Key Technology Research on Image Display and Processing of 3D Data Scenes

LiZhong WANG[1, a], Bin GUO[2], XiaoBo LU [3], Yumin LIU[4], ARIGELE [5]

1,2,3,4,5 Inner Mongolia Agricultural Univerisity
Hohhot city 010018 China
[a]anmgnydxwlz@icloud.com

***Abstract* —This paper explores graphic image display and processing for 3-D data visualization, and includes the most commonly encountered techniques in practice. Volume rendering is an important class visualization technology and, compared with the geometric surface extraction method of visualizing data from the field, it does not generate intermediate geometric elements, but goes directly to the data overall synthesis. Thus, different content can be displayed simultaneously in one image, to make it convenient for people to carry out comprehensive and integrated field data analysis and research. More obscure weaknesses of imaging volume rendering method are its slow speed and difficulties to express image information. This paper proposes some effective ways to develop volume rendering technology to better visualize and improve people's ability to analyze and understand the data.**

Keywords: 3D data scene; Volume rendering; Image display; Image processing

I. INTRODUCTION

Today, the rapid development of science and technology, meteorology, medicine, molecular biology and human activities all generate a lot of data. Thus, these data by visually effective analysis and processing, is a very important science, the science is scientific visualization. Scientific visualization through a strong visual effect, reveal the nature of things and variation plays an increasingly important role, it greatly improves the processing speed and quality of scientific computing data, modernized scientific computing tools and environments. Three-dimensional data field graphic image display and processing was developed in the late 1980s and to propose a new research area of rapid development. It refers to the use of computer graphics and image processing technology, scientific computing and data during the calculation result is converted to graphics and images displayed on the screen and the theory, methods and techniques of interactive processing [1-2]. It is widely used in meteorology, petroleum prospecting, medicine, molecular biology, computing fluid dynamics, finite element analysis. In the scientific visualization of many problems in the field of data visualization is the core, and the resulting data for scientific computing, engineering calculation result data and measurement data are usually three-dimensional data field. Thus, three-dimensional data fields and a graphic image display processing is the core of visualization in scientific computing problems.

In the three-dimensional data visualization of both approaches, volume rendering algorithm is a direct result of the two-dimensional image on the screen by the three-dimensional data field. This method is a three-dimensional data visualization method developed rapidly in recent years. Commonly used first by the three-dimensional structure of a data field and then the intermediate geometry from traditional computer graphics technology isosurface (line) drawing screen drawing algorithm, the image produced by this method can reflect the entire original data field the whole picture and the details, with high image quality, ease of parallel processing, etc., which is a very viable method for rendering 3D data [3]. However, this method of calculating the amount of large, long time to calculate, it is difficult to meet the needs of real-time rendering, especially for real-time rendering of three dimensional data field is a challenging research topic in the middle and low workstation and computer. Volume rendering is the essence of technology and resampling image synthesis. Resampling is to 3D data into two-dimensional discrete signal, image synthesis is the synthesis of all contributions to the two-dimensional image data values. The reason why it is difficult to achieve volume rendering real-time rendering, the main reason is that after the change when the viewing direction, the context data field between sampling points corresponding change occurred calculate resampling and the need to re-image synthesis, therefore, how improve the three-dimensional discrete data resampling rate has become a key issue for real-time volume rendering.

II. RELATED KEY TECHNOLOGIES

3D data visualization process that is based on a series of two-dimensional or three-dimensional volume data slice data, reconstruct three-dimensional solid object models with realistic effects, computer graphic image displayed in a way to provide interactive tools, in some cases for modeling and simulation applied research. 3D data visualization in scientific computing visualization as the core content, has its own number of key technologies and important content, we need researchers and numerous research efforts to explore and discover [4-5]. 3D data visualization from scientific visualization angle is defined as visualization tasks in a knowledge-dimensional data field, by many on this direction of research theory and application of technology analysis, the three-dimensional data visualization technology mainly includes two categories of key technology, that is, data analysis and processing and three-dimensional visualization [6]. In the three-dimensional data visualization process, data visualization is based, among which three-dimensional volume data based, involving a variety of data format conversion and processing technology is an important method of ensuring the realization of three-dimensional visualization of three-dimensional reconstruction excellent effect provided certain data structures and calculation rules, volume data to build mathematical models and three-dimensional engine library, on a computer screen showing a 3D solid object graphic images with realistic effects, and provide different levels of interactive operation [7].

Three-dimensional reconstruction of the surface mesh. Now for different data fields it has been proposed a variety of three-dimensional surface reconstruction algorithm. On the one hand the three-dimensional data field its own characteristics, you can choose different surface reconstruction algorithm; the other hand, different surfaces can be improved reconstruction algorithm applied to different data field types. Organization of this section in accordance with the characteristics of 3D point cloud classification describes the common surface reconstruction algorithm.

Isosurface surface regular data field reconstruction. Three-dimensional data field rules can be viewed as many hexahedral (or cube) unit, which is composed of voxels. By calculating the equivalent surface of each voxel and connected surface can be reconstructed three-dimensional object.

(1) Cuberille method. Cuberille method is suitable for intensive uniform grid data field. This method simply that the six faces of the hexahedron are isosurface, delete overlapping surfaces hexahedral but only the non-overlapping connecting surfaces can be approximated Isosurface three-dimensional object. The method is simple, easy to parallel processing; the main problem is to show the results of giving a "massive" feeling, especially not a good indication of the details of the object.

(2) Marching Cubes (MCs) method. In the three-dimensional structure of a regular grid data field isosurface method is the most representative Marching Cubes (Marching Cubes, MC). Each corner cube has a small space function value 0 or 1, the function value isosurface between [0,1]. Showing the comparison function value interpolation cube edges and vertices of the isosurface isosurface intersection with the edge of the cube is generated according to certain connected triangles as isosurface within the cube approximation.

(3) Dividing Cubes method. When the pixel density is so high that the three-dimensional data field Marching Cubes method of generating triangular piece with equal or more hours of screen, a small triangle by interpolation calculation is unnecessary. The basic idea is that when the cube contains Isosurface and projected on a screen pixel size is greater than the cube subdivided repeated testing and cube isosurface isosurface subdivision until the child cube contains the screen the projection is less than equal to the pixel size until the last cube containing the iso-surface is projected onto the screen as an approximate representation of the isosurface.

Scattered surface mesh data field reconstruction. D Scattered Data field is the most widely used 3D data, there have been a variety of surface reconstruction method, which is the most widely used Delaunay triangulation method based on grid reconstruction. Can prove that 2D Delaunay triangulation meet the minimum criteria to maximize interior angle, three-dimensional Delaunay triangulation meet external air balls guidelines. Based on Delaunay triangulation grid reconstruction can be divided into two direct projection based on triangulation Triangulation and space types.

Surface mesh semi-structured data field reconstruction. Typical of semi-structured data field is contour data field consists of a sequence of two-dimensional contour line consisting of, for example medical CT or MRI image slice distance between the two is much greater than the resolution of the image can be seen as a contour line data field, the data line structured light three-dimensional laser scanner is obtained contour data field.

III. PERSPECTIVE GRID FOR IMAGE DISPLAY AND PROCESSING

The perspective grid is used to resample the particle data inside the view volume. It is a structured grid that partitions the view frustum into $k \times l \times m$ oblique sub-frusta. Figure 1(right) illustrates this grid, which has the specific property that the sampling rate along z is the same as along x and y. The perspective grid is stored on the GPU in a 3D texture map (see Figure 1(left)). In the resampling process, the 3D texture map is used as a render target and resampled quantities are scattered to this target using accumulative blend. To perform the resampling efficiently on the GPU, a mapping of a point $r = (x, y, z)$ from 3D texture space to a point $r' = (x', y', z')$ in view space and vice versa is required [8-10]. In the following, we will derive this mapping and describe how to exploit it in the resampling process.

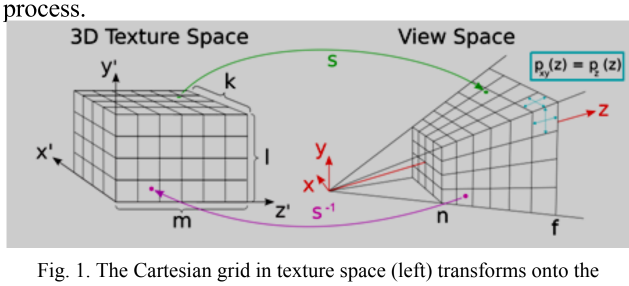

Fig. 1. The Cartesian grid in texture space (left) transforms onto the perspective grid in view space (right) under the mapping s

Perspective Grid Mapping. Let us denote by $s(r') = (s_x(x', z'), s_y(y', z'), s_z(z'))$ the mapping we are looking for, and let us note that the sampling

DOI 10.5013/IJSSST.a.17.18.12

ISSN: 1473-804x online, 1473-8031 print

rate $p_{xy}(z)$ along x and y is given by $(2 \cdot \tan(fov_y/2)/res_y) \cdot z$, where fov_y is the vertical field of view and res_y the vertical resolution of the viewport. The mapping of x and y in texture coordinates to x and y in view space is simply the inverse view-space to screen-space transformation:

$$s_x(x', z') = \left(\frac{2 \cdot x'}{k} - 1\right) \cdot s_z(z') \cdot \tan(fov_y/2) \quad (1)$$

and for $s_y(y', z')$, respectively. The mapping in texture space to $z' \in [0, m]$ in view space $z \in [n, f]$ are the near and far plane) needs some further explanations.

We observe that the sampling rate along z, $p_z(z)$, is equal to $\frac{ds_z(z')}{dz'}$, since z' is sampled uniformly with distance 1. Since $p_z(z)$ is supposed to be equal to $p_{xy}(z)$, we obtain:

$$p_z(z) = \sigma \cdot s_z(z') = \frac{ds_z(z')}{dz'} \quad (2)$$

Where $\sigma = \frac{2 \cdot \tan(fov_y/2)}{res_y}$ and z has been replaced by $s_z(z')$ in the second term. This differential equation is solved by any $s_z(z')$ of the form

$$s_z(z') = a \cdot b^{c \cdot z'/m} \quad (3)$$

Since $z' \in [0, m]$ is mapped to $z \in [n, f]$, and by setting $c = 1$, $s_z(z')$ equates to

$$s_z^{-1}(z) = m \cdot \frac{\ln(z/n)}{\ln(f/n)} \quad (4)$$

It's inverse is

$$s_z^{-1}(z) = m \cdot \frac{\ln(z/n)}{\ln(f/n)} \quad (5)$$

Note that this corresponds to the transformation to equally distribute the aliasing error in perspective shadow map parameterizations. This mapping is only correct, however, on the z axis. On every other view ray the sampling rate is larger, since all of them perform the same z sampling in world space. In the extreme case, which is along the edges of the frustum, the sampling rate scales by the factor

$$\lambda = \sqrt{\left(\frac{k \cdot \sigma}{2}\right)^2 + \left(\frac{l \cdot \sigma}{2}\right)^2 + 1} \quad (6)$$

We must further determine the number of sample points m along the $z-axis$ for a given range $[n \ldots f]$. By inserting equation 2 into 1 and solving for m, we obtain

$$m = \frac{\ln \frac{f}{n}}{\sigma} \quad (7)$$

Finally, to account for the increasing sampling distance towards the frustum boundaries, and thus to provide the appropriate sampling in the whole frustum, m must be scaled by λ.

IV. DATA RESAMPLING OF THE 3D DATA SCENE FOR IMAGE PROCESSING

To resample the particle data to the perspective grid, for every particle the grid vertices within the support of the particle's smoothing kernel have to be determined and the data is interpolated according to the kernel function:

$$A(\mathbf{r}) = m_j \cdot \frac{A_j}{\rho_j} \cdot W(|\mathbf{r} - \mathbf{r}_j|, h_j) \quad (8)$$

Here, $A(r)$ is the resampled data at position r, and m_j, p_j, r_j, and A_j are the particle's mass, density, position, and data value, respectively. W is the kernel function with a support h_j that can vary from particle to particle. We describe the particular kernel functions underlying the smoothed particle hydrodynamics simulations used in this work. The interpolated value A_j is added to the corresponding telex in the 3D texture map. Key to an efficient and high quality resampling of large particle sets is the use of a multi-resolution particle representation. In such a representation the particle set is encoded at different levels of detail by merging subsets consisting of smaller particles into one larger particle [11]. The hierarchical particle representation allows pruning particles that are too small to be reconstructed at the required sampling rate. Thus, aliasing artifacts can be avoided and the number of particles to be processed can be reduced.

Hierarchical Particle Representation. Our pre-computed multi-resolution particle representation is organized in an adaptive octree data structure. In particular, for large particle sets and high spatial resolution of the simulation we employ the same regular domain partition to allow for the construction of the particle hierarchy in parts. In addition, for the Millenium gas dynamics simulation we scale each particle component logarithmically and compress the data using vector quantization. Spherical pre-fetching regions are realized on the GPU and the CPU to exploit frame-to-frame coherence and thus reduce memory access latencies. In contrast, however, we use different rules for merging particles.

Starting with a uniform grid at the resolution at which the simulation has been performed, particles in contiguous blocks of 2 3 cells are merged into a single particle. The volume of this particle is the sum of the volumes of the merged ones, and the scalar quantities of the merged particles are averaged into this particle according to their mass contribution. The merging process is illustrated in Figure 2. This process is recursively repeated until a user-given resolution level is reached.

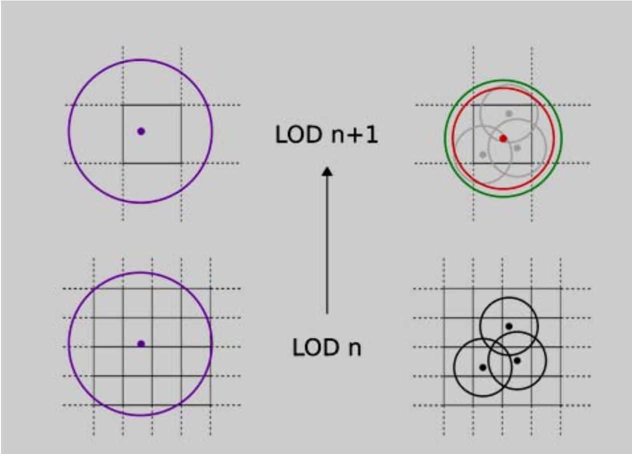

Figure 2. Bottom-up construction of the particle LOD-hierarchy

In the merging process, at a particular level only particles with a radius less than the cell size on the next coarser level are merged. If the radius of a new particle falls below this size, it is increased to this size and its density is decreased proportionally to reflect the volume increase. All other particles are copied to the coarser level to guarantee a consistent representation of the particle field at all levels of detail.

V. PROPOSED VOLUMETRIC DISPLAY AND STEREOSCOPIC RENDERING

The introduction of modern X-ray multi-detector computed tomography (MDCT), magnetic resonance imaging (MRI), and positron emission tomography (PET), has enabled the acquisition of highly detailed medical images. However, the large amount of data that comprises these images creates a challenge for their effective display and interpretation. Real-time stereoscopic volume rendering can address many of these limitations, and provide support for a variety of medical applications, such as diagnosis, radiation therapy planning and guidance, surgical planning, reconstructive surgery, medical education and research. In this section, we briefly introduce some medical applications of monoscopic and stereoscopic volumetric image display techniques.

Rendering methods. Direct volume rendering (DVR) is a major technique for 3D medical data display, creating images of an entire dataset without explicitly extracting surfaces corresponding to features of inter- est. DVR is therefore a promising technique for visualizing complex anatomical structures within a 3D dataset, eliminating many of the disadvantages of MPR, SR and MIP image rendering methods.

Stereoscopic rendering. Stereoscopic visualization adds depth to the perceived image by creating a binocular view of an object, which enables observers to more easily understand complex vessel structures, providing them with more efficient assessment of diseases such as vascular aneurysms, stenotic narrowing of vessels, and vascular malformations. Stereoscopic display is a compelling adjunct

Modern graphics hardware is designed to operate on large continuous streams of vertex and fragment data. A typical graphics pipeline is described in Figure 3: vertices (points in 3D space) are first transformed and assembled into primitives, such as triangles, polygons, rectangles, etc., by the vertex units. These primitives are then rasterized into geometric fragments, composed of a large stream of elements. Texture mapping is performed as a part of the rasterization procedure, during which the texture image is interrogated for the correct color to be added to the fragments. These fragments are further processed by the fragment units, and then written to the frame buffer, which is copied into the graphics card video memory to produce images on the screen.

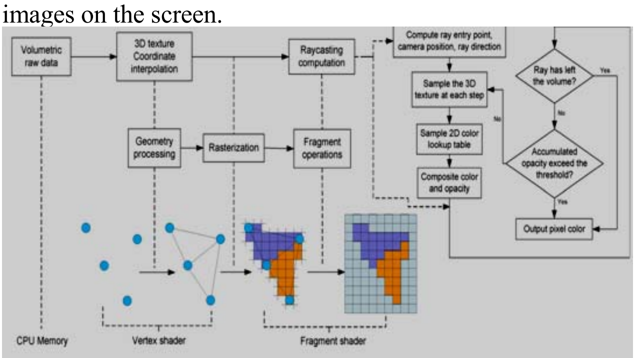

Figure 3. Schematic description of the programmable hardware graphics pipeline and the GPU-based volume algorithm.

In older generation graphics pipelines, the geometry sent to the GPUs was static, and the operations that could be performed on fragments were limited. GPU functions were therefore restricted to certain fixed pipelines. The new generation of graphics cards has evolved to a level where the vertex and fragment units are now user programmable by developing small programs called vertex shaders and fragment shaders, which permit direct control of the vertex and fragment processors during the graphics pipeline. Modern GPUs support the Shader Model 4.0 features, such as dynamic branching, multiple rendering targets, and computation of texture coordinates per pixel, etc. A further improvement in the new GPUs is the increase in pixel depth from 32 bits per pixel to 128 bits per pixel, which means that each red, green, blue, and alpha component can now have 32-bit floating point precision throughout the graphics pipeline. The increased data precision combined with the enhanced programmability means that complicated scientific visualization and computation algorithms can now be implemented directly on GPUs with much greater efficiency, accuracy, and realistic visual effects than ever before.

VI. CASE STUDY

Once the particle quantities have been resampled to the perspective grid, the data can be rendered in turn on the GPU using texture-based volume ray-casting. This enables using different rendering options like isosurface rendering or direct volume rendering simultaneously at very high speed. The major difference to classical texture-based ray-casting is in the kind of grid that is rendered. Usually, the data is given on a Cartesian grid in world space and has to be interpolated tri-linearly at the sample points along the rays. The data in the perspective grid, on the other hand, is already at the positions in view space where a sample is placed during ray-casting. Consequently, the data values in the 3D texture map that stores the perspective grid can be accumulated directly in front-to-back order along the z coordinate axis in texture space. To account for the varying sampling distances, opacity correction of the samples has to be performed. For simulating local illumination effects we compute the gradient of the resampled particle quantity. A gradient's x and y components can be approximated directly via central differences along x and y, respectively.

Example applications is shown in Figure4. 3D texture mapping hardware on the GPU supports tri-linear interpolation, which is not exact in our scenario due to the frustum-shaped cells underlying the texture grid. In order to improve the interpolation accuracy we have implemented distance-based interpolation in texture space.

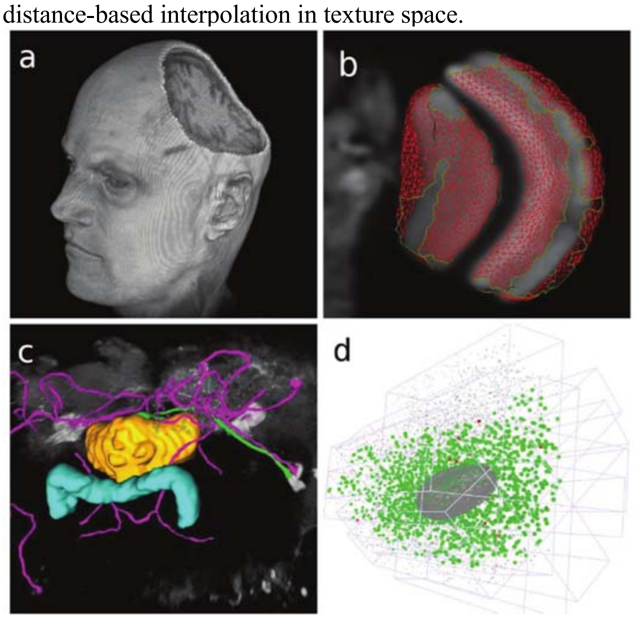

Figure 4. Example applications.

The 3D scene is a virtual 3D space in which any number of entities are hosted. These entities are volume renderings, isosurfaces and orthoslice sets. Volume renderings are representations of voxel data viewed from arbitrary angles, with transparency determined by the intensity values. Isosurfaces are meshes of triangles described by a list of vertices. Orthoslices are orthogonal planes that cut through an image volume. Each of these entities, or objects, has a number of properties, such as transparency, color, and a unique name. The 3D scene that hosts all objects has the following properties: a zooming level, an origin of viewer coordinates, and a scene background. For efficient rendering in computer graphics, we chose Java 3D: a low-level hardware-accelerated software library. Java 3D has the further advantage of being implemented for Java, thus enabling Java applications to interoperate with the graphics card of a computer, via either Open GL or Direct X low-level native layer. Java 3D provides a fine-grained representation of a virtual scene as a directed acyclic graph. Operations on any node of the graph affect its entire subtree. In practice, this means that, for example, zooming in and out of a scene is expressed as a scaling transformation in a low-order node. High-order nodes encapsulate images and meshes. A key feature of our library is to substantially simplify the usage of Java 3D. We define our 3D scene in terms of Java 3D nodes. We provide straightforward means to instantiate a new interactive 3D scene and to add objects to it. In the following, we describe the structure of the scene graph as implemented in our library.

VII. CONCLUSION

Image display and processing for 3d data scene is the core of visualization in scientific computing problems. In the three-dimensional data image display and processing of both approaches, volume rendering algorithm is a direct result of the two-dimensional image on the screen by the three-dimensional data field. This method is a three-dimensional data visualization method developed rapidly in recent years. In this paper, volume rendering in-depth study, with the commonly used first by the three-dimensional structure of a data field and then the intermediate geometry from traditional computer graphics technology isosurface (line) drawing screen drawing algorithm, this method of generating an image can reflect the whole picture and details of the entire original data field, with high image quality, ease of parallel processing advantages. In the field of three-dimensional graphic image display and data during processing, graphical data display and basic image processing, this paper focuses on a variety of format conversion and processing techniques for three-dimensional volume data is to achieve excellent effect of three-dimensional reconstruction of an important guarantee for three-dimensional visualization methods It provided some data structures and calculation rules, volume data constructing mathematical models and three-dimensional engine library, on a computer screen showing a 3D solid object graphic images with realistic effects, and provide different levels of interactive operation. Finally, the case shows that the effectiveness of the intuitive approach.

ACKNOWLEDGEMENTS

Inner Mongolia Natural Science Foundation, Number: 2014MS0616. Scientific research project of Inner Mongolia colleges and Universities, Number: NJZY14091. Inner Mongolia Agricultural University Doctor Foundation.

REFERENCES

[1] Gupta A, Satkin S, Efros A, et al. "From 3d scene geometry to human workspace". Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on. IEEE,pp. 1961-1968, 2011.

[2] Schmid B, Schindelin J, Cardona A, et al. "A high-level 3D visualization API for Java and Image". BMC bioinformatics, vol. 11, no. 1, pp. 274-278, 2010.

[3] Fedorov A, Beichel R, Kalpathy-Cramer J, et al. "3D Slicer as an image computing platform for the Quantitative Imaging Network". Magnetic resonance imaging, vol. 30, no. 9, pp. 1323-1341, 2012.

[4] Crassin C, Neyret F, Lefebvre S, et al. "Gigavoxels: Ray-guided streaming for efficient and detailed voxel rendering", Proceedings of the 2009 symposium on Interactive 3D graphics and games. ACM, pp. 15-22, 2009.

[5] Everts M H, Bekker H, Roerdink J B T M, et al. "Depth-dependent halos: Illustrative rendering of dense line data". Visualization and Computer Graphics, IEEE Transactions on, vol. 15, no. 6, pp. 1299-1306, 2009.

[6] John M, Liao R, Zheng Y, et al. "System to guide transcatheter aortic valve implantations based on interventional C-arm CT imaging", Medical Image Computing and Computer-Assisted Intervention–MICCAI 2010. Springer Berlin Heidelberg, pp.375-382, 2010.

[7] Fraedrich R, Auer S, Westermann R. "Efficient high-quality volume rendering of SPH data". Visualization and Computer Graphics, IEEE Transactions on, vol. 16, no.6, pp.1533-1540, 2010.

[8] Congote J, Segura A, Kabongo L, et al. "Interactive visualization of volumetric data with webgl in real-time", Proceedings of the 16th International Conference on 3D Web Technology. ACM, pp.137-146, 2011.

[9] Swennen G R J, Mollemans W, Schutyser F. "Three-dimensional treatment planning of orthognathic surgery in the era of virtual imaging". Journal of oral and maxillofacial surgery, vol. 67, no.10, pp. 2080-2092, 2009.

[10] Boucheny C, Bonneau G P, Droulez J, et al. "A perceptive evaluation of volume rendering techniques". ACM Transactions on Applied Perception (TAP), vol. 5, no.4, pp.23-28, 2009.

[11] Nießner M, Zollhöfer M, Izadi S, et al. "Real-time 3d reconstruction at scale using voxel hashing". ACM Transactions on Graphics (TOG), vol. 32, no. 6, pp. 169-175, 2013.