VECTOR AND RASTER DATA STORAGE BASED ON MORTON CODE

Guoqing Zhou, Qiuyu Pan, Tao Yue [*], Qingyang Wang, Hongjun Sha, Shengxin Huang, Xiaofan Liu

Guangxi Key Laboratory of Spatial Information and Geomatics, Guilin University of Technology,
No. 12 Jian’gan Road, Guilin, Guangxi 541004, China - (gzhou, yuetao)@glut.edu.cn

Commission III, ICWG III/II

KEY WORDS: Vector and raster, Integration, Morton code, Data storage, Georaster

ABSTRACT:

Even though geomatique is so developed nowadays, the integration of spatial data in vector and raster formats is still a very tricky problem in geographic information system environment. And there is still not a proper way to solve the problem. This article proposes a method to interpret vector data and raster data. In this paper, we saved the image data and building vector data of Guilin University of Technology to Oracle database. Then we use ADO interface to connect database to Visual C ++ and convert row and column numbers of raster data and X Y of vector data to Morton code in Visual C++ environment. This method stores vector and raster data to Oracle Database and uses Morton code instead of row and column and X Y to mark the position information of vector and raster data. Using Morton code to mark geographic information enables storage of data make full use of storage space, simultaneous analysis of vector and raster data more efficient and visualization of vector and raster more intuitive. This method is very helpful for some situations that need to analyse or display vector data and raster data at the same time.

1. INTRODUCTION

Vector and raster spatial data models are the based format for storing geographic data digitally. And both of them have their own advantages and disadvantages. Many of comprehensive geographic information systems (GIS) offering provide both integrated raster and vector analysis techniques (Turk A.G, 1992). Integrated raster and vector processing capabilities are most desirable and provide the greatest flexibility for data manipulation and analysis (Turk A.G., 1992). But there is still not a proper way to realize the integration of spatial data in vector and raster formats perfectly. Many researchers such as Wu Z. et al. (2017), Zhou G. et al. (2006) and Li C. et al. (2017) studied to unify the storage and management of vector data and raster data in spatial database, but they cannot realize the integrated vector and raster in analysis process and visualization. The other problem is that, with the maturing of UAV remote sensing true digital orthophoto map high precision mapping techniques (Zhou G. et al., 2008, 2009, 2016), the quality of TOM (raster data) is becoming higher and higher, and the amount of data is bigger and bigger. At the same time, there are more and more image information need to be stored in database.

Referring to the above problems, this paper proposes to store vector and raster data in Oracle Database 12C based on Morton code. For the purpose of storing a large number of high quality image data, the method takes full advantage of the powerful capabilities of Oracle database management in common data and Oracle GoeRaster in image (David B., 2009). GeoRaster is a feature of Oracle Spatial and Graph that lets us store, index, query, analyse, and deliver raster image and gridded data and its associated metadata (Xie Q. et al., 2012). Using Morton code instead of row number and column number to mark the position information of vector and raster data can make connecting vector and raster data more easily and more efficient. In the future, this method can also be applied to 3D city data storage and analysis, which is now being welcomed (Zhou G. et al, 2013, Guo J. et al., 2017).

2. DATA STORAGE

2.1 Data Table

As Figure 1b shows, the raster part of database has at least one Georaster table for storing raster image. All of information about the image, including an SDO_GEORASTER object, is contained in each record in the table. Many custom attributes of the image data such as ID, name can be added to the table. The metadata, the extent of the image and other spatial information will be described in detail in SDO_GEORASTER object.
Sample of create a Georaster table:

```
  CREATE TABLE GUT_images (
    image_id NUMBER,
    image_name VARCHAR2 (50),
    image SDO_GEORASTER);
```

Each image data should be divided into a number of small blocks called pixel. And each block is stored in a raster data table (RDT) using a BLOB object (Ravi K et al., 2007). It means that every raster data table corresponds to an SDO_GEORASTER object. In general, the position of the block is recorded using the row number and column number in RDT. This article, however, will use Morton code instead of row and column number to record the location of the block (the section 3 will describe Morton in detail). We can add some attributes such as elevation, grey scale to RDT if they are useful for spatial analysis or visualization.

```
  CREATE TABLE GUT_image_rdt OF SDO_RASTER (
    PRIMARY KEY (RASTERID, PYRAMIDLEVEL,
    BANDBLOCKNUMBER, ROWBLOCKNUMBER,
    COLUMNBLOCKNUMBER)
    )
```

[*] Corresponding author: Tao Yue; E-mail: yuetao@glut.edu.cn.

This contribution has been peer-reviewed.
https://doi.org/10.5194/isprs-archives-XLII-3-2523-2018

```
LOB (RASTERBLOCK) STORE AS (NOCACHE
   NOLOGGING);
```

The vector part is described in Figure 1a, there is at least one Geometry table records information of every polygon, line and point object. Every object is described in detail in one record and every record has a SDO_GEOMETRY object which includes the position of vertex, the order of vertex connection and other spatial information of polygon, line and point. Every Geometry table corresponds to a Point table. In this table, the information of every vertex such as ID, Elevation, Morton codes of polygon, line and point object are recorded individually. The method of create of the two table is resemble to raster table.

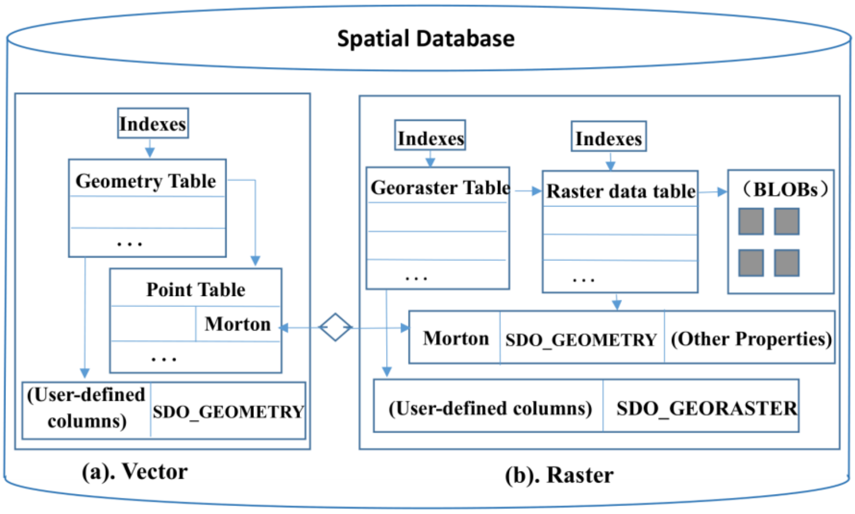


Figure 1. Data storage

2.2 Morton Code

In a 2n*2n image, *i* and *j* are used to represent the row and column number of image data. And *m*(*i*, *j*) is used to represent the Morton code of the *i*-th row and the *j*-th column (see Figure 2). The following conversion relationship can obtain Morton code from row and column number:

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

In this formula: $0 \leqslant i,\ j \leqslant 2^{\mathrm{n}}$, $\lfloor\ \rfloor$ presents down remainder arithmetic, and mod() represents rounding operation.

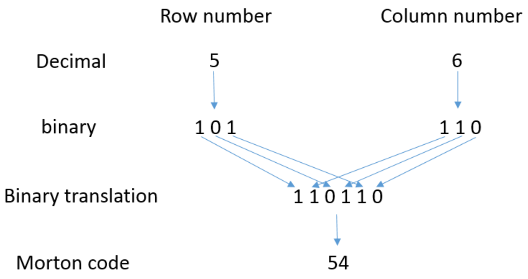


Figure 2. Conversion of Morton code

The Morton code conversion of vector data is little different form the raster, because the information of vector point: X Y are decimal.

$$m(\mathrm{X},\mathrm{Y}) = \sum_{k=0}^{n-1}(\lfloor |X|/2^{2k+1} \rfloor * \mathrm{mod}(2*2^{2k+1})) + \sum_{k=0}^{n-1}(\lfloor |Y|/2^{k} \rfloor * \mathrm{mod}(2*2^{2k})) \tag{2}$$

Figure 3 is a good explanation shows how vector data and raster data can be related by the Morton code. Assuming that the length of each block is 1 meter, then m_1=m(2, 2) represents a point of vector in image in Figure 3a. And its corresponding vector point in Figure 3b is M_1=M(2, 2). The Morton code of vector and raster data obtained by formula (1) are equal: m(2, 2)=M(2, 2)=12. When we want to do data processing, we can obtain vector information and raster information at the same time through Morton code. With two kinds of data information, it can make the data visualization more intuitive and the data processing more efficient.

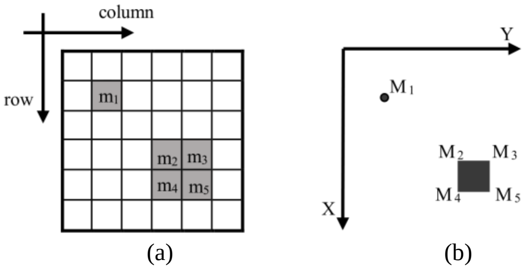


Figure 3. Morton code between vector and raster

Using Morton code to record the position information of vector and raster data has the following two important benefits:

(1) As we all know, raster image is two-dimensional, its position information needs row number and column number to mark together. The two-dimensional image data can translate into a one-dimensional data using the Morton code, as shown in Figure 2. It can reduce data redundancy to some extent when restoring raster data.

(2) What is most important, one-dimensional image retrieval is much faster than two-dimensional image, just as two-dimensional image is faster than three-dimensional image.

(3) After raster data and vector data are converted into one-dimensional at the same time, they can easily correspond to each other through their common attributes, that’s Morton code. In data retrieval, the location of the target can be quickly located with Morton code. It is just like the index which is useful.

3. APPLICATION

3.1 Data Import

There are many C++ interfaces that can access Oracle Database, such as ODBC and ADO that are simple to use (Li G., 2012). We use ADO interface, a comprehensive and easy-to-use API, to connect Oracle Database 12C to Visual C++. Then we import an image data of Guilin University of Technology to Raster table we created for test.

```
    INSERT INTO GUT_image (name, image) VALUES
      ('gut_image', SDO_GEOR.INIT('GUT_IMAGE_RDT'));
```

After importing image data to database, we storage the pixel information of the SDO_GEORASTER object to raster data table by using the LOB storage format SecureFile LOBs (SecureFiles). And then, we use a pointer to point to the data that needs to be saved and set some important properties. When choosing block size, we'd better trade off the size of a raster block and the number of blocks required for a Georaster object. In this paper, we divide the image into 512 rows and 512 columns. After successfully importing data to taster data table, we can check all information of image in raster data table (see Figure 6a) and use a comprehensive ETL tool, third parties provide visualization client, to display the image in Georaster table (see Figure 4). We can use some basic functions of ETL tool to browse images, including zooming in, zooming out and roaming. It's very convenient.

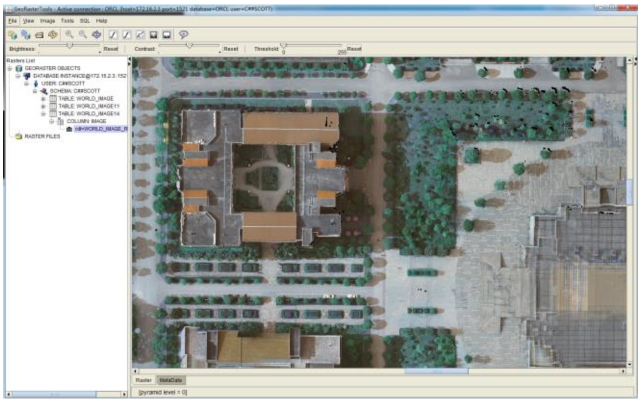
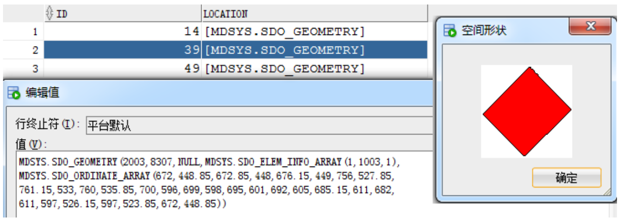


Figure 4. Storage of raster data

In order to import vector object to table, we first edit vector data to TXT in a certain format, it serves as control file. Spatial information such as spatial reference systems, geometry coordinates, and their connection order are stored in this TXT. And then, we use the SQL*LOADER tool to import vector objects. After saving vector data we can see the information in the table. In Figure 5, we see the shape of the building geometry.



	ID	LOCATION
1	14	[MDSYS.SDO_GEOMETRY]
2	39	[MDSYS.SDO_GEOMETRY]
3	49	[MDSYS.SDO_GEOMETRY]

编辑值

行终止符 (I)： 平台默认

值 (V)：

```
MDSYS.SDO_GEOMETRY(2003, 8307, NULL, MDSYS.SDO_ELEM_INFO_ARRAY(1, 1003, 1),
MDSYS.SDO_ORDINATE_ARRAY(672, 448.85, 672.85, 448, 676.15, 449, 756, 527.85,
761.15, 533, 760, 535.85, 700, 596, 699, 598, 695, 601, 692, 605, 685.15, 611, 682,
611, 597, 526.15, 597, 523.85, 672, 448.85))
```

Figure 5. Shape of building

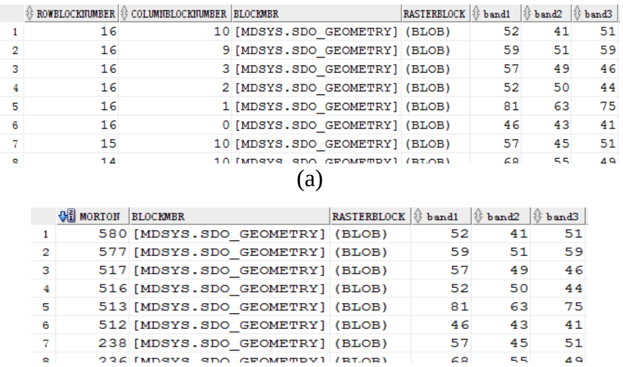
3.2 Convert to Morton Code

According to formula (1), we can convert row and column number to Morton code in Visual C++ environment: First of all, we create a function to realize of translation. Next, we convert one pair of row and column number to Morton code and update it to raster data table. Last but not least, we set up a loop statement to repeat the above steps to complete all data conversions.

```
int getMortonNum(row, column)
  {
    //Implementation of formula (1).
    return (Morton code);
  }
  for (int id = 1; id <Num ; id++) // Num represents the number
    of pixels needed to convert the Morton code.
  {
    //execute the query statement, get the row and column
    number of a pixel.
  int Morton= getMortonNum(row, column)
    //Execute the update statement and place the converted
    Morton code in the corresponding pixel record field.
   }
```

Just like Figure 6 shows, there are row number and column number record pixel of image respectively, but after converting to Morton code, there is only one column. When we select a pixel form raster data, then system will automatically determine whether the corresponding data exists in vector data. If the corresponding data is exists, system will automatically display or analysis it.

The conversion of Morton code of vector data is just like raster data and it should use formula (2). After storing all information of vector and raster data, we can view all the vector data and raster data information by selecting vector point data or directly through the Morton code, as shown in Figure 6.



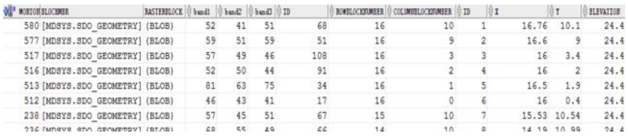
	ROWBLOCKNUMBER	COLUMNBLOCKNUMBER	BLOCKMBR	RASTERBLOCK	band1	band2	band3
1	16	10	[MDSYS.SDO_GEOMETRY]	(BLOB)	52	41	51
2	16	9	[MDSYS.SDO_GEOMETRY]	(BLOB)	59	51	59
3	16	3	[MDSYS.SDO_GEOMETRY]	(BLOB)	57	49	46
4	16	2	[MDSYS.SDO_GEOMETRY]	(BLOB)	52	50	44
5	16	1	[MDSYS.SDO_GEOMETRY]	(BLOB)	81	63	75
6	16	0	[MDSYS.SDO_GEOMETRY]	(BLOB)	46	43	41
7	15	10	[MDSYS.SDO_GEOMETRY]	(BLOB)	57	45	51
8	14	10	[MDSYS.SDO_GEOMETRY]	(BLOB)	68	55	49

(a)

	MORTON	BLOCKMBR	RASTERBLOCK	band1	band2	band3
1	580	[MDSYS.SDO_GEOMETRY]	(BLOB)	52	41	51
2	577	[MDSYS.SDO_GEOMETRY]	(BLOB)	59	51	59
3	517	[MDSYS.SDO_GEOMETRY]	(BLOB)	57	49	46
4	516	[MDSYS.SDO_GEOMETRY]	(BLOB)	52	50	44
5	513	[MDSYS.SDO_GEOMETRY]	(BLOB)	81	63	75
6	512	[MDSYS.SDO_GEOMETRY]	(BLOB)	46	43	41
7	238	[MDSYS.SDO_GEOMETRY]	(BLOB)	57	45	51
8	236	[MDSYS.SDO_GEOMETRY]	(BLOB)	68	55	49

(b)

Figure 6. Contrast before and after Morton code conversion



MORTON	BLOCKMBR	RASTERBLOCK	band1	band2	band3	ID	ROWBLOCKNUMBER	COLUMNBLOCKNUMBER	ID	X	Y	ELEVATION
580	[MDSYS.SDO_GEOMETRY]	(BLOB)	52	41	51	68	16	10	1	16.76	10.1	24.4
577	[MDSYS.SDO_GEOMETRY]	(BLOB)	59	51	59	51	16	9	2	16.6	9	24.4
517	[MDSYS.SDO_GEOMETRY]	(BLOB)	57	49	46	108	16	3	3	16	3.4	24.4
516	[MDSYS.SDO_GEOMETRY]	(BLOB)	52	50	44	91	16	2	4	16	2	24.4
513	[MDSYS.SDO_GEOMETRY]	(BLOB)	81	63	75	34	16	1	5	16.5	1.9	24.4
512	[MDSYS.SDO_GEOMETRY]	(BLOB)	46	43	41	17	16	0	6	16	0.4	24.4
238	[MDSYS.SDO_GEOMETRY]	(BLOB)	57	45	51	67	15	10	7	15.53	10.54	24.4
236	[MDSYS.SDO_GEOMETRY]	(BLOB)	68	55	49	66	14	10	8	14.19	10.99	24.4

Figure 7. Select information from database

4. CONCLUSION

This article proposed a method to store vector and raster data in Oracle database and convert row and column number and X Y

This contribution has been peer-reviewed.

of position information to Morton code. This method makes it easy to relate vector data to raster data. When retrieving or viewing the data, we can easily get all the information including vector data and raster data. These advantages can enable some spatial analysis and visualization to save time and calculation to a certain extent. Using Morton code instead of row and column and X Y to mark the positional information of vector and raster data can make retrieval of data faster.

ACKNOWLEDGEMENTS

This paper is supported by the National Natural Science of China under Grant numbers 41431179, the National Key Research and Development Program of China under Grant numbers 2016YFB0502500, the State Oceanic Administration under Grant numbers 2014#58, GuangXi Natural Science Foundation under grant numbers 2015GXNSFDA139032, Guangxi Science & Technology Development Program under the Contract numbers GuiKeHe 14123001-4, GuangXi Key Laboratory of Spatial Information and Geomatics Program under Grant numbers 151400701, 151400712, 163802512, Guangxi Doctoral student innovation under grant numbers YCBZ2018054.

REFERENCES

David B., 2009. The GIS Primer-An Introduction to Geographic Information Systems. pp. 22-27.

Guo J., Sun B., Qin Z., Wong S., Man S. and 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 Internationa*l, 65, 13-31.

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

Li G., 2012. Development of gis application system on occi and mapx. *Etcs Proceedings of Ird International Workshop on Education Technology & Computer S, 2*(7), pp. 235-238.

Ravi K., Albert G. and Euro B., 2007. Pro Oracle Spatial for Oracle Database 11g. Amarica, *Apres*s, pp. 1-9.

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

Wu Z., Li C., Wu P., Sun J., and 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. and 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),* pp. 91-96.

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

Zhou G. and 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. and 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), pp. 739-747.

Zhou G., Tan Z., Cen M. and Li C., 2006. Customizing visualization in three-dimensional urban gis via web-based interaction. *Journal of Urban Planning & Development*, 132(2), 97-103.