

DEM Data Based Stratigraphic Visualization Modeling

Qianlin Wu¹, Dandan Zhu^{1,*}, Chenchen Wang¹, Zhihui Ye², and Dong Chen²

¹College of Computer Engineering, China University of Petroleum – Beijing, China

²College of Petroleum Engineering, China University of Petroleum – Beijing, China

*Corresponding author

Abstract—We proposes a three-dimensional visualization modeling method of multilayer stratum by the DEM data, including the longitude, latitude and depth information. The interactive environment is mainly supported by a Python-embedded plotting library, Matplotlib. We adopted the DEM data to create the upper and lower surfaces of each stratum, which is the core component for stratum construction. To reinforce the visual perception of three-dimensional effect, we chose the appropriate angle of light and shaded the surfaces with geophysical map. The challenging point of this research is located in the creation of the solid between adjacent surfaces. Our solution is to extract the surface edge points according to the data storage structure, and generate 4 side surfaces for each stratum based on the polygon plotting principle. Consequently, a three-dimensional visualization model of multilayer stratum has been created.

Keywords—3D visualization; Matplotlib; DEM(Digital Elevation Model); multilayer stratum

I. INTRODUCTION

The techniques of terrain visualization has been evolving with the development of geographic information systems since the 1960s. The terrain expression can be traced back to the pictographic symbols, the perspective of writing the scenery, the topographic maps of the contour lines in the 17th century, and then, the aerial images of the 1940s. Under the assist of the fast-developing computer graphics and information technologies, the terrain expression has undergone an essential change. Especially, the Digital Elevation Model (DEM) and Geographical Information System (GIS) technologies encouraged the emergence of the terrain visualization technique [3,4]. The terrain visualization based on the DEM [2] or Digital Terrain Model (DTM) stems from computer graphics theory, and it has been an enduring hot spot in both industry and academia for nearly half a century.

Digital Elevation Model (DEM) is an important part of the spatial database with massive data. In real-world application, it is necessary to extract DEM-derived terrain factor data from various regions. Extracting a terrain factor from ArcGIS requires a lot of parameters and paths to be set and the data to be processed in a single directory. This is a waste of time and errors in the processing of large amounts of data. [7]. However, there are few studies on how to quickly and accurately extract DEM topographic factors in batches. Therefore, the modeling method adopted in this paper improves the processing speed and accuracy of the terrain factors to a certain extent, and improves the processing efficiency for complex DEM data.

A number of excellent 3D modeling and visualization soft wares have been developed, for example, GoCAD, LYNX, EarthVision, Ctech, Micromine, MineMap, Vulcan and 3D Earth Modeling. However, most are developed based on VC ++, generally requiring a long development cycle and large code workloads. To improve the efficiency of system development and the robustness of system processing capacity, we propose a 3D geological modeling method based on Python, and use Python library data to simulate formation parameters for visualization [14,15].

II. FRAMEWORK OF MODELING METHOD

The proposed modeling method uses the DEM data to construct stratum in the interactive environment supported by the Matplotlib embedded in Python.

DEM data is composed of a series of three-dimensional vectors $\{X, Y, Z\}$, where (X, Y) represents the plane coordinate while Z represents the elevation[6]. Although the DEM data is built based on the discrete spatial points, the meshed data is dense enough to simulate the continuity of terrain.

Modeling the multilayer stratum based on the DEM data, we consider the upper and lower surfaces of each stratum as the mainstay of the whole model. We utilized neighboring points sampled from the DEM data to construct plenty of adjacent spatial triangles and splice them together into a surface. It is notable that the surface approximation follows the rule of the Delaunay triangulation, that is, each triangle that makes up a triangle does not contain any data points. Because the Delaunay triangulation rule ensures that triangles in the mesh satisfy approximate equilateral angles, the DEM data bring the uniqueness of Delaunay triangulation.

In order to enhance the visual effect of the stereoscopic stratum surfaces, we conducted a series of visual processing on the drawn layer, including the light angle selection and geophysical texture shading.

After creating the upper and lower surfaces of each stratum, the framework of the stratum model has been established, and a common guideline is to simply fill the space between every two neighboring surface and form the stratum solids. However, both the Python built-in plotting libraries and the Matplotlib do not provide any direct function for such spatial filling. Therefore, we chose another way to construct the stratum solid. Because the points on the surface edge can be extracted according to the data storage structure, we could use them to generate the 4 side surfaces for each stratum, and with the upper and lower surfaces,

a layer of stratum could be consequently constructed. Then Give the framework of the whole modeling method as figure I.

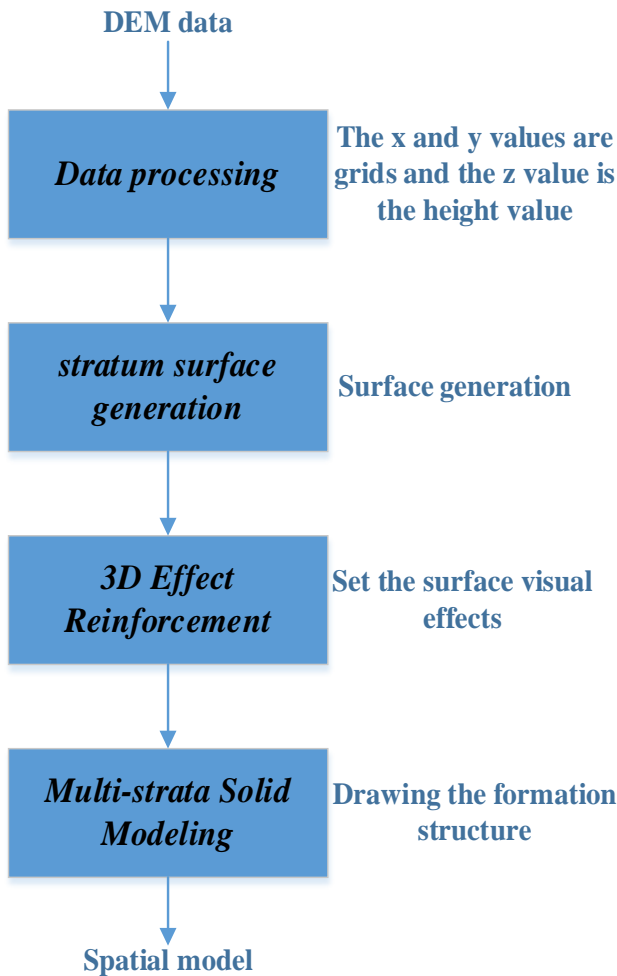


FIGURE I. SYSTEM FRAMEWORK

III. STRATUM SURFACE GENERATION AND 3D EFFECT REINFORCEMENT

In this paper, we use the built-in Matplotlib model in Python to realize the visualization of 3D terrain [9]. The basic process of 3D terrain visualization is to collect the DEM data and then process the data (segmentation and grouping), then construct the model, and then through the correlation function Parameter settings, including light calculation, projection transformation and texture mapping, the final realization of the three-dimensional terrain display, the following will focus on analysis of multi-layer geological model rendering.

A. Data Processing

The most common form of DEM is an elevation matrix or a regular rectangular grid [8], which can be obtained by interpolating irregular or regular spatial data points. Another manifestation of DEM is the irregular triangular network (TIN), which is a terrain model that uses a series of consecutively adjacent irregular triangles with indeterminate X and Y coordinates. In contrast, the TIN model is better able to take into

account the geomorphological features, allowing more information to be collected in areas with complex terrain, with less data redundancy and less distortion due to simplified data in this paper, the establishment of three-dimensional terrain model is the use of irregular triangular network (TIN) [10].

Since the original spatial data is discrete and irregular, in order to get the DEM data in the form of grid, we used the appropriate interpolation method for interpolation. There are several algorithms that could be used for grid interpolation, such as inverse method, Kriging method, surface spline function method, multiple quadric surface method and so on [10].

In this paper, we read the three-dimensional array as the DEM data, which defines the z value as the altitude, while the value of the x and y cut, will be divided into minimum and maximum value as the maximum row and column; and grid function in the drawing surface to make a grid [12].

Through the data gridding, the processed data is obtained, with the z value corresponding to the grid of x and y values as a data point for later plotting.

B. Stratum surFace Generation

Based on the resulting formation data, we can construct a three-dimensional surface of the formation:

- 1) After the data x, y, z are arranged in descending order, the matrices of the data x and y are meshed, and the coordinate axes are divided into a plurality of grids;
- 2) The z value as the height value and corresponding to each grid, by interpolation connected to each sub-plane, to make a mesh surface layer;
- 3) Repeat the above steps until the entire surface layer is drawn;

Then we obtained the stratum surface layer in Figure II:

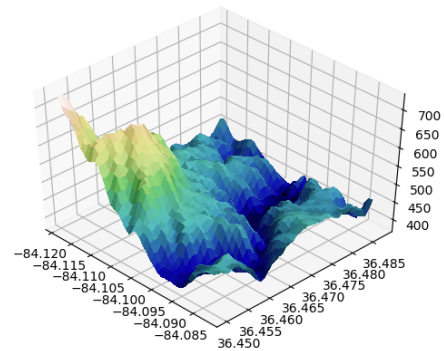


FIGURE II. STRATUM SURFACE WITH 3D EFFECT

As shown in figure, each grid of x and y values has a corresponding z value as the height value, and all the points are interpolated to obtain the surface map of the strata, and the stratigraphic image we obtained at this time is the upper surface layer of the terrain, and the lower surface geological layer was to make according to the same drawing method [13].

C. 3D Effect Reinforcement

After drawing the surface image, the visual effect of the surface image need to be set. By calling the material function of self-building surface textures in Python, we can render the surface texture of the surface layer and obtain the surface of the geological layer. At the same time, we optimized the light source effect, which is most suitable for the direction of 45 ° direction. Optimized the shadow effect, so that the surface of geographic information can be more intuitive observation by the reader.

IV. MULTILAYER STRATUM MODELING

In the upper and lower surfaces of the space between the upper and lower surfaces, are the corresponding geological layer. In order to construct the entity between two adjacent surfaces, we adopt the method of complementing of the geological layer between the two surface layers.

In order to ensure the integrity and accuracy of the stratum structure, different geological formations between the upper and lower surfaces need to be visually modeled. Because there is no specific data between layers to distinguish between the specific structure, and the Python library file does not have a fill-in rendering function, so we need to use the edge of the sample data for each layer, the use of the sampled edge data to re-draw it covers the four directions of observation at the border. However, after the data x and y are gridded, the formation data is stored in the form of matrix. At this time, the geological structure layer can not be rendered by conventional methods. Because of the characteristics of data storage at this time, the difficulty is greatly increased.

As can be seen from the above, the terrain data is stored in the form of a three-dimensional matrix, and x, y and z are 45 data matrices arranged in descending order. After gridding the data, the two matrices x and y must be the same number of rows and columns, and the numbers of rows in x and y are equal to the total number of elements in the input parameter y, and the numbers of columns in x and y are both Equal to the total number of elements in the input parameter x.

It is also worth noting that the different data storage and retrieval sequences after the successful extraction of the edge data can also have a dramatic effect on the rendering of the spatial structure. Drawing the data points in reverse order will result in the resulting image being drawn Mirror representation in the original drawing surface.

The main method used in this article is:

- 1) According to the characteristics of data storage, known data are arranged in descending order, extracting only the maximum and minimum x-edge y-value and z-value;
- 2) The above surface edge data as a starting point, extracted from top to bottom, from maximum to minimum of the top surface, from minimum to maximum of the bottom surface to re-save the data;
- 3) Use this data as a stratum structure surface and set its visualization as described above.;

Using the above principle, through the sampling of the boundary data can be completed between the upper and lower surface of the spatial structure of the drawing. With the Verts

function, the extracted boundary data is taken as the vertex. However, due to the difference in the order of the data storage structure, in particular, the side surface geological structure is drawn separately from the positive sequence of the upper surface to the lower surface. The images of the four observation planes are obtained by analogy and covered on the upper and lower surfaces. Taking the boundary values of x and y separately, we get the filling images of the other two observation surfaces, and then get the complete geological layer. The complete terrain image is shown in Figure III

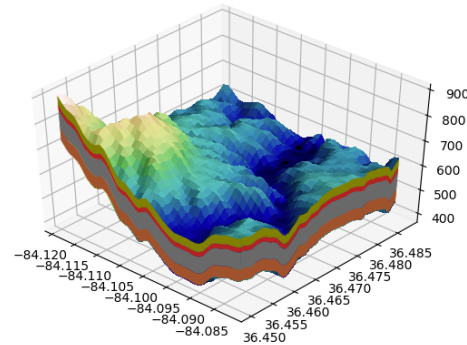


FIGURE III. MULTILAYER STRATUM MODEL

Then the structure of four different Stratum surface has been modeled in the upper and lower strata. The thickness of the strata is 35,20,70,50 from top to bottom; it can be seen that the boundaries between each layer are clearly visible and easy to observe.

V. CONCLUSIONS

As can be seen from the figure above, the basic topography and geological information have been visualized, and the boundary between geological layers has been clearly marked.

The basic terrain visualization has been realized through the Matplotlib library on the Python language. Thanks to the simple and efficient Python language, dozens of lines of code can accomplish the basic goal of terrain visualization. This shows that this method is completely feasible for visualizing the terrain Completion is quite high, but it is worth noting that there are still some problems that require the reader to think and solve:

- 1) This method basically implements terrain visualization, but it can not be added to the legend yet and multiple legends need to be manually added;
- 2) Although the method can visualize the DEM data but still need to pre-process the data, it may be abnormal;
- 3) It has not been possible to verify whether this method works for multivariate data;
- 4) Failed to complete the code package.

The visualization of the terrain, especially the large-scale, irregular terrain visualization, is a challenging task. Although many predecessors have done a lot of work in this area as discussed above, it is truly practical and suitable for a wide range

of terrain There are not many algorithms for visualization. Terrain visualization is of far-reaching significance to geological prospecting and oil exploration. Therefore, it is not only an industry demand but also an inevitable guide for the development of the industry to realize the terrain visualization more efficiently, conveniently and intuitively.

Python is rapidly becoming a programming language of increasing interest to data scientists. It provides a very broad programming language ecosystem and a scientifically calculated library of good computational power. Scientific ecosystem and data analysis libraries available to Python developers, easy to integrate with almost any other technology, and open source. Terrain visualization based on the Python language has unique advantages. As the development of the language progresses, a new round of revolution is bound to take place in the industry.

ACKNOWLEDGEMENTS

This study was financially supported by the Strategic Priority Research Program of the Chinese Academy of Sciences, Grant No. XDAXX040402, and Science Foundation of China University of Petroleum, Beijing(No.2462015YJRC008)

REFERENCES

- [1] ZHU Hai-Jin, WENG Wei-Lin. Data Extraction Based on Digital Elevation Model of Python [J]. *Geospatial Information*, 2013,11 (05): 136-138 + 14.
- [2] Wang Chun, Wang Zhanhong, Li Peng, Li Weitao. Technology of DEM Terrain Visualization Self-enhancement [J]. *Journal of Geographical Information*, 2009,7 (01): 38-45.
- [3] Wang Yongming. Terrain Visualization [J]. *Journal of Image and Graphics*, 2000, (06): 4-11.
- [4] Bian Haihong, Wang Feng. Research and Implementation of Terrain Visualization Based on 3D GIS [J]. *Computer Technology and Development*, 2006, (07): 230-232 + 235.
- [5] Li Baoyuan. Application of Matplotlib in Visualization of Calculation Results [J]. *Modern Computer*, 2007, (01): 81-84.
- [6] Wang Linxu, Li Sikun, Pan Xiaohui. A Real-time Visualization of Dynamic Terrain [J]. *Journal of Computer China*, 2003, (11): 1524-1531.
- [7] Yang Gang. Review of Three-Dimensional Terrain Visualization [J]. *Journal of Jilin Building and Construction Engineering*, 2012,29 (03): 69-72.
- [8] WAN Cong. Research on 3D Terrain Visualization Based on RS and GIS [D]. China University of Geosciences (Beijing), 2009.
- [9] HE Hui-ming. DEM modeling of digital elevation model and its 3D visualization [D]. Southeast University, 2004.
- [10] Han Yuanli. DEM data processing in large area and its 3D visualization application [D]. Wuhan University, 2004.
- [11] Xiao Min, Chen Xing. Characteristics and Applications Based on Python Language Programming [J]. *Computer Information and Technology*, 2014,10 (34): 8177-8178.
- [12] Liu Zhenping. Visualization Method for 3D Modeling and Computation of Engineering Geology [D]. Graduate School of the Chinese Academy of Sciences (Institute of Rock and Soil Mechanics, Wuhan), 2010.
- [13] Xavier Bonaventura, Aleksandra A. Sima, Miquel Feixas, Simon J. Buckley, Mateu Sbert, John A. Howell. Information measures for terrain visualization [J]. *Computers and Geosciences*, 2016.
- [14] Alejandro Graciano, Antonio J. Rueda, Francisco R. Feito. Real-time visualization of 3D terrains and subsurface geological structures [J]. *Advances in Engineering Software*, 2017.
- [15] Peng Chen. Research of Terrain 3D Visualization and Data Transfer Services within Distributed System [A]. The Chinese Institute of Electronics (CIE). Proceedings of 2006 8th International Conference on Signal Processing (Volume II of IV) [C]. The Chinese Institute of Electronics (CIE), 2006:4.