Visualize Oracle GeoRaster Objects in Geoglobe System under Internet Environment*

Longgang Xiang [a], Qingyun Xie[b], Charles Wang[c]

[a] LIESMARS, Wuhan University, Wuhan 430079, P.R. China - lgxiang@lmars.whu.edu.cn
[b] Oracle USA, Inc., One Oracle Drive, Nashua, NH 03062, USA - qingyun.xie@oracle.com
[c] Oracle China, Inc., Asia-Pacific Research Center, Beijing 100193, P.R. China - charles.wang@oracle.com

KEY WORDS: Raster Data, GeoGlobe, *GeoRaster*, Google Earth, Visualization, Pyramid

ABSTRACT:

Raster format data, mainly images and digital elevation models, is a very important spatial data source, as can be seen clearly from Google Earth. In this paper, we discuss the problem of efficient storage and visualization for huge raster data. The developed solution is a seamless integration of GeoGlobe system and Oracle GeoRaster module, where the latter stores raster data and the former visualizes them through network. We design extensible integration architecture and efficient database structure. We also discuss some key implementation issues, including invalid edge elimination, fast visual building and identification conversion. Our demo system and comparative experiments show that the integration combines the strength of GeoGlobe system and Oracle GeoRaster module, and therefore is a valuable network application of raster data, especially under internet environment.

1. INTRODUCTION

Due to the rapid advances in remote sensing technologies, the acquisition of raster data has become quite popular. Not only domain experts, but also general public, are interested and involved in a variety of applications related to raster data. To meet this requirement, Google released the software of Google Earth in 2005. It is now widely used, for its high resolution satellite images, fast network viewing and easy operating. Following this trend, many similar systems are developed, such as World Wind [1], Virtual Earth [2], ArcGlobe [3], and so on. They together bring geographic applications into a new era, which visualizes and shares spatial data in three dimensional (3d for abbreviation) way through network.

GeoGlobe, developed by LIESMARS (the state key Laboratory of Information Engineering in Surveying, Mapping And Remote Sensing, LIESMARS), is a Google Earth like system, but focus more on spatial information integration and sharing. With GeoGlobe, users can view spatial data (including image, DEM, vector, 3d builder, placemark, etc.) in 3d spherical scene. Besides, GeoGlobe can integrate GIS platforms, access OGC web services (including WMS, WCS, WFS, etc.), and connect to other location-related data sources (such as GPS, KML, and so on).

In order to store spatial data in raster format, Oracle Spatial provides SDO_GEORASTER data type (introduced since the version of 10g) [4,5]. Conceptually, an SDO_GEORASTER object is an N-dimensional matrix of cells, where the dimensions include row, column, band and other optional dimensions. With the Oracle GeoRaster component, user can perform efficient storage, query and manipulation of a variety raster data, including aerial photos, satellite images, DEM, etc.

In his paper, we introduce the integration of GeoGlobe system and Oracle GeoRaster module, using the latter to store raster data and present them with the former under internet environment. There are three main components in the integration system, i.e., visual importer, access interface and network service. To validate the effectiveness, we developed a demo system and conducted some comparative experiments, and the results show that Oracle GeoRaster works very well as the underlayer storage of GeoGlobe raster data.

The rest of the paper is organized as follows. Section 2 introduces two closely related works, i.e., GeoGlobe System and Oracle GeoRaster module. In section 3 and 4, the integration architecture and related key issues are discussed respectively. The demo system and comparative experiments are presented in section 5. Finally, we conclude this paper in section 6.

2. RELATED WORKS

There are many works related to the work of this paper, and among these, GeoGlobe system and Oracle GeoRaster are directly related.

2.1 GeoGlobe

GeoGlobe is a network application of spatial information, where raster format data is the main content to be scheduled and rendered. Different from traditional applications, GeoGlobe visualizes raster data in 3d manner from a global perspective. In order to provide fast interactive response, raster data in GeoGlobe is organized as multi-resolution global pyramid, so a GeoGlobe pyramid is not only multi-resolution, but also globally blocked at all levels. Figure 1 show a 3-level image pyramid, where level 0 has 1 tile, level 1 has 4 tiles and level 2 has 16 tiles. All these tiles have the same size in pixel. Currently, GeoGlobe uses file system to store global pyramids of raster data.

Based on the above organization for raster data, GeoGlobe client take pyramid as the logical unit for schedule and rendering. Once a pyramid falls into the current view range, its tiles are checked in a top-down manner, and only those

* This work is supported by the National Key Technology R&D Program of P.R.China (No. 2006BAB10B03-B) and the “863” Program of P.R.China (No. 2007AA12Z201)

intersecting with the current view range are requested from remote GeoGlobe server.

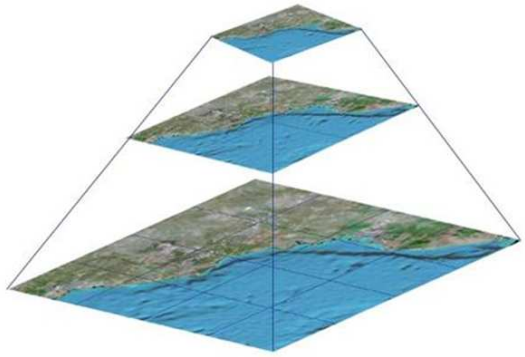


Figure 1. An example image pyramid

2.2 Oracle GeoRaster

GeoRaster is a feature of Oracle Spatial that lets user store, index, query, analyze, and deliver raster image and gridded data and its associated metadata. It uses a generic raster data model that is component-based, logically layered, and multidimensional. The core data in a raster is a multidimensional matrix of raster cells. Each cell is one element of the matrix, and its value is called the cell value.

The multidimensional matrix of cells is blocked into small subsets for large-scale GeoRaster object storage and optimal retrieval and processing. Each block is stored in a table as a binary large object (BLOB), and a geometry object (of type SDO_GEOMETRY) is used to define the precise extent of the block. Each row of the table stores only one block and the blocking information related to that block. (This blocking scheme applies to any pyramids also.)

3. INTEGRATION ARCHITECTURE

The integration system consists of three connective parts: visual importer, access interface and network service, as shown in Figure 2. Visual importer loads raster data files into Oracle database as GeoRaster objects. It accepts two kinds of input: file pyramid and raster data file, where the former comes from the existed file-based pyramids, and the latter includes various format files, such as geotiff files, img files, and so on. Access interface provides interface to retrieve GeoRaster blocks according to level, row and column. In the integration system, network requests are processed by the module of network service. Network service uses the "GET" method of the HTTP protocol to transmit data and the query string has the format of "T=dataset name&L=level number&X=column number&Y=row number".

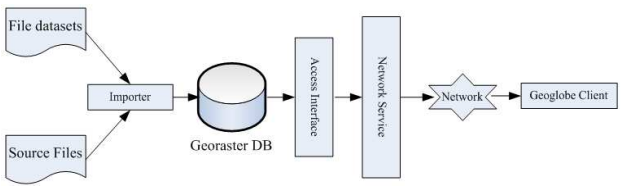


Figure 2. The integration system architecture

The database schema is shown in figure 3. It has two kinds of table: one is a normal table and the other is an object table, where the normal table records the meta information of a pyramid, while the object table stores tile data of the pyramid. In ther integration system, there exists only one normal table in the integration system, and each row of it represents a GeoGlobe pyramid. The normal table contains a column with the data type of "SDO_GEORASTER".

Obviously, a pyramid may be built from more than one raster data files, and it is stored in the normal table as a GeoRaster object. For fast response, visual importer will create a raster data table for each pyramid, so there may exist many raster data tables in the GeoRaster database. From figure 3, one can see it clearly that the tile data of a pyramid are actually stored in the corresponding raster data table.

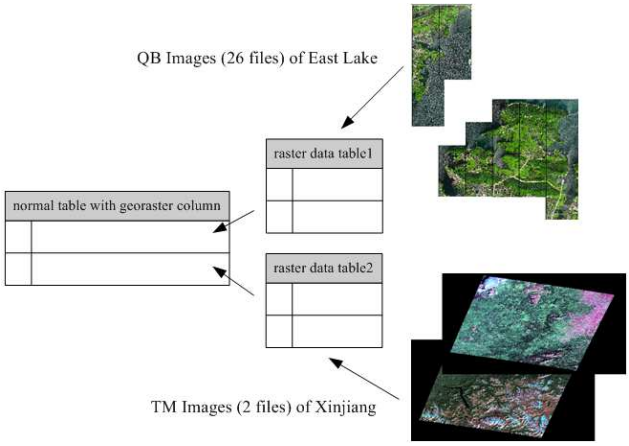


Figure 3. GeoRaster database schema

4. KEY ISSUES

This section discusses three issues related to GeoGlobe application, including invalid edge elimination, fast visualization, and identification conversion.

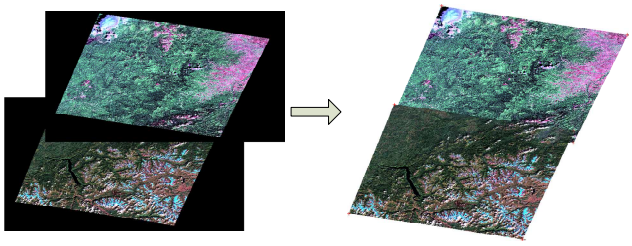
4.1 Elimination of invalid edge

Satellite images are usually not regular rectangle. When they are processed in computer world, invalid edge (often appears as black color) occurs inevitably, as can be seen in figure 3. The invalid edge must be filtered out in GeoGlobe application, and the reasons are as follows: 1) better visualization effect; 2) correct visualization for multiple overlapping images. In order to filter invalid edge, the integration system uses two different methods: invalid color and valid polygon. Figure 4 shows the result of invalid edge elimination for two example images, where (a) uses invalid color, while (b) uses valid polygon. It should be noted that only the valid polygon method works correctly in the case of figure 4 (b), because the valid parts also have cells with the color of black.

When stored in file system, the tiles containing invalid part are saved as PNG file and the others are saved as JPEG file. However it is some difficult and complex to handle this problem with GeoRaster, because in a Georaster object, only one format (raw or jpeg) is supported. To overcome it, we artificially create an alpha channel for each Georaster object, so there are 4 channels for RGB images and 2 channels for gray images. The alpha channel is used to indicate whether a cell is

filtered or not. It should be noted out that this design only allows raw-format for the Georaster object, because compression will change the content of the alpha channel, may resulting in error elimination of invalid edge.

(a) Invalid color filtering



(b) Valid polygon filtering

Figure 4. Elimination of invalid edge

4.2 Fast visual building

Visual importer requires fast visualization of raster data files; otherwise it is hard to be practical. Building pyramid in advance can improve the performance of roam and zoom, but it is a time-consuming job, especially for huge multiple raster data files. To resolve this problem, we use a three-level caching mechanism while browsing. In figure 5, there are 179 image files (totally 142G). The time to load is about 20 seconds, while the time for a zoom or a roam mostly finishes in 1-2 seconds.

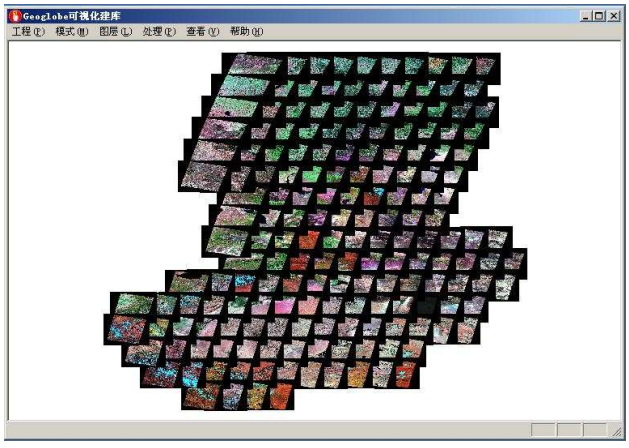


Figure 5. Visualize XinJinag TM Images

The three-level caching mechanism includes screen picture caching, cell value caching and GDAL block caching, where screen picture caching only works in roam operations, while other two caching works not only in roam operation, but also in zoom operations.

(1) GDAL block caching. Cache GDAL blocks into memory and when reading a cell, if its block is cached, query it from GDAL block cache directly and therefore, disk read is avoided.

(2) Cell value caching. For each pixel in current screen, the corresponding cell is cached into memory. When redrawing the screen, if the corresponding cell of a pixel is cached, its value can be retrieved directly.

(3) Screen picture caching. Save the current screen as picture and cache it into memory. When user drags mouse, the cached picture can be reused by clipping those out of new screen.

4.3 Identification conversion

To seamlessly cover the whole earth surface, Geoglobe apply the method of equal latitude & longitude to partition the input images into tiles, and the origin of partition is chosen at the point of (-180.0o, -90.0o). However in Georaster, the partition origin is the left upper corner of an image, so when a network tile request poses, we must compute the corresponding GeoRaster block id, and then use it to locate and fetch tile data. The conversion method is given in algorithm 1.

Algorirhm 1: convert from GeoGlobe tile id to GeoRaster block id
Input:
1, tile level number *tileLevNo*,
2, tile row number *tileRowNo*,
3, tile Column number *tileColNo*.
Output:
1, block level number *blkLevNo*,
2, block row number *blkRowNo*,
3, block Column number *blkColNo*.
Method:
01, *blkLevNo* = *levelNums*-*tileLevNo*-1;//levelNums is the total number of pyramid level
03, compute the tile end row *tileEndtRowNo* number at level *tileLevNo*;
04, compute the tile start column number *tileStartColNo* at level *tileLevNo*;
05, *blkRowNo* = *tileEndRowNo*-*tileRowNo*;
06, *blkRowNo* = *tileColNo*-*tileStartColNo*;
07, return;

5. DEMO AND EXPERIMENTS

5.1 Demo System

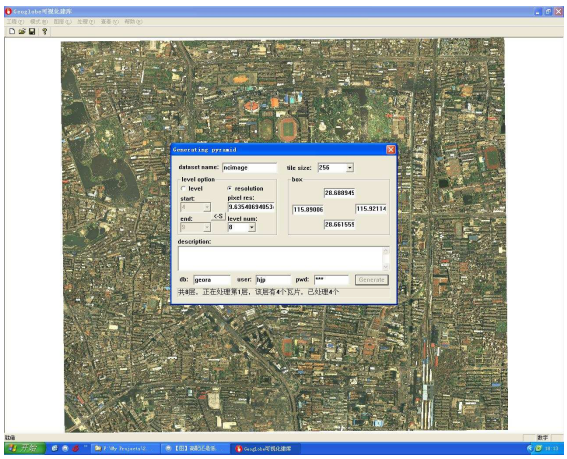
The demo system consists of four parts: image importer, tile server, catalog center and presentation client. Image importer, shown in figure 6(a), is used to load image files into GeoRaster database. Tile server, shown in figure 6(b), is used to provide tile data through network. Catalog center, shown in figure 6(c), is used to register tile server nodes. Presentation client, shown in figure 6(d), is used to visualize tile data in spherical 3d scene. This demo system works as follows:

1) Run image importer, loading image files into oracle as GeoRaster objects;

2) Run tile server, loading Georaster objects as raster pyramids;

3) Run catalog center, registering tile server nodes (including server address and raster pyramid parameters);

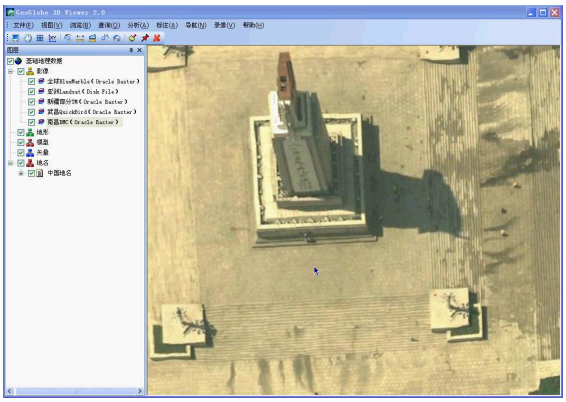
4) Run presentation client, dynamically visualizing tile data from tile server.



(a) Image importer

(b) Tile server based on GeRaster objects

(c) Geoglobe catalog center



(d) GeoGlobe 3d presentation client

Figure 6. Demo of the Integration system

5.2 Experimental results

In order to validate the importing efficiency of Oracle Georaster, we conducted three sets of experiments, comparing the importing performance with that of file system. For all experiments, we have the following configurations: 1) oracle software and the tested database were installed in local machine, which is a normal PC of Dell (Core 2 Duo CPU, 2GB DDR2 Memory and 250GB 7200 rpm Disk); 2) the table space used one physical data file (not bigfile mode), initially 100MB and increasing on next 100MB, and the storage of column "RASTERBLOCK" in raster data table is specified as “SECUREFILE”.

The first set of experiments used Wuhan quickbird images with the format of BMP+DOM, and its pixel resolution is 0.6m. Table 1 shows the comparative result, and in the table: the first column records data volume and file numbers and its volume, and the second and the third column records runtime of oracle Georaster and file system respectively. The runtime include both block partition and database writing (the same below).

The second set of experiments used Nanchang DMC images with the format of BMP+DOM, and its pixel resolution is 0.1m; the last set of experiments used Xinjiang TM images with the format of TIF+TFW, and its pixel resolution is 15m. Table 2 and Table 3 show the results respectively.

File Number / Total Volume	Oracle GeoRaster	Disk File
1/11.4M	4s	4s
2/22.9M	6s	7s
4/45.9M	10s	16s
8/91.8M	23s	41s
16/183M	90s	120s
32/298M	145s	190s

Table 1. Runtime of Wuhan quickbird images

File Number / Total Volume	Oracle GeoRaster	Disk File
1/294M	2m30s	2m42s
2/589M	5m10s	5m38s
4/1.15G	12m30s	13m30s

9/2.58G	25m20s	25m45s

Table 2. Runtime of Nanchang DMC images

File Number / Total Volume	Oracle GeoRaster	Disk File
1/895M	3m30s	5m50s
2/1.79G	12m50s	13m50s

Table 3. Runtime of Xinjiang TM images

From these results, one can see that oracle GeoRaster achieves faster importing than file system in all cases. So it is unnecessary to worry about the writing speed of Oracle GeoRaster, when compared to that of file system. Besides, Oracle GeoRaster is more extensible and flexible than file system, so for GeoGlobe system, Oracle GeoRaster is a better choice than file system to store and mange raster data.

6. CONCLUSION

This paper introduces the integration problem of GeoGlobe system and Oracle GeoRaster module. In the integration system, raster format data is stored in Oracle as GeoRaster objects, and they are visualized by GeoGlobe. Our demo system and comparative experiments show that Oracle GeoRaster achieves faster importing than file system, and imported GeoRaster objects can be efficiently visualized by GeoGlobe 3d client through network (including both internet and LAN). So the integration combines the strength of the two technologies, and is a valuable network application of raster data, especially under internet environment.

References from Books:

[1] *http://worldwind.arc.nasa.gov/*.

[2] *http://www.microsoft.com/virtualearth/*.

[3] *http://www.esri.com/news/arcnews/summer03articles/introducing-arcglobe.html*.

[4] Ravi Kothuri, Albert Godfrind, and Euro Beinat, *Pro Oracle Spatial for Oracle Database 11g*, 2008.

[5] http://www.oracle.com/technology/products/spatial/index.html