Accelerating Footprint Method Based on Comparability of Consecutive Image Slices*

Jianxun Zhang[1], Jiawan Zhang[2], Jizhou Sun[2]
[1]*School of Computer Science&Engineering, Chongqing Institute of Technology, Chongqing 400050, P.R.China*
[2]*School of Electronic & Information Engineering, Tianjin University, Tianjin 300072, P.R.China*
{zjx@cqit.edu.cn, jwzhang@tju.edu.cn,jzsun@tju.edu.cn}

Abstract

Volume rendering has been a key technology in the visualization of data sets from various disciplines. However, fast volume rendering of large-scale data sets is still a challenging field due to the vast memory, bandwidth and computational requirements. Two consecutive image slices of volume data usually have very high comparability. It is designed an accelerating footprint volume-rendering method based on comparability of consecutive image slices. When the contribution of the sampling point to screen pixels color value is computed, only the changed contribution of the next slice sampling point changed data value to screen pixels color value is computed. The influencing screen pixels color value is modified based on the changed contribution value. The experiment expresses that the new footprint volume-rendering method can obviously accelerate speed of volume rendering.

Keywords: visualization in scientific computing; volume rendering; footprint method

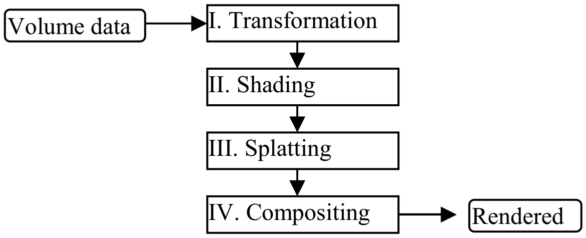
1. Introduction

Volume visualization is a research area that deals with various techniques for extracting meaningful visual information from abstract and complex volume data [1]. It has been successfully applied into several applications including medical visualization, computational fluid dynamics, seismology, and natural phenomena simulation. One of the most frequently applied techniques for visualization is direct-volume rendering, which produces 3D rendered images of high quality directly from volume data.

Generally, direct-volume rendering can be classified into five categories according to the implementation details: ray casting [2][3], shear-warp [4], footprint method [5][6],volume texture mapping and frequency domain method [7]. Footprint method, introduced by Westover,voxels are traversed in either front-to-back or back-to-front order consistent with a given viewpoint. During the traversal, each voxel is classified and shaded by user-supplied opacity and material transfer functions. Then it is projected into the image plane, and its contribution is accumulated to an image buffer using a projected reconstruction kernel, called a footprint.

The pipeline of footprint method is given in Figure 1.This show how the volume (the input data structure) is transformed through a 4-step rendering process to produce the final image (the output data structure).

Fig.1. Data flow diagram of footprint method



In every step rendering process, if rendering process speed was accelerated, total rendering time was reduced. In this paper, It is designed a footprint volume-rendering method in which accelerate 3rd step (splatting) rendering process based on comparability of consecutive image slices.

Over the years, many computer graphics researchers have developed accelerate splatting algorithms. Laur and Hanrahan [8] describe a hierarchical footprint method by using octree to exploit

---

* This work is partially supported by National Natural Science Foundation Project (NSFC, 60373061) and Natural Science Foundation Project (NSFCQ, 8647) of Chongqing,China; Corresponding author: Jianxun Zhang, Tel:+86-023-68667793

coherence of homogeneous region and to enable progressive refinement during rendering.

Crawfis and Max [9] have developed accelerate splatting algorithms using translucent polygons, two-dimensional texture mapping hardware.

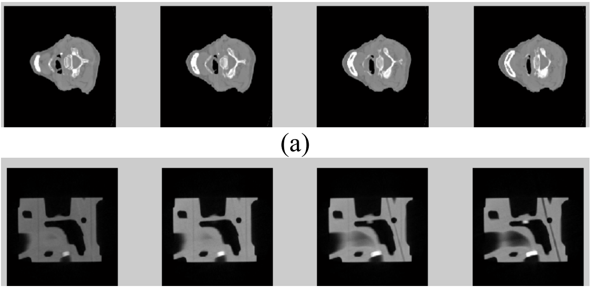
Insung [10] developed an efficient data structure to skip voxels that do not contribution to the rendering image.

Mueller and Yagel [11] combine footprint method with ray casting techniques to accelerate rendering with perspective projection.

Daqing Xue and Roger Crawfis [12] developed a splatting algorithm that utilizes the new capabilities of vertex programs and the OpenGL extensions on a PC equipped with an NVIDIA GeForce4 display card.

2. Comparability of consecutive image slices

Most three-dimensional (3D) volume data are sampled with the 1mm to 3mms distance between consecutive image slices. So two consecutive image slices of 3D volume data usually have very high comparability. In Fig. 2, four consecutive image slices of 3D volume data are showed. The sampling distance x/y/z direction of head CT images is 0.87mm/0.87mm/2mm, respectively. The sampling distance x/y/z direction of engine CT images is 1mm/1mm/1mm, respectively. In theory, the sample distance between consecutive image slices of 3D volume data are smaller, two consecutive image slices of 3D volume data are more comparability.



(a)

(b)

Fig. 2. Four consecutive image slices of 3D volume data: (a) Head CT images ;(b) Engine CT images.

3. The principle of accelerating footprint volume-rendering method

The main advantage of footprint method is the storage-order access to the data. This method, in combination with filter kernel approximations, allows to trade-off image quality for fast algorithm execution. Because the computation of the footprints and of the filter weights is expensive, Westover proposes to use lookup-tables for rotation-invariant Gaussian resampling filters. In practice, simpler 2D convolution filters replace the computationally expensive three-dimensional convolution. Lookup tables for the filter weights avoid excessive arithmetic complexity. Using this approximation, the footprint method reduces to projections of object space slices onto the image plane.

The volume reconstruction is equal to [5]

$$signal_{3D} = \iiint h_v(u-x, v-y, w-z) \cdot \sum \delta(x,y,z)\rho(x,y,z)\,dudvdz \tag{1}$$

where h_v denotes the volume reconstruction kernel, ρ denotes the density function, and ∑δ denotes the comb function.

Evaluating the integral at a point <x, y, z> results in:

$$signal_{3D}(x,y,z) = \sum_{i\in V} h_V(i_x - x, i_y - y, i_z - z)\rho(i) \tag{2}$$

The effect at a point <x, y, z> by a data sample <i> is:

$$effect_i(x,y,z) = \rho(i) h_V(i_x - x, i_y - y, i_z - z) \tag{3}$$

Because the footprint method is an object-order rendering, projecting the voxel onto the view plane at pixel <x, y> is:

$$effect_i(x,y) = \int_{-\infty}^{\infty} h_V(i_x - x, i_y - y, w)\rho(i)\,dw \tag{4}$$

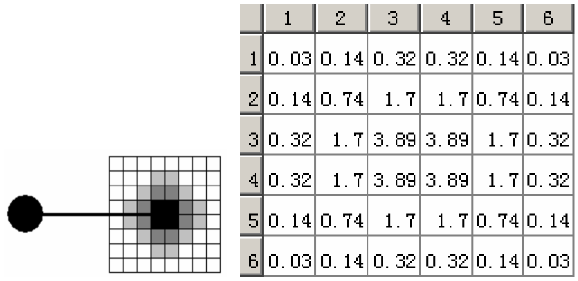
So the footprint [6] function can be defined as follows:

$$footprint(x,y) = \int_{-\infty}^{\infty} h_V(x,y,w)\,dw \tag{5}$$

Since the footprint function can be pre-computed and stored in a table, the speed of projecting the voxel to the view plane will be fast. In Fig. 3, a generally purpose footprint lookup table is showed. In parallel projection, the footprint lookup table is not changed.

$\rho(i)$ denotes the density value in 3D volume data sample <i>. When $\rho(i)$ is changed to $\rho(i) + \Delta\rho(i)$, the contribution of the 3D volume data sample <i> to screen pixels color value is also changed. The changed contribution to screen pixels color value is:

$$\Delta effect_i(x,y) = \Delta\rho(i) \int h_v(i_x - x, i_y, w)\,dw \tag{6}$$



（a） (b)

	1	2	3	4	5	6
1	0.03	0.14	0.32	0.32	0.14	0.03
2	0.14	0.74	1.7	1.7	0.74	0.14
3	0.32	1.7	3.89	3.89	1.7	0.32
4	0.32	1.7	3.89	3.89	1.7	0.32
5	0.14	0.74	1.7	1.7	0.74	0.14
6	0.03	0.14	0.32	0.32	0.14	0.03

Fig. 3. A generally purpose footprint lookup table: (a) the effect of a 3D volume data point; (b) 6X6 footprint lookup table based on Gaussian reconstruction kernel;

The equation (6) is used to accelerating footprint volume-rendering method. The density value of sample <i,j> in two consecutive image slices of 3D volume data L_{k1} and L_{k2} is $\rho(i,j,k1)$ and $\rho(i,j,k2)$,respectively. In parallel projection mode, the contribution of the 3D volume data sample <i, j, k1> to screen pixels color value based equation (6) is:

$$effect_{(i,j,k1)}(x,y)=\rho(i,j,k1)\int h_v(i_x-x,i_y,w)dw \quad (7)$$

The contribution of the 3D volume data sample <i, j, K2> to screen pixels color value based the equation (6) is:

$$effect_{(i,j,k2)}(x,y)=\rho(i,j,k2)\int h_v(i_x-x,i_y,w)dw \quad (8)$$

The different value of two contribution is：

$$\Delta effect(x,y)=(\rho(i,j,k2)-\rho(i,j,k1))\int h_v(i_x-x,i_y,w)dw \quad (9)$$

If the density value of sample <i,j> in two consecutive image slices of 3D volume data L_{k1} and L_{k2} is not changed, the contribution of sample <i,j> in two consecutive image slices of 3D volume data L_{k1} and L_{k2} to screen pixels color value is also not changed. If the density value of sample <i,j> in two consecutive image slices is changed, based the equation (6), the contribution of sample <i,j> in two consecutive image slices of 3D volume data L_{k1} and L_{k2} to screen pixels color value is also changed. The changed contribution is equal to the contribution of sample voxel which density value is $\Delta\rho=(\rho(i,j,k2)-\rho(i,j,k1))$ to screen pixels color value.

Let us assume that the contribution of all sample voxel in image slices L_{k1} of 3D volume data to screen pixels color value has been computed and saved in $eff_{k1}(i,j),1<=i,j<=M$. Where we assume that the footprint volume-rendering method resulting screen image have M rows and M columns. When the contribution $eff_{k2}(i,j),1<=i,j<=M$ of the sample voxel in consecutive next image slices of 3D volume data L_{k2} to screen pixels color value has been computed, if the density value of sample <i,j> in two consecutive image slices L_{k1} and L_{k2} of 3D volume data are equal to, the contribution of sample <i,j> in two consecutive image slices L_{k1} and L_{k2} of 3D volume data to screen pixels color value is not changed. The contribution of all sample voxel in image slices L_{k2} of 3D volume data to screen pixels color value, which was been express $eff_{k2}(i,j)$, is not changed. If the density value of sample <i,j> in the image slices L_{k2} is changed $\Delta\rho=(\rho(i,j,k2)-\rho(i,j,k1))$, only the influencing screen pixels color value is modified. The modified contribution value is given by the following equation $\Delta effect(x,y)=(\rho(i,j,k2)-\rho(i,j,k1))\int h_v(i_x-x,i_y,w)dw\cdot$

If image slice L_{k1} of 3D volume data are first slice, the contribution of all sample voxel in first image slices to screen pixels color value is computed in traditional footprint volume rendering method and saved in $eff_{k1}(i,j),1<=i,j<=M$, Where we assume that the footprint volume-rendering method resulting screen image has M rows and M columns. Assume that the contribution of all sample voxel in last image slices L_{k1} of 3D volume data to screen pixels color value has been computed, saved in $eff_{k1}(i,j),1<=i,j<=M$ and used to blend final image. When the contribution of all sample voxel in next image slices L_{k2} of 3D volume data to screen pixels color value has been computed, firstly compute $\Delta\rho=(\rho(i,j,k2)-\rho(i,j,k1))$, where $\rho(i,j,k2)$ and $\rho(i,j,k1)$ are the density value of sample <i,j> in current image slices L_{k2} and last image slices $L_{k1}$of 3D volume data. If $\Delta\rho$ is equal to 0，all $eff(i,j),1<=i,j<=M$ is not changed. If $\Delta\rho$ is not equal to zero, the influencing screen pixels (decided by footprint table) color value is modified based on the changed contribution value.

4. Experiments and result

All performance measurements have been conducted on a PC equipped with a 2.0GHz Pentium 4 CPU and 512M DRAM. Fig.4 shows the rendering results in parallel projection mode.

Table 1 shows the performance comparison between the traditional footprint volume-rendering method and the accelerating footprint volume-rendering method in this paper. Table 1 express that the new footprint volume-rendering method can obviously accelerate speed of volume rendering and the resulting image has same quality. In theory, two consecutive image slices of volume data have a high comparability, the accelerating footprint volume-rendering method in this paper have a high efficiency.

For example, two consecutive image slices of CT engine volume data have higher comparability, the accelerating footprint volume-rendering method in this paper have higher efficiency. Data in Table 1 explain rationality of our algorithm in this paper.

Table 1. Algorithm performance compare (in parallel projection mode)

Data set name	Resolution	Traditional footprint method rendering time	Accelerating footprint method rendering time	Accelerating ratio
Fuel	64x64x64	0.121s	0.015	8.06
Inner Ear	128 x128x30	0.264s	0.041	6.46
CT Head	256 x 256 x 225	9.252s	1.715s	5.39
CT Engine	256 x 256 x 128	4.382s	0.605s	7.24

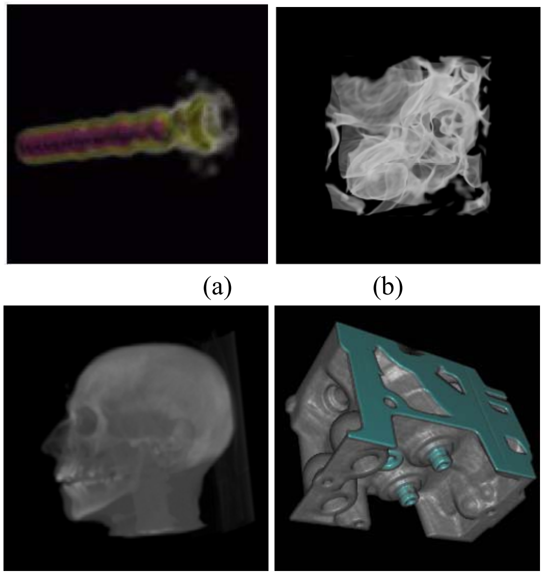

(a) (b)
(c) (d)

Fig. 4. Rendering results: (a) Fuel; (b) Inner Ear; (c) CT Head; (d) CT Engine.

5. Conclusions

Volume rendering has been a key technology in the visualization of data sets from various disciplines. However, real-time volume rendering of large-scale data sets is still a challenging field due to the vast memory, bandwidth and computational requirements. In this paper, to fast visualize the medium or large-scale data set, we first analyze attribute of two consecutive image slices of volume data. Two consecutive image slices of volume data usually have very high comparability. It is designed an accelerating footprint volume-rendering method based on comparability of consecutive image slices. When the contribution of the sampling point to screen pixels color value is computed, only the changed contribution of the next slice sampling point changed data value to screen pixels color value is computed. The influencing screen pixels color value is modified based on the changed contribution value. The experiment expresses that the new footprint volume-rendering method can obviously accelerate speed of volume rendering and the resulting image has the same quality.

References

[1] A. Kaufman (Ed.), Volume Visualization, IEEE Computer Soc., Los Alamitos, CA, 1991.

[2] J T Kajiya, B P Von Herzen, "Ray Tracing Volume Densities", Computer Graphics, Vol.18 (3), pp.165-174, 1984.

[3] M Levoy, "Efficient Ray Tracing of Volume Data", ACM Transactions on Graphics , Vol.9(3), pp.245-261, 1990.

[4] P. Lacroute , M. Levoy, "Fast Volume Rendering Using a Shear-Warp Factorization of the Viewing Transform", Computer Graphics, Vol.28(3), pp.451-459, 1994.

[5] L. A. Westover, "Interactive Volume Rendering", Proceedings of the Chapel Hill on Workshop Volume Visualization, University of North Carolina, Chapel Hill, NC, pp. 9-16,1989.

[6] L. A. Westover, "Footprint Evaluation for Volume Rendering", Computer Graphics, Vol.24 (4), pp.367-376, 1990.

[7] T. Totsuka, M. Levoy, "Frequency Domain Volume Rendering", SIGGRAPH '93, pp.271-278, California, 1993.

[8] D. Laur, P. Hanrahan, "Hierarchical Footprint method: A Progressive Refinement Algorithm for Volume Rendering", Computer Graphics, Vol.25 (4), pp. 285-288, 1991.

[9] R. A. Crawfis, N. Max, "Texture Splats for 3D Scalar and Vector Field Visualization", Proceedings of IEEE Conference on Visualization 1993, pp. 261-266, 1993.

[10] Insung Ihm, Rae Kyoung Lee, "On Enhancing the Speed of Splating with Indexing", Proceedings of the 6th IEEE Visualization Conference, pp. 69-76, 1995

[11] K. Mueller, P. Yagel, "Fast Perspective Volume Rendering with Footprint method by Using a Ray-driven Approach", Proceedings of IEEE Conference on Visualization 1996, pp. 65-72, 1996.

[12] Daqing Xue and Roger Crawfis,"Efficient Splatting Using Modern Graphics Hardware", journal of graphics tools, Vol. 8(3): pp.1—21