Geometric-Textured Bitree: Transmission of a Multiresolution Terrain Across the Internet

María José Abásolo Francisco Perales
Graphics and Vision Unit. Mathematics and Computer Science Department
University of the Balearic Islands
Cra. de Valldemossa km 7.5 (07001) Palma, Spain
{mjabasolo,paco}@uib.es

Armando De Giusti
Laboratorio de Investigación y Desarrollo Informático (LIDI)
La Plata National University
Calle 50 y 115 - 1er. Piso - (1900) La Plata, Argentina
degiusti@lidi.info.unlp.edu.ar

ABSTRACT

Transmission of large terrain databases across the Internet is still worrying. A large terrain database requires a big number of polygons and textures, and that represents a problem with current bandwidth and resource limitations. Multiresolution models permit progressive transmission, that is the transmission of a simple model followed by successive refinements.

In this work we present a new multiresolution model called Geometric-Textured Bitree. In contrast with most of recent works, it makes possible the progressive transmission of not only geometry but also the texture of a terrain model.

Wavelet Multiresolution Analysis is applied to the selection of geometry points and, texture segmentation. Also a new texture synthesis process based on Wavelet classification is presented.

Keywords

Progressive Transmission, Quadtree Triangulation, Multiresolution Model, Multiresolution Analysis, Wavelets, Texture Classification and Segmentation, Texture synthesis

1. INTRODUCTION

Large-scale terrain models require not only a big number of polygons but also large images for 3D textured representation. According to resource limitations and current bandwidth it represents a problem for visualisation and transmission of these models.

Multiresolution models provide different Level-Of-Details (LODs) of an object. Although the big size of the model, we can have a multiresolution representation to obtain real-time visualisation. An appropriate resolution can be used to display the model depending on viewing parameters, like screen size of the object, distance from the viewpoint and view direction. Also multiresolution models make possible selective refinement and progressive transmission over networks.

In general, previous works in multiresolution models only consider geometry of an object and also attributes like colour. In the work presented in this paper we propose a multiresolution model called *Geometric-Textured Bitree* that considers the geometry and the texture of a terrain.

Section 2 presents the state of the art of different related topics:

- *Wavelet Multiresolution Analysis*
- *Multiresolution Images*
- *Texture Synthesis and Generation*

Section 3 describes the research undertaken:

- *Overall Architecture*: we present a client-server architecture for progressive transmission of terrain models.
- *Geometric Model:* we use a *Wavelet*-based criterion for selecting points to simplify the geometric model. After that we make a *Bitree* triangulation presented in our previous work [2]
- *Model of Textures:* we make texture classification and segmentation based on a *Wavelet* representation of the image data. After that we extract some representative patterns of the segmented textures. Finally we make a *Bitree* triangulation and assign textures to triangles.
- *Progressive Transmission:* we propose an algorithm to progressively transmit the *Geometric-Textured Bitree*.
- *Reconstruction:* the client regenerates the complete model while receiving geometric and textural information. Particularly, textures are reconstructed by means of a synthesis process based on the *Wavelet* analysis of textural patterns received.

Section 4 presents current and future work of our investigation.

2. PREVIOUS WORK

This section presents the state of the art of *Wavelet* multiresolution models of meshes and images.

2.1. Wavelet Multiresolution Analysis

The *Wavelet* transform produces a hierarchical decomposition of functions. Mallat [18] describes a function as a low resolution function plus low to high-resolution detail functions. *Wavelets* provide means of frequency and space analysis.

The *lifting scheme* [8] is an efficient algorithm to make the forward and inverse *Wavelet* transform.

We focused *Wavelets* on:

- *Multiresolution Geometric Models*
- *Multiresolution Images*

2.1.1. Wavelets in Multiresolution Geometric Models

Lounsbery et al [17] subdivide a basic mesh applying *k-disc Wavelets* to functions defined over surfaces with connectivity properties. Bonneau [3] introduces a *Haar Wavelets* decomposition applied to any triangle mesh. Gross [14] proposes to use any *Wavelet* to control the approximation of an adaptive triangulation of height fields. He uses the detail signal of an inverse *Wavelet* as a criterion for vertices removal.

2.1.2. Wavelets in Multiresolution Images

Wavelets can be used for lossy and lossless compression based on the fact that small coefficients can be ignored because are less important. *EZW* (*Embedded Zerotree Wavelet*) [25] produces a progressive coding of an image. Recently has emerged a new standard for *Wavelet*-based image compression, the *JPEG2000*. Progressive transmission and region-of-interest coding are some representative features of it. A description of *JPEG2000* can be found in [6].

2.2. Multiresolution Images

In general, the texture of a terrain model comes from an entire image that is mapped to a geometric model. There are several ways to provide a multiresolution image:

- *Multiresolution Coding*
- *Image Layers*
- *Coloured Triangulation*

2.2.1. Multiresolution Coding

One of the first multiresolution solutions was the *Laplacian Pyramid*. They provide a redundant multiresolution representation and their merit is that they were the inspiration for *Wavelets*. One of the advantages of *Wavelets* is that they can be applied to obtain a non-redundant multiresolution representation.

2.2.2. Image Layers

A multiresolution image can be composed of different images that explicitly define the levels of details. *Iris Performer*, the *Silicon Graphics* standard, organises texture in a pyramidal structure where each level has twice the resolution of the following one [23]. The *clipmapping* technique determines which resolution is to be applied to each region depending on the view-point position. *GeoVRML*, the *VRML Working Group* [13] proposes a pyramidal structure to represent geometry and textural data. Each image in the pyramid is tiled. Each tile relates with four children tiles of the higher resolution level. This feature makes possible the selected refinement of a region.

2.2.3. Coloured Triangulation

This approach makes an adaptive sampling of an image by means of general triangulation. High frequency regions require more samples than low frequency ones. Progressive refinement is possible by adding more samples. Darsa et al [7] make a *Delaunay* triangulation and fill every triangle with a constant colour resulting from the average of dominant colours of the triangle. Hoppe [16] presents *progressive meshes*, an approach to store and transmit large-scale meshes. Assigning a plane colour to each triangle of a meshing square represents images. Certain et al [4] extend Lounsbery work [17] applying multiresolution *Wavelet* analysis to a triangulated surface with connectivity. They capture geometry and colour independently.

2.3. Texture Synthesis and Generation

Texture generation means producing a texture from some basic information of it. If the information is smaller than the texture itself, it brings the advantage of avoiding storage space and transmission time. We find these following approaches in texture generation:

- *Texture Synthesis*
- *Repetition of Texture Patterns*

2.3.1. Texture Synthesis

Image synthesis is to produce a new image from an example image, in such a way that: the new image is different enough from the original but seems generated by the same stochastic process that generates the original one [9].

Simoncelli et al [26] captures random and structured aspects of a texture by means of joint-probabilities of *Wavelets* coefficients. It synthesises a new image from a *Gaussian* noise image by forcing it to satisfy the given probabilities. De Bonet [9] uses the *Laplacian* pyramid of the original texture, and computes the joint occurrence of several features through different resolution levels. It considers that there exist interchangeably regions in low frequency levels. The synthesis is done level by level by means of a uniform sampling between these interchangeably regions.

2.3.2. Repetition of Texture Patterns

Neyret [19] introduces texture generation from a set of triangular samples of the original image. Whatever the geometry, the surface is divided in triangles and every one is assigned a texture pattern with the continuity considerations the user specifies. Praun [21] pastes texture samples repeatedly to a mesh. It considers the paste direction and scale specified by the user.

3. RESEARCH UNDERTAKEN

Previous section shows that most of authors consider only geometry of an object or at most attributes like colour also. We have no knowledge of relevant work that considers both geometry and texture of a multiresolution model.

A terrain model is generally a height field, that is a matrix of points that are distributed regularly on a two-dimensional grid. In the work presented in this paper we propose a multiresolution model called *Geometric-Textured Bitree,* that considers not only geometry but also the texture of a terrain. The main purpose of it is the progressive transmission of the model over Internet, as is explained in our previous work [1].

3.1. Architecture Overview

The general client-server architecture for the progressive transmission of maps by Internet is as figure 1 shows.

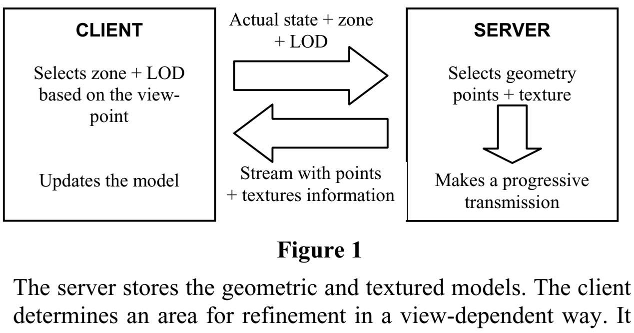


Figure 1

The server stores the geometric and textured models. The client determines an area for refinement in a view-dependent way. It makes a request to the server sending information about the required refinement and the actual state of its model. In response to a client request, the server selects a particular area of a model in different Levels-Of-Details.

In order to attend a client request, the server builds the model according to the selected level of detail. After that it transmit the refinement in a progressive way. In the other side, the client updates the model while it receives the information. The information sent is not only geometry but also textures. They can be both sent by the same logical channel or by separate logical ones.

Figure 2 shows a possible implementation using *Java applets* and *servlets*. In the client side an *applet* send the request to the server. In the server side a *Java servlet* attend the client requests and send data according to the progressive transmission algorithm.

The *Java applet* builds a *VRML(Virtual Reality Modelling Language*) model with the geometry and texture information received. *VRML* is a standard to describe 3D interactive worlds used in Internet, Intranets and local systems. The *VRML* model visualisation could be done by an Internet browser with a *VRML plug-in* installed.

The *VRML* model is controlled completely from the *Java applet* by means of *EAI* (*External Authoring Interface*). The a*pplet* can access the nodes of the scene sending and receiving *VRML* events. In this way the *applet* receives geometric and textural data and interprets them to update the real model and the *VRML* representation.

While the user navigates in the *VRML* scene the view-point changes position and the *VRML browser* informs the *applet* with an event. In consequence the *applet* can determine to initiate a new request to the server.

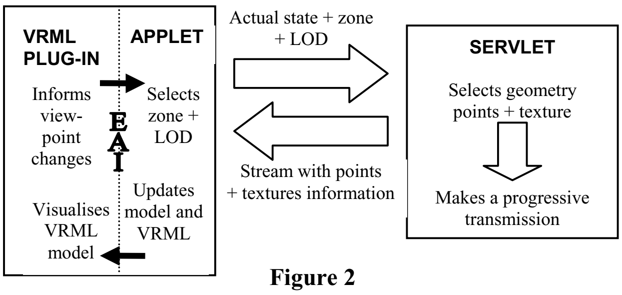


Figure 2

The whole process is shown in plates A-B. Following subsections explain every part of the process: subsection 3.2 describes the geometric model; subsection 3.3. describes the model of textures; subsection 3.4. describes the progressive transmission of the geometric and textural model; and finally subsection 3.5. describes the process of reconstruction in the client side.

3.2. Geometric Model

3.2.1. Polygon Simplification Based on Wavelets

The goal of a simplification surface process is to obtain a reduced model that uses fewer polygons that approximates the original surface in a certain degree. It is desirable that the simplification process be adaptive to the terrain structure because regions with fast geometry changes has to be modelled with more triangles per area unit than a low curvature region.

The points of a triangulation can be selected according:

- a criterion based in an object-space error measure of the approximation;
- a criterion over a parametric representation of the surface.

We follow the last alternative by decomposing the surface by means of a *Wavelet* transform. Points are selected according a criterion based on the *Wavelet* coefficients.

Authors like Lounsbery [17] and Gross [14] decompose the original model by means of a *Wavelet* transform. Gross computes the partial energy of the coefficients in regions surface and if this is low the approximation of the surface can be done with larger triangles and small number of points.

3.2.2. Triangulation

After the selection of points they are triangulated following the *Bitree* triangulation presented in our previous work [2]. *Bitree* is a non-restricted adaptive hierarchical triangulation for height fields. We describe here the main features of *Bitree:*

- Implicit hierarchy of triangles
- Non-restricted selection of points
- *Cracks-free* triangulation with *"fictitious"* points
- Adaptive Level-Of-Details
- Possible Implementation: *Triangle Strips*

Implicit hierarchy of triangles

Bitree divides a triangles into two children and in consequence produces a hierarchy of triangles. *Bitree* subdivision recursively split a triangle by adding the mid-point of its hypotenuse (Figure 3). The initial square block is composed of two triangles adjacent by its common hypotenuse.

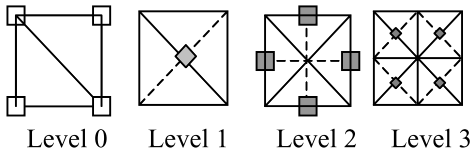


Figure 3

The triangulation is computationally very efficient because it is given implicitly. That means that given a point is possible to deduce the triangle which contain it as the mid-point of its hypotenuse. It is neither necessary to make geometric computations like in-circle tests nor to store triangle connectivity.

Non-restricted selection of points

Generally, authors impose restrictions on the presence of points to produce a *cracks-free* hierarchical triangulation ([20], [14]). In contrast, this approach is non-restricted because it aims the independence between the selection of points criterion and the structure of points in triangulation.

Whatever being the set of selected points *Bitree* triangulation completes the structure to avoid *cracks* by inserting *"fictitious"* points.

Cracks-free triangulation with *"fictitious"* points

The presence of *cracks* is undesirable in any triangulation approach. When rendered from certain viewing angles, even small gaps become very noticeable (figure 4.a). In other works the selection of points is restricted to avoid the presence of cracks. In contrast, *Bitree* avoids cracks by inserting *fictitious* points (figure 4.b).

Fictitious points are not selected according a criterion for triangulation but are required to build a triangulation without cracks between adjacent triangles of different levels. There is an implicit dependency between the points in the triangulation. A *fictitious* point is inserted when one of the needed vertices according to the dependency rules is not selected.

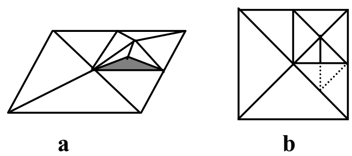


Figure 4

A *fictitious* point has interpolated height between the hypotenuse extremes. That means that its height is deducible from these vertices. In consequence there is no need of storing or transmission them. Besides, this approach produces less overhead triangles than other similar models (see [2] for cost comparisons).

Adaptive Level-Of-Details

Bitree permits adaptive Level-Of-Details through the terrain model. That means that is possible to make a local simplification or selective refinement having different LODs in different adjacent regions.

The LODs are not generated a priori but dynamically according to the specific request.

Possible Implementation: *Triangle Strips*

Triangle strips are a construction for efficiently supporting triangles by hardware rendering engines. They take less space than the implementation with separate *triangles.* Instead of using three points to define each triangle, a *triangle strip* builds triangles from an ordered list of points (*a,b,c,d,e,..*). by grouping every three successive triangles (*abc, bcd, cde,...).*

In this work we can systematically derive a *triangle strip* from the *Bitree* structure. For more details about this process see our previous work [2].

3.3. Model of Textures

3.3.1. Wavelet Multiresoution Analysis of an Image of Textures

A *Wavelet* transform decomposes a function in different Levels-Of-Detail. It provides frequency and space analysis.

Before transmission, textures are processed without considering the geometric model. In this work multiresolution *Wavelet* analysis of textures is done for:

- *Macro-level texture classification:* we analyse an image that is to be mapped to the geometric model. This image is supposed to be a satellite image composed of different natural textures. We segment this image according to an algorithm for texture classification based on characteristics from the *Wavelet* analysis. After that we locate and extract small regions with the same texture called *patterns*.

- *Micro-level texture classification:* in the phase of transmission, we transmit *pattern* texture images based on its multiresolution representation. Besides that in the client side a multiresolution analysis of every pattern is made for texture decomposition and synthesis.

Lifting scheme is a recent tool for *Wavelets* construction [8] [24]. In comparison with traditional methods *lifting* presents several advantages:

- All the operations are in the space domain
- Efficient implementation
- Calculation in place
- The inverse transform results from inverting operations of the forward one
- Operations in every step can be made in parallel

In every *lifting* step there are obtained 4 images or bands of half resolution. The LH, HL and HH bands represent the detail to be added to a low frequency LL band (L/H stands for low/high frequency filtering respectively; the order of the letters means that the filtering operations are applied to rows/columns respectively).

Figure 5 shows the decomposition into two *lifting* steps of a satellite image from Mallorca Island.

The steps to follow are:

1. Applying the *Lifting scheme* to every channel of the image (R,G,B) with a depth of N levels or steps.
2. Analysing the Level-Band-Channel hierarchical decomposition to obtain different characteristics from the *Lifting* coefficients.
3. Classifying every pixel of the analysed image (see subsection 3.3.2)

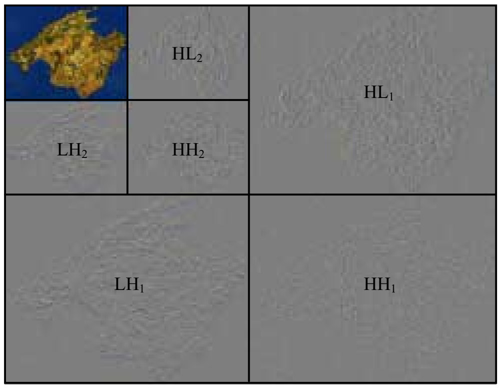


Figure 5

After the decomposition is possible to extract different characteristics from the coefficients:

- Plane *lifting* coefficients.
- First-order statistics of the histogram in the vicinity of a coefficient.
- Edge detection characteristics: gradient, Laplacian, sum of gradients of the channels, number of borders in the vicinity of a pixel, envelope detection, etc.
- Filters: Law, Chebyschev, Sobel, etc.
- Second-order statistics of lifting coefficients: same level-band-channel and different spatial positions, same position-different level-band-channel.
- First-order statistics of one of the previous characteristics.

We have the following bands to be analysed:

- LL bands: all the successive low bands obtained in every lifting step
- *Lifting* pyramid: the LH, HL and HH bands of every lifting step plus the lowest resolution LL band.
- All the generated bands: LL bands plus *lifting* pyramid.

The channels of a band can be analysed separately or its characteristics can be averaged.

3.3.2. Segmentation

After the phase of analysis and extraction of characteristics, the image is segmented. For doing that a classification of every pixel based on the extracted characteristics is done.

Authors like Chang [5] and Gross [15] use *Wavelet* coefficients as a characteristic for texture classification and segmentation. Fatemi [12] evaluates different segmentation algorithms based on *Wavelets*.

We have several alternatives of doing the classification of the original image:

- Direct classification of the original image based on the characteristics of all the channels of all the bands.

- Each band can be classified separately based on the characteristics of its channels. After that the original image is classified based on the classification of its bands.
- Each level can be classified based on the characteristics of the channels of its bands, or on the classification of it if it was done. After that the original image is classified based on the classification of its levels.

The last two options are what we call *progressive classification.* In this case, because of the high correlation between the coefficients of the corresponding bands in consecutive levels, also the characteristics or classification of the previous level can be considered.

Figure 6 shows a satellite image from Mallorca Island. Figure 7 shows the segmentation of this image after applying the classification algorithm *ISODATA (Iterative Self-Organising Data Analysis Techniques).* In this example, first each LL band was classified based on just the *lifting* coefficients of its channels and the classification of the immediate lower resolution level.

3.3.3. Pattern extraction

The pattern extraction phase follows the segmentation of the original image in its different textures. A *texture pattern* is a region of a uniform texture with some minimal size required.

Figure 9 shows the region in the original image (figure 6) corresponding to the texture referenced with colour shown in figure 10.a. Figures 10 b-d show three patterns of this texture obtained in different positions of the original image.

If it is not possible to find patterns of the minimum size required, the corresponding class is eliminated and its pixels are converted to the nearest class.

3.3.4. Triangulation

After the pattern extraction phase, the segmented image is triangulated with *Bitree* triangulation. As in 3.2.2 the triangulation of the image starts with two initial triangles that are recursively divided. If a triangle is considered to be uniformly textured within certain grade of tolerance, then the subdivision stops. Finally, every final triangle has a unique texture. Figure 11 illustrates the process of triangulation of an original image (figure 11.a) and the resulting texture-to-triangle mapping (figure 11.b).

Figure 8 shows the segmented image of figure 7 after triangulation with a tolerance of 95%. That means that a triangle is considered to have a unique texture if the rate between the amount of pixels of this texture within the triangle and the total amount of pixels of the triangle exceeds 0.95.

It should be noted that the difference between the images in figure 7 and 8 it is noticeable mostly in the border of the island. Figure 12 shows a detail of the original map (a) and the *Bitree* triangulated one (b).

Figure 6

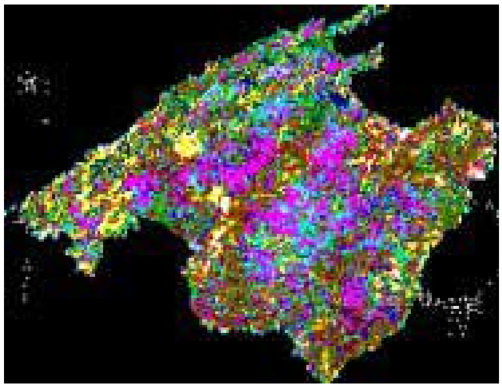


Figure 7

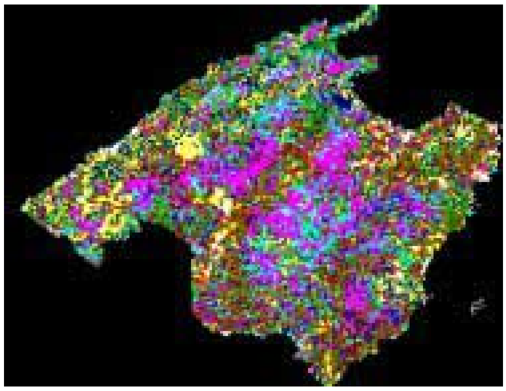


Figure 8

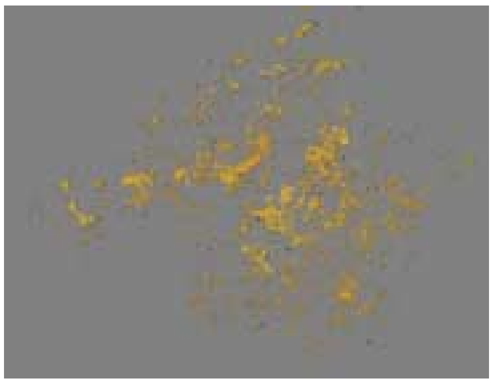
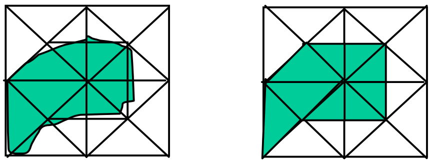


Figure 9

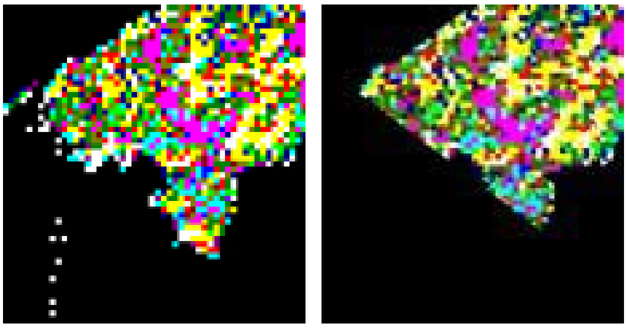
a **b** **c** **d**

Figure 10



a b

Figure 11



a b

Figure 12

3.4. Progressive Transmission

Progressive transmission of a model is the transmission of a basic approximation followed by successive refinements to obtain the desired approximation.

To quote just few relevant recent works in progressive transmission: Hoppe [16] introduces *progressive meshes*; Taubin [28] defines *progressive forest*, the algorithm used by the *MPEG-4* standard for transmitting progressive Level-Of-Detail of *VRML* meshes; De Floriani [11] describes a progressive transmission algorithm for *TINs*. Some recent works propose alternatives for progressive transmission over Internet: Certain [4] describes an application for transmitting coloured meshes over Internet and visualises it in an *HTML browser*; Suliman [27] describes an architecture for progressive transmission over Internet of simulation data; Reddy [22] presents *Terra Vision*, an application for navigating *VRML* terrain models in an *HTML browser*; De Floriani [10] presents a client-server architecture for view-dependent progressive transmission of triangulated meshes.

After both triangulation processes described in previous sections, geometrical and textural ones, we obtain a hierarchy of triangles with a texture reference and a hierarchy of geometric triangles. For transmission purposes, both models could be treated separately or if convenient could be synchronised.

We explain how to transmit the models independently in the two following sections. After that, we describe how to transmit both models in a synchronised way with short modifications.

3.4.1. Progressive Transmission of Geometric Information

The optimal transmission of a mesh or geometric model is to transmit just the geometric points. In general, a transmission algorithm adds a lot of *overhead* for transmitting mesh connectivity information. In contrast, our model try to exploits the simple connectivity of *Bitree* model to reduce overhead information.

The geometric model of this work is a height field. That is z values in Z axe corresponding to a regular grid (x,y) in the XY space. Since the hierarchical *Bitree* triangulation is implicit, it is possible to transmit only z values of the points that divides the triangle hypotenuse. It is possible to deduce (x,y) position on the grid from the order of reception of the information.

The algorithm to transmit the geometric model has the following characteristics:

- It transmits only z value of points ordered by level;
- Client deduces (x,y) position based on the order of reception of the point;
- Whatever the *Wavelet* analysis, Z-value information correspond to the original mesh;
- It transmits information to know that a point is not selected and in consequence its z value is not sent;
- *Fictitious* points are not transmitted, because clients deduce them according the reconstructed *Bitree*.

3.4.2. Progressive Transmission of Texture Information

Instead of transmitting the entire texture image that is to be mapped to a geometric model, we make a segmentation of the image based on its separated textures.

The progressive transmission of the image of texture is composed of:

- progressive transmission of texture patterns;
- progressive transmission of texture-to-triangle mapping.

Progressive transmission of texture patterns

The main characteristic this algorithm tries to exploit is that whatever being the area occupied by a texture in the original image only a fix size texture pattern is transmitted.

The *Wavelets* multiresolution representation of the pattern image can be progressively transmitted by any known algorithm like *EZW* or the recent standard *JPEG2000.*

Progressive transmission of texture-to-triangle mapping

The transmission of the texture-to-triangle mapping is based on the *Bitree* triangulation of the segmented image of texture. Both server and client assume the order of the triangles to be transmitted. For every triangle the following information is sent:

[(T)<texture pattern> | (R) <reference> | (N)] (STOP|CONTINUE)

where

(T): denotes that this triangle has a texture with the pattern that follows

(R): denotes that this triangle has a texture that was transmitted before and is identified with the reference that follows

(N): denotes that this triangle has the texture of its parent triangle

After that the server sends the bit information:

(STOP) the triangle is a final one;

(CONTINUE) the triangle is to be refined.

3.4.3. Progressive Transmission of Synchronised Geometry/Texture Information

The transmission of geometry/texture information can be synchronised. Since every geometric triangle have one or several texture triangles both geometric and texture triangulations can be combined.

A geometric point is the middle point of the hypotenuse of one or two triangles. That is why a point can be logically linked to the triangles it generates.

After transmitting the information of a geometric point the texture information of the triangles involved is transmitted.

Two cases are pointed out:

- When division of a geometric triangle finishes, only the texture information is sent;
- When division of texture triangle finishes, only the geometric information is sent.

3.5. Reconstruction by client

3.5.1. Reconstruction of Geometric Information

The client makes a progressive refinement of the model while receiving information of it. In every step the client waits geometric information, a z-value about a specific (x,y) grid point. It could happen that the server indicates that this point is not selected. In this case the client receives and updates this information to continue with the next point.

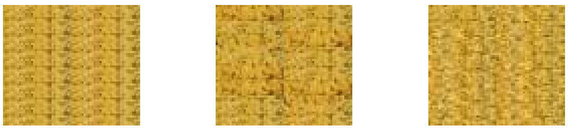
3.5.2. Reconstruction of Texture Information: Texture Synthesis

The client receives texture information of all the triangles in the hierarchy starting from the biggest ones. That is why the reconstructed model has always a texture assigned. The model is to be refined while receiving texture information of the children triangles.

For texture information reconstruction we combine the ideas of repetition of patterns and texture synthesis.

The client receives information about the texture-to-triangle mapping. It only receives a texture pattern image to fill an entire region of a texture.

Figure 13.a shows that the simple repetition of the pattern of figure 10.b adds a repetitive structure not present in the original texture. Figure 13.b shows that it can be improved by repeating the three different patterns of figure 10.b-d in a random way.



a b c

Figure 13

Is easy to see that the bigger number of patterns the more accurate texture generation. Instead of transmitting a lot of texture patterns extracted from the original texture by server, client synthesis new texture patterns from the one or ones received.

In contrast to [21] and [19] the texture repetition process is done without user intervention and no continuity considerations.

Texture pattern synthesis

De Bonet [9] uses the Laplacian pyramid of the original texture. He defines a similarity measure to determine if two regions can be interchanged. He synthesises each level of a new pyramid by sampling uniformly between interchangeably regions of the original one.

Following this idea we use similarities between *Wavelet* coefficient considering the dependence between coefficients of a same locality but adjacent levels. In each level similar coefficients are permuted to generate a different pattern but similar texture appearance.

A classification process like the one explained in section 3.3.2 is applied to every texture pattern. Two pixels of the *lifting* pyramid can be interchanged if they have the same class. We randomly permute pixels of the same class to generate the lifted pyramid of a new image. After that the *inverse lifting scheme* is applied to obtain the new synthesised pattern.

Figure 13.c shows the repetition of different synthesis of the pattern in figure 10.b.

Figure 14.a shows an original pattern and figures 14 b-c show two different synthesis of it. In this example, a *lifting scheme* with 4 steps is applied to the original image and after that a classification process follows. For doing each of the synthesis, we make 50 permutations of coefficients with the same classification in every band. The channels of a band are permuted in the same way to avoid changes in final colours. Finally, the lifting scheme is inverted.

a b c

Figure 14

A complete synthesis of Mallorca Island is shown in figure 15.

Figure 15

4. RESULTS AND CONCLUSIONS

Transmission of simple texture patterns followed by synthesis process allow the compression of the texture model. Bitree implicit hierarchy reduces overhead information of the geometric model in comparison with other models. Compression rates above 99% are obtained in final models with good visual quality. An acceptable low resolution approximation is transmitted with 1% of triangles and a bit rate of 0.03 bpp. A model with 75% triangles needs 1.82 bpp and don´t show differences with the original one.

The results obtained encourage us to believe that the progressive transmission of GTB bring a good balance between visual quality and transmission costs. At the moment we are evaluating progressive transmission with different terrain databases.

5. ACKNOWLEDGEMENTS

We acknowledge the *Laboratory of GIS* (University of the Balearic Islands) for supplying us with the geometric and textural data of Mallorca Island.

6. REFERENCES

[1] M.J.Abásolo, A.De Giusti, J. Blat. Transmisión progresiva por Internet de modelos topográficos texturados. *CACIC 2000,* Ushuaia, Tierra del Fuego, Argentina, October 2000.

[2] M.J.Abásolo, A.De Giusti, J. Blat. A Nonrestricted Triangulation Hierarchy for Multiresolution Terrain Model. *SCCG 99. Spring Conference on Computer Graphics and its Applications.* Budmerice, Eslovaquia, May 1999.

[3] G-P.Bonneau. Multiresolution Analysis on Irregular Surface Meshes**.** *IEEE Transactions on Visualization and Computer Graphics*, vol.4, nº4, Oct-Dec 1998

[4] A. Certain, J. Popovic, T. DeRose, T. Duchamp, D. Salesin, W. Stuetzle. Interactive Multiresolution Surface Viewing. *Computer Graphics. Proceedings of SIGGRAPH'96*, August 1996

[5] T. Chang, C.-C. Jay Kuo. Texture Analysis and Classification with Tree-Structured Wavelet Transform. *IEEE Transactions on Image Processing,* vol.2, nº4, October 1993

[6] C.Christopoulos, A.Skodras, T.Ebrahimi. The JPEG2000 Still Image Coding System: An Overview. *IEEE Transactions on Consumer Electronics,* vol.46, nº4, November 2000

[7] L. Darsa, B. Costa. Multiresolution Respresentation and Reconstruction of Adaptively Sampled Images. *IX SIBGRAPI*, October 1996.

[8] Ingrid Daubechies; Wim Sweldens. Factoring Wavelt Transforms into Lifting Steps. Priceton University, september 1996.http://cm.bell-labs.com/who/wim/papers/papers.html

[9] J.S. De Bonet. Multiresolution Sampling Procedure for Analysis and Synthesis of Texture Images.*Computer Graphics.Proceedings of SIGGRAPH'97,*1997.

[10] L. De Floriani, P. Magillo, F. Morando, E. Puppo. Dynamic view-dependent multiresolution on a client-server architecture. *Computer-Aided Design* (in press)

[11] L.De Floriani, P. Magillo, E. Puppo. Compressing Triangular Irregular Networks. *Geoinformática*, nro.4, vol.1, 2000

[12] N. Fatemi-Ghomi, P. Palmer, M. Petrou. Performance Evaluation of Texture Segmentation Algorithms based on Wavelets. *Workshop on Performance Characteristics of Vision Algorithms*, Cambridge, Abril 1996

[13] GeoVRML Working Group.

http://www.ai.sri.com/geovrml/

[14] M. Gross, O. Staadt, R. Gatti. Efficient Triangular Surface Approximation using Wavelets and Quadtree Data Structure. *IEEE Transactions on Visualization and Computer Graphics*, vol.2, nº2, June 1996

[15] M. Gross. Integrated Volume Rendering and Data Analysis in Wavelet Space. G.M.Nielsen:*Scientific Visualization*, IEEE Computer Society, ISBN 0-8186-7777-5, p.149-177, 1997

[16] H.Hoppe. Progressive Meshes. *Computer Graphics. Proceedings of SIGGRAPH ´96*, pages 99-108, 1996.

[17] M. Lounsbery, T. DeRose, J. Warren. Multiresolution Analysis for Surfaces of Arbitrary Topological Type. Computer Science and Engineering, University of Washington, Technical Report TR 93-10-05, October 1993

[18] S. Mallat. A theory for multiresolution signal descomposition: The Wavelet descomposition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.11, nº7, pp.674-693, 1989

[19] F.Neyret, M-P.Cani. Pattern-BasedTexturing Revisited. *Computer Graphics.Proceedings of SIGGRAPH'99*, 1999.

[20] R.Pajarola. Large scale Terrain Visualisation using the Restricted Quadtree Triangulation. Internal report.292 *Institute of Theorical Computer Science. ETH Swiss Federal Institute of Technology.* Zürich, Switzerland, 1998.

[21] E.Praun, A.Finkelstein, H. Hoppe. Lapped Textures. *Computer Graphics.Proceedings of SIGGRAPH 2000,* 2000.

[22] M. Reddy, Y.Leclerc, L.Iverson, N.Bletter. TerraVision II: Visualizing Massive Terrain Databases in VRML. *IEEE Computer Graphics and Applications*, March 1999.

[23] J. Rohlf, J.Helman. IRIS Performer: A High Performance Multiprocessing Toolkit for Real-Time 3D Graphics. *Computer Graphics. Proceedings of SIGGRAPH'94*, 1994.

[24] Peter Schröder. Wim Sweldens. Building Your Own Wavelets At Home: First Generation Wavelets. Second Generation Wavelets. *SIGGRAPH'96* Course Notes, curso nº13: Wavelets in Computer Graphics, 1996.

[25] J.M. Shapiro. Embedded Image Coding Using Zerotrees Of Wavelet Coefficients. *IEEE Transactions on Signal Processing*, Vol. 41, No. 12 (1993), p. 3445-3462.

[26] E. Simoncelli, J. Portilla. Texture Charaterization via Joint Statistics of Wavelet Coefficient Magnitudes. *Fifth International Conference on Image Processing*, Vol I, Chicago, IL; 4 -7 October 1998.

[27] S.H. Suliman, P.J. Metzger. A Tool for Visualizing SEDRIS Databases Across the WWW.

http://www.sedris.org

[28] Taubin, G. Gueziec, A. Horn, W. Lazarus. Progressive Forest Split Compression. *Computer Graphics. Proceedings of SIGGRAPH´98*, 1998.