

A PRACTICAL APPROACH FOR SERVING LARGE AMOUNTS OF GEOSPATIAL DATA VIA COMPUTER NETWORKS

E. Yilmaz^{a*}, M. Ocak^a, H. Tastan^a

^a GCM, General Command of Mapping, 06100 Cebeci, Ankara, Turkey - (erdal.yilmaz, mustafa.ocak, hayati.tastan)@hgk.mil.tr

KEY WORDS: Geospatial Data Transfer, JPEG 2000 for Geospatial Data, Remote Rendering

ABSTRACT:

In this paper we describe our efforts to develop a system that generates interactive Virtual Environment at the client PCs, by rendering geo-spatial data that is stored at a remote server. The end-users of geospatial data generally demand visually appealing, up-to-date and quite accurate data that covers a vast landscape. Considering these requirements we tried to implement a distributed servers/clients architecture system that significantly uses data compression, computer graphics and network technologies. Tera-bytes of geospatial data have been reorganized in a way that achieves reasonable data transfer rates even in low-bandwidth networks. Our current progress gives promising results for the future and the final system is supposed to meet the requirements of remote geospatial data users.

1. INTRODUCTION

Serving geospatial data of large areas such as a country or whole earth via the Internet or intranets is a hot application challenge. Google Earth™, World Wind™ and similar software tools set new standards in this area and showed that even at low bandwidths it is still possible to serve large data chunks, almost in real-time. Such applications encouraged geospatial data producers to serve their products via World Wide Web, abbreviated as web. Actually interactive web mapping has been in existence for almost a decade (Rhyne, Brutzman, & Macedonia, 1997) and various commercial or freeware tools to achieve web mapping by using different approaches (Huang, 2003). While some solutions only capable of displaying geospatial data in a limited extent, some can handle small amount of data and support only raster or vector and most of them suffer from speed and performance issues considering the network communication. In this sense only a few software tools are capable of combining elevation model, images and data to create interactive 3D virtual world by using network streaming technology (Remondino, 2005).

In this study our goal is to develop a multi-user remote rendering system that uses scanned topographic maps, aerial photographs, satellite images, relief maps, elevation data together with vector (graphic) and alphanumeric (descriptive) data. It is aimed to transfer huge amount of data via computer networks and generate 2D/3D seamless virtual environment (VE) at a reasonable interaction rate. Following the evaluation of cost and performance issues of existing web-mapping solutions we decided to build a new software tool from scratch by integrating wavelet technology, network communication and advance computer graphics algorithms. In order to achieve above mentioned goal we decided to develop distributed servers/clients architecture application as shown in Figure 1. In the servers, we store raster, vector and alphanumeric data that are organized in one degree geo-cells. No special geodatabase solution or third-party image management tool is dedicated at this stage. Data organization is designed to minimize search time and decrease response time to meet real-time application constraints.

3D visualization applications that are built on aerial photography or sub-meter satellite imagery and scanned maps require a strategic choice for data storage (Martin et al., 2006), since geospatial data of order of tera-bytes are common in such systems. Increased accuracy means higher resolution and larger file sizes. Although the cost for storage media has been decreasing continuously, data sets at tera-bytes size are still expensive to store and difficult to manage. Besides cost and management issues, larger files also require larger bandwidth and more I/O (Input/Output) operation. It is a well-known fact that data transfer is the largest bottleneck in modern computers even between the memory and processor. This problem grows significantly if it is aimed to send large amounts of data by using computer networks (Ottoson, 2001).

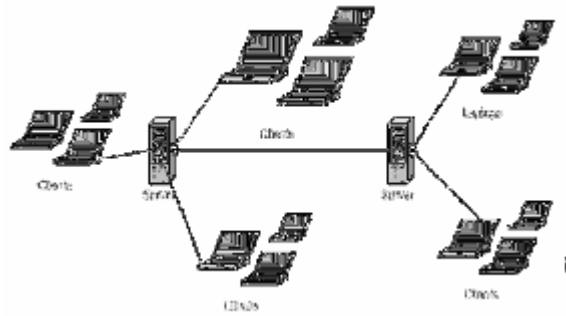


Figure 1. System Architecture

Since time is very crucial for on-demand remote access to data sets over the computer networks, current bottlenecks such as bandwidth and latency must be overcome. If it is not possible to increase bandwidth then file sizes must be decreased. At this point data compression seems to be the best solution for large amounts of geospatial data.

Network latency can be minimized by sending geospatial data in a single and small packet (Martin et al., 2006). Distributed servers choice has also positive contribution in this issue. We discuss these details in 2.3.

* Corresponding author.

Client side also plays an active role in this system by processing data in real-time and generating interactive 3D VE. Our goal regarding the final VE is to display everything from space to street level in a seamless and integrated manner. In order to achieve such a big rendering task we decided to use previously developed high performance VE editor (Yilmaz et al., 2004) as the basis of graphics engine. This tool is ready to render detailed scenes at lower altitudes (street level), so at this moment our focus is rendering virtual worlds as seen from higher altitudes.

2. IMPLEMENTATION

This part covers initial implementation which has a fairly simple interface but powerful framework which covers major issues regarding real-time web based visualization. We have already started to develop first stable version which is supposed to meet current end-user requirements regarding the remote rendering and geospatial data visualization in real-time. In this section we address topics that are related with geospatial data management in detail. Network and computer graphics issues are not covered deeply.

2.1 Raster Image Data

It is a well known fact that imagery is an essential component to everyone requiring land information since it provides reality and detail not found in other data sources. It is a very valuable source and vital for fast, effective and accurate decision-making (ER Mapper Solutions White Paper, 2004). Since it is closer to the usual human perception, it provides a better base for many GIS solutions as well as visualization applications.

As we already mentioned this application stores raster data at the servers. Servers are responsible for extracting geospatial data of requested regions and inserting them into data-transfer-packets. Although this analogy simplifies the process, things are not that simple in the practical life. The amount of geospatial data which is around the order of tera-bytes presents the following challenges.

- Data storage cost
- Data management complexity
- Data transfer latency (storage to processor via data bus)
- Processing time
- Network latency (server to client via wire/wireless networks)

Today data storage cost is less than \$1/giga-byte. Although the cost decreases rapidly, increased image resolutions keep this issue still alive. This cost is also important for us since we plan to use distributed servers approach. Besides cost another issue is the complexity of data management due to large data size. Disk I/O cost, which takes longer than CPU (Central Processing Unit) time with present COTS (Commercial of the Shelf) hardware causes latency and annoys end-users.

Network transfer time is the most important bottleneck and it must be overcome, as long as the goal is to develop web-based real-time visualization system. Current bandwidths and network technology force the use of small data chunks for web-based real-time applications. Network latency, packet loss, collision are among the problems introduced with network infrastructure. Since it is crucial to display the progress to the end-users data must be converted into transferable size. This issue is also covered in section 2.3.

The only existing solution to overcome above mentioned problem is to compress image data as much as possible while meeting visual quality expectations. Many image compression

algorithms have been proposed and considering the demand for better compression, it can be evaluated that this issue will always be a hot application topic. Most of the state-of-the-art image compression formats (MrSID, ECW, JPEG2000) use wavelet technology to achieve higher compression ratios. In the literature there exist several studies regarding the compression of geospatial data by using wavelet technology (Ueffing, 2001; Li and Hu, 2004). Extensive research has shown that wavelet technology based image compression algorithms provide perfect compression ratios and produce good visual quality. In this study we have also preferred wavelet method and decided to use future industry standard candidate, JPEG2000.

JPEG2000 addresses various problems related with imagery, which are listed as follows.

- Provides lossless and lossy compression
- Encodes very large images
- Supports progressive transmission by pixel accuracy and by resolution
- Robustness to the presence of bit errors and random code-stream access (Charrier et al., 1999).

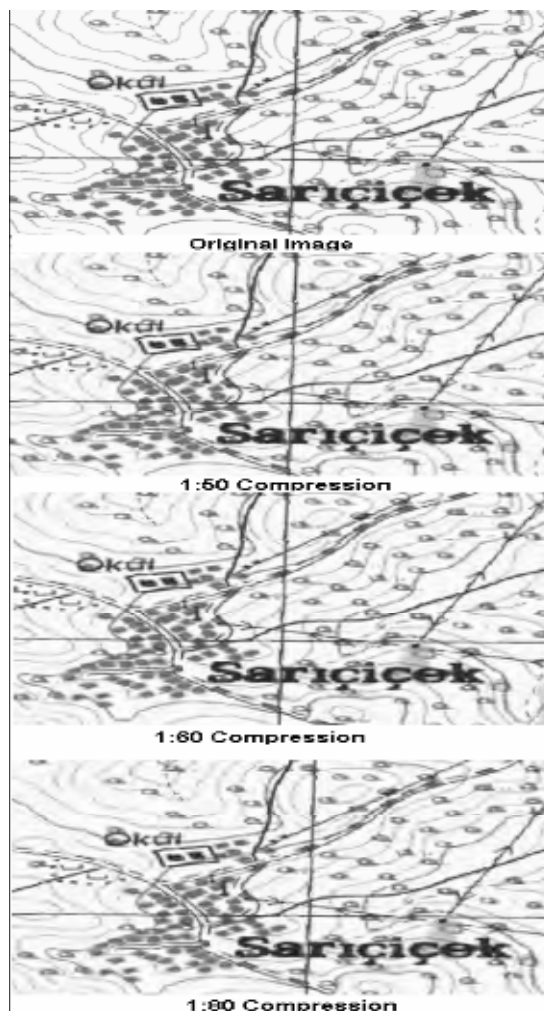


Figure 2. Compressed Images at Various Ratios
Regarding the determination of compression ratio and visual quality we have simply considered the readability of textual information on scanned-maps and ease of recognizing figures such as buildings, fountains or trees. After several trials we

observed that 1:50-1:80 compression ratios meet this constraint as shown in Figure 2. In this figure we can see the blur at the contour lines and the information lost at sharp details or edges as the compression ratio increases. This figure also shows that even 1:80 compression ratio can be used for web-based visualization applications where visual quality is not crucial. One critical issue with wavelet-based JPEG 2000 technique is the ability of encoding an image at multiple resolution levels. This is similar to the image-pyramiding approach but eliminates data redundancy and thus prevents file-size overhead. If it is compared to the conventional 1:4 pyramid structure, it can be seen that this property directly eliminates 30% data redundancy caused by mipmaps. This structure also helps us in rendering issues since it lets developers to control mipmaps manually by discarding graphics API (Application Programmers Interface).

2.2 DEM Management

Digital Elevation Model files keep height values at discrete samples of a continuous surface. Height data values are mostly considered in meters and represented by a two-byte short integer (-32767/32768) which keeps the highest peak on earth in this range. There are various DEM file formats in ASCII or binary structures such as USGS DEM, DTED, SRTM etc. File sizes of popular DEM file models make them unsuitable for real-time remote rendering applications. Considering current bandwidth, network technology and hardware constraints as explained in section 2.1, similar precautions must be taken in order to decrease DEM file size.

There are various approaches for DEM compression. Current trend in DEM compression issue is to compress data as much as possible without losing precision. This approach can be softened in our case since network communication performance comes first. Considering bandwidth limitations, our first interest is easily transferable file sizes rather than accuracy.

Several studies have been realized in the literature regarding the compression of elevation data by using wavelet technology (Creusere, 2001; Ottoson, 2001; Oimeon, 2003). Ottoson recommended 1:6-1:48 compression ratios for RMSE between 1.5-2.4 m (same RMSE with the original data). Oimeon concluded that using the discrete wavelet transform produces largely acceptable results with minimal storage overhead. The results of this study also showed that various terrain models reconstructed from %5 of wavelet coefficients look almost same of original models. Creusere worked on lossless compression techniques and obtained 1:5-1:13 compression ratios for different tiles at 3-10 meter resolution. He reported 1:2.5-1:6 compression ratios for the same area with LZ77 algorithm.

In the light of these studies, in order to compress elevation data we decided to use wavelet technology rather than conventional compression algorithms. After this decision we must decide for the encoding method and compression ratio. Since the main bottleneck in our implementation is bandwidth and network communication issues, we have to compress data as much as possible without sacrificing accuracy. Regarding the method, we preferred JPEG2000 encoder and decoder since we already used them in raster image modules. Second and most important point is the compression ratio which should be at least 1:10, considering the minimum performance requirements for our web-based real-time visualization implementation.

In the implementation we split 16 bit elevation data into 2 different channels as low-order byte and high-order byte. Low order byte channel keeps values of 256 m. discrete steps. High-order byte channel corresponds to in-between values as shown in Figure 3. The terrain profile in this figure also shows us that although first channel has possible 256 values, there are only 4

different entries. This practical result is valid for almost all geo-cells since elevation changes are limited in small regions.

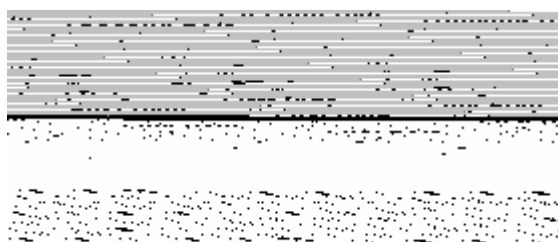


Figure 3. Elevation Data Structure

This channel splitting strategy helps us not only for having good compression ratios but also for maintaining data security. We plan to transfer and store this channel at the client PC since it is not useful without the second channel. This channel also has information that can be used by higher resolution DEMs. It is very clear that lossless compression must be applied to this channel since 1 unit shift means 256 meters of error.

Although Creusere reported 1:5-1:13 compression ratios for lossless wavelet decoding, this result is valid for 16 bit data set. In this implementation we achieved average of 1:20 compression ratio for the first channel since it has limited discrete values as shown in Figure 4.

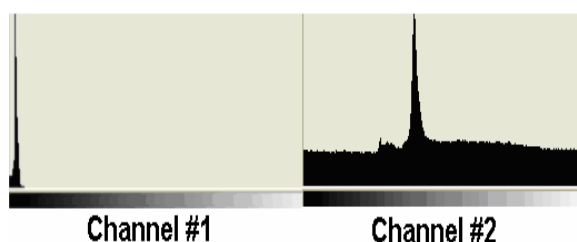


Figure 4. Histograms of Channel#1 and Channel#2 of 1st DEM Tile

	Compression Type			
	(lossless)	lossy		
		1:5	1:10	1:20
Channel #1	70,605 bytes (1:20)	-	-	-
Channel #2	1,023,699 bytes (1:1.5)	288,480 bytes	144,240 bytes	72,120 bytes
	RMSE= 0 m	RMSE= ±6 m	RMSE= ±16 m	RMSE= ±28 m

Table 1. Compression Results of 1st and 2nd Channels

Table 1 covers details about test for DEM compression. We used four different geo-cells in this test (1 degree*1degree, height value interval 3 seconds, RMSE=±16m). Each cell has 9,732,096 bytes in DEM format and 2,884,802 bytes in binary raw format. Binary raw format is used as the basis for determining compression ratios.

For the second channel we applied 1:5, 1:10, 1:20 compression ratios and determined to use 1:10, considering file size and reconstruction quality constraints. Overall compression ratio is 1:15.

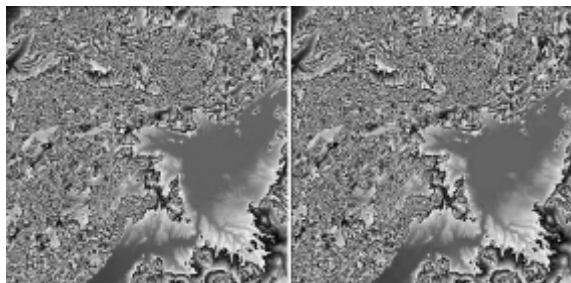


Figure 5. Grayscale Image of 2nd Channel of Tile #1 (left: original, right: 1:10 compressed)

Figure 5 shows the differences between the original and 1:10 compressed high-byte channels. It can be easily seen that compressed version still very close to the original channel.

2.3 Network Communication

Network communication is very important part of this study. Real-time application constraints and the amount of the data that must be transferred require special attention. Latency always exists since it takes time for a signal to pass through wire, but it is our responsibility to decrease this time to not to annoy end users. Distributed servers architecture choice helps in decreasing latency and share the network load, but something more has to be done. Transferring data in small packets is simple but effective approach in our case. Although opposite is advised because of increased amount of header bytes overhead (Tanenbaum, 1996), this rule of thumb can be ignored in our case to provide visual sensation of real-time. As the user moves around the VE, requested data is transferred dynamically. Executable at the client-side, i.e. client software tool, is designed to render newly arrived data immediately, in order to show progress to the user. We plan to increase number of communication ports in order to transfer more data in less time. Another approach is to let clients to store received data sets in storage device to eliminate data transfer duplication, if the client request data of the same area in that session or later. This action certainly prevents data transfer duplication but requires precautions regarding the data security.

2.4 Remote Rendering

The last important element of this study is related with computer graphics issues. It is client PCs task to render transferred data in real-time. At this moment we used OpenGL as rendering tool but we also plan to add DirectX option to benefit from the advantages offered by this API. In order to achieve interactive frame-rates we take advantage of well known techniques such as frustum-culling, LOD (Level of Detail), quad-tree structure etc. Current prototype application lets user to freely move around the VE and interact with the scene. We will skip this part since we use previously developed graphics engine. Detailed information has been given in related publications (Yilmaz et.al, 2004).

2.5 Software Architecture

Figure 6 illustrates the detailed software architecture. Server executable, i.e. server software tool, simply accepts requests from clients and replies them by transferring requested geospatial data. In order to keep simplicity we have not assigned any commercial or free database management system. Geospatial data is ready to insert into a single or a few network packets. Data folders are designed in a way to minimize search time. There exists a special module that handles requests from different clients. It tries to be fair by checking parameters such as # of requests from each client, packet send time etc. Data manager module provides an important functionality by caching requested data sets, since it is computationally expensive to extract them. This module simply checks use flag and releases least-used data set when memory became full. Prime time test to check server performance is not completed yet, so it is not possible to report up-to how many clients are supported.

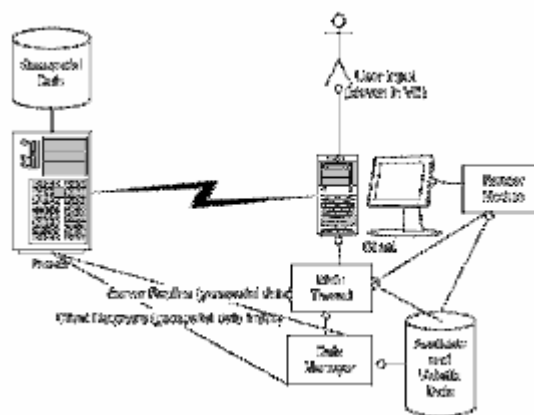


Figure 6. Detailed System Architecture

Client executable performs different tasks ranging from network communication to rendering. Initial scene is rendered by using very low resolution data set that is distributed with setup package. After handshaking with the server, remote data sets are ready for transferring. Data request is prepared according to virtual camera position. The lower the virtual camera the higher resolution data is requested. By using longitude, latitude and altitude of virtual camera geo-tiles are calculated and data-manager pushes a new item onto the stack if it is not available in the memory or hard-disk. Incoming tiles are sent to the data manager module and processed to be texture-maps or terrain model etc. Similarly incoming geo-tiles are kept in the memory for possible future requests. Currently we support limited data download to the hard-disk due to data security issues. User input is used to change virtual camera and trigger modules such as data-request, render etc. Render module is responsible of generating VE and refreshing the scene when needed.

2.6 Interaction, User Interface and Several Features

Besides setting new standards in web-based geospatial data serving, Google Earth™, World-Wind™ and similar digital globe applications introduced several interaction techniques and new GUI (Graphical User Interface) styles. Grid lines, gazetteer items, measuring tools, etc. are all designed to meet user friendliness and computational performance issues. In this study we mainly inspired by these techniques. Some of the features of our study are as follows.

We tried to realize dynamic transition from abstract model representation to reality as illustrated in Figure 7. In this figure single tree icon represents a forest at high altitudes (top image) and it is replaced with photorealistic virtual trees at lower altitudes (bottom image). In the future we also plan to include dynamic objects such as vehicles or human-beings at the lower altitude scenes.

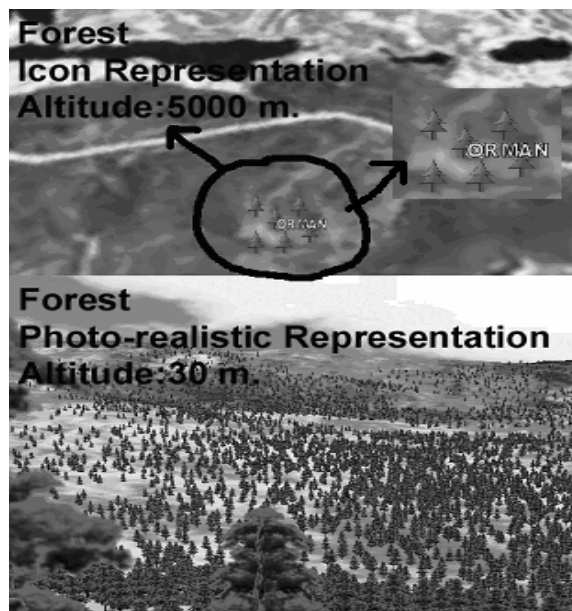


Figure 7. Transition from Abstract Model to Reality
 Another property that we tried implement is instant transition between different view modes. This property lets user to see area of interest in different view modes as shown in Figure 8. In these modes geometric model is draped with different texture maps. This is quite easy and fast if related texture maps exist in the buffer or already downloaded to the data storage. But it takes more time if texture maps are not available.

3. CONCLUSION AND FUTURE WORK

In this study we have developed a prototype system that is capable of generating interactive VEs at the client PCs in real-time by using remotely stored geospatial data. Tera-bytes of original data sets are converted into transferable and easily manageable sizes which is the first issue in such applications. Although developing this application from scratch gives us the freedom in implementation we have encountered many issues handled by third party solutions.

In the future we plan to improve network performance and develop a better and robust tool that is capable of managing traffic between a server and few hundred clients. Our urgent future work is prime-time test simulation by assigning a few hundreds of virtual users per server. Following this test we will have the evaluation criteria such as "Performance/# of users" or "Performance/data size" and try to shape up next version according to these values. This will also help us to compare our results with the existing solutions.

Better compression of elevation data is another future issue. We believe that better compression ratios than 1:15 can be realized by meeting accuracy constraints.

Another future work might be to analyze new ISO standard 19128, which specifies operations related to web map server interface and make improvements in the light of this document.

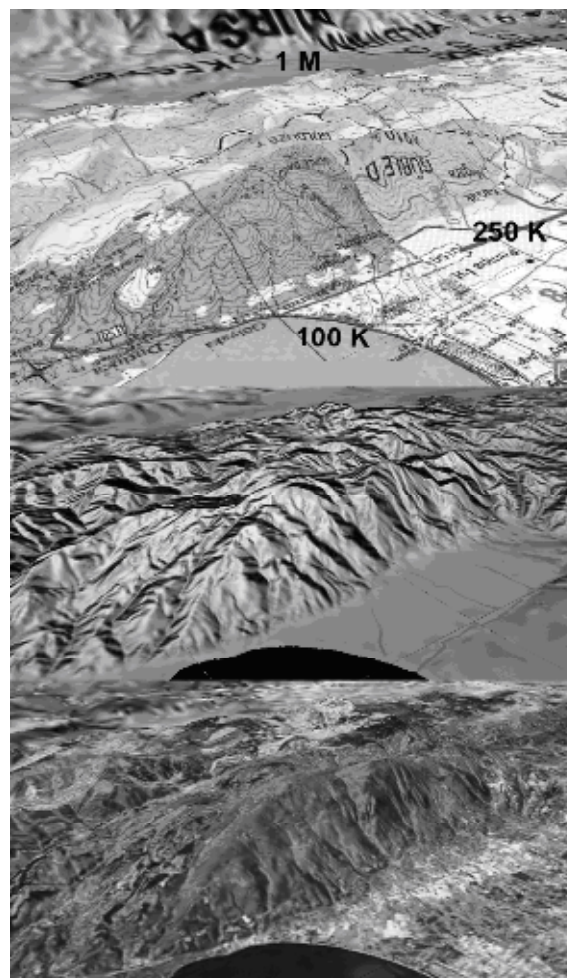


Figure 8. Visual Modes (Scanned Map/Relief Map/ Satellite Image)

REFERENCES

- Charrier, M., Cruz, D.S., Larsson, M., 1999. JPEG2000, the next millennium compression standard for still images. In: *Proc. of the IEEE International Conference on Multimedia Computing and Systems (ICMCS)*, Vol. 1, pp. 131-132.
- Creusere, C.D., 2001. Compression of digital elevation maps using nonlinear wavelets. In: *Proceedings of 2001 International Conference on Image Processing*, Vol.3, pp.824-827.
- ER Mapper Company, 2004., Integrating imagery with your GIS. www.ermapper.com/document/doc.aspx?doc_id=99 (accessed 12 Jun. 2006)
- Huang, B., 2003. Web-based dynamic and interactive environmental visualization. In: *Computers, Environment and Urban Systems*, 27, pp.623-636.

Li Q., Hu Q., 2004., 3d wavelet compression to multiple band remote sensing images based on edge reservation. In: *Proceedings of XXth ISPRS Congress*, Istanbul, Turkey, pp:55-60.

Martin, A., Gesquire, G., Puech, W., Thon, S., 2006. Real time 3D visualization of DEM combined with a robust DCT based data-hiding method. In: *Proceedings of SPIE Visualization and Data Analysis 2006*, Vol. 6060, pp.623-636.

Oimoen, M.J., 2003. The Effects of Wavelet Compression on Digital Elevation Models (DEMs).
<http://gis.esri.com/library/userconf/proc04/docs/pap1748.pdf>
(accessed 10 Jun. 2006)

Ottoson, P., 2001., Compressing digital elevation models with wavelet decomposition. In: *Proceedings of ScanGIS*, pp.15-31.

Rhyne, M.T., Brutzman, D.P., & Macedonia, M.R., 1997., Internetworked graphics and the web. In: *IEEE Computer*, 30(8), pp.99-101.

Remondino, F., Lecture Notes UNESCO Training Workshop for Site Managers, "Photogrammetry for Natural and Cultural Heritage Site Documentation, Mapping and Visualization" Campache, Mexico, 25-27 November 2005.
<http://www.radiig.informatik.tu-muenchen.de/ISPRS/WG-III4-IV2-Report.html> (accessed 24 Jun. 2006)

Tanenbaum, A. 1996. *Computer Networks*. 3d ed. Prentice-Hall, New-Jersey, pp:541-546.

Ueffing, C.M., 2001. Wavelet based ECW image compression. In: *Proceedings of Photogrammetric Week '01*. Heidelberg, Germany, pp:299-306.

Yilmaz E., Maras H.H., Yardimci Y.Ç., 2004., Photo-realistic scene generation for Pc-based real-time outdoor virtual reality applications. In: *Proceedings of XXth ISPRS Congress*, Istanbul, Turkey, pp:615-620.