Image Compression Terrain Simplification

Boaz Ben-Moshe* Liad Serruya† Ariel Shamir‡

Abstract

Surface simplification is an important application in geographic information systems. The goal is to obtain a new surface that is combinatorially as simple as possible, while maintaining a prescribed degree of similarity with the original input surface.

In this paper, we propose a new terrain simplification algorithm which is based on known Digital Image Processing compression methods (e.g. DCT, wavelets compression) that was specially adjusted to fit Digital Elevation Models. DEM-images are terrains or elevation maps represented as gray scale images. We investigate the special nature of such terrain-images and design a unique pre-compression process which defines the parameters to guide the image compression. We perform a large-scale experiment comparing several terrain simplification methods and conclude that the new suggested algorithm (named $ICTS$[1]) leads to significantly better compression results.

1 Introduction

Terrain models are commonly used to represent the surface of the earth or planets, as well as virtual worlds in games. The compression of such models is fundamental for a number of applications including storage, transmission, and real-time visualization in navigation systems. The storage and transmission of high-resolution elevation information can consume considerable amounts of resources. The increased interest in GIS application, in particular, mapping the earth surface and real-time map representation, emphasize the need to develop efficient compression techniques for elevation maps.

Lossless compression methods [12, 14]often leads to compression ratio which is not high enough. Therefore in cases where some loss of information is allowed, and drastic compression ratio is needed (e.g. video, images audio), lossy compression is used (e.g. $JPEG$ [8] for natural images). Lossy compression methods are mostly suitable when the distribution of elevation values is relatively flat. Elevation maps and terrains are similar in this sense to most natural images. In such scenes, high gradient values or discontinuities are rare and most content changes gradually across the domain. Examining the spatial frequency domain of elevation maps shows that the lower spatial frequency components contains more energy than the high frequency components, which often correspond to details and noises.

In this paper we assume the original surface is a grid based terrain (e.g. DEM, DTED). We choose to represent such terrain as a $2D$ image where the pixel values are the height. We call this representation *elevation image*. Using elevation images enables the use of known digital image processing algorithms for compression (we use the term *terrain image* for a compressed elevation image). This work demonstrate utilization of image compression for terrain simplification and representation. The suggested method ($ICTS$) is based on existing digital image processing method that was specially 'tuned' to fit the task of simplifying and representing terrain-images. We suggest a pre-compression stage which consider the terrain geometric properties. And show how $ICTS$ can support geometric point location queries. We compare $ICTS$ to the state-of-the-art image compression $JPEG2000$ and to standard terrain simplification algorithms and conclude that the suggested $ICTS$ method leads to better simplification results.

2 Related work

There are numerous papers dealing with terrain and surface simplification. Heckbert and Garland [7] surveyed general methods for simplifying 3D models (terrains are special case of 3D models). There has also been extensive work on many aspects specific to terrain simplification; most papers address terrain simplification using triangulation based representation and algorithms optimized to consider error norms such as maximum vertical distance and Hausdorff distance [3, 6, 15]. More recently several attempts to improve the terrain simplification accuracy and runtime results were made using semi-local triangulation [10].

On the other hand, terrains can be naturally represented by a $2D$ grid of heights (DEM). Such DEM terrain can be represented as a grayscale image, where each elevation sample is translated to a grayscale 'pixel' value. Few papers looked at such terrain-images from the digital image processing aspect; Franklin and Said

*Department of Computer Science, College of Judea and Samaria, Ariel 44837, Israel, benmo@yosh.ac.il

†Department of Computer Science, Interdisciplinary Center, Herzliya 46150, Israel

‡Department of Computer Science, Interdisciplinary Center, Herzliya 46150, Israel

[1]Patent pending IL182367

[2] showed that the *progcode* image processing algorithm yields to efficient compression results over terrain images with respect to RMS error norm. Rane and Sapiro [14] investigated terrain-images lossless compression using the standard $JPEG - LS$ [8]. Yea et al. [16] presented a detailed wavelets based compression for terrain-images which support elevation query mechanism allowing de-compressing only the parts of the terrain within an elevation range. Recently Owen and Grigg [13] demonstrated the use of $JPEG2000$ for compressing and querying DEMs. Gortler and Hoppe [5] even define geometry images as a representation for closed 3D surface meshes.

In this paper we compare between these two main approaches; (i) Computational Geometry: triangulation based terrain simplification and (ii) Digital Image Processing: terrain-image compression.

3 New Terrain Simplification Method (ICTS)

The general Frame-work of the $ICTS$ algorithm includes the following stages: **Preprocessing Stage:** converting the grid based terrain into a gray scale image; this step involves translating the elevation data (positive / negative values) into the gray scale value range. **Presetting Stage:** setting the compression parameters according to the geometric nature of the terrain. This stage is the main contribution of our algorithm, it allows us to compress a terrain image according to its geometric properties (e.g. water-flow), while standard compressions only consider it as a $2D$ natural image. **Compression stage:** perform the actual DIP compression. For most terrain-images the DCT compression is used. **Output testing stage:** The simplified terrain is compared to the original input. In case the output does not satisfy the user limitation (e.g. the error rate is too big) the compression parameters will be updated and a new compression will be computed. This stage is optional but might be needed in order to guarantee that the simplified terrain satisfies the user limitation.

3.1 The Parameters of Terrain Image Compression

Based on the well-known corollary that the values of the different parameters have a major effect on any DIP compression quality, we investigate the nature of the parameters which take part in images-compression general framework, and demonstrate the role each parameter-value takes in the compression process. The following hi-level parameters should be fixed in order to perform a DCT compression: **Block Size**: usually $8 * 8$ pixels, but may also be of a general rectangle dimension (e.g. $8 * 16$ $16 * 16$ $32 * 32$). **Quantization Table**: takes an important roll in the trade-off between resolution and compressed-image size. **File Size**: limits the output terrain size, implies the compression ratio.

In order to tune the compression parameters to fit the specific nature terrains the pre-compression stage includes the following steps:
Input settings: the original terrain and its meta-data regarding both the input and the output type, size and other constrains. For instance, the user can ask to compress the terrain to a certain size.Alternatively, the user can ask to simplify the terrain to the minimal size given an upper error rate bound. The user can also suggest the type of the terrain.
Compute simple local parameters: In this part several local parameters are computed including: The min/max height (the extreme values of the terrain). The average height difference between a pixel and consecutive pixels (usually 8 neighbors). The Standard deviation of the difference between a pixel and consecutive pixels. This stage is highly efficient in terms of runtime and memory.
Compute global approximation factors: A rough approximation of the water flow and the watershed of the terrain, We followed known algorithms for computing watershed and water flow [1], A statistical representation of the water flow/shed can be computed using the same single 'pass' performed to compute the local parameters. Therefore this step is also implemented efficiently, since only a rough approximation of the water flow/shed is being computed (and not the complete diagrams).
Classify the terrain: Using the above parameters we classify the terrain into the following types: Flat or almost flat terrain. Mostly dunes (dunes have unique shape of water flow/shed). Hilly terrain and Smooth Mountains (i.e. old mountains), peaks and cliffs (i.e. new mountains). Natural terrain with artifacts (watershed exist: lakes, buildings). Natural terrain without a water flow (ocean surface terrains, stars surface terrains). Artificial terrains (i.e. cities, gaming...)
Set compression parameters: using the above computed information, we turn to set the compression parameters, including the block size, the proper quantization table, and other parameters.

4 Geometric Queries

In this section we discuss the usability of compressed terrains as geometric surfaces. Regular compressed images mostly represents general-$2D$ images. In such case there is rarely a need of retrieving the value of a query 'pixel' without fetching all the image. On the other hand, simplified or compressed terrains represent geometric surfaces and thus should support local (point location like) queries. One of the most common queries performed on terrain is *point location.* In such query a $2D$ point $p(x, y)$ is given, and we are interested in computing the z-value (height) of p as implied by the terrain. Kirkpatrick [11] showed how a triangulation can support such query in $O(logn)$ time with only a linear (space) overhead. Other queries may include two points Line Of Site (LOS), and general local-region queries.

Compressed terrain image can support the above geometric quires efficiently and with very little overhead storage. *Region Of Interest (ROI)* is a general image compression technique which support querying desired blocks of pixels, without fetching the entire image. ROI is supported in the $JPEG2000$ standard and had been implemented on other image compression methods [13]. Using ROI approach all the above mentioned geometric queries can be supported efficiently. Moreover compressed terrain images can also support another geometric type of query; Said et al. [16] presented efficient image coding to access a pixel range using DCT compression. This way one can access only parts of the terrain within some elevation range.

5 Experimental Results

In this section we report on some of our experiments with $ICTS$, as well as comparisons with other terrain simplification methods $QSlim$ [3], Terra [6] and $JPEG2000$ [9]. We compared these terrain simplification methods using the following four measures (error norms): MSE (mean squared error), PSNR (peak signal-to-noise ratio), MAE (mean absolute error), RMS (root mean squared).

We used a data set of 17 input terrains representing different and varied geographic regions, including: terrains representing dunes, hills, mountains, craters, ocean surfaces, and more. Each input terrain covers a rectangular area of 10×10 - 100×100 km^2, and consists of $1,000,000$ - $16,000,000$ vertices.

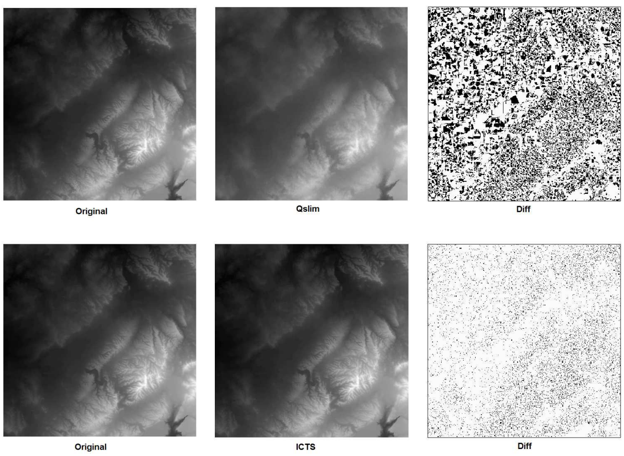

Figure 1: The difference between a triangulation based simplification method and the image compression terrain simplification (using DCT) is illustrated.

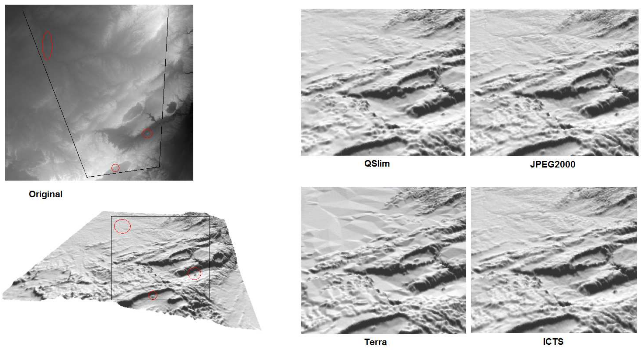

Figure 2: The original 100*100 km terrain was simplified using all four methods ($QSlim, Terra, JPEG2000, ICTS$) all simplified terrains have (more or less) the same maximal vertical error, yet the ICTS terrain has the smallest file size.

Norm	JPEG2000	ICTS
$Filesize(kb)$	198.27	168.78
MSE	2.9756	1.0481
MAE	1.1574	0.4251
RMS	1.6586	1.0525
$PSNR$	52.302	55.950

Table 1: ICTS vs. JPEG2000 Terrain simplification results (error norms), based on average results over all the terrain data set.

The first experiment we performed compared $ICTS$ to $JPEG2000$. We wanted to test if the suggested new terrain simplification method can do better than $JPEG2000$ ($JPEG2000$ is a new image coding system that uses state-of-the-art compression techniques based on wavelet technology).Preliminary results showed that terrains compressed by $ICTS$ have a better error rate and smaller file size than those compressed by $JPEG2000$. To experiment the advantages of $ICTS$ over $JPEG2000$ (with respect to terrain simplification) we forced the file size of the terrains compressed by $ICTS$ to be at lease 10% smaller than the corresponding $JPEG2000$ files, see table 1. AT the second experiment we compered between two types of terrain simplification methods: (i) standard; triangulation based terrain simplification ($QSlim, Terra$) and (ii) $ICTS$ method.

In order to compare the compression quality (error norms) we first simplified the input terrains using $Terra$ and $QSlim$ into five levels of simplification: 10,000, 30,000, 50,000, 100,000, 200,000 (vertices). Then we compute the error norms for each simplified terrain. We then used $ICTS$ to simplify each terrain to files of the same size as the corresponding $Terra$ (or $QSlim$) files[2].

[2] Each triangulation (simplified using $Terra$ or $QSlim$) was farther compressed using standard ZIP compression. Only then the corresponding $ICTS$ terrain was computed.

Norm	QSlim	Terra	ICTS
MSE	54.9915	25.4917	0.4434
MAE	5.2899	3.7039	0.1725
RMS	7.06465	4.9786	0.6638
$PSNR$	39.7121	42.5078	59.9158

Table 2: $ICTS$ vs. *QSlim* and $Terra$ Terrain simplification results (error norms), based on average results over all the terrain data sets for the five levels of simplification.

RMS value	QSlim	Terra	JPEG2000	ICTS
4.0	1082.1	1130.5	33.4	33.8
3.53	1418.6	1444.5	49.8	44.6
3.09	1839.5	1863.8	81.9	58.4
2.8	2113.1	2026.1	97.6	68.4
2.45	2163.2	2732.2	160.1	98.5

Table 3: The same error norm values(RMS) were used to test the file size (kb) of all simplification methods.

Table 2 shows the average error norms over all tested terrains and levels of simplification for the same file size compressed terrains.

Runtime results: For small terrains (elevation maps of 100,000 vertices), $ICTS$ runs on average 4-10 times faster than $Terra$ and *QSlim*. For larger terrains (1,000,000 vertices and more) the runtime gap grows significantly, often reaching a factor of 100.

File size results: For the same error level the files computed $ICTS$ were less than 15% of the corresponding same error ratio terrains simplified using *QSlim* or $Terra$ and farther compressed by standard ZIP.

6 Conclusion

We have tested several existing DIP formats (e.g. $JPEG, JPEG2000$) and shown that these formats often lead to significantly good compression of terrains. As shown in the experiment results above even a standard jpeg compression compresses terrains significantly better (smaller data, better quality, and faster runtime) than specialized terrain simplification methods such as $Terra$. Yet, because the DIP regular formats were originally designed for compressing images (usually natural images) and not terrains, their parameters and other fine details of the implementation can be specialized for terrain-images and therefore improved even further. Simplifying terrains using $ICTS$-like methods, can also support region of interest (ROI) queries, with very little over head. Thus, GIS queries such as line of site (LOS) can be performed on the compressed terrains. Yet another advantage of DIP like terrain simplification methods has to do with the actual implementation platform; DIP algorithms may be implemented on dedicated hardware (such as GPU [4]), therefore implementing $ICTS$ in hardware is feasible and might lead to further runtime-improvement.

References

[1] J. Y. Choi and B. A. Engel. Real-time watershed delineation system using web-gis. *J. Comp. in Civ. Engrg.*, 17:189–196, 2003.

[2] W. Franklin and A. Said. Lossy compression elevation data. In *Proc. 7th Int. Symp. on Spatial Data Handling*, pages 29–41, 1996.

[3] M. Garland and P. S. Heckbert. Surface simplification using quadric error metrics. In *Proc. SIGGRAPH '97*, pages 209–216, 1997.

[4] Graphic Processor Unit. *GPU.* http://www.gpgpu.org/.

[5] X. Gu, S. Gortler, and H. Hoppe. Geometry images. In *Proc. of SIGGRAPH*, pages 355–361, 2002.

[6] P. S. Heckbert and M. Garland. Fast polygonal approximation of terrains and height fields. Report CMU-CS-95-181, Carnegie Mellon University, 1995.

[7] P. S. Heckbert and M. Garland. *Survey of polygonal surface simplification algorithms*, 1997. http://cmex.ihmc.us/data/Tutorial/MDIMs.htm.

[8] Joint Photographic Experts Group - ISO 10918-1. *JPEG*, 1994. http://www.jpeg.org/.

[9] Joint Photographic Experts Group - ISO/IEC 15444-1. *JPEG2000*, 2000. http://www.jpeg.org/jpeg2000/.

[10] Y. Kang, T. Lee, S. Yang, and W. Park. A fast digital terrain simplification algorithm with a partitioning. In *Proc. of HPC*, 2000.

[11] D. Kirkpatrick. Optimal search in planar subdivision. *SIAM J. Comput.*, 12:28–35, 1983.

[12] P. Melnychuck and M. Rabbani. Survey of lossless image coding techniques. In *Proc. SPIE, vol 1075, (Digital Image Processing Applications)*, pages 92–100, 1989.

[13] M. J. Owen and M. W. Grigg. *The Compression of Digital Terrain Elevation Data (DTED) using JPEG2000.* Command and Control Division Information Sciences Laboratory, Australia, 2005.

[14] G. S. S. D, Rane. Evaluation of jpeg-ls, the new lossless and controlled-lossy stillimage compression standard, for compression of high-resolution elevation data. *IEEE Transactions on Geoscience and Remote Sensing*, 39:2298–2306, 2001.

[15] C. T. Silva and J. S. B. Mitchell. Greedy cuts: An advancing front terrain triangulation algorithm. In *Proc. 6th ACM Workshop on Advances in GIS*, pages 137–144, 1998.

[16] S. Yea, A. Said, and W. Pearlman. Efficient image coding for access to pixel ranges. In *ICIP'04 International Conference on Image Processing*, pages 425–428, 2004.