

Study on 3D Visualization of Electromagnetic Signal Strength

Wang Hongming^{1,2,*} and Dong Jun²

¹The State Key Laboratory of Complex Electromagnetic Environment Effects on Electronics and Information System (CEMEE), Luoyang, 471003, China; ²National Key Laboratory of Electromagnetic Environment, China Research Institute of Radiowave Propagation, Qingdao, 266107, China

Abstract: Based on the theory of Visualization in Scientific Computing, we use Marching Cubes isosurface algorithm, Multi-Isosurfaces algorithm and ray casting algorithms based on GPU to achieve the 3D visualization of electromagnetic signal strength. It wouldn't make much sense for user to display the distribution of electromagnetic environment in isolation, This paper, the distribution of electromagnetic signal strength is integrated in the battlefield scene, we achieve real-time interactive of the battlefield electromagnetic distribution using VC++6.0 language and 3D OpenGL tool, enhance the convenience and visual effects of operation. The result shows practical application value.

Keywords: Electromagnetic signal strength, marching cubes, multi-isosurfaces, ray casting, visualization.

1. INTRODUCTION

With the wide development and application of high and new technology, electronic warfare has become the main form of modern high-tech warfare. Regardless of battlefield communications, battlefield combat command or direct confrontation, electromagnetic waves have played an irreplaceable role. However, the current measurement of the electromagnetic signal strength mostly rely on the specialized receive or detection equipment, it greatly increases the difficulty of analysis because the electromagnetic field itself is not visible. Therefore, the study of spatial electromagnetic simulation and visualization has become the current hotspot [1-3], it also has great significance for modern electronic warfare.

The purposes of electromagnetic field simulation and visualization is to help users to assist decision and analysis, which requires the visualization of the electromagnetic field from two-dimensional, three-dimensional and so on many dimensions. The electromagnetic field simulation and visualization can help users to grasp the trend of the spatial distribution of electromagnetic field, monitor the direction and strength of electromagnetic propagation. At present, many experts at home and abroad do a large number of researches and obtain certain achievements in order to realize the efficient display of electromagnetic field [4, 5].

The current three-dimensional visualization research has focused on visualization in scientific computing, this paper, we use Marching Cubes isosurface algorithm, Multi-Isosurfaces algorithm and ray casting algorithms based on GPU to achieve the 3D visualization of electromagnetic signal strength. Because it is lack of performance ability to display the distribution of the electromagnetic environment isolately, the computing and visualization systems of

electromagnetic environment must regard geographic environment as a background. In order to give the user an immersive experience, we must add the 3D visualization of electromagnetic field to the geographical environment. In this paper, we achieve real-time interactive of the battlefield electromagnetic distribution using VC++6.0 language and 3D Open GL tool, enhance the convenience and visual effects of operation.

2. MARCHING CUBES ISOSURFACE EXTRACTION METHOD

2.1. Realization Ideas

Three-dimensional data can be regarded as a set of sampling points which have a certain physical properties related to the spatial region. The value of non sampling point can be calculated by the interpolation points near the sampling data at a regional scale. In the space region with a collection of same data value point will be composed of one or more curved surface, called the isosurface. Marching Cubes algorithm [6, 7] provides a simple and effective method for extracting the isosurface, each vertex of voxels were labeled and categorized according to different combination of vertex state, equivalence point of each edge of the cube is extracted by linear interpolation, and then use the three angle fitting a series of equivalent surface. MC algorithm using triangular patches as the basic elements of the geometric primitives, the triangular facets can be accelerated rendering using current graphics card, the MC algorithm is simple in principle, easy to realize itself. So it has been widely used, is considered to be one of the most popular isosurface extraction algorithm.

The basic idea of Marching Cubes algorithm is: processing voxel one by one in the field, classify those voxels which have intersection with isosurface, the intersection of voxels is connected to one or more triangular mesh by a certain generate. As an approximation representation, then use interpolation to calculate the vertex of each triangle mesh normal

vector, and finally using common graphics software packages or hardware rendering the isosurface.

2.2. Implementation Steps

The implementation steps of MC algorithm are as follows [8, 9]:

1. Read the three-dimensional discrete data fields according to hierarchical rule.
2. From the first layer slice, we process the voxel one by one from bottom to up, each voxel in the eight corner taken from the adjacent layers. The eight corner can compose a cube. Cube construction diagram is shown in Fig. (1).

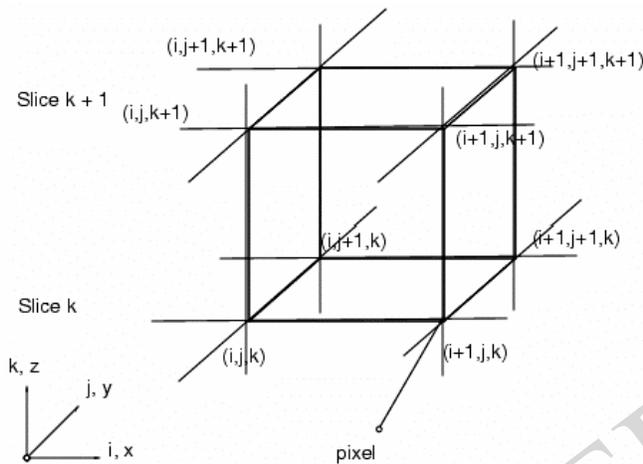


Fig. (1). Cube construction diagram.

3. According to the comparison result between the corner points of each voxel and ZValue given nominal value equivalent, we can build the voxel index table. When the existence of different points marked voxels, the description of the voxel intersect with the isosurface, called active voxels. The intersection of the active surface of the voxel located at the equivalent of the edge of the voxel, marks of the two opposite end edges of a certain, known as the active edges.
4. According to the index table, we can receive the voxel which have an intersection with the boundary. According to 15 kinds of split model, offline establish triangulation index table, when the algorithm is executed in ac-

cordance with the specific state voxel positioning index table, the way to get active triangulation edge. 15 kinds of partition models is shown in Fig. (2).

5. Depending on the location of the edge of space, by interpolation to calculate the coordinates and equivalent personally edge points:
6. Determine the normal vector of the triangular faces at each vertex. there are two methods, one is linear interpolation, another choice is the midpoint. Using the midpoint of the selected part of the surface can be more flat, the error is below half the side length of the cube, so that the image and the reconstructed image obtained by linear interpolation, and there is no significant visual difference.
7. Draw isosurface according to the coordinates and normal vectors at the vertices of triangular facets.

3. MULTILAYER ISOSURFACE EXTRACTION METHOD

The traditional isosurface technology only can express single contour information which will loss a large number of internal details of spatial data in the field [10]. It is not conducive to visualization for spatial data such as electromagnetic environment information. To compensate for this deficiency, we can extract the different characteristics of attribute values using multi-isosurface extraction technology, enhance the ability for representing the spatial information.

We can draw a semitransparent multi-isosurface by using the fusion of object level sort [1, 5], which is considered as the most direct method. The isosurface will blend into multi-isosurface according to the order sorted from back to front, until get the final result. In order to get the accurate results, we must be strict to each pixel according to the order sorted from back to front, we also must separate the segmentation primitives each other at the intersection and solve the occlusion problem of loop between primitives. Despite we can render the results accurately, we spent a large amount of processing time. This method is difficult to render efficiently in a large-scale model. In order to sort multi-isosurface quickly, we can sort the multi-isosurface from inside to outside according to the extraction threshold. To distinguish between the front and back surfaces of the sheet isosurface, we use the angle between the direction of the viewpoint. The angle is less than 90°, it is the front surface of the sheet, The angle is more than 90°, it is the back surface of the sheet.

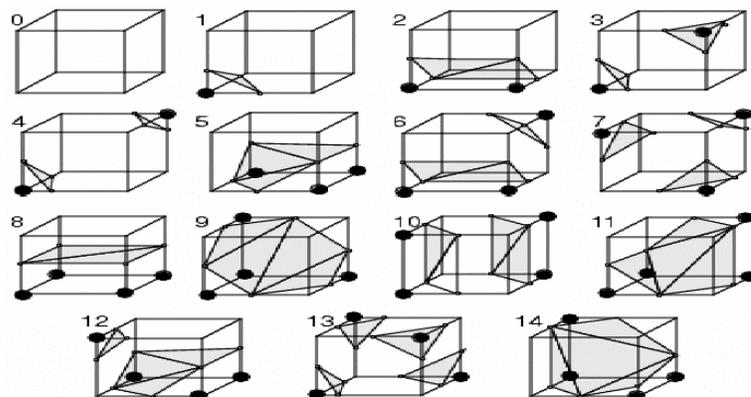


Fig. (2). 15 kinds of partition models.

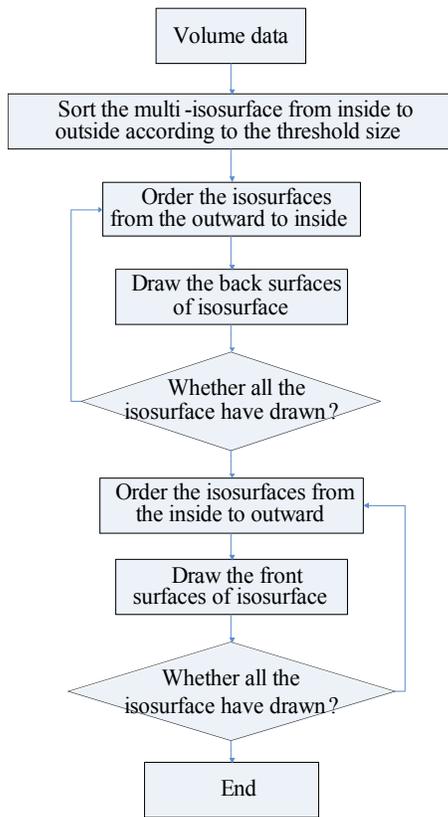


Fig. (3). The specific fusion of multi-isosurface rendering process.

The specific fusion of multi-isosurface rendering process is shown in Fig. (3).

4. RAY CASTING METHOD BASED ON GPU

4.1. Realization Ideas

The volume rendering technology which convert a three dimensional discrete sample values into a two-dimensional image on the screen by all means, is as far as possible to accurately reproduce the original 3D data field, allowing users to observe and discover the distribution and change of data value in 3D data field at any position. The principle of ray casting volume rendering algorithm is as follows [11, 12]. Send the light from the image plane of each pixel along the line of sight into the voxel space, select the sampling points in the light, calculate the resampling point value using the reconstruction function selected by interpolation, determine its opacity and color values for each sample using the transfer function. At last, synthesis the final image for each resampling point. In the ray casting method. It will limit the rendering speed because we must operate each voxel which will waste a lot of time, so it can not achieve the purpose of real-time rendering. The ray casting volume rendering algorithm based on GPU can make full use of software and hardware to achieve real-time rendering, solves the contradiction between rendering speed and rendering quality in real-time rendering image.

4.2. Implementation Steps

The implementation steps of ray casting volume rendering algorithm based on GPU are as follows [13]:

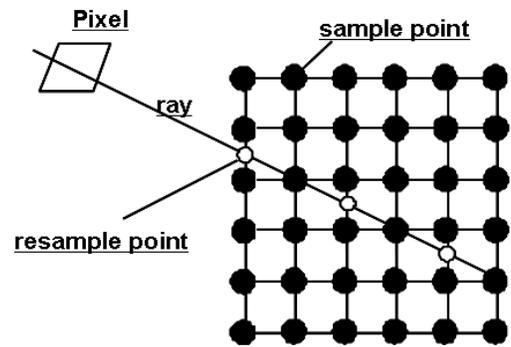


Fig. (4). The set diagram of resampling point in ray casting.

1. Data conversion, read the voxel values, then convert the voxel values into the 3D texture by the corresponding transfer function. This process is completed in CPU.
2. Coordinates conversion, we should convert the view coordinate system which is the coordinate system for the sample points into the world space coordinate system which is the coordinate system for the volume data.
3. Generate the end point of light, reach behind the rendering of bounding box, get the end point coordinates of light, for the regular volume data, we should normalize the three-dimensional size of bounding box to 0-1.
4. Generate the direction of light, in the front of the rendering bounding box, get the start point coordinates of light,
5. Calculate the brightness values, resample the point along the direction of ray casting light and calculate the brightness values. start from the current view position, send a light to every pixel on the screen through the data field, meanwhile, resample the point according to the same interval along the direction of ray casting light, then we obtain a series of resampling points. For resampling point PR in the data field, we find it in the voxel, execute the three linear interpolation to obtain the opacity and color of resampling point. The set diagram of resampling point in ray casting is shown in Fig. (4).
6. Accomplish ray casting, calculate the range of screen pixels according to the reproducing function, then provide the brightness of each pixel which is obtained by the accumulation of data points. This method for generating light direction, regardless perspective projection or parallel projection, can be obtained by rendering the bounding box from front to back.

The diagram of ray casting algorithms based on GPU is shown in Fig. (5).

For large amounts of data in real-time interactive, many scholars rely on octree structure and graphics hardware acceleration, the volume data is divided into different blocks, achieve rendering volume data for large amounts of data accurately. In this paper, we use volume data of entire earth and space as a example. The drawing process is as follows [14, 15]:

1. Direct volume rendering bounding box is determined, while roaming the earth space environment, because only one part of earth that can be seen at one time, so we

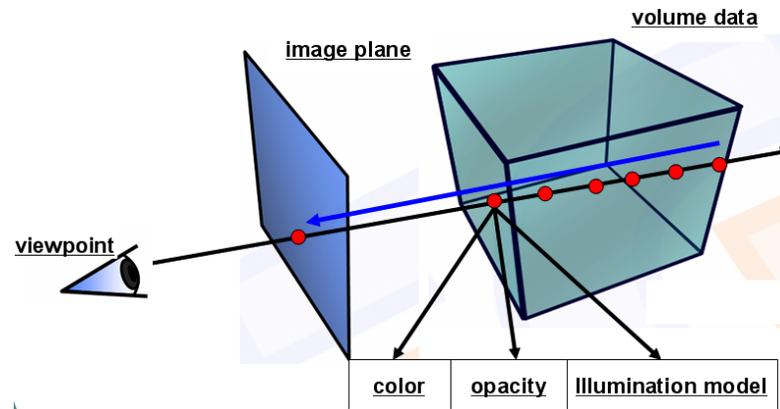


Fig. (5). The diagram of ray casting algorithms based on GPU.

do not need to draw the invisible part. In this paper, we use view frustum culling method to determine the specific space which circle the Earth grid visible.

2. Manage the bounding box vertex using VBO (Vertex Buffer Object), using VBO to manage these vertices can be implemented on the server side cache storage, so that the data processing need not be directly taken out from the server to speed up data transfer.
3. Generate the ray-casting light, from the view point of departure, a vector connecting the viewpoint and the pixels on the screen as the forward direction of the light, the light enters the calculation of the starting point of the bounding box (set A (x_1, y_1, z_1)), the light leaving the bounding box is calculated the end point coordinates (set B (x_2, y_2, z_2)), according to the sampling step and $|AB|$, the number of samples to calculate the radiation in the AB. OVER operator accumulation cycle with color and opacity.
4. Update the bounding box dynamically, in order to achieve the purpose of geospatial data in real-time dynamic display, establish automatic paging and storage mechanism based on the bounding box, Which is a very effective method. In the dynamic rendering process, with the movement of the viewpoint, the need to constantly update the bounding box of the data block. In order to speed up the rendering speed, this paper established a buffer zone around the station of two data pages, and to achieve the exchange of data content between two buffers through multi-threading technology.

5. EXPERIMENTAL RESULTS AND ANALYSIS

We use the VC ++ 6.0 language and OpenGL 3D tools to achieve the algorithm described in this article, the use of the machine's processors for AMD 2.7GHz, 4GB RAM, GeForce 6800 graphics card. Windows 7 operating system, programming environment is Microsoft Visual Studio 2008. On the test data, the basic parameters of propagation prediction are set as follows: operating frequency: 1000MHz, the percentage of time: 50%, the percentage of location: 50%, transmitting station Longitude: 118°, transmitting station latitude: 25°, station antenna height: 10m, antenna height of the transmitting station 20m to 340m.

MC isosurfaces extraction algorithm of electromagnetic signal strength distribution drawing effect is shown in Fig. (6).

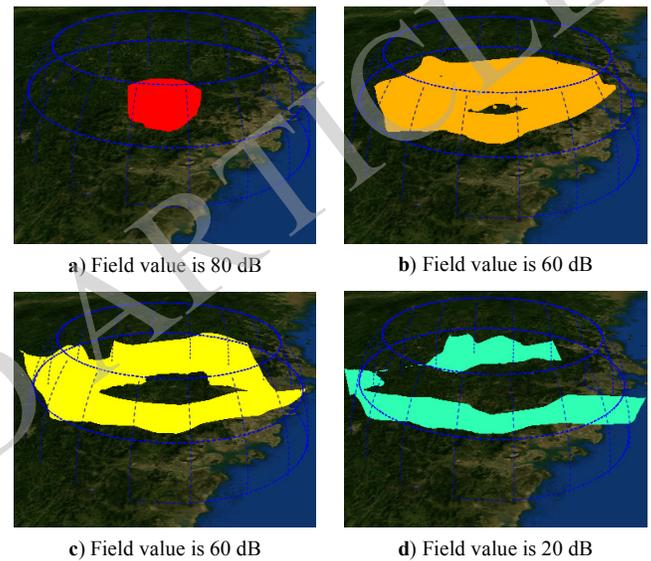


Fig. (6). MC isosurfaces extraction algorithm of electromagnetic signal strength distribution drawing effect.

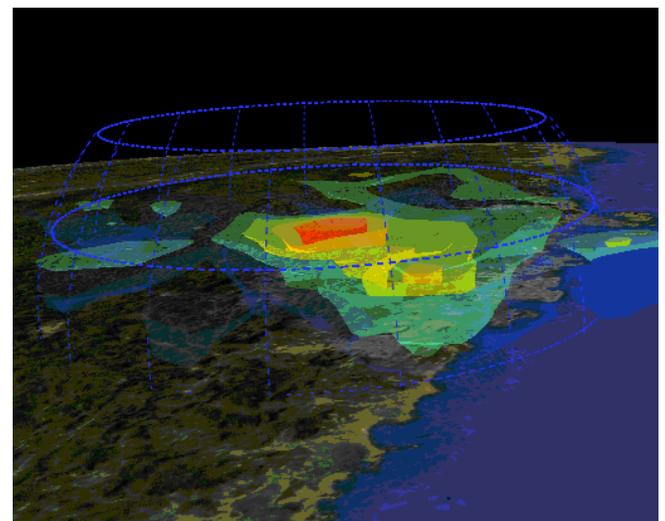


Fig. (7). Multi-isosurfaces rendering of the electromagnetic signal strength.

Multi-Isosurfaces Rendering of The Electromagnetic Signal Strength is shown in Fig. (7).

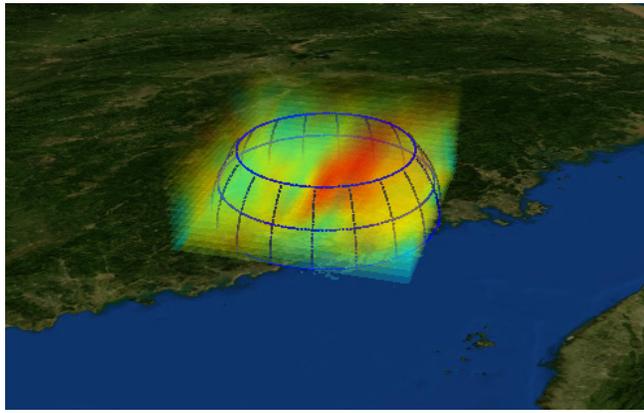


Fig. (8). Electromagnetic frequency device with volume rendering renderings situation.

In order to verify and test the proposed rendering algorithm of ray casting based on the earth sphere shell space grid model. The experimental data were three-dimensional electromagnetic momentum body data of 400 km * 200 km * 800m, data formats are 32 floating-point number. Electromagnetic frequency device with volume rendering renderings situation is shown in Fig. (8).

6. CONCLUSION

With the development of visualization in scientific computing and the virtual reality technology, the visualization technology is gradually penetrating to each subject. It has a broad application prospect, and the visualization of electromagnetic signal strength has become an important application field.

In this paper, we use Marching Cubes isosurface algorithm, Multi-Isosurfaces algorithm and ray casting algorithms based on GPU to achieve the 3D visualization of electromagnetic signal strength, meanwhile, we add the 3D visualization of electromagnetic field to the battlefield scene, achieve real-time interactive of the battlefield electromagnetic distribution. At present, many domestic and foreign scholars put forward many methods of large quantities of data volume rendering, the next step, we should integrate these methods and the real battlefield scene, so that we can achieve better results for the visualization of electromagnetic signal strength.

CONFLICT OF INTEREST

The authors confirm that this article content has no conflict of interest.

ACKNOWLEDGEMENTS

This work was financially supported by the State Key Laboratory of Complex Electromagnetic Environment Effects on Electronics and Information System Director Fund (CEMEE2014Z0103B).

REFERENCES

- [1] C. Yang, "Research on electromagnetic environment modeling and rendering three-dimensional method of virtual battlefield", D. thesis, National University of Defense Technology, Changsha, ON, China, 2010.
- [2] A. Ohshita, M. Okuhara, C. Matsuya, K. Hata, and K. Iida, "Direct Visualization of electromagnetic microfields by superposition of two kinds of electron holograms", *Microchimica Acta*, vol. 155, no. 1-2, pp. 225-228, 2006.
- [3] F. Li, Z. Zhang, G. Wan, "Three-dimensional electromagnetic situation battlefield visualization technology", *Geomatics Science and Technology*, vol. 29, no. 3, pp. 222-225, 2012.
- [4] W. Zhou, J. Li, and G. Huang, "Sea battlefield electromagnetic information visualization technology based on VTK", *Computer and Digital Engineering*, vol. 40, no. 5, pp. 121-124, 2012.
- [5] H. Hu, C. Yang, L. Wu, "Multilayer electromagnetic environment isosurface fusion", *Computer Engineering and Applications*, vol. 35, no. 5, pp. 206-209, 2011.
- [6] W.E. Lorensen, and H.E. Cline, "Marching cubes: A high resolution 3d surface construction algorithm", *ACM Computer Graphics*, vol. 21, no. 4, pp. 163-169, 1987.
- [7] C. Montani, R. Scateni, and R. Scopigno, "A modified look-up table for implicit disambiguation of Marching Cubes", *The Visual Computer*, vol. 10, no. 6, pp. 353-355, 1994.
- [8] J. Congote, A. Moreno, I. Barandiaran, J. Barandiaran, and O. Ruiz, "Extending Marching Cubes with Adaptive Methods to Obtain More Accurate Iso-surfaces", *Computer Vision, Imaging and Computer Graphics*, vol. 68, no. 6, pp. 35-44, 2010.
- [9] G. M. Nielson, L. Zhang, K. Lee, and A. Huang, "Parameterizing marching cubes isosurfaces with natural neighbor coordinates", *Advances in Geometric Modeling and Processing*, vol. 49, no. 75, pp. 315-328, 2008.
- [10] S. Yang, and T. Wu, "Compressing isosurfaces generated with marching cubes", *The Visual Computer*, vol. 18, no. 1, pp. 54-67, 2002.
- [11] J. Song, O. Gwun, S. Kim, and Y. Kim, "A boundary surface based ray casting using 6-depth buffers", *Computational and Information Science*, vol. 33, no. 14, pp. 1134-1140, 2005.
- [12] E. Gobbetti, F. Marton, and J.A.I. Guitián, "A single-pass GPU ray casting framework for interactive out-of-core rendering of massive volumetric datasets", *The Visual Computer*, vol. 24, no. 7-9, pp. 797-806, 2008.
- [13] K. Na, and M. Jung, "Curved ray-casting for displacement mapping in the GPU", *Advances in Multimedia Modeling*, vol. 49, no. 3, pp. 348-357, 2008.
- [14] C. Lux, and B. Fröhlich, "GPU-based ray casting of multiple multi-resolution volume datasets", *Advances in Visual Computing*, vol. 58, no. 76, pp. 104-116, 2009.
- [15] M. Zhu, M. Guo, L. Wang, and Y. Dai, "A ray casting accelerated method of segmented regular volume data", *Advanced Research on Electronic Commerce, Web Application, and Communication*, vol. 14, no. 4, pp. 7-12, 2011.