An Marching Cube Algorithm Based on Edge Growth

Xin WANG¹, Su GAO², Monan WANG¹ *, Zhenghua DUAN¹

1. Key Laboratory of Medical Biomechanics and Materials of Heilongjiang Province, Harbin University of Science and Technology, Harbin 15000, China
2. Beijing Normal University Hospital, Beijing 100875, China

* Corresponding author, mnwang@hrbust.edu.cn

Abstract Marching Cube algorithm is currently one of the most popular 3D reconstruction surface rendering algorithms. It forms cube voxels through the input image, and then uses 15 basic topological configurations to extract the isosurfaces in the voxels. Since it processes each cube voxel in a traversal manner, it does not consider the relationship between isosurfaces in adjacent cubes. Due to ambiguity, the final reconstructed model may have holes. We propose a Marching Cube algorithm based on edge growth. The algorithm first extracts seed triangles, then grows the seed triangles and reconstructs the entire 3D model. According to the position of the growth edge, we propose 17 topological configurations with isosurfaces. From the reconstruction results, the algorithm can reconstruct the 3D model well. When only the main contour of the 3D model needs to be organized, the algorithm performs well. In addition, when there are multiple scattered parts in the data, the algorithm can extract only the three-dimensional contours of the parts connected to the seed by setting the region selected by the seed.

Keywords 3D reconstruction, Marching Cube, Seed, Edge growth, Ambiguity

1. Introduction

Three-dimensional reconstruction algorithms are mainly divided into volume rendering and surface rendering. The surface rendering algorithm first extracts features through two-dimensional tomographic images, and then uses intermediate geometric primitives such as quadrilaterals, triangles, hexahedrons, cones, and tetrahedrons to fit the three-dimensional surface contour of the model to be reconstructed[1]. The volume rendering algorithm directly maps the input data into a three-dimensional figure on the screen without passing through the intermediate geometric primitives; the research in this article belongs to a surface rendering algorithm. Generally speaking, compared with the volume rendering algorithm, the operation of the surface rendering algorithm requires less memory and performs better in the details of the reconstruction model.

Currently, the main algorithms for surface rendering include: Triangulation of Contour Lines, Cuberille, Dividing Cubes, and Marching Cubes, etc.[2-5]. Among them, the Marching cube (MC) algorithm was proposed by Lorensen and Cline in 1987. Because of its good 3D reconstruction effect, fast reconstruction speed and simple algorithm principle, it has won the recognition of the majority of image developers in practical applications. The current MC algorithm has become a classic algorithm in the field of 3D reconstruction. With the in-depth research and application, we found that the MC
algorithm has the following shortcomings: ambiguity, it refers to the topological configuration between adjacent cubes does not match, and the finally reconstructed three-dimensional model has holes. In the calculation, because the MC algorithm processes all cube voxels in a traversal manner, part of the time is wasted in the calculation of empty voxels. Empty voxels refer to cubes that do not intersect with the isosurface, and generally occupy all More than 95% of the cube. The 3D mesh model that uses the MC algorithm for 3D reconstruction contains a large number of triangles, and too many triangles are not conducive to the rendering and storage of the model.

The ambiguity of the MC algorithm was first proposed by Dürst et al. in 1988 [7]. They found that the MC algorithm has some basic configurations that can be replaced by a variety of other forms of topological configurations. At this time, it is difficult to decide which topology configuration to use during the specific reconstruction. Due to the unreasonable choice of topological configuration, it is possible that the isosurfaces in adjacent cube voxels are not connected to each other, resulting in holes in the generated isosurfaces. The Marching Tetrahedra (MT) algorithm proposed in 1990 [8], as one of the earliest surface rendering algorithms to solve ambiguity, MT is based on the MC algorithm, which subdivides each cube voxel into multiple four sides Voxels, and then extract isosurfaces from each tetrahedron for reconstruction, but this method also has ambiguity when the cube is divided into tetrahedrons. Due to the ambiguity of segmentation, the reconstructed model may still have holes. In addition, MT will be much longer than MC algorithm in reconstruction time. In 1991, Nielson proposed the Asymptotic Decider [9]. This method can better judge the ambiguity of the cube surface. According to the criterion of judgment, the ambiguity of the cube surface can be resolved, but the method still cannot resolve the ambiguity of the cube. Regarding the ambiguity of MC algorithm body, the current main judgment methods include saddle point method, critical point method and generalized asymptotic method. For these judgment methods, although the ambiguity is resolved to a certain extent, it also increases the time for reconstruction. In 1990, Wilhelms proposed that in practical applications, generally only 3% of topological configurations have ambiguities [10]. Due to the low frequency of ambiguity, Wilhelms proposed an extended look-up table method, which expands the main ambiguity topology. The topological configuration is mainly based on the gray level of the cube vertex and the threshold. This kind of method is a method with special research value, because this kind of method not only has fast reconstruction speed but also good reconstruction quality under the premise of solving the reconstruction model hole. Based on this type of algorithm, Heiden et al. set up a corresponding reflection symmetric topology for each ambiguous topological configuration. In 1994, Zhou also proposed a similar method [11]. In 2013, Masala extended the 15 topological configurations of MC to 21 based on the extended lookup table method. This method can effectively solve the ambiguity problem, and it performs well in both the reconstruction speed and the quality of the 3D model. Some software has already used this method [12, 13].

In the optimization of the reconstruction time of the MC algorithm, the current methods used include parallel computing and octree methods to accelerate. The octree method can avoid the calculation of some empty voxels to a certain extent by dividing the input data into blocks. This method was proposed by Wilhelms and Van Gelder [14]. Montani proposed to use the midpoint to replace the linear interpolation point in the MC algorithm. This method reduces a lot of floating point operations and is a commonly used acceleration method for the current MC algorithm [15]. Based on this idea of Montani, some scholars also use golden section points, multi-segment points, etc. instead of linear interpolation points. When the amount of input data is large, the 3D reconstruction of the MC algorithm will generate a large number of triangles. The current main solution is to simplify the reconstructed 3D mesh
model. The Simplified Marching Cubes proposed by Vignolest in 2010 does not use interpolation to solve the intersection of the isosurface and the cube, but directly replaces the interpolation intersection with the vertices of the cube. This method can effectively avoid the ambiguity of the MC algorithm. And the number of triangles in the reconstructed 3D model is less than that of MC algorithm, but the accuracy of SMC algorithm reconstruction is not as good as MC algorithm, and its accuracy is between Cuberille and MC algorithm\(^{[17]}\). In this paper, aiming at the ambiguity of the MC algorithm's topological configuration, and based on the interconnection of isosurfaces in adjacent cube voxels, a surface rendering and reconstruction algorithm based on edge growth is proposed. The algorithm selects seed triangles, and then grows from the sides of the seed triangles to achieve the purpose of three-dimensional tissue reconstruction.

### 2. Algorithm description

#### 2.1 Method and process

In the process of MC 3D reconstruction, it mainly interpolates the isosurface corresponding to each cube based on 15 basic topological configurations, as shown in Figure 1.

![Fig.1 The 15 basic topological configuration of MC](image)

The ambiguity of the MC algorithm means that the topological configuration of the cube is not unique when selected, and the corresponding topological configuration can be replaced by other topological configurations. Due to the existence of ambiguity, and the MC algorithm processes each cube separately in the reconstruction process, it does not consider the connection between adjacent cubes, and sometimes it is reconstructed when it is uncertain which topology structure to use. There will be holes in the model, as shown in Figure 2.

![Fig.2 The hole caused by ambiguity](image)

As we all know, for an object, its external contour is a continuous surface, which corresponds to the isosurface in the three-dimensional reconstruction is continuous. From the perspective of the isosurface in the cube, the isosurface left by the three-dimensional surface contour of the object passing through the adjacent cube does not exist independently, and they are connected to each other, as shown...
in Figure 3.

Fig.3 The relationship between the isosurfaces of adjacent cubes

It can be seen from Figure 3 that when the isosurface passes through a pair of adjacent cubes, the isosurface will leave a line of intersection on the common surface of the adjacent cubes, and this line of intersection is the common edge of the isosurfaces in these two solids, and it plays the role of connecting the isosurfaces of the two sides. Based on the characteristics of this common edge, this paper proposes an MC algorithm based on edge growth. The core of the algorithm is to use the existing isosurface to generate the isosurface in the adjacent cube. As shown in Figure 3. If the isosurface in the cube on the left has been generated, but the one on the right has not yet, then the common edge of the isosurface in these two cubes can be used to generate the isosurface in the cube on the right. For this common edge, call it the growth side.

Like other types of growth algorithms, the algorithm proposed in this paper also has a growth queue to maintain the smooth progress of the entire reconstruction process. At the same time, it also has two tag arrays. The queue stores the growth edge information. The tag array records the processing of each cube and the growth information of the cubes placed in the growth queue. Among them, the growth edge information specifically includes the three-dimensional space coordinates of the two endpoints on the edge, the gradient values of the two endpoints, and the cube coordinates to which the growth edge will grow. The algorithm is mainly divided into the following steps:

Step 1: Select the seed triangle, and put the growth edge formed by the seed triangle into the growth queue.
Step 2: Take out the growth edge from the growth queue, and interpolate the intersection points of the isosurface and the cube edge according to the topology configuration, and connect the intersection points to form a triangular mesh.
Step 3: Put the new edge where the isosurface generated in Step 2 intersects with the cube surface as the growth edge and put it in the growth queue.
Step 4: Repeat steps 2 and 3 until the queue is empty.

2.2 Seed triangle selection

As can be seen from Figure 1, the isosurface corresponding to the MC algorithm topology configuration 1 is a triangle. In view of the simplicity of topological configuration 1, we used three sets of data to perform 3D reconstruction. The reconstruction results are shown in Figure 4. Figures (a), (b) and (c) are the reconstruction effects of data 1, 2 and 3, respectively, when using the MC algorithm. Figures (d), (e) and (f) show the reconstruction effect of data 1, 2 and 3 when only the cube of topology configuration 1 is reconstructed. From the comparison of the results of the same data reconstruction, although the number of triangles in the models (d), (e) and (f) is small, they can basically reflect the approximate outline of the object, and each independent triangular face piece is evenly distributed on each cross section. From the reconstruction effect, it can be seen that in the three-dimensional reconstruction of these models, the topological configuration 1 has a higher proportion.
Based on the simplicity and high proportion of topological configuration 1, this paper uses the triangle of topological configuration 1 as the seed triangle, and then uses the side of the seed triangle as the growth side for growth reconstruction.

In practical applications, the number of selected seed triangles must be reasonable, too few may not achieve the reconstruction effect due to the influence of noise, and too many will spend a lot of time on the selection of seed triangles. In most of the experiments in this article, all the triangles generated by topological configuration 1 in a layer are selected as seed triangles. This selection method is simple and easy to implement, and can achieve better reconstruction results. For the selection of the number of layers, it is necessary to select the number of layers with varying curvatures of the isosurface, because the content of topological configuration 1 in this layer is relatively large, and usually the intermediate layer can meet the requirements of seed selection. For the selection of seed triangles, it is not limited to selecting from a layer of cubes. It is also a good way to select artificially or in a designated three-dimensional space area. The core of the algorithm in this paper is to perform 3D reconstruction based on edge growth. The purpose of selecting the seed triangle is to obtain the growth edge, and the growth edge can also be obtained from other topological configurations in application.

2.3 3D reconstruction based on edge growth

2.3.1 Interpolation

In this algorithm, a three-stage interpolation method is proposed for interpolation reconstruction. This three-segment interpolation method is an interpolation method between the linear interpolation method and the midpoint selection method. It has better interpolation efficiency and high reconstruction accuracy.

Assuming that \( p_1 \) and \( p_2 \) are the three-dimensional coordinates of two vertices on one side of the cube, \( hu_1 \) and \( hu_2 \) are the gray values of the two vertices, and the threshold of the isosurface is \( Y \). When there is an intersection between this side and the isosurface, the intersection The calculation formula of coordinates using linear interpolation method is:

\[
p = p_1 + (p_2 - p_1)(Y - hu_1)/(hu_2 - hu_1)
\]

If the \((Y - hu_1)/(hu_2 - hu_1)\) in the formula is called the interpolation ratio \( k \), then:
\[ p = p_1 + k(p_2 - p_1) \]  \hspace{1cm} (2)

The selection of \( k \) value in the expression is the special feature of the three-segment interpolation method in this paper. Firstly, the lower limit \( m \), the upper limit \( n \), the lower value \( q \) and the upper value \( p \) should be set, in which \( 0 < q < m < n < p < 1 \) and \( q + p = 1 \) are required. Then, the absolute value of \( k \) is compared with \( m \) and \( n \). When the absolute value of \( k \) is less than \( m \), the absolute value of \( k \) is \( q \); When the absolute value of \( k \) is greater than \( n \), the absolute value of \( k \) is \( p \). In other cases, the absolute value of \( k \) is 0.5. In these three methods, the positive and negative requirements of \( k \) are always the same as before, only the absolute value of the value is normalized. In the practical application of this paper, \( q = 0.25 \), \( m = 0.3 \), \( n = 0.7 \), and \( p = 0.75 \).

2.3.2 Topological configurations

Because the three-stage interpolation method is adopted in the interpolation criterion in this article, each intersection point of the interpolation solution will only be between the two end points on each side.

![Fig.5 Two kinds of growth edges](image)

From the perspective of the growth edge, according to the different locations of the two end points of the growth edge, the growth edge can be divided into two types: the first is that the two end points of the growth side are on the two connected sides, and the growth side intersects the two sides to form a triangle, as shown on the left side of Figure 5. The other is that the two end points of the growing edge are on two opposite edges. The growing edge divides the cube surface into two quadrilaterals, as shown on the right side of Figure 5. In this paper, the former type of growth edge is called triangular edge, and the latter type of growth edge is called four-corner edge.

![Fig.6 Name of each vertex of the cube when growing with triangle edge](image)

When reconstructing with the growth edge, considering the uncertainty of the position of the growth edge, in order to facilitate the extraction of the isosurface in the cube. Therefore, the vertices of the cube need to be named based on the two end points \( I \) and \( II \) of the growth edge and the plane where the growth edge is located. Figure 6 shows the names of the vertices of the cube when the triangle edges are growing. The naming rule is: the names of all vertices on the surface where the growth edge is located begin with \( SS \), the surface where the growth edge is located is called the \( SS \) surface, the names of the
vertices opposite to the SS all begin with S_S, and the surface they are on is called the S_S surface. The vertices that just form a triangle on the SS surface and the growing edge are directly named SS, and the two vertices adjacent to the point SS on the SS surface are named SS_I and SS_II according to whether they are adjacent to the two end points I and II of the growing edge; the diagonal point opposite to the SS point on the SS surface is named SS_I_II. The name of each point on the S_S surface mainly refers to the name of the point on the adjacent SS surface. The names of the adjacent points on the two surfaces are the same in the suffix.

When using the triangular edge for growth three-dimensional reconstruction, take the point SS as the reference point, and compare its gray value $H_{ss}$ with the reconstruction threshold $Y$. If $H_{ss}$ is greater than threshold $Y$, the vertex with gray value greater than threshold $Y$ is said to meet the requirements, and the remaining points are points that do not meet the requirements; if $H_{ss}$ is not greater than threshold $Y$, then the vertex with gray value not greater than $Y$ is called The points that meet the requirements, the remaining points are the points that do not meet the requirements. As shown in Figure 6, the vertices that meet the requirements are marked with large dots, and the points that do not meet the requirements are marked with small dots. In the subsequent figures, large and small points are also used to mark each vertex. For some unmarked points in some topological configurations, they are called irrelevant points, and they do not participate in the specific isosurface generation. In addition, because the algorithm in this paper starts from the growing edge, only the isosurfaces that are connected to the growing edge are extracted in the cube body, so all the points that meet the requirements are connected to each other when extracting the isosurface.

According to the position of the cube surface where the growth edge is located, and the relationship between the gray value of each vertex of the cube and the threshold value, 11 topological configurations when the triangle edge is used for growth are summarized, as shown in Figure 7.

![Fig.7 The topological configurations when growing with triangle edge](image-url)
When growing with four corners, you need to first compare the gray values of the 8 vertices of the cube with the threshold $Y$, and then divide the vertices into two parts, the vertices with gray values greater than the threshold $Y$ and the gray values not greater than the threshold $Y$. The apex. Select the smaller number of points as the reference point, that is, the points that meet the requirements, and the number of points that meet the requirements will be in the closed interval of 1 and 4. As with the growth of triangular edges, when the four corners are used for growth, the vertices of the cube also need to be named. The naming rule is: all the vertices on the surface where the growth edge is located begin with SS, and the surface where the growth edge is located is called SS face; the face of the SS face-to-face is the $S_S$ face, and the names of the vertices on this face start with $S_S$; on the SS face, the vertex that meets the requirements adjacent to vertex I of the growth edge is called the $SS_I$ point, and the adjacent points that do not meet the requirements are called $SS_UP_I$. The vertex that meets the requirements adjacent to vertex II of the growth edge is called point $SS_{II}$ and the adjacent point that does not meet the requirements is called $SS_UP_{II}$; the vertices on the $SS$ surface and the vertices connected on the $SS$ surface are the same in the name suffix. Figure 8 shows the specific naming of each vertex.

The same as the reconstruction with triangular edges, according to the position of the cube surface where the growth edge is located, all the vertices that meet the requirements are connected to each other, and the six topological configurations for reconstruction with four corners are summarized, as shown in Figure 9.
topological configuration, it is also necessary to determine whether the intersection line between the iso-surface and the surface of the cube can become a new growth edge and continue to grow. The judgment is based on the following: the edge is not the first growing edge; neither the value of the processing flag nor the value of the growth flag in the adjacent cubes sharing this side is 4.

2.3.3 Cube Marks

When a cube is processed, its processing status needs to be recorded. Only when the processing mark value reaches a certain requirement, there is no need to do any further processing on the cube. Each cube has two marks, one is the processing mark, which records the situation of the cube being processed according to the growth criterion; the other is marked as the growth mark, which records the situation that the growth information of the cube is placed in the growth queue. For these two marks, there are 5 levels, which are represented by the numbers 0, 1, 2, 3, and 4. Level 0 is the lowest, indicating that the cube has not undergone any processing; level 4 is the highest, indicating that the cube has been processed and no further processing is required. Other numbers indicate that although the cube has been processed, it may still need to be processed. The above is the case when the 17 topological configurations grow with one growth edge. However, in actual applications, there may be multiple separate iso-surfaces in a cube, so it is necessary to continue processing such a cube.

The specific marker assignment criteria are as follows: ① Extract the cube corresponding to the seed triangle, directly set all the tags of the corresponding cube to 4 after the seed triangle is extracted; ② When the information of the growth side is put into the growth queue, it needs to be processed Add 1 to the cube growth tag value; ③ When the corresponding topological configuration of the cube is 4, 5, 7, 8, 10, 11, 13, 15 and 17, after the cube is processed, the processing tag value is set to 4; ④ When the corresponding topological configuration of the cube is 2, 3, 6, 9, 12, 14, and 16, after the cube is processed, the processing flag value is set to 3; ⑤ When the topological configuration of the cube is 1, add 1 to its processing tag value.

When the processing tag value of a cube is 4, there is no need for any subsequent processing of the cube; when the growth tag value of the cube is 4, there is no need to put the cube's growth information into the growth queue. In addition, when the processing label of a cube is greater than 1 and less than 4, only the iso-surface of topological configuration 1 is extracted in the cube.

3. Results and discussion

3.1 Ambiguous hole detection result

For the algorithm proposed in this paper, the primary problem is to solve the ambiguity of topological configuration. Because of the ambiguity, the generated 3D model will have holes in some details, which greatly reduces the overall quality of the model. Aiming at the hole problem, Masala et al. proposed a current simplest solution. It deleted one of the original 15 topological configurations of the MC algorithm, and then added 7 more, and finally set the topological configuration to 21 kinds.

For the inspection of whether the model generated by the 3D reconstruction has holes, this article uses the data used by Masala, which is called MC_example by Masala, and its size is 10 × 10 × 10, that is, there are 10 layers of data and each layer is of size 10 × 10. Except for the 5th, 6th and 7th layers, the data size of the other layers are all 0. The data of the 5th, 6th and 7th layers are shown in Figure 10. The data of the 6th and 7th layers are the same.
When using the set of data of MC_example for reconstruction, the reconstruction results of the standard MC algorithm, Masala's improved MC algorithm and the algorithm used in this article are shown in Figure 11. It can be seen from the figure that the reconstruction results of the standard MC algorithm have obvious holes. The reconstruction results of the algorithm used in this article are the same as the improved MC algorithm of Masala, and the generated 3D model has no holes. Figure 12 shows the mesh display results of each model in Figure 11. It can be seen that the standard MC algorithm mesh model has obvious jaggedness on the left side. Because of the lack of some triangles on these saw teeth, the final generated model is incomplete and there are some holes. The mesh model corresponding to the improved MC algorithm of Masala is roughly the same as the mesh model corresponding to the algorithm in this paper, and all generated are a closed mesh model.
3.2 Quality and time analysis of reconstruction model

As an improved version of the MC algorithm, Masala's MC algorithm mainly improves holes caused by ambiguity. In order to test the performance of the algorithm in this paper, this paper uses two sets of experiments to test the reconstruction effect of this paper. Table 1 shows the environment configuration of the overall experiment.

| Name                  | Parameter       |
|-----------------------|-----------------|
| Operating system      | Win10 64        |
| CPU                   | i5-8400         |
| Memory                | 8GB             |
| Development environment| VS2015         |
| platform              | VTK             |

![Fig.13 3D reconstruction model of the first group of experiments](image)

The first set of experimental data is a set of segmented medical image data, which consists of 145 two-dimensional tomographic images with a size of $125 \times 105$. The three-dimensional reconstruction model of Masala's improved MC algorithm and the algorithm in this paper is shown in Figure 13. The seed triangle of the algorithm in this paper is obtained from all the cube voxels in the middle layer. It can be seen from the figure that the two algorithms have the same grid on the overall three-dimensional external contour of the three-dimensional image, but from the inside of the model, the improved MC algorithm of Masala has a lot more internal grids than the model reconstructed by the algorithm in this paper. In practical applications, these internal grids do not participate in the display of the model. These internal grids are mainly caused by incomplete segmentation. In the case that only a complete external 3D contour mesh model is needed, the algorithm in this paper is better than the improved MC algorithm of Masala in the quality of the reconstructed model.

| Algorithm | Time(s)  | Triangular faces |
|-----------|----------|------------------|
| Masala's  | 0.219711 | 79728            |
| Our       | 0.212538 | 71472            |
Under the same environment configuration, the time and the number of triangles spent on the first set of data for the three-dimensional reconstruction of the two algorithms are recorded in Table 2. It can be seen that the algorithm in this paper is less than Masala's improved MC algorithm in the number of triangles in the reconstruction model. In terms of reconstruction time, there is little difference between the reconstruction of the two algorithms. The algorithm in this paper is slightly smaller than the improved MC algorithm of Masala.

The second set of experiments used 4 sets of data for reconstruction, and the same reconstruction threshold was used for these 4 sets of data. At the same time, the Masala improved MC algorithm and the algorithm used in this paper were used for 3D reconstruction. The reconstruction result is shown in Figure 14, where the seed triangle of the algorithm in this paper is obtained from all the cube voxels in the middle layer.

In Figure 14, Figures (a), (b), (c) and (d) are three-dimensional models reconstructed by Masala's improved MC algorithm under data 1#, 2#, 3# and 4#, respectively, (e), (f), (g) and (h) are the three-dimensional models reconstructed by the algorithm used in this paper under data 1#, 2#, 3# and 4#. It can be seen that the results of Masala's improved MC algorithm and the 3D reconstruction of this algorithm are roughly the same. Figure 15 is a partial enlarged view of the chest of the reconstructed 3D model during the 2# 3D reconstruction. It can be seen that the model of Masala's improved MC algorithm has many scattered small fragments compared with the algorithm in this paper, which is somewhat similar to the noise in the two-dimensional image. These fragments reduce the overall display effect of
Therefore, when some fragments that are not connected to the main contour are not needed, the reconstruction quality of this algorithm is better than that of Masala's improved MC algorithm.

![Fig.15 Partial enlarged view of the 3D reconstruction model of data No. 2](image)

Table 3 shows the time spent by the two algorithms in the reconstruction process of the second set of experiments and the number of triangles in the mesh model. It can be seen that, as before, under the same reconstruction parameters, the number of triangles in the reconstruction model of this algorithm is less than that of Masala's improved MC algorithm. In terms of reconstruction time, the algorithm used in this article takes more time to reconstruct the 2# data. The reconstruction time of the other three sets of data is less than that of Masala's improved MC algorithm.

### 3.3 The influence of seed region selection on reconstruction model

In order to facilitate the use of this algorithm for 3D reconstruction of medical images, we made a simple 3D reconstruction system. The system can set the selected area of the seed triangle to reconstruct a three-dimensional model connected to the seed triangle. We selected a set of human brain data to test the system. The data consists of 320 two-dimensional images. The results of the three-dimensional reconstruction are shown in Figure 16, where Figure (a) is a model of the entire data reconstruction using the MC algorithm. The model is the entire skull corresponding to the data, and each bone is scattered in each area. Figure (b) is the reconstruction result of the system when the seed region is \{(x,y,z) | 0 ≤ x < 400, 0 ≤ y < 400, 0 ≤ z < 10\}, and its corresponding three-dimensional model is the maxilla. Figure (c) is the reconstruction result of the system when the seed selection area is \{(x,y,z) | 200 ≤ x < 300, 400 ≤ y < 478, 270 ≤ z < 318\}, and its corresponding three-dimensional model is the mandible. It can be seen that when the system selects different seed regions, the reconstructed model is also different. Based on this reconstruction effect, for multiple scattered tissues, the algorithm in this paper can be used to set the seed area to perform three-dimensional reconstruction of the designated part of the image.
4. Conclusion

This paper proposes a Marching Cube algorithm based on edge growth for 3D reconstruction. First of all, the 3D model reconstructed by this algorithm does not have holes caused by ambiguity. Secondly, from the perspective of overall reconstruction quality, it is consistent with the appearance of the model reconstructed by Masala's improved MC algorithm, but when only the main contour of the model is needed, the reconstruction effect of this algorithm is better. In terms of reconstruction accuracy, this algorithm and MC and Masala's improved MC algorithm are based on cube voxel interpolation for 3D reconstruction, so their reconstruction accuracy is the same. In the application of this algorithm, when the tissues in the data are multiple scattered parts, the three-dimensional contour of the specified part can be extracted by setting the region of the selected seed.

For the algorithm in this article, its main limitations are as follows: The algorithm uses a queue to store the growth side information. Compared with other types of algorithms, it requires more memory space and is not suitable for large-scale reconstruction on a small machine. This algorithm generates a mesh model based on seeds, and is not suitable for reconstructing some scattered multiple objects such as broken bones. It needs to artificially set seeds for each scattered part, or expand the area of seed selection.

In the subsequent development, multithreading can be used to accelerate the reconstruction speed of the algorithm. It can also be combined with volume rendering. First, determine the area to be reconstructed according to the display results of volume rendering, then set the seed area and reconstruction threshold, and use the surface rendering algorithm in this paper to accurately reconstruct the tissue of the specified part.

References

[1] Newman T S, Yi H. A survey of the marching cubes algorithm[J]. Computers & Graphics, 2006, 30(5):854-879.
[2] Keppel E. Approximating Complex Surfaces by Triangulation of Contour Lines[J]. IBM Journal of Research and Development, 1975, 19(1):2-11.
[3] Herman, Gabor T, LIU, Hsun Kao. Three-dimensional display of human organs from computed
tomograms. Computer graphics and image processing, 1979, 9.1: 1-21.

[4] Cline H E. Two algorithms for the three-dimensional reconstruction of tomograms[J]. Medical Physics, 1988, 15(3): 320.

[5] Lorensen W E, Cline H E. Marching cubes: A high resolution 3D surface construction algorithm[J]. ACM SIGGRAPH Computer Graphics, 1987, 21(4):163-169.

[6] Jin J, Wang Q, Shen Y, et al. An improved marching cubes method for surface reconstruction of volume data[C].2006 6th World Congress on Intelligent Control and Automation. IEEE, 2006, 2: 10454-10457.

[7] Martin J. Dürst. Re: additional reference to marching cubes[J]. Acm Siggraph Computer Graphics, 1988, 22(5): 243.

[8] Shirley P , Tuchman A . A polygonal approximation to direct scalar volume rendering[J]. ACM SIGGRAPH Computer Graphics, 1990, 24(5): 63-70.

[9] Nielson G M , Hamann B . The asymptotic decider: resolving the ambiguity in marching cubes[C]. IEEE Conference on Visualization, Visualization. DBLP, 1991.

[10] Van Gelder, Allen, Wilhelms, Jane. Topological considerations in isosurface generation[C]. Workshop OnVisualization. ACM, 1990.

[11] Zhou C, Shu R, Kankanhalli M S. Handling small features in isosurface generation using marching cubes[J]. Computers & graphics, 1994, 18(6): 845-848.

[12] Masala G L, Golosio B, Oliva P. An improved Marching Cube algorithm for 3D data segmentation[J]. Computer Physics Communications, 2013, 184(3): 777-782.

[13] Cercos-Pita J L, Cal I R, Duque D, et al. NASAL-Geom, a free upper respiratory tract 3D model reconstruction software[J]. Computer Physics Communications, 2018, 223: 55-68.

[14] Wilhelms J, Van Gelder A. Octrees for faster isosurface generation[J]. ACM Transactions on Graphics (TOG), 1992, 11(3): 201-227.

[15] Montani C, Scateni R, Scopigno R. Decreasing isosurface complexity via discrete fitting[J]. Computer Aided Geometric Design, 2000, 17(3): 207-232.

[16] Lakshmipathy J, Nowinski W L, Wernert E A. A novel approach to extract triangle strips for isosurfaces in volumes[C]. Proceedings of the 2004 ACM SIGGRAPH international conference on Virtual Reality continuum and its applications in industry. ACM, 2004: 239-245.

[17] Vignoles G L, Donias M, Mulat C, et al. Simplified marching cubes: An efficient discretization scheme for simulations of deposition/ablation in complex media[J]. Computational Materials Science, 2011, 50(3): 893-902.

[18] Li X, Han C Y, Wee W G. On surface reconstruction: A priority driven approach[J]. Computer-Aided Design, 2009, 41(9): 626-640.