

3D Terrain Synthesis from SRTM data and Spatial Watermarking

Aditya Garg
JIIT Noida (India)
III-B-23, Lajpat Nagar-3,
New Delhi-110024

Ruchir Joshi
JIIT Noida (India)
C-55, Vasant Vihar,
Saharanpur-247001, UP

Vikas Saxena
JIIT Noida (India)
JIIT, A-10, Sector 82
Noida-201307, UP

ABSTRACT

This paper presents a methodology for synthesizing and visualizing 3D terrain model from SRTM data. The synthesized terrain model is color coded for different height levels of the corresponding region. The height information of different geographical region is obtained and terrain synthesized by segmenting the massive data file containing grid of height values. The paper also present a watermarking technique through which any string information can be embedded in the vertices of the model and successfully retrieved. This methodology provides disruption of the information content if it is subjected to attacks.

General Terms

SRTM (Shuttle Radar Topology Mission), Mesh segmentation, Terrain Visualization, Watermarking, Digital Elevation Model.

1. INTRODUCTION

Terrain Visualization of a particular area refers to the process of modeling and viewing the 3D surface from any orientation. This allows the users to have a real time view of different bare earth features Terrain Visualization of geographical area has become an important aspect in the area of Information accessing. There are many areas whose survey can't be done through existing land survey techniques [1]. With this increasing complexity in physical access, remote sensing data have proven to be of much help. Through high resolution camera height of any geographical area can be computed through different stereoscopic [2] and shape from shading [3, 4] methods. The end product of these processes is a Digital height map (Digital Elevation Model) which stores the height information in an image format at a particular resolution. With the help of these elevation values a 3d view of the area can be created, thereby providing support to defense agencies, civil planning softwares and acting as assets for Virtual Reality Applications.

However, creating a 3D view of massive Digital Elevation Model is difficult when the size of the data being handled is enormous as computation capability for rendering is limited. Therefore, divide and conquer strategy of breaking the file into

smaller process able units becomes indispensable. The requirement has certain tradeoffs with the viewable area of the terrain and the maximum rendering capacity of the system.

Section III provides insights into various 3D watermarking techniques. In Section IV we have discussed about our Terrain creation methodology from the SRTM data. This encompasses splitting of the data file, plotting the values of on the graphical interface, embedding the watermark and then extracting and validating it. Further in Section V, experimental results are presented which highlight the performance of extracting the embedded string from the terrain. Section VI concludes the findings and outlines the scope of future work to be done in the domain.

2. TERRAIN WATERMARKING

When 3D terrain models are used for defense purposes, then it becomes indispensable to create certain standards for their security, as the data might get corrupted and possibly deliver false information to the end user. In this regard, various 3d watermarking techniques have been proposed which protect the model from attacks such as affine transformation, scaling, cutting, etc. In [5] proposed an algorithm for encrypting information in the surface normals which represent a vector perpendicular to the surface considered, using the eight neighbor approach. The algorithm calculates a mathematical encrypting function that can be used to encrypt and decrypt the data. Point Clustering approach [6] has been used for creating imperceptible watermarks on point sampled surfaces.

3. METHODOLOGY

3.1 Data Splitting

Satellite data of SRTM type are huge and have very large complexity when viewed as a whole. Every single image file is a huge 65 MB (matrix of 6001*6001). Complexity increases when the whole data is plotted to 3D. It is therefore important to first split the whole data into number of smaller similar files (matrices of 300*300 in our case) and then subject them to any further operation.

Splitting the files were made even more complex by the fact that 6001*6001 values were actually represented as a single line of 36012001 without any limiters.

3.2 3D Plotting

The 300*300 height values stored in text files are plotted in 3D as single points and then each is joined to its neighboring 8 pixel height values in order to give a mesh model of the selected terrain area. If better hardware is available, the multiple 3D models of 300*300 sizes can be combined to create a larger view.

To provide better visualization to the models, texture mapping is provided to the mesh model. This texture mapping is dependent on the altitude as well as geographical coordinates of the area being visualized.

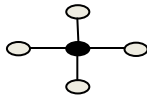


Fig 1. Four neighbor methodology used for grid formation

For each particular sample point connectivity is made to rest of the four neighbors using OpenGL [7] interface.

3.3 Watermarking Embedding & Retrieval

Once the Terrain Model is synthesized, a particular string is embedded in random block of the model using a predefined function.

$$f(v,n)= v \text{ modulo } n$$

where n is the total number of block of size $x*x$ being considered, x being the risk threshold decided for the mesh, v being the number of block. The value of v is usually kept fixed for a particular watermarking session. The string is encoded into bits and each bit is embedded into 1 block of size 30*30.

The storage pattern of the bits depends upon the relative height of points chosen in the selected block.

If $val_{[x,y]} > val_{[x+\delta x, y+\delta y]}$
 Bit 1 is stored successfully
 Else
 Bit 0 is stored successfully.

Using the reverse mechanism used in embedding technique, we can extract the text message embedded in the model using 4*4 height values to extract a single character. Two neighboring height values are used to find the value of every embedded bit.

3.4 Watermark Validation

In the proposed watermark verification scheme, we transfer significant material information from the Terrain model into a projection file. The whole terrain matrix is divided into a number of sites corresponding to each vertex. The entire terrain matrix is triangulated whereby the whole mesh is broken down into unit triangles and each triangle assigned an index. For each site definition, we calculate the mean of the vertices (the height value V_m). The feature value for that particular site having center vertex V_i is identified by using the function.

$$f(a) = \sum_{i=1}^N \frac{V_m - V_i}{N}$$

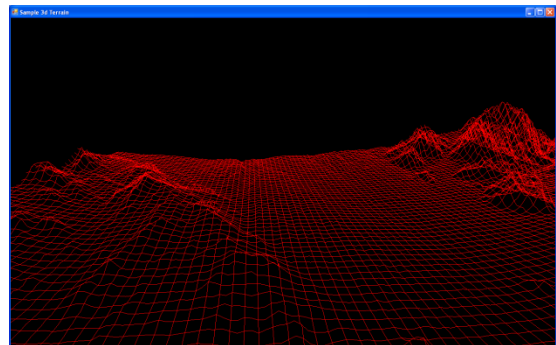
For the complete string, we take the first and last four bits of individual character alternatively and cumulatively form a result called strval. The computed secret value for that site is given by:

$$key(V_i) = (f(a) * strval) \% N$$

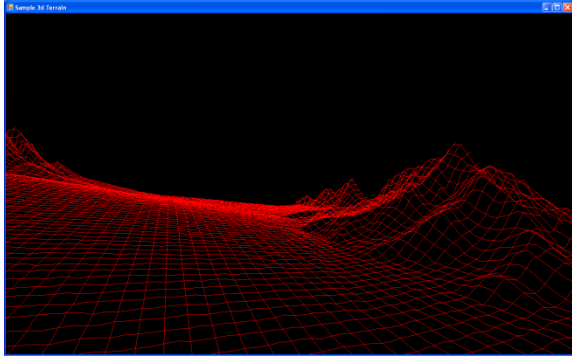
Where N is the number of neighbors of vertex V_i . This value is written down to a file keyval.txt for the entire 300*300 block. At the receiving end, this file is verified using the information about the string that is encoded. Any changes in the model like perturbation, scaling etc. would result in a different projection file, thus notifying about attack.

4. EXPERIMENTAL RESULTS

As a requirement to be able to view a location in 3D, we have successfully made a multiple parts of mesh model from GeoTiff SRTM image data. Camera has been added for navigation and an all side view of the visual model.



(a)

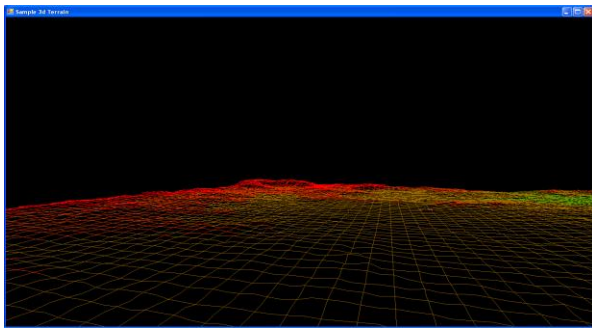


(b)

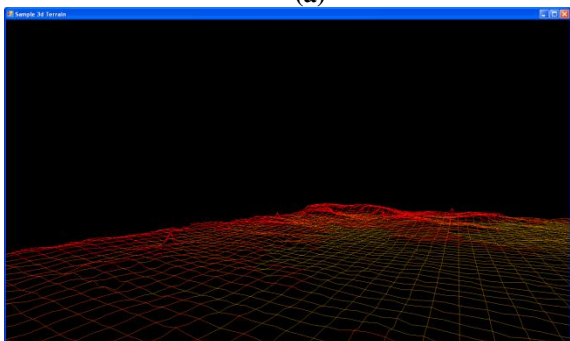
Fig 2 (a) Terrain Synthesized of Chattisgarh Region (India) (b) Karakoram Plateau (China)

A part model takes less than 1 second to paint on a Intel Core 2 Duo system facilitating easy repaint and navigation without jerks. A string data of maximum 99 characters has been successfully embedded into the part model and any variation encompassing more than 10*10 pixels will change the string thus indication tampering of model. Embedding of string and its retrieval takes about 3 seconds at the maximum.

Watermarking the string into the grid space is represented by slight modification in the peaks of the target sites. The selection of the target areas is made at random and depends upon a pre-determined function.



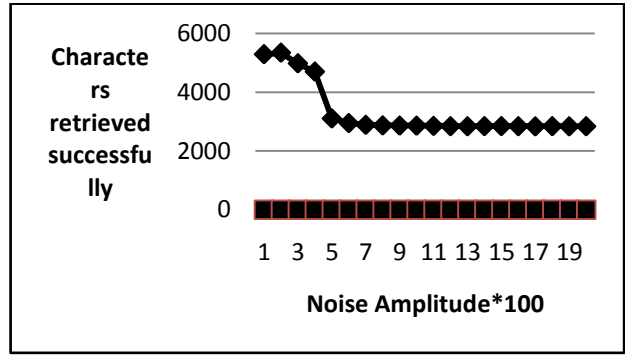
(a)



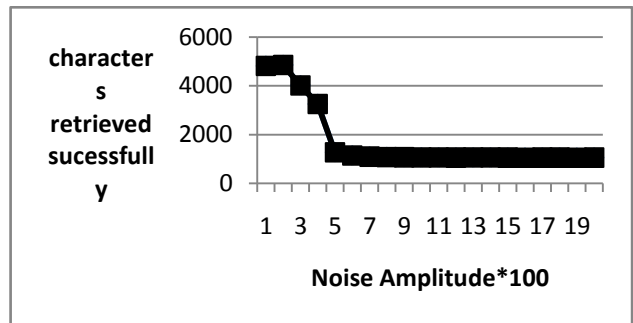
(b)

Fig 3(a) Original Terrain Model (b) Introduction of spikes to represent watermark.

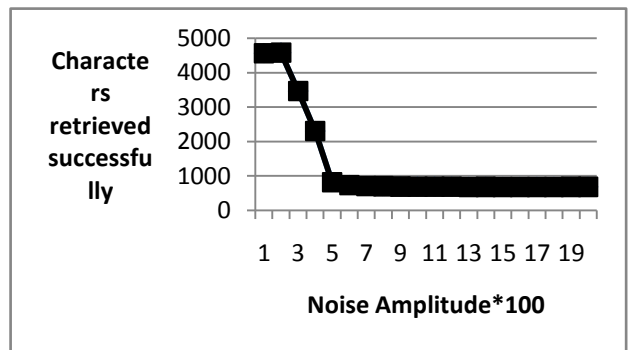
The string containing a maximum of 200 bits was embedded into the Terrain Model using the above mentioned algorithm. For checking the robustness of the embedding criteria, salt and pepper noise was introduced in different proportion of the total plotted area. Using the retrieval methodology, results were obtained for various noise levels corresponding to a single proportion of noise area.



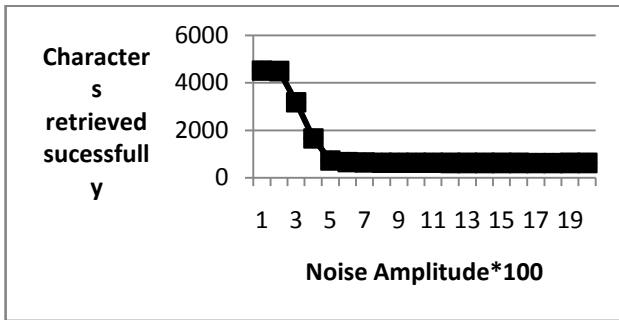
(a)



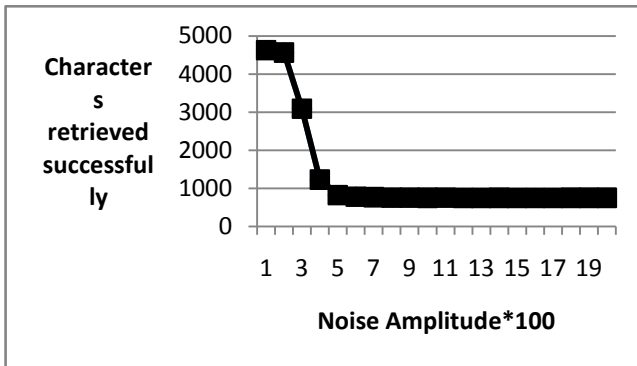
(b)



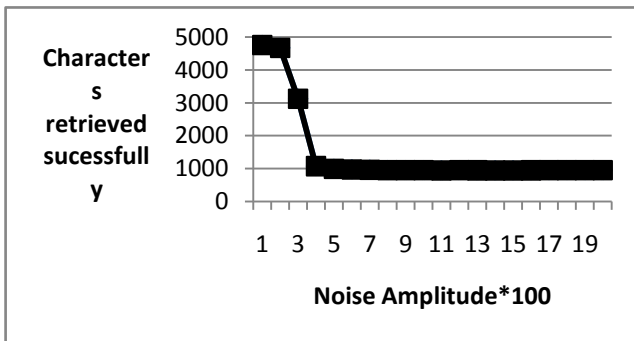
(c)



(d)



(e)



(f)

Fig. 4 Charts showing number of successful retrievals of the data embedded versus induced noise at (a) 10% (b) 30% (c) 50% (d) 70% (e) 90% and (f)100% of the total plotted area.

5. CONCLUSION & FUTURE WORK

We have successfully prepared a 3D terrain model from SRTM height cloud that can be visualized from any orientation. Texturing has been done on the wire mesh model using an existing database of images captured from google maps [6]. A string is embedded in the spatial domain of the grid matrix by breaking into bit string. Retrieval is successful until there is a perturbation on a less than 10*10 pixel scale.

The proposed system works for detecting 3D attacks through encrypting in the grid blocks. The idea can be extended to embedding details into the texture using 2D watermarking

techniques. A combination of skeleton features and texture would more accurately reveal the attacks.

6. ACKNOWLEDGEMENT

Our sincere thanks to our Head of Department, Prof Sanjay Goel, who was a constant source of motivation and guidance to us. We also wish to thank Ms. Suma Dawn, without whose valuable inputs and knowledge, this work would have remained incomplete.

7. REFERENCES

- [1] M. Eshraghi, and F. Samavati, "3D watermarking robust to accessible attacks", in Proc. First International Conference on Immersive Telecommunications, ImmersCom '07, October 2007.
- [2] P. Agarwal, and B. Prabhakaran, "Robust Blind Watermarking scheme for point sampled geometry", IEEE Trans. Information Forensics and Security, vol.4, no. 1, pp. 36 – 48, March 2009.
- [3] C. Bazin, J. M. L Bars, and J. Madelaine, "A Blind, Fast and Robust method for geographical data watermarking", in Proc. Second ACM symposium on Information, computer and communications security, pp. 265-272, March 2007.
- [4] K. Liu and L. Yang, "Watermarking of 3D polygonal meshes based on feature points", in Proc. Second IEEE Conference on Industrial Electronics and Applications, pp. 1837 – 1841, May 2007.
- [5] J. Bennour, and J. L. Dugelay, "Towards a 3D watermarking benchmark", in 9th Workshop on Multimedia Signal Processing 2007, pp. 369 – 372, Oct. 2007.
- [6] M. Corsini, E. D. Gelasca, and T.Ebrahimi, "Watermarked 3D-mesh quality assessment", IEEE Trans. Multimedia., vol. 9, no. 2, pp. 247 – 256, Feb. 2007.
- [7] Y. Maret, and T. Ebrahimi, "Data hiding on 3D polygonal meshes", in Workshop on Multimedia and security, MM&Sec '04, Magdeburg, Germany, pp. 68-74, 2004.
- [8] R. Ohbuchi, H. Masuda, M. Aono and M. Tokyo, "Watermarking Three dimensional Polygonal Models through Geometric and Topological Modifications", IEEE J. Selected Areas in Communication, vol. 16, no. 4, pp. 551 – 560, May. 1998.
- [9] V. Krishnamurthy, and M. Levoy, "Fitting smooth surfaces to dense polygonal meshes", in Proc. 23rd annual conference on Computer graphics and interactive techniques, pp. 313-324, 1996.
- [10] J. Baumann, and K. Hinrichs, "Texturing Techniques for Terrain Visualization", in Proc. Visualization 2000, pp. 227-234, 13-13 Oct, 2000.