Sphere clump generation and trajectory comparison for real particles

Mathew Price[1] Vasile Murariu[2] Garry Morrison[2]

[1]Department of Electrical Engineering
University of Cape Town
South Africa
mathew@dip.ee.uct.ac.za

[2]De Beers Group Services
Johannesburg
South Africa
(Vasile.Murariu, Garry.Morrison)@debeersgroup.com

Abstract

An automatic method for generating sphere clump models of real particles is presented. This is achieved by a process that seeks to optimise the surface correspondence between the sphere clump and a triangular mesh model of the particle. We explore the effect of the number of spheres in the sphere clump model and the extent to which realistic trajectories can be reproduced with DEM simulation. Particle trajectories of actual interactions are extracted from high-speed video sequences using a novel pose estimation technique. Comparisons of sphere clump trajectories with particle trajectories demonstrates that improved particle shape representation with multiple spheres does not necessarily imply better simulation accuracy, unless the interacting particles exhibit highly irregular mass distributions.

1 Introduction

Discrete element methods (DEM) are widely used in particle based analysis for determining the dynamic motion of a large number of particles in various processes by means of simulation. Particles are modelled by a group of basic elements (often spheres) for which the physical dynamics are well understood. This allows complex processes to be modelled and simulated.

In the event that more accurate shape representation is desired, many commercial DEM packages allow the specification of clumped elements (Itasca Consulting Group Inc., 2006; DEM Solutions, 2006). In this formulation groups of overlapping primitives are used to more adequately approximate the desired shape. One weakness lies in the generation of such a clump, since usually the user is required to interactively place the elements within a CAD type mesh model. This is a simple matter if all interacting particles are identical. However, in the case of multiple shape classes, this can be tiresome and inefficient.

Several techniques have been proposed for realistically modelling irregular particles. A broad overview is presented in Latham and Munjiza (2004). Circle fitting methods, e.g. Zeghal and Lowery (2002); Chang et al. (2003), have been popular in 2D DEM simulations where particle profiles are available. 3D sphere packing methods such as Hubbard (1996) approach the problem volumetrically. This results in sphere clumps with a high number of spheres, which are not representative of the surface curvature and do not take advantage of the fact that spheres may overlap. In addition, such clumps increase the computational overhead of the simulator and may limit the number of total particles that can be used simultaneously.

Towards providing an alternative, we present an automated 3D surface-driven method that uses the particle's mesh vertices to determine suitable sphere candidates. In our experiments, a calibrated six-camera setup is used to capture well distributed silhouette views of a particle. Volumetric carving and polygonisation is then used to produce an approximate 3D particle shape, referred to as the *visual hull* (Laurentini, 1994). An initial sphere clump is generated using a random sampling technique. This is followed by an iterative two step process that assigns mesh vertices to spheres after which least-squares optimisation is used to minimise the distance. In Matsushima and Saomoto (2002), a similar approach is used. However, the random surface sampling and clustering scheme we propose enables the clump to automatically find the optimum number of spheres for representation.

Comparative results are shown in which the similarity of various polyhedral mass properties are assessed. A second component investigates the extent to which particles modelled by sphere clumps interact more realistically in simulated processes. This is achieved by comparing simulated trajectories with those extracted from high-speed video (HSV) experiments using the method described in Price and Morrison (2007). The relation between the number of spheres in the clump and the simulated trajectory is also explored.

2 Sphere clump generation

Our primary objective is to produce a compact representation of a particle's shape that closely matches the surface of its visual hull mesh (or other 3D model representation). The method is divided into two processing stages: Initialisation and optimisation.

Initialisation involves populating the clump with suitable spheres by means of a random selection process. Iterative optimisation then adjusts the solution until the distance between the mesh surface and the combined sphere surface is minimised. It should be noted that the random element of the initialisation process means that repeated application of the method on the same particle may result in alternate sphere clump configurations. The following sections describe each processing stage in turn.

2.1 Clump initialisation

It is assumed that a 3D mesh of the target particle is available. In our case, a triangular mesh model is generated from multiple silhouette images of a particle using volumetric carving followed by polygonisation (Bloomenthal, 1994). Initialisation proceeds by repeatedly, fitting spheres to subsets of mesh vertices, which are selected at random from a selection list. The selection list initially comprises all vertices. Spheres are culled according to a distance measure and vertices that have been satisfied are removed from the selection list. Finally, nearest neighbour clustering is applied to find a set of unique spheres.

2.1.1 Randomised sphere sampling

A subset of four points (x_1, y_1, z_1), (x_2, y_2, z_2), (x_3, y_3, z_3), (x_4, y_4, z_4) uniquely defines a sphere by solving the following determinant (Beyer, 1987):

$$\begin{vmatrix} (x^2+y^2+z^2) & x & y & z & 1 \\ (x_1^2+y_1^2+z_1^2) & x_1^2 & y_1^2 & z_1^2 & 1 \\ (x_2^2+y_2^2+z_2^2) & x_2^2 & y_2^2 & z_2^2 & 1 \\ (x_3^2+y_3^2+z_3^2) & x_3^2 & y_3^2 & z_3^2 & 1 \\ (x_4^2+y_4^2+z_4^2) & x_4^2 & y_4^2 & z_4^2 & 1 \end{vmatrix} = 0. \quad (1)$$

This equation will have a unique solution as long as:

- No combinations consisting of three of the four points are collinear.
- All four points are not coplanar.

In terms of the minors of the above determinant, equation 1 can be written as:

$$(x^2+y^2+z^2)M_{11} - xM_{12} + yM_{13} - zM_{14} + M_{15} = 0. \quad (2)$$

This can be factorised in terms of the general sphere equation $(x-x_0)^2 + (y-y_0)^2 + (z-z_0)^2 = r^2$ and the sphere parameters are given by:

$$x_0 = M_{12}/(2M_{11}) \quad (3)$$
$$y_0 = -M_{13}/(2M_{11}) \quad (4)$$
$$z_0 = M_{14}/(2M_{11}) \quad (5)$$
$$r = \sqrt{x_o^2 + y_0^2 + z_0^2 - M_{15}/M_{11}}. \quad (6)$$

A list of potential spheres is generated by randomly selecting subsets of four vertices from the selection list and computing the sphere parameters. This produces spheres, whose surfaces pass through the mesh surface at the selected points, like the ones shown in the left hand image of Figure 1. By repeating this process many times, consistent sphere clusters emerge as seen in the right hand image of Figure 1.

2.1.2 Sphere culling

In certain cases, the fitted sphere can be a bad approximation to the particle's surface. For instance, if all four points of a particular subset fall close to one another, the sphere may be very large and its centre may fall outside the particle. Another example is when the points occur at a concavity of the mesh. Here, the fitted sphere may lie completely outside the mesh. Examples of these cases can be seen in Figure 2.

For this reason it is important to cull the list of potential spheres so that only suitable spheres are considered when determining the clump. We apply a two-step culling method. First, spheres whose minimum distances between their surfaces and the convex hull of the mesh exceeds a threshold $T_{\mathtt{ch}}$ are removed:

$$\min(\mathbf{D}_{\mathtt{ch}}) > T_{\mathtt{ch}}, \quad (7)$$

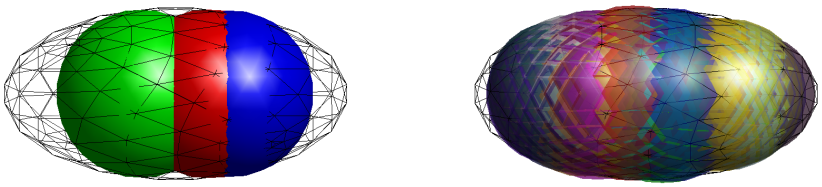

Figure 1: *Left: Example mesh with 3 spheres fitted to random subsets of vertices. Right: Example of repetitive sphere sampling.*

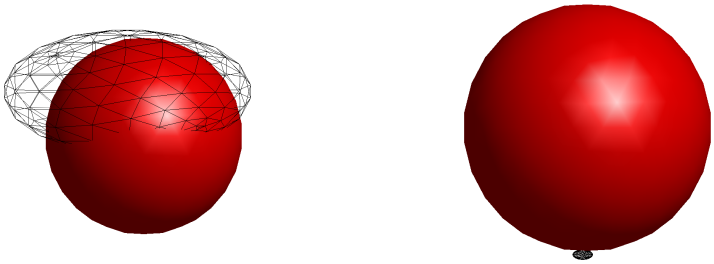

Figure 2: *Undesirable cases where fitted sphere passes through selected points, but falls outside the mesh.*

where the set of distances to the convex hull are computed by:

$$\mathbf{D}_{\mathtt{ch}} = (D_{\mathtt{ch}_1}, D_{\mathtt{ch}_2}, D_{\mathtt{ch}_i} \ldots, D_{\mathtt{ch}_n}) \quad (8)$$

$$D_{\mathtt{ch}_i} = \parallel (\mathbf{V_i} - (x_0, y_0, z_0)^{\mathtt{T}}) \cdot \mathbf{N_i} \parallel - r. \quad (9)$$

$\mathbf{V_i}$ is a vertex on the $i^{\mathtt{th}}$ face of the convex hull and $\mathbf{N_i}$ is the corresponding face normal.

Next spheres whose centres (x_0, y_0, z_0) fall outside the visual hull are removed, using a simple point-in-polyhedron test function. This step is necessary, since the visual hull mesh may have concavities which are not represented by the convex hull. Both steps are used consecutively for computational efficiency.

Following the culling stage, the minimum distance between each mesh vertex in the selection list and the set of potential spheres is computed. If a vertex is within a specified tolerance of its nearest sphere, it is deemed to be satisfied and is consequently removed from the selection list. Iterative sphere sampling and culling are applied until the proportion of mesh vertices that have not been satisfied is negligible.

2.1.3 Clustering

Once the sampling and culling stages have been completed. The spheres are clustered by binning them according to similarity. Similarity is defined as being the Euclidean distance $D_{\mathtt{sphere}}$ between two spheres' parameters, i.e.

$$D_{\mathtt{sphere}} = \sqrt{(x_{0_1} - x_{0_2})^2 + (y_{0_1} - y_{0_2})^2 + (z_{0_1} - z_{0_2})^2 + (r_1 - r_2)^2}. \quad (10)$$

If the distance between a sphere and a current list of sphere cluster centres is below a specified merging threshold $T_{\mathtt{merge}}$ then the sphere is added to the cluster. Otherwise, the sphere is added to a new cluster. When spheres are added to a cluster, the cluster centre is updated to represent the average of all its current members. After all the spheres have been assigned to their nearest cluster, the clusters are ordered according to population size.

2.2 Sphere clump Optimisation

The second processing stage is used to optimise the sphere clump computed in the initialisation stage by minimising the distance between the clump's surface and the original mesh. In our implementation we use a Levenberg-Marquardt (Moré, 1977) nonlinear least-squares based optimiser. An iterative optimisation approach is used. Mesh vertices are first assigned to a nearest sphere, after which an optimisation step minimises the distance between each sphere and its assigned vertices with respect to the sphere clump parameters. This two-step process is repeated until the solution converges.

2.2.1 Assignment step

Since the sphere clump is composed of several spheres, which may intersect, the exact outer clump surface is unknown. Therefore, in order to measure the distance between the clump's surface and the known mesh, the correspondence between the mesh vertices and their closest spheres must be determined. This is accomplished in an identical manner to the method that is used during the initialisation stage (Section 2.1.2). Mesh vertices are assigned to their closest sphere by measuring the Euclidean distance between each vertex and the set of sphere clump centres as shown in Figure 3.

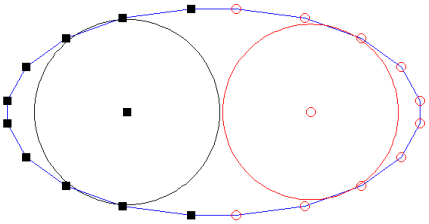

Figure 3: *Assignment step. Mesh vertices are assigned to their nearest sphere based on Euclidean distance.*

After assignment, spheres whose assigned vertex count falls below a minimum threshold are removed from the clump. This allows the clump to squeeze out redundant spheres that may be present. Additionally, spheres that fall outside the mesh after an optimisation step are also culled.

2.2.2 Optimisation step

Optimisation is parameterised by the centre and radius of each sphere in the clump. The distance between a sphere $\mathbf{S_i}(x_{0_i}, y_{0_i}, z_{0_i}, r_i)$ and an assigned mesh vertex $\mathbf{V_i}$ is defined as:

$$D(\mathbf{S_i}, \mathbf{V_i}) \quad = \quad \sqrt{(x_{0_i} - V_{i_x})^2 + (y_{0_i} - V_{i_y})^2 + (z_{0_i} - V_{i_z})^2} - r_i. \tag{11}$$

The distance between each vertex and its corresponding sphere is computed in this manner and concatenated to form the error vector used during optimisation. The optimiser thus minimises these distances with respect to the sphere clump's parameters.

Sometimes it is desirable for the user to limit the maximum number of spheres in the clump. In this case, the top n largest spheres (in terms of radius) are selected. Future experimentation will be necessary to determine whether a different selection approach should be used. Figure 4 shows the final output after optimisation. The effect of limiting the maximum number of spheres is shown in the right hand image.

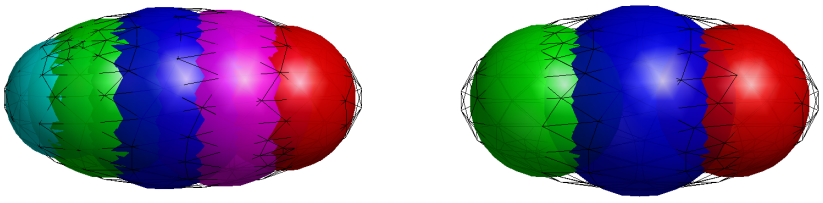

Figure 4: *Final sphere clump after optimisation. The right hand image shows the effect of limiting the maximum number of spheres in the clump to 3.*

3 Results

The purpose of generating sphere clumps is for improved prediction of particle interaction by means of simulation. Subsequently, three stages of evaluation are conducted. Firstly, sphere clumps for a test set of particles are generated and qualitatively assessed in terms of shape consistency with the original particles. Polyhedral mass properties are then used as a quantitative measure of similarity. Finally, the clumps are applied to DEM simulations and their trajectories are compared with those extracted from HSV (high-speed video) experiments.

3.1 Qualitative assessment

A test set of seven 3D models of diamonds are applied to our sphere clumping method. Figure 5 depicts typical sphere clump configurations for each test particle. Three test runs of each particle are conducted to demonstrate the repeatability of the method. As previously mentioned, the initial processing stage is based on a random sampling algorithm and therefore may result in alternate, but equally consistent, initial sphere clump configurations. In some

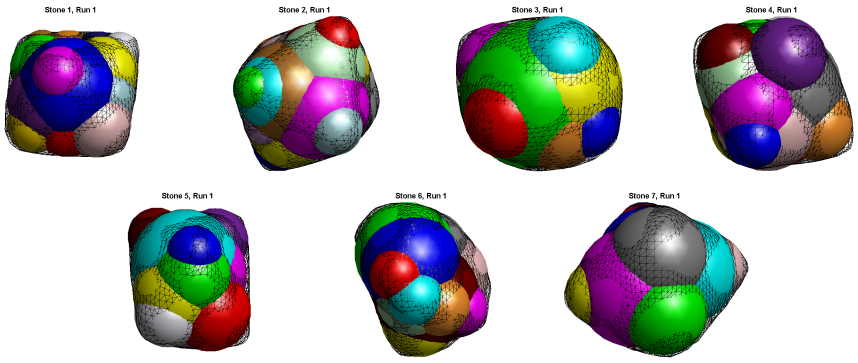

Figure 5: *Typical resulting sphere clumps and associated particle meshes for each particle.*

cases the final number of detected spheres also varies. This is caused by the domination of larger spheres, which may have not been detected in other runs. Table 1 summarises the total number of spheres detected for each run of each particle. The ability of the method to automatically scale the complexity of the final clump depending on the characteristic shape of the particle can also be seen by observing the result for the third particle. In this case, the particle is rounder, resulting in fewer spheres being required for representation.

Stone:	Run 1	Run 2	Run 3
1	18	15	15
2	19	14	15
3	10	10	8
4	14	14	16
5	13	15	17
6	12	16	12
7	9	11	14

Table 1: *Number of detected spheres in clump for each particle over three runs.*

3.2 Polyhedral mass properties

In order to quantify the performance of the sphere clump generation method, the errors of several polyhedral mass properties are computed using the method described in Lien and Kajiya (1984). Since the method operates on an explicit mesh, a sphere clump mesh is generated from the clump using the volumetric carving method previously discussed. The following property errors are computed:

Volumetric difference: The difference between the volumes of the original and sphere clump meshes. This is calculated as an RMS percentage error.

Centroid position error: The RMS percentage error in position of the centroid of the particle for each component.

Mass distribution similarity: This is the square root of the eigen values of the mass covariance matrix. It provides an estimate of the spread of mass along each of the three principal axes.

Correlation of the above properties between sphere clumps and associated particle meshes will result in similar simulated trajectories, because of their direct relation to particle mass and inertia. In our simulations we assume that the particle's density is constant. Therefore, this provides a convenient means for quantitatively forming a comparison between the original 3D mesh model and the mesh generated from the sphere clump. The covariance matrix and inertia tensor have identical eigen vectors, but different eigen values. By comparing the difference between the eigen values of the covariance matrices of two meshes, the similarity between their inertia tensors can be determined. This also allows the comparison to be independent of the actual principal directions of the meshes, which may vary greatly in the case of compact objects. Table 2 lists the RMS percentage errors computed for all the test stones over all runs. Figure 6 shows plots of the tabulated errors. Ground truth property values are plotted along

Property:	Error (%):
Volume	0.4%
Centroid	0.4%
Sqrt Eigen values	0.3%

Table 2: *RMS errors of mesh property differences measured between the sphere clump and original meshes. Errors are shown as a percentage of the true values averaged over all particles and runs.*

the x-axes, while properties computed from the sphere clump meshes are plotted on the y-axes. The graphs relating to the centroid and mass covariance comparisons are separated according to the differences of the respective x, y and z components.

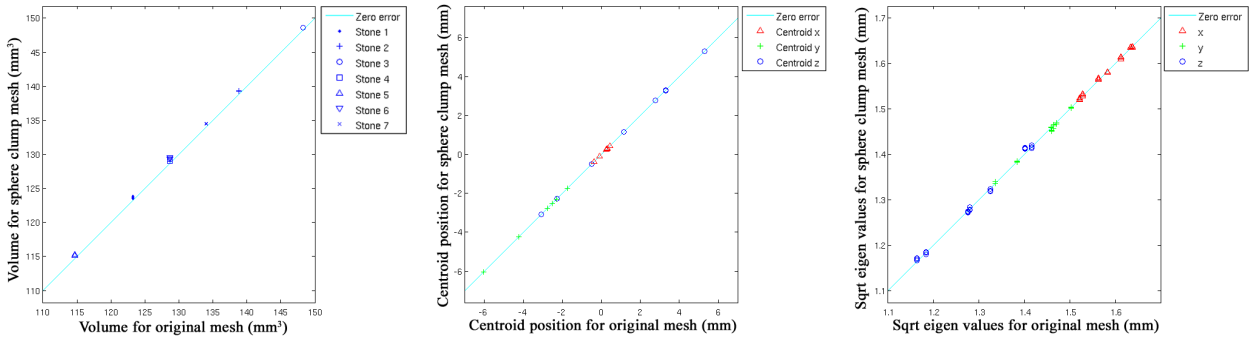

Figure 6: *Left: Volume comparison. Middle: Centroid comparison. Right: Mass covariance comparison. Original mesh properties are plotted on the x-axis while sphere clump values are plotted on the y-axis. The line* $x = y$ *represents zero error.*

The results indicate that the difference between the volumetric mesh properties of the original mesh and the sphere clump mesh is relatively small. The implication is that under simulation both models should exhibit similar motion characteristics. However, it is important to note that the actual surfaces are different. Most notably, while the sphere clump has equal principal curvatures locally, the original mesh may not. The small positive bias visible in the volume comparison in Figure 6 is attributed to the discrete nature of the space-carving method used to approximate the sphere clump mesh from the actual spheres.

It is also interesting to note that although the method will produce alternate configurations between runs, the final variance between mesh properties is exceptionally small. This shows that the method produces repeatable results even with the random sampling element. One improvement that may improve sphere clump particle replication, is to fill the small pockets that arise at the intersection between spheres. However, this is left for future endeavours.

For comparative purposes, Figure 7 shows the effect of limiting the number of spheres in the clump with regards to the error between the polyhedral mass properties. As expected, the error decreases as the number of spheres in

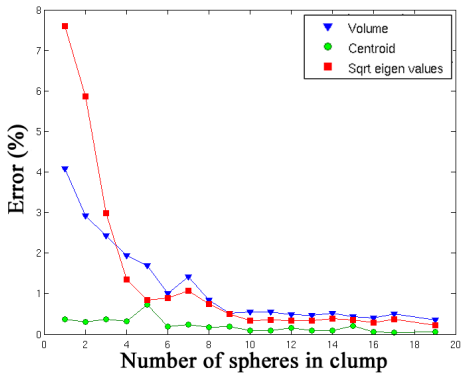

Figure 7: *Evolution of RMS error of sphere clump mesh properties versus the number of spheres in the clump.*

a clump is increased. However, an equilibrium is reached near 10 spheres where the addition of more spheres does not significantly improve accuracy. Naturally, the exact location of this point will depend on the complexity of a particle's shape. Owing to the compact nature of the test particles, the gross error between polyhedral mass properties is relatively small, even when only a single sphere is used.

3.3 Sphere clump trajectory comparison

In Price and Morrison (2007) a method is described, which allows 3D trajectories of real particles to be extracted from video sequences. The method was applied to four HSV experiments, captured at 4000 frames per second, consisting of an irregular diamond particle (Stone 6 from the test set) impacting an angled steel plate. The extracted 3D trajectories were then used as ground truth for comparison with DEM simulated impacts.

For robustness, the 3D trajectory extraction process was applied 10 times for each sequence, resulting in 40 corresponding DEM simulations. Sphere clumps of variable numbers of spheres were generated and used to represent the particle within the simulations. Figure 8 shows the post-impact linear and angular velocity error for each test. Results are shown for both the standard Hertz contact (Johnson, 1985) and the no-slip Hertz-Mindlin contact (Di Renzo and Di Maio, 2004), which appears in several commercial DEM codes. Owing to the material properties of diamond and the hardness of the steel used, we consider effects of plasticity during contact to be minimal. For the linear velocities, only the in-plane component (parallel to the HSV image plane) is used. This is because out-of-plane linear motion estimation can be unreliable due to the single HSV camera setup. Further details can be found in Price and Morrison (2007).

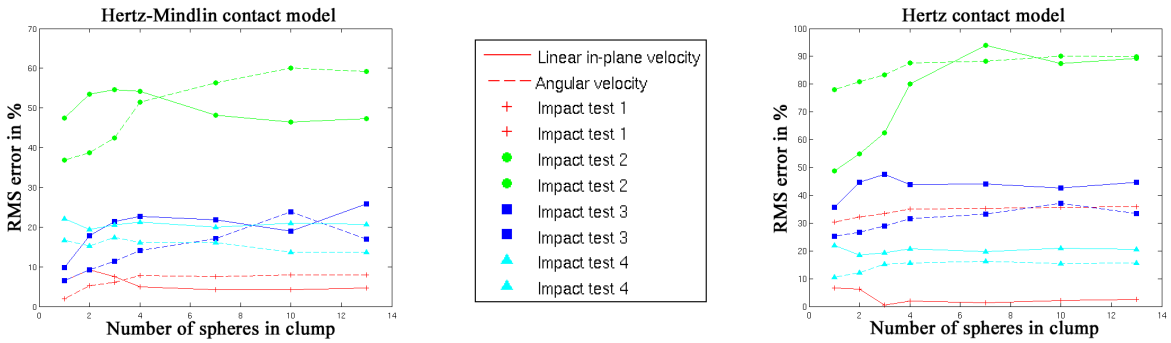

Figure 8: *Linear and angular post-impact velocity error for several HSV tests versus the number of spheres in the clump. Left: DEM with Hertz-Mindlin (no-slip) contact model. Right: DEM with Hertz contact model.*

Intuitively, one would expect simulations with a high number of spheres to achieve better accuracy. In particular, a notable variance in angular velocity would be expected between the single-sphere and multi-sphere cases. However, the results indicate that this is not the case for the particle in question. In fact, it is evident that better simulation accuracy is achieved through the use of an improved contact model (Hertz-Mindlin versus standard Hertz) than with improved mass representation with spheres.

In our tests, although the particle's shape is fairly irregular, it is also very compact. This leads to the centroid of the particle being located close to the centre of the sphere of equivalent volume. This can be verified by the consistently low centroid errors seen in Figure 7. Therefore, the rotational moment, caused by the position of the contact point relative to the centroid at the moment of impact, is similar for both the single-sphere and multi-sphere configurations. This leads to similar torques and hence similar angular velocities.

With regards to linear velocity, the rebound force is proportional to the square root of the radius of the contacting sphere (for planar impacts). Since the radius of the single-sphere configuration is typically close to that of the average sphere in a more complex clump, similar linear velocities are also encountered.

In summary, simulation accuracy cannot be significantly improved by improved shape representation through sphere clumping, unless the interacting particles are not compact and exhibit highly irregular mass distributions. This is illustrated by the comparative simulation experiment shown in Figure 9. A non-compact particle was gener-

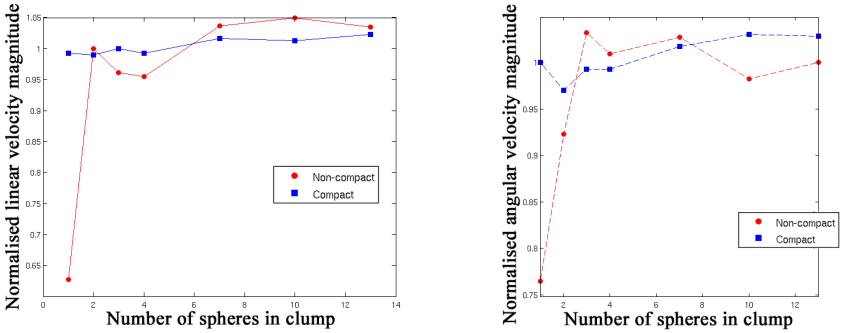

Figure 9: *Comparison of particle velocity variance for compact and non-compact particles in relation to the number of spheres in a clump. Velocities are normalised according to the median values over all results.*

ated by extruding a compact particle (Stone 6 from the test set) along one of its axes. The plots show how particle velocity variance between different sphere clump sizes is increased when the particle has a more irregular mass distribution.

Finally, although more sophisticated contact models offer improvement, ultimately the spherical assumption cannot adequately describe the underlying physical processes. This leads to simulations, which are not representative of actual particle interaction, e.g. the result shown for Impact test 2 in Figure 8.

4 Conclusions

A method has been described, which approximates 3D mesh models of particles with a clump of spheres. This allows improved particle shape representation for use with DEM simulation. The method is fully automatic, but does allow the operator to control certain aspects of the fitting process, for example the number of spheres in clump.

Random sphere sampling is used in conjunction with nearest neighbour clustering to form an initial sphere clump. A nonlinear least-squares optimisation process then minimises the distance between the sphere clump surface and the surface of the original mesh.

Qualitative and quantitative tests conducted on a sample of diamond particles has shown that the method is able to closely approximate particle shape with spheres. In particular variance of the polyhedral mass properties between the generated sphere clump and the original mesh has been found to be approximately 0.5%.

Comparative analysis of HSV and DEM trajectories of a compact diamond particle subjected to planar impact has shown that improved shape representation by using many spheres does not necessarily improve accuracy of simulation. However, particles, which exhibit highly irregular mass distributions, can be more realistically simulated using this technique. Finally, evidence suggests that sophisticated contact models may offer improvement in cases in which the assumption of a spherical surface is invalid.

References

W. H. Beyer. *CRC Standard Mathematical Tables*. CRC Press, 28 edition, 1987.

Jules Bloomenthal. *Graphics Gems IV*, pages 324 – 349. Academic Press Professional, Inc., 1994.

Y-L Chang, B-L Chu, and S-S Lin. Numerical simulation of gravel deposits using multi-circle granule model. *Journal of the Chinese Institute of Engineers*, 26(5):681–694, 2003.

DEM Solutions. *EDEM v1.1 Manual*. 2006.

Alberto Di Renzo and Francesco Paolo Di Maio. Comparison of contact-force models for the simulation of collisions in dem-based granular flow codes. *Chemical Engineering Science*, 59:525–541, 2004.

Philip M. Hubbard. Approximating polyhedra with spheres for time-critical collision detection. *ACM Transactions on Graphics*, 15(3):179–210, 1996.

Itasca Consulting Group Inc. *PFC3d Manual*. 2006.

K. L. Johnson. *Contact Mechanics*, chapter 4. Cambridge University Press, 1985.

J-P Latham and A. Munjiza. The modelling of particle systems with real shapes. *Theme issue of Philosophical Transactions of the Royal Society*, 362(1822):1953–1972, 2004.

Aldo Laurentini. The visual hull concept for silhouette-based image understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(2):150–162, 1994.

S. Lien and J. Kajiya. A symbolic method for calculating the integral properties of arbitrary nonconvex polyhedra. *IEEE Computer Graphics and Applications*, 4(10):35–41, 1984.

T. Matsushima and H. Saomoto. Discrete element modeling for irregularly-shaped sand grains. In *Proc. NUMGE2002: Numerical Methods in Geotechnical Engineering*, pages 239–246, 2002.

J. J. Moré. The levenberg-marquardt algorithm: Implementation and theory. In G A Watson, editor, *Lecture Notes in Mathematics*, chapter Numerical Analysis, pages 105–116. Springer Verlag, 1977.

M. Price and G. R. Morrison. Validating rigid body simulation of real particle shapes using pose estimation from high-speed video. *Submitted to DEM 2007*, 2007.

M. Zeghal and M. Lowery. Use of digital images to enhance discrete element modeling. In *Proceedings of the Third International Conference on Discrete Element Methods*, pages 107–111, 2002.