MATEC Web of Conferences **267**, 02016 (2019) https://doi.org/10.1051/matecconf/201926702016
*ISC 2018*

# 3D Reconstruction of Images Based on Embedded Processors

Jundi Wang[1,a], Chixin Li[2], Yaling Zhu[1] and Juan Wang[1]

[1]*College of Software Engineering, Lanzhou Institute of Technology, Lanzhou, Gansu, China*
[2]*College of Electronic Information Engineering, Lanzhou Institute of Technology, Lanzhou, Gansu, China*

**Abstract.** 3D image reconstruction has very important value in reality. In this paper, based on embedded system, we use Marching Cubes algorithm to realize 3D image reconstruction and furthermore, display the result in web page. By introducing principle and whole process of reconstruction, we can prove that comparing with personal computer, this method has advantage of lower price, small volume, and has good prospect in future.

## 1 Introduction

3D reconstruction of images is to convert 2D image data obtained from imaging equipment into 3D data, thereby displaying the 3D shapes of object and performing qualitative and quantitative analysis based on visualization technology in scientific computing[1]. The 3D image construction system based on embedded processors utilizes the existing mobile computing platform with strong computing capability to develop a low-cost, portable and easy-to-operate 3D visualization medical system.

This paper introduces a 3D image construction system based on embedded processors on the mobile platform using the ARM company's CORTEX-A8 processor development board, network Web server, and Matching Cubes algorithm. This system enables doctors to observe the generated 3D images on a web browser on a remote computer for the promotion in diagnosis and treatment level.

## 2 Principles of reconstruction

The 3D reconstruction of images studies the sequence of cross-sectional 2D images acquired by various imaging equipment and constructs a 3D geometric model of the object, which is finally drawn and displayed on a computer screen. Currently, there are two main methods for 3D reconstruction of medical images, 3D surface rendering method and 3D volume rendering method.

### 2.1 3D surface rendering method

The basic idea of the surface rendering method is to firstly segment the surface of the object to be displayed in the volume data, and then form the surface of the object by interpolating the geometric unit, and finally display the image by rendering and blanking the model with illumination. In the field of computer graphics, the surface rendering algorithm has moved into a mature stage. There are two specific forms: boundary contour line method and surface method.

The contour line-based method first extracts contour lines from the cross-sectional 2D image by segmentation, and then splices the corresponding contour lines of each layer together to represent the surface boundary of the object of interest. This algorithm is relatively easy -to-operate, but the displayed result is rough and not intuitive. The surface-based method is to reconstruct the surface of the object with the contour. Triangles or polygon facets (or surfaces) are used to form the surface of the object between adjacent boundary contour lines with a specific algorithm. The classic algorithms mainly include: Cuberille method, Marching Cubes method, Marching Tetrahedral method, Dividing Cubes method. What these algorithms have in common is that many small triangular patches are used to fit the equivalent surface of the object. The advantage is that it reduces the amount of calculation and speeds up the operation by adopting the drawing method of traditional graphics and the existing interactive algorithm, graphics hardware and graphics equipment as well.[2] The disadvantage is that some details in the 3D data field may be lost, thus impacting the veracity of the result.

### 2.2 3D volume rendering method

The main idea of the volume rendering method is to assign an opacity to each voxel, letting the light passes through the entire data field and taking the factors such as transmission, emission and reflection into

[a]Corresponding author: Jundi Wang, email:511378759@qq.com

consideration. Each pixel in the 3D image is considered to be a hexahedral element, which is regarded as the voxel. The transmission of light depends on the opacity of the voxel, the emission of light depends on the objectness of the voxel (The higher the objectness, the stronger the emitted light.), and the reflection of the light depends on the relationship of the angle between the plane where the voxel located and the incident light. The volume rendering method includes four steps: projection, blanking, rendering, and integration. It directly studies the relationship between the light passing through the volume data field and the voxel without the intermediate surface.[3] Many details of the voxels are preserved and the veracity of the result is greatly enhanced.

For these two methods, the image quality of the volume rendering method is usually better than the surface rendering method. However, the volume rendering method has high requirements for hardware and is slow in operation, while the surface rendering method has lower requirements for hardware and is suitable to be applied to the embedded mobile platform. This paper chooses the surface rendering method.

## 3 Platform construction

This experiment chooses the TQ210 development board produced by Guangzhou Tianqian Computer Technology Co., Ltd.. The hardware system consists of the core board and back board. The core board is the central component of the computer system, including S5PV210 chip, memory, and Nand Flash memory chip. The back board is responsible for the interaction between external information and the processor, containing a variety of external devices. During the development, the network interface and UART serial port are mainly provided by the back board.

### 3.1 U-boot transplantation

In an embedded system, there is usually no firmware program like the BIOS. The loading and starting tasks of the entire system are completed by the BootLoader. This program initializes the hardware device and creates an appropriate environment for the hardware and software in the system for the final invocation of the operating system. The full name of U-boot is Universal BootLoader, which is a development source code project that complying with the GPL. U-boot can be used to boot multiple operating systems and support various CPU architectures[4].

During the transplantation, firstly set the maximum clock frequency of the S5PV210 processor to 1000MHz by setting the PLL (phase-locked loop) inside the processor. Secondly, complete the memory initialization in U-Boot. Finally, set the memory and network. The external memory of the ARM development board is Nand Flash. In this project, we check the final result on the webpage, so we only need to set the network chip to access the remote file system through NFS (Network File System).

### 3.2 Linux kernel image transplantation

The transplantation of the Linux kernel is used for the specific target platform that only runs properly after the cutting of the installed Linux operating system. In the first phase of the Linux kernel boot, initialize the related code, hardware register, and memory. Then transplant the control code to the kernel, mainly changing the parts related to the architecture. This test uses the Linux 3.10.100 kernel, which provides a graphical interface for users to cut and configure the kernel. After the kernel is configured, run the Make command to start compiling the kernel and finally generate the image file.

### 3.3 Linux file system transplantation

The NFS file system is built using the open source software Busybox. All commands are centralized into an executable file via "plug-ins". In the compilation of Busybox, it is easy to cut (add or delete) its functions, and finally a usable file system is created. The configuration of Busybox is similar to that of the Linux kernel. After the Busybox is configured, run the Make command for compilation.

## 4 3D reconstruction of images with the MC algorithm

### 4.1 Principle of the algorithm

The basic idea of the MC algorithm is to process the cubes (voxels) in the data field one by one, classify the cubes that intersect the isosurfaces, and calculate the intersections of the isosurfaces and the cube edges using the interpolation. Connect these intersections in a certain way to generate an isosurface as an approximation representation of the isosurface in the cube according to the relative position of each vertex of the cube to the isosurface. There is a corresponding scalar value for each vertex of the voxel. When the value at the voxel vertex is greater than or equal to that of the isosurface, it indicates that the vertex is outside the isosurface, which is marked as “0”. When the value at the voxel vertex is less than that of the isosurface, it indicates that the vertex is inside the isosurface, which is marked as "1". Each voxel unit has 8 vertices, and there are $2^{8} = 256$ cases. Figure 1 shows 15 basic cases of the MC algorithm. The other 241 cases can be realized by the rotation and mapping of these 15 basic cases[3].

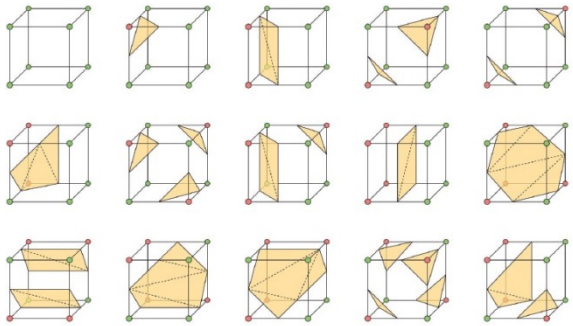

**Figure 1.** 15 basic cases.

The index rules for vertices and edges on each voxel unit are shown in Figure 2. If the value of vertex 3 in the lower part of the voxel is less than that of the isosurface and the values on the other vertices are greater than that of the isosurface, as shown in Figure 3, a triangular patch intersecting the voxel edges 2, 3, 11 can be generated, and the specific position of the vertices of this triangular patch needs to be linearly interpolated based on the value of the isosurface and the edge vertices 3-2, 3-0, and 3-7.

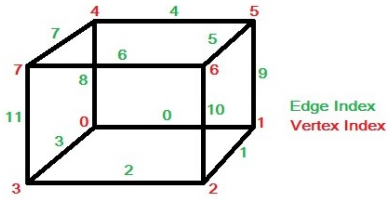


**Figure 2.** The index rules for edges.

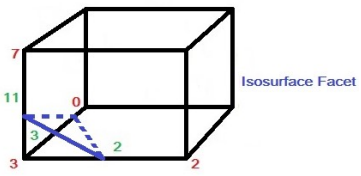


**Figure 3.** Generation of the triangular patch.

For an edge, if its two endpoints v1 and v2 are marked differently, the isosurface must intersect this edge, and the coordinates of the intersection are[5]:

P=P1 + (isovalue - V1 ) (P2 - P1 ) / (V 2 - V 1 )

P represents the coordinate of the equivalence point, P1 and P2 represent the coordinates of the two endpoints, V1 and V2 represent the gray values of the two endpoints, and isovalue represents the threshold or the value of the isosurface.

### 4.2 Steps of the algorithm

(1) Using the MC's triangle table edgeTable to determine which side of the voxel unit intersects the isosurface. The index number definition rule of the voxel unit vertex state as follows.

cubeindex=v7v6v5v4v3v2v1v0

For example, only vertex 3 is marked as "1" and other vertices are marked as "0". Then the voxel unit has a vertex state of 0000 1000 or 8. Obtain edgeTable[8] = 1000 0000 1100 in the table, which means the edges 2, 3, and 11 of the voxel unit intersect with the isosurface, and then calculate the position of each intersection by linear interpolation.

(2) Generating the composition of the corresponding triangle patch with the triTable table

Taking the example shown in Figure 4, check the table to get triTable[8] = {3, 11, 2, -1, -1, -1, -1, -1, -1, -1, -1, - 1, -1, -1, -1, -1}, which means that the vertex state can generate the triangle patch (3, 11, 2), representing the intersections of three vertices of the triangle patch as edges 3, 11, 2 and the isosurfaces.

(3) In the last two steps, the information about the points and surfaces of the isosurface can be obtained. In order to further improve the display effect, it is necessary to adjust the vertex normal.[6] Assuming that the value at vertex (i, j, k) is f(i, j, k), the gradient vector at that point can be calculated using the center difference method. As shown in the following equation:

### 4.3 Implementation of the algorithm

The MC algorithm is firstly implemented on the PC. With the OpenGL tool, generate a sphere with the triangular patch based on the core idea of the MC algorithm. As shown in Figure 4. Press the button to change the size of the isosurface, and the size of the sphere changes as well. After generating the fixed-point normal vector, a 3D sphere is generated under a certain light source with special effect rendering. As shown in Figure 5. The main code is as follows:

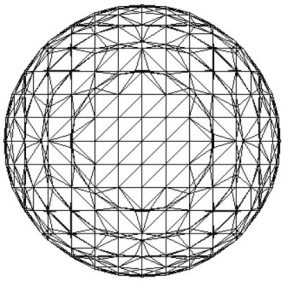

**Figure 4.** MC algorithm generates spherical patch.

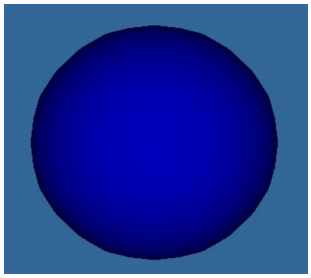

**Figure 5.** MC algorithm generates spherical rendering patch.

## 5 3D graphics reconstruction based on the S5PV210 development platform

Building a 3D image reconstruction system on the S5PV210 embedded platform requires the WebGL (Web Graphics Library) technology. WebGL is a 3D graphics protocol that allows the combination of JavaScript and OpenGL ES 2.0. By adding a JavaScript binding to OpenGL ES 2.0, WebGL can provide hardware 3D accelerated rendering for HTML5 Canvas and a series of JavaScript APIs for developers. These APIs can be used for graphical rendering to obtain 3D WebGL in the web page. The reconstruction process is show in figure 6.

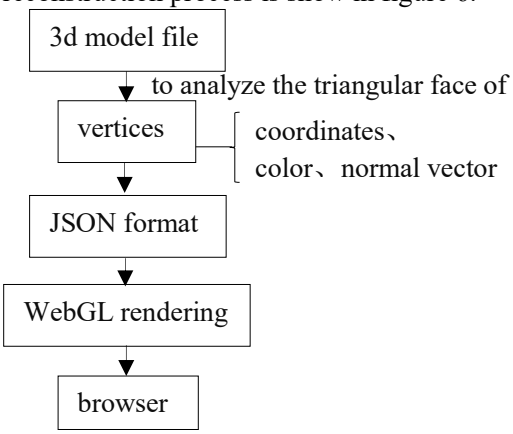


**Figure 6.** The process of the reconstruction.

The final result is shown in the following figure. Figure 7.1 shows the basic 3D image generation based on WebGL. (a), (b), and (c) of Figure 7 are 3D images constructed by setting different isosurface values in the MC algorithm.

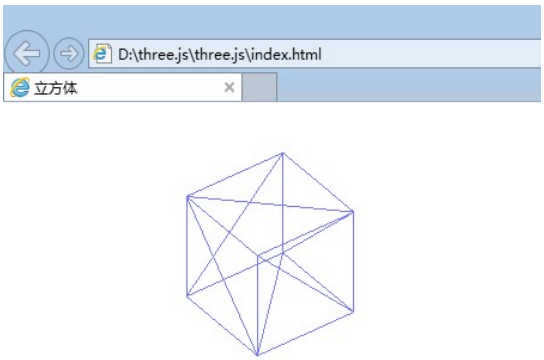


**Figure 7.** Simple 3D image generation patch.

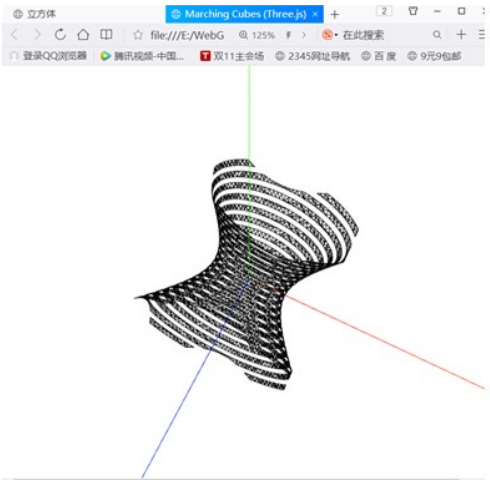


**Figure 7(a).** MC algorithm implementation patch.

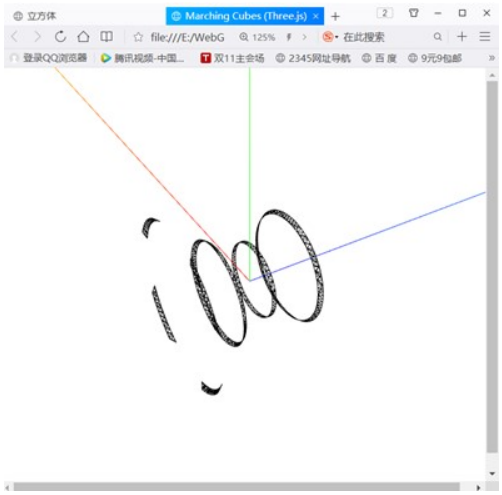


**Figure 7(b).** MC algorithm implementation patch.

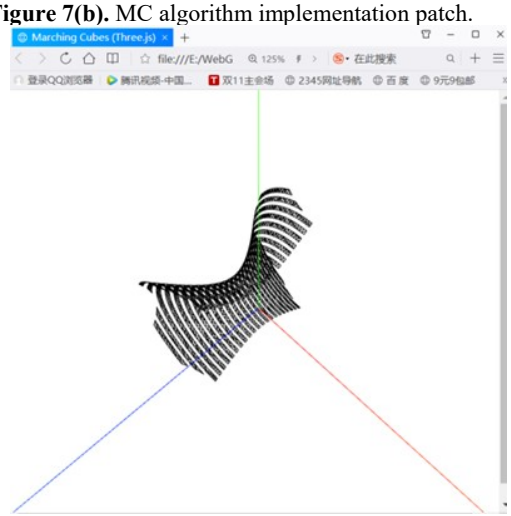


**Figure 7(c).** MC algorithm implementation patch.

## 6 Summary

This paper takes the reconstruction technology of 3D images as the research object and studies the

construction system of 3D images based on embedded processors. It chooses to transplant the Linux file system on the TQ210 development board and transplant the MC algorithm to the embedded system. A low-cost, portable, and easy-to-operate 3D visualization medical system is developed using the existing mobile computing platform with strong computing capability. The effectiveness of the method is proved by testing the real image sequences.

## Acknowledgements

The paper is supported by: (1) National Social Science Fund Project under grant NO. 15XMZ035 (2) Scientific research project of colleges and universities in gansu province under grant NO.2018B-059 (3) Natural Science Fund Project of Gansu province NO. 18JR3RA228 (4) Lanzhou science and technology planning project NO2018-4-56.

## References

1. Xia Liu, et al. Vertebrae CT Images 3D Reconstruction with Improved Marching Cubes Algorithm, Journal of Harbin Institute of Technology[J]: 2018(05)
2. Kai Zhu, et al. Three - Dimensional Reconstruction of Brain Map Based on Improved MC Algorithm [J]. Computer Applications and Software. 2016(04)
3. Minho Chang, et al. Interactive marching cubes algorithm for intraoral scanners. The International Journal of Advanced Manufacturing Technology. 2017(09)
4. Xingxing Xiong, et al. U-boot Analysis and Transplantation Based on S5PV210 [J]. Computer Systems & Applications. 2015(01)
5. Christian Linder, Xiaoxuan Zhang. A marching cubes based failure surface propagation concept for three-dimensional finite elements with non-planar embedded strong discontinuities of higher-order kinematics [J]. Int. J. Numer. Meth. Engng. 2013 (06)
6. Seungki Kim, et al. A Novel Interpolation Scheme for Dual Marching Cubes on Octree Volume Fraction Data. Computer Graphics. 2017(10)