The speed up factor of using a three-dimensional grid can be estimated analyzing these results and the algorithm itself.

Considering a square two-dimensional grid with a resolution \( n \times n \), the maximum number of particle visited by a ray is \( 2n - 1 \), as verified in [11]. Having this in mind, if each cell has an average of \( \mu \) particles, the maximum number of particles considered for a ray is

\[
   n_{\text{av}} = \mu (2n - 1) .
\]

Using a ray tracing system with no acceleration technique to render a particle system is much more demanding. Considering a very simple case where only one particle is intercepted in each cell, \( 2n - 1 \) particles are processed. However, to determine what particle is to be processed in each cell, it is necessary to compute the intersections with all particles of a system, that is, \( \mu n^2 \) particles.

Then, the speed up factor can be estimated and is given as:

\[
   s = \frac{\mu n^2 (2n - 1)}{\mu (2n - 1)} .
\]

Simplifying the expression reduces the speed up factor to \( n^2 \). The same procedure can be extended to three dimensions and results in a speed up factor of \( n^3 \). Observe that, in this estimation, the time to transverse the grid is not considered, but even so the speed up is considerable and enough to validate the technique.

Another topic to be analyzed is the memory requirements reduction. Considering that the reduction achieved was over 70% and the method is approximative, it is necessary to verify the error factor introduced by these approximations.

To analyze the error factor of the approximations, 100 millions three dimensional points were generated randomly. Then these points were reduced and after restored to their float point representation. The error factor for each particle was computed using the following equation:
\[ \epsilon = \frac{\| \vec{w} - \vec{v} \|}{\| \vec{v} \|} \]

where \( \vec{v} \) is the original point and \( \vec{w} \) is the point reduced and restored to its float point representation. The average and maximum error factors are shown in table 3.

Observe that the average error factor of the discrete reduced representation is smaller than the average error factor of the spherical representation. However the maximum error of the spherical representation is smaller. This is due to the fact that spherical representation uses the vector modulus to represent the point and it doesn’t use a constant spacement to represent the Cartesian coordinates of this point. As a result of that, when a vector modulus is very small, the error factor using discrete reduced representation is large, but this doesn’t happen using spherical representation.

<table>
<thead>
<tr>
<th></th>
<th>Average error</th>
<th>Maximum error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Discrete Reduced Representation</td>
<td>4.4 x 10^{-5}</td>
<td>0.015395</td>
</tr>
<tr>
<td>Spherical Representation</td>
<td>7.76 x 10^{-4}</td>
<td>0.009621</td>
</tr>
</tbody>
</table>

Table 3. Error factors obtained using the discrete reduced and spherical representation.

The error factors in table 3 show that both the discrete reduced representation and the spherical representation are suitable for use in particle system modeling and animation.

4 Conclusions and Future Works

This work presents a technique for integrating particle systems and ray tracing engines. The technique uses a three-dimensional grid to achieve feasible processing times to render an environment containing particle systems with massive number of particles.

The technique presented in this work for integrating particle systems and ray tracing systems, compared to traditional techniques for rendering particle systems in a complex scene, has the following features:

1. It is possible to obtain realistic effects, such as reflections, shadows, transparency, using a ray tracing engine;
2. Using a modulation function in transparency and color computation of a given particle solves aliasing problems that some particle systems rendering engines usually have;
3. The memory requirements for storing particle systems can be considerably reduced using the discrete reduced representation and spherical representation;