Image Compression
Terrain Simplification

M.Sc. dissertation for applied project

Submitted by Liad Serruya

Supervisor: Dr. Boaz Ben-Moshe\(^1\)

Supervisor: Prof. Ariel Shamir\(^2\)

April 2011

\(^1\)Department of Computer Science, Ariel University Center of Samaria, Ariel 44837, Israel

\(^2\)Department of Computer Science, Interdisciplinary Center, Herzliya 46150, Israel
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, 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 grayscale images. We investigate the special nature of such terrain-images and design a unique pre-compression process which defines the parameters for 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.

\(^1\) Patent pending IL182367
Acknowledgements

First and foremost I offer my sincerest gratitude to my supervisor, Dr Boaz Ben-Moshe, for providing valuable ideas and helpful discussions, for guiding me and supporting me during all stages of the research and the process of writing the thesis.

Last but not least to my husband, Evgeny, who listened and discussed with me ideas related to this thesis, and has always given me warm encouragement, unconditional support and love throughout this long process.


## Contents

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Introduction</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>1.1 Motivation</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>1.2 Outline of the paper</td>
<td>9</td>
</tr>
<tr>
<td>2</td>
<td>Related Works</td>
<td>10</td>
</tr>
<tr>
<td>3</td>
<td>New Terrain Simplification Method (ICTS)</td>
<td>12</td>
</tr>
<tr>
<td></td>
<td>3.1 The ICTS framework</td>
<td>12</td>
</tr>
<tr>
<td></td>
<td>3.2 The Parameters of Terrain Image Compression</td>
<td>13</td>
</tr>
<tr>
<td>4</td>
<td>Geometric Queries</td>
<td>15</td>
</tr>
<tr>
<td>5</td>
<td>Experimental Results</td>
<td>16</td>
</tr>
<tr>
<td></td>
<td>5.1 Compression Quality</td>
<td>16</td>
</tr>
<tr>
<td></td>
<td>5.2 First Experiment: Terra and QSlim vs. ICTS</td>
<td>18</td>
</tr>
<tr>
<td></td>
<td>5.3 Experiment Details</td>
<td>18</td>
</tr>
<tr>
<td></td>
<td>5.4 Summary of Results</td>
<td>18</td>
</tr>
<tr>
<td></td>
<td>5.5 Second Experiment: JPEG2000 vs. ICTS</td>
<td>21</td>
</tr>
<tr>
<td></td>
<td>5.6 Experiment Details</td>
<td>21</td>
</tr>
<tr>
<td></td>
<td>5.7 Summary of Results</td>
<td>21</td>
</tr>
<tr>
<td>6</td>
<td>Conclusions</td>
<td>25</td>
</tr>
<tr>
<td>7</td>
<td>Application</td>
<td>26</td>
</tr>
<tr>
<td></td>
<td>7.1 Installation</td>
<td>26</td>
</tr>
<tr>
<td></td>
<td>7.2 Usage</td>
<td>26</td>
</tr>
<tr>
<td></td>
<td>7.3 JPEG2000 Conversion</td>
<td>26</td>
</tr>
<tr>
<td></td>
<td>7.4 JPEG Conversion</td>
<td>28</td>
</tr>
<tr>
<td></td>
<td>7.5 Image Comparison</td>
<td>29</td>
</tr>
<tr>
<td></td>
<td>7.6 Settings</td>
<td>31</td>
</tr>
<tr>
<td>8</td>
<td>References</td>
<td>32</td>
</tr>
</tbody>
</table>
List of Tables

Table 1: 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. 19
Table 2: The same error norm values (RMS) were used to test the file size (kb) of all simplification methods. For very large values of RMS Terra compresses better than QSlim, but the RMS values ICTS performs much better than the other. 19
Table 3: The same file size (kb) was used to test different error norm values at the two methods. 21
List of Figures

Figure 1: The original elevation map compressed using all four simplification methods, and presented as simplified geometric surfaces................................................................. 18
Figure 2: The difference between a triangulation based simplification method and the image compression terrain simplification (using DCT) is illustrated.......................... 20
Figure 3: The difference between DCT and wavelets (DWT) terrain-images compression is illustrated........................................................................................................... 24
Figure 4: 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........................................... 25
Figure 5: JPEG2000 Tab – Conversion to JPEG2000 file format .................................. 27
Figure 6: ImageMagick Tab – Conversion to JPG file format ........................................ 29
Figure 7: Compare Tab - Image comparison .................................................................... 30
Figure 8: Setting Tab ....................................................................................................... 31
List of Graphs

Graph 1: The same file sizes (kb), compression ratio, were use to test the error norm values (MAE) of the two methods. For very large values of compression ratio the JPEG2000 performs much better than the ICTS. ................................................................. 22
Graph 2: JPEG2000 vs. ICTS Terrain simplification run time result, based on average results over 100 times. ........................................................................................................ 22
Graph 3: JPEG2000 vs. ICTS Terrain simplification file size based on the same error norm in different compression quality. .............................................................................. 23
Chapter 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 applications, in particular, mapping the earth surface and real-time map representation, emphasizes the need to develop efficient compression techniques for elevation maps. Lossless compression methods [18, 21] often lead 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 [12] 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 reflect the height. We call this representation an 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).

1.1 Motivation

We were motivated by several GIS applications dealing with huge data sets of DEMs\(^2\), which should be compressed in the most efficient and accurate way. An example for such application is NASA’s project of mapping Mars (MDIM project [19]). Other key applications which may benefit from the suggested method are navigation tools for mobile devices (e.g., GPS, smart-phone, PDA), as these devices can hardly support

\(^2\) Elevation maps may be stored as DEM (Digital Elevation Model) or DTED (Digital Terrain Elevation Data) - a standard digital dataset, which is a uniform matrix of terrain elevation values providing basic quantitative data for systems and applications that require terrain elevation, slope, and/or surface roughness information.
triangulation and 3D visualization algorithms, due to limited computation power. Yet such devices often have basic image manipulation capabilities (zoom, tilt) over standard image formats (e.g., JPEG), which can be used for implementing the suggested ICTS method on such devices.

Another advantage of using standard image processing algorithms for terrain simplification and representation is improved runtime - DIP algorithms are extremely fast and can support huge data set due to the 'local' nature of the compression. Moreover, these algorithms can be implemented in dedicated hardware (GPU [7]) which can lead to further runtime improvement.

**Our contribution:** This work demonstrates the utilization of image compression for terrain simplification and representation. The suggested method (ICTS) is based on an 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 considers the geometric properties of the terrain, 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.

### 1.2 Outline of the paper

The rest of this paper is organized as follows; in section 2, we review previous work and several basic methods for terrain simplification and terrain image compression. Then, in section 3, we present our new ICTS terrain simplification method. We discuss the importance of setting appropriate parameters in order to convert a general compression algorithm into a practical terrain simplification method. The actual value of these parameters has a significant influence on the quality of the compression. In Section 4 we discuss the usability of compressed image terrains as geometric surfaces. We demonstrate how terrain-image can support local queries such as: point location, line of sight (LOS), or region of interest (ROI) without decompressing the entire image. In section 5 we report large scale experiment with ICTS, as well as other terrain simplification software packages. Finally, section 6 offers a short discussion and some conclusions.
Chapter 2

Related Works

Numerous studies dealing with terrain and surface simplification have been published. Heckbert and Garland [10] surveyed general methods for simplifying 3D models (terrains are special cases 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 [6, 9, 22]. More recently several attempts to improve the terrain simplification accuracy and runtime were made using semi-local triangulation [14, 15], and a hybrid approach [24]. Little and Shi [17] (extending earlier work of Fowler and Little [4]), used linear features, along ridges and channels, to guide their triangulation algorithm with the goal of minimizing the root-mean-square (RMS) error in approximating a DEM with a triangulation.

On the other hand, terrains can be naturally represented by a 2D grid of heights (DEM). Such a DEM terrain can be represented as a grayscale image, where each elevation sample is translated to a grayscale 'pixel' value. A few papers looked at such terrain-images from the digital image processing aspect; Franklin and Said [5] showed that the progcode image processing algorithm yields to efficient compression results over terrain images with respect to RMS error norm. Rane and Sapiro [21] investigated lossless compression of terrain images using the standard JPEG–LS [12]. Yea et al. [25] 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 [20] demonstrated the use of JPEG2000 for compressing and querying DEMs. Gortler and Hoppe [8] even define geometry images as a representation for closed 3D surface meshes.

In this paper we compare between the following two main approaches: (i) Computational Geometry: triangulation based terrain simplification and (ii) Digital Image Processing: terrain image compression. We compare the suggested new ICTS method with the following simplification methods: QSLim [6], Terra [9], GcTin [22] and JPEG2000 [13].

QSLim. QSLim, developed by Garland and Heckbert [6], is an algorithm designed for more general simplification of all types of surfaces and not just terrains. QSLim uses simple edge contraction to perform simplification, while using a quadric error measure for efficiency and for visual fidelity.

Terra. This algorithm, implemented by Garland [9], is based on a simple greedy insertion algorithm with some optimizations to make it run faster. The input is assumed to be a height field given by a regular grid of elevation data. It begins with a trivial triangulation of the domain and then iteratively adds vertices according to which the
input point has the greatest vertical error with respect to the approximating surface. Retriangulation is done using the Delaunay triangulation.

**GcTin.** GcTin, developed by Silva et al. [22], uses an advancing-front technique for simplification of digitized terrain models. The algorithm takes greedy cuts (“bites”) out of a simple closed polygon that bounds a connected component of the yet-to-be triangulated region. The method begins with a large polygon, bounding the whole extent of the terrain to be triangulated, and works its way inward, performing at each step one of three basic operations: ear cutting, greedy biting, or edge splitting. One of the main advantages of GcTin is that it requires very little memory beyond that for the input height array.

**JPEG2000.** JPEG2000, [13], is a new image coding system that uses state-of-the-art compression techniques based on wavelet technology. This image compression format is not yet commonly in use but has a compression advantage over JPEG [12] by roughly 20% on average. The compression gains over JPEG are attributed to the use of DWT and a more sophisticated entropy encoding scheme.

Clearly there exist other image compression methods. Indeed, as shown in this paper, any image compression format can be thought as a terrain simplification method. We used JPEG2000 as a point of reference as the most advanced general image compression format available.
Chapter 3

New Terrain Simplification Method (ICTS)

In this section we present our new Image Compression Terrain Simplification (ICTS) method that is based on existing DIP formats. We first cover the general notion of image compression using spatial frequency domain and investigate the difference between a (natural) image compression method, and a terrain image compression method in terms of error norms. Lossy image 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 contain more energy than the high frequency components, which often correspond to details and noises. However, psychophysical experiments suggest that humans are more sensitive to losses in higher spatial frequency components than to losses in lower frequency components. Indeed, the quality of a general image compression method is often tested according to perception awareness.

Moreover, image compression methods like JPEG and JPEG2000 were originally designed to eliminate 'artifacts' in the compressed images, although, terrains can be efficiently compressed using DIP methods. Different error norms should be used, since terrain images represent geometric surfaces (not just images). One can think of several ways to measure the geometric quality of a simplified terrain. Ben-Moshe et al. [1] suggested a measure of quality based on preserving inter-point visibility. Other geometric measures may test the topological elements such as watershed and the water-flow. Methods for measuring topological and geometric approximation quality of compressed terrains are strongly motivated by GIS applications, yet these measures are not mathematical norms and often depend on implementation parameters. Therefore, in this paper we only report on experiments conducted with the following standard error norms: MSE, MAE, RMS and PSNR.

3.1 The ICTS framework

The general framework of the ICTS algorithm includes the following stages:

- **Preprocessing Stage**: converting the grid-based terrain into a grayscale image; this step involves translating the elevation data (positive / negative values) into the grayscale 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 as 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 are updated and a new compression is computed. This stage is optional but might be needed in order to guarantee that the simplified terrain satisfies the user limitation.

### 3.2 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 general image compression frameworks, and demonstrate the role each parameter-value takes in the compression process. The following high-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**: plays an important role 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 of 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 surrounding pixels (usually 8 neighbors), The standard deviation of the difference between a pixel and surrounding 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 [2]. 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 exists: lakes,
buildings), Natural terrain without a water flow (ocean surface terrains, stars surface terrains) and Artificial terrains (e.g., cities, gaming...).

**Set compression parameters**: using the above computed information, we set the compression parameters, including the block size, the proper quantization table, and others.
Chapter 4

Geometric Queries

In this section we discuss the usability of compressed terrains as geometric surfaces. Regular compressed images mostly represent general 2D images. In such case there is rarely a need of retrieving the value of a query ‘pixel’ without fetching the entire 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 [16] 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 Sight (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 supports querying desired blocks of pixels, without fetching the entire image. ROI is supported in the JPEG2000 standard and has been implemented in other image compression methods [20]. Using ROI approach all the aforementioned geometric queries can be supported efficiently. Moreover, compressed terrain images can also support another geometric type of query; Said et al. [25] 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.
Chapter 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 [6], Terra [9] and JPEG2000 [13]. 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) and RMS (root mean squared).

We have implemented a preliminary version of ICTS using the several open source tools. Our implementation of triangulation was used to evaluate and sample the triangulation based simplifications (i.e. QSlim, Terra). The JJ2000 [3] application was used to manipulate and test various JPEG2000 compression options. The ImageMagick [11] application was used to verify the computed error norms for all simplification algorithms and for JPEG compression. The Kakadu [26] application was used for JPEG2000 compression with the ability to compress according to the desired bit rate. All experiments were performed on the following platform: AMD Athlon™ 64 Processor 3500+, 2.2 GHz, 1.00 GB of RAM, Windows XP Professional. The current version of QSlim (2.1, 2004) was used, while for Terra and GcTin implementation, the original sources [9, 22, and 23] were used.

For the first experiment we used a data set of 17 input terrains representing different and varied geographic regions including: dunes, hills, mountains, craters, ocean surfaces and more. Each input terrain covers a rectangular area of 10 x 10 - 100 x 100 km², and consists of 1,000,000 - 16,000,000 vertices. For the second experiment we used a data set of 3 terrain inputs representing different geographic regions, including urban surface, with each input having different file size (2 MB, 5 MB and 55 MB).

5.1 Compression Quality

In order to compare lossy image compression methods, we add several criteria to quantify the quality of the compressed images for a given compression rate.

We tested the quality of the simplified terrains using the following four measures: MSE (mean squared error), PSNR (peak signal-to-noise ratio), MAE (mean absolute error) and RMS (root mean squared). We tested the compression time and the size difference between the compressed file and the original. We discuss each of these below.
MSE measures the distortion introduced by the compression technique and represents the cumulative squared error between the compressed and the original image. The formula gives a high value to error, since the measure grows proportionally to the square of the error.

\[
\text{MSE} = \frac{1}{MN} \sum_{y=1}^{M} \sum_{x=1}^{N} [I(x, y) - I'(x, y)]^2
\]

PSNR is an engineering term for the ratio between the maximum possible power of a signal and the power of corrupting noise that affects the fidelity of its representation. The PSNR is most commonly used as a measure of quality of reconstruction in image compression. It represents an unbiased measure of the fidelity of the rebuilt image. The larger the PSNR is, the smaller the MSE gets, the better the rebuilt image quality.

\[
\text{PSNR} = 20 \log_{10} \left( \frac{\text{maximum pixel value of the image}}{\sqrt{\text{MSE}}} \right)
\]

MAE is a measure to evaluate the accuracy of simplified models. It gives the average of the absolute values of elevation differences. MAE measure is more robust against isolated errors and infrequent large errors, since MAE weights all errors equally. A large MAE value means that the image is of poor quality.

\[
\text{MAE} = \frac{1}{MN} \sum_{y=1}^{M} \sum_{x=1}^{N} |I(x, y) - I'(x, y)|
\]

RMS error is sensitive to noise and outliers. It is a good measure for the overall fitting quality of the simplified surface and the original terrain.

\[
\text{RMS} = \sqrt{\frac{1}{MN} \sum_{y=1}^{M} \sum_{x=1}^{N} [I(x, y) - I'(x, y)]^2}
\]

Compression Ratio as the ratio between the uncompressed size and the compressed size used to quantify the reduction in data-representation size.

Run Time is measuring the compression time. The value is calculated as the average of 100 compressions of each image.
Figure 1: The original elevation map compressed using all four simplification methods, and presented as simplified geometric surfaces.

5.2 First Experiment: Terra and QSlim vs. ICTS

The first experiment we performed compared between two types of terrain simplification methods: (i) standard triangulation-based terrain simplification (QSlim, Terra) and (ii) ICTS method.

5.3 Experiment Details

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, and 200,000 (vertices). Then we computed 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)\(^3\).

5.4 Summary of Results

Error norms results: Table 1 shows the average error norms over all tested terrains and levels of simplification for the same file size compressed terrains.

---

\(^3\) To make sure the comparison is fair; each triangulation (simplified using Terra or QSlim) was further compressed using standard ZIP compression. Only then the corresponding ICTS terrain was computed.
<table>
<thead>
<tr>
<th>Norm</th>
<th>QSlim</th>
<th>Terra</th>
<th>ICTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSE</td>
<td>54.9915</td>
<td>25.4917</td>
<td>0.4434</td>
</tr>
<tr>
<td>MAE</td>
<td>5.2899</td>
<td>3.7039</td>
<td>0.1725</td>
</tr>
<tr>
<td>RMS</td>
<td>7.06465</td>
<td>4.9786</td>
<td>0.6638</td>
</tr>
<tr>
<td>PSNR</td>
<td>39.7121</td>
<td>42.5078</td>
<td>59.9158</td>
</tr>
</tbody>
</table>

Table 1: 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.

**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. The larger the original terrain is, the greater the gap. We were unable to run QSlim or Terra because of memory limitations, while JPEG2000 and ICTS compress such terrain in seconds (with minor memory overhead).

**File size results:** For the same error level the files computed ICTS were less than 15% of the corresponding terrains simplified using QSlim or Terra and further compressed by standard ZIP⁴.

<table>
<thead>
<tr>
<th>RMS value</th>
<th>QSlim</th>
<th>Terra</th>
<th>ICTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.0</td>
<td>1082.1</td>
<td>1130.5</td>
<td>33.8</td>
</tr>
<tr>
<td>3.53</td>
<td>1418.6</td>
<td>1444.5</td>
<td>44.6</td>
</tr>
<tr>
<td>3.09</td>
<td>1839.5</td>
<td>1863.8</td>
<td>58.4</td>
</tr>
<tr>
<td>2.8</td>
<td>2113.1</td>
<td>2026.1</td>
<td>68.4</td>
</tr>
<tr>
<td>2.45</td>
<td>2163.2</td>
<td>2732.2</td>
<td>98.5</td>
</tr>
</tbody>
</table>

Table 2: The same error norm values (RMS) were used to test the file size (kb) of all simplification methods. For very large values of RMS Terra compresses better than QSlim, but the RMS values ICTS performs much better than the other.

---

⁴ Note: there exists several compression methods designed specially for compressing triangulations [15]. Using these compressions might decrease the size of the compressed triangulation by factor of 1.5-2.5 comparing to standard ZIP compression, yet even with such compression these files will be at least 3 times larger than the terrains compressed by ICTS.
Compression techniques configuration results: The original terrain was compressed using triangulation-based simplification method and the image compression terrain simplification, using DCT. The monochromatic images represent the error rate. A black area represents an error rate larger than some constant (one meter). Observe that while the errors of the ICTS compression seem almost as random points, the QSlim compression errors regions cover a larger area.

Figure 2: The difference between a triangulation-based simplification method and the image compression terrain simplification (using DCT) is illustrated.
5.5 Second Experiment: JPEG2000 vs. ICTS

In the second experiment we compared ICTS to JPEG2000. We wanted to test whether 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).

5.6 Experiment Details

To test the performance of ICTS versus JPEG2000 (with respect to terrain simplification) we forced the file size to be the same. By compressing the terrains at 14 different quality levels, we received the same compression ratio with both methods. Then we ran the compression 100 times to calculate average run time and computed the error norms from a data set of 3 terrains, representing different geographic regions, including urban area, with each input file being of different size.

5.7 Summary of Results

**Error norms results:** Table 3 shows that the norm value of JPEG2000 is somewhat better than ICTS at the same compression ratio. Graph 1 shows the difference between the two methods at different tested norm errors. The error value is very small, but at a higher compression ratio the gap becomes more pronounced.

<table>
<thead>
<tr>
<th>Norm Types</th>
<th>JPEG2000</th>
<th>ICTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSE</td>
<td>10,205</td>
<td>31,687</td>
</tr>
<tr>
<td>MAE</td>
<td>65.5</td>
<td>109</td>
</tr>
<tr>
<td>RMS</td>
<td>101</td>
<td>178</td>
</tr>
<tr>
<td>PSNR</td>
<td>56</td>
<td>51</td>
</tr>
</tbody>
</table>

**Table 3:** The same file size (kb) was used to test different error norm values at the two methods.
Graph 1: The same file sizes (kb), compression ratio, were used to test the error norm values (MAE) of the two methods. For very large values of compression ratio the JPEG2000 performs much better than the ICTS.

Runtime results: To calculate the runtime we compressed the input terrains at different compression ratios in the two methods. Each compression was run 100 times and the average run time was calculated. Graph 2 demonstrates the major difference between JPEG2000 and ICTS: for small terrains ICTS runs on average 5-6 times faster and for larger terrains the gap grows significantly.

Graph 2: JPEG2000 vs. ICTS Terrain simplification run time result, based on average results over 100 times.
**File size results:** After compressing the images at different ratios, error norms were calculated for each image. Graph 3 demonstrates the file size differences between the two compression methods when the MAE norm error values are the same. The gap between the file size gets smaller as the error grows. JPEG2000 has an advantage at small error rates, but that advantage becomes minor at large error rates.

**Graph 3:** JPEG2000 vs. ICTS Terrain simplification file size based on the same error norm in different compression quality.
Compression techniques configuration results: Another big difference between JPEG2000 (wavelet) and ICTS (DCT) is in the error image received by subtracting the compressed image from the original one. In Figure 3, the original terrain (on the left) was simplified using both DCT and DWT. The monochromatic images represent the error rate. A black dot presents an error rate larger than some constant (one meter), while a white dot implies that the approximated elevation value is within one meter distance from the original elevation. Observe that while the error of the DCT compression seems almost as random points, the wavelet-based compressed image divides into large regions of errors.

Figure 3: The difference between DCT and wavelets (DWT) terrain-images compression is illustrated.
Chapter 6

Conclusions

We have tested several existing DIP formats (e.g. JPEG, JPEG2000) and shown that these formats often lead to very good compression of terrains. As shown in the experimental results above, even 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 fine-tuned for terrain images, further improving the performance. Simplifying terrains using ICTS-like methods can also support region of interest (ROI) queries, with very little overhead. Thus, GIS queries such as line of sight (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 [7]), therefore implementing ICTS in hardware is feasible and might lead to further runtime improvement.

Figure 4: 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.
Chapter 7

Application

Image Processor is the application which allows generation and comparison of JPEG and JPEG2000 files. The application interface is designed to simplify experimentation with parameters fine-tuning during image generation.

The application is a user interface wrapper for several open-source graphic processing applications. It makes use of JJ2000 Java applet to generate JPEG2000 images and ImageMagick suite to generate JPEG images and compare the generated images with the original ones.

7.1 Installation

Run imageprocessorsetup.exe to install Image Processor application. The JJ2000 and ImageMagick are required to run Image processor. They will be installed automatically when imageprocessorsetup.exe is run.

After the installation process is complete the Image Processor icon will be put in Windows Start Menu.

7.2 Usage

The application has several tabs: JPEG2000, JPEG, Compare, Settings. The first three tabs are used to work generate and compare images. The Settings tab contains the paths to JJ200 applet and ImageMagick executables. You do not normally need to change the paths unless you wish to use JJ2000 or ImageMagick that were installed at location other than default.

The Log box is at the bottom part of the application window. The Log box displays the output produced during command execution. The log messages are also written into log.txt so that the conversion results can later be reviewed and further processed if needed.

7.3 JPEG2000 Conversion

The application uses PGM, PGX and PPM files as the source for JPEG2000 conversion.

- Activate JPEG2000 tab.
Click Browse button to select the source file.

The input file name will appear in the Input file text box.

The output file name will be generated automatically using the input file name and the counter near the output file name.

The counter is incremented automatically each time you generate a file so that output files are never overwritten. If needed, one can set the initial counter value manually.

All the options in this tab are passed to JJ2000 application via a parameters file which has .p extension. The name of the parameters file is the same as the name of the output file. Thus one can always know what options were used to create a specified file.

![Image Processor 1.2.9](image.png)

**Figure 5:** JPEG2000 Tab – Conversion to JPEG2000 file format

Input File: This field specifies the full path to the input file.

Output File: After the input file is selected the output file name is generated automatically by appending a numeric suffix to the input file name. After each run the suffix is incremented so that the conversion results do not override each other.

Qtype: Specifies which quantization type to use for specified tile-component.
Qstep: This option specifies the base normalized quantization step size (BNSS) for tile-components. It is normalized to a dynamic range of 1 in the image domain. This parameter is ignored in reversible coding. The default value is 1/128.

QGuardbits: The number of bits used for each tile-component in the quantization to avoid overflow.

Filters: Specifies which filters to use for specified tile-component. If this option is not used, the encoder chooses the filters of the tile-components according to their quantization type: If this option is used, a component transformation is applied to the first three components.

Wlev: Specifies the number of wavelet decomposition levels to apply to the image. If 0 no wavelet transform is performed. All components and all tiles have the same number of decomposition levels.

Bitrate: This is the output bitrates of the code stream in bits per pixel. When equal to -1, no image information (beside quantization effects) is discarded during compression.

Lossless: Specifies a lossless compression for the encoder. This options is equivalent to use reversible quantization and 5x3 wavelet filters pair (`-Ffilters w5x3`). Note that this option cannot be used with Bitrate. When this option is off, the quantization type and the filters pair is defined by Qtype and Filters respectively.

Run: This button initiates the conversion process.

7.4 JPEG Conversion

The application can use BMP, PGM, PGX and PPM files as a source for JPEG conversion.

- Activate JPEG tab.
- Click Browse button to select the source file. Select the desired output quality.
- The output file name will be generated automatically using the input file name and the selected quality percentage.
- Click the Run button to perform a conversion.
Figure 6: ImageMagick Tab – Conversion to JPG file format

Input File: This field specifies the full path to the input file.

Output File: After the input file is selected the output file name is generated automatically by appending image quality percents to the input file name.

Quality: Compressed image quality in percents.

Output format: Switches between JPG and JPEG2000 output formats.

Run: This button initiates the conversion process.

7.5 Image Comparison

The application can compare two images. The result of the comparison is a file containing graphical representation of the differences between the files and a numerical result.

- The following options can be used to perform comparison.
  - AE: absolute error count, number of different pixels.
  - MAE: normalized mean absolute error, average channel error distance.
- MSE: mean error squared, average of the channel error squared.
- PAE: normalize peak absolute.
- PSNR: peak signal to noise ratio.
- RMSE: normalized root mean squared.
- To select files for comparison double click the grid at the both Input File row.
- The difference file name is a concatenation of the both input file names.

**Figure 7:** Compare Tab - Image comparison

Comparison type: Allows selecting the type of the measure of the differences between images.

Input file: Double-click the cells to select the two images to be compared.

Diff File: After the input files are selected the output file name is generated automatically by combining the input file names and the comparison type.

Size: Displays the size of the input files.

Result: Displays the measure of the differences between images according to selected comparison type.
7.6 Settings

![Image Processor 1.2.9](image.png)

**Figure 8:** Setting Tab

JJ2000: Specifies the folder in which JJ2000 application is installed.

ImageMagick Installed in Folder: Specifies the folder in which ImageMagick application is installed.

Repeat each run: Specifies how many times each conversion should run when you press the Run button. The purpose of multiple runs is to allow more precise calculation of the time it took to execute one conversion.
Chapter 8

References


