High fidelity compression of irregularly sampled height-fields

Patrick Marais, James Gain

Department of Computer Science, University of Cape Town, Rondebosch 7700, South Africa

ABSTRACT

This paper presents a method to compress irregularly sampled height-fields based on a multi-resolution framework. Unlike many other height-field compression techniques, no resampling is required so the original height-field data is recovered (less quantization error). The method decomposes the compression task into two complementary phases: an in-plane compression scheme for (x, y) coordinate positions, and a separate multi-resolution z compression step. This decoupling allows subsequent improvements in either phase to be seamlessly integrated and also allows for independent control of bit-rates in the decoupled dimensions, should this be desired. Results are presented for a number of height-field sample sets quantized to 12 bits for each of x and y, and 10 bits for z. Total lossless encoded data sizes range from 11 to 24 bits per point, with z bit-rates lying in the range 2.9 to 8.1 bits per z coordinate. Lossy z bit-rates (we do not lossily encode x and y) lie in the range 0.7 to 5.9 bits per z coordinate, with a worst-case root-mean-squared (RMS) error of less than 1.7% of the z range. Even with aggressive lossy encoding, at least 40% of the point samples are perfectly reconstructed.

KEYWORDS: height-field, point compression, multi-resolution height encoding, irregular sampling

1 INTRODUCTION

Digital representations of 3D data have become increasingly common as the cost of the hardware required to gather and present such data has declined. There has also been a steady growth in data volumes, since higher sampling rates inevitably lead to more accurate analysis and/or better visualization. To facilitate the efficient use of limited data storage resources in the face of this rapid data growth, one needs to employ appropriate 3D data compression techniques. A great many compression algorithms exist, optimized for a number of different 3D representations.

In the computer graphics domain, for example, triangle-mesh compression schemes dominate [1, 2, 3, 4]. Such schemes compress the 3D coordinates of a *triangle-mesh*, a popular model representation, along with a description of how to re-connect the mesh vertices. In contrast, in the field of digital image processing, image data are viewed as regularly spaced, discrete samples of an image function $I(x, y)$ — a height-field. Such a structured 3D data representation can be compressed very efficiently using transform techniques, such as the Discrete Wavelet Transform [5].

There are, unfortunately, 3D data sets which do not readily conform to either of these two dominant paradigms. This paper addresses the compression of data that exists as a height-field, but one that is *irregularly* sampled — perhaps with a very high degree of irregularity. Examples of such data occur frequently in the domain of GIS: contour lines, valley/ridge lines, Triangulated Irregular Networks (TINs) and so on. In each case there is no requirement for a regular underlying sampling of the $x-y$ plane. Such representations are not well suited to compression schemes presented above since they do not exhibit the predictable structure these techniques are predicated on. Furthermore, image compression schemes are usually lossy (the data is not perfectly recovered): in many cases the sample sites are chosen to optimize certain criteria and allowing the compression algorithm to modify these values may render the data less useful. However, one would like the freedom to allow lossy compression in either the sampling plane or the associated height values, should this prove desirable. In particular, one may wish to accommodate some amount of error in the height-field values.

The central goal of this work is to provide a compression framework which will compress irregularly sampled height-field data in such a way that the sample sites are preserved as far as possible, whilst allowing additional compression gains to accrue by loosening our requirement on lossless compression of the z values. To achieve these objectives we decouple the compression of the sample sites (in the $x - y$ plane) from their associated z values. The sample site values may be scattered arbitrarily about the image plane and there is thus little chance of exploiting correlations without some prior knowledge, which we wish to avoid. Fortunately, there are existing schemes that are well suited to compressing such data [6]. To compress the height-field values, we assume that some underlying surface function is being sampled and that the function is single-valued at each point. Since there is usually some degree of correlation in samples drawn from a surface, even if they are fairly scattered, we have developed a multi-resolution representation

Email: Patrick Marais `patrick@cs.uct.ac.za`, James Gain `jgain@cs.uct.ac.za`

to represent the height information. While our primary goal is lossless compression, the technique can accommodate a degree of lossyness in the z values. This approach is in marked contrast to general 3D compression schemes which allow 3D points to drift through space as different degrees of lossy compression are applied. An obvious benefit of this decoupling is that incremental improvements in either compression phase can be seamlessly integrated without changing the other component.

Results are presented for several different kinds of data and demonstrate that consistently good compression performance is obtained with widely varying sampling strategies.

This paper is arranged as follows: Section 2 covers necessary background to place the scheme in context and also examines related prior work. The compression framework is presented in Section 3. Section 4 evaluates the technique and provides results which quantify both lossy and lossless performance. Section 5 concludes the paper and discuss some possible areas for future work.

2 BACKGROUND AND RELATED WORK

Data compression is an enormous field and spans many disciplines. This section will focus on the areas most relevant to this research: the compression of images (which may be viewed as regularly sampled height-fields) and (3D) surface compression/approximation schemes.

Image compression has a long history, but came into its own with the advent of linear transform techniques such as the Karhunen-Loeve transform, Discrete Cosine Transform and more recently, the wavelet transform [7]. The JPEG image standards provide for both lossy and lossless encoding, depending on the application. An image is a simple height-field in which z samples are regularly spaced in the $x - y$ plane. As such it is amenable to clever schemes such as Wavelet Zero-Tree coding [5], which implicitly encodes the position of important wavelet coefficients. This provides a huge saving in terms of coding efficiency. While some heighfield data, such as digital elevation maps, are well suited to such techniques, other height data, such as contour samples or ridge-line data completely breaks the regularity of the sample sites. Transform image compression techniques are likely to perform poorly on such sparse data since the underlying transformations are designed to exploit the natural intensity correlations of neighbouring pixels across an image. Introducing sharp deltas in intensity will damage image compressibility. Furthermore, sampling the $x-y$ coordinates at the required resolution (typically 2K X 2K or more) will require the compression of large images, consisting mostly of useless information.

These objections suggest the use of a 3D point compression scheme, which can directly encode the positions of points in space. Spatial decompositions such as those based on the kd-tree representation [8] implicitly encode empty spatial regions and, for sparsely scattered points, reduce the amount of information required to represent the point set. A recent advance in this area [6] improves on the performance of kd-tree representations by effectively discarding the ordering information of a point-set sequence. This scheme has been used for the compression of point sets in 2D and 3D [9] and produces results which are competitive with the standard geometry encoders used to compress triangle mesh geometry. Unfortunately, it treats point locations as offset values in a cuboid volume and this does not allow for effective use of the underlying height-field structure. Another class of 3D point compressors, exemplified by [10] and [11], build a 3D graph structure and use simple prediction techniques to encode point coodinates. Such techniques perform badly when the underlying point distribution is unpredictable, making them a particularly poor choice for irregular point set compression.

An alternative approach to 3D point compression is to generate a proxy triangulation and apply one of the many compression techniques that exist for triangle meshes. In this case one assume that the 3D points are sampled from some underlying surface and are not arbitrarily scattered throughout the 3D volume. This is not a problem for height-field encoding: the sample sites have a single associated z value. Unfortunately, the schemes commonly employed to perform such compression [1, 4] use the connectivity to help reconstruct the geometry, and this extra information needs to be stored in addition to the compressed point scheme. There is also no obvious mechanism to control the accuracy of the z reconstruction value while leaving the sample sites untouched. For some schemes [2], the addition of a boundary for the proxy surface also requires additional storage, further reducing the benefit. Finally, the geometry encoding is usually accomplished by means of local predictor schemes (see for example [1]) which make assumptions about the distribution of points in space. For data which has peculiar sample sites (such as contour data), the proxy triangulation will contain an undesirable mix of triangle slivers and mismatched triangle sizes which will significantly degrade geometry predictor performance.

A class of mesh compressors which bear special mention are those based on the wavelet transform. The wavelet transform defined over a mesh exhibits the same desirable properties as its 2D counterpart. Unfortunately, such schemes are problematic for our application. The most general wavelet mesh compressors (see, for example, [12]) require costly topological information to encode connectivity changes across levels in the wavelet hierarchy. Alternatively, one may opt for schemes which require "subdivision connectivity" [13] or a predictable pattern to refine/decimate sample sites [14]. Both of these methods effectively resample the underlying mesh approximation and are thus unsuitable for our purposes. Approaching the compression problem in a somewhat more piece-meal fashion, [15] decompose the input mesh into a set of height-fields which are then wavelet encoded. Unfortunately, a fundamental component of this technique,

which enables incredibly low bit rates, is the regular resampling of mesh patches onto rectangular image regions. Since our goal is to preserve the location of the sample sites, any scheme which perturbs this site data is unacceptable.

For these reasons we have rejected mesh compression schemes and instead focused on a scheme which considers only the sample sites and their associated z values.

3 MULTI-RESOLUTION HEIGHT-FIELD ENCODING

The approach presented below is a synthesis of a number of existing techniques. The algorithm preserves sample site locations, but can accommodate slight deviations in the height values attached to each sample.

3.1 Encoding the $x - y$ site data

Rather than assume a known sample site distribution, or otherwise restrict the generality of the algorithm, we have chosen a spatial decomposition scheme well suited to representing scattered points. This approach uses a kd-tree type decomposition to partition space into a number of disjoint rectangular cells each of which contains a single point. The location of each point is encoded using $\log_2(x.y)$ bits by means of arithmetic coding [16], where x is the width and y the height of the cell. The reader is referred to [6] for details. It should be noted, however, that any 2D point encoding scheme can be used – there is nothing in the z encoding component that requires the use of a specific site encoding scheme. Although this work emphasizes lossless encoding of the $x - y$ site data, a lossy technique could also be employed if the site locations were deemed no less important than the z information.

3.2 Encoding the z data

Since the z values will, in general, be correlated across the base plane, the algorithm can exploit this fact to facilitate a more compact encoding. We have developed a multi-resolution (MR) scheme which progressively coarsens the input point set and generates a sequence of scalar height offsets which contain the detail necessary to reconstruct the original point set. While this bears superficial resemblance to schemes such as [14], we do not create new and arbitrary sample sites while decomposing or reconstructing the point set. Each sample site exists in the 2D encoded site information and one thus recovers data only at the original points.

To coarsen the mesh, a 2D proxy (Delaunay) triangulation is constructed: this enables one to get some sense of the sample-site distribution. The proxy triangulation is then used to build a Maximal Independent Set (MIS) [17] to select a scattered set of 'in between' points. This generally selects points which do not share a common edge in the triangulation, and can be implemented efficiently. The MIS typically only discards $\frac{1}{6}$th of the points in an input set [17], which is much less than the $\frac{3}{4}$ reduction that the prototypical 2D wavelet image schemes exhibit. Nonetheless, it provides a means of reducing the sample set in manner that ensures a good point spread across the sample domain. Other point decimation schemes exist (for example [18]) but these are generally more complex to implement and consequently may be less robust. The MR algorithm is independent of the precise decimation algorithm employed, requiring only that the selected points are scattered throughout the 2D plane.

The points that remain after sample point reduction ('decimation') are used to construct an interpolated 'approximation surface'. This is accomplished by the use of a Thin Plate Spline [19] interpolant. The interpolant provides a predictor for the z values at discarded sample sites, and allows us to calculate the z deltas required to recover the discarded z values. This is similar to the approach used in [14], but does not introduce arbitrary sample sites and uses a *global* interpolant (see Section 4). The algorithm is applied iteratively until only a small set of sample sites and

```
Xc = compress2D(X)
repeat J times
    D = { }
    T(X) = triangulate2D(X)
    Y = MIS(T) // construct an 2D MIS of T
    P0 = Y and P1 = X \ Y // partition points
    S = Interpolate(P1)
    for each point (Px,Py,Pz) in P0
        dz = Pz – S(Px, Py)
        D = {D, dz}
    arithmetic code D
    X = P1
arithmetic code z values of P1
```

Table 1: Compression algorithm.

```
// decode 2D vertices
X = {(Px, Py, 0)} = decode2D(Xc)
// build decimation hierarchy
H = {}
repeat J times
    T = triangulate2D(X)
    Y = MIS(T) // build MIS based on T
    P0 = Y and P1 = X \ Y // partition points
    H = {P0, H}
    X = P1
// decompress z values
decode ||X|| (approx) values from bit-stream
for i = 1 to J
    S = Interpolate(X)
    P0 = H[i] // the ith subset of H
    DZ = decode ||P0|| values from bit-stream
    for k = 1 to ||P0||
        dz = DZ[k]
        (P0[k])z = S( (P0[k])x, (P0[k])y) + dz
    X = {X, P0} // yields z values for X & P0
```

Table 2: Decompression algorithm.

associated z values remain (the 'approximation set'), along with a sequence of z deltas which show a generally increasing trend. Together with the retained z samples, the interpolated surface, and knowledge of the 2D sample sites, this information is sufficient to recover the input point set. The user specifies a parameter, J, which determines the number of decomposition steps to be performed.

The final height values and deltas are encoded using adaptive arithmetic coding [16] to ensure that each level of this multi-resolution structure uses as few bits as possible. Each level in the MR structure will require 2 parameters for the arithmetic coder: the minimum and maximum values of the range of symbols. The number of symbols for each level is inferred from the input 2D site information. The complete algorithm is summarised in Table 1.

Note that the deltas and smoothed point set height values are all integers: the quantized input values are integers and the interpolant is thus forced to return integer height values.

Decompression has the same computational complexity, but requires that one first compute the decimation hierarchy. This is the set of point indices which show the order in which the 2D point set should be processed to generate the correct z information. The algorithm is summarised in Table 2.

Compression is lossless (to within initial data quantization error). Should one wish a small amount of lossyness in the z values, one can employ a simple thresholding scheme. This does not involve sophisticated rate-distortion optimizations, but constructs a simple zero-threshold for each level and seems to work well in practice. An effective rate-distortion framework is an area of future research.

To generate a threshold, he user specifies a number r, between 0 and 1, and this is used to generate an initial threshold of $r2^L$, where L is the number of bits used to represent height values. This is used as a direct threshold for the first level of detail: all detail (delta) values falling below this are set to 0. For each subsequent level, the threshold is halved and the same procedure applied. This ad hoc process works well and produces representations which are quite compact with little average error.

4 RESULTS AND DISCUSSION

The algorithm described above has been applied a number of data sets which encapsulate a wide variety of z sampling strategies. While the data set tested is small, the compression gains are consistent across all sample types and we do not expect much variance in compression ratios should more examples of these height-fields be used.

To allow for meaningful comparison, a well known 3D point cloud compressor [6] was implemented. This serves as a base case for all compression results. The progressivity feature of this approach was not implemented, since this allows no means to accurately control site placement and changes the number of samples in the point set. The authors also allude to a point distribution model that can help improve results under their scheme. Unfortunately this is not fully described, and, in any event, this assumes certain characteristics w.r.t. the sample sites, which amounts to using *a priori* knowledge.

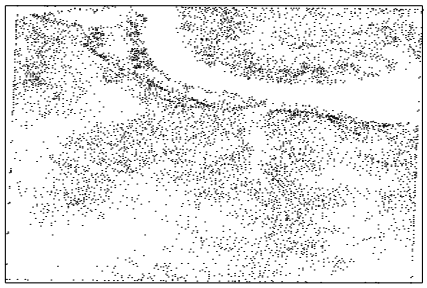
Ten levels of decomposition were used for all experiments i.e. $J = 10$. This generates a single low resolution point set and ten detail levels which need to be compressed. In general, higher values of J are desirable since they allow the MR hierarchy to come into play. However, if the hierarchy is too deep one soon reaches a point of diminishing returns due to compressor overheads and increasingly irregular (uncorrelated) delta values. Other J values may be suitable for different kinds of data; however, ten levels of decomposition provided a good trade-off for our application.

Each data set requires two numbers to specify the arithmetic coder: the minimum and maximum values for the sequence of numbers. Twelve bits were used for each signed number, resulting in 12*2*10 + 12 = 252 bits (since the coarsest level has an implicit lower bound of 0). This cost is included in the reported z bit rates, and results in a negligible contribution to the total bit-rate in all but the smallest of data sets. Bit rates are reported in bits per point (bpp).

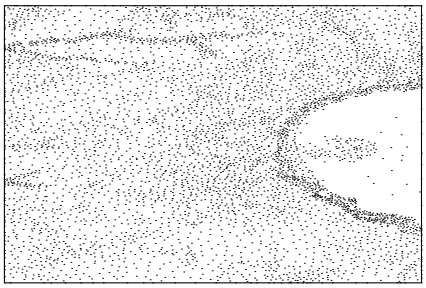
Results do not include running times for the system since this is an proof-of-concept prototype and is not optimized. The largest time component is attributable to thin-plate spline interpolation. This is not surprising since a naive thin-plate spline implementation requires the inversion of a large $N \times N$ matrix for N points. Our implementation does include a simple modification to the interpolation step to reduce computational overhead. The initial sample domain is split into an 2D array of disjoint rectangular regions, none of which have more than a user-specified number of points (this upper limit was arbitrarily set at 150). The interpolant is then solved over these rectangular regions, and includes points from an overlap zone that extends 30% into all neighbouring regions. Very little time was spent experimenting with optimal values for the overlap size or the maximum number of points in each patch, but the above values proved satisfactory for all point set types that were processed. With this modification, it takes about 125 seconds to process the 18,384 points for the 'valley' data set on a 3GHz P4 system with 1GB of memory. About 120 seconds of this time is spent solving for the interpolant.

The point data sets we used are shown in Figure 1. The data set includes three contour point sets, two TINs and one sampled network. An additional very small network, *Sparse*, is included in the compression performance figures for completeness. The network has too few points to render effectively and was thus not included in subsequent figures.

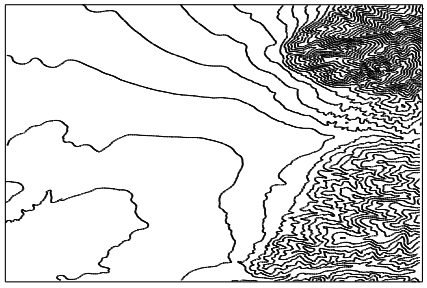
Examining Table 3, one can see that lossless compression gains are heavily dependent on the underlying sampling pattern. The contour data sets seem to be best suited to this compression scheme, although this



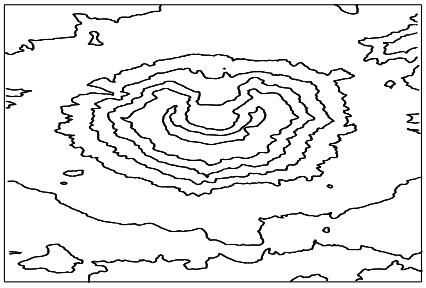
River



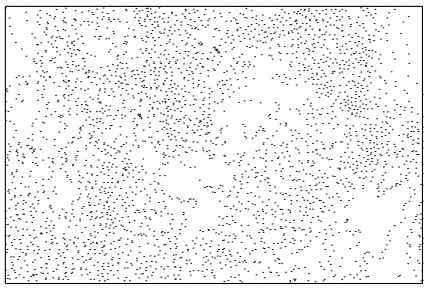
Crater



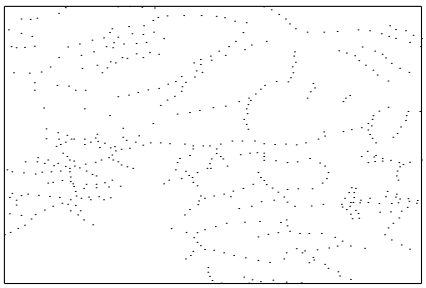
Valley



MtStHelens

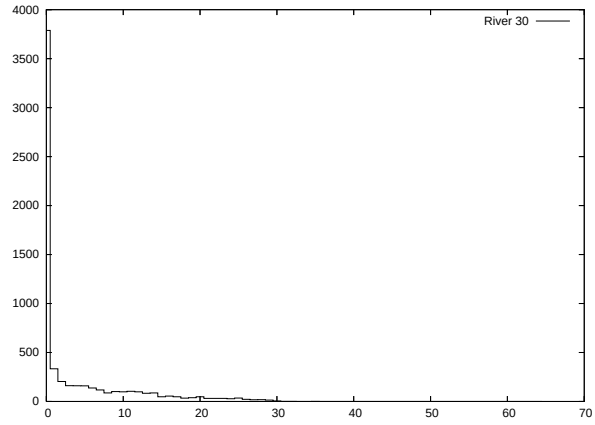


Mountainous

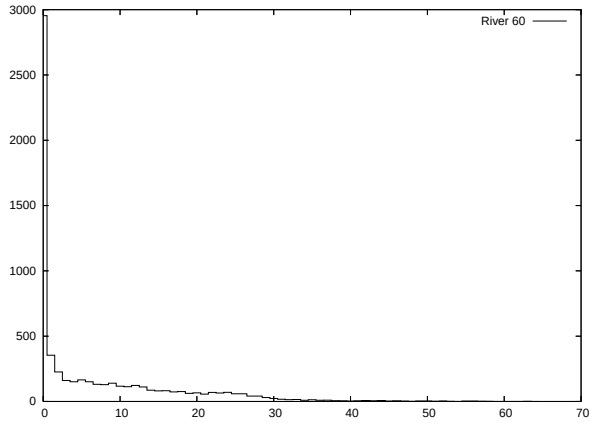


Terrain

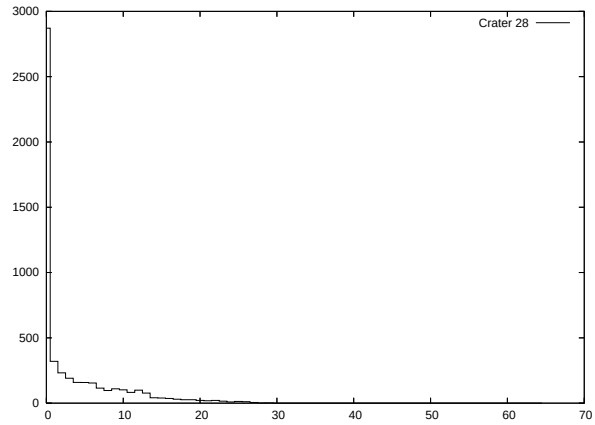
Figure 1: Data: point set models used for compression experiments.



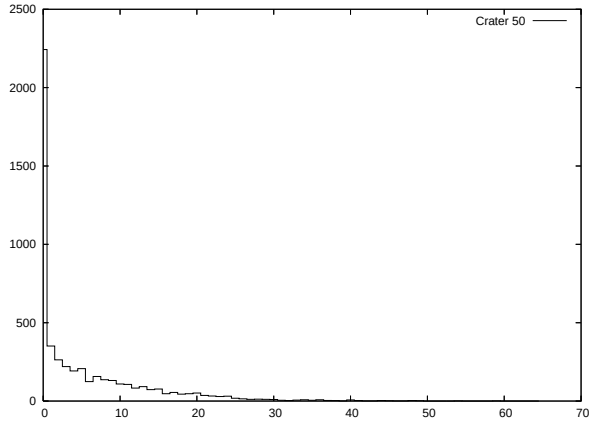
River (rms=0.71%)



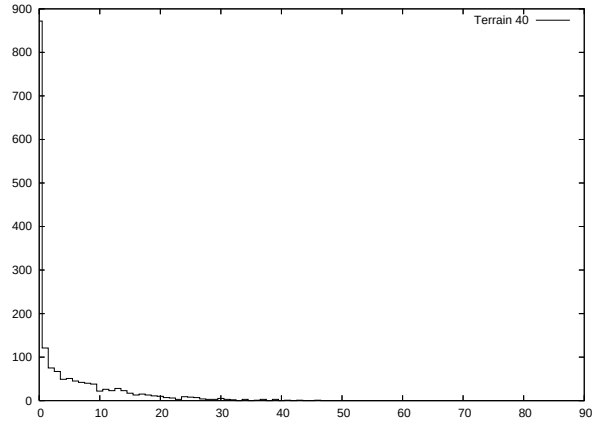
River (rms=1.13%)



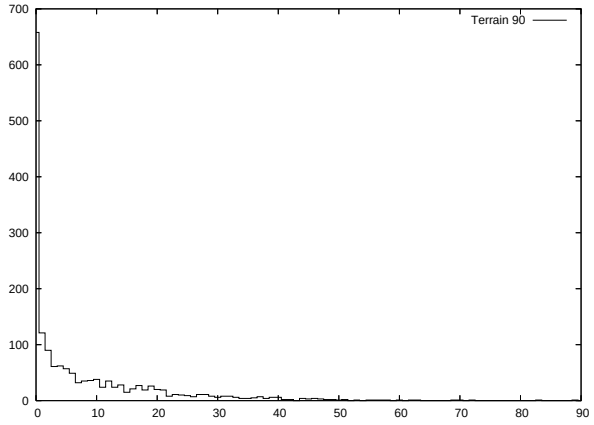
Crater (rms=0.61%)



Crater (rms=0.90%)

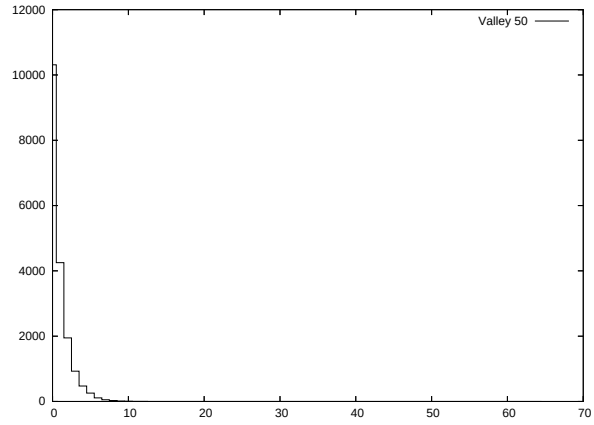


Terrain (rms=0.80%)

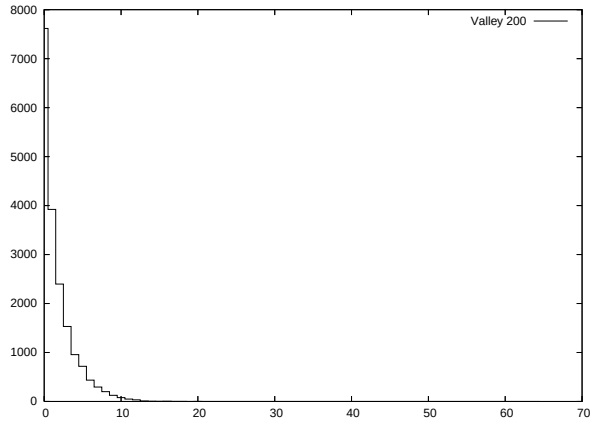


Terrain (rms=1.35%)

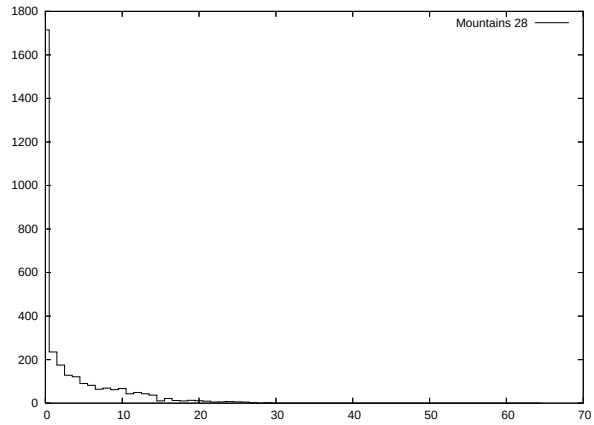
Figure 2: Z Reconstruction error histograms (A): The x-axis corresponds directly to absolute Z error; the counts (y-axis) are the number of points with the corresponding error. The left column shows a higher fidelity reconstruction, while the right column shows the result for a more aggressive threshold (the numbers are provided in the accompanying table).



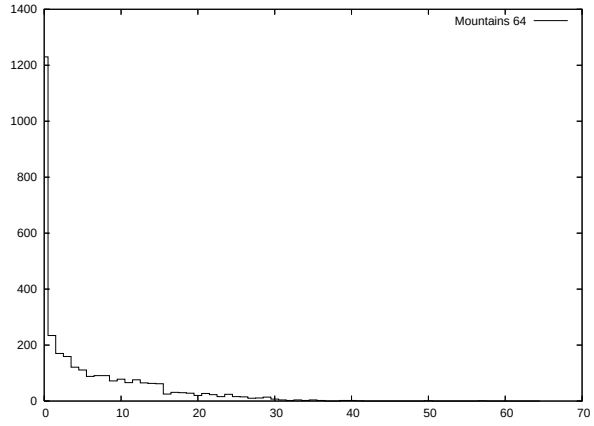
Valley (rms=0.15%)



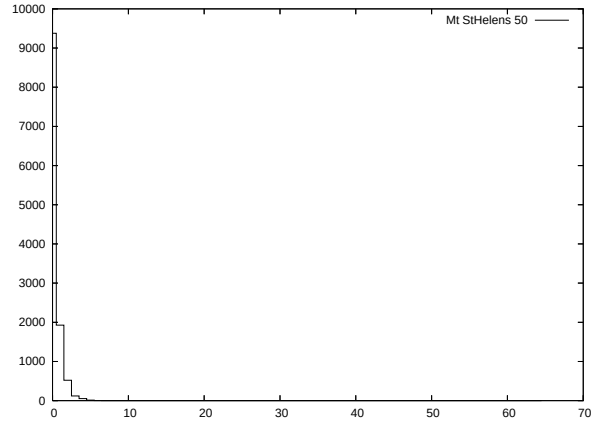
Valley (rms=0.27%)



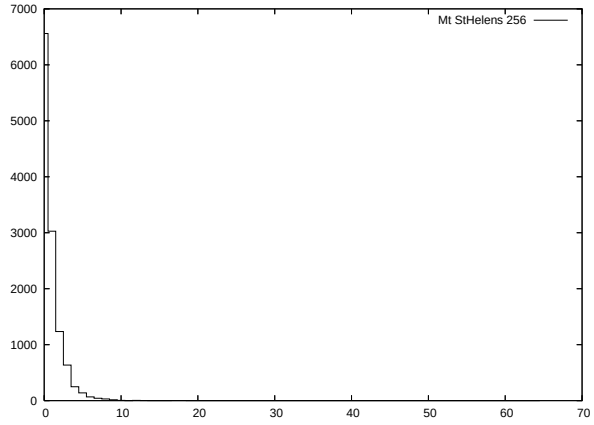
Mountainous (rms=0.56%)



Mountainous (rms=0.91%)



MtStHelens (rms=0.07%)



MtStHelens (rms=0.16%)

Figure 3: Z Reconstruction error histograms(B): The x-axis corresponds directly to absolute Z error; the counts (y-axis) are the number of points with the corresponding error. The left column shows a higher fidelity reconstruction, while the right column shows the result for a more aggressive threshold (the numbers are provided in the accompanying table).

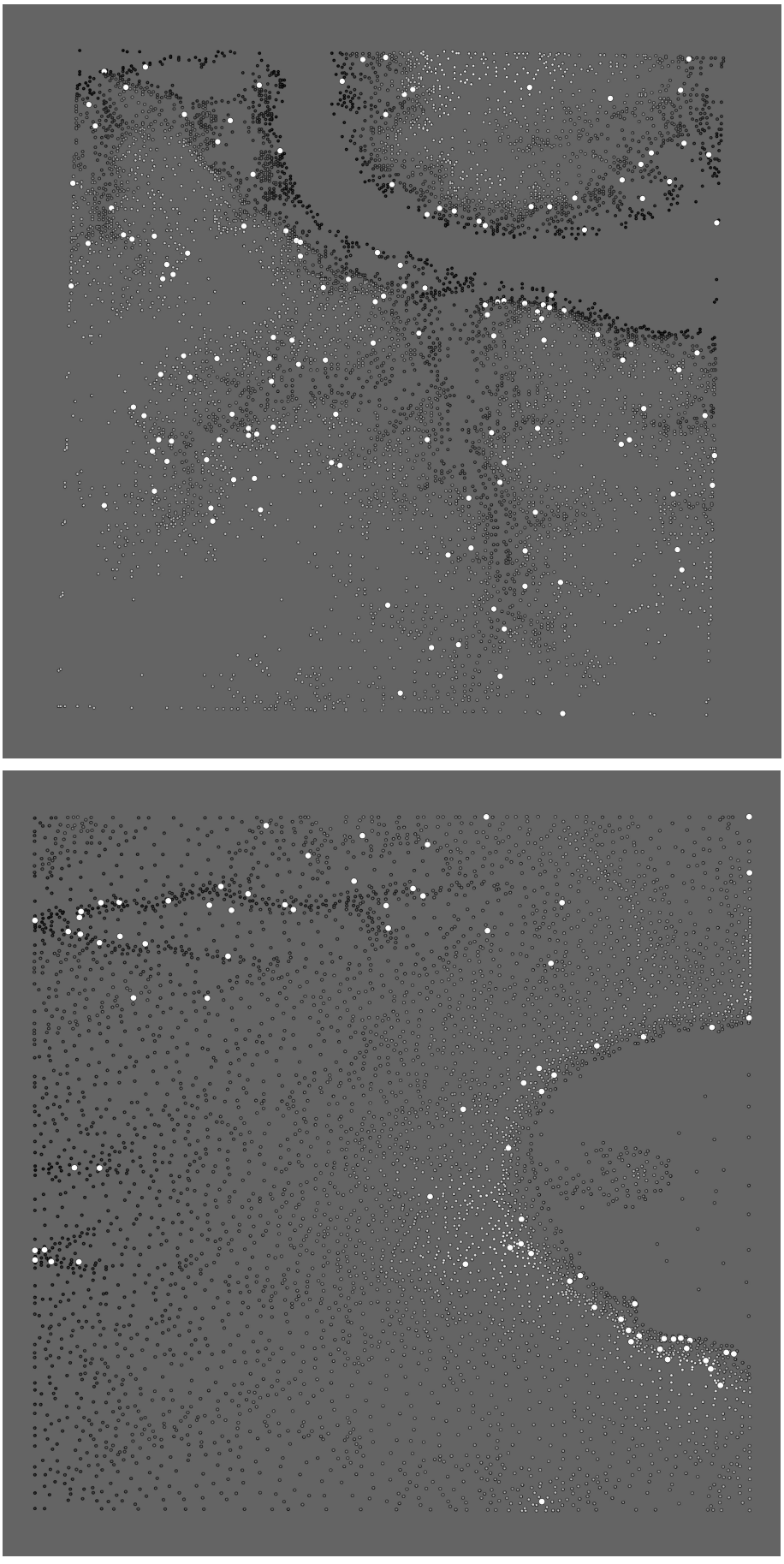


Figure 4: Reconstruction Error — largest error contributions: the greyscale values encode height on an increasing scale from black to white. The larger white points show the locations of the largest reconstruction error (in this case 50% of the error range).

Name	#points	Type	D3D	x-y	z	Total	gain z	gain x-y	gain total
Valley	18384	Contour	14.54	9.99	2.89	12.88	71%	58%	62%
Mountainous	3098	Contour	21.50	14.35	6.60	20.95	34%	40%	38%
MtStHelens	12014	Contour	13.18	9.53	1.51	11.04	85%	60%	67%
River	6226	TIN	21.10	12.90	7.41	20.31	26%	46%	40%
Crater	5080	TIN	20.95	13.58	7.03	20.61	30%	43%	39%
Terrain	1673	Network	23.53	15.02	8.08	23.10	19.2%	37.4%	32%
Sparse	519	Network	25.34	16.63	7.63*	24.18*	23.7%	30.7%	29%

Table 3: Lossless encoding: The point sets were all quantized using the same parameters: 12 bits each for x and y, and 10 bits for z. D3D represents the bit-rate obtained by the reference scheme. $x-y$ is the bit-rate for the 2D compressor we used, and z is the bit-rate for the height sample encoding. Total is the total bit rate for our scheme. The gains represent the % improvements over the input data size (24 and 10 bits per point for $x-y$ and z, resp.) * Note that for the Sparse network we have discounted 252 bits required for the z header due to the small number of points.

Name	lossless	lossy	CR	RMS Error %	Max Error %	z perfect (%)
Valley	2.89	1.34	7.5	0.15%	1.37%	66.1%
		0.95	10.5	0.27%	1.95%	41.4%
Mountainous	6.60	4.55	2.20	0.562%	2.83%	54.3%
		3.50	2.86	0.91%	5.57%	39.7%
MtStHelens	1.51	0.95	10.5	0.073%	1.37%	79.1%
		0.67	14.9	0.16%	1.76%	54.6%
River	7.41	5.53	1.8	0.71%	3.42%	60.8%
		4.66	2.15	1.13%	6.15%	47.4%
Crater	7.03	4.98	2.0	0.61%	2.83%	56.5%
		4.24	2.36	0.90%	5.76%	44.1%
Terrain	8.08	5.87	1.70	0.80%	4.49%	52.1%
		5.01	1.99	1.35%	8.68%	39.3%
Sparse	7.63	6.61	1.51	0.90%	4.10%	63.2%
		5.92	1.68	1.67%	8.20%	46.1%

Table 4: Lossy z encoding: When the z value is permitted a degree of error we can achieve significant improvements in z compression, with little loss of reconstruction fidelity. Compression error is measured as the square root of the average of the sum of squared differences (RMS) between the uncompressed and compressed z values at the sample sites. The error is expressed as a percentage of the z quantizer range (1024 in our experiments). The maximum error (as a percentage of the z quantizer range) is also reported, along with the number of sample sites at which perfect reconstruction (the correct bin value) is still achieved. CR is the overall z compression ratio of compressed to input data (10 bits per sample). Compression results are in bits per point.

is affected by the underlying coherence of the contour samples. For example, the 'mountainous' data set is a very coarsely sampled contour, and the lack of sample coherence makes it hard to interpolate the sudden changes in neighbouring z values. The smooth contours also show larger $x-y$ compression gains when compared to the z compression gains. While z values are correlated, the decimation strategy pays no attention to this 2D structure, and will discard sample sites arbitrarily. Of course, if one uses *a priori* knowledge then this could be improved. Doing so would, however, restrict the utility of the z compressor. Both TINs and network data (which are generally unstructured) require large amounts of data for both $x-y$ and z components. Again, the scattered nature of the data means that one can expect little else. Final data rates lie in the range 21-24 bpp, which is still a notable improvement over the 34 bpp required for each quantized input sample. These data sets, in particular, can benefit from some degree of lossy compression.

Lossy compression results are presented in Table 4. As explained in the text, a threshold is specified by the user and this forces small delta values to be truncated to zero. The table shows two compression settings which correspond to a lower and higher compression ratio, the latter with correspondingly greater reconstruction error. Reconstruction errors are reported as RMS, normalised to the z range (1024 bins (10 bit range) for our examples). All the contour data sets compress very well when some lossyness is permitted. The highly unstructured data sets show less of an improvement, but the gains are still noteworthy. The RMS error generally remains low (less than or around 1%), even in the presence of aggressive thresholding. The quality of reconstruction is not unduly compromised: the table shows that even in the worst case,

almost 40% of the z data values are perfectly reconstructed. The maximum error can grow quite large (8.7% for the worst case reconstruction), but these errors are infrequent, as demonstrated by the RMS figures.

The distribution of reconstruction error is presented in Figures 2 and 3. These histograms show the number of data set points associated with each (unit size) error bin. The bin corresponding to zero reconstruction error contains the largest number of points and bins which lie further away from the zero bin, contain a decreasing number of points. More agressive thresholding (the right-hand column) results in greater error and has the effect of lengthening the 'tail' of the appropriate histogram. The histogram counts decay in an approximately exponential fashion, which is necessary for good entropy coding/data reduction. Observe that the histogram counts do not decay monotonically. This is a consequence of the crude "rate-control" mechanism — a simple threshold. Nonetheless, the statistical properties of the error distribution are still well suited to entropy coding, as illustrated the by compression results.

Qualitative data on error distribution is presented in Figure 4, which examines the two TIN data sets ('river' and 'crater') in greater detail. The height values are coded on an increasing greyscale ramp (from black to white), while the errors lying above 50% of the error range are shown in as larger white points. As expected, the most significant errors lie along sharp ridges, such as the crater lip or the sharp sides of a canyon. There are also other scattered error sites. Some of these arise because of boundary approximation problems. Others simply reflect the choice of threshold in a region where the interpolant cannot properly match terrain roughness.

The above results show that the errors introduced by lossy compression allow us to reduce data sizes significantly at the cost of only minor z degradation. For all the test data sets, RMS errors of less than 1% can be accommodated while achieving perfect reconstruction at more than half the sample sites in the images. While the gains are poor for sampled networks, it is unlikely that generic point schemes could improve on this without prior information.

5 CONCLUSION AND FUTURE WORK

We have presented a framework for the lossless and near-lossless compression of irregularly sampled height-fields. By decomposing the 2.5D compression problem into two complementary components — a 2D sample site encoder, and multi-resolution z encoder — we ensure that the sample sites are accurately preserved, even in the presence of lossy compression, and we are free to substitute any reasonable compression scheme for either phase should one become available. For the diverse sampling strategies we tried, the lossless compression performance is better than that provided by the best generic point encoder we could find [6]. Of course, lossless compression provides limited gains, so we can also enable lossy compression of the z values, without disturbing the sample sites, and this ensures notable gains with low reconstruction error.

This was a prototype system, designed to test the feasibility of the algorithm. There are several places in which additional research time could be profitably spent.

Interpolant: Thin plate splines are convenient and intuitive: they require no additional parameters and fit the data in a sensibly constrained way. However, they are slow from a computational point of view. A local interpolation (and thus faster) scheme, perhaps based on a Fast Radial Basis implementation [20] could be considered.

Rate-Distortion: A threshold is a crude means of controlling and quantifying data loss. One could perform a rate-distortion optimization to optimally allocate bin widths to each of the detail quantizers.

Better $x-y$ compression: The $x-y$ compressor we used works well enough, but will becomes less attractive as the sample site density increases. In this case, even a simple run-length encoding scheme may provide better compression ratios. An approach which scales well to any irregular sampling is desirable.

ACKNOWLEDGMENTS

This work was supported by a UCT University Research Council grant and the NRF. We hereby acknowledge the various institutions and research groups which hold copyright on the terrain models used.

REFERENCES

[1] C. Touma and C. Gotsman. "Triangle Mesh Compression". In *Proceedings of Graphics Interface*. 1998.

[2] J. Rossignac. "3D Compression Made Simple: Edgebreaker with Zip&Wrap on a Corner-Table". In *Proceedings of the International Conference on Shape Modeling & Applications*, p. 278. IEEE Computer Society, 2001. ISBN 0-7695-0853-7.

[3] P. Alliez and M. Desbrun. "Valence-Driven Connectivity Encoding for 3D Meshes". In A. Chalmers and T.-M. Rhyne (editors), *EG 2001 Proceedings*, vol. 20(3), pp. 480–489. Blackwell Publishing, 2001.

[4] H. Lee, P. Alliez and M. Desbrun. "Angle-Analyzer: A Triangle-Quad Mesh Codec". In *Eurographics conference proceedings*, pp. 383–392. 2002.

[5] Z. Brahimi and K. A. Saadi. "Color Image Coding based on Embedded Wavelet Zerotree and Scalar Quantization". *ICPR*, vol. 01, pp. 504–507, 2004. ISSN 1051-4651.

[6] O. Devillers and P.-M. Gandoin. "Geometric compression for interactive transmission". In *Proceedings of the conference on Visualization '00*, pp. 319–326. IEEE Computer Society Press, 2000. ISBN 1-58113-309-X.

[7] A. Drozdek. *Elements of Data Compression*. Brooks/Cole Publishing Co., Pacific Grove, CA, USA, 2001. ISBN 053438448X.

[8] M. de Berg, M. van Kreveld, M. Overmars and O. Schwarzkopf. *Computational geometry: algorithms and applications*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1997. ISBN 3-540-61270-X.

[9] P.-M. Gandoin and O. Devillers. "Progressive lossless compression of arbitrary simplicial complexes". In *Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, pp. 372–379. ACM Press, 2002. ISBN 1-58113-521-1.

[10] B. Merry, P. Marais and J. Gain. "Compression of dense and regular point clouds". *Computer Graphics Forum*, vol. 25, no. 4, pp. 709–716, 2006. doi: 10.1111/j.1467-8659.2006.00993.x.

[11] S. Gumhold, Z. Kami, M. Isenburg and H. Seidel. "Predictive point-cloud compression". In *SIGGRAPH '05: ACM SIGGRAPH 2005 Sketches*, p. 137. ACM Press, New York, NY, USA, 2005. doi: http://doi.acm.org/10.1145/1187112.1187277.

[12] S. Valette and R. Prost. "Wavelet-Based Multiresolution Analysis of Irregular Surface Meshes". *IEEE Transactions on Visualization and Computer Graphics*, vol. 10, no. 2, pp. 113–122, 2004. ISSN 1077-2626.

[13] E. J. Stollnitz, T. D. DeRose and D. H. Salesin. *Wavelets for Computer Graphics: Theory and Applications*. Morgan Kaufmann Publishers, San Francisco, CA, USA, 1996.

[14] J. Wu and K. Amaratunga. "Wavelet triangulated irregular networks." *International Journal of Geographical Information Science*, vol. 17, no. 3, pp. 273–289, 2003.

[15] T. Ochotta and D. Saupe. "Compression of point-based 3D models by shape-adaptive wavelet coding of multi-height fields". In *Proceedings Symposium on Point-Based Graphics*. Zürich, Switzerland, June 2004.

[16] A. Moffat, R. Neal and I. Witten. "Arithmetic coding revisited". *ACM Trans. Inf. Syst.*, vol. 16, no. 3, pp. 256–294, 1998. ISSN 1046-8188.

[17] B. Jünger and J. Snoeyink. "Selecting Independent Vertices for Terrain Simplification". In V. Skala (editor), *WSCG'98 Conference Proceedings*. 1998.

[18] C. Moenning and N. Dodgson. "A new point cloud simplification algorithm". In *The 3rd IASTED International Conference on Visualization, Imaging and Image Processing*. 2003.

[19] F. Bookstein. "Principal warps: thin-plate splines and the decomposition of deformations". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, pp. 567–585, 1989.

[20] D. Suter. "Fast evaluation of splines using poisson formula". *Journal of Applied Science and Computations*, vol. 1, no. 1, pp. 70–87, June 1994.