1 Introduction

We present an interactive algorithm to compute Voronoi diagrams for protein molecules. In the research area of biochemistry, a molecule is generally represented as a set of 3D spheres with various radii. In this paper, we propose a method to compute Voronoi diagrams for a set of spheres in the 3D discrete domain. We achieved interactive construction of Voronoi diagrams through our adaptive subdivision scheme and massively parallel processing supported by current graphics hardware.

2 GPU Based Voronoi Diagram Computing

We get the Voronoi diagrams for the given sphere sites. Each sphere site \( s_i \) is described as its center point \( c_i = (x_i, y_i, z_i) \) and its radius \( r_i \). Let \( S = \{ s_i | 1 \leq i \leq n \} \) be a set of sphere sites in the 3D space. From a query point \( q = (x, y, z) \), the Euclidean distance to a sphere site \( s_i \) is calculated as \( \text{dist}(q, s_i) = \| q - c_i \| - r_i \). The Voronoi region of a sphere site \( s_p \) can be formed as:

\[
\text{reg}(s_p) = \{ p | \text{dist}(p, s_p) = \text{dist}(p, s_i), \forall s_i \in S, s_i \neq s_p \}
\]

The above algorithm should be executed for all voxels, and thus its CPU-based sequential execution may require a considerable amount of time. We achieved its interactive execution with CUDA-based massively parallel implementation. Voxel-related data are stored into the CUDA texture memories and the above voxel-specific algorithm is realized as a CUDA kernel program to finally be executed as CUDA threads. Overall CUDA framework with adaptive refinement can be represented as follows:

```
procedure framework
  do
    step 1. subdivide the space into 8 × 8 × 8 voxels, which is the current CUDA hardware limit.
    step 2. invoke kernel program for each voxel
    step 3. find any voxels belonged to boundary areas
    step 4. if any, process framework recursively for those voxels.
  until user-specified accuracy limit is achieved.
```

3 Experimental Results

As shown in Table 1, we achieved at least 15 times and at most 129 times faster performance in comparison with CPU-based sequential implementations. Our CUDA-based implementation can be used in an interactive manner, while CPU-based ones not.

<table>
<thead>
<tr>
<th>subdivision</th>
<th>execution time (msec)</th>
<th>Speed ups (times)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU-based</td>
<td>CUDA-based</td>
<td></td>
</tr>
<tr>
<td>32 × 32 × 32</td>
<td>750</td>
<td>15.63</td>
</tr>
<tr>
<td>64 × 64 × 64</td>
<td>6,016</td>
<td>87.19</td>
</tr>
<tr>
<td>128 × 128 × 128</td>
<td>47,750</td>
<td>118.19</td>
</tr>
<tr>
<td>256 × 256 × 256</td>
<td>381,625</td>
<td>129.85</td>
</tr>
</tbody>
</table>

CPU: Intel Core2 Duo E8400, 3GHz CPU with 3GB RAM
GPU: nVIDIA GeForce GTX 285 with 1GB Video RAM
with 1468 sphere sites from a protein molecule description file.

Table 1: Execution times for our implementations

4 Conclusions

In this paper, we presented an interactive-time algorithm to compute Voronoi diagrams for protein molecules. With respect to the user-specified accuracy limits, our algorithm shows at least 15 times to at most 129 times better performance in comparison to the single-core CPU-based implementations. Based on our work, we expect to develop interactive-time algorithms to compute the volume of the protein and molecular surfaces as further researches.