monosize packing and binary packing with size ratio equal to 10. The packing density for monosize particles is 0.636. For binary packing with size ratio equal to 10, when the number of small particles is 300 times as the number of big particles, the maximal packing density 0.824 is achieved.

DOI: 10.1103/PhysRevE.75.036710 PACS number(s): 02.70.Ns, 05.10.−a, 45.70.−n, 81.05.Rm

I. INTRODUCTION

The contact detection methods aim at detecting potential physical contacts among all the objects in the system. Contact detection methods have numerous applications in various areas such as physically based simulations, granular materials, computer graphics, molecular dynamics (MD), and the discrete element method (DEM) [1]. Contact detection is usually the most computing-intensive process in the simulation of multiple discrete objects [2]. Categorized by how the object's spatial coordinates are related to its storage place in the memory (RAM), there are mainly two types of contact detection methods, namely, spatial sorting methods [3,4] and spatial hashing methods [5,6]. Spatial sorting methods make use of a hierarchical tree decomposition of the domain, or sort objects based on their spatial coordinates. In spatial hashing methods, the domain is usually partitioned into a set of orthogonal cells, and the objects are mapped to those cells based on spatial coordinates. After the relationship between the object's spatial location and its storage place being established, we can detect exact contacts by checking every object and their corresponding neighbors. If all pairwise objects are checked directly, the time complexity is $O(N^2)$. Spatial sorting methods have a time complexity of $O(N \ln N)$. Spatial hashing methods, on the other hand, have a time complexity of $O(N)$, but they are much more sensitive to the object size variances (e.g., the size ratio) [3,6].

For real-life physical problems, sizes of objects in the system are often widely diverse. Some researchers extended the basic spatial hashing method to solve the contact detection problem for diverse object sizes. The natural approach is to discretize every large object into several small objects, and detect contacts for these small objects. The multilevel boxing method proposed by Iwai *et al.* [7] organizes objects into multiple level of boxes, and for each object and its smaller potential neighbors, the contact is checked in the corresponding level according to the size of the potential neighbors. When the object sizes conform to bimodal distribution, let W_b be the size of big objects, W_s be the size of small objects, and D be the number of dimensions of the system. In their method, the contacts between big objects and small objects are detected in the fine level. Since each big object needs to be checked against $\left(\frac{W_b}{W_s}\right)^D$ small cells in the fine level, the time complexity is $O\left[N_b \times \left(\frac{W_b}{W_s}\right)^D\right]$. If the object sizes conform to a uniform distribution, $O\left[N_b \times \left(\frac{W_b}{W_s}\right)^D\right]$ becomes $O\left[N \times \left(\frac{W_b}{W_s}\right)^D\right]$ (see the Appendix). CGRID [5] solves this problem by making large objects alive until all of their neighboring cells expire. A review of various approaches to contact detection is given by Munjiza [8]. These algorithms face two contradictory sides. If discretizing the information of big objects to the fine grid, the time complexity for this discretization itself is $O\left[N_b \times \left(\frac{W_b}{W_s}\right)^D\right]$. Or, other objects cannot get the spatial information of those neighboring big objects.

This paper presents a general multigrid contact detection method (MGCD). By integrating the idea of multigrid and solving different ranges of contacts at grids with different coarsenesses, the MGCD has $O(N)$ time complexity and is insensitive to object size ratio $\frac{W_b}{W_s}$. This paper is organized as follows. Following the introduction in Sec. I, Sec. II presents the multigrid method for contact detection. Section III discusses the performance of MGCD and compares it with the

*Corresponding author. Electronic address: kejinghe@ieee.org

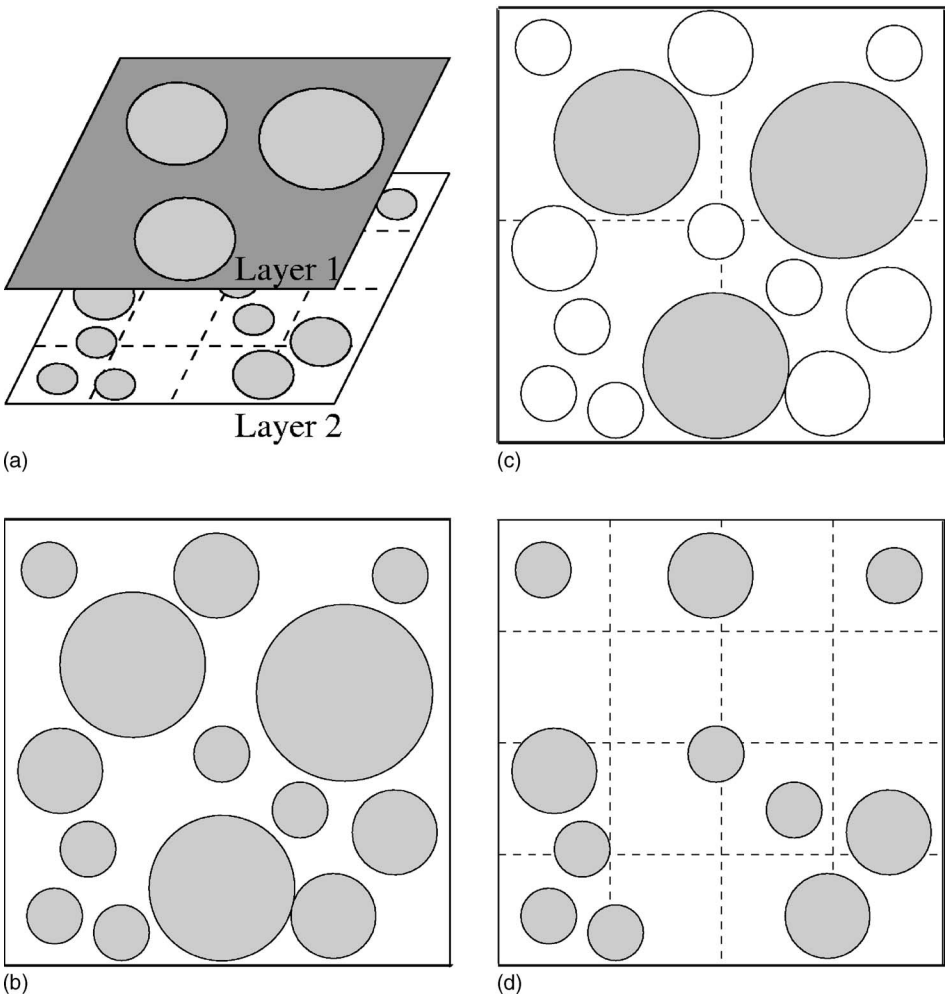

FIG. 1. Schematic view of a system with two grids in 2D. Objects are circles. (a) According to the objects' size, they belong to different grid layers. (b) Objects in the system. (c) Layer 1: S^1_{native} is shaded; $S^1_{\text{target}} - S^1_{\text{native}}$ is in white. (d) Layer 2: S^2_{native} is shaded.

no binary search (NBS) method [6] and the multilevel boxing method [7]. In Sec. IV, the MGCD is applied to the simulation of binary packing of granular materials with the size ratio equal to 10. Finally, some conclusions and discussions are made in Sec. V.

II. THE METHOD

In the MGCD, each object is completely contained in an axis-aligned bounding box (AABB), which is aligned with the axes of the coordinate system. AABB is used to improve the efficiency of contact detection, and when the shapes of objects meet some reasonable assumptions, we can expect that the number of intersections between AABBs remains proportional to the number of actual object intersections [9].

In standard spatial hashing methods, the contact detection process is composed of two phases. In the first phase, the domain is divided into monosize cells using rectangular grid. Each object is set to belong to a particular cell according to its coordinates, and objects in the same cell are organized as a linked list. In the second phase, for each cell, the method detects the contacts between the objects in current cell and all other objects in neighboring cells. For objects with the same size W, the cell size C is often chosen as $C=W+\Delta$, where $\Delta \geqslant 0$ is a small number. This strategy ensures all neighboring objects belong to neighboring cells. Since the search region for each particle is narrowed to neighboring cells, the time complexity is reduced to $O(N)$. However, if the system contains objects with diverse sizes, to ensure the neighboring objects belong to neighboring cells, the cell size needs to be $C=W_b+\Delta$ and each cell on average contains $N \times (C/L)^D$ objects, where L is the size of the domain. The density of big objects is $\rho_b = \frac{N_b \times W_b^D}{L^D}$, and the density of small

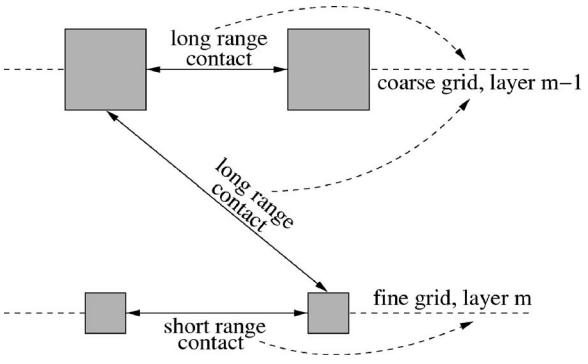

FIG. 2. The contacts among small objects are detected in the fine grid. The contacts among big objects are detected in the coarse grid. The contacts between big objects and small objects are also detected in the coarse grid.

objects is $\rho_s=\frac{N_s\times W_s^D}{L^D}$. Since each object needs to be checked against $O[N\times(C/L)^D]$ objects, the time complexity is

$$\begin{aligned} &O[N\times(N\times(C/L)^D)] \\ &\quad = O\{N\times[N_b\times(W_b/L)^D + N_s\times(W_b/L)^D]\} \\ &\quad = O\left[N\times\left(\rho_b + \rho_s\frac{W_b^D}{W_s^D}\right)\right]. \end{aligned} \tag{1}$$

The increase of time complexity is caused by the increase of search ranges. For detecting contacts between small objects, a search range which is slightly bigger than W_s is enough. However, for standard spatial hashing methods, a search range bigger than W_b is required.

To solve this problem, MGCD integrates the idea of multigrid and detect contacts with different ranges on grids with different coarsenesses (Fig. 2). Figure 1 illustrates a schematic view of a system with two grids.

Let the object set in the problem domain Ω be denoted by $O=\{O_i:i=1,2,\ldots,N\}$ and set $\{s_i:i=1,2,\ldots,N\}$ be the length of the longest dimension of the AABB of O_i. Let a sequence $\{G^m:m=1,2,\ldots,M\}$ of increasingly finer grids be given and $C^m(C^{m-1}>C^m)$ be the corresponding cell size of G^m. In each grid G^m, there are $\lfloor\frac{L}{C^m}\rfloor^D$ cells. Object O_i belongs to a particular cell if and only if the center of the AABB of O_i lies within this cell. For each grid G^m, there are two object sets: native object set S^m_{native} and target object set S^m_{target}, which are defined by

$$S^m_{\text{native}} = \{O_i: C^{m+1} \leqslant s_i < C^m\}, \tag{2}$$

$$S^m_{\text{target}} = \{O_i: s_i < C^m\}. \tag{3}$$

The following rules are satisfied:

$$\bigcup_m S^m_{\text{native}} = O, \tag{4}$$

$$S^m_{\text{native}} \cap S^n_{\text{native}} = \varnothing, \quad m \neq n, \tag{5}$$

$$S^m_{\text{target}} \subseteq S^{m-1}_{\text{target}}, \tag{6}$$

$$S^m_{\text{target}} \cup S^{m-1}_{\text{native}} = S^{m-1}_{\text{target}}. \tag{7}$$

The pseudocode of multigrid contact detection method is defined as follows.

```
1. for each grid G^m
2.   let {U^m_native} be the result of BUILD-CELLS (S^m_native)
3.   let {U^m_target} be the result of BUILD-CELLS (S^m_target)
4.   for each U_{i,j,k} ∈ {U^m_native}
5.     let {neighbor(U_{i,j,k})} be the set of cells neighboring to cell U_{i,j,k}
6.     for each U_{i',j',k'} ∈ {U^m_target} ∩ {neighbor(U_{i,j,k})}
7.       for each O_i ∈ U_{i,j,k}
8.         for each O_j ∈ U_{i',j',k'}
9.           DETECT (O_i, O_j)
```

BUILD-CELLS (S^m) partitions elements of set S^m into a set of no-empty cells $\{U^m\}$. DETECT (O_i,O_j) does the actual contact detection between object O_i and object O_j. The algorithm presented here focuses more on the multigrid aspect of the MGCD, and the implementation chosen is the classic spatial hashing. MGCD uses NBS, a clever, optimized implementation of the classic spatial hashing method (Secs. III and IV).

III. RESULTS AND DISCUSSION

Performance. In standard spatial hashing methods, the computing time is composed of two parts: the time $T_{\text{retrieval}}$ used for retrieving information from cells, and the time T_{detect} used for detecting contacts between the objects in $U_{i,j,k}$ and other objects in neighboring cells $\{\text{neighbor}(U_{i,j,k})\}$. In MGCD, for each grid G^m, the optimized NBS implementation is adopted. Rather than checking with all the cells, we only check those nonempty cells. For a sparse system, this strategy can achieve significant performance improvement and save a lot of memory [6]. There are at most $|S^m_{\text{native}}|$ nonempty cells in grid G^m, and for each native object $O_i \in S^m_{\text{native}}$, it will be checked against $|S^m_{\text{target}}|\times C_m^D/L^D$ averagely, so the computing time on grid G^m can be denoted by

$$\begin{aligned} T^m = T^m_{\text{retrieval}} + T^m_{\text{detect}} &= O(|S^m_{\text{native}}|) + O(|S^m_{\text{native}}|\times|S^m_{\text{target}}| \\ &\quad \times C_m^3/L^3) = O(|S^m_{\text{target}}|\times\rho^m_{\text{native}}), \end{aligned} \tag{8}$$

where ρ^m_{native} is the density of native objects of grid G^m. The computing time in whole is

$$\begin{aligned} T = \sum_m T^m &= \sum_m O(|S^m_{\text{target}}|\times\rho^m_{\text{native}}) < \sum_m O(N\times\rho^m_{\text{native}}) \\ &= O(N)\times\sum_m \rho^m_{\text{native}} = O(N). \end{aligned} \tag{9}$$

The time complexity T is independent of the number of grids and the number of cells.

The intrinsic multigrid property of MGCD leads to its efficiency. Contacts with different ranges are calculated on grids with different coarsenesses, so we don't have to maintain the long range contact information in the fine scale. Since different grids are independent of each other, we do not need to allocate memory for all grids at the same time. At any time, only one grid needs to be maintained in the memory. The memory consumption of MGCD is independent of the number of grid layers.

The tree-based contact detection methods have a $O(N\ln N)$ time complexity. It is meaningless to compare the $O(N)$ MGCD method with $O(N\ln N)$ methods. Here we compare MGCD with two $O\left[N\times\left(\frac{W_b}{W_s}\right)^D\right]$ methods, NBS [6], and the multilevel boxing method [7]. When the object sizes conform to bimodal distribution, following Sec. I, let W_b and W_s be the sizes of big objects and small objects, respectively. To solve this problem, we need two grids, G^b and G^s, the cell sizes of which satisfy $C_b=W_b$ and $C_s=W_s$. From Eqs. (2) and (3), $|S^b_{\text{target}}|=N$ and $|S^s_{\text{target}}|=N_s$. The density of big objects is $\rho_b=N_b\frac{W_b^D}{L^D}$, and the density of small objects is $\rho_s=N_s\frac{W_s^D}{L^D}$. From Eq. (8), solving the contacts between big objects in the coarse grid costs $N_b\frac{N_b\times W_b^D}{L^D}$, solving the contacts between big objects and small objects in the coarse grid costs $N_b\frac{N_s\times W_b^D}{L^D}$, and solving the contacts between small objects in the fine

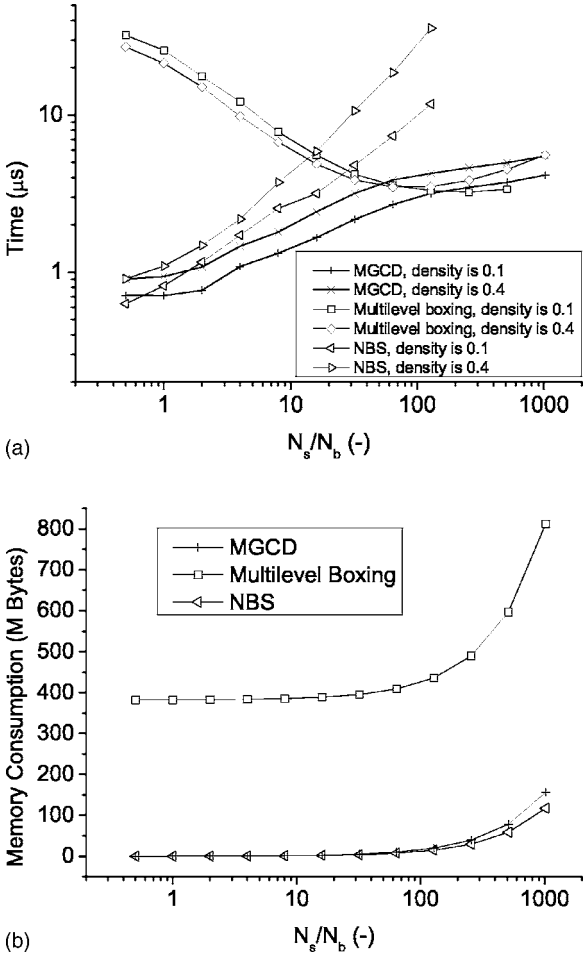

FIG. 3. Performance comparison between MGCD (+,×), the multilevel boxing method (□, ◇), and NBS method (◁, ▷) for different object composition. The computing time is normalized by the number of objects. $\frac{W_b}{W_s}$ is 10. The number of big objects is 10^4. Density is the sum of ρ_b and ρ_s ($\rho_b+\rho_s$). The host machine has one Pentium 42.8 G processor and 1 GB RAM. Due to excessive memory consuming, when the density is 0.1 and number of small objects is 10^7, the data for the multilevel boxing method are not available.

grid costs $N_s\frac{N_s\times W_s^D}{L^D}$. The total computing time is the sum of these three parts and to give a final time:

$$T = N \times \rho_b + N_s \times \rho_s. \tag{10}$$

Figure 3 illustrates the performance comparison for different object composition with $\frac{W_b}{W_s}$ equal to 10. In this case, MGCD includes two grids. The classic linked-list implementation [8] is adopted for coarse grid, and the NBS implementation is adopted for fine grid. MGCD is much faster than the multilevel boxing method when N_s has the same scale with N_b. Both MGCD and NBS consume much less memory than the multilevel boxing method. MGCD and NBS have similar performance when ρ_s is approximate to zero, but when ρ_s is not neglectable, MGCD outperforms NBS significantly. The computing time and memory consumption of MGCD is proportional to the number of objects rather than the number of

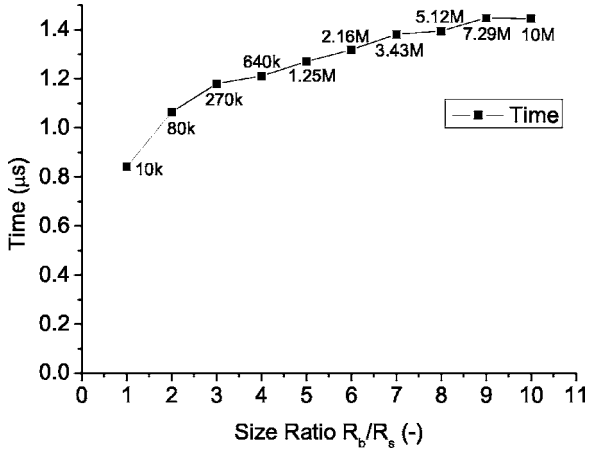

FIG. 4. The relationship between different size ratios and computing time. The computing time is normalized by the number of objects. N_b is 10^4, W_b is 10 cm, and the density (ρ) is 0.1. W_s varies from 1 to 10 cm, N_s (labeled in the figure) varies from 10^4 to 10^7, but $N_s\times W_s^3$ is kept fixed. The host machine is the same as the one for Fig. 3.

cells. For a sparse system, an algorithm being related to number of objects can save a lot of memory than being related to number of cells [Fig. 3(b)]. The relationship between the normalized computing time and different size ratios is depicted in Fig. 4. The performance of MGCD is insensitive to the size ratio and object size distribution.

MGCD can be parallelized effectively. Different grids can be calculated on different computers independently. The parallelization of contact detection in each grid can be achieved by domain decomposition method. For each grid, besides the NBS implementation, we can also only index those nonempty cells by building a hash mapping between cell coordinates and cell physical storage place. Generally, NBS is more efficient while the hash mapping based implementation is more straightforward.

IV. APPLICATIONS

The packing of particles has been studied for several years on account of its interesting properties, its technological importance and its potential applications. The research on the packing of particles can be applied to many fields, such as the microstructures of liquids [10], granular materials [11], amorphous materials [12,13], and powder metallurgy (PM) [14]. According to the packing method, the packing of particles can be classified into two classes: ordered packing and random packing. For the ordered packing, it is often easier to achieve larger packing density, while, random packing is more widely applied in nature and industry.

Random packing has been studied experimentally [15,16] and theoretically [17,18]. Based on DEM, scientists have studied the packing of monosize particles [19,20]. In the simulation of granular matter, the most time-consuming process is contact detection. We have integrated MGCD with DEM, and studied the influence of particle composition on packing density in three dimensions. Here, two packing

TABLE I. Parameters for the simulation of particle packing.

Parameter	Value	Parameter	Value
k_n	10^7 N/m	k_s	10^7 N/m
μ	0	γ_n	0.7
ρ	7800 kg/m^3	$\mathbf{g}$	9.8 m/s^2

schemes are compared: monosize packing and binary packing with size ratio equal to 10.

A. The model

According to Newton's second law of motion, every particle in the system undergoes two types of motion, namely, translational motion and rotational motion. In our model, the forces considered are the contact force in normal direction $\mathbf{F}_{cn,ij}$, the damping force in normal direction $\mathbf{F}_{dn,ij}$, the contact force in tangential direction $\mathbf{F}_{ct,ij}$ and gravity. All these forces cause translational motion. Since $\mathbf{F}_{ct,ij}$ is the only force that does not pass directly through the center of mass, the only cause of rotational motion is $\mathbf{F}_{ct,ij}$. The motion of particles can be described by

$$m_i \frac{d\mathbf{v}_i}{dt} = \sum_j (\mathbf{F}_{cn,ij} + \mathbf{F}_{dn,ij} + \mathbf{F}_{ct,ij}) + m\mathbf{g}, \tag{11}$$

$$I_i \frac{d\omega_i}{dt} = \sum_j (\mathbf{R}_i \times \mathbf{F}_{ct,ij}), \tag{12}$$

where m_i, $\mathbf{v}_i$, I_i, and ω_i are the mass, translational velocity, moment of inertia, and angular velocity of particle i, respectively. $\mathbf{R}_i$ is a vector running from the center of particle to the contact point with its magnitude equal to particle radius R_i. A simple linear elastic-damping model is adopted:

$$\mathbf{F}_{cn,ij} = k_n \xi_n \hat{\mathbf{n}}_{ij}, \tag{13}$$

where ξ_n is the displacement in normal direction and $\hat{\mathbf{n}}_{ij}$ is the unit vector point from the center of particle j to the center of particle i.

The damping in normal direction $\mathbf{F}_{dn,ij}$ is

$$\mathbf{F}_{dn,ij} = -\gamma_n |\mathbf{F}_{cn,ij}| \mathrm{sgn}(\dot{\xi}_n), \tag{14}$$

where γ_n is the damping constant in normal direction. The hysteretic damping model applied here is similar to the model proposed by [21].

Elastic model in tangential direction is adopted from [1], in which a tangential "virtual" spring is put at the contact point when two particles start to touch each other:

$$\mathbf{F}_{cs,ij} = -\min[|k_s \xi_s|, |\mu(\mathbf{F}_{cn,ij} + \mathbf{F}_{dn,ij})|] \times \mathrm{sgn}(\xi_s), \tag{15}$$

where k_s is the tangential stiffness, and ξ_s is the total shear displacement of the tangential spring that took place since the time t_0 when the contact was first established, i.e.,

$$\xi_s = \int_{t_0}^{t_1} \mathbf{v}_s(t) dt. \tag{16}$$

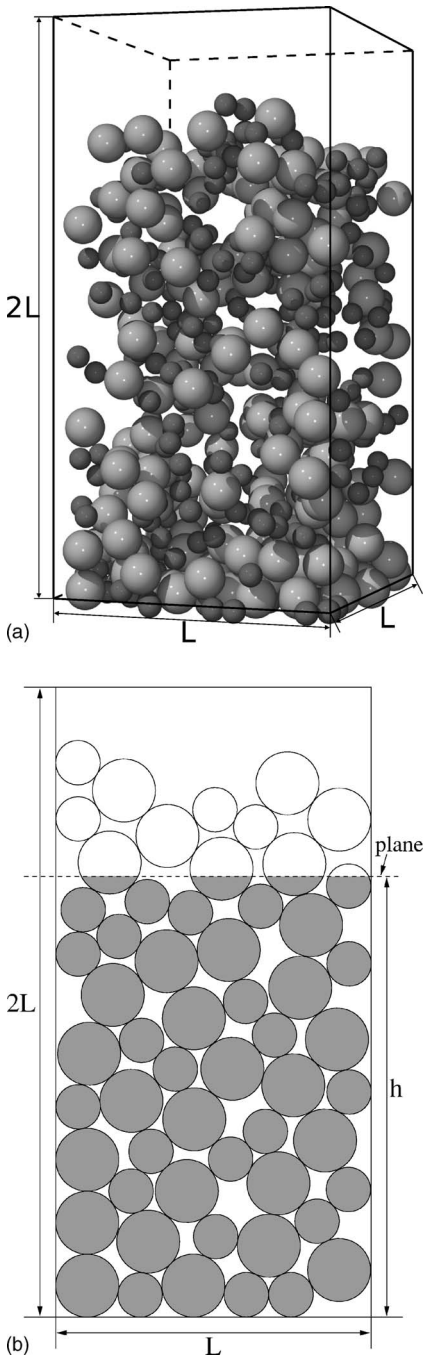

FIG. 5. (a) Snapshot showing the formation of a binary packing of 200 big particles (R_b=6.5 mm) and 300 small particles (R_s =3.9 mm). (b) Schematic view of calculation cuboid in 2D. Here, (particles $\cap$ calculation cuboid) is the shaded area, and the area of calculation cuboid is $L \times h$ [Eq. (17)].

B. Simulation condition

The simulation is composed of two stages, namely, the initial stage and the settling stage. The initial position of particles requires special handling. If they were located randomly, overlaps between some of them would be inevitable.

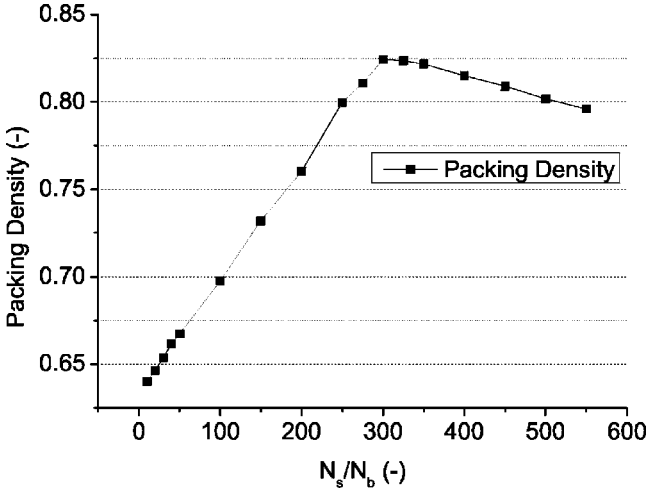

FIG. 6. The relationship between particle composition and packing density. The size ratio $\frac{R_b}{R_s}$ is 10. The number of big particles N_b is fixed at 2000.

That would induce instability right after the packing simulation starts. To avoid such initial overlaps, a "growing process" was introduced in the initial stage. At the beginning of the initial stage, particles are initialized with radius equal to zero and target radius conforming to a prescribed distribution. These particles are positioned within a rectangular box randomly and the acceleration due to gravity **g** is set to be zero. Then the particles begin to grow and touch with each other. The contacts are detected and processed. The growing process ends when particle radii reach their target radii. Then the simulation switches to the settling stage. At the settling stage, gravity is switched on, particles move according to gravity and interparticle forces. They will collide with neighboring particles and bounce back and forth. The system energy will dissipate by damping and friction and particles will reach their stable position eventually. In this application, the simulation time step Δt is 10 μs, the initial stage lasts 2×10^4 steps (0.2 s), and the whole simulation lasts 3×10^5 steps (3 s). Other simulation parameters are listed in Table I.

The radius of big particles (R_b) and small particles (R_s) are 5 cm and 5 mm, respectively. The number of big particles (N_b) is 2000, and the number of small particles (N_s) varies from 2×10^4 to 10^6. Particles are put in a rectangular box, and the volume of box is twice as the total volume of all particles. The vertical dimension of the box is twice as long as the two horizontal dimensions (Fig. 5). Periodic boundary conditions are applied to four vertical walls. No top boundary condition is applied and the bottom wall has the same physical property as particles. When the simulation is finished, since the top profile of these particles may not be flat, a virtual plane is placed 30 cm below the top particle. This virtual plane, the bottom wall, along with four vertical walls compose a calculation cuboid [Fig. 5(b)]. Packing density is defined by

$$\text{packing density} = \frac{\text{volume of particles} \cap \text{calculation cuboid}}{\text{volume of calculation cuboid}}. \tag{17}$$

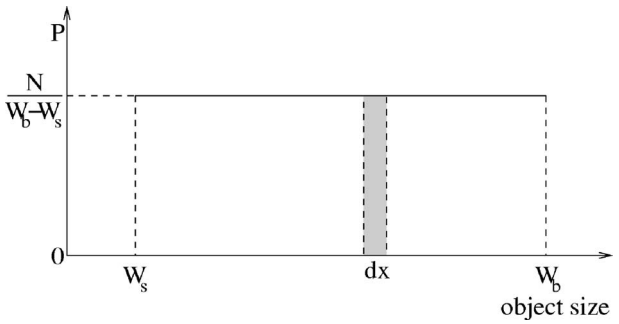

FIG. 7. Uniform distribution.

Figure 6 illustrates the relationship between different composition and packing density. By using 10^4 monosize particles, we get the packing density of 0.636. It agrees well with the classic experimental value of 0.637 obtained by Scott and Kilgour [15], which pours ball bearings into a large container, vibrates the system to achieve maximum densification, and extrapolates the results to eliminate finite-size effects. And when $\frac{N_s}{N_b}$ is 300, the maximal packing density 0.824 is achieved. In this application, we can make polydisperse packing get larger density than monosize packing with appropriate particle composition.

V. CONCLUSIONS

This paper has presented a multigrid method for contact detection. It has $O(N)$ time complexity and memory consumption. The performance of MGCD is insensitive to object size distribution. We have compared MGCD with other contact detection methods. MGCD has the same performance as NBS method when the system contains merely monosize objects, but MGCD outperforms other methods when the object sizes are diverse. MGCD can be applied to general contact detection problems as long as the objects in the physical system can be contained by bounding boxes. We have applied MGCD to a granular simulation problem, and simulated the dynamics of 10^6 metal particles. The packing density for monosize particles is 0.636. We have also obtained the binary packing density function for a big object size ratio. If the size ratio is fixed at 10, when the number of small particles is 300 times as large as the number of big particles, the maximal packing density 0.824 is achieved.

ACKNOWLEDGMENTS

We are grateful to Puqing Chen of the Singapore-MIT Alliance (SMA) for valuable discussions and suggestions.

This research was supported by China Next Generation Internet Project (CNGI-04-15-7A).

APPENDIX: THE COMPUTING TIME FOR ORDINARY METHODS WHEN PARTICLE SIZES CONFORM TO UNIFORM DISTRIBUTION

When object sizes conform to a uniform distribution (Fig. 7), the number of particles in the range $[x,x+dx]$ is $\frac{N}{W_b-W_s}dx$. According to Sec. I, the time complexity for discretizing the information of these objects to the fine grid is $\left(\frac{N}{W_b-W_s}dx\right)\times\left(\frac{x}{W_s}\right)^D$, so the computing time in whole is

$$\begin{aligned}\text{computation time} &= \int_{W_s}^{W_b}\left(\frac{N}{W_b-W_s}dx\right)\times\left(\frac{x}{W_s}\right)^D \\ &= \frac{N}{W_b-W_s}\frac{1}{W_s^D}\left.\frac{x^{D+1}}{D+1}\right|_{W_s}^{W_b} \\ &= \frac{N}{W_b-W_s}\frac{1}{W_s^D}\frac{W_b^{D+1}-W_s^{D+1}}{D+1} \\ &\overset{W_b\gg W_s}{=} O\left[N\left(\frac{W_b}{W_s}\right)^D\right]. \end{aligned} \tag{A1}$$

[1] P. Cundall and O. Strack, Geotechnique **29**, 47 (1979).
[2] J. Williams and R. O'Connor, Arch. Comput. Methods Eng. **6**, 279 (1999).
[3] E. Perkins and J. R. Williams, Eng. Comput. **18**, 48 (2001).
[4] J. Bonet and J. Peraire, Int. J. Numer. Methods Eng. **31**, 1 (1991).
[5] J. Williams, E. Perkins, and B. Cook, Eng. Comput. **21**, 235 (2004).
[6] A. Munjiza and K. Andrews, Int. J. Numer. Methods Eng. **43**, 131 (1998).
[7] T. Iwai, C. Hong, and P. Greil, Int. J. Mod. Phys. C **10**, 823 (1999).
[8] A. Munjiza, *The Combined Finite-Discrete Element Method* (Wiley, New York, 2004), Chap. Contact Detection, pp. 73–130.
[9] S. Suri, P. Hubbard, and J. Hughes, ACM Trans. Graphics **18**, 257 (1999).
[10] J. Bernal, Nature (London) **185**, 68 (1960).
[11] C. H. Rycroft, M. Z. Bazant, G. S. Grest, and J. W. Landry, Phys. Rev. E **73**, 051306 (2006).
[12] J. Finney, Nature (London) **266**, 309 (1977).
[13] A. Clarke and J. Wiley, Phys. Rev. B **35**, 7350 (1987).
[14] R. M. German, *Particle Packing Characteristics* (Metal Powder Industries Federation, Princeton, NJ, 1989).
[15] G. Scott and D. Kilgour, Br. J. Appl. Phys., J. Phys. D **2**, 863 (1969).
[16] J. Finney, Proc. R. Soc. London, Ser. A **319**, 479 (1970).
[17] J. G. Berryman, Phys. Rev. A **27**, 1053 (1983).
[18] H. J. H. Brouwers, Phys. Rev. E **74**, 031309 (2006).
[19] R. Y. Yang, R. P. Zou, and A. B. Yu, Phys. Rev. E **62**, 3900 (2000).
[20] L. Liu, Z. Zhang, and A. Yu, Physica A **268**, 433 (1999).
[21] O. R. Walton and R. L. Braun, J. Rheol. **30**, 949 (1986).