Efficient Texture Mask Prefiltering in Global Volume Rendering

Chengyu Lin1*, Meizheng Zhu1 and Jiarun Wang1

1 The 3rd Fundamental Department, North China Institute of Computing Technology, Beijing 100083, China

*Corresponding author’s e-mail: zengyoupeng@huawei.com

Abstract. In the application of ray-casted volume rendering on digital earth, the data in the region of interest (ROI) need to be pre filtered: culling or highlighting. At present, there are few researches on this problem in the field of volume rendering. Referring to the image mask in the field of image processing, an efficient pre filtering transformation technology based on texture mask is proposed: the pre filtering region is transformed into a 0-1 mask array, which is uniformly described as a two-dimensional texture. The texture is transferred into GPU Shader, and the pre filtering operation is transformed into an efficient texture query based on GPU hardware to realize filtering. And high performance rendering is based on GPU. Based on the global meteorological data, the experiments of land culling and administrative region highlighting are carried out, and the experimental results verify the effectiveness of the technology. The overall rendering effect is good, reaching the level of smooth display: the rendering frame rate is not less than 24 frames / s.

1. Introduction

Digital earth was proposed by former US President Al Gore in 1998, and has been widely concerned in the world. Google Earth, World Wind, AGI Cesium and so on are all well-known implementations of digital earth. People share global knowledge by visualizing various kinds of rich information on the digital earth. In recent years, deep space exploration, such as exploring the moon and Mars, has been strengthened, and the geospatial space of human understanding of the world has been expanded. Data visualization technology in line with human cognition is also constantly improving human cognitive ability.

Volume rendering is one of the most important visualization techniques. Volume rendering technology has a distinctive feature: it can show the internal interesting features of visual data, which is not available in general visualization. Ray casting algorithm is a typical representative of volume rendering algorithm[1], and it is also one of the mainstream methods of 3D scalar field data visualization. Ray casting algorithm is mainly used in medicine, geological exploration, radar detection, meteorology and other fields. The medical field is mainly used in nuclear magnetic resonance and ultrasonic medical imaging[2]. In geology, it is mainly used for 3D visualization of geological data[3]. Traditionally, ray casting volume rendering is mostly based on regular cube grid data, while spherical volume rendering is relatively less. Mei Honghui[4] and others designed a 3D meteorological data visualization system supporting global scale. In the system, the global volume rendering of meteorological data is realized, and the viewpoint roaming into the rendering volume is supported. Chen Jian[5] designed a GPU based spherical shell volume rendering algorithm and applied it to typhoon visualization. The development of
digital earth promotes a new requirement: volume rendering of large-scale global geospatial data. Recently, global volume rendering technology has become a research hotspot.

Due to the large amount of computation, ray casting algorithm is slow in CPU and difficult to achieve real-time rendering. At present, there are two main methods to optimize the computational efficiency: (1) optimize the sampling process of the algorithm, such as skipping invalid voxels\cite{6,7} and terminating the sampling ahead of time\cite{7,8,9}. The algorithm is accelerated by reducing the number of operations; (2) GPU is used for 3D texture sampling. GPU has many computing units and high parallelism, so it is suitable for pixel sampling parallel computing. At present, GPU is widely used in volume rendering, and there are many related research results.

However, in the practical application field, people often face the problem of eliminating or highlighting irregular areas on the digital earth. For example, it is necessary to highlight an administrative region or a country. At present, there are few researches on volume rendering. In this paper, a texture mask pre filter transform technology is proposed to solve this problem in the world wide volume rendering. The texture mask is applied to volume rendering by referring to the image mask in image processing. Main realization ideas: through coordinate transformation and statistics of irregular areas, a texture mask pre filter array is generated, which is transferred into the shader as texture. The shader can determine whether the pixels need to be filtered by obtaining corresponding mask values through high-performance texture query. The GPU hardware can transform the complex spatial interpolation geometry into simple and efficient texture query, and finally eliminate or highlight the specific area.

2. Pre filtering technology of texture mask

Texture mask pre filtering processing technology includes two core algorithms. One is to judge whether the point is in the closed area, which can realize the explicit and implicit control of the specific area pixels by judging the position of the point. The other is spherical ray casting algorithm based on GPU.

2.1. Algorithm for judging the position relationship between point and closed area

Given an irregular region, the region needs to be removed or highlighted. The core problem is to determine whether the pixels are in the irregular region. According to the classical computational geometry judgment method: the ray is emitted from the pixel, passing through the irregular area, and the number of intersections between the ray and the irregular area is judged. If the number of intersections is odd, it is in the irregular area, otherwise, it is outside the irregular area. The schematic diagram is shown in Figure 1.

![Figure 1. Schematic diagram of point position judgment with closed area](image)

As can be seen from Figure 1.1, two points a and B are inside and outside of irregular area X, respectively. The method to determine whether a and B are inside X: emit rays from two points a and B to one direction respectively, and calculate the number of intersections between ray and irregular body X. If the number of intersections is odd, it is in the irregular area, otherwise, it is outside the irregular area. There are three intersections between point a and irregular area, so it is judged that point a is inside X. Point B has two intersections with X, so point B is outside of X.
2.2. GPU based spherical ray casting volume rendering algorithm

The basic principle of the traditional ray casting algorithm [10]: light is emitted from the viewpoint and passes along the direction of the viewpoint and screen pixels. When the light enters the interior of the volume texture, the volume texture sampling starts. According to the set sampling interval, the pixel values of the obtained volume texture data are mixed and superimposed. When the light passes through the object or the transparency is greater than 1, the sampling process will be terminated. When the pixel value of superposition operation is output as the pixel value of screen, the final result of ray casting operation is obtained.

Ray casting algorithm consumes a lot of computing resources. Traditional algorithms rely on CPU for computing, which has low computational efficiency and slow rendering speed, and it is difficult to meet the real-time requirements. In recent years, with the rapid development of graphics hardware and the gradual enrichment of shader languages such as OpenGL, ray casting algorithm based on GPU has become the mainstream technology. Graphics hardware accelerates the rendering pipeline of OpenGL, and also provides many extended functions. After OpenGL2.0, programmable pipeline (shader programming interface) is provided, and programming technology moves from fixed pipeline to programmable pipeline era, so that special effects such as shading [11] and lighting [12] can be realized, and ray casting algorithm can be accelerated by GPU, among which the most representative is the algorithm proposed by Kruger [7] and Roettger [13].

Compared with CPU based computing, GPU based ray casting algorithm has the following advantages: 1. The intersection of light and object can be calculated automatically, which saves the extra computing cost; 2. Using GPU hardware computing unit to realize the complex trilinear interpolation, the speed is faster; 3. GPU has more computing units, high parallelism and faster rendering speed [14].

However, traditional ray casting algorithms generally use regular cubes as proxy geometry for rendering. However, the visualization of global meteorological data requires volume rendering on the earth surface, which brings new challenges to the texture mapping and sampling calculation in the operation of ray casting algorithm. Therefore, we first design a GPU based spherical volume rendering algorithm.

Step 1: according to the direction of the line between the viewpoint and the coordinates of the pixel, take the pixel as the starting point to emit a ray. Let the coordinates of the starting point be \( (d_x, d_y, d_z) \), the direction coordinates of the ray be \( (tx, ty, tz) \), and the sampling step be \( L \).

\[
\begin{align*}
F_x &= d_x + L \cdot tx \\
F_y &= d_y + L \cdot ty \\
F_z &= d_z + L \cdot tz 
\end{align*}
\]

(1)

Step 3: change the texture coordinates of the body to the spherical coordinate system coordinates \( (G_x, G_y, G_z) \):

\[
\begin{align*}
G_x &= \arctan \left( \frac{F_y}{F_x} \right) \\
G_y &= \arctan \left( \frac{F_z}{\sqrt{F_x^2 + F_y^2}} \right) \\
G_z &= \sqrt{F_x^2 + F_y^2} \cos G_y
\end{align*}
\]

(2)

Step 4: normalize the coordinates of spherical coordinate system and map them to the range of texture coordinate system. The final texture value is obtained by trilinear interpolation of texture coordinates.
WGS-84 coordinate system is generally used in digital earth, which is based on reference ellipsoid. However, due to the small difference between long radius and short radius, ellipsoid approximation is often used to replace ellipsoid in practical application. The difference between GPU based spherical volume rendering algorithm and traditional volume rendering algorithm lies in the different calculation methods of texture coordinates. The traditional volume rendering algorithm is based on the trilinear interpolation in rectangular coordinates, while the spherical volume rendering algorithm is based on the trilinear interpolation in spherical coordinates. As shown in Figure 2:

![Texture coordinate interpolation along sphere surface](image)

Figure 2. Texture coordinate interpolation along sphere surface

In the process of light sampling through the sphere, the traditional trilinear interpolation is to interpolate the texture coordinates of the sampling point P in Cartesian coordinate system. However, the data of spherical volume rendering is often organized according to longitude, latitude and height. Therefore, we transform the coordinate of P point from Cartesian coordinate system to spherical coordinate system, and logically establish the mapping relationship: X axis and Z axis correspond to longitude and latitude, and Y axis corresponds to height. In this way, the trilinear interpolation of texture coordinates in spherical coordinate system can meet the requirements of data spherical distribution in spherical volume rendering.

2.3. Texture mask pre-filtering algorithm

On the basis of GPU based spherical volume rendering, we propose a texture mask pre-filtering (TMPF) algorithm based on computational geometry to achieve the effect of region highlighting or eliminating. According to the position information of the drawing area, a texture mask array is generated and passed into the shader in the form of texture. The shader queries the mask of the currently drawn pixel to determine whether to highlight or reject the pixel. The algorithm requires that the area eliminated or highlighted be input in the form of polygon connected by coordinate points. The schematic diagram of TMPF algorithm is shown in Figure 3.
Figure 3. Schematic diagram of the masked texture pre filtering technology

The light is emitted from the viewpoint, and the spherical texture is sampled along the direction of the viewpoint and the screen pixels. Before volume texture sampling, ray pre filtering is performed by texture mask array. In Figure 1.3, the light continues to sample as it passes through a specific area (white area, green line). If the ray is outside a specific area (gray area, red line), the ray will directly terminate the sampling calculation. When the light passes through the sphere, the sampling point is determined according to the step length along the direction of the light, and the texture value is obtained by the spherical trilinear interpolation calculation of the sampling point. The texture value is superposed according to the weight, so as to calculate the final pixel value of the point on the screen.

The whole process of TMPF algorithm is as follows:

1. Visual data construction

The location data of irregular area is counted and abstracted into area coordinate array. Read the position data of the irregular area. Here, it is assumed that the position data of the irregular area is a set of coordinate points, and the points in the set of coordinate points are connected in order to form a closed area. Taking the minimum longitude of the regional position data points as the starting point of the regional coordinate array, the maximum longitude as the ending point, and the sampling precision as the interval of the array, the regional coordinate array is created.

![Area coordinate array](image)

Suppose the longitude range of the irregular area is from 110.3° to 125.6°, the accuracy is 0.1°. The created array is shown in Figure 1.4. Each longitude point corresponds to an array of latitude points. The latitude point array records the latitude information of the point where the ray emitted along longitude intersects the irregular area. All the latitude point arrays corresponding to longitude are statistically constructed, and the area coordinate array is constructed.

2. Mask array construction

The global mask array is generated according to the area coordinate array. The global longitude and latitude space is divided into two-dimensional grids according to a certain accuracy to form a global mask array (the accuracy needs to be consistent with the accuracy of the regional coordinate array). For
example, in Figure 4, the precision is 0.1°. Then the global longitude and latitude are divided by 0.1°. The size of the global mask array is 3600 * 1800. First, set the mask of all elements in the array to 0. According to the algorithm of judging the relationship between the midpoint and the closed area position in Section 1.1, in the latitude point array corresponding to the longitude point, taking two adjacent latitude points as a group, it can be inferred that the region between two latitude points in the same group at that longitude is in the irregular region. According to this conclusion, we traverse the area coordinate array and set the mask of all coordinate points in the irregular area to 1. The global mask array is shown in Figure 5.

| longitude | latitude |
|-----------|----------|
| -180°     | 180°     |
| -90°      | 90°      |
| 113.2°    | A        |
|           | B        |
|           | C        |
|           | D        |

Figure 5. global mask array diagram

An irregular area is drawn in the graph according to its longitude and latitude coordinates. The latitude value of the intersection of each longitude line and the irregular area has been generated in the area coordinate array. For example, at longitude 113.2°, the intersections with the irregular area are A, B, C and D. The latitude values of these four points are stored in the area coordinate array. Take points A and B as a group, and points C and D as a group. It is known that at longitude 113.2°, the regions between points A and B, and between points C and D are all in the irregular region. Set the masks of the corresponding regions in the global mask array to 1. Global mask array can be generated by traversing area coordinate array and performing mask setting operation.

3. Pre filtration fusion

The global mask array is transformed into a two-dimensional texture input shader to participate in rendering. The global mask array is passed into the shader as a two-dimensional texture. When drawing pixels, the mask value of the mask texture is queried according to the pixel coordinates. If the mask is 1, it means that the pixel is in the irregular area, and the corresponding culling or highlighting operation is carried out. If the mask is 0, the normal drawing operation is performed. In order to improve the computational efficiency and reduce the computational overhead, the preprocessing of irregular regions and the generation of mask texture are completed before the rendering process. In the real rendering process, we only need to increase the operation of texture query, because it is implemented by GPU hardware processing unit, the computational complexity is not large. In addition, if the accumulated pixel transparency reaches 1, the ray sampling will be terminated in advance. The algorithm will also determine whether the pixels are in the cone region, and the points outside the cone region will be removed to improve the computational efficiency.

3. Experiment

Experimental environment: Graphics card: NVIDIA GTX750, CPU: INTEL CORE i7 4770K, operating system: Windows7, Memory: 8GB, 3D rendering engine: OpenSceneGraph (OSG) 3.0.1, Development tool: Visual Studio 2010.

Experiments are divided into two categories: removing irregular areas and highlighting irregular areas.

(1) Experiment of eliminating irregular area

The experiment uses global ocean salinity data. The global ocean salinity in July 2010 released by China Meteorological Network was selected. The longitude range is from -175° to 175°, The latitude
range is from \(-75^\circ\) to \(75^\circ\). The depth ranges from 0 m to 3500 m, with a total of 35 layers. The grid scale of data is \(720 \times 345 \times 35\).

The rendering effect of ray casting volume without culling is shown in Figure 6.

![Figure 6. global ocean salinity field](image)

![Figure 7. rendering after removing land](image)

As can be seen from the above figure, the global volume rendering effect based on GPU is basically as expected, but the land area is filled with blue, which is not conducive to observation. This is because the global ocean salinity field data is filled with default values in the land area, and the default values are also drawn in the drawing process, which is not in line with the actual situation that salinity is only valid for the ocean. Therefore, the display effect is affected. For this reason, we extract the global coastline, and construct the polygon set of constraint region by outputting the contour line from GIS, and generate the texture mask, and remove the land part by pre filtering. The final implementation effect is shown in Figure 7.

As can be seen from the above figure, the land part is removed, and the rendering effect is clearer than before. From the junction of the mainland and the ocean, we can see the hierarchical changes of the data, and the display is more intuitive. Therefore, the effect of elimination basically meets the design expectation.

(2) Highlight irregular area experiment

The experiment uses global wind data. The global wind data of China Meteorological Network in April 2012 were selected. The longitude range is from \(-180^\circ\) to \(179^\circ\), the latitude range is from \(-90^\circ\) to \(90^\circ\), the statistical height of the wind is from 10 meters to 5000 meters, a total of 30 stories. The grid scale of data is \(360 \times 181 \times 30\).

The effect of ray casting volume rendering using global wind data is shown in Figure 8.

![Figure 8. global wind visualization](image)

As can be seen from the above figure, the global wind display effect has reached the expected level. However, if users want to observe and analyze the wind intensity in China separately, they need to narrow the display range. Therefore, we use GIS to extract the polygon set of China's regional contour boundary line, generate the texture mask prefilter array, and then highlight the wind data in China. The effect is shown in Figure 9 and Figure 10.
As can be seen from figure 3.4 and figure 3.5, the wind data in China are highlighted separately, and the display effect is good. Through texture mask pre-filtering technology, irregular areas can be highlighted separately, which creates conditions for further observation and analysis of specific areas that users focus on.

The rendering frame rate statistics of the whole experiment are shown in table 1.

| Experiment name                                           | Render frame rate (frames / sec) |
|-----------------------------------------------------------|----------------------------------|
| Global ocean salinity field (before elimination, Figure 3.1) | 36                               |
| Global ocean salinity field (after elimination, Figure 3.2) | 39                               |
| Global wind data (before highlighting, figure 3.3)         | 42                               |
| Global wind data (after highlighting, figure 3.4)          | 53                               |

It can be seen from the above table that the texture mask pre-filtering technology has little effect on the rendering efficiency, and even slightly improves the rendering frame rate. Overall, the rendering frame rate reaches fluency level (no less than 24 frames per second). After analysis, we think that there are two reasons: (1) highlight and cull operations will reduce the number of pixels to be rendered, so the number of ray casting sampling calculation will be reduced, which improves the rendering speed. (2) Most of the calculation of texture mask pre-filtering technology is completed before rendering. There is only one linear interpolation operation in the rendering process, which has little impact on the overall rendering performance. On the whole, we achieve the high efficiency of the algorithm through the hardware support provided by GPU.

4. Conclusion

In this paper, a texture mask pre-filtering technique is proposed to solve the problem of eliminating or highlighting specific areas in the process of ray casting volume rendering. The next research content: 1. 3D texture mask. At present, the texture mask in this paper only supports two-dimensional texture. In the future, it can be extended to more general three-dimensional texture to achieve more general and complex pre-filtering; 2. Performance optimization of global volume rendering; 3. Dynamic scheduling of large-scale display data. Large scale data can not be read into memory completely at one time, so it needs to be partitioned for real-time scheduling according to the drawing requirements. We hope that texture mask pre-filtering technology can accelerate the practical application of volume rendering in engineering.
References

[1] M. Levoy. Display of surfaces from volume data. IEEE Transactions on Visualization and Computer Graphics. 8(3), 1988:29-37.

[2] ZHANG Y. S., CHEN F. M. Volume Rendering of 3-D Medical Volumetric Data[J]. Computer Engineering and Applications, 2002(08):18-19+122.

[3] Hoppe H. Progressive meshes[C]//Proceedings of the 23rd annual conference on Computer graphics and interactive techniques. ACM, 1996: 99-108.

[4] MEI H. H., CHEN H. D., ZHAO X., LIU H. N., ZHU B., CHEN W. Visualization System of 3D Global Scale Meteorological Data[J]. Journal of Software

[5] Intelligent Information Technology Application Association. Proceedings of the 2011 International Conference on Future Computer Science and Application (FCSA 2011 V1) [C].Intelligent Information Technology Application Association,2011:4.

CHEN J., et al. "GPU-based Spherical Shell Volume Rendering Algorithm for the Typhoon Visualization". Proceedings of the 2011 International Conference on Future Computer Science and Application（FCSA 2011 V1）.Ed.. International Industrial Electronic Center, 2011, 500-503.

[6] Li W, Mueller K, Kaufman A. Empty space skipping and occlusion clipping for texture-based volume rendering [C]//Proc of the 14th IEEE Visualization (VIS’03), 2003:317-324.

[7] Kruger J, Westermann R. Acceleration techniques for GPU-based volume rendering [C]//Proc of the 14th IEEE Visualization(VIS’03),2003;232-239.

[8] Markus H, Patric L, Christof R S.et al. Advanced illumination techniques for GPU volume ray casting [C] // Proc of ACM SIGGRAPH ASIA 2008 Courses, 2008:136-141.

[9] Levoy M. Efficient ray tracing of volume data [J]. ACM Transactions on Graphics,1990,9(3) ;245-261.

[10] Li Bao-quan, Gordon J, Feng Dong, et al. Octree rasterizer: accelerating high-quality out-of-core GPU volume rendering [J]. IEEE Transactions on Visualization and Computer Graphics,2013,19(10):1732-1745.

[11] Westermann R, Ertl T. Efficiently using graphics hardware in volume rendering applications [C] //Computer Graphics Proceedings, Annual Conference Series, ACM SIGGRAPH, New York, 1998:169-178.

[12] Meissner M, Hoffmann U, Strasser W. Enabling classification and shading for 3D texture mapping based volume rendering using OpenGL and extensions [C] //Proceedings of IEEE Visualization. San Francisco. 1999 110-119.

[13] Roettger S, Guthe S, Weiskopf D, et al. Smart hardware-accelerated volume rendering. Proceedings of Joint Eurographics/IEEE TCVG Visualization Symposium, Grenoble. 2003, 231-238.

[14] LIANG C. Z. GPU-based Ray-casting Volume Rendering[D]. XIDIAN UNIVERSITY,2009.