Final project: Multi-Resolution Isosurface Rendering

NAME: QIYUKUN STUDENT NUMBER: 2020533002 EMAIL: QIYK@SHANGHAITECH.EDU.CN
and NAME: DIHAICHUAN STUDENT NUMBER: 2020533116 EMAIL: DIHCH@SHANGHAITECH.EDU.CN

1 INTRODUCTION

- Brief introduction of related paper.
- Basic task of this project.
- Load VDB data.
- Direct volume rendering.
- Data travsersal in multi-resolution grids.
- Interpolation.
- Classification.
- Some other tasks.

2 IMPLEMENTATION DETAILS

2.1 Brief introduction of related paper.

In [2021] Ray Tracing Structured AMR Data Using ExaBricks, this paper designs a techonolgy of data to enable simulations to adapt the domain resolution to save computation and storage, which is called Adaptive Mesh Refinement(AMR). This techonolgy adapts the sampling rate to the corresponding cell size and when overlap happens it will choose the finer resolution. It can also skip some empty area.

2.2 Basic task of this project.

The task of this project is to rendering four given data efficiently and beautifully. The four given data are VDB files which store results of fluid simulations. Two of them are single-resolution and two others are multi-resolution and have overlap of different resolution. Because the VDB files store a velocity field which is a vector field , we need to convert it to a scalar field to visualize it. For example, we can convert the vector field into a speed field or use Q-criterion.

2.3 Load VDB data.

In a VDB file, it contains one or more uniform girds and each uniform grid may have different resolution. Between uniform grids, there may have overlaps. In one unform gird, every cells have the same size and some cells may be empty. In metadata of a VDB file, every cells' location is contained by an origin cell's bottom front left corner world location and cell's local coordinate and gird size.In this project, we use library OpenVdb to load in VDB files.

2.4 Direct volume rendering.

There are many ways to visualize volume data. In this we choose the most directly way. We use direct volume rendering method. In this method, we generate rays to traverse the girds, and then calculate the approximate value by interpolation. If the value is in the data range we need, we give it a color based on its value, and use volume rendering to show layers of data. In our method, we think that there are vacuum between two satisfied data, so transparency will be only determined by how many layers of data it has hit not by distance.

2.5 Data traversal in multi-resolution grids.

In library OpenVdb, it gives a function to do data traversal, however the function only make sure that it will march a grid but not make ensure the step length is same. Thus, we write a our own function to do data traversal. In this function, the step is decided by the resulotion of the grid and it will skip choose the finer one when overlap happens. Also, our function can skip empty grids.

Algorithm 1 Data traversal

```
1:  function DDA($scene, ray, interaction$)
2:      $object.intersect(ray)$                ▷ intersection with surface
3:      while $t < t_{max}$ and t is in grid do
4:          if grid is active then
5:              find finest resolution
6:              doing interpolation
7:          end if
8:          $t += finestresolution$
9:      end while
10:     $sort(interactionlist)$
11: end function
```

2.6 Interpolation.

In each cell, it contains the value of its bottom front left corner, so we need to do interpolation.Basicly, we can use trilinear interpolation, which do linear interpolation on line, surface and volume of nearby 2X2X2 data. However, it will cause stripe of girds on the result. Thus, we use cubic B-spline interpolation, which use nearby 4X4X4 data.

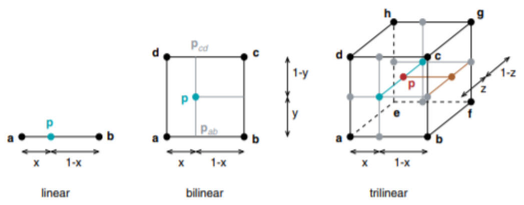


Fig. 1. Tensor-product linear interpolations: linear, bilinear, and trilinear

$$\begin{aligned} f(P_{ab}) &= (1-x)f(a) + xf(b) \\ f(P_{cd}) &= (1-x)f(c) + xf(d) \\ f(P_{abcd}) &= (1-y)f(P_{ab}) + yf(P_{cd}) \\ f(P) &= (1-z)f(P_{abcd}) + yf(P_{efgh}) \end{aligned} \tag{1}$$

Above figures and functions shows the procession of trilinear interpolation. Likely, cubic B-spline interpolation also use nearby values to approximate. Following equations showing how to process a linear cubic B-spline interpolation.

$$\begin{aligned} g(x) &= \Sigma_{i=0}^{4} w_i(u) f_i \ with\, u = x - [x] \\ w_0(u) &= \frac{1}{6}(-u^3 + 3u^2 - 3u + 1) \\ w_1(u) &= \frac{1}{6}(3u^3 - 6u^2 + 4) \\ w_2(u) &= \frac{1}{6}(-3u^3 + 3u^2 + 3u + 1) \\ w_3(u) &= \frac{1}{6}u^3 \end{aligned} \tag{2}$$

2.7 Classification.

If we choose to visualize speed field, we just need to directly calculate the norm of velocity vectors. However, if we decide to show another field calculate by velocity field like Q-criterion, we need do some pre-interpolation.

$$\begin{aligned} Q &= -\frac{1}{2}((\frac{\partial u}{\partial x})^2 + (\frac{\partial v}{\partial y})^2 + (\frac{\partial w}{\partial z})^2) - \frac{\partial u}{\partial y}\frac{\partial v}{\partial x} - \frac{\partial u}{\partial z}\frac{\partial w}{\partial x} - \frac{\partial v}{\partial z}\frac{\partial w}{\partial y} \\ f(x)' &= \frac{f(x+h) - f(x-h)}{2h} \end{aligned} \tag{3}$$

After pre-interpolation, we normalize all values.We decide the color by the value of interpolation by the function Color = ($value^{-0.3}$, $1 - value^{-0.3}$, 0) and convert it into a integer vector in range of [0,255].

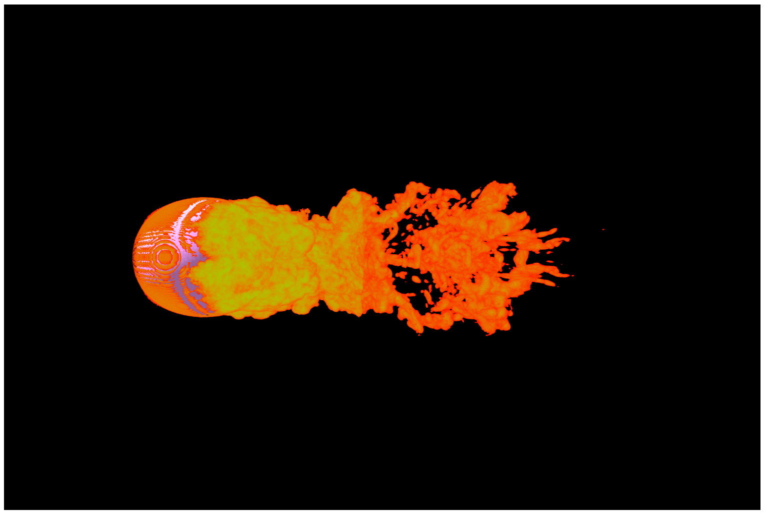
2.8 Some other tasks.

We use super-scaling method to do anti-aliasing and use BVH tree to store the mesh of sphere.

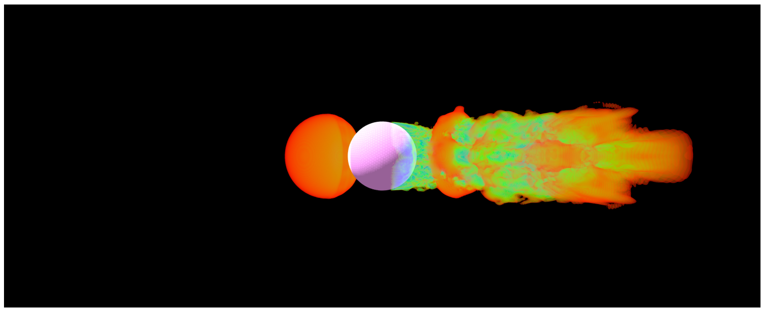
3 DIVISION OF WORK

Qi Yuchun mainly complete the load of data and data traversal in VDB. Di Haichuan mainly complete volume rendering and value interpolation.

4 RESULTS



Q-criterion



Speech field