INTEGRATED DYNAMIC MODEL FOR MULTI-RESOLUTION DATA IN THREE-DIMENSIONAL GIS

WENZHONG SHI[a] , BISHENG YANG[a,b] , QINGQUAN LI[b]

[a]Advanced Research Center for Spatial Information Technology, Department of Land Surveying and Geo-informatics,The Hong Kong Polytechinc University, Hung Hom, Kowloon, Hong Kong. lswzshi@polyu.edu.hk
[b]Research and Development Center for Spatial Information and Network Communication Wuhan Unviversity, Wuhan, Hubei, China 430079. bsyang@263.net

Working Group IV/1

KEY WORDS: Multi-Resolution, Integrated Model, Level-of-Detail, 3D GIS

ABSTRACT:

With the development of digital earth and cyber city, there is an increasing equipment on management and vitalization of integrated 3D model data and image data. How to improve the speed and visualization effect of three-dimesional (3D) model is a major research issue in 3D GIS. In order to improve the speed of 3D visualization, such as flying/walking through, many algorithms have been developed to reduce the data volume of 3D model, for instance, Level_of_Detail (Lod) algorithm. However, image data is also another important factor effecting the operation speed and reality of 3D model. Image data as the texture of surface of 3D model can enhance the reality of 3D model. For the reason that different graphic cards may have different restriction in the size of texture, the texture has to be managed very carefully. This paper proposes a new method to manage image data when a image is used as the texture for a 3D GIS environment. First, the data structure of multi-resolution texture model, which is developed to manage the large texture, is presented. Secondly, the algorithm that creates the multi-resolution texture model is developed, which is of vital importance for the texture mapping. Thirdly, the method of determining a proper texture resolution to map texture on the surface of 3D model based on viewer conditions is discussed. Finally, several experiments are illustrated.

1. INTRODUCTION

Image that is used as texture in a 3D GIS environment is of vital importance, the procedure that map image on the surface of model is texture mapping. Texture mapping is an efficient method to enhance the visualization effect of 3D model scene without adding the geometrical data of 3D vector model (Gahegan, 1999). Usually, the hardware of computer has a certain limitation for a certain texture size. For example, the OPENGL rendering engine limits the maximum texture size, which does not exceed 2048X2048. In 3D GIS system, the field of scene is usually very large, for instance, the terrain surface model can be of 4096X4096 grid size, and the terrain surface can be mapped with texture of 8000X8000 pixels, which exceeds the limitation of graphics hardware. In order to map the texture on the surface of 3D model, many techniques such as Mip-mapping have been developed to map the texture on the surface. It usually generates a serial of textures from the original texture, which called Mip-map pyramid (Williams, 1983), different layer of pyramid has different resolution. But Mip-map usually could not acquire better visualization effect when the texture size exceeds the limited size of graphics hardware, because the large texture has to be formatted in order to meet the requirements, many information of the texture is lost. In order to overcome the drawback of the limit of texture size, many graphics rendering systems such as (SGI□IRIS Performer 2.1,1999) developed Clip-Map method to overcome the drawbacks, Clip-map uses clip-size to clip the amount of texture data. By the clip-size, it can decide which layer texture data to be loaded from pyramid layers (Tanner, 1998). So when mapping texture on the model, only part of texture data is loaded into memory compared with Mip-map method. The difference between Clip-map and Mip-map is illustrated as figure. 1.

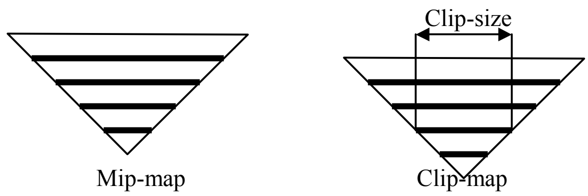

Figure 1. The difference between Mip-map and clip-map

In 3D GIS environment, a 3D city model for example, the terrain surface is large, and the texture of the terrain also exceeds the limit size of texture, moreover, there are so many buildings textures in 3D city model, So how to acquire better visualization effect when mapped the texture on the 3D model is a problem because of the limitation of graphics hardware and system memory. Because the scope of 3D scene is relatively large, only a fraction of the whole scene can be seen when flying/walking through the 3D scene. Moreover, it is evident that different parts of the 3D scene occupy different scope on the screen. For a terrain, supposing the terrain is divided into many tiles, the farther tiles occupy less area on the screen, for building, the farther buildings also occupy less area on the screen. Supposing the same resolution texture is used to map the texture on the different tiles and buildings, it is possible that the resolution of texture is lower for the near tiles of terrain and buildings and higher for the farther tiles of terrain and buildings. Therefore the visualization effect is worse. Supposing the higher resolution texture is used for the near tiles of terrain and building, and the lower resolution texture is used for the farther tiles of terrain and buildings. The visualization effect is thus better. It is evident that the different tiles, which are different distance to the viewer, are with different resolution of texture data. Thus we called the texture multi-resolution texture model. we proposed in this paper, that (1) multi-resolution texture model is developed to manage the large texture, which is useful for large texture mapping, (2) the method to decide the proper resolution of texture in different tiles based on viewer conditions. The frame of multi-resolution texture model is figure 1.

The rest of the paper is arranged as the following. Section 2 presented the data structure of multi-resolution texture model and the algorithm of creating multi-resolution texture, Section 3 propose the method of determining the resolution of texture, Section 4 illustrates several experimental results.

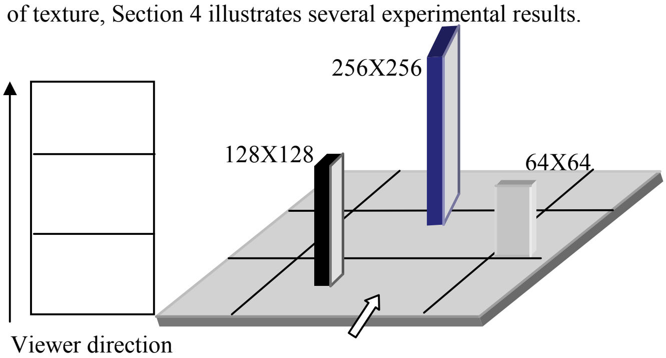

Figure 1. The multi-resolution texture model based on the viewing conditions

2. MULTI-RESOLUTION TEXTURE DATA MODEL

Based on the introduction of section 1, in order to obtain a better visualization effect and map arbitrary size texture on the surface of 3D model, the original texture has to be divided into many small tiles, moreover, so as to overcome the limited size of texture because of graphics hardware, the tile size does not exceed the limited size of graphic card. The regulations about how to divide large texture into many tiles are explained as the following.

Supposing I is the original texture, and $I_1, I_2, ..., I_n$ is the sub-area of original texture region. The following conditions are the constraint conditions when divide the original texture data into different small tiles

1. $I = \bigcup_{i=0}^{n} I_i$
2. $I_m \bigcap I_n = \phi, m \neq n$
3. $I_m \subset I, m < n$
4. $I_m \subset \{x1, y1, x2, y2\}$

Where the condition 1 demonstrates that the original texture can be divided into n titles, condition 2 demonstrate that supposing that any two titles do not intersect with each other , condition 3 demonstrates all the small tiles consist of the original texture, the title is an element of, condition 4 illustrates that each title can be a given size spatial region, where $(x1, y1)$ is the left-down corner of the spatial region, and $(x2, y2)$ is the right-up corner of the spatial region. So , based on the above conditions, the original texture can be divided into many titles. Moreover, each title has a certain spatial region, which can be used to decide proper texture mapping. In order to mange the texture data efficiently, viable data structure has to be developed to mange them. This is the foundation of texture mapping and texture data management.

2.1 Data structure of multi-resolution texture model

Based on the above analysis, the texture data includes two types, one is the texture data of the terrain surface, the other is the texture data of building surface. Because of the fact that the texture size of building surface is relatively small, it need not to be further divided. The different resolution layers texture of building surface textures can be generated from original texture. However, the large texture of terrain surface has to be divided into many titles. In order to ensure efficiency of texture mapping, the viable data structure has to be developed to manage the texture data of building surface and terrain surface. Table.1 illustrates the data structure of texture data, which is the foundation of retrieval of texture data and texture mapping.

Table 1. Data structure of multi-resolution texture

$Image_1$	$Image_2$	………..	$Image_n$

type	space region	layers	texture size	texture data
1	x1,y1,x2,y2	0	1024X1024	01010101.
1	x1,y1,x2,y2	1	512X512	01010101.
… …	……..		….	………
0	x1,y1,x2,y2	3	64X64	01010101.

Table. 1 illustrates the data structure of texture data, where 'type' represents the texture type, 0 is building texture data and 1 is terrain texture data. 'Space region' represents the surface scope that the texture data will be mapped on. 'Layers' represents the texture layers of original texture data generates. 'Texture size' represents the width and height of original texture data. 'Texture data' is the data of original texture data. '$Image_i$' represents the texture index of each tile or building surface, which is used to retrieve the texture data of different tiles of terrain and building textures.

2.2 The Algorithm of creating multi-resolution texture data

So as to creating the multi-resolution texture data, the size of each tile has to be decided, once the size of each tile is decided, the multi-resolution texture data can be created from the original texture data. The main step of creating multi-resolution texture data is presented as below.

2.2.1 Steps of creating multi-resolution texture data

The method of dividing of large texture is the foundation of multi-resolution texture mapping. Based on the data structure developed, the following algorithm is developed to divide the large texture into tiles.

Step1 Read the original texture data and judge the size of original texture (width and height of original texture)

Step2 According to the minimum size of tile, calculate the number of tiles,
Step3 Based on the criterion of partition, get the texture data of each title and store them into the data structure,
Step4 Create the pyramid model of each tile,
Step5 Storing the texture data of each title or building surface according to data structure□
Step6 End.

Figure 2 illustrates the results of original texture, which is divided based on the algorithm. After the original texture is divided into many small tiles, the index of each tile can be created according to the spatial region of each tile is mapped. When mapping them on the surface of 3D model, according to the distance and direction with the viewer, the proper resolution can be decided based on the direction and distance with the viewer, the criterion about how to decide the resolution of texture is proposed in Section 3.

Figure 2. The dividing result of original texture

2.2.2 Mapped Texture on the surface model

Texture mapping is a procedure that mapping the color region to geometrical region, which can be represented as $\xi : C \to R$, so each vertex of the 3D model has a texture coordinate. The procedure of texture mapping includes two steps (Hill, 2001)□

- Loading the texture data of spatial region,
- Determine the relationship between color space and geometrical space (X,Y,Z) and (U,V) □so each vertex can has a texture coordinate (u,v) □

Based on the analysis above, in order to map the texture on the surface of 3D model, the following four steps have to be finished, illustrated as figure 3.

- Based on the spatial index of each tile and viewer spatial position, decide the resolution of texture data of each tile,
- Load the texture data of each title into memory,
- Calculate the relationship between texture data and surface region of model,
- Mapping texture on the surface according to the relationship.

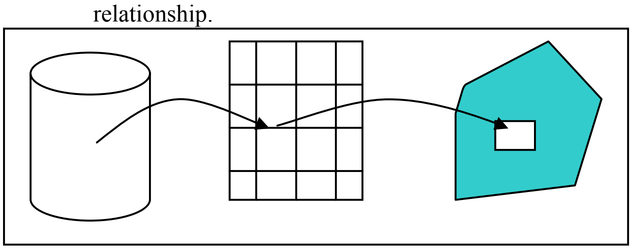

Figure 3. The procedure of mapping texture on the surface of model

For the reason that the limitation of system memory of computer, it is impossible that load all the original texture data of each tile into memory. In order to ensure the efficiency of mapping texture on the surface and the speed of 3D visualization, it is necessary to decide the proper resolution of texture data in each tile and different building surface textures. According to the analysis in Section 1, the tiles of terrain and the surface of building are far away from the viewer, they can be mapped with lower resolution texture data, and the tiles of terrain and the surfaces of building are close to viewer, they can be mapped with higher resolution texture data. According to the facts above, the texture resolution of different tiles and surfaces of buildings are related with viewer conditions, so in order to decide the proper resolution of different tiles, the relationship between texture resolution and viewer conditions has to be decided.

3. DETERMINING THE RESOLUTION OF TEXTURE

According to the theory of perspective projection, a certain length in XoY plane has a certain size on the projection plane. It is evident that the value on XoY plane has relationship with the viewing conditions (viewer direction and distance). The farther that the length is away from the viewer, the smaller that the value on the projection. Supposing the transformation relationship between the length and viewer is invariable, when the length becomes smaller, it's projection value on the projection plane also will become smaller. If the changes of the projection value does not exceed a limited value (for example, the minimum resolution of human eyes), the smaller length can be used to replace the larger length. The same principle also applies to texture mapping. For example, when we mapped texture on a certain surface, different resolution texture data can be generated from the original data. If different resolution texture data of original texture is selected as the texture data to be used as the texture of the same tile or surface, the changes in color space will be generated. According to the principle above, if the changes in color space do not exceed a limited value, the lower resolution texture data can be used as the texture of the surface or the tile. Compared with higher resolution texture data, the lower resolution texture data has less data amount, it needs less memory usage and also costs less time in texture mapping. So it is necessary to explore the relationship between the resolution of texture and viewer conditions (direction and distance). Next, the criterion of determining the resolution of texture data based on viewing conditions will be proposed.

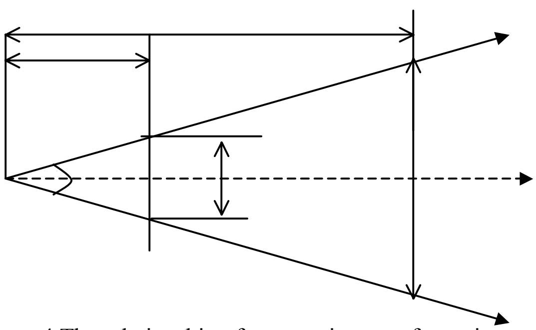

Figure 4 The relationship of perspective transformation

$$\delta / h = f / d$$
$$\delta = f * h / d \quad (1)$$

where f is the distance from the viewer to the projection plane, and d is the distance between the viewer and projection length, h is the projection value on XoY plane.

Figure.4 illustrates the relationship between the size in XoY and the value on the projection plane based on transformation of perspective projection. According to the theory of perspective projection, when the direction of the value in XOY is parallel with projection plane, the value on the projection plane is maximum.

Equ.1 illustrates the relationship between viewer conditions and projection value. If the scale between projection plane and screen is 1:1, the δ is the pixel numbers on the screen. It is evident that when the distance d is farther, the δ will be smaller. According to the scheme of texture mapping, if the same region is mapped with different resolution texture data of original texture, the space that each pixel occupies on the spatial region is different. Based on the mapping relationship between texture data and spatial region, the space that each pixel occupies can be calculated, so the difference between spatial region of each pixel occupies, which is generated by the different resolution texture data, also can be calculated. For example, supposing the size of the original texture data is 256X256, according to the scheme of texture mapping, the following resolution image can be generated as the size of 128X128, 64X64, and so on. If the spatial region of each pixel occupies is ξ_0 when the region is mapped with original texture data (256X256), the spatial region of each pixel occupies is $2\xi_0$ when the same region is mapped with the resolution texture data (128X128). Thus the difference between the spatial region is ξ_0 . According to the Equ.1, the projection value of ξ_0 on the projection plane can be calculated. The calculation result is the changes in the color space, which will have an impact on the 3D visualization effect. So we can think the impact on the visualization is brought by using different resolution texture data on the same region.

Based on the above analysis, the changes, which is generated by using different resolution texture data on the surface, can be calculated by Equ.1. Supposing the changes do not exceed a limited value, the lower resolution texture can be used to replace the higher resolution one. According to the data structure of multi-resolution texture model, each tile or building surface perhaps has several different resolution textures. For each tile, supposing there are $\{\zeta_1, \zeta_2, \bullet\bullet\bullet, \zeta_n\}$ different resolution texture data. And the threshold of limited value is τ , the resolution of current texture data on the tile is ζ_i . Thus so as to decide which resolution of texture data is proper for the spatial region, the following algorithm is developed to decide the proper resolution of texture data for the tile or surface.

Algorithm Decide_Texture_Resolution_Of_Tile( )

Step1 According to the relationship between spatial region and different resolution texture data, the length in space that each pixel of different resolution texture occupy can be calculated. Supposing the length sets of each pixel of different resolution occupies is $\{h_1, h_2, \bullet\bullet\bullet, h_n\}$,

Step2 According to the relationship between texture data and spatial region and the resolution of texture data ζ_i , the length in space that each pixel occupies (h_i) can be calculated, Based on Equ.1, δ_i (the projection value of h_i on the projection plane) can be calculated. $\delta_i = f * hi / d$,

Step3 Based on the threshold of limited value τ , so the optional value (δ) on the projection plane meet the condition. $\delta_i - \tau \le \delta \le \delta_i + \tau$, so length in space that each pixel occupies also can be calculated, the value of the length meet the condition. $(\delta_i - \tau) * d / f \le h \le (\delta_i + \tau) * d / f$,

Step4 Based on the result of **step 3** and length sets**,** the proper resolution can be selected from length sets $\{h_1, h_2, \bullet\bullet\bullet, h_n\}$. The criterion of selection optional resolution texture data is $|h - hi| = \min, (1 \le i \le n)$,

Step5 Based on Step 2,3,and 4, the resolution of texture data of each tile or surface can be decided,

Step6 End.

Based on the algorithm, the resolution of texture data of each tile and surface of building surfaces can be determined. It is evident that the resolution of texture is lower for the farther surface and is higher for the closer surface. The better visualization effect can be acquired based on the scheme of multi-resolution texture model.

4. EXPERIMENTS AND ANALYSIS

Based on the multi-resolution texture model and algorithm we proposed in this paper, the multi-resolution texture model is implemented in our system Spaceinfo and many experiments are finished in SpaceInfo, and many experimental results are as follow. Figure 5 is the result of mapping texture on the terrain surface based on pyramid texture model, and Figure 6 is the result of multi-resolution texture model. The size of the original texture data is 4096X4096 and the tile size is 256X256. Because of the limitation of graphics hardware, the original texture has to formatted as 2048X2048 resolution based on pyramid texture model.

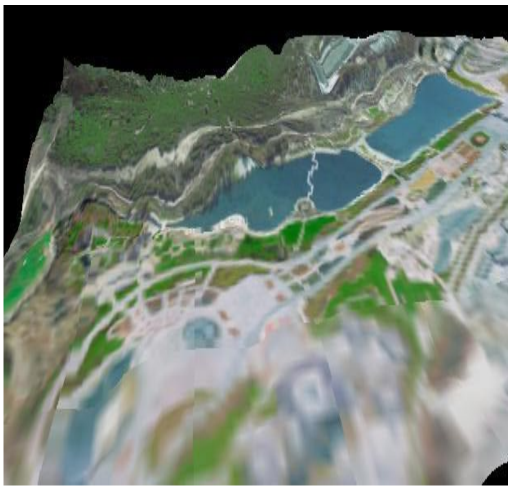

Figure 5. Texture mapping based on pyramid texture model

Figure 6. Texture mapping based on multi-resolution texture model

Based on figure 5 and 6, it is easy to see that the multi-resolution texture model can acquire better visualization effect than that of pyramid texture model. In order to test the performance of different tile size.Table2 lists the time performance of different tile size and pyramid texture model.

Table 2 Time performance of different tile size and standard resolution

Texture Size	512X 512	256X 256	128X 128	64X 64	Pyramid texture
6992X5328	19.09s	18.24s	17.18s	18.56s	19.72s
2048X2048	2.68s	2.78s	2.65s	2.64s	4.16s
1024X1024	0.86s	0.82s	0.81s	0.92s	0.70s

Table 2 lists the time of texture mapping of different tile size and pyramid texture. For the texture size 6992X5328, in order to use pyramid texture model to map texture, it has to be formatted so as to meet the requirements of graphics hard. The time span is from texture loading to map the texture on the surface. According to the result of Table 2, it is easy to see that the tile size is 128X128, the best time performance can be acquired. Thus we use the 128X128 tile size as the minimum unit when dividing a large texture.

Pyramid texture model has some advantages such as the simple data management, easy texture mapping and so on, but the size of texture is limited by the graphics hardware. Multi-resolution texture model can map arbitrary size texture, it overcome the limitation of graphics hardware. Table 3 lists the difference between them.

Table.3 Comparison between pyramid texture model and multi-resolution texture model

	Pyramid Model	Multi-resolution model
Data management	Simple	Complicated
Hardware limitation	Has limit	No limit
Texture coordinate	Calculated one time	Each tile is calculated
Edge handle	No extra handling	Extra handling
Visualize Effect	Worse	Better
Programming	Simple	Much difficult

5. CONCLUSIONS

Texture data is of vital importance for a 3D GIS. It can enhance the reality of 3D model without adding geometry data of model. The method of texture mapping and the management of texture data are important for a 3D GIS system. In this paper, we propose multi-resolution texture method to map large texture on the surface of 3D model, such as terrain model. A viable data structure for the management of texture data is presented, and the algorithm is developed to decide the proper resolution texture data. Moreover, the method is implemented in our software SpaceInfo and acquire many better results. Compared with the method of pyramid texture model, multi-resolution texture method has several advantages.
Firstly, it can map arbitrary size texture on the 3D model. Secondly, A new method for management texture data and map texture is proposed. Thirdly, The system memory of this method occupies is less that that of pyramid texture model. Fourthly, the fast speed of texture mapping, the resolution of texture data is related with viewer conditions, so the visualization effect of 3D model is much improved.

Although multi-resolution texture can acquire better visualization effect, how to improve the speed of 3D roaming further needs to be explored in the next step. For the reason that the resolution of texture data is related to viewer conditions, the texture data in the memory has to be exchanged timely when flying/walking through 3D model. Thus a high efficiency exchange schemes has to be developed, which is the next research.

ACKNOWLEDGEMENTS

The work described in this paper was substantially supported by grants from the Research Grants Council of the Hong Kong Special Administrative Region (Project No.: PolyU 5050, 3-ZB40) and The Hong Kong Polytechnic University (1.34.9709), and partially supported by a grant from the China National Natural Science Foundation (No.69833010).

REFERENCES

Blow, J., 1998. Implementing a Texture Caching System. *Game Developer*. http://www.gdmag.com

Gahegan, M., 1999. Four barriers to development of effective exploratory visualization tools for geosciences, *International*

Journal of Geographical Information Science, 13, pp.289-309.

Hill, Francis S, 2001. *Computer graphics: using OpenGL*. Upper Saddle River, N.J.: Prentice Hall.

IRIS Perform 2.1 API references, 1999. http://reality.sgi.com/performer/perf-99-07/0103.html.

Tanner, C., Migdal, C.J., 1998. The Clipmap: A virtual mipmap. In: Computer graphics (*SIGGRAPG'98*).

Williams, L., 1983. Pyramidal parametrics. In: computer graphics (*SIGGRAPG'83*). pp 1-11.