Serving cartography raster data in the Internet, a performance study

Eloína Coll, José-Carlos Martínez, Jorge-Gaspar Sanz
Department of Cartographic Engineering, Geodesy and Photogrammetry
Technical University of Valencia

INTRODUCTION

Map servers are applications oriented to show geographic information in a web site. These pieces of software cooperate with other applications as web servers, geographic enabled databases, etc. On the other hand, the evolution of the massive storage of images and compression algorithms allow acceding to huge amounts of geographic raster information faster and more efficient than few years ago.

In this work we will present a study of the performance of different geographic imagery data formats. First a web application is developed to accede repeatedly to a image and to store the amount of time elapsed, storing this time in a database. Executing this application in equal conditions over different formats and varying the scale of the image requested, we can establish relative differences between these data formats and try to find the most convenient to serve this kind of images through the web.

OBJECTIVE

Comparing performance offered to accessing different data raster images data formats for their spreading with a mapserver is the main objective of this work. Initial image will be wider enough (1.7 thousands of millions of pixels) and it will be translated to different widely used raster formats on storing aerial images or orthoimages.

Secondly, a experiment will be designed to reproduce the conditions of a image service through Internet. Using a form, a starting point will be defined, a window image side size defined in real meters, a window image side size generated by the mapserver in pixels, number of rows and columns of the grid that the application will runs over the initial image. On every access to the image, the time used in this task will be stored in a shapefile that will recreate the grid generated.

This way an empiric and enough repeated experiment is created to watch the performance of a mapserver serving diferent data formats. Furthermore, this application could be useful to test other services as vectorial data or connection to other mapservers using standardized protocols as Web Features Service or Web Map Service.

DATA FORMATS

Data formats used in this work are briefly described in the next lines. Two of them (ECW and TIFF), are oriented to storing images thinking in visualizations (in a remote or local way). On the other side, the other two, --IMG and BIN-- are oriented to the analysis multiespectral images as is done in remote sensing and so on.

In general, both first formats store with certain loss the information with the objective to reduce their disc storage drastically. This it is an important aspect since in the last years the power of processing of the microprocessors has increased spectacularly, whereas the speed of access to the secondary memory (hard disks, tapes, etc) has advanced much less. Therefore, the necessity to accede to a great amount of information in disc has slowed down enormously the treatment of images of great sizes.

The arising of new formats, that are allowed to compress information with high ratios --but they need an important processing work to their (de)compress-- has made possible to generate images with sizes actually huge, that on the other hand, are possible to access with very short times.

These formats will be studied:

- **Erdas Imagine**[38]:This format allows to store multispectral images, admits pyramids and it's easily accessed from other applications as well as Erdas. If the image is over 2GB, this is stored in two different files: one file with extension .img stores the header and pyramids, and in the other file (.ige) the image information. This format doesn't loose information, thus the image generated takes up 5.4GB of the hard disk.
- **ENVI**[39]**:** Like the previous one, this format permits to store multispectral images, with a somewhat smaller size to the previous one (4.8GB), without lost. It is a format used traditionally for the accomplishment of analysis of multispectral images in the field of the remote sensing. In any case, this type of images can be offered in Internet by means of a mapserver and it's interesting the study of its performance.
- **GeoTIFF**[40]: It is a variation of one of the most used formats to store images, allows to geocode the image without using any header file (tfw). It stores in one unique file the image and the pyramids and permits some different typos of compression. This format is limited on the file size to 2GB, reason why the original image has been tiled in four images mosaic. In this case any type of compression has been used, and the resulting tiled images occupies 6.4GB.
- **ER Mapper**[41]: This formats uses a wavelet algorithm to reduce the image with ratios over 20:1. This format is been used to store huge images if geographic information. This format doesn't needs pyramids because the algorithm builds the image at the requested resolution on every request.

GEOGRAPHIC INFORMATION

The initial image of this study is a orthoimage of the city of Valencia and suburbs (see fig. 1). Originally it’s an ECW image with a resolution of 30 centimeters, 45899 columns and 37169 rows (1.7 millions of pixels). The space used by this image is 232MB. The coordinate system is UTM, zone 30 on the Hayford's International Ellipsoid. Next three datasets are generated from this initial image.

With ERDAS software, a image is generated with its pyramids (from 4X to 100X), it has three bands tiled with 128 pixels cells. To create the GeoTIFF image, the GDAL library (Warmerdam, 2005) and its utility *gdal_translate* is used. As the uncompressed image is up to 2GB, the GeoTIFF image has been tiled in four cells. To present this tiled image as one only image, *gdaltileindex* has been used. This tool writes a shapefile with as many features as tiled images are and stores in the attribute table the location of every image to present them together. Finally, the ENVI image is also created with *gdal_translate*, stored in three bands too.

[38] http://gis.leica-geosystems.com/products/

[39] http://www.rsinc.com/envi/

[40] http://www.remotesensing.org/geotiff/geotiff.html

[41] http://www.ermapper.com

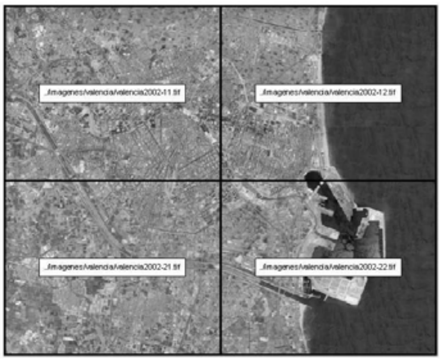

../imagenes/valencia/valencia2002-11.tif
../imagenes/valencia/valencia2002-12.tif
../imagenes/valencia/valencia2002-21.tif
../imagenes/valencia/valencia2002-22.tif

Figure 1: Tiles of the GeoTIFF image

THE MAP SERVER

The University of Minnesota Map Server (Lime and Morissete, 2005) is a long term project, started in 1994 as a very simple Arc/Info script to generate images for the web. In 1995 the ForNet[42] project starts to manage natural resources in Minnesota (DNR). This project, sponsored by NASA, supports the Remote Sensing laboratory of the UMN to develop a cartography server through the Internet. First version (2.0) is published in 1998 as ForNet Mapserver. During last years this project is being supported by different instituttions, growing in resources. Nowadays 20 developers work in the project as well as thousands of testers and users all over the world.

This maps server is free software. This implies that everyone can use it with no restrictions, being allowed to modify it, to adapt to their necessities and so on. This project, as usual in the free software world, uses other related projects. One of the most interesting is the Graphics Data Abstraction Library (GDAL) commented previously. That is to say, all the functionalities and formats supported by GDAL can be used in Mapserver. This allows this map server accessing many raster data formats to serve data over the Internet to any user.

Another interesting feature of Mapserver is the possibility of using it not only as a basic server, but it provides a set of tools for developers with many different programming languages. Therefore, it's possible to accede to the objects model called MapScript through these languages. The most interesting language is PHP because it's specially oriented to serve dynamic web pages.

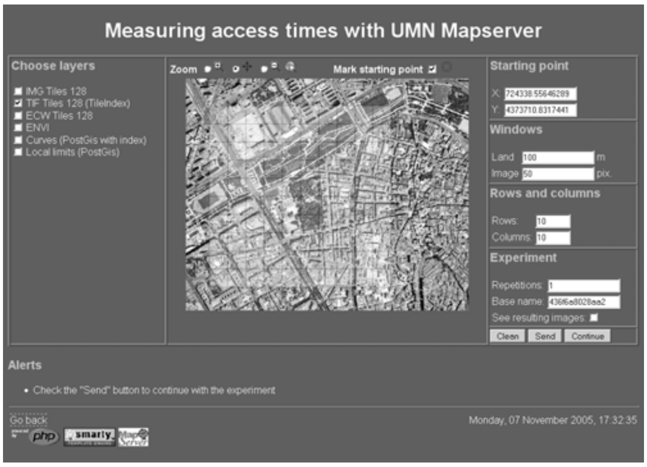
In this way, an application is developed to simulate a images service in different data formats, but focusing in the iterative realization of requests controlling the total time used to read the image. Next, the application is presented.

THE EXPERIMENT

Web application

As it's commented, a web application is developed using the programming language PHP to accede to the objects model MapScript. Also, the Smarty template engine is used to separate the graphic interface from the logic of the application.

[42] http://www.gis.umn.edu/fornet



Measuring access times with UMN Mapserver

Choose layers

- □ IMG Tiles 128
- ☑ TIF Tiles 128 (TileIndex)
- □ ECW Tiles 128
- □ ENVI
- □ Curves (PostGis with index)
- □ Local limits (PostGis)

Mark starting point ☑

Starting point

X: 724338.55646289

Y: 4373710.8317441

Windows

Land: 100 m

Image: 50 pix.

Rows and columns

Rows: 10

Columns: 10

Experiment

Repetitions: 1

Base name: 436f6a8028aa2

See resulting images: □

Alerts

- Check the "Send" button to continue with the experiment

Monday, 07 November 2005, 17:32:35

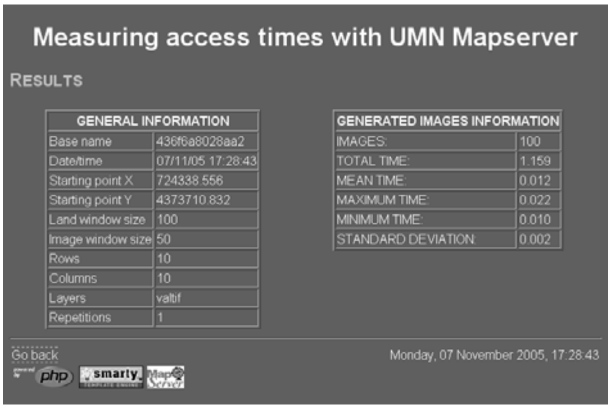
Figure 2: Establishing parameters of the measuring

Some web pages are designed, first a simple cartographic viewer to check the correct visualization of the different images. Next, a page were parameters of the experiment are established (fig. 2) and finally a page were the images are requested and results are presented (fig. 3).

The parameters page shows the resulting grid because on every parameters modification a shapefile is created and presented. Also, shows messages indicating steps needed to complete the experiment. Once defined, these parameters are passed to the next page as a simple string in the URL (GET).

In the last page the parameters are verified, the resulting grid is created and the image is explored storing in the shapefile all the data (row, column and seconds). The experiment parameters as well as the summarizing results (mean, maximum, and so on) are stored in a separate database.

This way, we have individual experiment results of every cell explored (as alphanumeric data associated to a shapefile) as well as statistic information of every experiment.



Measuring access times with UMN Mapserver

RESULTS

GENERAL INFORMATION	
Base name	436f6a8028aa2
Date/time	07/11/05 17:28:43
Starting point X	724338.556
Starting point Y	4373710.832
Land window size	100
Image window size	50
Rows	10
Columns	10
Layers	valtif
Repetitions	1

GENERATED IMAGES INFORMATION	
IMAGES:	100
TOTAL TIME:	1.159
MEAN TIME:	0.012
MAXIMUM TIME:	0.022
MINIMUM TIME:	0.010
STANDARD DEVIATION:	0.002

Monday, 07 November 2005, 17:28:43

Figure 3: Results of a experiment

Hardware and software used

The experiment has been done in a workstation with the following characteristics:

- *Microprocessor*: AMD Athlon XP 2600 1.96Ghz
- *Main memory (RAM)*: 512MB
- *Secondary memory (Hard disks)*: 80GB + 120GB
- *Operating system*: Windows XP Professional (SP2)
- *Web server*: Apache 1.3.33 + PHP 4.3.11
- *Map server*: UMN Mapserver 4.6 + MapScript 1.235

Even though it's not a system with the features of a real web server, it provides a relative results that can be extended to any other architecture.

Method of measuring times

After switch on the workstation, it's left without starting any user account, to avoid any unnecessarily process charge. On another computer, using a web browser, one can accede to the project pages and go on with the measurements. All the results are stored in the workstation, using the other one as a simple light weight terminal that shows the user interface.

This way the principal computers only provides the web service, working with the images, generating databases, etc. The other user tasks are executed in other computer allowing to repeat the experiment practically always in the same conditions.

The first image (fig. 1) has been measured with grids of different sizes depending the number of windows that can be explored. This way, the maximum number of cells that has been possible to measure has been taken to allow its representativeness.

Overall, every experiment has been done three times except the two last sizes where five times have been needed to get enough measures. The generated image side has been always 256 pixels, even though this size is below the usually used in web mapping, it's enough to represent a real server conditions (Hendry, 2004).

After measuring all the sizes and formats, a new huge database with all data is created from the shapefiles and the experiments resume database. This database has the next schema:

- **Form**: Image format (ECW,ENVI,TIF,IMG)
- **TamImg**: Size of the image cell in pixels (128-16372)
- **TamVent**: Size of the generated web image (128)
- **Pas**: Number of the pass (1-5)
- **Col**: Column of the grid
- **Fil**: Row of the grid
- **Secs**: Time in seconds used to read the original image

ANALYSIS OF STUDIED FORMATS

After adding all the measures to the in a one only table, a multidimensional information structure is created. The fact of the star schema is the time used to read the image and the dimensions are the rest of attributes. These dimensions will be used to aggregate the facts with different criteria.

We have used to achieve the data exploration analysis mainly the statistics software R (R Development Core Team, 2005). Also, a client to execute some SQL statements has been useful to do some aggregations by the database server.

General plots

17760 measures have been done. Histograms by format can be presented differentiating times below 95th percentile and above it (figs. 4 and 5 respectively). This percentile is considered to allow a convenient representation. Box and whisker plots are also presented separating both pairs of formats (ECW-TIFF and IMG-ENVI). This plot presents how ENVI format generates more scattered and with higher values than others.

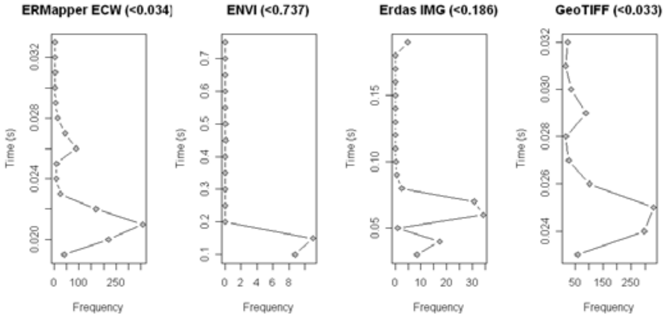


Figure 4: Time of different formats below 95% quantile

Moreover, analyzing differences between different image formats, table 1 reflects to every format the mean times for every size of image. Figures 6 and 7 present mean times and standard deviation for every format. In this figures, the horizontal axis present the size of the window side expressed in pixels in a logarithmic scale to facilitate reading.

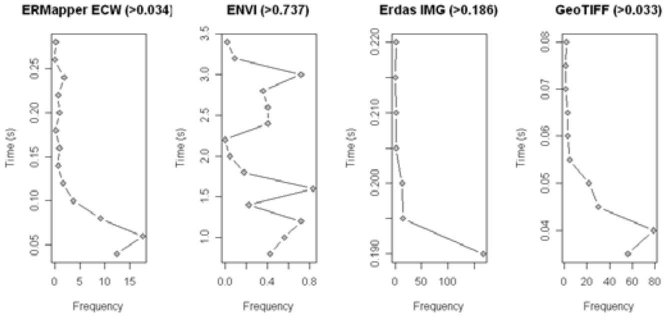


Figure 5: Time of different formats over 95% quantile

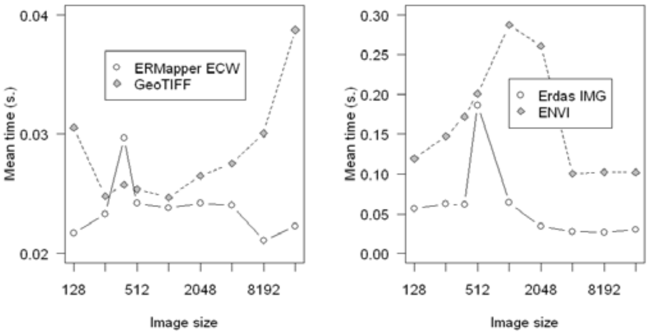


Figure 6: Mean time calculated for every different image size

It's necessary to separate formats into two figures to present mean times and standard deviations with values completely different. As we saw in figure 8, deviations by size and format are also pretty higher at ENVI binary format (fig.7).

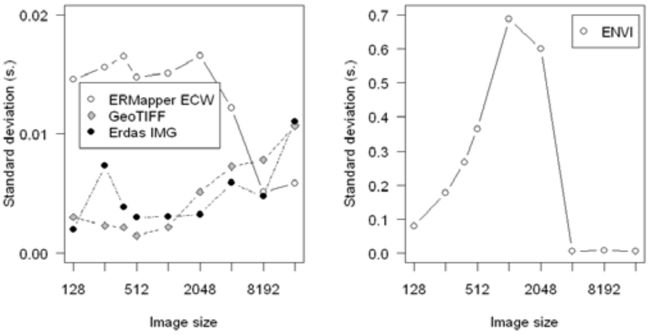


Figure 7: Standard deviation calculated for every different image size

	ECW	ENVI	IMG	TIFF
128	0.022	0.120	0.057	0.031
256	0.023	0.148	0.062	0.025
384	0.030	0.172	0.062	0.026
512	0.024	0.201	0.190	0.025
1024	0.024	0.287	0.064	0.025
2048	0.024	0.261	0.034	0.027
4096	0.024	0.100	0.027	0.028
8192	0.021	0.102	0.026	0.030
16384	0.022	0.102	0.030	0.039

Table 1: Average times by image side size (in pixels) and format

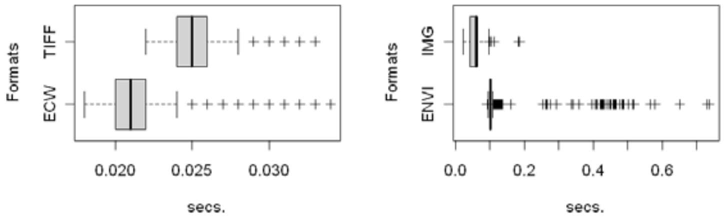


Figure 8: Time of different formats below 95% quantile in a boxplot

Anomalous results

IMG format

One of the most outstanding characteristics of figure 6 is the peak produced by the IMG format on the fourth image size corresponding with 512 pixels in the original image. Mean times are in general higher than the rest of formats studied. This anomalous behaviour has been repeated in later measures, to rule out any system incidence.

ECW format

ECW format works quite well, no increasing time is detected when the image size is higher. In any case, it's true that in some cells the time used to generate the image is quite higher. The regular arrangement of these anomalous measures, probably it's due to the necessity of taking to extract one image some tiles from the original image. As an example, figure with times for the experiment with 1024 pixels (every cell is the mean of the three passes.

TIFF format

This format offers times slightly higher than ECW, but the increase of these times since 1024, without been excessive (0.04 secs), could be significative if the offered image is huge. One feature of this format is that in fact the final image is a mosaic of images due to it limitation in file size. If one observes accessing times in an image that needs more than one tile of the mosaic, a slightly increase of measured time can be observed where the images join. As well as this difference is not outstanding (below 0.1 seconds) it's perfectly acceptable.

ENVI format

This raw binary format, even though useful in analysis tasks, it's clearly insufficient to serve images in Internet, because the accessing times offered are excessive, appearing unacceptable values. Thus, in general, the first access of every row has been around three seconds while the rest of cells has been around 0.1 seconds.

CONCLUSIONS

After doing different experiments and analyze the results, is possible to rule out ENVI format as well as IMG. The binary format of ENVI has shown times pretty larger than the rest, with unacceptable outliers. This format, being perfectly useful for remote sensing and image analysis tasks, results inadequate for visualization of geographic information through Internet. IMG format besides a regular performance, in some cases it shows anomalous results. Again, a widely used format in geomatic fields by its simplicity, pyramid enabled and so on is not very convenient to be used with a map server.

Both best formats have been ECW and GeoTIFF. They entail in an excellent way with practically every condition studied. ECW format results better in compression rate, easiness management, and simple creation. Perhaps the worst inconvenient is the impossibility of changing any creation parameter except compression ratio. Finally, this format has turned out some irregular access times.

GeoTIFF format has been the most convenient according to our results as it offers pretty good and regular times. The worst characteristic of this format is the amount of hard disk needed as the image is not compressed. Moreover, it is necessary to create mosaics to obtain detailed images of large areas and in consequence, it is more difficult to manage than the rest of formats. The best advantage on the other hand is the possibility of changing the creation parameters to adapt the image to every situation.

Our future work in this project is the repetition of this experiment to improve the results working with the GeoTIFF image and varying the creation parameters (block sizes, mosaics, compression algorithms …). This way, we hope to get better times, obtaining conclusions about the most adequate parameters to optimize a GeoTIFF image to present large amounts of raster data in Internet.

ACKNOWLEDGMENTS

This work has been partially supported by the research project “Information and Management in Local Administration” BIA2003-07914 from the Spanish Government (CICYT) and the European Union (ERDF funds).

BIBLIOGRAPHY

Clancey, William J., 1997 Situated Cognition: On Human Knowledge and Computer representations. Cambridge University press. The Edinburgh Building, Cambridge CB2 2RU, United Kingdom. ISBN 05214449004, pp 406.

Coll, Eloína, Martínez, José-Carlos and Sanz, Jorge-Gaspar, 2005, Introducción a la publicación de cartografía en Internet. Technical University of Valencia, Valencia.

Hendry, Flavio, 2004, Best practices for web mapping design, Mapserver User Meeting.

Levene, Mark and Loizou George, 2003, Why is the snowflake schema a good data warehouse design?, Information Systems, 28(3), pp 225—240.

Lime, Steve, Morissete, Daniel et al., 2005, University of Minnesota Mapserver Project, http://mapserver.gis.umn.edu

Maindonald, John and Braun John, 2003, Analysis and Graphics using R, an example approach, Cambridge University Press, Cambridge.

R Development Core Team, 2005, R: A language and environment for statistical computing, R Foundation for statistical computing, Vienna, Austria.

Warmerdam, Frank, 2005, Geospatial Data Abstraction Library (GDAL), http://www.gdal.org