Ocean Temperature Field 3D Visualization Key Technology Research Based on Pseudo-octree Model

Yuanyuan Feng 1, *, Beibei Han 2
1 School of Information Engineering, Zhengzhou University, Henan, China.
2 Industrial Technology Research Institute, Zhengzhou University, Henan, China.

*Corresponding author: ieyuanfengzzu@163.com

Abstract. This paper presents the pseudo-octree model to organize data which is non-uniformly sampled in the depth direction and combines with early ray termination to realize accelerated rendering based on the analysis of features of ocean temperature field. VC and OpenGL render are used to realize the 3D visualization of ocean temperature field, which can help people understand the spatial distribution of ocean temperature elements. The experimental results show that the proposed method can provide technical reference to accelerate volume rendering of scalar field data.

1. Introduction
The ocean, which accounts for 71% of the Earth's surface area, is the cradle of earth's life and an important space for mankind's survival and sustainable development. It has unique advantages in terms of resources, environment, space and strategy. The study and exploration of the oceans will enable us to better understand the ocean world and ocean phenomena and learn the changing laws of the oceans. This is of great significance to the development and utilization of oceans and the protection of the ocean environment. As one of the three basic physical parameters in oceans [1], ocean temperature has a great influence on the distribution of ocean resources [2], and most of the ocean phenomena are related to ocean temperature. Through the study and analysis of the ocean temperature, the distribution features and the variation patterns of the ocean temperature are obtained, which help humanity to improve the exploitation and utilization of ocean resources and ensure the sustainable development of the ocean environment. As an effective means of studying ocean temperature, visualization technology can visualize the effective information contained in ocean data and reveal the phenomena and scientific laws contained in oceanic data.

At present, the research at home and abroad on ocean visualization mainly focuses on two aspects: one is the ocean simulation and visualization platform (system) and the other is visualization techniques and methods. More mature and well-known visualization platforms are Google Ocean [3] developed by Google, World Wind [4] developed by NASA, and domestic EV-Globe [5,6]. The visualization of ocean data mainly focuses on visualization of vector field data [7] and visualization of scalar field data [8–10]. Volume rendering [11] is a common scalar field visualization method, which can obtain high quality images and can clearly display the internal details of the research object. Ray casting algorithm is the most widely used volume rendering method.

Based on the analysis of ocean temperature field data features, this paper proposes a three-dimensional space model-- pseudo-octree model. The pseudo-octree model is suitable for non-
equidistant sampling data and is applied to the organization of temperature field data. This model is combined with ray casting algorithm to realize the three-dimensional visualization of ocean temperature.

2. Features of temperature field Data

The ocean temperature field has the space and time characteristics, and has four dimensions, namely longitude, latitude, depth and time. The time dimension reflects the time range of temperature; the longitude, latitude and depth dimension reflect the three-dimensional space of temperature. Changes in ocean temperature are spatially continuous, but the actual data collected is discrete due to the limitations of acquisition methodologies and instrument. The sampling data has two features: (1) In the depth dimension, the seawater temperature data is sampled according to the depth of the seawater, and the data sampling layers are unevenly distributed, that is, the intervals of the sampling layers are non-equidistant. The characteristic of the sampling layer distribution is that the closer to the surface of the ocean, the denser the sampling layer is, and the more sampling data; the closer to the sea floor, the less the sampling layer, the less sampling data. (2) In the longitude and latitude dimension, the sampling data in the same depth layer is a uniform two-dimensional grid data, that is, the interval in the longitude dimension and the interval in the latitudinal dimension are both fixed values.

There are three ways to organize this type of volume data [12]:

- Treating the least interval in the depth dimension as a standard and interpolating the sampling layer in the entire depth direction. However, this way can lead to the dramatic increase of data size, which has a great influence on the later rendering efficiency.
- Ignoring the sampling interval in the depth dimension and organizing the data with the original sampling layers. This way may lead to deviations in spatial features of data and even mislead readers.
- Organizing the data with octree. This way divides data into 8 subtrees in latitude and longitude and depth dimension, which may cause fragmentation of data and increase the complexity of the data organization.

In consideration of own feature of ocean data and its application in analysis of need, the above three kinds of organization are not applicable, and as a result, this paper presents a pseudo-octree model.

3. Pseudo-octree model

Because quadtree can only store single layer data, and octree splits the data equally into 8 parts. While the data in the depth dimension is multi-layer and non-equidistant. Furthermore, in comparison with the span in the latitude and longitude dimension, the span in depth dimension is relatively small, normal octree is not applicable, so here pseudo-octree model is constructed.

3.1. Pseudo-octree model

The pseudo-octree model is described as follows: In terms of the spatial structure, the pseudo-octree refers to dividing the three-dimensional space \( V \) by \( X \) and \( Y \) two dimensions from the middle position and dividing the three-dimensional space into four regular cuboids evenly. According to the target contained in each box, it is determined whether there is a need to further divide each box until each box satisfies the requirement of no longer being divided. That is, each box is filled by one target or the size of the child cuboid has satisfied the predefined non-recoverable dividing element. All the data, as the root node of the pseudo-octree, are continuously expanded downward by the division. Each node is divided into four sub-nodes until to leaf nodes, and the leaf nodes indicate the highest resolution. When pseudo-octree divides the data, the division is only in the \( X \) axis and \( Y \) axis direction, and there are always \( N \) depths in the \( Z \) axis direction. For example, the temperature field data used in this paper has 33 layers of data in the depth dimension, that is, \( N = 33 \). Pseudo-octree spatial structure is as shown in Fig 1.
3.2. The establishment of octree model
The ordering and delamination of pseudo-octree enable fast indexing and positioning, accelerating the entire visualization process. Pseudo-octree structure in C language described as following:

```c
//Define the pseudo-octree node
template<class T>
struct Pseudo-OctreeNode
{
    T data;    // Node data
    T xMin,xMax;  //Node coordinates
    T yMin,yMax;
    T zMin,zMax;
    Pseudo-OctreeNode <T>*top_left,*top_right; //four child nodes of the node
    Pseudo-OctreeNode <T>*bottom_left,*bottom_right;
};
```

Pseudocode of creating pseudo-octree:

1. /* root is the root of the pseudo-octree and maxdepth is the maximum recursion depth */
2. CreatePseudo-Octree (root, maxdepth, xMin, xMax, yMin, yMax, zMin, zMax)
3. {
4.     Every recursion maxdepth-1;
5.     Assign a node;
6.     Assignment for node coordinates;
7.     Calculate the half-length of two nodes in two dimensions;
8.     Depth dimension, the data stored in the list lst_Depth;
9.     // Recursively create subtree (according to the location of the node to determine the location of its children)
10.    CreatePseudo-Octree ();
11.    CreatePseudo-Octree ();
12.    CreatePseudo-Octree ();
13.    CreatePseudo-Octree ();
```

Data organization diagram based on pseudo-octree is as shown in Fig 2.

![Figure 1. Pseudo-octree spatial structure](image)
3.3. Pseudo-octree coding

Pseudo-octree coding is similar with the way of octree coding, each block stores two-dimensional information, regardless of the depth interval, and only take the number of layers of data into account. The first two-bit position codes can be acquired by calculation, and the third bit position code is obtained by the data layer mapping. According to this pseudo-octree node encoding, we can easily locate any point in the space.

Assuming that the depth of the pseudo-octree is $N$, the leaf nodes are encoded as,

$$q_1q_2...q_iFF...F,$$

where

$$q_1,...,q_n \in 0,1,2,3, i \in [0,N],$$

and F is a symbol different from 0, 1, 2, and 3. The purpose of the symbol F is to ensure that the encoding length of each tree is the same. Pseudo-octree coding diagram as shown in Fig 3.

Taking the experimental data used in this paper as an example, each layer of data has $780 \times 480$ data points, requiring 

$$\log_2(780 \times 480) \approx 19$$

bits code, which means that the binary code needs 19 bits. Using pseudo-octree organization data, so each time we can determine two codes, a total of 10(according to $\frac{19}{2} \approx 10$) comparisons are required.
It is known that the span of latitude and longitude is respectively lonS ~ lonE, latS ~ latE, as shown in Fig 4.

![Figure 4. Pseudo-octree code](image)

Input lon, lat, and compare \( \text{lon} - \text{lon}_S \) with \( \frac{\text{lonE} - \text{lon}}{2} \), and compare \( \text{lat} - \text{lat}_S \) with \( \frac{\text{latE} - \text{lat}}{2} \). If both of the former is greater than the latter, then the first digit of the code is 3. If the former of the first one is greater than the latter while the second is less than the latter, the first digit of the code is 1. If both of the former is less than the latter, the first digit of the code is 0. If the former of the first is less than the latter, while the second is greater than the former, the first digit of the encoding is 2. According to the above encoding rules, \( \text{lon} - t_1 \) and \( \frac{\text{lonE} - \text{lon}}{2^2} \), \( \text{lat} - t_2 \) and \( \frac{\text{latE} - \text{lat}}{2^2} \) will be compared with, and so on, until the determination of the 10 positions encoding, then we get the first two encoding of pseudo-octree. The third encoding is obtained by mapping the number of layers in which the data is located.

4. Improvement of ray casting algorithm

Among the algorithms of volume rendering, the application of ray casting algorithm is the most common and widest for the three reasons: (1) ray casting algorithm can obtain higher image quality to meet people's requirements for rendering; (2) it’s easy for ray casting algorithm to transferred to the GPU to achieve, to meet the people's drawing speed requirements; (3) it’s easy to understand because it accords with human common sense. In view of the above advantages, this paper uses an improved ray casting algorithm to complete the visual rendering of ocean temperature field data.

4.1. Basic Theory of Ray casting Algorithm

Ray Casting algorithm is proposed by M. Levoy in 1988[11], which is a classical direct volume rendering method based on image sequences. The basic idea [12, 13, 14] is described as follows: From each pixel of the image, a ray passing through the volume data (that is, the ray traverses the entire image sequence) is emitted along the ray of sight. and then, equidistant sampling points along the ray are selected in the part which is within the volume data. The color values and opacity values of the sampling points are determined by interpolation. and then compositing all the sampling points with front-to-back or with back-to-front in order to get the sampling point in the screen color and opacity, and final get the final rendered image. The schematic diagram is as follows:
Figure 5. Ray Casting Algorithm Schematic

Although the ray casting algorithm can get high quality images, it has great calculation and long calculation time which means slow rendering. It is a research hot spots to improve the rendering speed of ray casting algorithm while ensuring image quality. The researchers at home and abroad mainly focus on the improvement in two aspects: software improvement and hardware improvement. Among these improvements, the software improvements mainly focus on the sampling of light, such as early light termination method [15], space leaping [16] and adaptive sampling [17]; while the hardware improvements mainly focus on GPU-accelerated volume rendering [18,19].

4.2. Light Early Termination

Due to the large calculation amount and low rendering speed of ray casting algorithm, in this paper, resampling step uses light early termination method to reduce unnecessary calculations. Taking advantage of the front-to-back composition, the opacity $\alpha$ of the sampling points on each projected ray is accumulated. When opacity value $\alpha$ accumulates to 1, the ray stops propagating. The sampling point is regarded as the critical sampling point, that is to say, the color and opacity of all the voxels before the sampling point are cumulatively combined to form the color value of the pixel point on the image. Voxels after the critical sampling point have no contribution to the result, and we can avoid calculating operations on these points, thus reducing the amount of computation and speeding up the volume rendering. Flowchart of the light early termination is as shown in Fig 6.

Ray may pass through the volume data at any angle, and according to the pseudo-octree division, we can quickly locate the incident point of projected ray and then resample along the projected ray.

5. Experiments and analysis

The experiment platform of this paper is Windows7, Intel Core i7-6700U processor, 8GB memory, C++ programming language, OpenGL [20] rendering engine. The experimental data we used is AIPDAILY_20100101 from the Ocean Reanalysis Data Set--AIP Ocean 2.0 in the joining area of Asia and Indian-Pacific Ocean(APIO). This data is with a time range of 1 day and a spatial range of 29.99997E to 184E and 32.28827S to 51.66615N (span of latitude and longitude), and depth dimension range of 5-5500 meters with a total of 33 layers, and the resolution of about 0.2° × 0.2°. There are 9 variables, longitude, latitude, depth and time 4 dimensions.
The data is stored in the local disk in NetCDF format and contains longitude, latitude, depth, time and attribute information. The data is extracted on demand through the interface function library of C++ version of NetCDF. This article uses the ocean temperature field data from this dataset. The features of ocean temperature field data are as shown in table 1.

**Table 1. Features of ocean temperature field data**

| variables | long_name | units     | range             | resolution | missing value |
|-----------|-----------|-----------|-------------------|------------|---------------|
| lon       | Longitude | degrees_east | 29.99997~184E    | 0.2°       | --            |
| lat       | Latitude  | degrees_north | 32.28827S~51.66615N | 0.2°       | --            |
| depth     | depths    | meters     | 5~5500 meters    | --         | --            |
| time      | time      | days       | 1day             | 1day       | --            |
| t         | temperature | Deg C     | -1.89~33.05      | --         | 0.1f          |

Visualization results are as shown in fig7-fig9.
Fig 7 is a panoramic view of the visualization of ocean temperature from a distance. Fig 8 is a partial view of the visualization of ocean temperature after closing the earth carrier. Fig 9 is a side elevation view of the ocean temperature obtained by stretching 30 times in the depth dimension. It can better show the distribution characteristics of seawater temperature in the vertical direction. It can be seen that, as the water depth increases, the temperature gradually decreases.

From the results of visualization, we can get the temperature distribution in the space characteristics: (1) As the latitude increases, the temperature gradually decreases, and the temperature near the land is low. (2) As the depth increases, the ocean temperature exhibits a non-uniform decreasing characteristic. The closer to the ocean surface, the greater the temperature change; the closer to the sea floor, the smaller the temperature change.

6. Conclusion

In this paper, we study ocean temperature field and the ray casting algorithm, by reading and preprocessing the NetCDF data, and using the pseudo-octree model to organize the data and combining with the light early termination to achieve the 3D visualization of the temperature field data. The spatial distribution of ocean temperature elements is also analyzed. Compared with the traditional ray projection algorithm, this method can make up for the shortcomings of large amount of calculation and slow rendering speed and can better represent the spatial distribution characteristics of ocean elements in the study area, especially in the research of multi-layer data visualization analysis of more advantages.

Acknowledgments

First of all, thanks to the junior, Beibei Han, for discussing and solving the problems with me when I did the experiment. And secondly thanks to the experimental data, Ocean Reanalysis Data Set--AIP Ocean 2.0, for providing the data to verify the correctness of the method in this paper.
References

[1] J. Guo and Z. Tian, "Research on Visualization of Ocean Temperature Field Based on Volume Rendering ", Marine surveying and mapping comprehensive academic seminar, 2007.
[2] W. Dong and X. Zhang, "Sphere based ocean environment scalar field element 3D visualization technology research ", Journal of Oceanography in Taiwan Strait, 2010, 29(4), pp. 571-88.
[3] http://www.worldoceanobservatory.org/content/google-ocean.
[4] https://worldwind.arc.nasa.gov/.
[5] https://baike.baidu.com/item/EV-Globe/4382314?fr=aladdin.
[6] C.C.Yan and X.Zhang, "Design of solo-soldier 3D visualization simulation in aerial defense system based on EV-Globe ", Electronic Design Engineering, 2012.
[7] H.S.Li and W.J.Niu, "Today and Tomorrow of Vector Field Visualization ", Application Research of Computers, 2001, 18 (8), pp. 11-14.
[8] C.Chen, "Research and Implementation of Visualization for Marine Scalar Field Data ", master.thesis, Graduate School of National University of Defense Technology, 2012.
[9] Johnson C and Hansen C, "Introduction" in Visualization Handbook, edited by C.D.Hansen and C.R.Johnson , pp.5-13.
[10] D.Chen, "Key Technology Study on Marine Hydrological Data Visualization", PLA Information Engineering University, 2015.
[11] Levoy M. "Volume rendering, display of surfaces from volume data", IEEE Computer Graphics & Applications, 1988, 8(3), pp. 29-37.
[12] L.S.Li, "Research on 3-D Visualization Technology for Ocean Scalar Field", master.thesis, Ocean University of China, 2015.
[13] C.Zhu, "GPU-based multi-dimensional visualization of ocean water environment technology ", master.thesis, Ocean University of China, 2015.
[14] C.Z.Liang, "GPU-based light projection algorithm ", master.thesis, Xidian University, 2009.
[15] M. Levoy, "Efficient Ray Tracing of Volume Data", ACM Transactions Computer Graphics and Applications, 1990; 9(3), pp. 245-261.
[16] R Avial, "Towards a Comprehensive Volume Visualization System", In: Proceedings of Visualization92, Boston, 1992.
[17] Ljung P, "Adaptive Sampling in Single Pass, GPU-based Raycasting of Multiresolution Volumes", Institute of Technology, 2006, pp. 39-46.
[18] H.B.Zhu, "The Analysis and Improvement of Ray Casting by GPU", master.thesis, Shandong University of Science and Technology, 2014.
[19] H. Scharsach, "Advanced GPU raycasting", Proceedings of Cescg, 2005, 16(5), pp. 69--76.
[20] E. J. Kessenich and Version, , et al, The OpenGL® Shading Language. www, 2010, 26(4), pp:853--854.