An Accelerating 3D Image Reconstruction System Based on the Level-of-Detail Algorithm

Tsung-Yu Lee, Member, GSTF, Ren-Guey Lee, Sheng-Chung Tien, Robert Lin, Wei-Hua Su

***Abstract*-This paper proposes a research of An Accelerating 3D Image Reconstruction System Based on the Level-of-Detail Algorithm and combines 3D graphic application interfaces, such as DirectX3D and OpenCV to reconstruct the 3D imaging system for Magnetic Resonance Imaging (MRI), and adds Level of Detail (LOD) algorithm to the system. The system uses the volume rendering method to perform 3D reconstruction for brain imaging. The process, which is based on the level of detail algorithm that converts and formulates functions from differing levels of detail and scope, significantly reduces the complexity of required processing and computation, under the premises of maintaining drawing quality. To validate the system's efficiency enhancement on brain imaging reconstruction, this study operates the system on various computer platforms, and uses multiple sets of data to perform rendering and 3D object imaging reconstruction, the results of which are then verified and compared.**

***keywords*—Level of Detail ; GPU ; 3D Reconstruction**

I. Introduction

Before 2001, there is no official issue about vertex processor or pixel processor, 3D graphical computation is usually achieved by Vector Processor [1].The main purpose of Vector processor is accelerating the speed of matrix-computation, which uses VLIW (Very Long Instruction Word) or Floating Point Multiple Accumulator Processing Element of SIMD (Single Instruction Multiple Data) to accelerate the computation.

3D, or multiple dimensions, medical imaging reconstruction is being more concerned in medical field. However, it requires vast computation and data flow capacity, so that it fails to fulfill the need for immediate reconstruction. Around this time, the multi-core processor architecture brings forth new visions and new possibilities for digital signal processing and image processing because of its coming into the market and being widely adopted.

Due to the digital medical imaging has become more acceptable by medical field, plus the exponential growth of the computation capacity of the central processing unit (CPU) and graphic processing unit (GPU), the integration of 3D medical imaging and surgical planning has become a trend in the surgical field [3][4]. Recently, the need for dental implanting, etc., accelerates the research being invested in 3D imaging in the dental field also results in medical experts paying more attention to the development of 3D imaging [5][6][7]. In order to establish 3D modeling for medical imaging, data processing capability of the hardware increase for coping with vast data computation, so does the cost of the system.

MRI, which has the sharper imaging ability on soft tissues, has been usually adapted on diagnosis of soft tissues diseases, and its discriminating ability of hard tissues inside is better than CT or X-ray. Due to MRI has been used widely on clinical, MRI image discrimination and 3-D model reconstruction also get more attention gradually, recently, research of image-reconstruction on joint cartilage becomes important[8][9].

It is under these premises that the motivation for this research arises. While building the 3D modeling of MRI, we use an algorithm related to LOD [10] to simplify the vertex data of the model so as to reduce data volume needed for the computation of 3D imaging reconstruction, also to achieve real-time computation by the hardware platform.

II. System Design

The concept design process of the system is shown in Figure 1, and is divided into two parts.

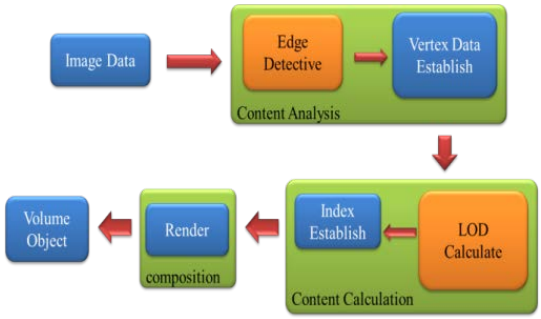

Figure 1 System Design Process Diagram

After reading image data, noise in the image data is decimated first, then valid vertex information is established, followed by dynamic reading of valid information, based on view angle and distance, to build 3D models. Details are described as follows：

1) Data Analysis and Setup:

After reading the data file, the rendering environment must be set up for computation. The immediate next step is to perform edge detection with the Canny Algorithm, and then perform logical operations on the original image to decimate noise in the file and confirm valid data. The valid data is then converted into screen coordinates. Then use the conversion specialized memory architecture to store vertex color and coordinate location information.

DOI: 10.5176/2251-3043_3.3.274

2) LOD Algorithm:

The Vertex Clustering Algorithm is then applied to the established data to perform partial LOD computation, then dynamically adjusts vertex to build indexes by referencing the application's parameters, such as adjusted view angles, and distances. Using the indexes of real-time computation, a GPU can dynamically read information of the intended vertex to be drawn, so that it can reduce the required data volume for computation while maintaining model details.

A. LOD

Discrete Level of Detail, which is the level of detail technique appeared earlier, it created each kind of different version of LOD for model at the preprocessing stage, then choose the most appropriate level of model in executing stage to draw; Continuous Level of Detail is creating a data structure point at original model, and encoding the continuous spectrum of detail creation, in executing stage, it will retrieve required resolution from data structure depends on different condition; View Dependent Level of Detail is a technology extended from continuous level of detail, except for containing characteristic of normal continuous level of detail, multi-layer resolution which depends on view point changes is Anisotropic, that is the same model can derive many kinds of simplified level, for example, the closer part of object can be depicted as higher resolution; in contrast, the farer part of object is drawn as lower resolution, shown as figure 2.

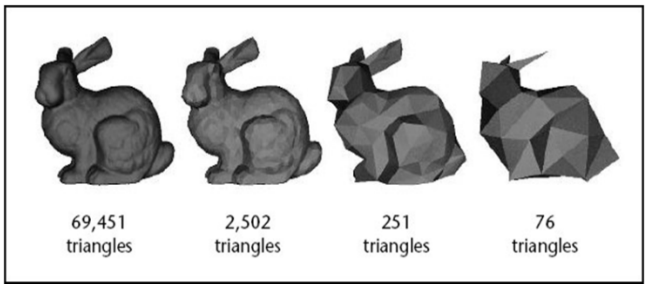

Figure 2 Different resolution of multi-layer detail.

As the distance between a model and an observer's viewpoint increases, the occupied area of the model in the observer's viewing scope decreases. At the meantime, using the simplified model not only has less levels of detail, but less numbers of polygons, risks little in reducing the visual quality, but can reduce the burden of computation [11]. This technology is called Level of Detail. Interpretation is shown as figure 3.

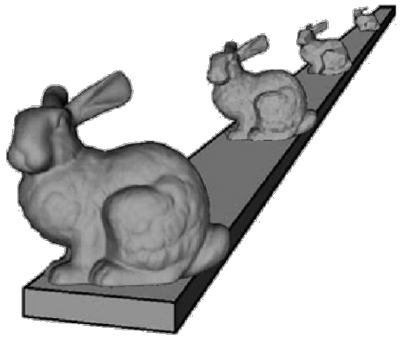

Figure 3 Appearance of model fine degree under different ranges.

This research adopts view dependent level of detail. It's retaining general characteristics of continuous LOD, and the multi-layer resolution is anisotropic, i.e., the same model can generate multiple levels of simplification.

B. Local Model Simplification of LOD Algorithm

Local simplification means to make adjustments to local areas so as to simplify the model, popular algorithm such as Vertex Removal [12], Edge Collapse [13], Vertex Clustering [14].

1. Vertex Decimation：

This method is depending on the feature degree of geometric shape around vertex to repeatedly choose a candidate vertex, this candidate vertex will be remove in next stage, each vertex decimation process contains two steps, first, vertex removal, this step will remove the triangle which close to two vertices, then mesh model will left a hole there; Second step is Triangulation, using mesh triangulation algorithm to fill the vacancy.

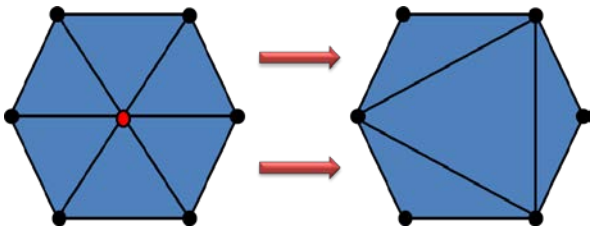

Figure 4Vertex Decimation.

2. Edge Collapse：

This method can simplify a vertex and two triangles at most once, algorithm such as Hoppe and Garland et al. , their method can establish multi-layer resolution tree structure of mesh model, by the relationship between each vertices from each time simplified steps record, we can create a hierarchy structure made of each vertices information, it's called Vertex Binary Tree. However, Edge Collapse is difficult to deal with boundary, as shown in figure 5, it will produce another new vertex after computation.

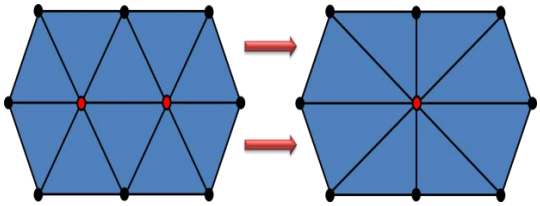

Figure 5 Edge Collapse.

3. Vertex Clustering：

The multi-layer resolution mesh model simplification of Vertex Clustering can be regarded as a resample method to geometric vertex. If there are many close vertices almost around the same pixel after projecting to image surface, these vertices can be replaced by a representative vertex, in this way, not only will the complexity being reduced, but also maintaining output quality.

In normal solution, the original model bounding box will be uniform spatial subdivision, making the occupied space of each mesh model being separated into several tiny grids, then choosing the most representative vertex to replace other vertices in the same tiny gird, eventually, all of the tiny grids will be re-triangulation to retrieve lower complexity geometric model, which is depicted as figure 6.

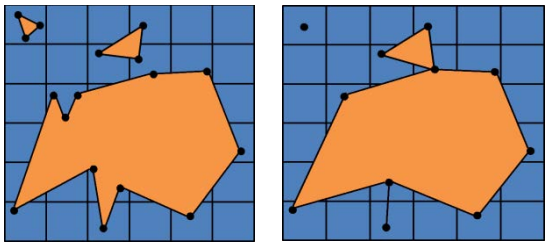

Figure 6 Vertex Clustering.

Here we use Vertex Clustering, it can be regarded as a multi-resolution mesh modeling simplification method that re-samples vertices of a geometric model.

In the other hand, “Partial Simplification” means adjusting data on partial fields to simplify the model, but it will cause the medical image having more edges and corners (or holes) while being transferred from 2-D to 3-D, and such of this kind of image won’t be able to be utilized by medical stuffs, it’s important to issue an algorithm-simplification model with boarder appliance to produce the presentation of LOD, so that we divide algorithm simplification into two parts, “Surface-Direction Computation ” and “Geometry Simplification”. This paper assumed that we all deal with the triangle-meshes, if the target isn’t a triangle-meshes, it has to be triangulated first, and we will assign a triangle connection index of an integral model, then compute it in triangle-surface-direction to setup the simplified degree of each partial fields, after computing, we will do the geometry-simplification to the target, then the LOD details will be presented through “Surface-Direction Computation ” and “Geometry Simplification” , as figure 7 shows.

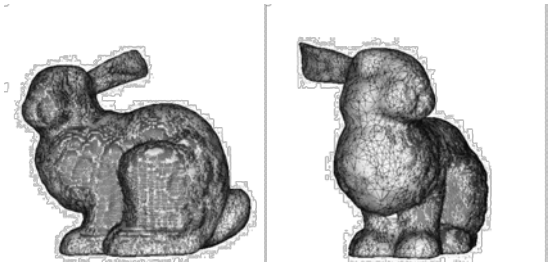

Figure 7 Level fineness modification based on viewing angle.

When saving data, a file resolution algorithm is used to subdivide mesh uniformly. Various levels of geometry simplification data are set up while establishing a data file. Assuming that vertex data of the model contains n_a points, we can then use this expression: $(X_i, Y_i, Z_i), \mathrm{i} = 1, \ldots, n_a$. To speed up data processing, we can use a method of regularity to store vertex array data, which reads many vertices at the same time. This data structure is called grid data structure. Vertex data can be easily partitioned for storage with grid data structure. When there is a need to carry out geometry calculations of the neighboring points, the rudimentary method of reading and calculating each and every point can be avoided, which significantly shortens the time for data processing, and enhances the efficiency of system computation.

Grid data structure is established with the following steps:

1. Calculate the values of $x_{min}, x_{max}, y_{min}, y_{max}, z_{min}, z_{max}$.

2.Calculate the sizes of Grid：

$$\mathrm{size} = \alpha \times \sqrt[3]{\frac{(x_{max}-x_{min})(y_{max}-y_{min})(z_{max}-z_{min})}{n_a}} \quad (1)$$

Wherein n_a is the total number of points of a point cloud, and α is a constant, the grid is adjusted as the rate of simplification differs.When α=1, the physical meaning of size is the average distance between two adjacent points. As α's value increases, the grid size increases, and number of points within the grid increases; conversely, as α decreases, the grid size decreases.

3. Formulae for calculating the size of the grid in X, Y, Z directions are as follows:

$$x_{res} = \left\lfloor \frac{x_{max}-x_{min}}{size} \right\rfloor + 1 \quad (2)$$

$$y_{res} = \left\lfloor \frac{y_{max}-y_{min}}{size} \right\rfloor + 1 \quad (3)$$

$$z_{res} = \left\lfloor \frac{z_{max}-z_{min}}{size} \right\rfloor + 1 \quad (4)$$

Grid model requires adaptable adjustment to set up differing grid simplifications while data files are being constructed. Therefore, the setting for the grid size is

adjusted according to the planar directions and distances of the triangle.

C. Projection Algorithm

The projection methods for 3D image drawing can be divided into two categories: Orthogonal View, which is sometimes called Orthographical View, and Perspective View [15][16]. 3D objects of the same size drawn with the former will look the same regardless of their distance to the lens; however, when drawn with the later, objects that are farther away look smaller, which is closer to realistic view, that's the reason why we use it.

The most important point that a perspective view can generate a 3D feeling is to show distance, make the object which in the distance smaller. Therefore, the calculation for the projection must be adjusted. Not only making the 3D vector to be resizing, the 3D vector scale must be changed proportionally based on the distance between the vector and the lens, so projection of orthogonal view angle is just an easy scaling computation for linear zooming of camera lens coordinate axes vertices on X, Y and Z axes. Orthogonal projection matrix computing formula of Direct3D:

$$\begin{bmatrix} \frac{2}{w} & 0 & 0 & 0 \\ 0 & \frac{2}{h} & 0 & 0 \\ 0 & 0 & \frac{1}{Z_{near}-Z_{far}} & 0 \\ 0 & 0 & \frac{Z_{near}}{Z_{near}-Z_{far}} & 1 \end{bmatrix} \quad (5)$$

Z value transformation of orthogonal will be more complicated, lens setting adapts Z_{near}、Z_{far} to express the closest and farthest location. The Z axis transformation in projection matrix except for scaling Z value, it will use the horizontal shifting to transform the vertex which distance from lens is Z_{near} and Z_{far} to 0 and 1.

The reason why non-orthogonal view angle can make object looked like 3-D, the first requirement is creating visual distance to shrink farer object, so there must have some adjustment on projecting computation, besides scaling 3-D vector, the shrinking proportion of 3-D vector has to be changed according to distance between vector and lens.

The non-orthogonal projection formula used in Direct3D is described as below:

$$\begin{bmatrix} xscale & 0 & 0 & 0 \\ 0 & yscale & 0 & 0 \\ 0 & 0 & \frac{Z_{far}}{Z_{near}-Z_{far}} & -1 \\ 0 & 0 & \frac{Z_{near}*Z_{far}}{Z_{near}-Z_{far}} & 0 \end{bmatrix} \quad (6)$$

$$xscale = cot(fovY * 0.5) * ScreenSize_x/ScreenSize_y \quad (7)$$

$$yscale = cot(fovY * 0.5) \quad (8)$$

When establishing non-orthogonal view projecting matrix, the difference between orthogonal projecting matrix is that the former uses lens' view angle as parameter, then decides shrinking proportion of X and Y axes with screen's length and width proportion. In fact, the shrinking rate after transforming by formula is the distance between vertex and lens on Z value, the farer the value, the farer the distance, the bigger rate will shrink vector smaller.

The view angle projecting matrix after computing will be saved in constant register by CPU, after computing the matrix of rendering spatial projection and object projection, a mixing computation of matrix will be executed to transform following object's vertices to screen coordinate axes.

D. Edge Detection Algorithm

Due to the image will not be the original one after undergoing analog to digital signal transformation, that is, distortion happened, in order to increasing accuracy of other following data computation and establishment, and raising display quality of 3-D reconstruction, the image pre-processing step must be implemented first.

Our system have compared and analyzed these three algorithms, Sobel, Prewitt and Canny Edge Detection, while executing data analyzing and establishing, at last, this paper adapts Canny Edge Detection algorithm then adjusts and modifies it for our experiment, and we are preparing to issue and descript this in oral presentation. As shown in figure 8 and figure 9.

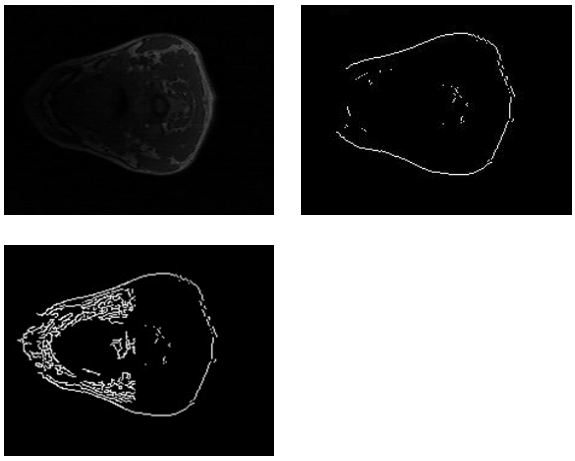

Figure 8 Detecting result of image after partial strengthening the soft tissues.

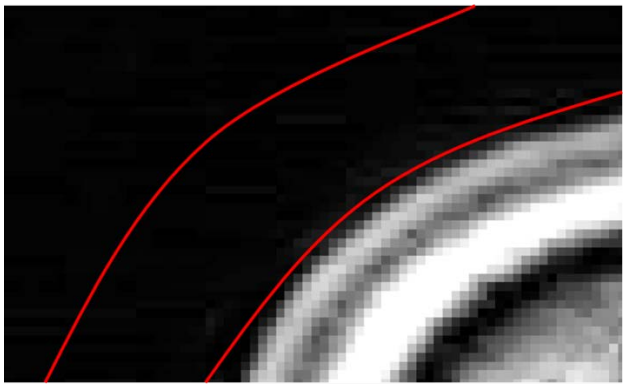

Figure 9 Modification of the noise which is hard to be distinguished in Canny algorithm.

E. System Optimization

The data source of a Vertex Shader mainly comes from the vertex data of a Vertex Buffer and Constant Registers. The information of constant register must be processed by the CPU for inputting to constant registers that, is to be used later by the GPU.

After the information is stored in a register, the GPU can proceed with the following calculation. Thus, the data read and write speed of a CPU can affect the operating efficiency of a program. Other than using a CPU with a higher clock rate to further enhance system operation performance, would be using parallel computing [17].

A popular parallel program running under a shared memory architecture that is used on stand-alone personal computers is Multi-Thread. Also, other commonly used APIs are Pthreads from Microsoft, and an open source application, OpenMP.

In a computer program, with the premise of not affecting information accuracy, OpenMP multi-thread function is usually added to the data analysis and data handling portions, so that system computation capacity and performance can be enhanced.

III. SYSTEM IMPLEMENTATION

A. Operating Environment and Software pecifications

For system implementation and performance evaluation, this research applies this system on personal computer platforms with 4 different specifications, and uses 3 different sets of MRI image data to perform system performance testing. The software versions and operation systems for each platform are listed in the table below:

Table1 Computer specifications of experimental group (A) and (B)

	Experiment (A)	Experiment (B)
CPU	Intel(R)Core(TM)i7-2600@3.40GHz	
RAM	DDR3-1333 4GB	
VGA CARD	NVIDIA Quadro 600	
OS	Windows 7 Enterprise x32	
SOFTWARE	Visual Studio 2008 SP1	
LIBARY	(1)Open CV Library V2.3(2)DirectX	

The difference between the experimental group (A) and experimental group (B) is in the use of a professional grade GPU or not. The purpose of this arrangement is to test the system performance when using a CPU to simulate a GPU executing drawing operations.

Table 2 The Computer Specifications of experimental group (C) and (D)

	Experiment (C)	Experiment (D)
CPU	Intel(R)Core(TM)i7-2630QM@2.00GHz	AMD Phenom(tm)II X4 945 Processor 3.00GHz
RAM	8GB	DDR3-1333 4GB
VGA CARD	NVIDIA GeForce GT550M	NVIDIA GTS250
OS	Windows 7 Enterprise x32	
SOFTWARE	Visual Studio 2008 SP1	
LIBARY	(1)Open CV Library V2.3(2)DirectX	

The major difference between experimental group (C) and experimental group (D) is that the testing is executed on different platforms, such as observing how differing CPU clock rates impact system performance, and performing a system operating speed comparison while using differing GPU clock rates and different numbers of rendering pipelines. The results are used to specify basic equipment criteria for the platform required in reconstructing medical 3D imaging.

All three sets of MRI files used in system testing are human head T1 images, provided by the Brain Connectivity Laboratory of Yangming University.

Table3 Data of Test

	No.1	No.2	No.3
Resolution	256*224	384*508	384*484
Files	175	176	176
Data size	12.3MB	16.5MB	16.5MB
File architecture	JPEG	JPEG	JPEG
The total number of vertex	1,868,971	5,020,896	3,615,990

The number of vertex and the levels of resolution of testing data No. 1 are smaller, the purpose of which is to test system computation and simplification capabilities when testing data volume is small, and to test the system performance prior to scope simplification.

B. User Interface

The system provides a 3D image displaying function along with the 2D image display function to assist in observation and analysis.

The user interface includes three 2D image display windows and one 3D image display window. Presented in the 2D image display windows are section images of the same image data with 3 different dimensions: X-Y, Y-Z, and X-Z, which are controlled by buttons below the windows, so that users can switch between section images of different dimensions.

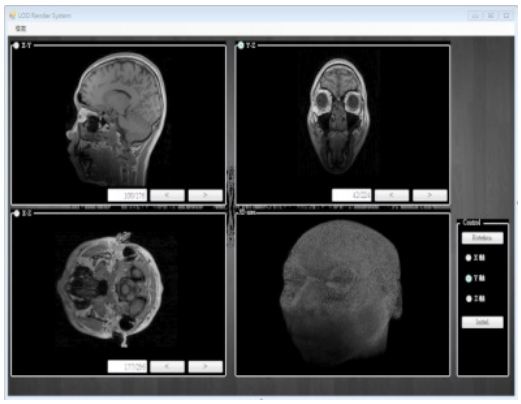

Figure 10 User Interface

In the 3D image display window, images can be controlled by the mouse to zoom-in, zoom-out and rotate. Under Rotation control, a user can choose to rotate the 3D model on each of the axis clock-wise; while the Initial control enables a user to return the model back to the original location in the window.

IV. System Implementation Results

There are two steps in the analysis process to enhance the system performance. One is the enhancement of pre-process performance. The other is the improvement of real-time rendering performance.

A. Edge Detection Performance and Enhancement

In the edge detection phase, all image data must first go through edge detection operations. Logical operations will then be performed on graphic edges resulting from the detection operation and the original data to eliminate noise in the images.

Because the edge detection computation is handled by a CPU, the operation speed is determined by the clock rate of the CPU. As such, the difference in the operation time between experimental group (A) and experimental group (B) is very small; while the clock rate of the CPU in experimental group (C) is only 2.00 GHz, which differs significantly from the 3.40 GHz of experimental group (A) and (B) and the 3.00GHz of experimental group (D), thus taking more time. To further enhance operating speed and reduce the dependence on the CPU clock rate, a multi-thread application is added to the system. Although a total number of 8 multi-threads are added to the experimental groups (A), (B) and (C), the CPUs in the operations are quad-core, and there is a delay in the synchronization between the parameters used and data access, thus, the performance using parallel computation was only enhanced by about 4 times, as shown in Figure 11-13.

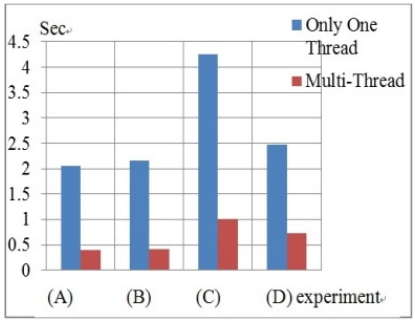

Figure 11 Edge Detection Results of Test Data No. 1 Using Various Testing Platforms

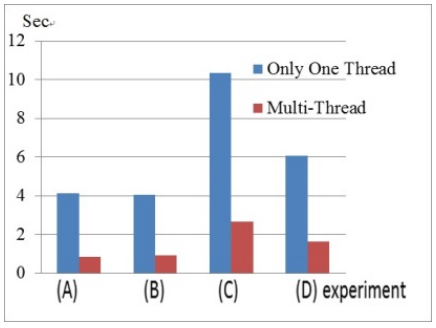

Figure 12 Edge Detection Results of Test Data No. 2 Using Various Testing Platforms

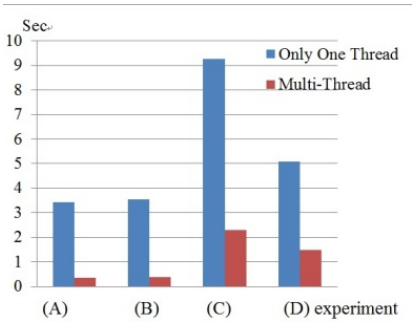

Figure 13 Edge Detection Results of Test Data No. 3 Using Various Testing Platforms

B. Performance of Data Construction

Data construction includes 2D image and spatial coordinate conversion, and the retrieval of color value and transparency image data, as well as, using the spatial coordinate locations of the vertices to construct the triangulation connectivity index of the neighboring vertices.

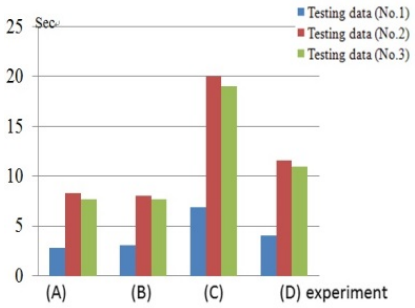

Figure 14 Constructing Valid Data Performance with Various Testing Platforms

Along with the data that must be passed by the CPU to memories or specific registers for GPU processing, data for the construction phase is also processed by the CPU because, each vertex of the image data has continuity, which is then used to build triangulation indices. However, after adding multi-thread for speed, the discontinuity and irregularity of vertex data will add difficulty and complexity to the construction of triangulation indices, resulting in more computation and time spent. Therefore, multi-thread parallel processing is not used in this phase for speed, as shown in Figure 14.

C. Data Volume Simplification

Prior to performing image rendering, the system will perform a stress test on each platform, it's approach is to run a rendering operation with a set of data preset by the system, then record the time required for the system to complete the execution 30 times, and then perform preset data volume reduction until the computer platform can perform rendering at 30 times per second; i.e., testing the data volume of each frame that the system can handle at 30 FPS (Frames per second). The preset model data of the system contains 1.5 million vertices and a spherical model with 3 million triangles. The stress test results of each platform are shown in Figure 15.

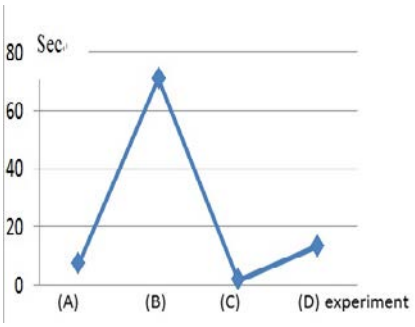

Figure 15 Requiring time for Each Platform to Render 30 Frames

The required time for experimental group (B) is significantly higher than other platforms. This shows that, even with the higher CPU clock rate, the performance of simulating a graphic operation is still comparably different than the performance of a low level GPU. The performance difference ranges from 5 to 40 times.

As for the performance differences between experimental groups (A), (C), and (D): the GPU of experimental group (A) is a professional grade GPU, but there is no advantage in handling data volume; while the performance differences between experimental group (C) and (D) are caused by the differences of GPU clock rate and memory space.

The system uses the times required to stress test each testing platform as the parameters to calculate maximum data volume that can be handled by each platform for image reconstruction, the data volume of which is close to the resulting data volume after partial simplification. Figure 16 shows the simplification results of the testing data on each testing platform.

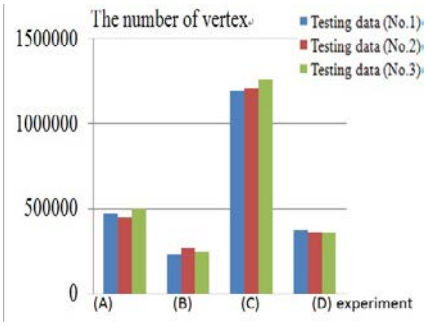

Figure 16 Data Volume of Each Testing Data after Partial Simplification

The reason that there is a difference between the projected data volume standard observed through stress test, and the simplified data volume of testing data No. 2, and testing data No. 3 of experimental group (B) are because, if the data simplification ratio is great, the precision of model presentation will be significantly decimated. Thus, the simplification process stops after reaching a certain ratio, thereby, the interactivities of testing data No. 2 and No. 3 of experimental group (B) are poor.

This research performs accuracy comparison on the image generated with image reconstruction after image data is simplified and the image generated without data simplification. The comparison method is to compare visual errors of each axial plane (XY, YZ, XZ) of the model. The comparison results are shown in Figure 17 below:

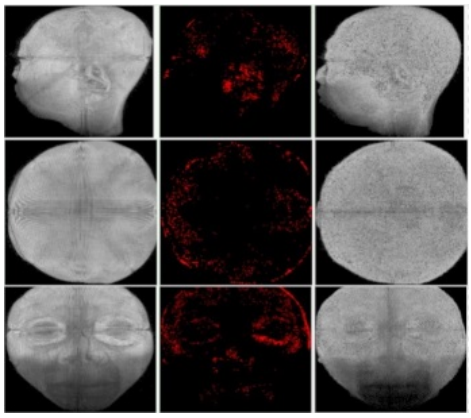

Figure 17 Accuracy comparison of Testing Data No. 1 After Being Processed Nearly 10 Times with Data Simplification

Testing data No.1, after being processed nearly 10 times with data simplification, is beginning to generate visual errors on the model. Although testing data for No. 2 and No. 3 files have higher resolution, the models generated by the data with 20 time and 10 time simplifications are beginning to show errors, as shown in Figure 18-19.

In the comparison diagram we find that a high ratio of data volume simplification can significantly reduce system computation volume, while maintaining features of overall appearance of the original model; however, the detail presentation of the model is beginning to show errors, and the color of each vertex is losing its accuracy, resulting from the significant data simplification.

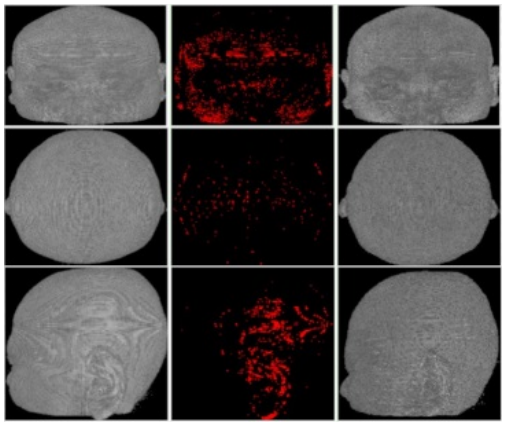

Figure 18 Accuracy Comparison of Testing Data No. 2 after Nearly 20 Times of Data Simplification

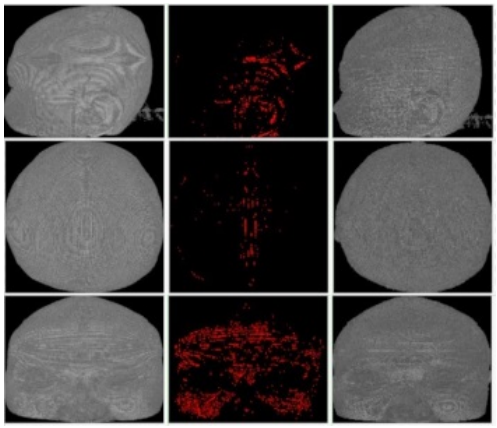

Figure 19 Accuracy Comparison of Testing Data No. 2 after Nearly 10 Times of Data Simplification

V. Conclusion

To obtain rapid 3D imaging reconstruction, this research explores various hardware platforms with their potential application on this function in mind. With the goal of implementing the LOD algorithm, this research performs a study of GPU computation capabilities on various platforms. Subsequently, this research completed algorithm implementation step by step, performed analysis to further understand its effect, and used multi-thread methods to enhance data preprocessing efficiency.

In this research, under the premise of disregarding image fidelity while pursuing operating fluency, testing data of the system can be simplified for 10 to 20 times without affecting the appearance of the model, while maintaining good operability. Under the premise of maintaining model scope fidelity, the model simplification level can reach 5 to 10 times. While its operability may be reduced, its FPS rate is still maintained at between 10 to 25 simplifications.

As for the model simplification of using a CPU for graphic simulation, even with higher clock rates, its

operability is significantly reduced when image fidelity is taken into consideration, which shows that it is less suitable to be applied in the medical or its related fields. As for a CPU and GPU integrated platform, the clock rate speed of the CPU affects data preprocessing time, while the construction and presentation of the model is affected by GPU clock rate and its specification. The results of this study show: if CPU clock rate is higher than 2.0 GHz, and there are 4 or more cores, data preprocessing time can be reduced to less than 30 seconds; and if GPU clock rate is higher than 1.5 GHz, and it has at least 96 graphic cores, there will be good fidelity in image reconstruction.

The edge detection part, the parameters used in the system were adjusted according to the data source, if it is to be applied to varying data, adjustments must be made, especially for MRI grayscale imaging, enable the system's general usage and enhance its precision.

A large amount of data volume is required to process 3D image construction, which is why the entire computation process consumes so much time. Strictly speaking, current systems can only be operated off-line. In the future, new algorithms can be developed to reduce the preprocessing time, so that the entire computation process can truly achieve computation in real time.

REFERENCE

[1] N. E. Nahi C. A. Franco “Recursive Image Enhancement-Vector Processing” IEEE Transactions On Communications, Vol. 21, No.4, April 1973., pp.305-311.

[2] Talla, D.; John, L.K.; Lapinskii, V.; Evans, B.L. “Evaluating signal processing and multimedia applications on SIMD, VLIW and superscalar architectures” IEEE International Conference on Computer Design, pp. 163-172,2000.

[3] Qi Zhang, Roy Eagleson, and Terry M. Peters ,“GPU-Based Image Manipulation And Enhancement Techniques For Dynamic Volumetric Medical Image Visualization”IEEE International Symposium,April 2007, pp. 1168-1171.

[4] Wenfeng Shen, Daming Wei, WeiminXu, Xin Zhu, Shizhong Yuan, “Parallelized computation for computer simulation of electrocardiograms using personal computers with multi-core CPU and general-purpose GPU”Computer Methods And Programs In Biomedicine,June 2010 , pp.87-96.

[5] Hajeer M Y, Millett D T, Ayoub A F, et al. Current products and practices applications of 3D imaging in orthodontics: part I[J]. Journal of orthodontics, 2004, 31(1): 62-70.

[6] Tymofiyeva O, Proff P C, Rottner K, et al. Diagnosis of Dental Abnormalities in Children Using 3-Dimensional Magnetic Resonance Imaging[J]. Journal of Oral and Maxillofacial Surgery, 2013.

[7] Ronghua Liang; Zhigeng Pan; Krokos, M.; Mao Chen; JiaruiBao; Cheng Li “Fast Hardware-Accelerated Volume Rendering of CT Scans”,Journal of Display Technology, volume: 4, pp. 431 - 436,2008.

[8] Cromer M S, Foster S L, Bourne R M, et al. Use of 3T MRI and an unspoiled 3D fast gradient echo sequence for porcine knee cartilage volumetry: Preliminary findings[J]. Journal of Magnetic Resonance Imaging, 2012

[9] Bloecker K, Wirth W, Hunter D J, et al. Contribution of regional 3D meniscus and cartilage morphometry by MRI to joint space width in fixed flexion knee radiography—A between-knee comparison in subjects with unilateral joint space narrowing[J]. European journal of radiology, 2013.

[10] Luebke, D.: Level of detail for 3D graphics, vol. 1. Elsevier Science Inc. (2002)

[11] Liang Hu, Pedro V. Sander, and Hugues Hoppe, “Parallel View-Dependent Level-of-Detail Control” IEEE Transactions On Visualization And Computer Graphics, Vol. 16, No. 5, September/October 2010.

[12] Reinhard K. Multiresolution representations for surfaces meshes based on the vertex decimation method[J]. Computers & Graphics, 1998, 22(1): 13-26.

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

[14] Low K L, Tan T S. Model simplification using vertex-clustering[C]//Proceedings of the 1997 symposium on Interactive 3D graphics. ACM, 1997: 75-ff.

[15] JorikBlaas, Charl P. Botha, H. Post “Extensions of Parallel Coordinates for Interactive Exploration of Large Multi-Timepoint Data Sets”IEEE Transactions On Visualization And Computer Graphics, Vol. 14, No. 6, November/December 2008.

[16] Fang, T. P. and L. A. Piegl, “Delaunay Triangulation in Three Dimensions”, IEEE Computer Graphics and Applications, volume 15, No. 5, 1995, pp. 62-69,

[17] Matthew D. Jones, Rutao Yao, “Parallel Programming for OSEM Reconstruction with MPI, OpenMP, and Hybrid MPI-OpenMP” IEEE Nuclear Science Symposium Conference Record, Vol. 5, 2004 , pp.3036-3042.

Tsung Yu, Lee was born in Chang-Hua, Taiwan, in March 1990. He received the bachelor degree in department of electronic engineering from National Taipei University of Technology, Taipei, Taiwan, in 2012. Since September 2012, he has been working toward to the Master degree at graduate institute of computer and communication engineering in National Taipei University of Technology, Taipei, Taiwan.

Currently, his main research interest domain is computer science and graphical image processing of medical electronic engineering, now his current research project is about relationship between disease feature with ECG and Oxygen concentration on clinical.

Mr. Lee is a student member of Global Science and Technology Forum.

Ren-Guey Lee was born in 1965. He received the M.S. degree from Department of Electrical Engineering, National Chen Kung University (NCKU), Taiwan, in 1989, and the Ph.D. degree from Department of Electrical Engineering, National Taiwan University (NTU), in 2000.

Since 2002, he has been with the Department of Electronic Engineering, National Taipei University of Technology, where he is currently a full Professor. His research interests include medical informatics, tele-care and mobile care system, wearable technology and wireless sensor network for biomedical applications.

Sheng-Chung, Tien was born in Taiwan, in September 1978. He received the Master degree in department of electronic engineering from National Taipei University of Technology, Taipei, Taiwan, in 2008. Since September 2009, he has been working toward the doctor degree at graduate institute of computer and communication engineering in National Taipei University of Technology, Taipei, Taiwan.

His main research interest domain is embedded system devices software integration and graphical image processing of medical electronic engineering. Now, he focuses on android system integration and ECG devices design.

Robert Lin was born in Chang-Hua, Taiwan, ROC, in 1957. He is currently a Associate Professor in Electrical Engineering Department of Lunghwa University of Science and Technology. He received the M.S. degree in electronic engineering from Chung Yuan Christian University, Taiwan, ROC, in 2001. He received the Ph. D degree in Bio-industrial Mechatronics Engineering Department of National Taiwan University. At present, his interested research includes mix-mode IC design of VLSI, embedded systems used in electrical engineering, and the application of medical engineering.

Wei-Hua, Su was born in Taipei, Taiwan, in February 1987. He received the bachelor degree in department of computer science from Tamkang University, New Taipei City, Taiwan, in 2010. Since September 2012, he has been working toward the Master degree at graduate institute of computer and communication engineering in National Taipei University of Technology, Taipei, Taiwan.

His main research interest domain is parallel computing and 3-dimensions rendering, now his current research project is medical graphic reconstruction and visualization.