Eleventh International Multi-Conference on Information Processing-2015 (IMCIP-2015)

Hardware Accelerated Multi-coordinate Viewing Framework for Volumetric Visualization of Large 3D Medical Dataset

Piyush Kumar* and Anupam Agrawal

Graphics and Visual Computing Lab, Indian Institute of Information Technology, India

Abstract

The advances in Computed Tomography (CT) and Magnetic Resonance Imaging (MRI) scanning techniques are improving the resolution and size of the volume datasets. The prevalence of three dimensional volumetric data is rapidly expanding with the internal information in the different resolution sizes of the dataset. In this paper we proposed an approach that can visualize the inner organs structure of the visible human male dataset in Multi-coordinate Viewing (MCV) framework. This can help to medical experts that are able to peer inside anatomy of the human medical dataset. The volume rendering part has been carried out by the utilization of enhanced ray casting algorithm for the crossing points of 3D square strategy for voxels. We present this system using Graphics Processing Unit or GPU-accelerated Compute Unified Device Architecture (CUDA) based approach for the focusing a specific region while zooming operation. The final results would allow the doctors to diagnose and analyze the atlas of 8-bit CT-scan data using three dimensional visualization with the efficient frame rate rendering speed in multi-operations like zooming, rotating, dragging. The framework is tested for visible human male dataset prepared by National Library of Medicine (NLM, USA) of size 1.2 GB.

© 2015 The Authors. Published by Elsevier B.V.

Peer-review under responsibility of organizing committee of the Eleventh International Multi-Conference on Information Processing-2015 (IMCIP-2015).

Keywords: Volume rendering; GPU-accelerated; Visualization; CUDA; Multi-coordinate viewing; Visible human male.

1. Introduction

Volume data visualization has been a priority for many fields of technology, especially in medical, designing, automobile engineering and simulation engines. Generally for a real life simulation in the medical domain we have to scan a bulk continues slicing of data and this volume dataset is too dense.

There have been many generations which are presenting both the focused region and a surrounding context visualization has not change much. We still acquire the most information about the unknown inner data of the anatomic organs with their shape and the position of the human body. Among these problems the plethora of examples demonstrating the necessity of man-made focus and context approaches are the anatomic sketches of Leonardo DaVinci1-8 as shown in Fig. 1.

The major issue behind the explanation of these things is that the abstraction is to condense the information in the dataset allows quick interactive and intuitive understanding.

*Corresponding author.

E-mail address: piyushkumar1786@gmail.com

1877-0509 © 2015 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

Peer-review under responsibility of organizing committee of the Eleventh International Multi-Conference on Information Processing-2015 (IMCIP-2015)

doi:10.1016/j.procs.2015.06.065
The structure of the data is in the form of a simple three-dimensional array of volume elements as shown in Fig. 2. Actually volume is used for three dimensions, therefore the 3D data here we used for volume visualization.

The resolution sizes of volumetric datasets we have dealing with increased from \(32^3 \rightarrow 512^3\) to \(1024^3\) voxels. This kind of dataset contains a lot of essential data and extraneous information as well, which may or may not be the interest for the user. Here, we are talking about the structure of the volume dataset as explained below. To keep the essential data focused to the user’s perspective we are required to hide extraneous information of volume dataset. The additional extraneous information usually referred to as the context. The user can communicate the exact shape and position of the inner organs of the intuitive anatomic volume data.

2. Related work

The research in focus+context has been started from 1500 A.D. In the ancient time an artistic Leonardo DaVinci\(^1,8\) use discovered many information related to human body anatomy. The way which is used by his was sketches of the human body as shown in Fig. 1. But during that time he sketches only the parts of human body like: brain, heart and other many parts in the field of neuroanatomy. Now time to do some improvement on the previous research era.

In flow of the scientific volume data visualization in the area of the focus and context visualization is divided in many categories: like if some researchers are working on the magic lens and magic boxes then some are working on
Table 1. Ray intersection approach.

| Algorithm: Ray Intersection Multiple Box Method |
|------------------------------------------------|
| Compute the intersection distance of the planes \( r \rightarrow \text{ray and } d \rightarrow \text{direction} \) |
| 1. \( T_1 = (X_l - X_o)/X_d \) |
| 2. \( T_2 = (X_h - X_o)/X_d \) |
| 3. If \( T_1 > T_2 \) swap \((T_1, T_2)\) since \( T_1 \) intersection with near plane *|
| 4. Inverse view matrix = view matrix \((1.40)/r.d//\text{intersection of ray with all six bbox planes} \) |
| 5. If \( T_1 > T_{\text{near}} \) set \( T_{\text{near}} = T_1/\ast \text{want largest } T_{\text{near}} \ast/ \) |
| 6. If \( T_2 < T_{\text{far}} \) set \( T_{\text{far}} = "T_2"/\ast \text{want smallest } T_{\text{far}} \ast/ \) |
| 7. If \( T_{\text{near}} > T_{\text{far}} \) box is missed so return false |
| 8. If \( T_{\text{far}} < 0 \) box is behind ray return false end |

the fisheye system. Milan et al.⁶ presented a GPU-based volume rendering by introducing an interactive lens that is also a type of magic lens. It is the process through magnification of the data voxels. The authors provide the finger details within the region of interest with high-quality rendering. Lujin et al.¹⁰ proposed a focus + context framework that uses various advanced magnification lens rendering approach. They are using the rendering technique to magnify the feature of the interesting area.

Yingcai et al.¹¹ proposed a framework for the focus and context volume data set visualization for some novel animation techniques. The animation part is used to help the users to understand complex and dense structure of voxels contained from the medical domain and scientific simulation fields. Actually, the 3D relationships of the data cannot be revealed by using only one image. Therefore, the authors presented animation technique, in which they demonstrate the visualization with relationship between the focus regions and the context part. There are visualizing different stages of an animation. The author proposed an image-centric method which is used for layered depth images (LDIs) to get correct depth cues⁹. The novel transfer function technique for smoothing transition between frames.

Matej et al.¹² presented a focus+context volume data visualization in the parallel coordinates in which the small-scale features are detected before visualization and specially this will be during the contextual visualization. The binned data representation is used for introducing the outlier detection and the context visualization in parallel coordinates. Here, the performance of this solution is dependent on the rendering size not on the data size. Yu-Shuen et al.¹³ presented a feature-preserving data reduction method which is based on the transfer function system. Their method utilizes the input transfer function values to continue partitioned regions of the volume data. When the user interact this system, it can then magnify the regions like magnifying lens corresponding to the features of the interest. This method avoids the need to smooth the transition between the low and high-resolution regions which is required by multi-resolution methods.

Yanlin Luo et al.¹⁴ proposed a distance-based system for the focus + context visualization of the volumetric medical datasets. In this paper the author used the direct volume rendering (DVR) based ray casting with maximum importance difference accumulation (MIDA). Nadezhda et al.¹⁵ presented a generalized temporal focus+context framework for improved medical data exploration.

3. Materials and Proposed Methodology

The materials which we used for the focus and context based volume visualization system are based on some hardware and some conceptual approach of basic computer graphics. In hardware, we are using NVIDIA Tesla C1060 GPU with CUDA enabled personal computer system. On the other hand, the basic concepts of 3D transformation with the navigating by the mouse events using some OpenGL functions.

3.1 Ray intersecting in a rectangular box

In ray intersection for a geometry model like rectangular/square box we expects the box with ordered corners indicates min and max, and a ray \( R(t) \), and a valid intersection points as \( T_{\text{near}} \) and \( T_{\text{far}} \) to be given as shown in Fig. 3.

568
3.2 Proposed methodology

Most of the volume visualization techniques make an abstract representation of the original volume data. In the framework for focus and context in volume visualization we have taken a Cartesian ROI (Region of Interest). Focus defines the nature of the specific region for focusing. This is happen when we click the mouse left button on the cubical grid of visualized 3D volume data. This is a rectangular region. In the 3D Cartesian bifocal region, this rectangular region will be uniformly focused by a factor and the remaining sections of the volume data will be context. We proposed a multi-coordinate viewing system for the streaming of the volume elements (voxels) in any coordinates as shown in Fig. 4. We can explore the 3D medical data and the focusing of the inner voxels in the same rendering speed.

The color and opacity of those colors will be evaluated in ray tracing method. This is a backward mapping approach, in which we traverse a ray for different cube known as voxels for volume data. The ray fired the plane surface composition structure for every voxel.
Now consider that each volume element is indexed by \( V = (x, y, z) \). The color \( C(v) \) and opacity \( \alpha(V) \) are initialized for every voxel. If \( \alpha = 1 \) implies that this is an opaque voxel otherwise \( \alpha = 0 \) implies transparent voxel. If the coordinate is \((i, j)\) then we define as a vector \( R(i, j, k) \). The \((i, j) = r \) is the voxel index of the ray, and \( k \) is the depth along the ray which will be, \( k = 1, 2, 3, \ldots, k \). For each voxel we have to re-sample the volume data at every interval and computing color \( C(R) \) and \( \alpha(R) \). Now we calculate those values which are going along a ray in a front-to-back order. This will evaluate the process for the standard transparency formula and for the sampling of \( R \) (ray) we have to calculate the color and opacity first then processed as following equations:

\[
C^\bullet_{\text{out}}(r, R) = C^\bullet_{\text{in}}(r, R) + C^\bullet(R)[1 - \alpha_{\text{in}}(r, R)]
\]

\[
\alpha^\bullet_{\text{out}}(r, R) = \alpha^\bullet_{\text{in}}(r, R) + \alpha^\bullet(R)[1 - \alpha_{\text{in}}(r, R)]
\]

where:

\[
C^\bullet_{\text{in}}(r, R) = C_{\text{in}}(r, R)\alpha_{\text{in}}(r, R)
\]

\[
C^\bullet_{\text{out}}(r, R) = C_{\text{in}}(r, R)\alpha_{\text{in}}(r, R)
\]

Then we get,

\[
C^\bullet(r, R) = C(r, R)\alpha(r, R)
\]

All the colors are first multiplied by its associated opacity then \((R, G, B)\) will become \((R_\alpha, G_\alpha, B_\alpha)\) in the evaluation purpose.

In the end of the loop for each ray \( R(i, j, k) \), we get the final colors for lookup table from following equation:

\[
C(r, R) = \frac{C_{\text{out}} \ast (r, R)}{\alpha_{\text{out}} \ast (r, R)}
\]

4. Experimental Results and Analysis

We have tested the results in the GPU with CUDA enabled system. The configuration of this system is Intel(R) Core(TM) i5 CPU 650@3.20 GHz with 4 GB RAM. Also we are using NVIDIA GPU Geforce GTX470 with 448 CUDA core enabled system. The datasets can be downloaded from http://www.gris.uni-tuebingen.de/edu/areas/scivis/volren/datasets/new.html. Also we have tested the multi-coordinated viewing system of focusing region on the visible human male volume dataset of resolution 1760 x 1024 x 700 and the data size is 1225523200 Bytes (1.2 GB).

4.1 Experimental results

This viewing Multi-coordinate Viewing (MCV) framework has been done by a single CUDA kernel. The opacity required for the changing the sampling rate globally or locally\(^2\) and this will be evaluated by the equation (3) as given below:

\[
\alpha_{\text{corrected}} = 1 - (1 - \alpha_{\text{stored}})^\bullet
\]

where, \(\alpha_{\text{corrected}}\) is adjusted opacity and \(\alpha_{\text{stored}}\) is opacity stored in transfer function\(^2\). Transfer functions shows variation between opacity and scalar values. The transfer function would be computed by the following scale as shown in Fig. 5.

In case of the zooming operation, when we use the medical volume data in Multi-coordinate Viewing System (MCVS) then this results proper shape and position of the parts of the anatomy of the Visible Human Male dataset as shown in Fig. 6.
Table 2. Result analysis of the tested volume datasets.

| Data set            | Dimension              | Execution speed (FPS) | Rendering speed (FPS) | Operation |
|---------------------|------------------------|-----------------------|-----------------------|-----------|
| MRI_ventricle       | 256*256*124            | 0.54                  | 72–76                 | 57–60     |
| Skull               | 256*256*256            | 0.63                  | 72–75                 | 53–56     |
| Mrt8_angio2         | 256*320*128            | 0.75                  | 71–74                 | 51–55     |
| StatueLeg           | 341*341*93             | 0.88                  | 71–73                 | 49–54     |
| Angio               | 416*512*112            | 0.96                  | 68–70                 | 48–50     |
| Stent8              | 512*512*174            | 1.12                  | 66–69                 | 47–50     |
| Supine8             | 512*512*426            | 1.30                  | 62–65                 | 42–45     |
| Prone8              | 512*512*463            | 1.35                  | 61–66                 | 41–44     |
| Vertebra8           | 512*512*512            | 1.41                  | 60–64                 | 40–42     |
| VHM                 | 1760*1024*700          | 1.72                  | 59–62                 | 39–42     |

4.2 Result analysis

The analysis of the tested datasets are shown in Table 2 as given above. According to the table, results shown above are visualized in the efficient execution time with the effective rendering speed in frame per second.

The size of the block can be varying in a limited flow then the rendering speed of the visualized data will be increased. We got 65–75 fps rendering speed for the block size of 32 * 32 * 32. Therefore, the variable BlockSize is totally inverse proportional to the rendering speed FPS.

\[
\text{BlockSize} = \frac{1}{\text{FPS}}
\]
However, the same is applied for the variable GridSize, therefore

$$\text{GridSize} = \frac{1}{\text{BlockSize}}$$

The above equation means that the GridSize is totally inversely proportional to and depend on the used BlockSize. If we vary the BlockSize the size of the variable GridSize will change accordingly. The graph related to analysis of this computation is shown as in Fig. 7 in which the execution time in seconds and the rendering speed.

The graph as shown in Fig. 8 is a proper evaluation of the rendering speed in different terms. First is the rendering speed in case of focusing a particular region in a continuous manner. Another is the operation speed in case of the performing operations.

5. Conclusion and Future Work

The advances of the GPU techniques is the changing the scenario of the complex computations were done till now. It would play a major role in the diagnosis of the large 3D medical volume data visualization. We proposed some GPU-accelerated multi-coordinate viewing framework for focusing of specific voxels. We can also perform some operations through mouse and keyboard like zooming, rotating, and dragging of the exploring of the inner organ location and shape of 3D medical dataset. This will help to the physicians for exploring the slices of 3D volume dataset.

Now a days, the GPGPU enabled CUDA systems may compute out-of-core dataset with the complex calculations in remote visualization\(^5,19,21\). We will try to go through some hardware like hand dataglove\(^22\) for navigating to visualize medical dataset.
References

[1] Krüger Jens, Schneider Jens and Westermann Rüdiger, ClearView: An Interactive Context Preserving Hotspot Visualization Technique, IEEE Transaction on Visualization and Computer Graphics., vol. 12, No. 05, pp. 941–948, (2006).

[2] Kumar Piyush, Agrawal Anupam, GPU-Accelerated Interactive Visualization of 3D Volumetric Data using CUDA, World Scientific International Journal of Image and Graphics, vol. 13, issue 02, pp. 1340003–13400017, (2013).

[3] Doleisch Helmut, Gasser Martin, Hauser Helwig and Interactive Feature Specification for Focus + Context Visualization of Complex Simulation Data, IEEE TCVG Symposium on Visualization, pp. 239–248, (2003).

[4] Ikits Milan, Kniss Joe, Lefohn Aaron and Hansen Charles, Volume Rendering Techniques, in Book Randima Fernando, GPU Gems NVIDIA. http://http.developer.nvidia.com/GPUGems/gpugems_ch39.html. Last accessed, 15 November 2013, pp. 667–672, (2004).

[5] A. Agrawal, Kohout Josef, Clapworthy J. Gordon, Mcfarlane J. Nigel, Dong Feng, Viceconti Marco, Taddei Fulvia and Testi Debora, Enabling the Interactive Display of Large Medical Volume Datasets by Multiresolution Brickling, Springer The Journal of Supercomputing, pp. 3–19, (2010).

[6] Ikits Milan and Hansen Charles D., A Focus and Context Interface for Interactive Volume Rendering. http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.119.9198, Last accessed on 23, March (2014).

[7] Fogal Thomas, Schiewe Alexander and Krüger Jens, An Analysis of Scalable GPU-Based Ray-Guided Volume Rendering, IEEE Transaction on Visualization and Computer Graphics, (2013).

[8] L. DaVinci, Quaderni D’anatomia. I-VI. Fogli Della Royal Library Di Windsor, pp. 1452–1519.

[9] J. Fairfield Adam, Plaseneia Jonathan, Jang Yun, Theodore Nicholas, R. Crawford Neil, H. Frakers David and Maciejewski Ross, Volume Curtaining: A Focus + Context effect for Multimodal Volume Visualization, In Proceeding of SPIE Journal of Medical Imaging; A Computer-Aided Diagnosis, vol. 9035, (2014).

[10] Wang Lujin, Zhao Ye, Mueller Klaus and Kaufman Arie, The Magic Volume Lens: An Interactive Focus + Context Technique for Volume Rendering, IEEE International Conference on Visualization, USA, pp. 367–374, (2005).

[11] Wu Yingcai, Qu Huamin, Zhou Hong and Chan Ming-Yuen, Focus + Context Visualization with Animation, Springer International Conference on PSIVT, pp. 1293–1302, (2006).

[12] Novotný Matej and Hauser Helwig, Outlier-preserving Focus+Context Visualization in Parallel Coordinates, IEEE Transaction on Visualization and Computer Graphics, vol. 12 no. 05, pp. 893–890, (2006).

[13] Wang Yu-Shuen, Wang Chaoli, Lee Tong-Yee and Ma Kwan-Liu, Feature-Preserving Volume Data Reduction and Focus+Context Visualization, IEEE Transactions on Visualization and Computer Graphics, vol. 17, no. 02, pp. 171–181, (2011).

[14] Luo Yanlin, Distance-Based Focus + Context Models for Exploring Large Volumetric Medical Datasets, IEEE International Conference on CS and AIP, pp. 63–72, (2012).

[15] Radeva Nadezhda, Levy Lucien and Hahn James, Generalized Temporal Focus +Context Framework for Improved Medical Data Exploration, Springer International Journal of Digital Imaging, vol. 27, pp. 207–219, (2014).

[16] Gallo Luigi and A. P. Placitelli, High-Fidelity Visualization of Large Medical Datasets on Commodity Hardware, Hindawi Journal on Biomedical Engineering, vol. 2013, pp. 1–9, (2013).

[17] B. Nelson, R. M. Kirby and R. Haimes, GPU-Based Volume Visualization from High-Order Finite Element Fields, IEEE Transactions on Visualization and Computer Graphics, vol. 20, pp. 70–83, (2014).

[18] C. Tominski, S. Gladisch, U. Kister, R. Dachselt and H. Schumann, A Survey on Interactive Lenses in Visualization, Eurographics Conference on Visualization, (2014).

[19] A. P. Singh, P. Kumar and A. Agrawal, Interactive 3D Remote Visualization of Medical Volume data in Distributed Environment, International Journal of Advanced Computer Research (IJACR) vol. 3(3), (2013).

[20] J. Beher, M. Hadwinger and H. Pfister, A Survey of GPU-Based Large-Scale Volume Visualization, Eurographics Conference on Visualization (EuroVis), (2014).

[21] A. Agrawal, J. Kohout, G. J. Clapworthy, N. J. B. McFarlane, F. Dong, M. Viceconti, F. Taddei and D. Testi, Interactive Out-of-Core Exploration of Large Volume Datasets in VTK-Based Visualization Systems, Eurographics Int’l Conf. on Theory and Practice of Computer Graphics, UK, pp. 10–15, (2008).

[22] P. Kumar, S. S. Rautaray and A. Grawal, Hand Data Glove: A New Generation Real-Time Mouse for Human-Computer Interaction, IEEE International Conference on Recent Advances in Information Technology, pp. 750–755, (2012).