WHEN SIMULATION IS AS COMPLEX AS REALITY: VISUALIZATION AS A DATA EVALUATION TOOL FOR MANY-PARTICLE MODELS

Roland Reichardt, Stefanie Grollius, Peter Droste, Wolfgang Wiechert

University of Siegen, Faculty of Mechanical Engineering,
D-57068 Siegen, Germany

Wolfgang.Wiechert@uni-siegen.de (Wolfgang Wiechert)

Abstract

This contribution deals with an event discrete many particle simulator for high energy milling processes. The simulator generates flight trajectories for several thousand particles. Although the simulation is built on a simplified model of the real system, it produces a huge amount of data that – at first glance – is equally hard to evaluate as data from the real system. This makes the design of simulation experiments, parameter estimation for ball mills and the generation of meaningful and reliable results a challenging task.

Particularly, the simulation results are much harder to evaluate than this is the case for standard granular gas simulations, because the motion and trajectories of the milling balls are highly sensitive to changes in the process and model parameters and the particles are accelerated by an external energy source – the rotor. In some states the statistical distributions of particle attributes have no equilibrium, not even locally. Additionally it is possible that the system is oscillating, but becomes rapidly stable at certain thresholds of the model parameters.

The development of tailored visualization tools turned out to be an extremely useful method for generation practically useful information from such milling simulations. In fact, visualization efforts turned out to be more important and time consuming than the simulation implementation itself. It will be shown how different types of visualizations can be used for debugging the simulation code, analyzing the motion of the complete ball mill population, single particle tracking, statistical analysis of local distributions, estimation of physical parameters, and the analysis of oscillation phenomena. Some details on the implementation of the visualization tools will be given.

Keywords: many particle system, nonlinear dynamics, visualization, statistical output analysis, particle tracking,

Presenting Author's biography

WOLFGANG WIECHERT studied mathematics and computer science at the University of Bonn and obtained his PhD in 1991. From 1991 to 1996, he worked at the Jülich Research Center. In 1996, he became a full professor for simulation at the Institute of Systems Engineering at the University of Siegen.

1 Introduction

Many particle simulations are a very special domain of scientific computing. Usually these kind of simulations, e.g. molecular dynamic simulations or granular gas simulations, are done by physicians, e.g. to calculate thermodynamic properties of gases and materials. As a technical system, the high energy ball mill studied in this contribution is an untypical example. However, in contrast to many other many-particle studies, it is characterized by highly nonlinear dynamic phenomena which cannot always be described in terms of statistical equilibrium concepts.

The technical system is shown in Fig. 1. Inside of the mill's cylindrical grinding chamber the grinding media (steel balls) are accelerated by rotor blades. Powder inside the mill is reduced in particle size due to the colliding balls. Figure 1 shows a transparent grinding chamber without powder with rotor speed 0 rpm (top picture) and 1300 rpm (bottom picture). Because all balls are freely moving in the milling chamber and only pair wise collisions occur, this many-particle system can be considered as a granular gas.

A discrete event simulator (DES) has been developed to describe the motions of grinding balls inside the mill. The particles exchange their impulses and energy by surface forces only and during their flight any damping forces like air resistance is absence. Hence the ODE for the position of the balls has an analytical solution, namely the parabolic trajectory. But in this special case DES is applicable to simulate the flight paths of the grinding balls.

This article does not deal with the DES itself. The interested reader is referred to [1] regarding the simulator, [2] regarding its use in a optimization loop and [3] regarding its coupling with Monte Carlo methods to additionally describe the powder inside the mill.

The following chapters are rather concerned with visualization tools to generate meaningful practical information from the huge data files generated by the simulator. To get an impression of the generated amount of data, notice that the simulator calculates every collision event and stores it in a time ordered trace file, which are about 3 million per real time second. Every record contains the time of the collision event, the index of the involved balls, their position and the new velocity vector due to the collision response. Between two collision events the balls are flying on a parabolic flight path. Hence, it is possible to calculate the position and velocity of any ball at a arbitrary time.

This simulation output is the input for the different visualizations and statistical analysis. Particularly, the simulation results are much harder to evaluate than standard granular gas simulations, because the system is behaving in a highly dynamic and nonlinear way. The motion and trajectories of the milling balls are very sensitive to changes in the process and model parameters and the particles are accelerated by an external energy source – the rotor. In some states the statistical distributions of particle attributes have no equilibrium, not even locally.

For this reason during the development of the DES and the analysis of its simulated results it was extremely useful to visualize the data. Special visualization tools had to be developed for data evaluation. In the following chapters several applications of visualization are shown, which might also be applicable while developing other simulators. Particularly, it will be demonstrated how visualization tools can generate information that is hardly accessible by any other method.

Figure 1: The real System: Balls are moving quickly inside a cylindrical system driven by a rotor in the centre.

2 Visualization as a Debugging Aid

The particles in the example system are moving along circular pathways inside a hollow cylindrical system border (Fig. 1). The geometrical constraints given by the cylinder and rotor geometry are difficult to assure in the simulation code because the slightest numerical roundoff errors can produce strange effects.

For example, due to an implementation failure or caused by rounding errors it could happen that balls are leaving the system boundary (Figure 2, one blue ball is outside). This situation occurs very rarely and therefore it is hard to detect in the debugging phase. Debugging of DES is generally difficult, mainly because of two reasons:

- The DES is implemented in C++ and its release version is compiled with code optimization, which includes optimization of the floating point operations. Usually this results in slightly different results of the mathematical calculations in comparison to the debug version of the DES. Due to the different rounding procedures of the debug and release version the calculation of the collision events are different after a number of collisions. If a ball of the release version leaves the system boundary at a certain event number, it does not in the debug version. Furthermore the execution time of the debug version is about 10 times slower than the release version.

- DES calculates collision events directly without "moving" the particles about a time step between the collisions. Hence, balls are able to penetrate in case of an implementation failure or they are able to leave the system boundary, if the collision with it has not been found. For example the DES finds a collision of two balls as the next event, but a collision of the wall should happen before. In this case the ball leaves the system boundary, moves outside on a parabolic flight path, reenters the system and collides with the predicted ball. Consequently, all balls are inside the system, if their position is checked at collision events only.

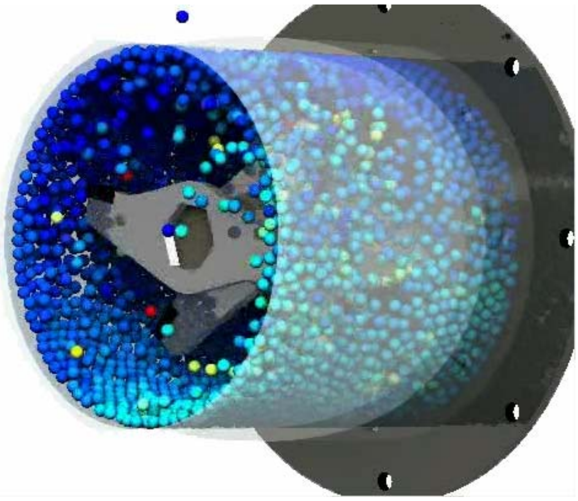


Figure 2: The blue colored ball on top has left the cylindrical system due to numerical rounding errors.

In contrast, it is easy to detect abnormal behavior in motions for a human being just by visual observation. Hence, these particles where easy to detect in the computer animations and it was possible to eliminate the reason for the problem by debugging the source code just before this event.

For the photorealistic animations the software POV-RAY [5] has been used. The ball's position and velocities are calculated at certain time steps and a script for POVRAY is generated for every time step. The animation video is made of these single pictures. A compute cluster is used to accelerate making of the movie.

3 Motion analysis

Even if 5000 or even more particles are shown in the animation of the example system at once, visualizations help to estimate the main properties of motions very quickly, e.g. that most of the particles are moving on a circular path at the system's border, whereas the particles nearby the centre are moving randomly (Figure 2).

This information is important for geometrical design optimization of the milling chamber which has been performed in [2]. For this purpose, the mill geometry has been changed in the simulator. Fig. 3 shows an optimized geometry with a non cylindrical wall. Clearly, it was very important to study the behavior of the balls when they are moving closely to the sinusoidal wall.

Particularly, beside the optimization target function, the maximization of the collision velocity, some restrictions have to be observed, e.g. that all balls are moving nearby the system border (no *dead zones*), even if they move through the valleys of the shape (Figure 3). This restriction has been set by mechanical process engineers, and the visualization of the grinding media's motion is helpful at the evaluation of the optimal shape. It has been detected just by observing the overall motion of the particles, that some shapes are not valid for productive usage, because some balls are lying in the valleys (Figure 4). In this case the valleys turn out to be too deep.

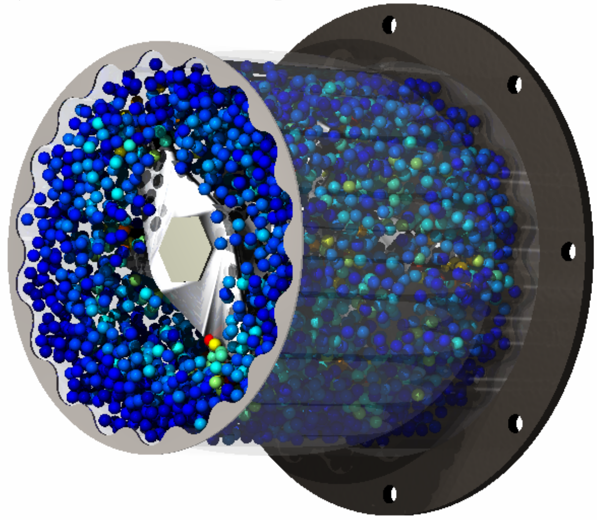


Figure 3: Design optimization of the grinding chamber. Balls must move through the valleys of the shape, which is checked by visual observation.

4 Statistical analysis on cross sections

For the statistical analysis of the ball motions the 3D system has been cut in 2D planes in axial and radial directions and these planes have been subdivided into sectors and rectangles respectively [4]. Standard volume visualization tools available in MATLAB divide the volume into Cartesian voxels, which is applicable in the centre of this example system, but not at its boundary. Hence the cylindrical volume has been subdivided into axis parallel sectors of the same volume, whereby the number of sectors can be freely chosen within some limitations due to integer rounding [4]. Sectors are similar to a radial cross section.

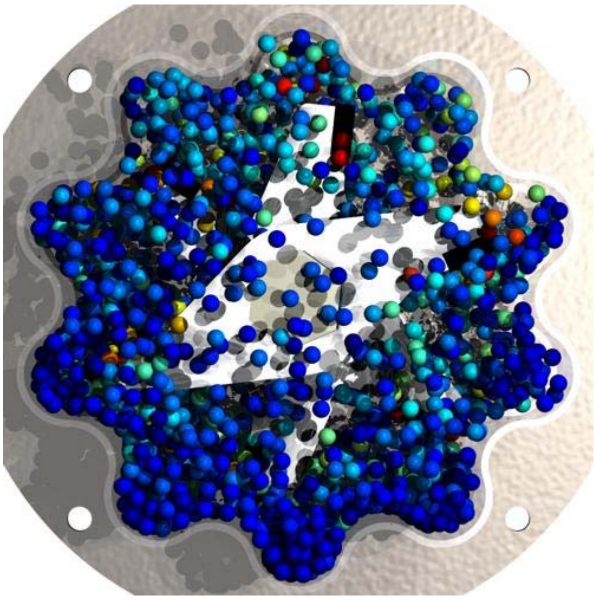


Figure 4: Design optimization of the grinding chamber. In this case the valleys are too deep. Some balls are lying in the valleys and the others are moving above them.

Another problem of 2D volume visualization of a cylinder is the subdivision into rings with profiles of equal area. Because the volume of the inner rings is less than the volume of the outer rings, the statistical data have to be divided by the volume of the rings, to keep them comparable (data per volume). A cut through these rings is similar to a axial cross section.
Special properties like the mean collision velocity or the sum of collision events within these areas have been visualized by color coding. As a result the intensity of collisions of the particles inside the system can be evaluated qualitatively. In Figure 5, bottom, a snapshot of the corresponding ball locations is shown.
The velocity per ball is color coded. In Figure 5, top and middle position, this situation is analyzed by the sectors. The picture in top positions shows the number of balls per sector color coded. It is observable that balls are more concentrated in the upper left and lower right region of the systems border. In the photorealistic picture, this information remains hidden. The middle picture of Figure 5 shows the mean velocity per sector. Red (high speed) balls are observable in 30° steps which is equal to the angular offset step size of the rotor blades.
The number of balls situated in the center region of the cylinder can be by orders of magnitudes smaller than close to the cylinder wall. For this reason, to obtain a meaningful color encoding of the ball frequencies, a logarithmic scale turned out to be well suited. It is also used in Fig. 5.

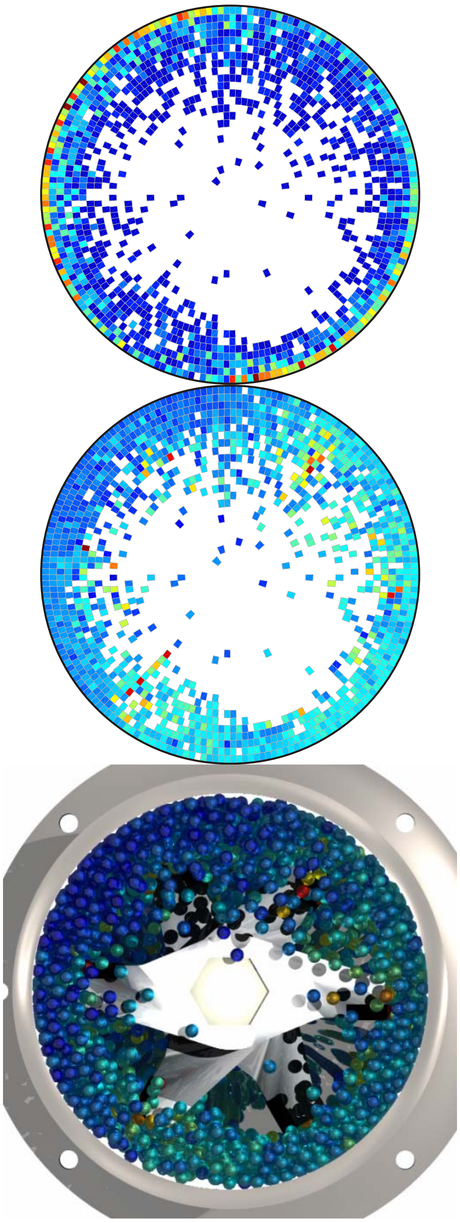


Figure 5: Top: The number of balls per volume is color coded. Middle: The mean velocity of the balls per volume is color coded. Bottom: Photorealistic snapshot of the situation of the balls. The velocity per ball is color coded.

Figure 6 shows an axial cut through the ring volumes. The collision velocity is color coded. The highest collision velocity is in the domain of the five rotor tips

5 Interactive motion tracking

A special program has been developed to visualize single particles interactively. The visualization is implemented in C++ and uses the *Coin3D package* [6] to render the objects in OpenGL. Hence, the execution speed depends on the performance graphic card. The graphical user interface (GUI) has been implemented using the *Qt class library* [7]. The software reads the records of the collision database and interpolates the position of the balls between the collision events at certain time steps. The step size and thereby the animation speed is arbitrary selectable, e.g. for slow motion. In opposite to animated videos it is possible to change the camera position while animation (Figure 7).

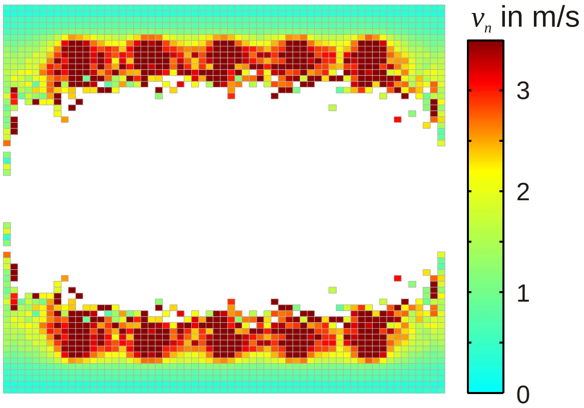


Figure 6: Axial cut: The collision velocity is color coded. The highest collision velocity is in the domain of the five rotor tips

Single particles can be shown, whereas most other particles are hidden. By doing this, it is possible to track the trajectory of one particle. It is observable that particles remain in their axial position within a certain region. This information was helpful for process engineers, because if the balls are not mixed in axial direction it is the same with the milled powder.

Furthermore it is possible to show just the particles within a certain range of the velocity distribution. Due to this feature it was possible to observe, that the particles have a high velocity just after a collision with the rotor blades only, and they are slowing down rapidly after a few collisions with other balls.

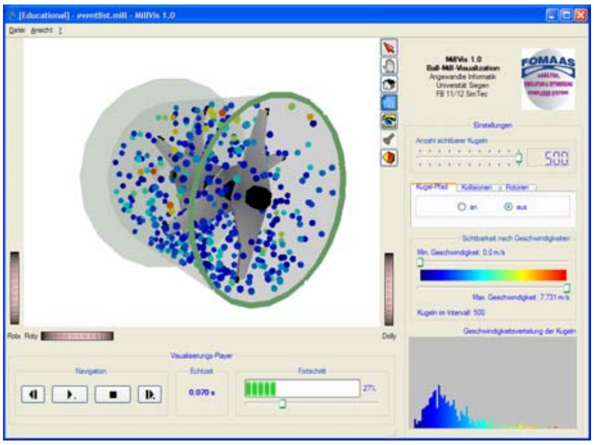
The history of the ball's trajectory can be visualized as a comet's tail.

6 Parameter estimation

Some ball collision parameters are extremely difficult to measure and there is in fact no data available. Particularly, the knowledge of rotation parameters is very important for the realism of the simulation. For example, oil coated surfaces of the steel balls are almost unable to transfer rotational energy between balls, in contrast to rusted surfaces.

Rotational energy transfer between balls depends mainly on the roughness and coating of the surface of the balls. Precisely, the so called rotational restitution coefficient describes the transmission of rotational energy during a ball-to-ball collision.

Although high speed videos of transparent milling chambers are available (Fig. 1) it is almost impossible to evaluate ball-to-ball collisions in such a video. The reason is that no information on ball rotation and the depth coordinate in 3 dimensions is available.



Anzahl sichtbarer Kugeln: 500

○ an ● aus

Min. Geschwindigkeit: 0.0 m/s

Max. Geschwindigkeit: 7.731 m/s

Kugeln im Intervall: 500

Echtzeit: 0.070 s

Fortschritt: 27%

Figure 7: Interactive visualization with an OpenGL Software.

Interestingly, it was possible for the first time to estimate the rotational restitution coefficient by comparing a ball mill visualization with a high speed video. (Figure 8). To this end the simulation was run with different restitution coefficients (Fig. 9). It turned out that the rotational restitution coefficient very sensitively influences the thickness of the ball boundary layer. Consequently, just by visual comparison between videos and simulation animation, it was possible to estimate the parameter for the rotation within a tolerance of 1%.

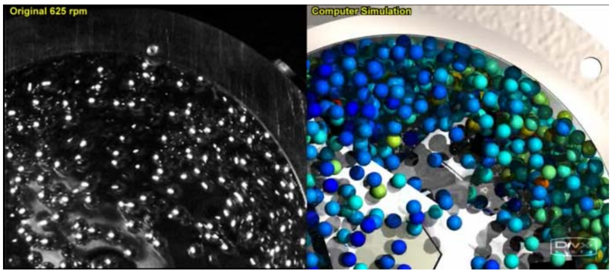


Figure 8: Snapshot of a rendered video (right) and the high speed video (left). The velocity of the balls is color coded (right).

7 Exploring oscillations

Less rotation of the balls leads to a higher density of balls at the inner surface of the grinding chamber (Figure 9, bottom right). Furthermore, the balls are synchronizing their speed. This leads to an oscillation phenomenon that has been experimentally observed only recently. The oscillation cannot be observed for balls with a high rotational energy coupling.

Generally, the system tends to oscillate in dependence of certain model parameters. Oscillation in this context means, that the balls are moving along the inner surface of the grinding chamber, all with nearly the same speed. Because of the gap between the tips of the rotor blades and the inner surface, the moving balls are not hit by the rotor. Hence, these balls reduce speed due to energy loss by ball-ball and ball-wall collisions. When their speed drops under a critical threshold, all balls are falling into the chamber, nearly at the same time. There, the balls are accelerated by the rotor towards the wall again.

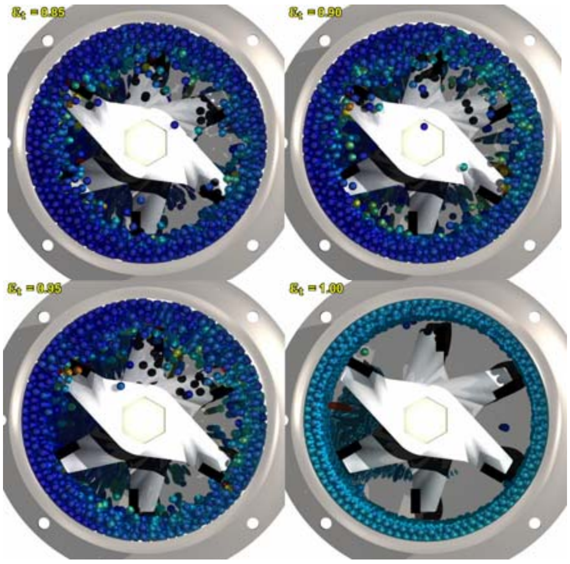


Figure 9: Influence of the rotation parameter. The sequence top-left, top-right, bottom-left and bottom-right shows the influence of less rotation.

The frequency of the oscillation changes proportional to the value of one of the model parameters, namely the rotation parameter, but if this parameter reaches a certain threshold, the oscillation disappears immediately and the system becomes stable. It is currently not very well understood why this strange nonlinear behavior is encountered and what the major influencing factors are. The reason for this behavior is currently being investigated by visualization, e.g. of the median of the velocity distribution.

Furthermore this effect has been studied by analyzing the speed changes. In a transition matrix, shown in Figure 10, the probability for a speed change from velocity v_1 to velocity v_2 is color coded. The probability of zero is transparent. The sum of the probabilities per line is normalized to one. The diagonal line in the transition matrix denotes that the velocity has no speed particles can change their velocity to nearly any other velocity, but low speed particles (about 1 m/s), keep their speed. These particle are located at the system border.

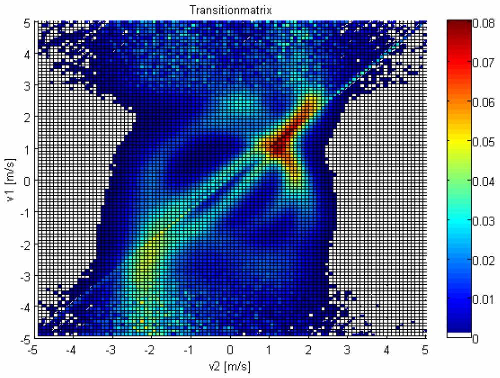


Figure 10: color coded transition matrix.

8 Conclusion

The high energy ball mill studied in this contribution is a special example for a many particle simulation. The grinding ball's motions have been simulated by a discrete event simulator. Since the system has a highly dynamic nonlinear behavior standard statistical tools are not always suitable to draw practically meaningful conclusions from the simulation runs.

It has been demonstrated how specially tailored visualization components can be used in this situation to make sense of the huge amounts of data produced by the simulator. Four types of visualization tools have been implemented for this purpose:

1. Photorealistic animations have been generated by ray tracing software. It has been shown, that applications for these animation are, debugging, monitoring of the motion of particles and parameter estimation by comparing animations with the real system.
2. The statistical analysis of the system has been visualized by color coded sections through volume elements using MATLAB. The shape of the volume elements have been chosen to fit the system.
3. The oscillations of the system caused by the rotation parameter of the balls has been analyzed by a color coded transition matrix of the velocity changes between time steps.
4. For interactive animations of individual ball paths an special OpenGL software has been presented.

References

[1] Reichardt, R.; Wiechert, W.: Event Driven Algorithms applied to a High Energy Ball Mill Simulation, *Ganular Matter*, 2007, DOI: 10.1007/s10035-006-0034-y

[2] Reichardt, R., Wiechert, W.: *Design Optimization of a High Energy Ball Mill by Discrete Event Simulation*, 21st EUROPEAN Conference on Modelling and Simulation ECMS 2007, June 4th - 6th, Prague, Czech Republic, 2007

[3] Reichardt, R., Wiechert, W., Weinhold, F.: *Particle Based Modelling of Mechanical Alloying by Parallel Monte Carlo Methods*, 21st EUROPEAN Conference on Modelling and Simulation ECMS 2007, June 4th - 6th, Prague, Czech Republic, 2007

[4] Reichardt, R.: *Ereignisdiskrete Simulation einer Hochenergie-Kugelmühle*, PhD Thesis, 2005, urn:nbn:de:hbz:467-1914

[5] Persistence of Vision Raytracer: http://www.povray.org/

[6] Coin3D package: http://www.coin3d.org/

[7] Qt class library: http://trolltech.com