

Optimization Techniques in an Event-Driven Simulation of a Shaker Ball Mill

Marina Gavrilova¹, Jon Rokne¹, Dmitri Gavrilov², and Oleg Vinogradov³

¹ Dept of Comp. Science, Universit of Calgary, Calgary, AB, Canada, T2N1N4
marina@cpsc.ucalgary.ca, rokne@cpsc.ucalgary.ca

² Microsoft Corporation, Redmond, WA, USA dmitrig@microsoft.com

³ Dept of Mech. and Manufacturing Eng., University of Calgary, Calgary, AB,
Canada, T2N1N4 ovinogra@ucalgary.ca.

Abstract. The paper addresses issue of efficiency of an event-driven simulation of a granular materials system. Performance of a number of techniques for collision detection optimization is analyzed in the framework of a shaker ball mill model. Dynamic computational geometry data structures are employed for this purpose. The results of the study provide insights on how the parameters of the system, such as the number of particles, the distribution of their radii and the density of packing, influence simulation efficiency.

1 Introduction

Principles of mechanical alloying were first established in 1960's by J.S. Benjamin [3]. Since then a number of authors contributed to the theory of efficient ball mill design [12, 2]. Both a continuous and a discrete simulation models were considered in those studies [2, 4].

In any simulation model, one of the important aspects is scheduling collisions between particles. The collision detection optimization (CDO) in multi-particle systems has been proven to be a crucial task that can consume up to 80% of the simulation time and thus significantly influence the efficiency of the simulation [4]. Up until recently, the cell method was the most popular method for collision detection optimization in molecular dynamics, computational physics and granular materials simulation [10, 11, 13]. In this method, the simulated space is divided into rectangular subdomains and the collision detection performed only on the neighboring cells. The known downside of this method is its dependency on the distribution of particle sizes that can deem the approach completely inefficient. A variety of other methods based on hierarchical planar subdivisions were recently developed. Successful applications are found in motion planning, robotics and animation [7, 1, 9].

The paper presents the results of an experimental comparison of a number of collision detection algorithms as applied to the problem of efficient simulation of a mechanically alloyed system. The event-driven approach was employed to develop the shaker ball mill simulation model. The performance of algorithms

was studied for data sets containing mono-sized and poly-sized particles, with various particle distribution and configurations. To the best of our knowledge, this is the first systematic study focused on understanding the correlation between specific features of the developed algorithms and the parameters of the simulated system.

2 The shaker ball mill model

A *shaker ball mill* comprises a massive cylinder filled with steel balls oscillating in a vertical direction with a specified amplitude and frequency. Balls are moving along parabolic trajectories. Collisions between the balls and between the balls and the cylinder are assumed to be central and frictionless (the tangential velocity component of colliding particles is conserved). The cylinder is assumed to have an infinite mass, so that its velocity is not affected by the collisions.

The inelastic nature of collisions can be represented by introducing restitution coefficients, which represent the ratio between particle velocity components before and after collision. The restitution coefficient ε is calculated as a function of particle velocity and size by using the Hertz contact theory to find the deformations of particles due to a collision [4]. It is assumed that the energy lost due to impact was spent on heating this volume. It is also assumed that the temperature returns to normal after each impact.

Dynamics of a mechanical alloyed system is described by a system of ordinary differential equations of time [14]. The solution of the system describes, in particular, the trajectories of bodies in the time interval $(t, t + \Delta t]$ and the velocities of the bodies in the next time step. In a shaker ball mill, balls move in a gravitational force field. The general equation of motion of a body in a force field is given according to the 2nd Newton's law. For the gravitational field, this equation can be solved analytically. The solution is

$$\mathbf{x}(t + \Delta t) = \mathbf{x}(t) + \mathbf{v}(t)\Delta t + \mathbf{g}\frac{\Delta t^2}{2}, \mathbf{v}(t + \Delta t) = \mathbf{v}(t) + \mathbf{g}\Delta t, \quad (1)$$

where $\mathbf{v}(t)$ is the velocity vector of the particle, $\mathbf{x}(t)$ is the position vector of the body, and \mathbf{g} is the gravitational constant. The time of collision between two particles is found as the minimal positive root t_0 of the equation

$$\|\mathbf{x}_1(t) - \mathbf{x}_2(t)\| = r_1 + r_2, \quad (2)$$

where $\mathbf{x}_i(t)$, $i = 1, 2$ are the positions of the centers of two particles, and \mathbf{r}_i , $i = 1, 2$ are the radii of the particles. For parabolic motion in the gravitational field, the root is found exactly by solving a second degree algebraic equation. If the root does not exist, the particles will not collide.

The normal velocities after collision has taken place are found as:

$$\begin{pmatrix} v_1^{after} \\ v_2^{after} \end{pmatrix} = \begin{bmatrix} A - \varepsilon & 1 - A + \varepsilon \\ A & 1 - A \end{bmatrix} \begin{pmatrix} v_1^{before} \\ v_2^{before} \end{pmatrix}, \quad A = \frac{(1 + \varepsilon)m_1}{m_1 + m_2}, \quad (3)$$

where m_1, m_2 are the masses of particles, and ε is the coefficient of restitution [4]. When the energy of the system is high, and there are no long-term interactions between particles, a common approach is to employ the event-driven simulation scheme, as described below.

3 Event-driven simulation scheme

As of today, most of the research on collision detection of particle systems is limited to consideration of a relatively simple simulation model. The idea is to discretize time into short intervals of fixed duration. At the end of each time interval, the new positions of the moving particles are computed. The common problem with such methods is related to choosing the length of the interval.

A much more precise and effective approach, the *dynamic event-driven simulation* of a particle system, relies on discrete events that can happen at any moment of time rather than on the fixed time steps [14]. This can be accommodated by introducing an event queue. We employ this scheme with the following events: collision events, predict trajectory events and topological events. The collision optimization problem is considered in the framework of the dynamic event-driven simulation.

In mechanically alloyed materials simulation, particles are usually approximated by balls (disks). A *collision event* occurs when two balls come into contact with each other or with a boundary. A *predict trajectory event* occurs when the trajectory and the velocity of a ball is updated due to re-calculation of the system state at the next time step. A particle travels along a trajectory defined by a function of time between two consecutive events. In most cases, the trajectories are piecewise linear.

The task of detecting collisions can be optimized by maintaining a set of neighboring particle for every particle in the simulated system. A computational overhead, related to collision detection algorithm, is related to *topological events*. New neighboring pairs appearing due to a topological event must be checked for collisions.

4 Dynamic Computation Geometry Data Structures

Consider a problem of optimizing the collision detection for a set of moving particles in the context given above. In a straightforward approach each pair of particles is considered to be neighbors, i.e. the neighbor graph contains $\frac{1}{2}n(n-1)$ edges. For a large scale computation the method performs poorly. The number of neighbors considered on each step can be reduced by employing a geometric data structure and by dynamically maintaining the list of neighbors.

The dynamic generalized Delaunay triangulation, the regular spatial subdivision, the regular spatial tree and the set of segment trees based methods were chosen as candidates for analysis. Dynamic generalized Delaunay triangulation

in power metric was built using the sweep-plane technique. *INCIRCLE* tests allowed to dynamically maintain the the set of neighbors [6]. A topological event occurs when the proximity relationship in Delaunay triangulation changes. An important characteristic of this method is its versatility - its performance is independent of the distribution of particle sizes and their configuration.

In regular spatial subdivision, the space is divided onto cells. The topological event happens when a particle moves from one cell to another. The size of cells in the regular spatial subdivision is selected so that it guarantees that no more than a constant number of particles resides in each cell at any moment of time. This allows imposing limit on the number of neighbors for each particle, which ensures a good performance for monosized particle systems. A topological event happens when a particle moves from one cell to another. Introducing hierarchy for regular spatial subdivision reduces storage requirements. An AVL tree is used for this purpose. The method is called the regular spatial tree method.

The set of segment trees is the final data structure considered. A tree of intersecting segments, obtained as a projection of the bounding boxes of particles onto one of the coordinate axes, is dynamically maintained. The particles are said to be neighbors if their bounding boxes intersect. A topological event takes place when two segment endpoints on one of the axes meet. If the bounding boxes of the corresponding particles intersect, then the particles become neighbors. A detailed description and performance analysis of the above data structures can be found in [5].

5 Experimentations

The event-driven simulation environment for shaker ball mill was created at the Dept. of Mechanical and Manufacturing Engineering, University of Calgary. Algorithms were implemented in Object-Oriented Pascal in the Borland Delphi environment and run under Windows 2000 operating system. The size of the simulation space (shaker area) was set to 200 by 200 mm. Balls radii were in the range from 1 to 10 mm. The duration of the simulation run was set to 10 sec. Predict trajectory events were computed every 0.005 sec, which defines the duration of the timestep.

5.1 Monosized Data Sets

The first series of experiments considered data sets comprising 100 monosized balls (i.e. particles with the same radius). The density of the distribution is defined as the ratio between the combined volume of balls in the simulation space and the volume of the simulation space. The density was gradually increasing from 5% to 70% (see Table 1). The number of collisions and the number of predict trajectory events are independent of the collision optimization method. The number of collisions grows as a linear function on a packing density until it reaches 50

Table 1. Number of collision and predict trajectory events for monosized data sets

Packing density	5%	20%	33%	50%	70%
Ball radius (mm)	2.5	5	6.5	8	9.5
Collision events	174	387	661	1143	4170
Predict trajectory	199855	199459	198941	198009	192371

In Table 2, the elapsed time, the total number of topological events (TE) and the total number of collision checks (CC) performed during simulation run are recorded. From Table 2, observe that the number of topological events is the

Table 2. Experimental results for monosized data sets.

Density	5%	5%	5%	33%	33%	33%	70%	70%	70%
Algorithm	Time	TE	CC	Time	TE	CC	Time	TE	CC
Direct	1114.61	0	20645172	1218.85	0	20783008	1255.10	0	21899782
Regular sub.	96.50	1718	93137	117.71	582	625788	166.15	759	1443880
Spatial tree	103.04	1718	93137	128.36	582	625788	171.81	759	1443880
Segment trees	106.94	13038	11542	121.72	13362	116468	151.87	30094	337770
Dynamic DT	304.84	778	1926240	312.64	601	1925194	333.89	770	1979036

largest for the segment tree method. This can be justified by the fact that the topological event in a segment tree happens every time the projections of two particles collide on a coordinate axis. Thus, the number of topological events increases significantly for a densely packed system. The number of topological events is the smallest for the dynamic Delaunay triangulation method.

The number of collision checks for the straightforward method is approximately 10 times larger than the number of collision checks in dynamic Delaunay triangulation, 20 times larger than that of the regular subdivision and spatial tree methods and 100 times larger than that of the segment tree method. The smallest number of the collision checks for the segment tree method is justified by the fact that collision checks are only performed when bounding boxes of two balls intersect, which happens rarely. The number of the collision checks is the largest for the dynamic DT method, due to the number of neighbors. Note, that the DT method is the only method where this number does not depend on the density of the distribution.

The dependence of the time required to simulate the system of 100 particles on the density of packing is illustrated in Fig. 1. Note that the actual time required to process the simulation on a computer is significantly larger than the duration of the "theoretical" simulation run. Immediate conclusion that can be drawn is that the use of any collision optimization technique improves the performance of the simulation algorithm at least by an order of magnitude.

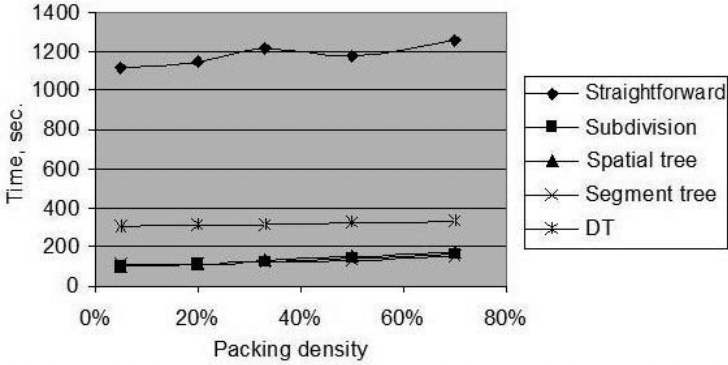


Fig. 1. Time vs. packing density for monosized data set.

The dynamic Delaunay triangulation method performs approximately 3 times slower than all the other collision optimization methods for distributions of low density. It is, however, almost twice as slow for high density distributions. Note that the difference in performance is partly due to the fact that every topological event for this method requires the solution of a 4th order polynomial equation. Also note that the number of collision checks is practically constant in the dynamic Delaunay triangulation, thus the performance remains unchanged as the density increases.

The spatial tree method is only 5% slower than the regular subdivision method, while it requires significantly less memory. The segment tree method is the most efficient among all methods considered for the monosized particle system.

5.2 Polysized Data Sets

The second series of experiments were conducted for polysized data sets of 100 balls. The radius of largest ball was approximately 10 times larger than the radius of the smallest ball. The packing density changed from 5% to 70%. The number of collision events and predict trajectory events were measured (see Table 3).

The comparison analysis on the number of topological events, collision checks and the elapsed time depending on the type of the CDO method and the density of the particle distribution is given in Table 4. As the packing density increases, the number of collision checks raises significantly for the regular subdivision and spatial tree methods. It is evident that these methods are inappropriate for CDO for a system of polysized particles. Two of the best performing methods, the segment tree and the dynamic Delaunay triangulation, are now compared.

Table 3. Number of collision and predict trajectory events for monosized data sets.

Packing density	5%	20%	33%	50%	70%
Ball radius (mm)	0.4-4	1-10	1.8-18	3-27	7.5-50
Collision events	189	375	488	695	3127
Predict trajectory	199816	199416	199209	198912	220282

Table 4. Experimental results for for polysized data sets.

Density	5%	5%	5%	33%	33%	33%	70%	70%	70%
Algorithm	Time	TE	CC	Time	TE	CC	Time	TE	CC
Direct	1332.66	0	20657627	1284.71	0	20741959	1690.39	0	27485297
Regular sub.	103.26	1161	298667	242.66	257	3648103	1880.48	30	27358811
Spatial tree	111.99	1161	298667	247.72	257	3648103	1922.31	30	27358811
Segment tree	109.68	12432	14154	121.77	11050	91584	395.65	36450	575909
Dynamic PD	303.85	881	1934517	318.73	626	1934358	397.27	735	2221611

It was established that the number of collision checks for the dynamic DT is independent of the distribution density. Interestingly, this number approaches the number of collision checks for monosized data set. Number of collision checks for the segment tree method is approximately 14 times less than that of the dynamic DT method in case of low packing density. It is four times smaller than the number of collision checks in DT for distribution with high density of packing. The opposite trend can be noticed in regards to the number of topological events. This number twice smaller for DT method for the distribution with low packing density and almost 30 times smaller for the high density particle distribution.

The graph exploring elapsed time versus density of packing relationship is found in Fig. 2. It can be observed that the DT is a very steady method with the time behaving as a constant function. As the density increases over 45% , the regular subdivision and the spatial tree methods require more time than the dynamic DT or the segment tree method. The segment tree method outperforms the dynamic DT method for low densities. It matches the performance of the DT method at packing densities close to 70%.

5.3 Increasing the Number of Particles

In the final series of experiments, the number of particles gradually increases from 10 to 100 with their radii selected in the 5 to 10 mm range. Density of particle distribution varies from 4.5% to 35% (see Table 5). Graph in Fig. 3 illustrates the dependance of the elapsed time on the number of particles. It can be seen that the time required for the straightforward algorithm grows as a quadratic function while the time required by all the other methods shows just a linear growth. Once again the segment tree outperforms all other methods while

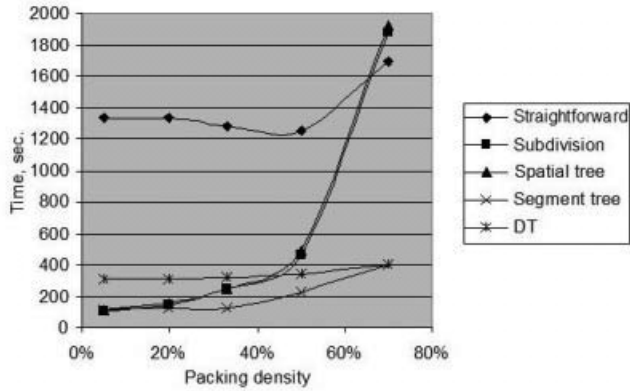


Fig. 2. Time vs. packing density for polysized data sets.

Table 5. Experiments for shaker ball mill data set.

Number of particles	10	20	30	40	50	60	70	80	90	100
Density	4.5%	8.5%	12.5%	15.0%	20.0%	23.0%	26.5%	31.0%	33.5%	35.0%
Collision events	3	5	8	24	31	52	47	79	121	128
Predict trajectory	4014	8021	12026	16006	20000	23985	27972	31902	35871	39854

the dynamic Delaunay triangulation method requires the most time among all collision detection optimization data structures.

6 Conclusions

Based on the results obtained, suggestions about the most appropriate data structure and algorithms for some other simulation problems can be made. They are applicable to any simulation model that can be described by specifying the number of particles, their size distribution, density of packing and functions defining their trajectories. The guidelines are:

1. The regular spatial subdivision, the regular spatial tree and the segment tree methods show similar performance for monosized particle systems with low packing density.
2. The segment tree method outperforms the other methods for monosized particle systems with high packing density (higher than 50%).
3. The dynamic Delaunay triangulation method performs worst for monosized distributions.

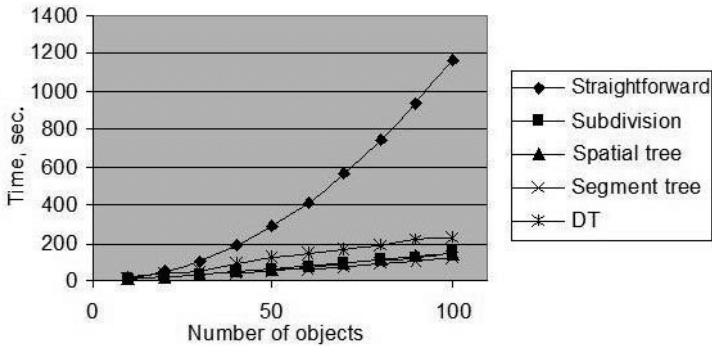


Fig. 3. Time vs. number of particles.

4. The dynamic Delaunay triangulation method is as efficient as the segment tree method for polysized high-density distributions (when the density of the distribution is 70% or higher). Both of them outperform other methods for this type of simulated system.
5. The segment tree method is the best method for polysized low-density distributions (for density lower than 70%).
6. The regular spatial subdivision and the regular spatial tree methods can perform almost as bad as the straightforward method for polysized high-density distributions (density higher than 50%) and thus shouldn't be used for collision detection optimization in such systems.
7. The regular spatial subdivision and the regular spatial tree show similar performance with regular subdivision method being 5% faster but requiring more space. The space required for DPT, spatial tree and segment tree method is practically the same.

Some general conclusions can be drawn from the above discussion. Based on the overall performance, the segment tree can be considered as the best candidate for almost all types of the systems considered. The dynamic generalized Delaunay triangulation method demonstrates consistent performance that practically does not depend on the distribution of radii of the particles or their packing density. It is shown that despite its complexity the method is a good candidate for collision detection optimization for polysized particle systems.

References

1. Agarwal, P., Guibas, L., Murali, T. and Vitter, J. Cylindrical static and kinetic binary space partitions, 13th Annual Symp. on Comp. Geometry (1997) 39–48.
2. Beazley, D.M., Lomdahl, P.S., Gronbech-Jensen, N. and Tamayo, P. A high performance communications and memory caching scheme for molecular dynamics on the CM-5, 8th Int. Parallel Processing Symposium, Cancun, Mexico, (1994) 800–809.
3. Benjamin, J.S. Dispersion Strengthened Superalloys by Mechanical Alloying, *Metallurgical Transactions*, **1** (1970) 2943–2951.
4. Gavrilov, D., Vinogradov, O and Shaw W. J. D. Simulation of grinding in a shaker ball mill, *Powder Technology*, **101**(1) (1999) 63–72.
5. Gavrilova, M. Collision Detection Optimization using Dynamic Data Structures, *Proceedings of the ICCS'02* (2002), in print.
6. Gavrilova, M. and Rokne, J. Swap conditions for dynamic Voronoi diagram for circles and line segments, *J. of Computer-Aided Geom. Design*, **6** (1999) 89–106.
7. Gottschalk, S., Lin, M. and Manocha, D. OBBtree: A hierarchical data structure for rapid interference detection, *Computer Graphics Proceedings, Annual Conference Series*, (1996) 171–180.
8. Hubbard, P. Approximating polyhedra with spheres for time-critical collision detection, *ACM Transaction on Graphics*, **15**(3) (1996) 179–210.
9. Klosowski, J.T, Held, M., Mitchell, J.D.B, Sowizral, H. and Zikan, K. Efficient Collision Detection Using Bounding Volume Hierarchies of k-DOPs, *IEEE Trans. Visualizat. Comput. Graph.* **4**(1) (1998) 21–36.
10. Krantz, A. Analysis of an efficient algorithms for the hard-sphere problem, *ACM Transaction on Modeling and Computer Simulation*, **6**(3) (1996) 185–209.
11. Marin, M., Russo, D. and Cordero, P. Efficient algorithms for many-body hard particle molecular dynamics, *Journal of Computer Physics*, **109** (1993) 306–329.
12. McCormick, P.G. , Huang, H., Dallimore, M.P., Ding J. and Pan J. The Dynamics of Mechanical Alloying, in *Proceedings of the 2nd International Conference on Structural Application of Mechanical Alloying*, Vancouver, B.C. (1993) 45–50.
13. Medvedev, N.N. Voronoi-Delaunay method for non-crystalline structures, *SB Russian Academy of Science*, Novosibirsk (2000)
14. Vinogradov, O., Explicit equations of motion of interacting spherical particles, *Recent Advances in Structural Mechanics*, **248** (1992) 111–115.