VECTOR AND RASTER DATA LAYERED FUSION AND 3D VISUALIZATION

Yongsheng Huang[1], Guoqing Zhou[1,*], Tao Yue[1], Hongbo Yan[1], Wenxi Zhang[1], Xin Bao[1], Qiuyu Pan[1], Jinsheng Ni [2]

[1] Guangxi Key Laboratory of Spatial Information and Geomatics, Guilin University of Technology, No. 12 Jian'gan Road, Guilin, Guangxi 541004, China

[2] Spatial Data Service Center ,China Aerospace Science and Industry Corporation, No.8 Fucheng Road, Haidian District , Beijing 100048, China

KEY WORDS: Partial Fusion,Vector Raster Fusion,Morton Code, Oracle Database, Data Organization, 3D Visualization

ABSTRACT:

Although contemporary geospatial science has made great progress, spatial data fusion of vector and raster data is still a problem in the geoinformation science environment. In order to solve the problem, this paper proposes a method which merges vector and raster data. Firstly, the row and column numbers of the raster data, and the *X, Y* values of the vector data are represented by Morton code in the C++ environment, respectively. Secondly, we establish the the raster data table and the vector data table in the Oracle database to store the vector data and the raster data. Third, this paper uses the minimum selection bounding box method to extract the top data of the building model. Finally, we divide the vector and raster data into four steps to obtain the fusion data table, and we call the fusion data in the database for 3D visualization. This method compresses the size of data of the original data, and simultaneously divides the data into three levels, which not only solves the problem of data duplication storage and unorganized storage, but also can realize vector data storage and the raster data storage in the same database at the same time. Thus, the fusion original orthophoto data contains the gray values of building roofs and the elevation data, which can improve the availability of vector data and the raster data in the 3D Visualization application.

* Corresponding author: Guoqing Zhou; E-mail: gzhou@glut.edu.cn

1. INTRODUCTION

At present, there are two kinds of data structures commonly used in geographic information systems: vector data structures and raster data structures, and most of the other data structures are derived from these two data structures (Xie et al., 2009). Among them, integrated structure for raster and vector processing functions are ideal, providing maximum flexibility for data processing and analysis (Turk A.G.et al., 1992). However, there is still no suitable way to perfectly implement spatial data integration in vector data and raster data. Although, many researchers such as (Wu et al., 2017; Zhou et al. 2006; Li et al. 2017) study the storage and management of vector data and raster data in a unified spatial database, integrated vectors and rasters for 3D modeling of buildings and orthophoto hasn't realized in the analysis process and visualization process. Regarding data storage methods and organization methods, many researchers such as (Liang et al., 2012; Zhu et al., 2008; Gu et al., 2011; Wang ., 2004; Zhu et al., 2011) have done many research, but storage and organization issues are still a difficult problem to solve. Beside that, with the maturity of high-precision mapping technology for UAV(unmanned aerial vehicle) remote sensing true digital orthophoto maps (Zhou et al., 2008, 2009, 2016), the quality of TOM (raster data) is getting better. Memory for storing data has increased dramatically, and the effective reduction of the memory occupied by the data has become an urgent problem to be solved. Because vector data and raster data lacks organized storage, there are duplicate storage problems for the data. For example, (Zhou G., 2018) the same information of data is stored twice. In addition, many building vector data is only the outline data of the building. The data inside the top profile of the building is empty and there is no actual data, which greatly reduces the diversity of data usage. With vector and raster data for 3D urban data storage and analysis is becoming increasingly popular (Zhou G., et al. 2013, Guo J., et al. 2017), 3D visualization of vector and raster fusion data has also become an important research direction.

In order to slove the problem of data storage and vector raster data fusion, this paper proposes a method based on Morton code to layer and store vector raster data in Oracle Databas0e 12C. This paper studies the following four directions:

(1) The *X* value and the *Y* value of the three-dimensional building model data are represented by Morton code, and the storage operation is performed;

(2) The row number and column number of the raster data are represented by Morton code, and the storage operation is performed;

(3) Using the contour of the building to generate the smallest rectangular frame on the top surface of the building, and then extracting the top data of the building and performing the storage operation;

(4) Organize the vector and raster data into a table, and call and perform 3D visualization from the fused data table.

2. THE FUSION METHOD

2.1 The Conversion of Raster Data by Morton Code

In the orthophoto of *2n*2n*, the number of rows and the number of columns of image data are represented by *i* and *j*. The Morton code of the *i*-th row and the *j*-th column is represented by *m(i, j)*, and the following conversion relationship can be obtained by the row number and the column number:

$$m(\mathrm{i},\mathrm{j}) = \sum_{k=0}^{n-1}(\lfloor i / 2^{2k+1} \rfloor * \mathrm{mod}(2 * 2^{2k+1})) + \sum_{k=0}^{n-1}(\lfloor j / 2^{k} \rfloor * \mathrm{mod}(2 * 2^{2k})) \quad (1)$$

In formulas (1): $0 \le i, j \le 2n$, $\lfloor\ \rfloor$ presents down remainder arithmetic, and mod represents rounding operation. As shown in Figure 1, the gray area indicates that the number of rows and columns of the grid which rows is 3 and columns is 3, that is *i*=3, *j*=3. Use the formula (1) to get the Morton code representing 3 rows and 3 columns is 15.

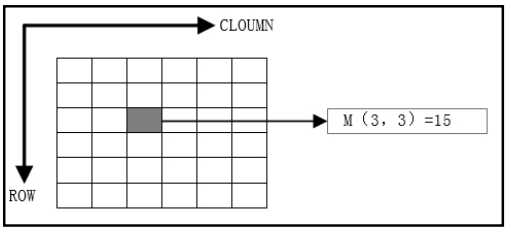

Figure 1.Schematic diagram of raster data to Morton code

Use the Morton code to represent the row number column number of each grid. The obtained data structure is shown in Table 1.

Table 1. Raster data table

Row	Column	Morton code	Grayscale
I1	J1	M1	G1
.....	.....	.....	.....
.....	.....	.....	.....
I20	J20	M20	G20

2.2 The Conversion of Vector Data by Morton Code

Figure 2 shows the relationship between vector data and raster data. Vector data is data based on a certain coordinate system. In the same coordinate system, the position of the row and column

expression and the position of the vector expression are the same. The row and column numbers *(i1, j1)* representing the vector points are then derived from equations (2) and (3).

$$i1 = 1 + \text{floor}((\text{Ymax-Y}) / dy) \quad (2)$$

$$j1 = 1 + \text{floor}((X - X\,\text{min}) / \text{dx}) \quad (3)$$

In formulas (2) and (3): *(i1, j1)* is the row number and column number indicating the original vector point *(X, Y)*. In formulas *dy=dx* is the resolution of the raster image, where *Ymax* is the maximum ordinate of the raster and *Xmin* is the minimum horizontal coordinate of the raster image. The function of floor is the rounding operation, its function is "rounding down" or "rounding down", that is taking the largest integer not greater than *x*. The function of the floor function is different from "rounding off". Rounding down is directly taking the left value closest to the required value on the number axis, meaning that the largest integer value is not greater than the required value. Then *(i1, j1)* is converted into a Morton code representing its position by equation (4), expressed by *m(x, y)*.

$$m(y,x) = \sum_{k=0}^{n-1}(\lfloor i1 / 2^{2k+1} \rfloor * \text{mod}(2 * 2^{2k+1})) + \sum_{k=0}^{n-1}(\lfloor j1 / 2^{k} \rfloor * \text{mod}(2 * 2^{2k})) \quad (4)$$

As shown in Figure 2, M2 and M1 are the maximum and minimum vector points of the orthophoto, where the resolution of the image is 1 meter, assuming M1(2,2), M2(10,10). The rectangular area Build1 (represented by the gray area) is the projection of the building on the orthophoto. A1, A2, A3, and A4 are corner points of the building, and the coordinates of the vector point A1 are *X*=2.5 and *Y*=6.5. Using equations (2) and (3), it can be concluded that the raster image row number where the A1 point falls is 4 rows and 1 column. Using the formula (4), the Morton code representing the vector point A1 is 33.

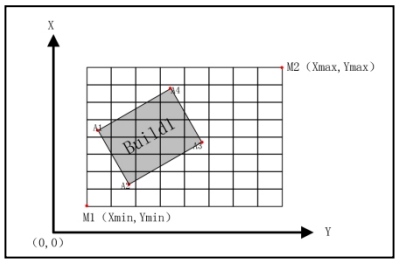

Figure 2. Schematic diagram of the relationship between vector data and raster data

Perform vector to Morton code processing on building data to obtain data table 2:

Table 2. Vector data table

Build ID	X value	Y value	Morton code	Height value
Build1	X1	Y1	M1	Z1
.....	.....	.....	.....	.....
.....	.....	.....	.....	.....
Build20	X2	Y20	M20	Z20

2.3 Extracting the Top Information of the Building

The building vector data is used to obtain all the *X* values and *Y* values of the projected contour of the building on the plane, and all the *X* values *Y* are compared to the maximum value *xmax* and the minimum value *xmin* and the maximum value *ymax* and the minimum value y*min* of the *y*. As the formula (5)-(8), the maximum values of X and Y are rounded up to *Xmax*, *Ymax*, and the minimum values of X and Y are rounded down to obtain X*min*, *Ymin*, where ceil is the upward rounding function, where floor is the rounding function. Rounding up and down allows the selection bounding box to contain all the top areas of the building, and the floating point data can be rounded up for data point to Morton code processing.

$$\text{Xmax} = \text{ceil}(\text{xmax}) \quad (5)$$

$$\text{Xmin} = \text{floor}(\text{xmin}) \quad (6)$$

$$\text{Ymax} = \text{ceil}(\text{ymax}) \quad (7)$$

$$\text{Ymin} = \text{floor}(\text{ymin}) \quad (8)$$

As shown in Figure 3 below, the selected bounding box A (*Xmin, Ymin)*B(*Xmax, Ymin*)C(*Xmax, Ymax)*D(*Xmin, Ymax)* can be calculated. Compared to calculating a building from an image, it is faster to extract the top data of the building from the minimum bounding box, and it only needs to compare the outline of the building with the data points in the minimum bounding box. Using the algorithm to get all *X*, *Y* values inside the bounding box ABCD separated by orthophoto pixels. Then use ray tracing method (Liu, 2008) to determine whether the point is inside the building outline *EFGI*. If the point is inside the building outline, we record the *X* value of the point, the *Y* value and convert the *X* value and the *Y* value into the row number and the column number. Position the row and column numbers to the raster pixels, and finally combine OpenCV to extract the gray value of the pixel.

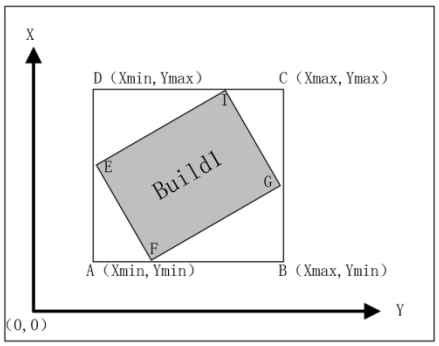

Figure 3. Schematic diagram of top data extraction from buildings

Finally, using equation (4) to convert the row and column numbers into Morton code, the data structure diagram of the following Table 3 is obtained.

Table 3. Building top data table

Build ID	Morton code	Height value	Grayscale
Build1	M1	Z1	G1
.....	.....	.....	.....
.....	.....	.....	.....
Build20	M20	Z20	G20

2.4 Hierarchical Fusion

In the same position, the Morton code representing the vector is the same as the Morton code representing raster. Combined with the building elevation, building ID, etc., this experiment divided the data fusion operation into four steps. The specific steps are as follows:

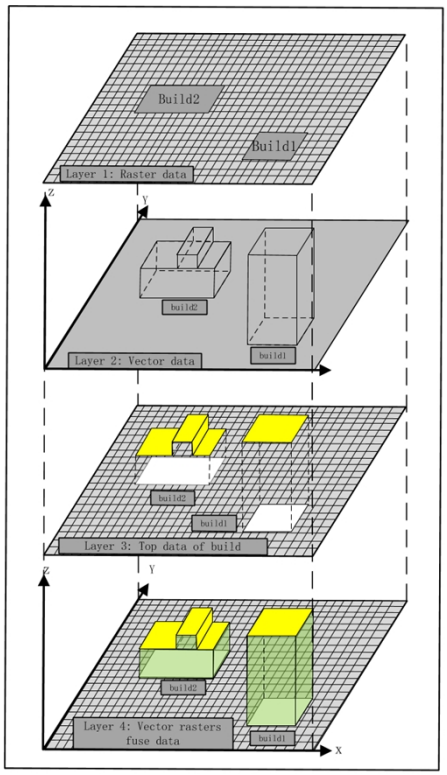

Figure 4. Schematic diagram of vector raster data layered fusion

As shown in Figure 4:

1. As shown in layer 3, the yellow area indicates the top data of the building, which obtained from the extracting the top information of the building. In order to eliminate the problem of data duplication storage, the data in the raster data table is retrieved using the Morton code in the top data of the building, and then deleted the repeated Morton code in the raster data table. Since the projection of the top surface data on the orthophoto image does not include the outline of the building, the Morton code in the building vector table is used to retrieve the data of the raster table, and the same Morton code is deleted. The repeated data is represented by the white area.

2. As shown in layer 4, after the duplicate data of the white area is removed, the data of the gray area will be stored. Because the data in the gray area part has no building data, it is identified as 0 in the building ID, and the data of this area is assigned the elevation at the same time. The data structure table as shown in Table 4 is obtained:

Table 4. Gray area data structure table

Build ID	Morton code	Height value	Grayscale
0	M1	Z1	G1

.....	.....	.....	.....
.....	.....	.....	.....
0	M20	Z20	G20

3. As shown in layer 4, the top data of the number of data buildings represented by the yellow area is directly stored into the top data structure of the building. Then we get the data structure table shown in the following figure:

Table 5. Data structure table represented by the yellow area

Build ID	Morton code	Height value	Grayscale
Build1	M1	Z1	G1
.....	.....	.....	.....
.....	.....	.....	.....
Build20	M20	Z20	G20

4. As shown in layer 4, the green area indicates the outline data of the building, where the top data of the building does not include the outline of the building. The data is stored in the database, and the data structure table shown in the following figure is obtained:

Table 6. Data structure table represented by the green area

Build ID	Morton code	Height value	Grayscale
Build1	M1	Z1	G1
.....	.....	.....	.....
.....	.....	.....	.....
Build20	M20	Z20	G20

3. APPLICATION

3.1 Raster Conversion of Morton Code and Storage

The OpenCV library is used to process grayscale images, read the row and column numbers of each raster and convert the row and column numbers into Morton codes. At the same time, read the grayscale values of each raster and output them to the TXT text to get orthophoto image**.** As the raster data TXT text has more than 22 million pieces of data, using the PLSQL(Database Upload Software) tool of the Oracle database to upload data to the database, which takes a long time and is not stable. This experiment uses the CMD(command prompt) command combined with the CTL file to upload data. Finally, the raster data table is obtained in the Oracle database. As shown in Figure 5, the PID is the data ID number, I is the line number, J is the column number, G is the gray value, and MD is the Morton code.

	RID	I	J	G	MD
1	1	0	0	0	0
2	2	0	1	0	1
3	3	0	2	0	4
4	4	0	3	0	5
5	5	0	4	0	16
6	6	0	5	0	17
7	7	0	6	0	20
8	8	0	7	0	21
9	9	0	8	0	64
10	10	0	9	0	65

Figure 5. Raster data table in oracle

3.2 Vector Conversion of Morton Code and Storage

For building 3D data, it is first read into the computer's memory, and then compiled in C++ language to load it into the database in an orderly manner. There are many C++ interfaces that can access Oracle databases such as ODBC and ADO, which are simple to use (Li G., 2012). We use the ADO interface, a comprehensive and easy-to-use API to connect the Oracle12C database to Visual C++. In the C++ environment, the *X*, *Y* values of the vector points are converted into Morton codes, and then the ADO interface is used to execute the SQL statement to perform the vector data insertion operation, and the table for storing the building data is obtained in the Oracle database (see Figure 6).

	PID	BID	X	Y	Z	MD
1	521	build1	1516.849976	402.000000	5330.192871	12311771
2	522	build1	1516.150024	412.000000	5330.192871	12311634
3	523	build1	1516.150024	414.000000	5330.192871	12310266
4	524	build1	1514.000000	415.149994	5330.192871	12310252
5	525	build1	1510.849976	411.000000	5330.192871	12311581
6	526	build1	1511.000000	409.000000	5330.192871	12311605
7	527	build1	1510.000000	407.000000	5330.192871	12311612
8	528	build1	1518.000000	396.850006	5330.192871	12312148
9	529	build1	1521.150024	393.000000	5330.192871	12312353
10	530	build1	1517.000000	387.850006	5330.192871	12312275
11	531	build1	1507.150024	378.000000	5330.192871	12317711
12	532	build1	1503.849976	379.000000	5330.192871	12317704

Figure 6. Building Vector Data table

3.3 Top Data Selection of Buildings

Use C++ to read the building vector data, and judge the point in the minimum selected bounding box. For the point where is within the top profile of buildings, OpenCV used to extract the gray value of the orthophoto image. And the (*x*, *y*) converted to The Morton code is processed into the database. The data structure diagram shown in Figure 7 is obtained, in which the Morton code and the Z (elevation value) are used to mark the spatial position of each pixel raster data, and the BID is the arrangement ID number of the building to facilitate the joint call with the vector raster fusion data.

	G	MD	BID	Z
1	0	11400747	build2	5239.348299
2	0	11400745	build2	5239.348299
3	0	11400739	build2	5239.348299
4	0	11400737	build2	5239.348299
5	0	11400715	build2	5239.348299
6	0	11400713	build2	5239.348299
7	0	11400707	build2	5239.348299
8	0	11400705	build2	5239.348299
9	0	11400363	build2	5239.348299
10	0	11400361	build2	5239.348299
11	0	11400355	build2	5239.348299

Figure 7. Building top data table

3.4 Layered Fusion of Vector Data and Raster Data

After getting the vector data tables, raster data and building top data table tables in the Oracle database, this experiment procedures performs vector and raster data fusion. Mainly in the Oracle database to execute SQL statements, this experiment has also used the ADO interface to connect the C++ environment to execute SQL statements. Because the long time to go into and out of the database, the operation of data in the C++ environment is taking a long time, executing SQL language in an Oracle database is faster than executing in a C++ environment. The following are the steps to perform the fusion:

1. Use the Morton code in the top surface and contour data of the building as the search condition, and set the elevation of the building outline (Z=5200m) to screen the outline of the bottom of the building. The SQL in the database is used to retrieve the data in the raster data table one by one. When retrieving data with the same Morton code, the data is deleted from the raster data table.

2. The data represented by the gray area is stored. This part of data is the data of the original orthophoto after deleting the white area data. This part of the data has no elevation and building ID. In this experiment, the building ID and elevation are assigned to this part of the data, which is convenient for other researchers to make data calls. The building ID is assigned a value of 0 to identify the part of the data does not contain building data. When the data is called, the building ID can be used to distinguish the building ID. The bottom elevation of this dataset is consistently (Z=5200m), and the data assignment elevation of this part of the dataset is 5200m.

3. The SQL language is used to directly copy the data in the top data table of the building into the library, but the top data of this part of the building does not include the contour data of the building. This article will deal with building contour data in the fourth step.

4. The data of the green area part in Figure 4 is stored. In this part of the data, the vector table has a Morton code corresponding to the raster table and the elevation value in the vector table is greater than the ground elevation value. Using the elevation value as the retrieval condition, the data with the elevation greater than the bottom elevation is retrieved in the building table. At the same time, the gray value of the data retrieved by the Morton code in the raster data table is assigned to the building outline data.

After performing the above steps, a fusion data table as shown in Figure 7 is obtained:

	BID	MD	Z	G
1638	build37	5821921	5592.200195	0
1639	build37	5821931	5592.200195	0
1640	build37	5867819	5592.200195	0
1641	build37	5866488	5592.200195	0
1642	build37	5866458	5592.200195	0
1643	build37	5865521	5592.200195	0
1644	build37	5865509	5592.200195	0
1645	build37	5865486	5200.000000	0
1646	build37	5821921	5200.000000	0
1647	build37	5821931	5200.000000	0
1648	build37	5867819	5200.000000	0
1649	build37	5866488	5200.000000	0
1650	build37	5866458	5200.000000	0
1651	build37	5865521	5200.000000	0
1652	build37	5865509	5200.000000	0
1653	build38	33908963	5251.299805	0
1654	build38	33908954	5251.299805	0
1655	build38	33908946	5251.299805	0
1656	build38	33909003	5251.299805	0

(a) Top and contour data structure of the building

	BID	MD	Z	G
97	0	5120	5200	0
98	0	5121	5200	0
99	0	5124	5200	0
100	0	5125	5200	0
101	0	5136	5200	0
102	0	5137	5200	0
103	0	5140	5200	0
104	0	5141	5200	0
105	0	5184	5200	0
106	0	5185	5200	0
107	0	5188	5200	0
108	0	5189	5200	0
109	0	5200	5200	0
110	0	5201	5200	0
111	0	5204	5200	0
112	0	5205	5200	0

(b) Raster data structure

Figure 7. Vector and raster Data table

3.5 Result Analysis and 3D Visualization

In order to verify whether the data can still maintain the characteristics of the original data after the conversion, the Morton code is compiled into x, y of vector. As shown in Figure 8, Figure 8.a is the original database, Figure 8.b is the data after the vector and raster fusion, Figure 8.c is the original building data in the database and the Morton decode to (x, y), after the data obtained from the database. Each row of data before the conversion of Figure 8.a and Figure 8.c corresponds to each row of data after conversion, and the results show that the error of X value and Y value before and after conversion is about 1 meter, which is in accordance with the standard to be reached in this experiment.

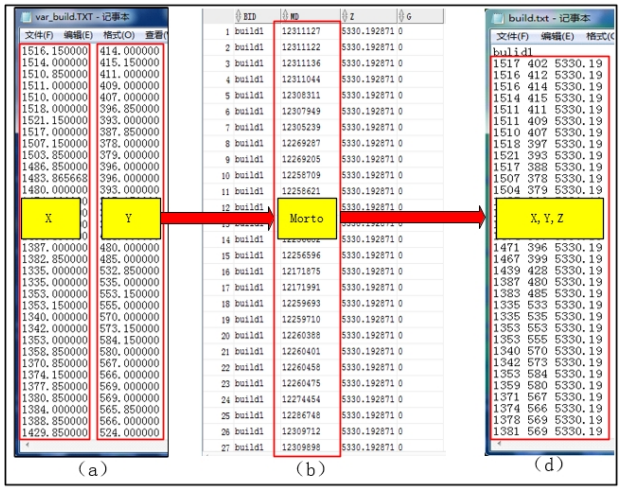

Figure 8. Schematic diagram of Data inspection

Input the Morton code interpretation parameters as shown in Figure 9.b, and call the data in the Oracle database in Figure 9.a (Figure 9.a is part of the data in the fusion data table) and interpret it. For example, the start and end ID numbers of 14 mean that the data of the No. 14 building is interpreted and retrieved. The No. 14 building outline data is extracted first, and then the top surface data is extracted also. At the same time, the data with the building ID of 0 is extracted, which is the underlying orthophoto data. Finally, the information of the data that does not select to display the three-dimensional building, such as the gray level and the position, is extracted from the data of the building outline data portion and the top surface texture portion to fill the blank of the orthophoto image. Then OpenGL is used to model it in C++ as shown in Figure 9.d, and Figure 9.c shows it in 2D. It can be seen in the 3D visualization of Figure 9.d that the fusion process uses a unified algorithm for conversion, so the stitching seam generated by multiple data calls in the display can be ignored.

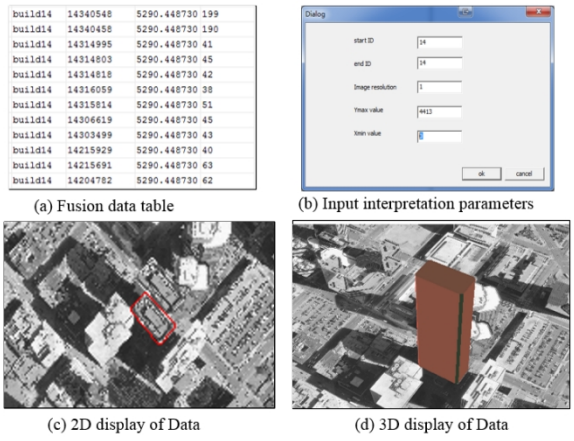

Figure 9. Schematic diagram of use the fusion data

4. CONCLUSION

This paper proposes a method of layering and merging vector raster data, in which the *X* value, *Y* value of the vector data and row number and column number of raster data are replaced by Morton code, which compresses the original data amount. At the same time, this paper tiers the data into the database, which not only solve the problem of data duplication and unorganized storage, but also stores all the information that the original data needs to express to the database. In addition, the original orthophoto data after fusion has elevation data, and the data at the bottom of the building model has gray values to improve the availability of data. Finally, this paper makes a call and 3D visualization of the fusion data, and proves the usability of the 3D modeling of the fusion data from the imaging effect.

ACKNOWLEDGEMENTS

This paper is financially supported by the National Natural Science of China under Grant numbers 41431179,41961065; Guangxi Innovative Development Grand Grant under the grant numbers GuikeAA18118038, GuikeAA18242048; the National Key Research and Development Program of China under Grant number 2016YFB0502501 and the BaGuiScholars program of Guangxi (Guoqing Zhou).

REFERENCES

David B., 1997. *The GIS Primer-An Introduction to Geographic Information Systems*. Pacific Meridian Resources, USA. Buckley.

Guo J., Sun B., Qin Z., Wong S., Man S., Chi W., et al. 2017. A study of plot ratio/building height restrictions in high density cities using 3d spatial analysis technology: a case in Hong Kong. Habitat International, 65, 13-31.

Gu S., Li J., Tian L., Cui S., 2011. Organization-Oriented 3D Spatial Data Model Organization Method. *Geospatial Information*,9(04):75-77+191.

Li C., Wu Z., Yin J., 2017. Research on Oracle-based Integrative Storage and Management of Spatial Data. I*nternational Conference*, 16-22.

Liang J., Li F., 2012. The Massive Data Organization Method in Urban 3D GIS. *Science and Mapping Science*, 37(06):91-93.

Turk A. G., Kuhn W., 1992. An Introduction to Geographic Information Systems (GIS). Wiley.

Liu D., Qi W., Lan X., 2008. Algorithm and Application of Inside and Outside Polygons Based on Reverse Ray and Vertex Degradation. *Science of Surveying and Mapping*, (04):84-86.

Wang Y., 2004. Research on storing GIS data based on object-relationship model. Zhejiang University.

Wu Z., Li C., Wu P., Sun J., Sun W., 2017. Integrated Storage and Management of Vector and Raster Data Based on Oracle Database. *Acta Geodaetica Et Cartographica Sinica.*

Xie Q., Zhang Z., Ravada S., 2012. In-database raster analytics: map algebra and parallel processing in oracle spatial georaster. ISPRS- International Archives of the Photogrammetry, *Remote Sensing and Spatial Information Sciences*, XXXIX-B4 (1), 91-96.

Xie S., Feng X., Wang J., 2009. A quadtree structure based on superior attribute storage and its construction algorithm. *Journal of Wuhan University*, 34(6).

Zhu G., Ma Z., Sun L., Li C., 2008. Database organization and management of massive data in urban three-dimensional geographic information system. *Science of Surveying and Mapping*, (01): 238-240+254.

Zhou G., Wang Y., Yue T., Ye S., Wang W., 2016. Building occlusion detection from ghost images. *IEEE Transactions on Geoscience & Remote Sensing,* 55(2), 1074-1084.

Zhou G., Xie M., 2013. Three-dimensional Morphological Change Analysis of Assateague Island National Seashore via GIS and Remotely Sensed Series Datasets. International Conference on Business Intelligence & Financial Engineering (pp.458-462). IEEE Computer Society.

Zhou G., Xie W., Cheng P., 2008. Orthoimage creation of extremely high buildings. *IEEE Transactions on Geoscience & Remote Sensing*, 46(12), 4132-4141.

Zhou G., 2009. Near real-time orthorectification and mosaic of small UAV video flow for time-critical event response. *IEEE Transactions on Geoscience & Remote Sensing*, 47(3), 739-747.

Zhou G., Tan Z., Cen M., Li C., 2006. Customizing visualization

in three-dimensional urban gis via web-based interaction. *Journal of Urban Planning & Development*, 132(2), 97-103.

Zhou G., 2018. VECTOR AND RASTER DATA STORAGE BASED ON MORTON CODE. ISPRS Technical Commission III on Remote Sensing. Proceedings of the ISPRS Technical Commission III Midterm Symposium on "Developments, Technologies and Applications in Remote Sensing". ISPRS Technical Commission III on Remote Sensing.

Zhu Q., Li X., Zhang Y., Liu G., 2011.Design and Implementation of an Efficient 3D GIS Database Engine. *Journal of Wuhan University (Information Science Edition)*, 36(02):127-132+139.