A Web-based Technique to Generate and Visualize 3D Scenes from Global to Local Views

Liqiang Zhang[1], Chongjun Yang[1], Donglin Liu[1], and Hongwen Lin[2]

[1]Institute of Remote Sensing Applications, Chinese Academy of Sciences, P.O.BOX 9718,Beijing, 100101,P.R.China,
[2]Science College,Southwest University of Science and Technology,Mianyang,Sichuan province,China.

Abstract

Global terrain visualization is a very challenging subject. Adopted a client/server architecture for complex, dynamic and distributed 3D scenes, based on hypertext transfer protocol and oriented object methods, a virtual scene for the earth which merges satellite and aerial imagery with DEM data using the Earth itself as an interface is built. Database organization functions in the server, data transfer and real-time display mechanisms are discussed in this paper; then pyramid data structure and wallet analysis principles are introduced to simplified data. How to construct high efficient indexes and algorithms of operating 3D landscape on the browser are also discussed. Finally an experiment is performed using different levels of detail data.

I. Introduction

3D scene virtual reality technique is one of the most important components that consist of "Digital Earth" framework, which has made great progress in recent years. A large body of previous work has addressed issues in modeling, representing, and manipulating spatial data for geographic information systems [1], e.g. Jürgen [2] illustrates integrating a GIS with a visualization toolkit by embedding visualization objects into the GIS and by specializing visualization classes by object-oriented techniques. Some researchers have also developed their 3D terrain application systems. VGIS [1] uses multiresolution representation and rendering techniques for large, complex terrains and height fields using polygonal meshes, it also relies on multiresolution techniques to allow truly interactive visualization with its geometrically complex terrains. Keyhole [3] has developed *EarthViewer* which fuses massive vector, image and points of interest (POI) datasets into a multi-terabyte 3D model of the globe, compiling this data into a scalable server environment, and delivering this data using patent-pending technology to PCs, PDA's, set-top boxes and wireless devices.

For the most part, some of the terrain models that have been produced are not multi-resolution, and hence are limited in the amount of geometry and imagery that can be effectively displayed. In addition, only little work has yet faced the issue of modeling large-scale areas at resolutions below 1 meter. However, most of the existing applications run on PCs or the organization's intranet, but not on the Internet.

3D virtual scene for the earth having been built in this paper put DEM data, satellite and aerial imagery of the globe, the details of which can be as fine as 1m,into a database, releasing them through the Internet or Intranet, constructing 3D virtual terrain of the earth. Users can see a country or an area, even a small city or even specific buildings on the Brower through Internet.

II. PROPOSED CONSTRUCTING 3D MODELS AND DISPLAY

Constructing 3D models

A geodetic (or geographic) coordinate system related to the ellipsoid is used to model the earth, e.g. the latitude-longitude system. *GlobalViewer* uses a right-handed, geodetic coordinate system [7] to model all objects in 3-D space. In the coordinate system, the Z axis is defined as the outward normal of the surface at the origin, while the Y axis is parallel to the intersection of the tangent plane at the origin and the plane described by the North and South poles and the origin, that is the Y axis is orthogonal to the Z axis and points due North. The X axis is simply the cross product of the Y and Z axes, so the three axes form an orthonormal basis [3], where all locations are specified in units of meters as an (x, y, z) offset from the center of the planet. This choice of orientation is very natural as it allows us to approximate the "up" vector by the local Z axis, which further lets us treat the height field as a flat-projected surface with little error. Hence, the height field LOD algorithm, which is based on vertical error in the triangulation, does not have to be modified significantly to take the curvature of the Earth into account.

To any point T on the earth surface, the conversion relationship between the world geodetic coordinate system $(B,L,H)$and geodetic coordinate system $(X,Y,Z)$is as follows:

1082-4006/03/0901~2-14 $5.00

$$\begin{cases} X = (N + H)\cos B \cos L \\ Y = (N + H)\cos B \sin L \\ Z = [N(1 - e^2 + H)\sin B] \end{cases} \quad (1)$$

In the above equations (1), N is the curvature radius of the ellipsoid; e is the first eccentricity of the ellipse; B, L respectively presents the Geodetic longitude and latitude; H is the geodetic elevation; where,

$$\begin{cases} N = \dfrac{a}{W} \\ W = (1 - e^2 \sin^2 B)^{\frac{1}{2}} \\ e^2 = \dfrac{a^2 - b^2}{a^2} \end{cases}$$

where, a, b respectively presents longer radius and smaller radius, $a = 6378.140km$, $b = 6356.755km$.

We divide the ellipsoid into two parts: one is the North and south poles; the other is the part beside the two poles. In the second part, we make sure the normal of the patch is defined as outward from the origin. One of the advantages is that the scene is more stereo and smooth. In the poles, there is only one point respectively on the two poles. Therefore, *GlobalClient* constructs the triangulation fans centered in the poles. Figure 6 shows a part of the global triangle model.

The major problem with 3D virtual earth terrain models is their big size, which effects both rendering speed and the download time. The models of terrain and their details can be nearly infinite. High-quality online-visualization of virtual earth in real-time is a demanding task even for a state-of-the-art PC, although hardware is improved in dramatic pace. The rendering and data transfer problems are even more severe for any PC [7]. So data had to be made highly simplified to enable reasonable rendering and download time. In general, the frame rate of real-timely creating 3D scene is no less than 15 frames per second and the slowest rate shouldn't less than 10 frames per second. Therefore, in order to ensure the precision of the scene, accelerate the display speed and obtain the effect of real-time interaction. We must simplify and reasonably organize data blocks, besides using some techniques, e.g.OpenGL double buffer technique, display list technique, accelerated functions of map and textures brought by computer hardware and software techniques.

Data organization, simplification and display

An M*N regular grid may be taken as an image having M*N pixels, the height value on every grid's point of intersection may be taken as the gray value of the image, we take the basic physiognomy as low frequent information, and the detail physiognomy as high frequent information. So we may transform the DEM data simplification into two-dimension image simplification [1]. Because image $f(x, y)$ is two-dimension matrix, sampled image is usually accomplished in the binary grid.

$$\Gamma_j = 2^{-j} Z^2 \quad (2)$$

In equation (2), when $j \to +\infty$, Γ_j becomes refine; when $j \to +\infty$, Γ_j becomes coarse.

Wavelet analysis is an applied mathematic theory that has been developed in recent years, it is called 'microscope on mathematics', because it has such properties as good temporal domain and frequent domain, moreover, it handles sample space from coarse to refine in temporal and spatial domain to resolve the high frequent components, so we can find out any detail of the image like microscope that adjusting the focus automatically to see out the different distance's objects. Wavelet analysis is introduced to simplify DEM data and image data in this paper, it can finish the following task: 1) With sampled grids changing from refine to coarse, the information of original images is not needed. Wavelet transmission can simplified sampled image f_{j-1} on Γ_{j-1} according to the sampled image f_j on Γ_j 2) it can directly simplified original DEM data into a multiple of M^2, and we can get continuous model data in the virtual geographic environments (see Figure 1).

According to the methods and principles of multi-system wavelet transmission, quad-system wavelet is adopted to simplified global data into pyramid data structure. Every layer data also are divided into some blocks based on performance of the computer's hardware; finally the simplified data is input into database.

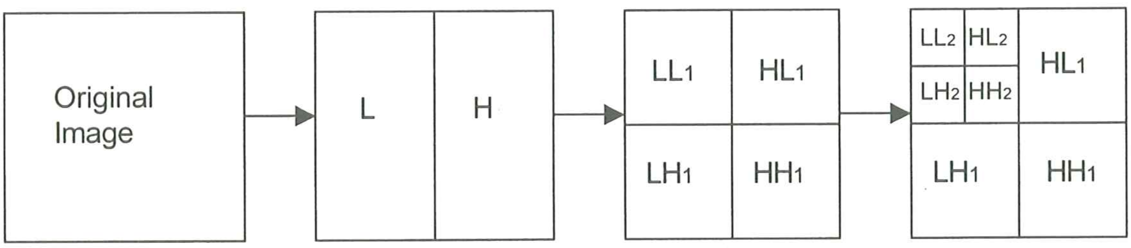

Figure 1. Hierarchical decomposition of an image

The simplification of DEM data and image data follow the same principle. Namely, they have the same levels and every block covers the same spatial extent. An image block only has a DEM data block and a DEM data block only has an image block. The main purpose is that DEM data and over headed image data can be organized conveniently and the programs achieve the function.

File storage is a traditional method for data storage; the advantage of file storage is flexible and convenient. However, file storage unsatisfied massive data organization. Large database system provides convenience for massive data storage using its strong database organization functions.DEM geometric data blocks have spatial positions, using their left-bottom corner's longitude and latitude(L_{min} , B_{min})and right-top corner's longitude and latitude(L_{max} , B_{max}) to present each block's spatial position, so the database also stores spatial information besides data itself information. In order to ensure network and data security, we take some measures to deal with the delivered data such as transforming DEM data and other vector data's formats into binary data formats and providing visiting power.

In general, High efficient data index mechanism is built, after data are input into the database. We should consider such information as data blocks' geo-coordinates, level of detail and spatial position etc.the format of the index file adopted in this paper is as follows:

<EarthVR-1.0>// file header
[Layers] // flag
Begin // file begins
Geometric data record's total number
... ...
Level *iLevel*, data's *ID* number (1,2,3), location information of data $(L_{min}, B_{min}, L_{max} B_{max})$, data's filename
... ...
End// file end

In the above index file format, the signification of the ID number is: 1) denotes that there are no image data blocks and only are DEM ones in the *iLevel* level; 2) denotes that there are no DEM data blocks and are only Image ones in the *iLevel* level; 3) denotes that there are both in the *iLevel* level. If there are none in the *iLevel* level, the application will automatically download the coarser level's data to render the landscape. In this paper, we make DEM data filename same as Image data filename, except the extension of the filenames are different.

When the scene is near to the point of view, the resolution of the terrain is higher and visual field becomes accordingly smaller, the application should request DEM data and image data that have higher level of detail. When different level's data switch each other, we should ensure that they are spatial and temporal successive. Namely, the connectivity between the neighboring level's data is continuous and seemless. With the eye closing to the earth,' jump' on vision should be avoided when changing in neighboring data. In order to smoothly switch between the levels, we must deal with the relationship between the display data's level and the height, visual angle. This is a very important problem for the virtual terrain roaming, which is mentioned in some reference, but some references have discussed very simply [12]. The authors in this paper mainly put forward to a algorithms to solve it.

The viewers can browse the landscape from the whole earth to one continent, a country, a province, a city, or even a smaller region. In this procedure, the line of sight and view of angle are changing. The line of sight refers to the distance between point of view and the reference point, the point of view can be determined by the reference point and the given scaling factor. The author adopted the equation(3), (4)to determine the scaling factor f_i and view of angle α_i in the i level:

$$\begin{cases} f_i = f_{min} + \dfrac{(f_{i-1} - f_{min})}{n} - \dfrac{e^i}{1+e^{i+1}} & i>1 \\ f_1 = f_{max} \end{cases} \quad (3)$$

$$\alpha_i = \alpha_{min} + (\alpha_{max} - \alpha_{i-1})(f_i - f_{min})/(f_{max} - f_{min}) \quad (4)$$

With the point of view closing to the virtual earth surface, the landscape's details are clear, in the condition of $f_i < f \le f_{i-1}$, the clients request data blocks in the i level from the server (the crucial scale and view of angle are shown in Table 1, to the 8th level's data blocks).

In order to accelerate the display speed, only data blocks inside the view volume are transferred every time, and the ones outside of the view volume are not displayed. Because data blocks are organized in the form of blocks and layers, switching among different levels' data blocks, the users request the searching areas $\{fx\min, fy\min, fx\max, fy\max\}$ which centers in the reference point, the width is h_i , and h_i is determined by the following equation (5).

$$h_i = 2(f_i - 1)tg\frac{\pi\alpha_i}{360} \quad (5)$$

In the above equation, i is the level of detail, f_i is the scale factor between the reference point and the eye point, α_i is the visual angle.

We can finish searching data blocks using the index file. Now we introduce the algorithms that search the current level's data from the database.

1) When the application is initializing, it reads index data into memory array *m_pPool* [i];
2) According to the values of the line of sight and visual angle (determined by the equations (3), (4)), the application computes the current display level of DEM data and image data.
3) Finding out *iCurLevel* level in the index file, comparing the bound $(fx\min, fy\min, fx\max, fy\max)$ with

the four coordinates of every data block to search the ones falling into the bound. However, we should consider the blocks which one part of it locating in east hemisphere and the other locating in west hemisphere.

4) If DEM data and texture data are not opened and not visible,then loading data and creating display lists.

III. BROWSING 3D TERRAIN VIEWS

The viewer can see any place on the earth from different angles while the line of sight remains unchanged. Because the vector direction of one point on the earth is different, operating the terrain is not as simple as we expect. The methods are introduced to implement the function in this paper.

We have known the longitude and latitude coordinates(B,L) of the reference point *A*, the angle between the coordinate of original eye point *E*(the line between *E* and A points to sphere center) and the current view line (see Figure.2). Now we need to find out the coordinate (x, y, z) of the current eye point R. The algorithms are as follows:

1) Calculating the coordinate (x_1, y_1, z_1) of original eye point *E*.

To any point *T* on the earth surface, the conversion relationship between world geodetic coordinate system (B, L, H)and spatial right-angle coordinate system (X, Y, Z)is as follows:

$$\begin{cases} X = (N + H)\cos B \cos L \\ Y = (N + H)\cos B \sin L \\ Z = [N(1 - e^2 + H)\sin B] \end{cases} \tag{6}$$

In the above equations (6),*N* is curvature radius of the ellipse; *e* is the first eccentricity of the ellipse; *B, L* respectively presents the Geodetic longitude and latitude;*H* is the geodetic elevation; a,b presents longer radius and smaller radius respectively, $a = 6378.140km, b = 6356755km$. So

$$\begin{cases} N = \dfrac{a}{W} \\ W = (1 - e^2 \sin^2 B)^{\frac{1}{2}} \\ e^2 = \dfrac{a^2 - b^2}{a^2} \end{cases} \tag{7}$$

Through the equation (7), we can calculate the coordinate (x_0, y_0, z_0) in the spatial right-angle coordinate system of the reference point *A:*

$$\{x_1, y_1, z_1\} = f_s\{x_0, y_0, z_0\} \tag{8}$$

Calculating the coordinate $(x_1, y_1, z_1)$of the eye point *E* through the equation (8). f_s is a scale factor and its value is more than 1.0.

2) Calculating the normal plane APQ that passes through the reference point *A*.The normal vector of the normal plane *APQ* is:
$n = \{a,b,c\} = \{(x_1 - x_0),(y_1 - y_0),(z_1 - z_0)\}$, so the equation of the normal plane *APQ* is:

$$a(x - x_0) + b(y - y_0) + c(z - z_0) = 0 \tag{9}$$

3) Calculating one point Q(x, y, z) on the normal plane. From the equation of the normal plane (equation 9), we can calculate the point of intersection Q(x_3, y_3, z_3) between the normal plane and the vertical line through the original eye point *B*.

$$\begin{cases} x_3 = x_1 \\ y_3 = y_1 \\ z_3 = z_0 - [a(x_1 - x_0) + b(y_1 - y_0)]/c \end{cases} \tag{10}$$

In the above equation (10), $c \neq 0$, namely $z_1 \neq z_0$; if $z_1 = z_0$, the reference point *A* is on the equator, so the coordinate of point R can be calculated directly.

4) From the following equation, the point of intersection P(x_2, y_2, z_2) between the circle $\overset{\frown}{ERP}$ and the normal plane APQ.

Table 1.

Level i	Min f	Max f	Min α	Max α
1	1.7	10	40.01	60
2	1.5	1.7	26.69	40.01
3	1.35	1.5	16.7	26.69
4	1.25	1.35	10.04	16.7
5	1.15	1.25	3.38	10.04
6	1.13	1.15	2.05	3.38
7	1.1001	1.13	0.057	2.05
8	1.100001	1.1001	0.00501	0.057

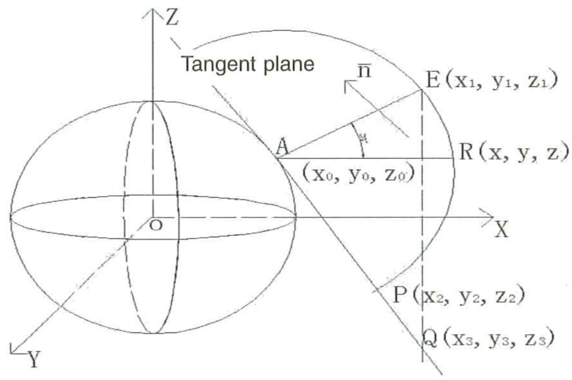

Figure 2. Sketch map of browsing the terrain from different angles

$$\begin{cases} \dfrac{x-x_0}{x_1-x_0} = \dfrac{y-y_0}{y_1-y_0} = \dfrac{z-z_0}{z_1-z_0} \\ (x-x_2)^2+(y-y_2)^2+(z-z_2)^2 = S^2 \end{cases} \quad (11)$$

$$S = \sqrt{(x_0-x_1)^2+(y_0-y_1)^2+(z_0-z_1)^2}$$

Considering the shape of the whole scene is a sphere and the northern direction points to the front of the viewer, so to the reference point on the north sphere, $z - z_0 > 0$,the vector's direction of the line of sight is (0,0,1).To the reference point on the south sphere, $z - z_0 < 0$,the vector's direction of the line of sight is (0,0,1).During calculating the above equations (11), we should discuss the conditions that the coefficients of X, Y, Z are equal to zero and the vector of the circle on the plane XOY.

5) Calculating the coordinate (x, y, z) of the eye point on the scene from which we can browse the terrain from any angle.

$$\begin{cases} aX + BY + cZ = S^2 \cos\alpha \\ dX + eY + fZ = S^2 \sin\alpha \\ gX + hY + jZ = 0 \end{cases} \quad (12)$$

So

$$\begin{cases} x = X + x_0 \\ y = Y + x_0 \\ z = Z + z_0 \end{cases}$$

IV. SYSTEM ARCHITECTURE FOR IMPLEMENTATION

Our application named *GlobalViewer* is designed for scalable distribution and visualization of web-mapping applications. The architecture of *GlobalViewer* basically consists of two layers: *GlobalServer* and *GlobalClient*. *GlobalViewer* has a minimal response time because a major part of the content to be manipulated is generated on the client side, rather than on the server side. *GlobalServer* is a *Daemon* program running on the WWW server. It integrates ISAPI solution,so *GlobalServer* communicates well with *GlobalClient*. The responsibility of *GlobalServer* is to receive *GlobalClient* request, dequeues the highest priority request, and finds out the proper data sets according to the index files

GlobalServer then reads the compressed data sets (*ECW* or *CAB*) from disk if they exists, or else compresses the data into CAB and ECW formats. After dispatching the needed data to the client, the server may have to do additional processing. LOD state and other parameters are also generated and initialized before the request is completed, and *GlobalClient* making the request is notified by submitting an acknowledgement message. Modules can be attached to the servers to handle paging of "arbitrary" data types. The core of the server is merely a dispatcher of jobs to be executed by these modules. In this particular case, the elevation data may not even have to be read in if it already resides in memory since the servers have access to the shared memory cache.

GlobalClient is a COM component, which is a client module. *GlobalClient* is responsible for identifying the position of the view point, LOD state and the vision sensitive area, making data requests to *GlobalServer* whenever data of some type and resolution is needed for the area, and taking the appropriate actions upon notification by *GlobalServer* that the request has been serviced. When data need to be updated, *GlobalClient* allocates space for the data within a shared cache and sends a message via a shared memory priority queue to the server. Then *GlobalClient* create 3D global model using DEM, 3D models and imagery acquired from the *GlobalServer*.In the virtual scene, users can view the places they interest, overlap vector map on it, query information and do some analysis. Figure 3 shows the overall architecture of the application.

The server

GlobalServer's data organization layer provides a series of multithread COM components: *DataOrganize*, *DataTranslate*, *Dataompress and DataTransfer*.

- *DataOrganize*: To accommodate data paging, level of detail management, and view culling, a pyramid data structure [5] is used to spatially subdivide and organize the terrain raster data. The globe is subdivided into a small number of pre-determined areas, each corresponding to a separate level. Yet the hierarchy must be flexible so that detail can be added or deleted as needed. Such flexibility is quite important due to database size, as the global datasets used with the application often require ten or more gigabytes.
- *DataTranslate*: The source data may come in a variety of file formats, which *DataTranslate* must translate to a single common format. In addition, for each data type, the source

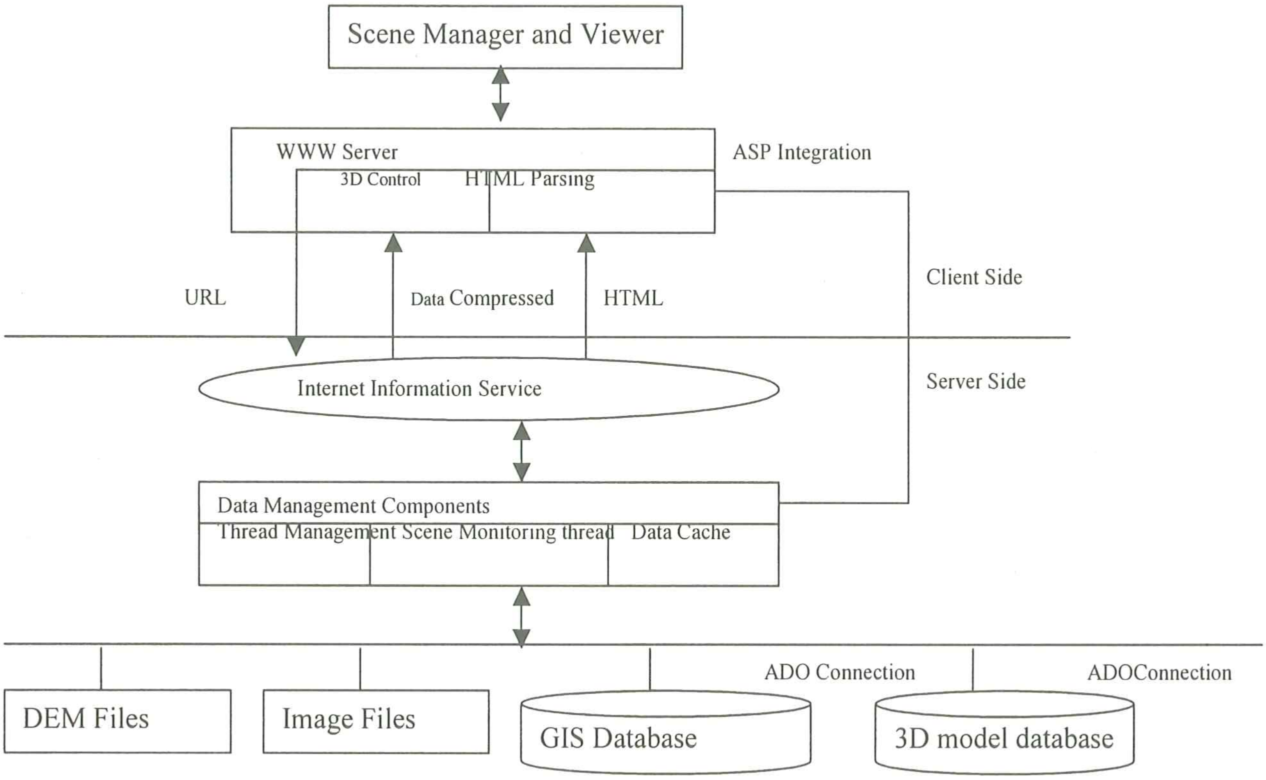

Figure 3. Web application architecture of our system

data may consist of multiple, variable resolution, possibly overlapping/nested datasets. *DataTranslate* has the ability to layer these and composite them into a single dataset if so desired.

- *Dataompress*:I t compresses the proper DEM file into CAB format, airborne and satellite imagery into ECW format.
- *DataTransfer*: *GlobalServer* starts a new thread when it receives a notification, then searches the appropriate dataset using the index file and updates the scene cache.

The Client

GlobalClient performs program execution after the executables downloaded to the client as a COM control. *GlobalClient* also includes another COM control, named *GlobalGuid*. The users may manipulate *GlobalGuid* to view whichever area on the earth and whichever level of the data sets. The client applies new dynamic loading framework (namely, displaying and loading happened at the same time) to visualize the global scene, parallel loading make the users invisible to the pause caused by data loading. After *GlobalServer* submits the record sets,to the cache and starts a transferring thread, the client receives data and then begins to deal with them ,in the meanwhile, the thread receiving data shifts to in the back. After the client finishes the needed functions, it dellocates all the occupancy resources. When data need to be updated, *GlobalClient* allocates space for the data in a shared cache and sends a message via the shared memory priority queue to *GlobalServer*. Message priorities in this queue are changed dynamically according to the importance of the associated request as determined by the level of detail manager [1]. Thus, requests that gradually become less important, or even obsolete, shift towards the end of the queue and get serviced only when no higher priority requests remain in the queue.

V. TEST AND RESULTS

We have used VC++6.0,OpenGL and ATL template library to build an application system *GlobalViewer* that runs on Internet. It fuses vectors, images, and GPS coordinates into a single, seamless dataset to create the ultimate geographic reference platform. The result is a whole new way of looking at your world. *GlobalViewer* delivers the unprecedented ability to fluidly interact with a global geospatial database. It is a powerful analysis and presentation system. It enquires different levels' data according to the distance between the earth surfaces and the eye point. It is downloaded from the server to the client, and *GlobalViewer* runs on the client. The system provides the functions, such as GPS navigation, the flying devices flying along the designed routes, Rotating along one point on the earth surface, changing pitching angle freely, operating the landscape using the mouse and the keyboard, querying the coordinates and attribute query. Through the mouse, the keyboard and the navigation sphere, the users can browser any place of the world; therefore one can get what he can see.

We have tested the usability of 3D visualization. The 3D system runs on an 800 MHZ PIII PC capable of fast 3D graphics card with a broadband (128 kbps or greater) Net connection. DEM data consist of global terrain data *JGP95E 5´*edited by the Defense Mapping Agency and NASA/GSFC,their spatial resolution is 5´,and GTOPO30 terrain data edited by U.S. Geological Survey's EROS Data Center (EDC), their resolution is 30?. Image data consist of global satellite image, Modis image of the whole China which resolution is 250m,TM data of Beijing Granting reservoir. Figure 4 - Figure7 is the graphic interfaces on the browser of *GlobalViewer* (n:the number of the rendering triangles per frame,f:the number of the frame per second,v:the velocity of data translation per second,p:the maximum time of delaying(s))

VI. FUTURE WORK AND CONCLUSIONS

The application system is an ongoing project with more upcoming contents and features. The 3D models and database will be expanded. There are many technical problems needed to be solved, such as how to integrate with virtual scene better, and making the application's GIS functions stronger. In the same time, we should put the database into the different servers and ensure the loading balance of the servers, due to the massive data of the virtual system. In the future work, the application system will be designed for scalable distribution and visualization of web-mapping applications. Combining satellite and aerial images with a variety of other datasets from relational databases, served by a MapServer, *GlobalViewer* distributes rich mapping applications across the Internet or an organization's Intranet.

The main ideas developed in this paper have depicted new methods for building a 3D virtual scene for the earth. A major goal in our project is to realize the framework of a client/server architecture for complex, dynamic, distributed 3D scenes. In contrast to the approach [9], which relies on the high-bandwidth MBone concept [10], we aim at environments with very limited network resources. We have also presented a wavelet-based multiresolution algorithm for simplified massive data given view-dependent error metrics. The method which users view the scene from any angle has also been discussed.

The advantages and applications of visualizing the real-time real world with 3D models are numerous. It provides an intuitive and user-friendly way to view your world - reach out and "grab" the earth with your mouse, pan quickly to your area of interest (or select it from a placemark) and then zoom smoothly down to an individual building. Watch as the city unfolds

Figure 4. The initial interface of global viewer, f=36.8FPS, n=60155, v=7. 5, p=1. 9

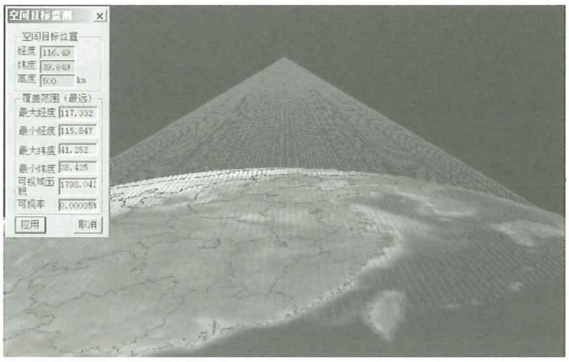

Figure 5. Analysis of visual field

Figure 6. 3D visualization of a part of the earth, f=26.8FPS, n=50125, v=10, p=2. 8

Figure 7. Visualizing east of China using DEM data, f=28. 5FPS, n=58364, v=9, p=2

itself before your eyes, revealing new perspectives and fresh insights. A 3D representation of a real place makes it easier to perceive, e.g., proportions, distances,and landforms, and to recognize landmarks. The 3D objects and environments are highly interactive and allow unrestricted movement.

ACKNOWLEDGEMENTS

This research was funded by the knowledge innovation project of the Chinese Academy of Sciences and Program of Data Fusion for Flood Analysis and Decision Support (ANFAS).The source codes of the application system mostly have been developed in National Engineering Research Center for Geoinformatics of China.

REFERENCES

[1] Cordier, F., H. Seo and N. Magnenat, 2003, Thalmann Made-to-Measure Technologies for an Online Clothing Store. *IEEE Computer Graphics and Applications*, 23(1): 38-48.

[2] Jürgen, D., Klaus, H., 2000. An object-oriented approach for integrating 3D visualization systems and GIS. *Compueters & Geosciences* 26: 67-76.

[3] http://www.earthviewer.com/company/index.html.

[4] Zhang, yongsheng, 2000, *Remote Sensing Image Information System*, Science Publishing House.

[5] Macedonia, M. R., Zyda, M. J., Pratt, D. R.,Brutzman, D. P. and Barham, P. T., 1995, Exploiting reality with multicast groups: a network architecture for Large-scale virtual environments. *IEEE Computer Graphics and Applications*, 15(5): 38-45.

[6] Daniel Cohen-Or and Yishay Levanoni, 1996, Temporal continuity of levels of detail in delaunay triangulated terrain. In Proc. of 7th IEEE Visualization '96 Conference ,October 27 - November 01, San Francisco, California, pp.37-41.

[7] Miller, Mark C., 1996, Multiscale Compression of Digital Terrain Data to meet Real Time Rendering Rate Constraints.PhD thesis, University of California, Davis, 1995~96, pages 37–42. IEEE Comput. Soc. Press, 1996.

[8] Lindstrom, Peter, David Koller, William Ribarsky, Larry F. Hodges, Nick Faust, and Gregory A.Turner, 1996, Real-Time Generation of Continuous Levels of Detail For Height Fields. In *Proceedings of ACM SIGGRAPH 96*, August 1996, pp. 109-118.

[9] Duchaineau, Mark, M. W. Llnl, D. E. S. Lanl, LANL, M. C. M. Lanl, C. A. Llnl, M. B. M.-W. Lanl Lanl, 1996, ROAMing Terrain:Real-time Optimally Adapting Meshes.www.llnl.gov/graphics/ROAM/.

[10] OpenGL. http://www.opengl.org

[11] Rakkolainen, Ismo, Teija Vainio, 2001, A 3D City Info for mobile users. *Computers &Graphics,* 25:619-625.

[12] EarthViewer Software.http://www.earthviewer.com

[13] Zhang, Liqiang, Chongjun Yang, Donglin Liu, 2002, Study On 3D Terrain Visualization On Internet.In *Proceeding of ISPRS Commision II Symposium.Integrated Systems For Spatial Data Production,Custodian And Decision Support*. XI'an, P. R. China, pp.589~594.

[14] Wan, Gang, Changqing Zhu, 1999, Application of multi-band wavelet on simplifying DEM with lose of feature information. *Journal of Surveying and Mapping*, 28(1): 36-40.

[15] zhou, zhongmo , Jiejun Yi, Qi Zhou, 1992, Principle and Application of GPS Satellite Surveying, Serveying and Mapping Publishing House. Dec., 1992.

[16] Digital Elevation Data. http://www.vterrain.org/Elevation/index.html.

[17] MSDN. http://msdn.microsoft.com/.