

# Using MC Algorithm to Implement 3D Image Reconstruction for Yunnan Weather Radar Data

Zhongneng Liu<sup>1</sup>, Zhenzhong Shi<sup>1</sup>, Murong Jiang<sup>1</sup>, Jie Zhang<sup>2</sup>, Liqing Chen<sup>1</sup>, Tian Zhang<sup>1</sup>, Gongqin Liu<sup>1</sup>

<sup>1</sup>School of Information Science and Engineering, Yunnan University, Kunming, China

<sup>2</sup>Yunnan Meteorological Information Center, Kunming, China

Email: jiangmr@ynu.edu.cn

**How to cite this paper:** Liu, Z.N., Shi, Z.Z., Jiang, M.R., Zhang, J., Chen, L.Q., Zhang, T. and Liu, G.Q. (2017) Using MC Algorithm to Implement 3D Image Reconstruction for Yunnan Weather Radar Data. *Journal of Computer and Communications*, 5, 50-61.

<https://doi.org/10.4236/jcc.2017.55005>

**Received:** February 23, 2017

**Accepted:** March 27, 2017

**Published:** March 30, 2017

Copyright © 2017 by authors and Scientific Research Publishing Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

---

## Abstract

3D image reconstruction for weather radar data can not only help the weatherman to improve the forecast efficiency and accuracy, but also help people to understand the weather conditions easily and quickly. Marching Cubes (MC) algorithm in the surface rendering has more excellent applicability in 3D reconstruction for the slice images; it may shorten the time to find and calculate the isosurface from raw volume data, reflect the shape structure more accurately. In this paper, we discuss a method to reconstruct the 3D weather cloud image by using the proposed Cube Weighting Interpolation (CWI) and MC algorithm. Firstly, we detail the steps of CWI, apply it to project the raw radar data into the cubes and obtain the equally spaced cloud slice images, then employ MC algorithm to draw the isosurface. Some experiments show that our method has a good effect and simple operation, which may provide an intuitive and effective reference for realizing the 3D surface reconstruction and meteorological image stereo visualization.

## Keywords

Weather Radar Data, 3D Reconstruction, MC Algorithm, Cube Weighting Interpolation

---

## 1. Introduction

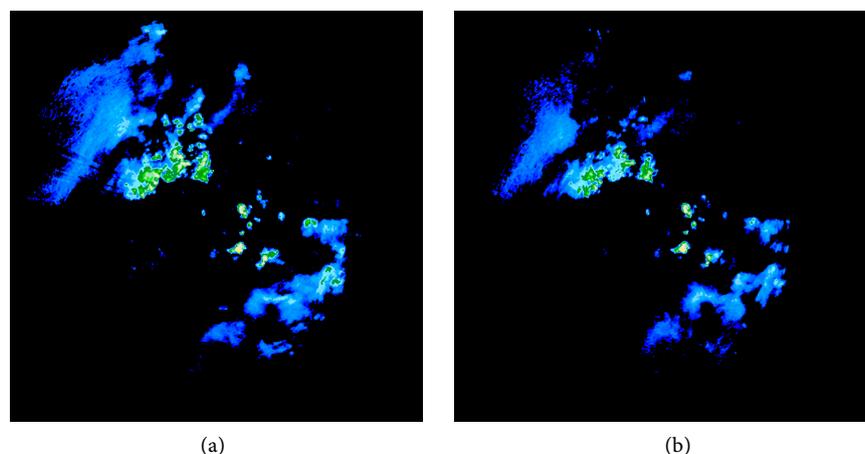
In recent years, most of the meteorological radar products are still expression by using two-dimensional data structure to represent three-dimensional cloud state. This process is both abstract and difficult to understand. The three-dimensional image reconstruction for the radar data can not only transform the

abstract meteorological data into the visual image information which make people understand and accept easily, but also explore the further potential information from the meteorological targets through spatial three-dimensional modeling and stereoscopic displaying, provide more accurate data basis for the short-term weather forecasts.

Three-dimensional image reconstruction has more extensive and mature application in medical image processing, mechanical manufacturing, computer animation, film and television design and many other fields [1] [2] [3]. In the meteorological field, owing to the weather objects in the entire detection space are discrete, irregular and uneven distributed, most of the visualization products are confined into the two-dimensional structure, as shown in **Figure 1**, the images reconstructed from the radar data with two elevation angles in Kunming Radar Station at 18:23 on July 18, 2013. If we could provide the three-dimensional cloud structure of this moment by using the three-dimensional reconstruction technology, we would better understand more meteorological object feature information and know the weather changes.

There are two kinds of three-dimensional reconstruction technologies. One is known as the surface drawing method, which approached through the geometric unit splicing fitting surface to describe the object three-dimensional structure, the classical algorithms including Marching Cubes (MC), Marching Tetrahedral and Dividing Cubes, etc.; another is called the volume rendering method based on the volume data, it projected directly the voxel into the display plane, the classic algorithms have: light projection, projection imaging and Frequency domain transform method. The commonality of these algorithms is that using many triangular patches to fit the equivalent surfaces, with the advantage that hardware acceleration can be achieved when rendering some 3D reconstructed images [4]-[9]. But more intermediate data may be generated from the original data to the final display; the amount of calculation is very large.

Due to the radar detecting elevation angle is small, the data form is also close to the plane data, then we can obtain the equally spaced cloud slice images by



**Figure 1.** Reconstructed images with two elevation angles: (a) Elevation angle 3.4 degree; (b) Elevation angle 4.3 degree.

data preprocessing. Besides Marching Cubes algorithm on the three-dimensional reconstruction for the slice images are superb both in speed and performance, therefore, this paper uses MC algorithm to carry out three-dimensional reconstruction for cloud scanning images.

Firstly, we need to establish the 3D data field because the weather radar data are not the equidistant slice images data, then use the MC algorithm to reconstruct the 3D cloud image of the weather radar, and realize the stereo visualization of the two-dimensional meteorological data.

## 2. Cube Weighting Interpolation Method

Based on the working principle of the weather radar, we know that each layer of the scanning data obtained by the weather radar is a cone scanning surface, all the objects in one cone scanning surface have the different high attitude, we must search each layer of scanning data to obtain the specific attitude firstly. Secondly, we compute contour graphic data in specific attitude by doing some data preprocessing, such as the coordinate transformation, data interconnection and blindness etc. After above processes finished, the object three-dimensional data field is formed. Then, we use spatial interpolation to produce a three-dimensional weather object image.

In order to establish the 3D data field, we propose a projection method called Cube Weighting Interpolation (CWI) to obtain the equally spaced cloud slice images.

CWI includes the following three steps:

- 1) Standardize the txt data and map it into the 3D space;
- 2) Build standard 3D grid data;
- 3) Correct and optimize the grid data.

### Step 1:

Setting the original data to  $d_o(x_o, y_o, z_o)$ , where  $x_o$ ,  $y_o$  respectively represent the data elements in a radar map  $P(1000 \times 1000)$  within the row and column coordinates;  $z_o$  denotes the data element where the radar map number  $P_{no}$ ;  $d_o$  denotes the echo intensity detected by the radar. Besides, the echo intensity is standardized as follows:

$$\begin{cases} d_o = 0, & \text{if } d_o < 5 \\ d_o = 85, & \text{if } d_o > 85 \end{cases} \quad (1)$$

Let the mapping data be  $d_i(x_i, y_i, z_i)$ , then the conversion equation of  $d_o$  and  $d_i$  is as follows:

$$\begin{cases} x_i = x_o \times \cos \alpha \\ y_i = y_o \times \cos \alpha \\ z_i = \sqrt{x_o^2 + y_o^2} \times \sin \alpha \\ d_i = d_o \end{cases} \quad (2)$$

where,  $x_i, y_i, z_i$  represent the spatial position coordinates separately when the original data mapping to the three-dimensional space,  $\alpha$  is the angle between the scanning cone surface and the horizontal plane,  $d_i$  indicates the echo in-

tensity at the coordinate position, each mapping data  $d_i$  corresponds to the echo intensity data about a space sampling point during the observation process.

**Step 2:**

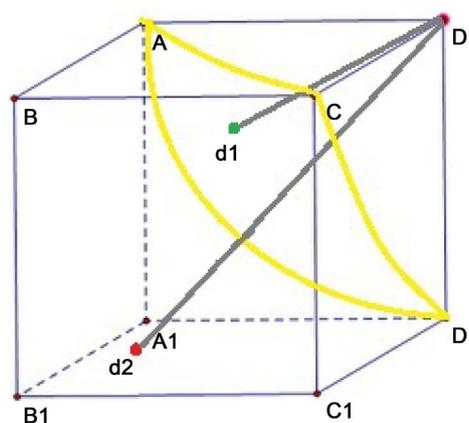
It must to use spatial interpolation, since the radar scanning sampling data is distributed in 14 tapes, and distribution of the data samples is irregular and uneven.

For any sample point  $d_i$ , its real spatial position falls into a unit cube  $M$  surrounded by an integer axis with a length and width of 1 km, and the distance between the sampling point  $d_i$  and the cube  $M$  is close, as shown in **Figure 2**.

Assumed that vertex  $D$  is center of sphere whose radius is length of the unit cube  $M$  in **Figure 2**, and these curves in **Figure 2**, which indicate  $1/8$  spherical surface of sphere. The weight of each sampling point  $d_i$  can be calculated according to the magnitude of the distance between the sampling points  $d_i$  and vertex  $D$  in the eight unit cube adjacent to vertex  $D$ , and the weighted average of the sampling points  $d_i$  can be used as estimated echo intensity value of the vertex  $D$ , and specific equation is as follows:

$$d(x, y, z) = f(D) = \frac{\sum_{i=1}^n [q(i) \times d_i]}{\sum_{i=1}^n q(i)} \tag{3}$$

Let the echo intensity be  $d(x, y, z)$ ,  $(x, y, z)$  is position coordinate of echo intensity in matrix, where  $x \in [-150, 150]$ ,  $y \in [-150, 150]$ ,  $z \in [0, 20]$ ,  $d(x, y, z)$  expresses the observation data that stored in the three-dimensional matrix  $(x, y, z)$ , and also represents echo intensity at the  $(x, y, z)$  in real space.  $d(x, y, z)$  can be obtained by taking the weighted average of the sampling point data  $d_i$  near the real space  $(x, y, z)$ .  $D$  represents the vertex of a standard cube, and  $n$  represents the number of data sampling points of the eight unit cubes adjacent with vertex  $D$ , and  $n$  is a nonnegative integer. When  $n > 0$ , which means that exists data sampling point in the eight adjacent unit cubes, and we can estimate  $D$  point with weighted average. When  $n = 0$ , Equation (3) is meaningless, point  $D$  cannot use this equation to obtain, then estimated data at  $D$  is empty, we can fill it with data constants that beyond the normal data scale, acts



**Figure 2.** Cube  $M$ .

as a placeholder, which is helpful to repair data after that.

$$q(i) = \frac{1}{\Delta l^\gamma} - 1 \tag{4}$$

where:

$$\gamma > 0$$

In the Equation (4),  $q(i)$  means that the weight is assigned to the different sampling points.  $\gamma$  is a parameter related to the weight distribution, which determines the relation between the weight assignment and  $\Delta l$ , here  $\gamma$  is 1. When  $\gamma = 1$ , there is an inverse relationship between the weight size and  $\Delta l$ . It is worth noting that  $q(i)$  is the weight, and its minimum value is 0, and is not negative. So when  $q(i)$  is negative, we unified with 0 instead it.

$$\Delta l = \sqrt{\Delta x^2 + \Delta y^2 + \Delta z^2} \tag{5}$$

where:  $\Delta l$  represents the distance between the sampling point and the standard cube vertex  $D$ .

$$\begin{cases} \Delta x = |x_i - \text{floor}(x_i)| \cup |x_i - \text{ceil}(x_i)| \\ \Delta y = |y_i - \text{floor}(y_i)| \cup |y_i - \text{ceil}(y_i)| \\ \Delta z = |z_i - \text{floor}(z_i)| \cup |z_i - \text{ceil}(z_i)| \end{cases} \tag{6}$$

In the Equation (6), using the commands round up and round down in MATLAB, we can get quickly the coordinate distance between the data sample points  $d_i$  and the adjacent eight standard cube vertices. The coordinates of the eight vertexes of the standard cube associated with the data sample points can be quickly obtained by combination of  $x$ ,  $y$ , and  $z$  with rounding up or rounding down respectively, and the echo intensity of the sample point  $d_i$  can be assigned to the eight vertices according to the weight rule.

Through this step, we can establish a three-dimensional space grid with 1 km interval. We call this grid as a standard grid. The data  $d$  of grid intersection is the basis of the next three-dimensional display and model reconstruction.

**Step 3:**

In step 2, we have faced two major problems:

- 1) Data sampling points are sparse where away from the radar, there will be the case of  $n = 0$  in the Equation (3), there is no data to estimate the vacuum point;
- 2) Data samples are dense and the large amount of data is allocated where near the radar. Although it helps to offset the outliers, the influence of the key data is weakened and the estimation results are balanced and cannot be fully characterized.

In order to solve the above two problems, this paper uses a method to change the sphere radius of the estimated grid point data. When constructing a standard grid point, we always use 1km as the sphere radius of the point  $D$ . If the sphere radius was increased, we could reduce the risk of problem 1, and if the sphere radius was reduced, we could alleviate conflict of problem 2.

The concrete operation of this method is to do the scaling of the real space in  $d_i(x_i, y_i, z_i)$ , and when the coordinates  $(x_i, y_i, z_i)$  become half of the original,

the grid node spacing will turn into 2 km, and sphere radius of the data point D also turn into 2 km. Then the number of data in the sphere will increase, which will alleviate problem 1. Although the mesh density is reduced by half, the net interpolation can be used to compensate for the mesh density, and we can get a grid with a precision of 2 km. The grid data is recorded as  $d^{2\text{km}}(x, y, z)$ , the original grid data is recorded as  $d^{1\text{km}}(x, y, z)$ . Using the similar method, we convert coordinate  $(x_i, y_i, z_i)$  to  $2 \times (x_i, y_i, z_i)$ , the grid node distance becomes 0.5 km, the grid density becomes 2 times the original. Then we restore the original grid density by extracting even bits in grid nodes as the target grid, this grid alleviates the contradiction of problem 2, and its data is recorded as  $d^{0.5\text{km}}(x, y, z)$ .

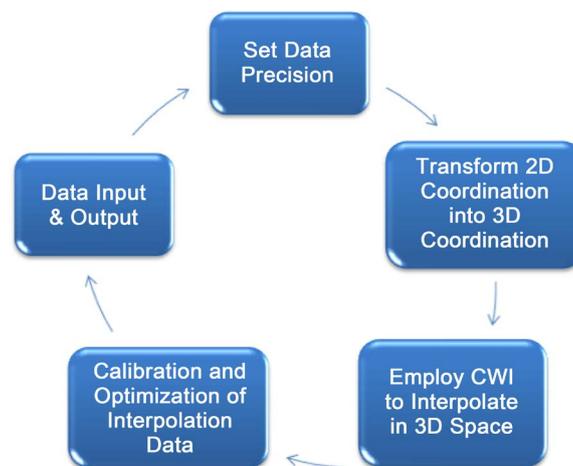
$d^{0.5\text{km}}(x, y, z)$ ,  $d^{1\text{km}}(x, y, z)$ ,  $d^{2\text{km}}(x, y, z)$  compared to the three, their estimated accuracy sort is from high to low, their vacuum points number is from more to less. Therefore, the final corrected mesh data is filled by  $d^{0.5\text{km}}(x, y, z)$ , the remaining data vacuum points are filled by  $d^{1\text{km}}(x, y, z)$ . Remaining data vacuum points are further filled by  $d^{2\text{km}}(x, y, z)$  and finally we can get the composite grid data  $d(x, y, z)$ . Experiments show that, after filling the grid data, 18 km above sea level where still does not appear data vacuum point. Clouds above 18 km are scarce according to actual observation, which are almost negligible, thus the results  $d(x, y, z)$  of the composite structure can meet the requirements of observation and analysis.

The details of CWI flow diagram can be shown as **Figure 3**.

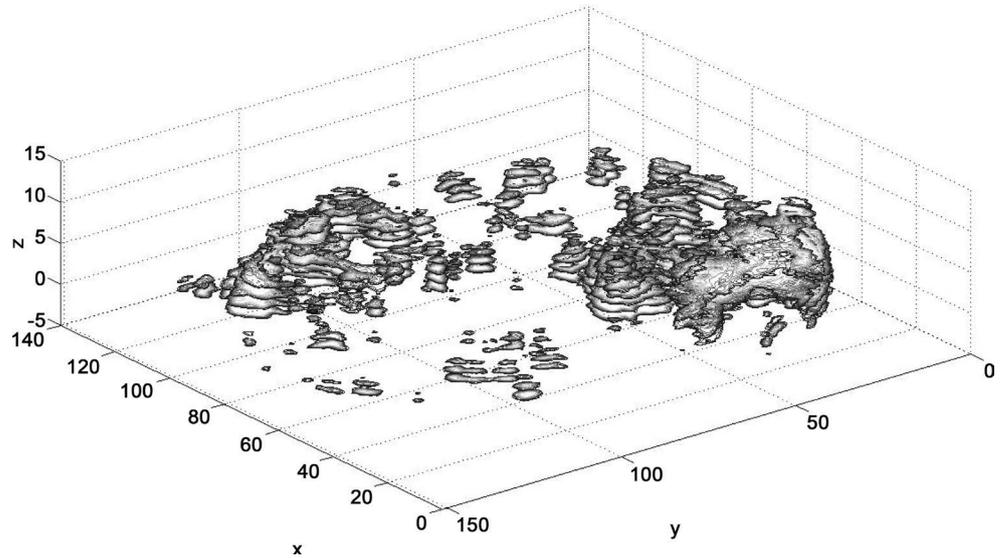
After above three steps, we can get the three-dimensional data field, as shown in **Figure 4**.

### 3. Using MC Algorithm to Reconstruct 3D Image of Cloud Data

Marching Cubes (MC) algorithm is the most representative method among the isosurfacing algorithm with the three-dimensional spatial rule data field. This method is simple and easy to implement, additionally it has been used widely [10] [11] [12] [13] [14].



**Figure 3.** CWI flow diagram.



**Figure 4.** 3D data field.

The essence of the MC algorithm is to view a series of two-dimensional slice data as a three-dimensional data field, from which extract a substance with a certain threshold, and then using a topological form to connect as a triangular. The core of the MC algorithm is to individually deal with each voxel in the three-dimensional data field, then use the compare data between each peak in each voxel and the isometric surface to determine the intersection of the voxel and the isometric surface. Computing the isosurface normal vector to draw the equivalent surface, among which the determination of the intersection form between the voxel and the isometric surface and the drawing of the isosurface are two key operations.

### 3.1. Voxel Model

First, we evenly sample the data in the 3D data field. Let  $x, y, z$ , in the direction of the sampling interval corresponds to  $\Delta x, \Delta y, \Delta z$ . We use this sample pitch to sample the three-dimensional data field. The cube region, formed by eight adjacent sampling points, is just a voxel. Simultaneously, these eight sample points referred as the corners of the voxels. Supposing these eight points in the spatial position index are  $(i, j, k), (i+1, j, k), (i, j+1, k), (i, j, k+1), (i+1, j+1, k), (i, j+1, k+1), (i+1, j, k+1), (i+1, j+1, k+1)$ , as shown in **Figure 5** [15].

Assumed  $P_6(x, y, z)$  is any point within the voxel, it is necessary to convert its spatial coordinates into image coordinates  $(i_6, j_6, k_6)$  in order to calculate the value of voxel, where  $i_6 = x/\Delta x, j_6 = y/\Delta y, k_6 = z/\Delta z$  are represented respectively the position index of the voxel in directions of  $X, Y$ , and  $Z$  in the body data. Using the three-linear interpolation can obtain the value of voxel for the point, and its value can be expressed as

$$\begin{cases} f(p_6) = f(p_4)(i+1-i_6) + f(p_5)(i_6-i) \\ f(p_4) = f(p_0)(j+1-j_6) + f(p_3)(j_6-j) \\ f(p_5) = f(p_1)(j+1-j_6) + f(p_2)(j_6-j) \end{cases} \quad (7)$$

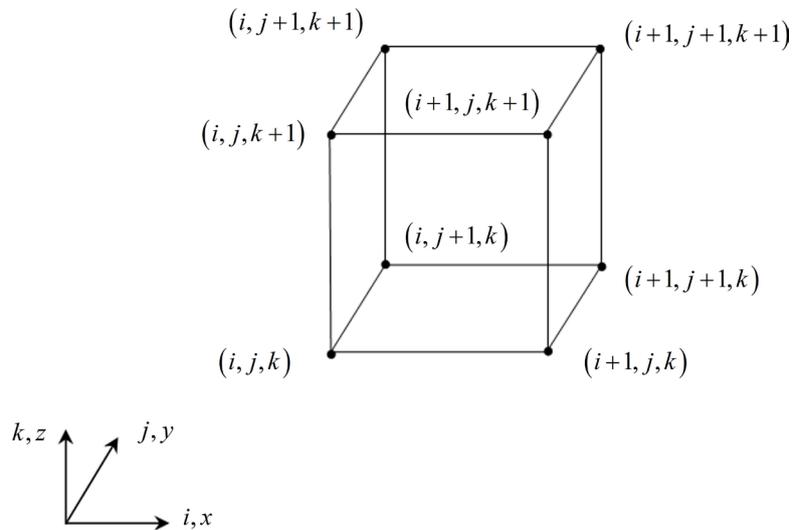


Figure 5. Voxel model.

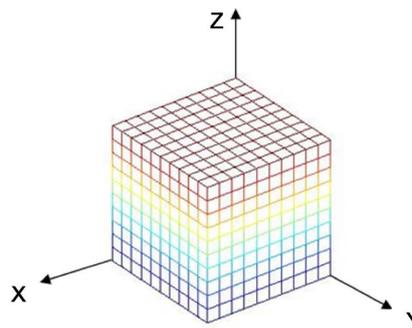


Figure 6. Volume data based on Voxel.

$$f(p_n) = f(i(n), j(n), k)(k+1-k_6) + f(i(n), j(n), k+1)(k_6 - k) \quad (n = 0, 1, 2, 3) \tag{8}$$

where:

$$i(n) = \begin{cases} i & n = 0, 1 \\ i+1 & n = 2, 3 \end{cases}, \quad j(n) = \begin{cases} j & n = 0, 3 \\ j+1 & n = 1, 2 \end{cases}$$

By the above equation, we can get the equation as follows:

$$f(x, y, z) = a_0 + a_1x + a_2y + a_3z + a_4xy + a_5yz + a_6xz + a_7xyz \tag{9}$$

In Equation (9),  $a_0, a_1, \dots, a_7$  are constants, which are uniquely determined by the values of the eight vertices in the voxel. According to Equation (9) we can obtain the property value of any point inside the voxel.

Assumed that the three-dimensional data field constructed by the slice sequence of the meteorological image is a rectangular that the length is  $L \times VX$ , the width is  $M \times VY$ , and the height is  $N \times VZ$ . The surface of the rectangular is parallel to the coordinate plane, and for the voxel

$(i, j, k)$ ,  $i = 1, 2, \dots, L$ ,  $j = 1, 2, \dots, M$ ,  $k = 1, 2, \dots, N$ , the value of volume data is  $f(i, j, k)$ . The volume data is a three-dimensional array, that the sizes is  $L \times M \times N$ , as shown in Figure 6.

### 3.2. Algorithm Steps

The basic assumption of MC algorithm is that the bulk data is locally continuous. According to this assumption, if two adjacent sampling points are greater than and less than the value of isosurface, there be an equivalent point on the edges of them. If we get the equivalent points on each edge, we can regard these points as the vertices, and use a series of triangles to approach isosurface.

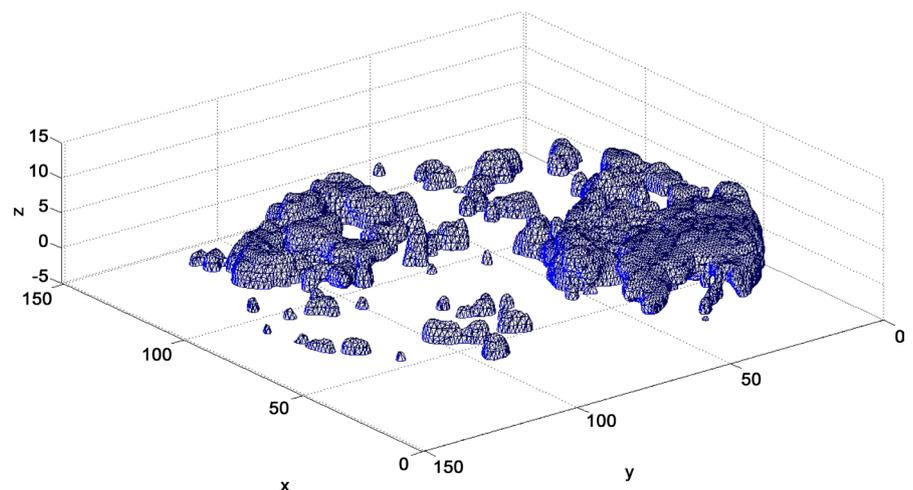
The main steps of MC algorithm reconstruct the three-dimensional image are as follows:

- The three-dimensional discrete rule data field is read hierarchically.
- Scanning two layers of data, to construct voxels one by one. The 8 corners of each voxel take the adjacent two layers.
- Comparing the values of each corner of the voxel with a given isosurface dimension. According to the comparison result, the index table of the voxel is constructed.
- According to the index table, we obtain the boundary voxels, which have an intersection with the isometric surface.
- Using the linear interpolation algorithm to calculate the intersection of voxel edge and isometric surface.
- Using firstly the central difference method to calculate the normal vector at the corners of voxels. Then calculating the normal vector at the vertex of the triangular patches by the linear interpolation method,.
- Finally, drawing isomorphic images according to the coordinates of each triangle points at the coordinates and the normal vector.

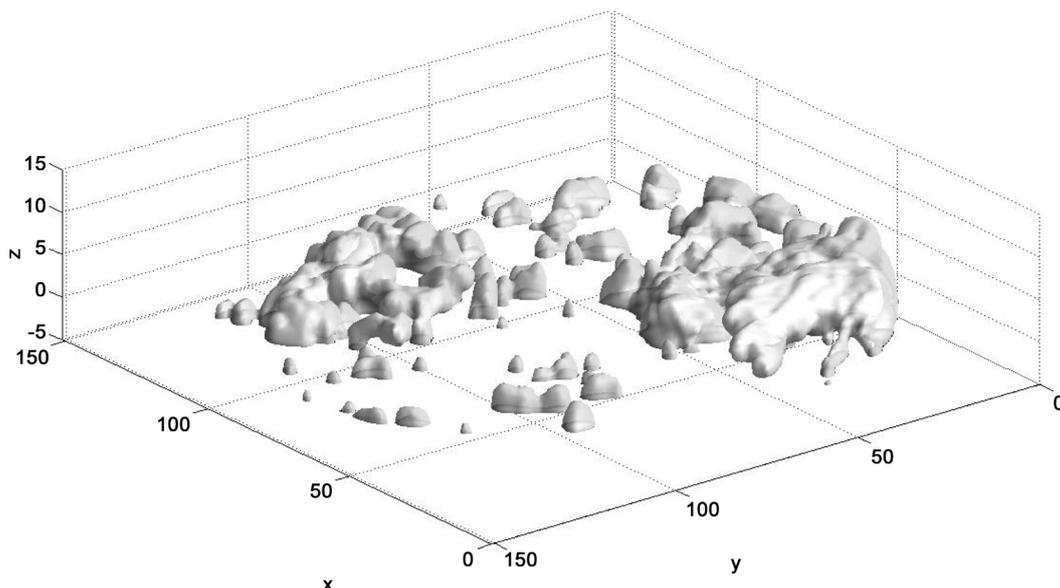
After the above steps, we can get a three-dimensional reconstruction map of weather radar data. Firstly, we can get triangle Grid of clouds data as shown in **Figure 7**. Then, we get isosurface map of cloud data.as shown in **Figure 8**.

### 3.3. Experiment and Results

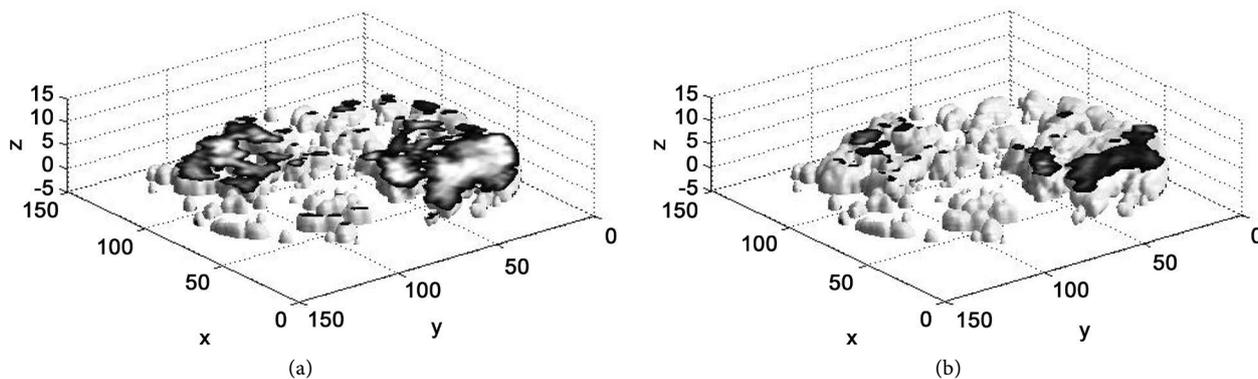
We got the cloud map that meteorological department need, by adding a certain realistic rendering to map that have been reconstruction with MC algorithm.



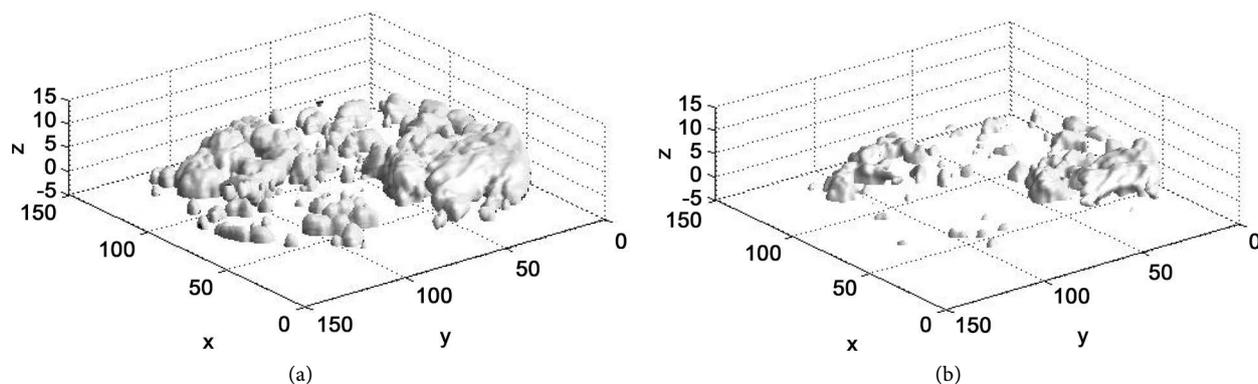
**Figure 7.** Triangular grid.



**Figure 8.** Isosurface map.



**Figure 9.** Cloud section map at different heights, echo intensity  $\geq 1$  dBZ. (a) Height = 4 km; (b) Height = 6 km.



**Figure 10.** Different echo intensity map, height = 10 km. (a) Echo intensity  $\geq 1$  dBZ; (b) Echo intensity  $\geq 40$  dBZ.

**Figure 9(a)** is cloud section map which the Kunming radar station on the ground height of 4 km was obtained at 18:23 on July 18, 2013. **Figure 9(b)** shows cloud section map on the ground height of 7 km for the same time. **Figure 10** shows the different intensity maps for the same time at 5 km. Weatherman can employ **Figure 9** and **Figure 10** to grasp and understand state about cloud, such

as thickness and intensity distribution of the cloud.

#### 4. Conclusion

In this paper, we use the proposed Cube Weighting Interpolation (CWI) and MC algorithm to implement the 3D cloud image. Due to that the weather radar data are distributed on some cones, we need to transform the data into the slice image data at first. Using CWI to project the raw radar data into the cubes, we obtain the equally spaced cloud slice images, then employ MC algorithm to reconstruct the 3D weather image. Some reconstruction results show that the proposed method has a better effect and more simple operation, which provides a reference for the simple realization of 3D surface reconstruction of meteorological images and also provides a basis for the realization of dynamic 3D meteorological images.

#### Acknowledgements

This work has been supported by Chinese National Science Foundation Grant No. 11161055.

#### References

- [1] Elias, M. and Kebisek, M. (2010) An Overview of Methods for 3D Model Reconstruction from 2D Orthographic Views. *Innovation in Information Technologies: Theory and Practice*, Dresden, 1-5.
- [2] Jiang, D. and Jiang, D. (2001) Overview of Camera Calibration and 3D Reconstruction for Computer Vision. *Computer Engineering and Applications*, **13**, 53-55.
- [3] Höhne, K.H., Bomans, M., Pommert, A., Riemer, M., Schiers, C., Tiede, U., *et al.* (1990) 3D Visualization of Tomographic Volume Data Using the Generalized Voxel Model. *Visual Computer International Journal of Computer Graphics*, **6**, 28-36. <https://doi.org/10.1007/bf01902627>
- [4] Liang, C.G. and Huang, Y.X. (2007) Weather Radar Data 3D Visualization Based on Particle System. *Computer Knowledge and Technology*, **2**, 518-527.
- [5] Li, L., Wan, W., Li, X. and Wang, Z. (2011) Weather Phenomenon Simulations in 3D Virtual Scenes Based on OSG Particle System. *1st International Communication Conference on Wireless Mobile and Computing*, Shanghai, 14-16 November 2011, 254-257. <https://doi.org/10.1049/cp.2011.0885>
- [6] Lu, Z.Y., Jin, X. and Han, C.Y. (2013) 3D Reconstruction of Strong Convective Cell Based on Doppler Radar Base Data. *International Conference on Machine Learning and Cybernetics. IEEE*, **2**, 845-849.
- [7] Guan, L., Wei, M., Wei, K. and Wang, J. (2015) 3D Visualization of Doppler Radar Data Based on MATLAB Platform. *Journal of Henan Normal University*, **43**, 43-48.
- [8] Han, Y., Liu, X., Lu, X. and Li, H. (2016) The 3D Modeling and Radar Simulation of Low-Altitude Wind Shear via Computational Fluid Dynamics Method. *Integrated Communications Navigation & Surveillance*, 4D3-1-4D3-9.
- [9] Y.J., Yu. and Li, Y. (2016) Method for Detecting and Simulating 3D Turbulence Field of Airborne Weather Radar. *Systems Engineering & Electronics*, **38**, 293-297.
- [10] Lorensen, W.E. and Cline, H.E. (1987) Marching Cubes: A High Resolution 3D

- Surface Construction Algorithm. *ACM SIGGRAPH Computer Graphics*, **21**, 163-169. <https://doi.org/10.1145/37402.37422>
- [11] Delibasis, K.S., Matsopoulos, G.K., Mouravliansky, N.A. and Nikita, K.S. (2001) A Novel and Efficient Implementation of the Marching Cubes Algorithm. *Computerized Medical Imaging & Graphics*, **25**, 343-352. [https://doi.org/10.1016/S0895-6111\(00\)00082-3](https://doi.org/10.1016/S0895-6111(00)00082-3)
- [12] Rajon, D.A. and Bolch, W.E. (2003) Marching Cube Algorithm: Review and Trilinear Interpolation Adaptation for Image-Based Dosimetric Models. *Computerized Medical Imaging & Graphics*, **27**, 411-435. [https://doi.org/10.1016/S0895-6111\(03\)00032-6](https://doi.org/10.1016/S0895-6111(03)00032-6)
- [13] Newman, T.S., Byrd, J.B., Emani, P., Narayanan, A. and Dastmalchi, A. (2004) High Performance SIMD Marching Cubes Isosurface Extraction on Commodity Computers. *Computers & Graphics*, **28**, 213-233
- [14] Newman, T.S. and Yi, H. (2006) A Survey of the Marching Cubes Algorithm. *Computers & Graphics*, **30**, 854-879.
- [15] Ernvik, A. (2002) 3D Visualization of Weather Radar Data. Department of Electrical Engineering.



Scientific Research Publishing

**Submit or recommend next manuscript to SCIRP and we will provide best service for you:**

Accepting pre-submission inquiries through Email, Facebook, LinkedIn, Twitter, etc.

A wide selection of journals (inclusive of 9 subjects, more than 200 journals)

Providing 24-hour high-quality service

User-friendly online submission system

Fair and swift peer-review system

Efficient typesetting and proofreading procedure

Display of the result of downloads and visits, as well as the number of cited articles

Maximum dissemination of your research work

Submit your manuscript at: <http://papersubmission.scirp.org/>

Or contact [jcc@scirp.org](mailto:jcc@scirp.org)