Studying on Conversion of Oblique Photogrammetry Data Based on OSG and ARX

Ruan Ming 1,2, Xie Meiting 3,4, Liao Hua1, and Wang Fen3,4

1 Nanning Exploration Survey & Geo-information Institute, 530022 Nanning, Guangxi, China
2 Nanning Planning Information Technology Center, 530022 Nanning, Guangxi, China
3 Guangxi Key Laboratory of Spatial Information and Geomatics, 541004 Guilin, Guangxi, China
4 Guilin University of technology, 541006 Guilin, Guangxi, China
* Xie Meiting: xiemt95@163.com

Abstract. Oblique photogrammetry data processing is a hotspot of current urban 3D model application research. Aiming at the problem that oblique photogrammetry data cannot be directly loaded and displayed on AutoCAD, based on OSG 3D rendering engine and ARX secondary development technology, take the OSGB data into the DWG file as an example, this paper analyzes the oblique photogrammetry data format and the organization of the DWG model on AutoCAD and summarizes a conversion method of oblique photogrammetry data. The example shows that the method introduced in this paper can correctly realize the conversion of the oblique photographic data, and provides a reference for more industries to apply oblique photogrammetry data.

1. Introduction
Oblique photogrammetry is a high-precision aerial survey. Because of its high efficiency, low cost and other characteristics, the use of oblique photogrammetry to draw topographic maps has become more and more important[1,2], and oblique photogrammetry data processing has gradually become one of the hotspots in the field of oblique photogrammetry[3]. In recent years, researchers have carried out a lot of research on oblique data processing methods such as oblique model monomers and multi-source data fusion[3,4] and a number of oblique photogrammetric data processing software have been introduced in the market[5-9]. However, the related research on the application of oblique photogrammetric data to AutoCAD modelling is still less. Most of oblique photogrammetry data is stored in the OSGB format, if the OSGB data format can be extended, such as that the OSGB model is converted to a the DWG model, it will greatly promote the application of the oblique photography model on AutoCAD to meet the business needs of the land and planning departments, and better solve the problem of oblique photography data processing and application in actual production. OpenSceneGraph (OSG) is a high-performance, cross-platform application programming interface (API) developed based on OpenGL technology. There are a variety of library files in OSG, among which OSG read-write library (osgDB) adopts plug-in management architecture, which allows users to load, use and write 3D database, and supports a large number of common 3D graphics file formats[10]. ObjectARX (ARX) is a development software package developed by Autodesk for the
secondary development on AutoCAD platform. It provides an object-oriented development environment and application interface based on C++, and many of AutoCAD's own modules are also developed with ARX, which has the most powerful and highest performance, and can access the DWG files quickly and completely. The execution speed of functions programmed with ARX is greatly improved[11]. This paper will consider using OSG and ARX secondary development techniques to convert oblique photographic data.

2. Anatomy of the OSGB and the DWG model files

2.1 The OSGB model file
Generally speaking, there are multiple folders in the OSGB 3D model dataset. Each folder contains multiple data files in the OSGB format. Each OSGB file contains one root node (the type of PagedLOD) and the middle level node (the type of PagedLOD or Group) and the lowest node (the type of Geode). The middle level node contains the geometric information, the texture information of the model and the parent-child relationship between the upper and lower nodes. The lowest node contains only the geometry and texture information of the model. Essentially, every OSGB model is the mesh mode[12].

2.2 The DWG model file
The DWG file is an unpublished graphic format file introduced by Autodesk, which consists of points, lines, faces and other entities to form a CAD graphics database[13]. The content of the DWG file is divided into five parts — HEADER, ENTITIES, TABLES, BLOCKS and CONTINGENCY HEADER. Most of the information of the graph is in ENTITIES, so for the conversion from the OSGB model to the DWG model, how to create the data of the physical part of the DWG is the focus.

3. The conversion of OSGB into DWG

3.1. The OSGB model parsing based on OSG
The OSGB model data generated by oblique photography mainly involves two aspects of data: model data, including vertex coordinates, face index and normal; texture data, including texture image and texture coordinates. According to the characteristics of the OSGB format, this paper completes the OSGB model data analysis for the above two aspects.

3.1.1 Extraction of model data
The OSG mainly includes three basic class nodes: Node, Geode, and Group. The file read/write library (osgDB) in the OSG can be used to call the readNodeFile function to load the OSGB model data into the Node class, then design vertex accessor and texture accessor, inherit both NodeVisitor classes, and overload the apply function, traverse the entire OSGB model scene function and call the function of the accessed child node, and the Node, Geode, StateSet nodes are processed in turn. Geode inherits from the Node, which contains geometry information for managing geometric primitives. The analysis of model data is mainly for this node.

3.1.2 Extraction of texture data
Texture data is the basis for the true expression of the oblique model. Correct interpretation of the texture ensures high fidelity of the converted DWG model. In the OSGB model, Texture2D manages the texture objects in the scene, and uses Image to manage the pixel data of the image. If you want to use 2D image files as texture graphics, you must assign the file name to the Image object and associate the Image to Texture2D. Because the Geometry object stores vertex and vertex attribute data (including primitive vertices, vertex colours, vertex associations, normal, texture coordinates, and other basic information) in an array, you can map vertex arrays to colours, normal by array indexes, or an array of texture coordinates. Considering that the texture coordinates of the OSGB model are
different from the Y-axis reference point of the DWG model, it is necessary to convert the texture coordinates when caching the texture coordinates. When parsing the texture coordinates, the parsed texture coordinates need to be calculated according to the formula. The relationship between the two texture coordinates is as shown in equations (1) and (2), where the subscript d represents the texture coordinates in the DWG model, and the subscript o represents the texture coordinates in the OSGB.

\[
X_d = X_o
\]  \hspace{1cm} (1)

\[
Y_d = 1 - Y_o
\]  \hspace{1cm} (2)

Texture data is parsed into 2D image, thus completing the OSGB data parsing work. The OSGB file parsing process is shown in Fig.1.

Fig. 1. The parsing process of the OSGB file.

According to the above analysis, the conversion steps of the OSGB model are as follows:

1. Design two accessors that inherit from the NodeVisitor class — GeometryVisitor and TextureVisitor.
2. Pass a specific accessor object to the node and traversing the entire OSGB model scene.
3. The parsing node contains data from the Drawable object.

3.2 Reconstruction of DWG 3D model based on ARX

The OSGB model generated by oblique photography is essentially a mesh model, which is rendered in a triangular patch plus texture. The data organization is similar to the data type AcDbSubDMesh in AutoCAD. AcDbSubDMesh can create 3D models composed of multiple triangle faces and support texture mapping. Using chapter 3.1 parsed model data and texture data, according to the creation method of AcDbSubDMesh, the triangle patch model is organized, and the mesh is created as an entity to complete the DWG model reorganization. Thus, the transformation from the OSGB 3D model to the DWG 3D model is realized. The DWG 3D model reorganization method is shown in Fig.2.

Fig.2. The main reconstruction process of the DWG 3D model

The main reconstruction steps are as follows:

1. Create an AcDbSubDMesh object (meshObj) by using vertex data (pVertexPnts) and its corresponding face index data (pFaceIdx).
2. Create an AcDbMaterialobj material object (Material) by using a texture image.
(3) Set the Texture coordinates (setVertexTextureArray (…)) and the normal (setVertexNormalArray(…)).

(4) Assign the material Material to meshObj.

4. Application example

The data source of this example is a block of data in the OSGB data by using the UAV to collect in Hangyang, Nanning City, and then using Smart3D to process. The block data belongs to the 18th level and contains buildings, trees, vehicles, ground, roads, etc. The method proposed in this paper is implemented in the Visual Studio 2015 environment. The hardware environment as follows: Dell OptiPlex 3020 desktop, I5 quad-core/four-thread CPU (clocked at 3.30GHz), memory is 4G, hard disk is 1000G. Software configuration as follows: The operating system is Microsoft Windows 10 Professional, AutoCAD 2017, and OpenSceneGraph 3.6.1.

The experimental data size is 803 KB, including 23309 vertex coordinates, 23309 texture coordinates and 14467 face indexes. The partial vertex coordinates, texture coordinates, and face index data of the model is respectively showed in Table 1, Table 2 and Table 3.

| ID | Vertex Coordinates |
|----|--------------------|
|    | x      | y      | z      |
| 1  | 14529.7 | 709.996 | 81.2608 |
| 2  | 14529.4 | 708.964 | 80.0068 |
| 3  | 14526.1 | 707.100 | 79.5166 |
| 4  | 14526.9 | 14526.9 | 14526.9 |
| ... | ...    | ...    | ...    |

| ID | Texture Coordinates |
|----|---------------------|
|    | u      | v      |
| 1  | 0.762734 | 0.541712 |
| 2  | 0.759709 | 0.541804 |
| 3  | 0.950821 | 0.303745 |
| 4  | 0.954089 | 0.305133 |
| ... | ...      | ...    |

| ID | Face Id |
|----|---------|
|    | a      | b      | c      |
| 1  | 4/4/   | 5/5/   | 6/6/   |
| 2  | 7/7/   | 8/8/   | 9/9/   |
| 3  | 10/10/ | 11/11/ | 12/12/ |
| ... | ...    | ...    | ...    |

In order to convert the OSGB model into a DWG 3D model automatically, this paper first uses OSG to parses the OSGB file to obtains the data of vertex coordinates, texture coordinates, face indexes, textures, normal and so on. Then the data is stored as a cache file, and the 3D model and texture mapping composed of several patches are re-created by ARX. Finally, the DWG 3D model file is obtained. The technology route is show in Fig.3.
Fig. 3. The technology route of the conversion of oblique photogrammetry data based on OSG and ARX

Firstly, the vertex coordinate data and the face index data are extracted from the OSGB. Then the data is reconstructed into an untextured mesh model by using ARX. Fig. 4(a) shows the pre-conversion model displayed in the wireframe usage on Acute3D Viewer and Fig. 4(b) shows the mesh model on AutoCAD 2017. It can be seen that the triangle face tissue composed of vertex data is basically consistent, which shows that the OSGB model can be correctly converted into an untextured DWG model by using this method of this paper.

(a) Before conversion (on Acute3D Viewer)

(b) After conversion (on AutoCAD 2017)

Fig. 4. Comparison of before and after convert the OSGB model into the DWG (in wire-frame mode)

Secondly, the texture image and texture coordinates are extracted. The texture data is stored in the system file as the 2D image and is used to create a model material by using ARX. Then the material and texture coordinates are applied to the mesh model. Finally, the DWG 3D model with real texture was successfully converted. Fig. 5(a) shows the display effect of the original OSGB model on Acute3D Viewer and Fig. 5(b) shows the display effect of the converted DWG 3D model on AutoCAD. It can be seen that the converted DWG 3D model has a consistent visual effect with the OSGB model.
5. Conclusion

This paper expounds secondary development technology of OSG and ARX, and describes the parsing method of the OSGB model and the reconstruction method of the DWG model based on the analysis of the organization structure of the OSGB 3D model and the DWG 3D model. Finally, this paper uses OSG and ARX to realize the method of automatically converting the OSGB model into the DWG model, and compares the data organization of the model before and after conversion to verify the effectiveness of the proposed method. Using this method, the model display effect before and after conversion has a high consistency, which can be loaded and displayed on AutoCAD 2017, and meets the accuracy requirements of the 3D model data used by the national land and planning department.

The method proposed in this paper is only for the analysis of a single OSGB file, but a complete oblique photogrammetry model is composed of multiple OSGB files. When displaying the 3D model of oblique photogrammetry, it also needs to consider the LOD display effect of the OSGB data. Therefore, further research and exploration are needed in the reality of LOD effect.

References

[1] S. Q. Xu, X. F. Huang, F. Zhang, X. L. Yong, Z. M. Xia, T. Wang. Bull Surv Map, (02),111-115 (2018).
[2] Y. Tian, Y. Xiang, F. Gao. Bull Surv Map, (02), 59-62+66 (2013).
[3] Y. Li, Z. Liu, N. S. Peng, Y. Y. Gong. Struct. Eng. Mech, 43(07): 103-108 (2018).
[4] B. H. Peng. Bull Surv Map, (S1):117-120+162. (2018).
[5] W. Wang, W. W. Huang, J. Zhen. Geomat Spat Inf Technol, 34(03):181-183 (2011).
[6] Y. Li, Z. J. Lin, G. Z. Su, Y. Yang. Sci Surv Map, 42(09):88-93 (2017).
[7] J. P. Huang. Mod Chem Res, (05):179-181 (2017).
[8] W. Xu, Y. Ren, X. Duan, W. Lei. International Conference on Electronic Information Technology and Intellectualization (ICEITI, 2016).
[9] Z. Q. Zhan, Y. H Li, X. Y. Gui. Bull Surv Map, (5):71-74 (2017).
[10] Wang R, Qian X Li. OpenSceneGraph 3.0 Beginners Guide, 263- 265 (2010).
[11] W. Zhou, X. X Dai. Bull Surv Map, (06):16-19. (2010).
[12] SuperMap 2015. SuperMap 7C (2015) SP1.
   https://wenku.baidu.com/view/c52f8a1783d049649b6658d5.html.
[13] F. Xiao, B. Q. Lv, X. Y. He, X. P. Lu. Bull Surv Map, (12):63-65 (2015).