A New Three-Dimensional Terrain Model Generation and Real Time Rendering Algorithm

Yang Bin

Department of Information Engineering, North China University of Water Conservancy and Electric Power, Zhengzhou, China, 450011, yangbin@ncwu.edu.cn

Abstract

In order to make full use of the topographic map with DWG format and construct 3D terrain model, a new 3D terrain model generation algorithm based on conversion technology is put forward. Firstly, by processing the topographic map with DWG format, we can get the standardized data which can be stored in spatial database. Secondly, the elevation points and the elevation data note text are matched automatically to generate the objects with 3D information. Thirdly, with discrete sampling technology, we can extract the 3D data points in the contour lines. Finally, with the VRML model, texture mapping, blending, the system based on OpenGL was accomplished. The effect of system can favorably meet the need of real time and realistic rending.

Keywords: 3D Modeling, ARX, Texture Mapping, Real Time Rendering

1. Introduction

With the rapid development of computer hardware and software technology, geographic information systems have been used more and more widely. For the massive data of terrain plane and elevation data, geographic information systems provide an effective means of management, storage and maintenance. However, it is a time-consuming work to generate terrain digital elevation model (DEM) data. At present, there are many commonly used methods of collecting DEM data, such as digital photogrammetry, ground survey, map digitizing and so on. Although these methods possess high accuracy, the cost of get these original data is too high which prevents the applications of geographic information systems to some extent. Currently, the terrain map data with DWG format based on AutoCAD are widely used. Therefore, we can generate the terrain DEM data from DWG format data. As the DWG and DEM are different data format, we need to design a conversion algorithm between the two types of data. With the DEM data, we can construct the wire frame model and surface model of three-dimensional terrain scene.

In view of the analysis above, an algorithm for vector conversion is designed based on ARX technology in this paper which possesses the performance of low cost and high precision. Finally, some simulation experiments are made to test the performance of this approach.

2. Organizational structure analysis of DWG format

2.1. Procedure call pattern

AutoCAD is a universal computer-aided design software which is widely used in construction, GIS machinery, electrical and electronics and other fields. AutoCAD provides AutoLISP, ADS, ARX and other embedded programming language which can be used to expand the function of AutoCAD. AutoLISP is an interpreted language that provides a simple mechanism for adding commands to AutoCAD. Although there is some variation depending on the platform, AutoLISP is logically a separate process that communicates with AutoCAD through inter process communication (IPC). AutoLISP communicates with AutoCAD by internal process communication (IPC). ADS applications are written in C and are compiled. However, to AutoCAD, ADS applications appear identical to AutoLISP applications. An ADS application is written as a set of external functions that are loaded by and called from the AutoLISP interpreter. ADS applications communicate with AutoLISP by IPC. The call relationship is shown in figure 1.
Object ARX (AutoCAD runtime extension) is a development package based on AutoCAD platform which provides C++ based object-oriented development environment and application program interface and can really access graphics database of AutoCAD quickly. Different from AutoLISP and ADS, Object ARX application program is a DLL (dynamic link library) which share address space of AutoCAD and can call AutoCAD functions directly. Therefore, the execution speed of application programs can be increased greatly.

2.2. Organizational structure of DWG based on AutoCAD

The graphic data file of AutoCAD (.DWG) is viewed as class objects and the objects are organized hierarchical structure (shown in figure 2). The relationship among level objects is inclusion and the root of the collection is object of graphical database (AcDbDatabase). The second layer consists of blocks table, layer table, data dictionary, group and Xrecord and so on. The third layer consists of block table records, layer table records and other symbol table records. Other objects are included in the named object dictionary. The fourth layer consists of the specific content of all the entities.

2.3. Conversion technology roadmap and platform

In order to generate terrain DEM data, firstly, we should realize the data format conversion between graphic data and attribute data. In this paper, the conversion process is implemented by MFC, ARX and database technology. The conversion principle is as follows: Firstly, extract the attribute information of each basic entity with ARX and then write this information into database with ADO and ODBC. In this way we can realize the conversion from graphic data to attribute data. Conversely, read attribute data from database and then drawing the vector graphic with ARX technology. In this way we
can realize the conversion from attribute data to graphic data. The conversion process is shown in figure 3.

![Diagram of conversion process between graphic data and attribute data](image)

**Figure 3.** Conversion process between graphic data and attribute data

### 3. Design and Implementation of Vector Conversion Algorithm

In this section we will discuss the principle of conversion algorithm from vector to DEM terrain data. The main contents include standardization of the terrain contours, extraction of elevation data and elevation data processing.

#### 3.1. Standardization of terrain contours

Some contours are generated by digitizer. Therefore, it is inevitable to bring about some data exception problems, such as inconsistent direction of contours, fold points, redundant points and so on. These data exception may cause the DEM data is incredible. So in order to get credible and accurate data, the data standardization is necessary to eliminate the data exceptions.

1. **Contour direction checking**
   Contours are closed curves and we assume that they have the same direction (clockwise or counterclockwise). For the vertexes sequence in the contour, processing algorithm is as follows:
   1) Choose any vertex in the contour and calculate the vector direction angle formed by the connection between the vertex and each vertex of the enclosed area. Then set the sum of the direction angles is \( \theta \).
   2) If \( \theta =360 \), the vertexes sequence of the contour is counterclockwise; If \( \theta =-360 \), the sequence is clockwise.
   3) Adjust the sequence based on the results.

2. **Fold points removing**
   Due to some unpredictable reasons, the contour can be like figure 4.

![Fold points contour](image)

**Figure 4.** Fold points contour

This problem can cause unpredictable results in the subsequent process. For this problem the processing algorithm is as follows:

1) Read the adjacent two vertexes \( (x_i, x_{i+1}) \) in the contour and the segment is named \( l \).
2) Check that whether $l$ intersects with other segments formed by all the adjacent two vertexes in the contour.
3) If $l$ intersects some segment, it shows that the fold points exist and need to be removed. Otherwise go to the fourth step.
4) Let $i = i + 1$, if the two vertexes $(x_i, x_{i+1})$ are same with the begin vertexes, the algorithm is over. Otherwise go to the first step.

(3) Redundant points removing

Redundant points have no affect on generating terrain contour. For example, if the contour is a straight line, then the vertexes between the two endpoints are redundant points. The emergence of redundant points not only increases the storage space of database, but also reduces the processing speed. Therefore, we should remove the redundant points if the contour is straight line. The processing algorithm is as follows:
1) Read the coordinates of the adjacent two points $(x_i, x_{i+1})$ and name the line $l$.
2) Check that whether the vertex $x_{i+2}$ is on line $l$. If so, remove the vertex $x_{i+1}$; otherwise go to the third step.
3) Let $i = i + 1$, if the vertexes $(x_i, x_{i+1})$ are the same with begin vertexes (read in firstly), the algorithm is over. Otherwise, go to the first step.

3.2. Algorithm of elevation data extraction and matching

The data structure of elevation points are formed by elevation point and elevation data note text. Generally, elevation point provides accurate position $(x, y)$ but sometimes there is no attribute value $z$. In this case the attribute value $z$ is written in the note text. If the elevation points in the DWG topographic have elevation attribute value, then the attribute value can be extracted directly and stored. Otherwise the elevation points need match with elevation note text. There is a solution to this problem with which we can obtain accurate elevation value of accurate location. Firstly, we can obtain the location data tables of all the points (including $x, y$). Then find out the elevation data tables (including $x, y, z$). Finally, realize the data match between the two tables. The specific method can be like this: the location data $(x, y)$ of points and elevation value $z$ are stored in two arrays. Starting from the first vertex, find out the elevation note text of which the distance with it is less than a given value in the elevation data. The text note is the elevation value of this vertex. Then the elevation text note should be deleted to reduce calculated amount in the subsequent processing. The algorithm for extracting elevation data and matching from the terrain contour is shown in figure 5.
In the match processing, taking the elevation point to be matched as center, traverse all the elevation note text and calculate the distance from the elevation point to all the elevation note text. Usually, if the minimum distance is less than two meter, we can regard the elevation value marked by the elevation note text as the elevation value of the vertex to be assigned.

### 3.3. Contour discretization

As elevation points, the contour is the important three-dimension data source. Effective extraction for contour lines is very important for generating DEM data. In this paper we use discretization method to extract three-dimension data points in the contour lines. Firstly, for the selected contour line, we should check that whether it has elevation note text. If the contour line has elevation information, then it will be discretized directly and the eligible points will be extracted from the contour line. Otherwise, the elevation value of the contour should be input manually. And then the discretization operation can be carried out. In the discretization operation how to control the density of selected points is very important. If the density is too low, then the accuracy of DEM data to be generated can not meet the requirements. Otherwise, if the density is too high, the data volume will be too large for calculation, storage and management. Therefore, in order to control the selection density, a threshold (the distance between two feature points in the contour line) is defined. The extraction algorithm for feature points in the contour line is shown in figure 6.
4. DEM data generation and analysis

From the mathematical point of view, elevation $z$ is a continuous function of the two arguments $(x, y)$ in the plane coordinate system. The mathematical functional expression is as follows:

$$ z = f(x, y) \quad x, y \in DEM \text{ region} $$

Since the observation methods and data acquisition means are usually different, the DEM data can be divided into two types: grid data model and discrete data model. Grid data model divides the covered region into small regular grid and each grid has the same size and shape. The grid model realizes the 2D spatial orientation by row and column of the matrix elements and the third dimension is elevation value. The discrete data model expresses the plane coordinates of irregular distribution feature points and the third dimension is elevation value too. With grid data model, the contour lines, gradient, aspect, mountain shadows and basin outline can be figured out easily. In order to get the grid data model from discrete data model, it is necessary to carry on data interpolation. The commonly used methods include inverse distance weighting, kriging interpolation, minimum curvature method, multiple regressions and so on. This paper adopts kriging interpolation algorithm. To check the performance of DEM data generation algorithm, some experiments are made with real data. Figure 7 shows the terrain contour of some region (the scale of the topographic map is 1:500).
With the DEM generation algorithm proposed in this paper, we can obtain the grid data of this region. Then with the DEM data we can construct the surface model of the given region easily (shown in figure 8).

With the 3D visualization technology, the terrain DEM data model can be transformed into three-dimensional terrain model (shown in figure 9).
In order to construct realistic three-dimensional scene, some visual rendering techniques is necessary such as adding lighting in the scene and setting the scene surface texture and so on. For the scene texture mapping, in order to simplify calculation, we choose image texture which can be realized easily with “ImageTexture” node in VRML. The syntax of this node is as follows:

```plaintext
ImageTexture {
    url       # texture resource location
    repeats        # SFBool, this domain value specifies how to map the texture on the object surface.
    repeatT        # SFBool
}
```

Figure 7 shows the visual rendering result (texture mapping).

To test the effect of the algorithm proposed in this paper, some roaming simulation experiments are made on HP graph workstation (HP z4000). The standard configuration of workstation is as follows: two AMD Opteron™ 2.6GHz CPU, 6 GB ECC DDR memory, NVIDIA QUADRO FX6400 video card, Microsoft® Windows® Server2003 operating system and 1024×768 resolution. During the scene roaming, the average frame rate is shown in figure 11.

![Frame rate graph](image)

Figure 11. Statistics of scene rendering frame rate

5. Conclusions

Terrain scene modeling is an important component of the geographic information system. At present the topographic data are mostly DWG format. Therefore, it is necessary to generate DEM data from topographic Map with DWG format. A conversion algorithm is put out based on the analysis of the AutoCAD data structure in this paper. The experiment results show that the algorithm can not only
generate DEM data quickly with high accuracy, but also the conversion operation is simplified greatly. In the terrain scene modeling fields, the algorithm will have a broad application prospects.

6. Acknowledgment

This work is supported by the Heng Yang Joint Funds of The Hunan Provincial Natural Sciences Foundation of China (NO.11JJ9010) and the foundation of He'nan Educational Committee (No.12A520005).

6. References