TEST Engineering & Management

July – August 2020
ISSN: 0193-4120 Page No. 535 - 541

Volume Rendering for Weather Data using Ray-casting Algorithm on Web

Palak Chatwani[*1], Purvi Ramanuj[2], Shivani Shah[3]
*1Information Technology Department, LD College of Engineering, Ahmedabad, Gujarat, India
palakchatwani46@gmail.com
2Information Technology Department, LD College of Engineering, Ahmedabad, Gujarat, India
purviramanuj@yahoo.com
3DWD/EPSA, Space Applications Centre, Indian Space Research Organization, Ahmedabad, Gujarat, India
shivanishah@sac.isro.gov.in

Article Info
Volume 83
Page Number: 535 - 541
Publication Issue:
July - August 2020

Article History
Article Received*: 06 June 2020*
Revised*: 29 June 2020*
Accepted*: 14 July 2020*
Publication*: 25 July 2020*

Abstract

Volume rendering is the key to visualize data in 3D. Ray-casting algorithm is a direct volume rendering technique that makes volume rendering possible. Imaging satellites are used to collect weather data (temperature, humidity, wind, rain) present in NetCDF (Network Common Data Form). Volume rendering of weather data cannot be achieved by NetCDF data format. By converting the data from NetCDF format to NRRD (Nearly Raw Raster Data), 3D volume rendering of weather data has been performed using modified ray-casting algorithm in this paper. Weather data is rendered in 3D using Ray-casting based WebGL (Web Graphics Library) tool which maps textures with images of different volumetric data while changing the range of dimensions as well as colormaps.

Keywords: Volume rendering, Ray-casting, Weather data, WebGL

I. INTRODUCTION

Volume rendering is the system of showing two-dimensional projections of three-dimensional data, i.e. volumetric information. In scientific 3D visualization, volume rendering is widely used. The essential standards of Volume rendering were first grown by Marc Levoy [1] and further conveyed by Robert A. Drebin [2] and numerous different researchers in the representation field. Levoy has given the point by point review and execution of Volume rendering in different fields [3][4]. Structured volume can be represented as a simple 3D array of scalar values that implicitly defines a grid. Eight neighboring values in the volume define the basic volume element, a voxel [5].

A volume described as a group of voxels can be visualized by volume rendering. Volume rendering works on volume rendering pipeline. Here, Fig. 1 represents the diagram of the volume rendering pipeline. As volume rendering is widely used in medical imaging, so mostly volumetric information can be CT, MRI, ODT or, any 3D data. As the weather data needs to be visualized here,volumetric information will be used as it as.A volume related information group is normally situated of V of the examples (x,y,z,v), likewise named voxels, speaking to the worth v of some attribute of the information, on three-dimensional area (x,y,z).

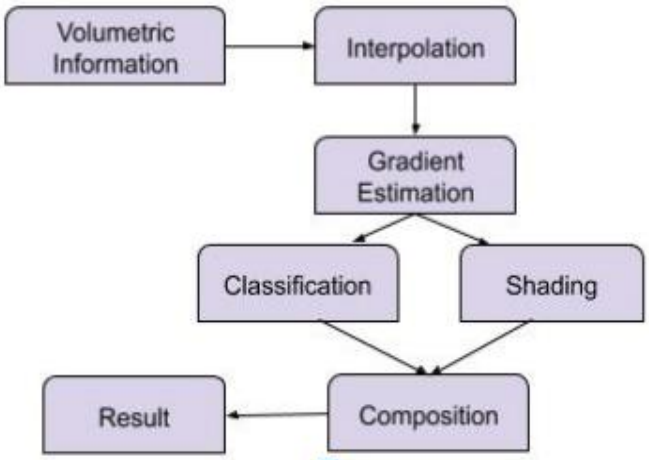

Figure 1: Volume rendering pipeline

Next phase, the interpolation can be defined as the generation of new values which are missing in between based on existing values. Linear, bilinear and trilinear interpolation can take place as per the dimensions. Gradient estimation is done for assessing the measure of light redirected from volumetric exteriors towards the eye. After gradient estimation, classification and shading

July – August 2020
ISSN: 0193-4120 Page No. 535 - 541

are done parallelly. Classification can be done using a transfer function where information qualities are mapped to shading and darkness, with the outcome that elements can be stressed or covered up. Shading is the procedure of figuring an illumination prototype where the outcome is the shading for a polygon or for distinct pixels. Compositing is the final stage to reach the image by binding all steps of rendering. Compositing can be done using back-to-front or front-to-back traversal [6].

II. RELATED WORK

Volume rendering techniques can be categorized into 2 categories. Direct Volume Rendering (DVR) and Indirect Volume Rendering (IVR). DVR primarily offers flexibility; it can be used to obtain an initial overall view of the data, and, by changing transfer functions (which are directly analogous to color maps), Data's particular features can also be focused incrementally. Direct volume-rendering algorithms consist of three major components: sampling, classification, and compositing. Sampling deals with selecting the piecewise steps taken through the volume; classification is the process of computing a color and opacity for each step using the volume-rendering integral; and compositing blends these classified steps together to form an image as a final step [7].

DVR can be categorized as Image order and Object order techniques. With the image order approach, pixels are partitioned among processors. The partitioning creates groups of consecutive pixels, also known as tiles. Each processor begins by loading the cells that contribute to its tile. Then each processor generates the portion of the image corresponding to its tile, by operating on the cells that it loaded. The portion of the image produced by each processor is also known as a sub-image. The sub-images from all the processors are then collected onto one processor to produce the final image [8]. Ray casting is an example of image order technique. With the object order approach, data is divided into blocks among processors. Each processor volume renders its own cells independently of the other processors. Then the contributions from all the processors are composited together to produce a final image, e.g. splatting [9]. Splatting technique is compared to tossing a snowball (voxel) at a glass dish. The snow commitment at the focal point of effect will be more and the commitment will drop away more far from the focal point of effect. Shear warpping shears the image in 2D and warps the intermediate image in the end [11]. Texture Based slicing generates immediate slicing while rendering. Indirect volume rendering extracts certain areas of interest from the entire volume and transforms them into polygonal models e.g. marching cubes algorithm, which approximates a polygonal model from a voxel-based dataset [10].

Ray-casting is based on surface interaction of rays with objects. Ray casting algorithm collects the colour and opacity details along its path through the volume. The rays start from the viewpoint and traverses almost all voxels in the dataset at the specified inclination. Rays which originate from the viewpoint are sampled at equidistance within the volume data. These samples are interpolated in trilinear way and all the interpolated values of individual rays are composited correspond to the image plane pixels that is crossed by the correspondingrays [12][18]. Fig.2 shows the working of ray-casting.

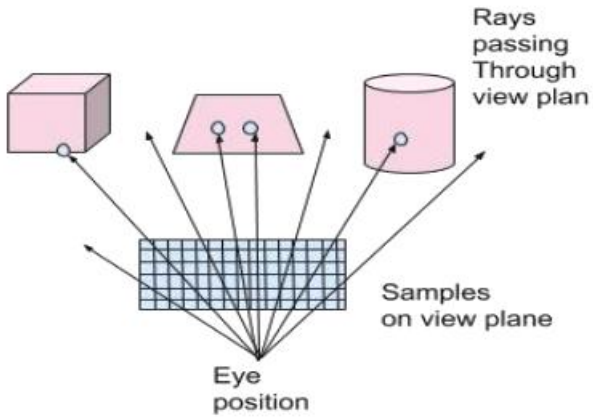

Figure 2: Ray-casting

For volume rendering, ray casting follows these steps.

(1) Finding the visible surface, for that consider $V0$ =View Point= $VP = (VPx, VPy, VPz)$ And $P0$= pixel = $(x1, y1, 0)$.

Now intersect this ray with your scene and set a colour of the intersected surface to the pixel.

$$Fp = (IR, IG, IB).$$

(2) Adding diffuse shading, for that consider $x = x0 + t * dx, y = y0 + t * dy, \quad z = z0 + t * dz,$ then find unit normal vector(U) to the sphere at (x, y, z) and find the unit vector from (x, y, z) to light (Lx, Ly, Lz).

July – August 2020
ISSN: 0193-4120 Page No. 535 - 541

Now compute the $fact = cos(U, L)$ and set the pixel to

$$Fp = (ka * IR, ka * IG, ka * IB) + (kd * fact * IR, kd * fact * IG, kd * fact * IB),$$

where kd is the diffuse-coefficient and ka is theambient-coefficient.

(3) Add the shadows to the sphere so that it would look more round and real, for that consider

$$Fp = (ka * IR, ka * IG, Ka * IB) + (0.5 * kd * fact * IR, 0.5 * kd * fact * IG, 0.5 * kd * fact * IB).$$

(4) At last add the phong highlights by adding highlight factor(hf) to it with

$$Cp = (ka * IR, ka * IG, Ka * IB) + (kd * fact * IR, kd * fact * IG, kd * fact * IB) + (ks * hf, ks * hf, ks * hf)$$

where ks = coefficient defining shine of sphere. After applying these four steps the object will be rendered using raycasting.

III. DATA PREPARATION

Unlike most commonly observed volumetric data, such as medical images, which are not georeferenced and can conveniently be visualized in an isolated environment without resorting to multi-source data integration, atmospheric data have the followingcharacteristics:

(1) Atmospheric volumetric data are composed of arrays of spherical voxels located by longitude, latitude and altitude.

(2) Atmospheric volume data cover broad geographic areas, in some cases even the entire Earth, but represent a very narrow vertical range due to the thinness of Earth's atmosphere [13].

Dataset used for this study are the output from weather forecast model. It contains 3 hourly forecasts of different weather parameters like temperature, humidity, rainfall, wind etc. for the next 72 hours. These datasets are in NetCDF (Network Common Data Form) formats. NetCDF supports the creation, access, and sharing of array-oriented scientific data. The datasets contain a total 4 dimensions; Time, Latitude, Longitude, and Elevation as the forecast is available at 35 different vertical atmospheric levels. As the data represents weather parameters at 3 hourly time scales over the Indian region, keeping the time as constant value the three-dimensional array of latitude, longitude and elevation is used for volume rendering and visualization. The volume of daily data files is around 4.2 GB. The dataset covers 35 vertical levels of elevation with a horizontal resolution of 5x5 km. Dataset represents values at 400*400 ($lat\ x\ long$) with the area coverage of values -35 to +49.

IV. PROPOSED METHODOLOGY

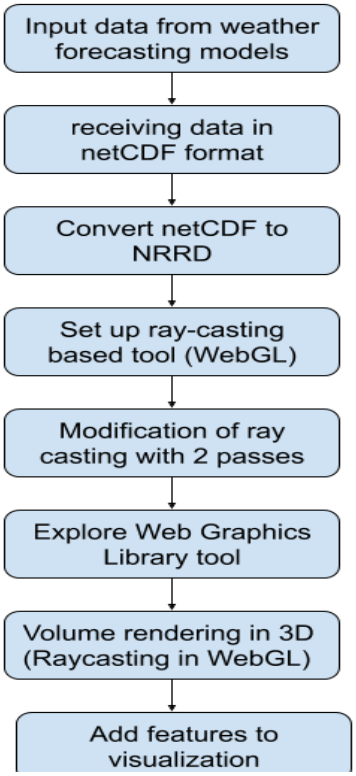

Figure 3: Workflow of proposedmethod

Fig. 3 represents the workflow of proposed method. For visualization, the following stepswill be performed:

(1) Convert raw data format to that format which can do volume rendering. (2) Explore the ray-casting implementation tool (3) Performing volume rendering of the data on the web (4) Add features to the visualized data.

A. Conversion of Data Format

July – August 2020
ISSN: 0193-4120 Page No. 535 - 541

NetCDF (.nc) files cannot be used directly for visualization on the web because these files are very large in size. To overcome this problem, NetCDF files are converted to another format for visualization. A rasterized format is chosen for visualization because of its rasterization characteristic.Rasterization can be defined as taking an image described in a different shape (vector graphics like shapes) and converting it into a series of pixels, dots or lines, which, when displayed together, create the image which was represented via shapes. Though it doesn't provide a way to compute the color of pixels, it simply is the process of computing the mapping from scene geometry to pixels. It can help in the volume rendering of data.

Keeping this advantage of rasterization under consideration,NRRD (Nearly Raw Raster Data) format is chosen as new data format for volume rendering.NRRD (.nrrd extension) represents and processes n-dimensional raster data. NRRD file contains header describing field and description of the field. The header also contains key value pairs. NRRD file can have attached or detached header and it can contain a single multi-dimensional array but it can't contain multiple arrays.

NRRD can read and write data in either byte-ordering, it can have the data in a separate file from the header (detached header). It can store the data of any dimension, and any C scalar type. It encodes data not just in ascii and binary, but also in gzip and bzip2.

It can also store more peripheral information, such as axis labels and units. In NRRD, vectors and tensors are represented implicitly, by using a short non-spatial axis prior to the spatial axes. Because of these advantages of NRRD, ncis converted to nrrd using python script.

Here, each NRRD file would contain 3-dimensional data. These files will have sizes in MB. By converting nc to nrrd, 13 nrrd files of 13 different time intervals are received. These fileswill be visualized in further implementation. For that, tool exploration is needed.

B. Exploration of Tool

As ray-casting algorithm is used for doing volume rendering on the web, WebGL tool can be used. As per name suggests WebGL stands for Web Graphics Library and it uses ray-casting for volume rendering.

WebGL is proposed as a new standard for plugin-less graphics computing from within a web browser. It provides a cross-platform, immediate mode, and royalty-free 3D graphics API. Since WebGL is based on the OpenGL ES 2.0 API which is a subset of OpenGL for embedded devices, it uses the same shader language framework as OpenGL. From the implementation point of view, WebGL is an extension of the HTML5 canvas element. This element provides fast and high-performance graphics constructs for rendering high-quality graphics in a browser window. For 3D rendering, rendering context is provided by OpenGL ES 2.0 functionalities. Both of these APIs are controlled through JavaScript in the web browser [15].

Moreover, WebGL has a three.js library. As per name suggests this provides geometry, shading, lighting, to visualize the data in 3D. Here, vertex shader transfers the geometry from one place to another, it works with shapes and coordinates. With fragment shaders different effects like color, lighting, shadow effect can be applied to the final image [16]. With ray casting, terrain visualization has also been done using WebGL tool [17]. This tool can be used for exact implementations of volume rendering on the web.

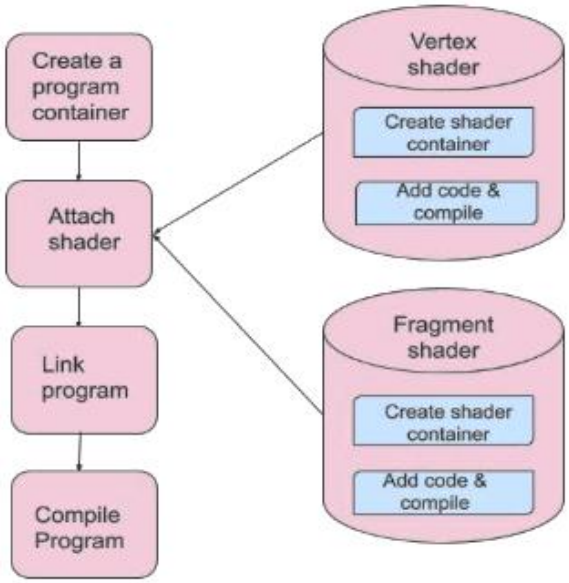

Figure 4: WebGL workflow

C. Volume rendering with ray-casting on WebGL

Ray-casting collects color and opacity at each point and render the volume. First nc files is converted to nrrd files. As WebGL is based on ray-casting, it renders the three-dimensional array as an object. WebGL supports NRRD

July – August 2020
ISSN: 0193-4120 Page No. 535 - 541

format for visualization but nrrd format can't have direct color or opacity. That's why another pass needed to be applied to the ray-casting algorithm for visualization.

1. In the first pass of ray-casting, data is visualized by applying 3D textures on the data. When ray is passed through the data, it collects information at each data point. (x,y,z,v) will be the coordinates denoting longitude, latitude, elevation, value of the data respectively. WebGL provides a geometry to give 3D shape to that collected data. In three.js, the geometry is visualized using textures, here the DataTexture3D method is used. In the second pass, 2D image iswrapped around that 3D visualized data to provide colors to the rendered data.
2. In the second pass, using texture loader, the 2D image is loaded to wrap the rendered data. This way, the three-dimensional array of weather data can be visualized.

In the next segment, features will be added to the visualized weather data.

D. Adding features to the visualized data

Here, ray-casting is applied in 2 passes to visualize the data in 3D. Now if data needs to be seen at more or less height, or in more depth or at different width then the dimensions of data should be variable. The feature of moving dimensions is added to see the data as per the requirements.

Different colors to the datacan also be given by applying colormap using different images containing shading of different colors. Colors of rainbow with 0 color blindness simulation, grayscale colormap, and green shades containing colored image have been applied to see the variation in the data.

Different rendering styles iso-surface and mipmap can be applied to the rendered data. To develop thesefeatures, controls are defined in three.js. Each control can be operated by a user on the browser. Let's see the experiments done on the three-dimensional nrrd file to generate the results.

V. EXPERIMENTS AND RESULTS

The experimental setup used a comprise of:

1) Dell Precision Power 7810 Desktop with Red hat Enterprise Linux Server 7.2, 64-bit operating system and IntelR Xeon ® CPU E5-2650 V3 @ 2.30GHz*40 processor.
2) The graphics card required to run WebGL 2.0 experiments contains certain requirements. The graphic card here contains NVIDIA Quadro M4000/PCIe/SSE2. The GNOME version for Linux environment is 3.14.2.

TABLE I
FEATURE DESCRIPTION

Feature1	Visualization is done at elevation level 1 with isosurface style and rainbow-colored image is wrapped around the rendered data shown in figure5(a)
Feature2	Visualization is done at elevation level 1 with mipmap style and the grayscale image is wrapped around the rendered data shown in figure5(b)
Feature3	Visualization is done at elevation level 1 with isosurface style and green shades containing image is wrapped around the rendered data shown in figure 5(c). It shows the variable dimensions.

After using WebGL for reading nrrd file and visualizing it, results are received as following. Result explanation are summarized in Table 1.

Figure 5 (a): Rainbow colors with isosurface

July – August 2020
ISSN: 0193-4120 Page No. 535 - 541

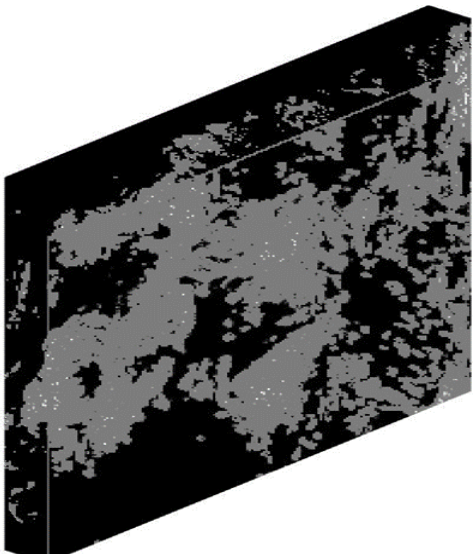

Figure 5 (b): Grayscale with mipmap

Figure 5 (c): Green scale with isosurface and variable dimensions

VI. CONCLUSION

Successfully achieved raw weather data rendering in 3D by mapping textures with images ofdifferent volumetric data while changing the range of dimensions as well as colormaps with thedata present in NRRD format performed on WebGL tool using Ray-casting algorithm. In future, at each elevation level, to observe data more precisely the slicing of 3D data can be achieved.For real time 4D web visualization system, generic tools can be developed to implementrendering of weather data.

ACKNOWLEDGEMENT

This research is acknowledged by Space Applications Center, Indian Space Research Organization (SAC, ISRO) enough resources and encouragement is provided for the research. This research couldn't be completed without the constant support of Dr. HiteishiDiwanji, Head of the Department (Information Technology) at LD College of Engineering.

REFERENCES

1. LEVOY M., Display of surfaces from volume data.
2. IEEE Computer Graphics and Applications 8, 3(1988), 29–37
3. Drebin, R. A., Carpenter, L. and Hanrahan, P .Volume rendering. Computer Graphics 1988,22,65-74
4. Levoy, M., "Volume Rendering, A Hybrid Ray Tracer for Rendering Polygon and Volume Data," IEEEComputer graphics and Applications, Volume 10, Number 2, March 1990, 33—40.
5. Levoy, M., "Efficient Ray Tracing of Volume Data," ACM Transactions on Graphics, Volume 9, Number 3, July 1990, 245-261.
6. Callahan, Steven P., et al. "Direct volume rendering: A 3D plotting technique for scientific data." Computing in Science & Engineering 10.1 (2008): 88-92.
7. Dubey, Rashmi, Sarika Jain, and R. S. Jadon. "Volume Rendering: A Compelling Approach to Enhance the Rendering Methodology." 2016 Second International Conference on Computational intelligence & Communication Technology (CICT). IEEE, 2016.
8. Lee, Ahyun, and Insung Jang. "Mouse Picking with Ray Casting for 3D Spatial Information Open-platform." 2018 International Conference on Information and Communication Technology Convergence (ICTC). IEEE, 2018.
9. Binyahib, Roba, et al. "A scalable hybrid scheme for ray-casting of unstructured volume data." IEEE transactions on visualization and computer graphics 25.7 (2018): 2349-2361.
10. Hamada, Kazuma, and Masaki Aono. "3D Indoor Scene Classification using Tri-projection Voxel Splatting." 2018 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC). IEEE, 2018.
11. Tan, Jianhao, et al. "Design of 3D visualization system based on vtk utilizing marching cubes and ray casting algorithm." 2016 8th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC). Vol. 2. IEEE, 2016.
12. Lacroute, Philippe, and Marc Levoy. "Fast volume rendering using a shear-warp factorization of the viewing transformation." Proceedings of the 21st annual conference on Computer graphics and interactive

July – August 2020
ISSN: 0193-4120 Page No. 535 - 541

techniques. 1994.

13. Mehaboobathunnisa, R., AA HaseenaThasneem, and M. Mohamed Sathik. "Ray grouping based ray casting for visualization of medical data." 2016 International Conference on Computing Technologies and Intelligent Data Engineering (ICCTIDE'16). IEEE, 2016.
14. Liang, Jianming, et al. "Visualizing 3D atmospheric data with spherical volume texture on virtual globes." Computers & Geosciences 68 (2014): 81-91
15. Y. Zhang, P. Gao and X. Li, "A novel parallel ray-casting algorithm," 2016 13th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP), Chengdu, 2016, pp. 59-61
16. Mobeen,Movania Muhammad, and LinFeng. "high-performance volume rendering on the ubiquitous webgl platform." 2012 IEEE 14th International Conference on High Performance Computing and Communication & 2012 IEEE 9th International Conference on Embedded Software and Systems. IEEE, 2012.
17. Lin, Yiming, Fan Zhang, and Wei Hu. "A novel 3D visualization method of SAR data." (2013): 0166-0166.
18. Luo, Jianxin, et al. "An efficient Ray Casting method for terrain visualization." 2011 International Conference on Multimedia Technology. IEEE, 2011
19. Palak Chatwani, Shivani Shah, PurviRamanuj, "Different Ray-Casting Algorithm Implementations for Volume Rendering" International Journal of Innovative Technology and Exploring Engineering (IJITEE) ISSN: 2278-3075,Volume-9 Issue-7S, May 2020.