

THE RPI GEOSTAR PROJECT

FRANKLIN W.R.

Rensselaer Polytechnic Institute, TROY, UNITED STATES

INTRODUCTION

The RPI GeoStar project was sponsored by The Defense Advanced Research Projects Agency from March 2005 to January 2009. DARPA also sponsored several related projects at other universities and companies. This paper is about only the RPI project, which enabled us to pursue terrain ideas that we had been working on for many years.

Franklin et al. [3] summarized an earlier stage of this project. The web site [2] gives more information. This paper is the first publication after the project's completion.

GeoStar's success was due to many people, of which the following co-authored at least one paper. The project lead was W. Randolph Franklin, with a background in GIS and Computational Geometry. The assistant director was Prof Barbara Cutler, with an expertise in Computer Graphics. Prof Frank Luk, now at Hong Kong Baptist University, supplied the numerical analysis knowledge. Prof Marcus Andrade visited from the Federal University of Viçosa in Brazil for more than a year. Dr Metin Inanc[8] and Dr Dan Tracy[17] completed their PhDs, while Chris Stuetzle and Zhongyi Xie's theses are in progress. Jake Stookey[13], Zhongyi Xie[19], and Jon Muckell[12] completed masters projects.

The GeoStar tasks included TERRAIN REPRESENTATION and TERRAIN OPERATORS. In this context, TERRAIN means the elevation of some land surface above a reference geoid. The specific tasks were not completely defined at the project's start, but were updated from time to time depending on our progress and the sponsor's current needs.

TERRAIN REPRESENTATION

Terrain representation included morphological terrain sculpting, Overdetermined Laplacian PDE (ODETLAP), Triangulated Irregular Networks (TIN), and lossy compression of terrain and slope. Terrain operators included a siting/intervisibility toolkit, trajectory (aka path, motion) planning, and drainage analysis.

MORPHOLOGICAL TERRAIN SCULPTING was an attempt to represent terrain by a sequence of operators designed to mimic how the terrain was formed. The prime operator was a SCOOP, which acted as follows. A scoop started followed a trajectory starting at some point in the terrain and ending at a point on the edge of the terrain. The scoop's elevation remained constant or decreased, and as the scoop moved, it removed a cross section of terrain around its position. The operation was analogous to an ice cream scoop. Terrain formed by a sequence of scoops has the desirable property of never having a local minimum. While morphological terrain sculpting remains our most promising long term form of terrain representation, most of our resources were dedicated to more short term projects. remains only potential.

An OVERDETERMINED LAPLACIAN PDE (ODETLAP) was our major terrain representation, and forms the basis of most of our current work. Key differentiating factors of ODETLAP include the following. When used to interpolate elevation contours, it produces a smooth surface in which the original contours are usually invisible. It facilitates progressive transmission of the surface. It can conflate inconsistent partially overlapping data sets. It can fit a surface to complete contour lines, partial contours lines, isolated points and partial sets of elevation posts. When fitting a surface to a set of nested contours, it will infer a local maximum inside the topmost contour. We will discuss ODETLAP more later in this paper.

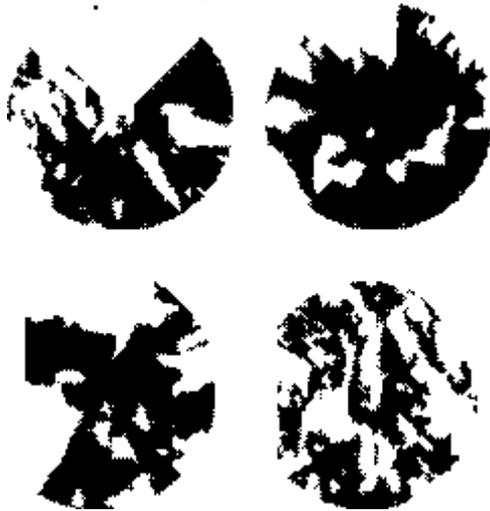
A TRIANGULATED IRREGULAR NETWORK (TIN), or piecewise linear triangular spline, was our third terrain representation. There have been many variants of TINs since the first TIN in cartography was implemented by Franklin, under the direction of Tom Poiker and David Douglas, in 1973[5]. Our version has some advantages. It can easily process a grid of 10000 by 10000 elevations on a laptop. From those potential points of the TIN, it chooses actual points for the TIN incrementally in order of their importance, rather than in a fixed scan-line order. Therefore it produces a sequence of approximates of ever increasing accuracy. Finally, it operates in core, which greatly increases its speed.

Using these representations, we were able to perform a lossy compression of terrain, with a reasonable error, compared to storing the elevations as binary integers, by a factor of 100 or more.

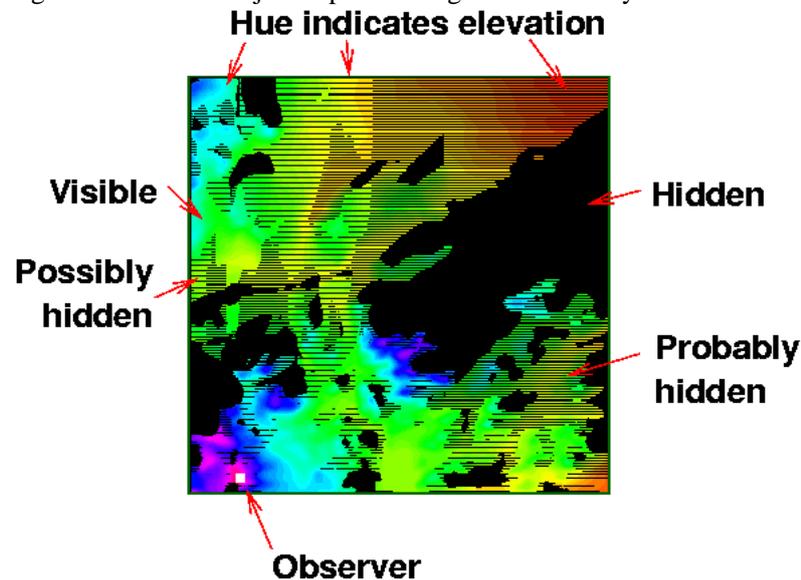
TERRAIN OPERATIONS

A SITING/INTERVISIBILITY toolkit was our first terrain operation. We produced the algorithm, then contracted the implementation, as an ArcGIS DLL, to ESRI, and moved on to other tasks. The siting problem is to find sites (locations) for observers that will jointly cover the maximum amount of terrain. The observers are placed at a given height above the terrain, and can each see out to a given distance, the Radius of Interest, to look for potential targets that may be at a given distance above the terrain. If the lines of sight are blocked by the terrain, the viewshed, or the terrain that each observer can see, is reduced. There may be hundreds of observers, and the terrain cell may be 3600x3600 posts. The intervisibility problem is to restrict the observers' sites so that the enough pairs of observers can see each other that any pair of observers can exchange messages, at least indirectly. Our contribution was to handle so many observers with high-resolution viewsheds on large terrains.

The following Figure shows four sample viewsheds.



The following Figure shows viewshed uncertainty. In this example, a small change in the interpolation algorithm between adjacent posts changed the visibility of one half of all the points.



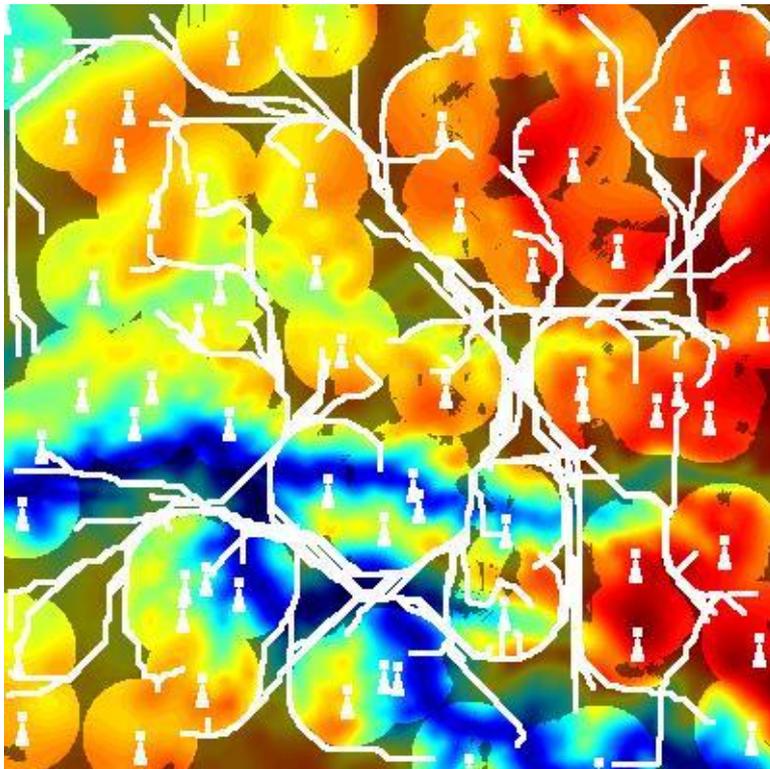
PATH PLANNING, or MOTION PLANNING, was our next terrain operation. We implemented this on terrains up to 3600x3600, such as urban LIDAR data. Our computed paths are not necessarily quasi-straight, but can have complicated curves as they optimize a realistic non-symmetric objective function. It combines a weighted function of (1) the Euclidean distance between the source and goal, (2) the amount of elevation gained, and (3) the amount of distance traveled inside given forbidden regions, such as the viewsheds of opponents' observers. The exact metric is given by the following equation whose value is integrated along the path.

$$c = \sqrt{\Delta x^2 + \Delta y^2 + \Delta z^2} \cdot \left(1 + \max \left(0, \frac{\Delta z}{\sqrt{\Delta x^2 + \Delta y^2}} \right) \right) \cdot (1 + 100v)$$

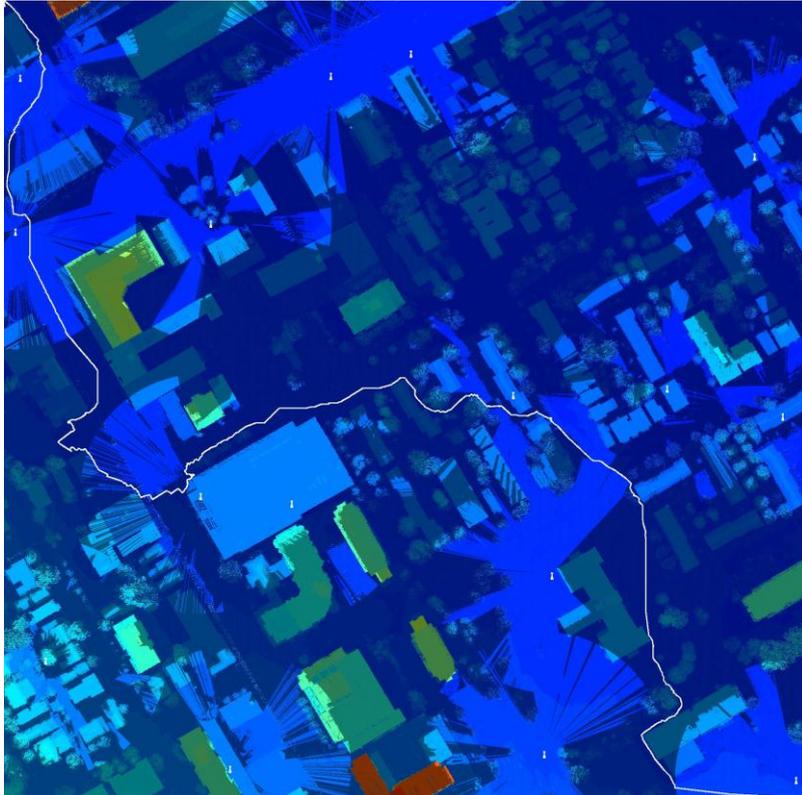
v is a bit that tells whether the path is in a forbidden zone. Note that this function is not a mathematical metric function since it is not symmetric in the source and goal. Uphill is more expensive than downhill[18].

Combining those two operations permits red-blue games over some terrain. One opponent might optimize observers to cover the terrain, while the other opponent then computes paths to avoid them. Another possible round could be for the first opponent to observe the choke points where many paths converge and add new observers to block them.

The following Figure shows many optimal paths computed between pairs of points on opposite sides of the terrain. The little lighthouses represent the observers, and the bright regions their viewsheds. The elevation is represented by the hues. The choke points are clear.



The following Figure shows path planning on some urban LIDAR data.



LOSSILY COMPRESSING terrain while representing the SLOPE accurately was another terrain operation. Slope is important for mobility, erosion, aircraft, visibility, and recognition. From calculus, it is well known that the accuracy of an approximation $g(x)$ to a function $f(x)$ says nothing about how close $g'(x)$, the derivative of $g(x)$ is to $f'(x)$. Indeed, as an approximation to terrain elevation gets better, the slope accuracy may get worse. Commercial GIS tools can represent slopes so badly that the terrain is not recognizable; for copyright reasons, we do not include examples here.

ODETLAP, the OVERDETERMINED LAPLACIAN METHOD was the fundamental terrain representation for GeoStar. In the context, ODETLAP fits a complete array of terrain elevation posts to a small set of known elevations. The method is to set up and solve an overdetermined system of linear equations. The idea was inspired by a digital Laplacian partial differential equation system; however its properties are quite different.

Let $\{(i,j,h[i,j])\}$ be the set of known elevations or heights. The height at location (i,j) is $h[i,j]$. Our output will be the complete array of elevations $z[i,j]$. The $z[i,j]$ are the unknown variables, to be solved for. We construct two classes of equations. The first class has one equation for each known point:

$$z[i,j] = h[i,j]$$

The second class has one equation for each nonborder point:

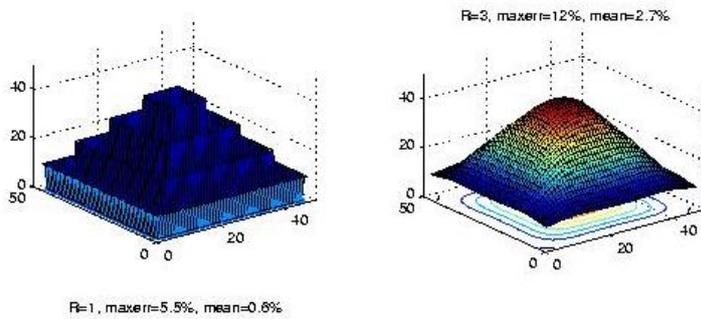
$$4z[i,j] = z[i-1,j] + z[i+1,j] + z[i,j-1] + z[i,j+1]$$

Therefore each known point has two equations, making the system overdetermined. We solve for a best fit, using a normalized overdetermined sparse linear system solver in Matlab. Childs[1] uses a Page-Saunders method to process 1201×1201 cells. We can emphasize accuracy vs smoothness by weighting the two types of equations differently.

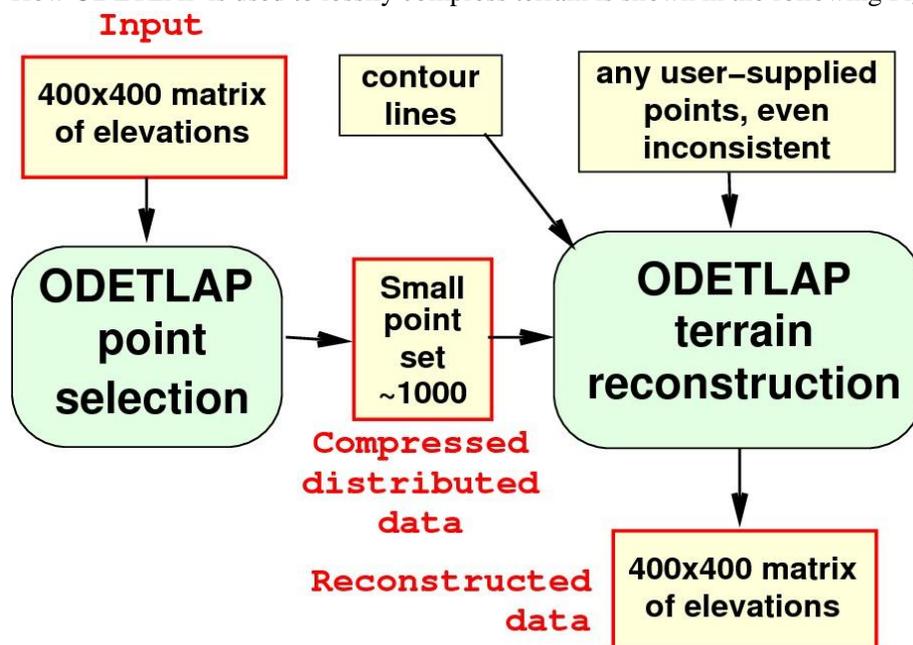
The original goal of ODETLAP was to interpolate from elevation contours to a grid without showing terraces on the contour lines[6]. Many competing methods have the problem that information does not flow across the contours, which makes the slopes discontinuous at the contours. Also, the common technique of firing a ray from the test point to the first known point is not conformal. That means that it cannot handle nested contours shaped like kidney beans.

The equations at the borders of elevation array are messy since there are not four neighboring points to make this border point the average of. However, depending on the number of known points, omitting that equation can make the whole system underdetermined. Making each border point the average of its two or three neighbors biases the surface to be horizontal at the border, which is undesirable. The same problem is faced in Computer Aided Design at the end point of a spline.

The following Figure shows how well ODETLAP handles a hard example. The input is a set of nested contours with sharp corners. Nevertheless, the output has a smooth silhouette, with no terracing, and even an inferred top. The viewpoint was chosen to display the silhouette edges since there any irregularities in the generated surface become very evident.



How ODETLAP is used to lossily compress terrain is shown in the following Figure.



ODETLAP plays slightly different roles in the compression and in the decompression. During the compression of a 400x400 array of elevations, a greedy algorithm selects a set S of representative posts for the terrain. S is initialized either with a coarse regular grid of perhaps 10x10 points, or with a set of perhaps 1000 points selected by our incremental Triangulated Irregular Network program. Then a surface is generated with ODETLAP, compared to the original terrain, and the points with the greatest error found. Those points are inserted into S and the process repeated. S , suitably coded, is the compressed representation of the terrain.

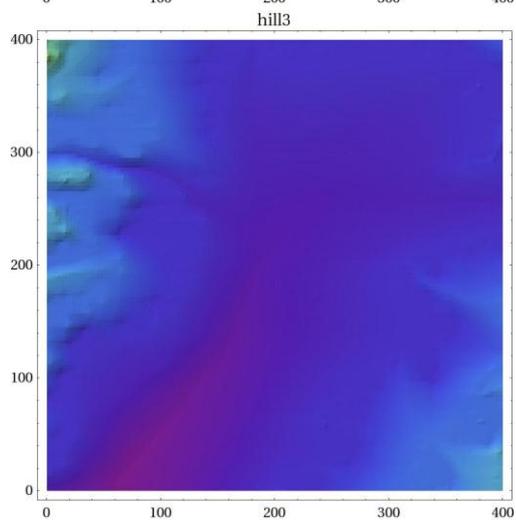
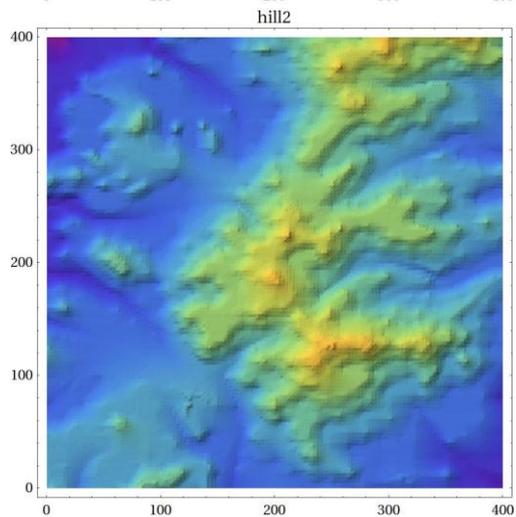
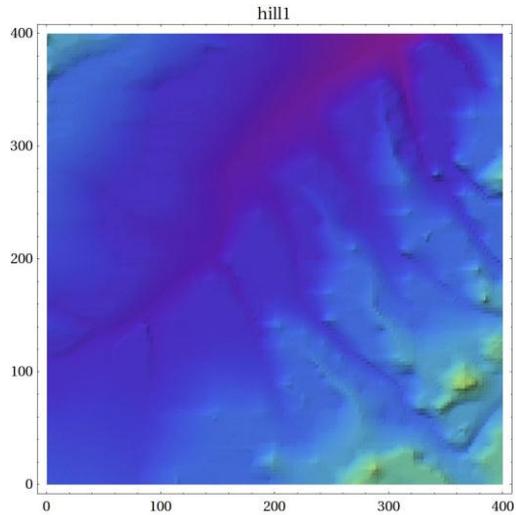
During decompression, S , or any other desired set of points, is expanded into a full elevation array with ODETLAP.

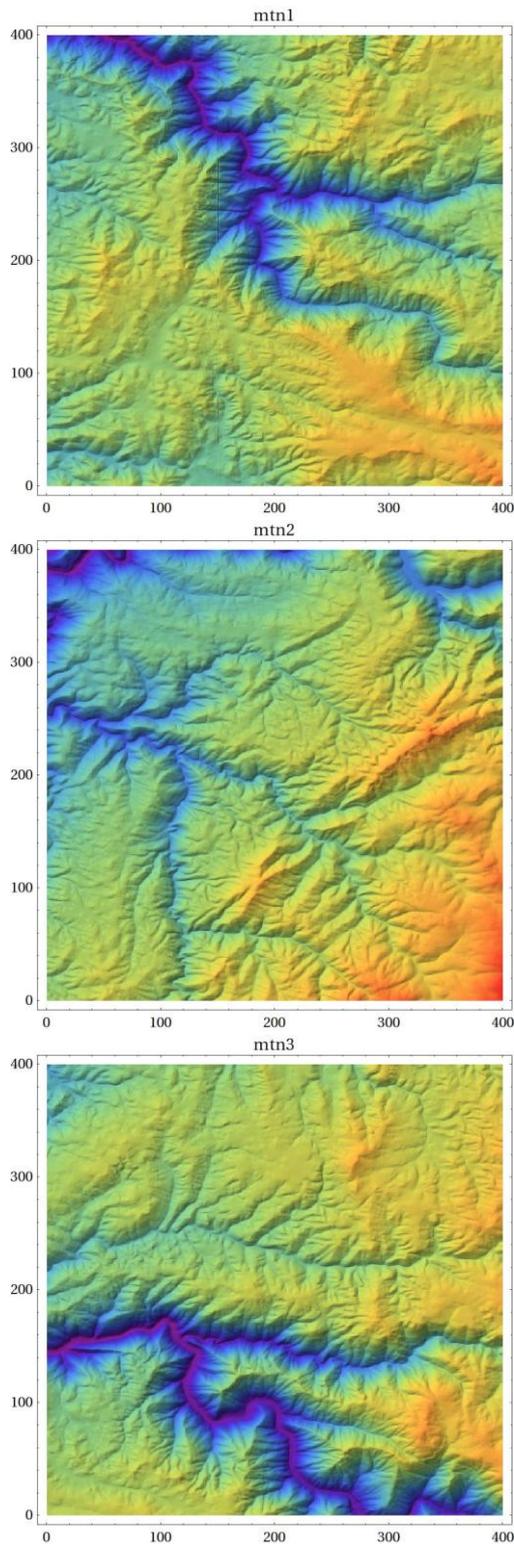
The goal of terrain compression is to minimize the file size, not merely to minimize the number of ODETLAP points. Those objectives may be quite different. For example, if the points were on a regular grid, then coding the $\{(x,y)\}$ would take no space, so more points would be possible in the same total size. Therefore CODING the points, or converting them into bits, is as important as selecting them. To code $\{(x,y,z)\}$, we separate the $\{(x,y)\}$ from the (z) . Note that the order of the $\{(x,y)\}$ is immaterial, but that once an order is chosen, then the order of the (z) is fixed.

We observe that $\{(x,y)\}$ can be considered as an array of bits, similar to a facsimile original. Therefore we utilize the considerable existing science for facsimile compression and use CCITT G4. Assuming no structure in the bits, the information theoretic limit can be computed as shown in this example. Assume that 1000 of the 400x400 bits are 1, the others 0. Then the number of combinations, or ways to select 1000 from 160000, is 160000 choose 1000. The information content is the base-2 log of that, or 8754 bits. We are within 20% of that, which is good.

That leaves the sequence of z to code. We compress that with `bzip2`, a common text compression algorithm. It might be possible to do better since `bzip2` is lossless and some imprecision would be tolerable here. We are researching that.

Our sample datasets were six 400×400 sections from level-2 DEMs. Three are hilly, three mountainous. The elevation ranges in each dataset were from 500m to 1040m. They are shown below; the images were generated in Mathematica.



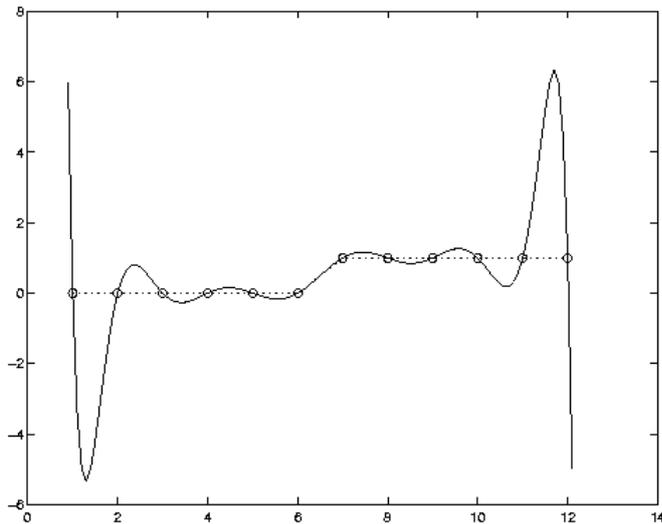


For example, we compressed hill1, which has a 505m elevation range, down from 320KB (with the elevations stored as 2-byte binary integers) to 4071 bytes, which is a factor of 80. The resulting average absolute elevation error was 2.5m, or 1/200 of the the elevation range. Sometimes terrain compression results are reported relative to an original ASCII file size. By that standard our compression was a factor of 200. Our results for the other datasets were similar or better.

Slope can be as important as elevation, as described earlier. We use the common Zevenbergen-Thorne definition of slope. Let $p[i,j]$ be the 3D coordinates of the point at $[i,j]$. The slope is defined to be the z-component of the vector cross product $(p[i-1,j]-p[i+1,j]) \times (p[i,j-1]-p[i,j+1])$. The formula is symmetric about $[i,j]$, but $p[i,j]$ does not enter in the slope computation there. If the elevation array has 1m resolution and

30m post spacing, then the slope precision is 2 degrees. Slope has a smaller autocorrelation distance than elevation, but it also has less relative precision.

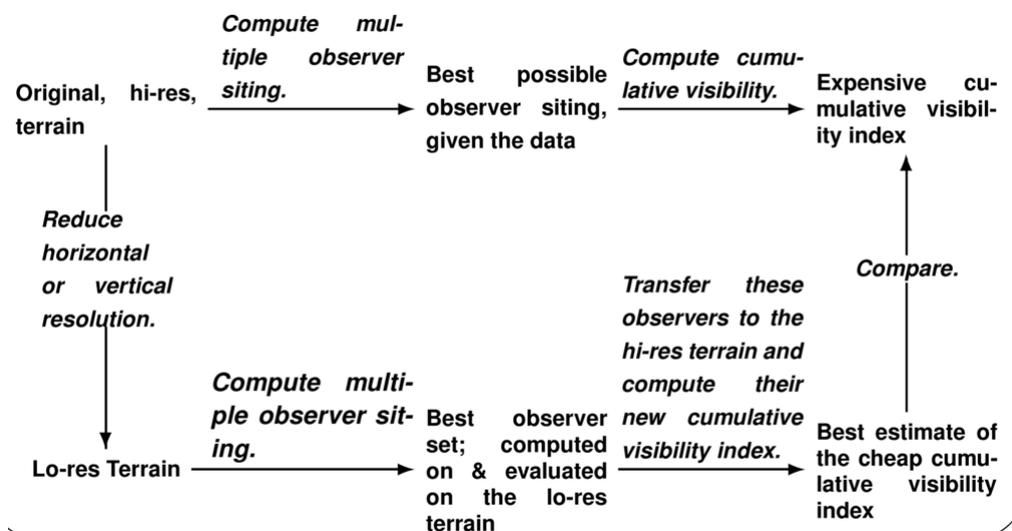
Fitting a surface to a set of points can be counterintuitive. For example, consider the following Figure, where a polynomial is fitted exactly to 12 points that are equally spaced in x, using Lagrange interpolation. The first six points have y=0, the last six have y=1. The polynomial goes exactly through all the points, but between them, swings up as far as y=6. Worse, the maximum slope is over 100. As more points are added, both these numbers increase exponentially.



Therefore our solution was to extend ODETLAP by explicitly incorporating slope into a new overdetermined linear system. This resulted in a better slope compression. When we compressed mtn2 by a factor of 15 relative to the original binary file, the average slope error was 4 degrees, or only double the inherent slope precision of the data. More details are in [4].

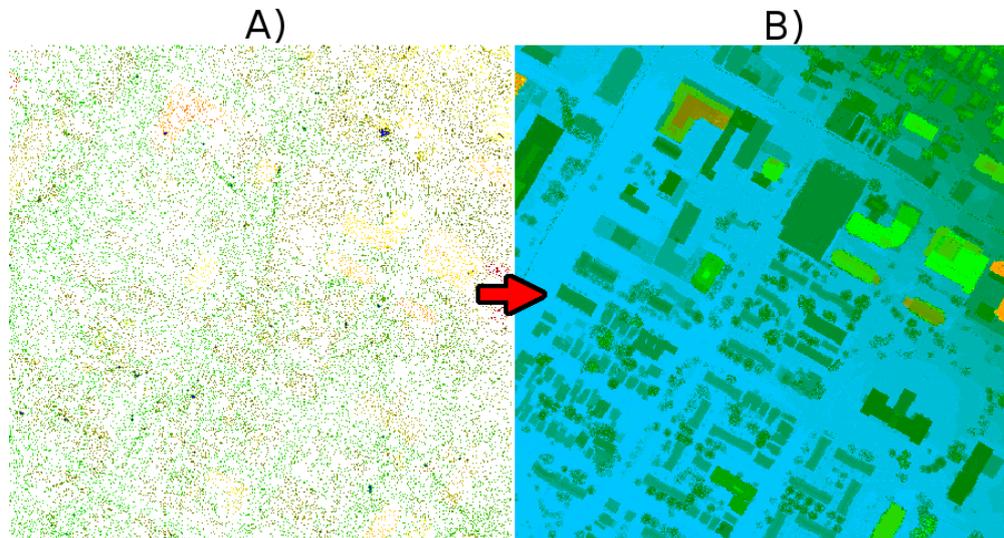
How best to evaluate the quality of a lossy compression is not entirely clear. The easy metric is RMS elevation (and perhaps slope) error. However, the terrain is being compressed for use in some application, and so suitability of the compressed terrain for that application is also important. However, that itself has some complexity. Suppose that our application is multiple observer siting. We might site observers on the original terrain, and then separately on the compressed and restored terrain. The second set of observers may be quite dissimilar to the first set. However, if its joint viewshed as just as large, then we should not care. Likewise, suppose that our application is optimal path planning.

The path computed on the compressed terrain might be quite different from the path computed on the original terrain. However, if its cost is almost the same, then the compression was adequate. The following Figure shows a commutative diagram to illustrate this.

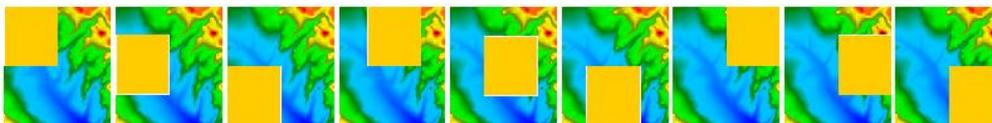


ODETLAP is quite computation intensive. Therefore we developed a parallel implementation using RPI's IBM Blue Gene/L with 32K processors. The idea was to process overlapping patches of terrain, with one patch per processor, and then to blend the resulting arrays of computed elevations into one large array. The major challenge was to manage the communication to avoid bottlenecks, since each processor has limited memory and no virtual memory. Therefore our Preliminary results were achieved on a 128-processor 2.6 GHz AMD Opteron[14].

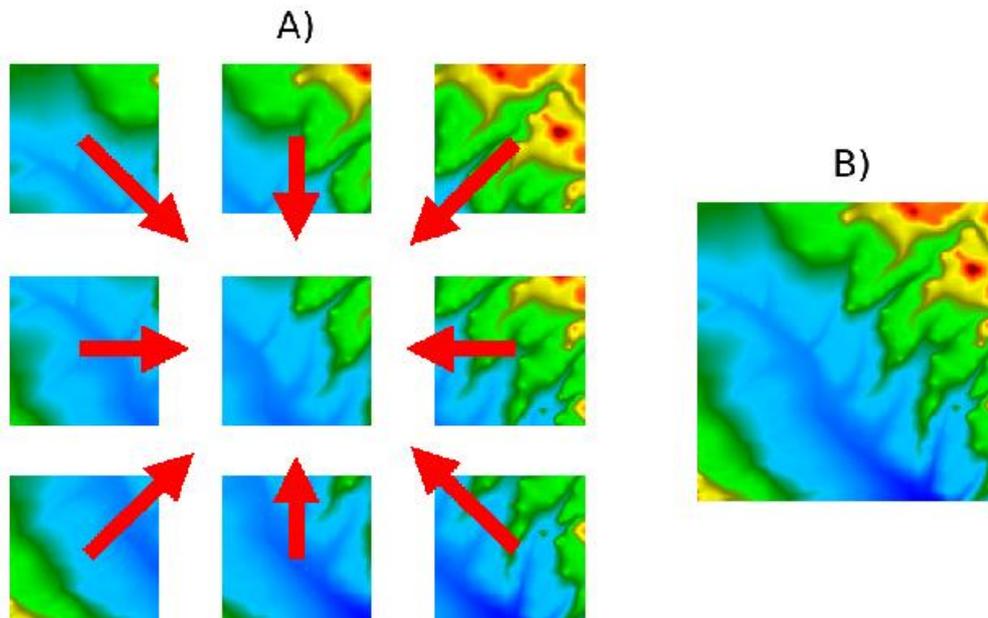
Our parallel version of ODETLAP was initially used to generate a dense 10000x10000 elevation array from raw urban LIDAR data consisting of 13 million points on the AMD cluster.



We applied ODETLAP to overlapping layers of patches. In this example, four patches would be enough to cover the entire heightmap, but we would see errors at the patch edges. Instead, we ran ODETLAP on none overlapping patches.

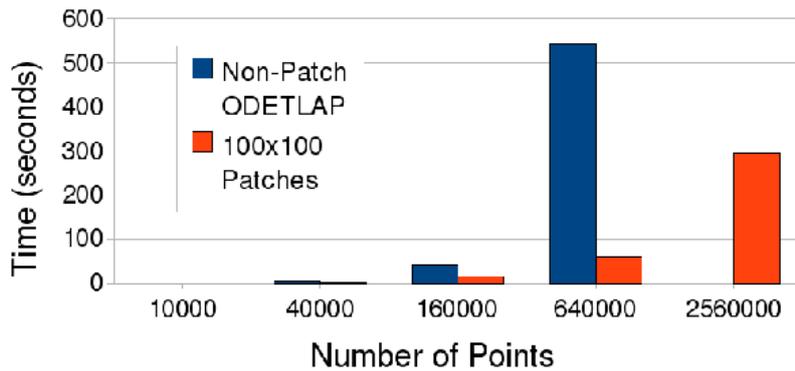


Our final step is to merge the overlapping patches (A) into the complete reconstructed elevation map (B), below.



The execution time for the non-patch version of ODETLAP grew quadratically with the number of pixels, while the patch version of ODETLAP had a linear growth. This improvement was also gained while running serially.

Non-Patch ODETLAP vs. Serialized Patch ODETLAP



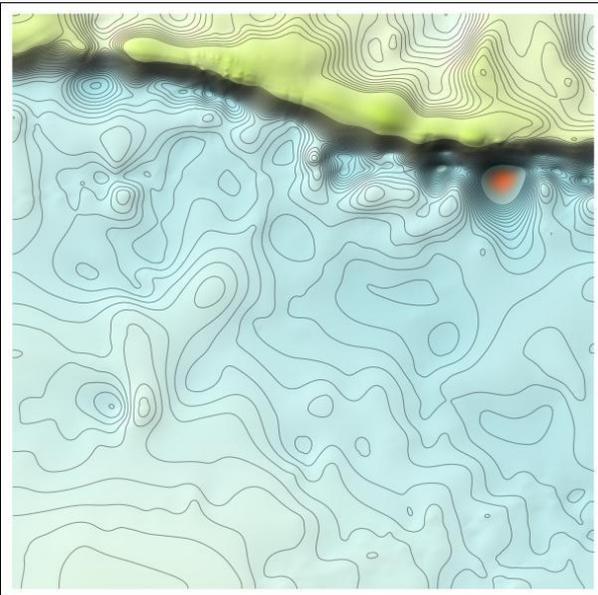
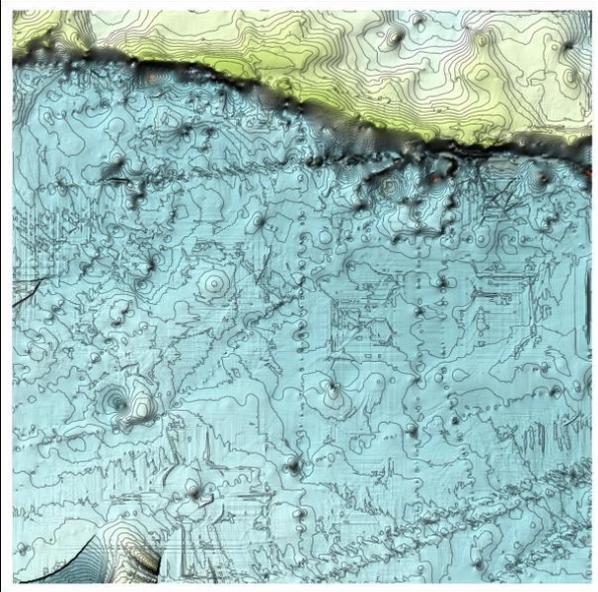
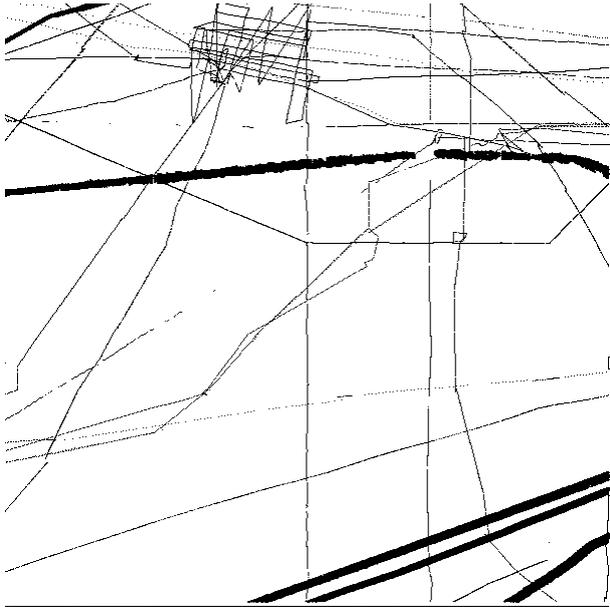
The following Figure shows the parallel speedup with the 128-processor Opteron cluster. The task was to reconstruct a 2000x2000 grid from 1% of original points by running ODETLAP on 2.6GHz AMD Opteron cluster. For up to 128 processors, there was a linear speedup. After that, the file I/O overhead dominated. The merging central process was not included above.

Patch ODETLAP Running Time when Parallelized



HYDROLOGY and BATHYMETRY

These were two applications of our terrain research. The first was to compute hydrology on terrain and then investigate the effects of compression on it[15]. The second was to use ODETLAP to fit a bathymetric surface to unevenly spaced tracklines measured on the ocean floor[10]. The goal was to produce a surface that did not exhibit the input data as artifacts. The following three Figures shown the distribution of the input data for our test case, the bad result computed by kriging, and our surface computed with ODETLAP.



SUMMARY AND FUTURE WORK

The RPI GeoStar project allowed us to extend some existing terrain research with considerable new results and achieve useful synergy. We are now engaged in various followon projects. They include more lossy compression while maintaining slope accuracy, Xie et al. [20], compressing multidimensional environmental datasets, Li and Franklin [11], computing valid river networks from incomplete observations, Lau and Franklin [9], and studying, including modeling and simulating, how levees fail when overtopped, Gross et al. [7], Stuetzle et al. [16].

ACKNOWLEDGEMENT

This research was partially supported by NSF grant CMMI-0835762.

REFERENCES

- [1] John Childs. Development of a two-level iterative computational method for solution of the Franklin approximation algorithm for the interpolation of large contour line data sets. Master's thesis, Rensselaer Polytechnic Institute, 2003. Geostar, March 2009.
- [2] W. Randolph Franklin. <http://www.ecse.rpi.edu/Homepages/wrf/Research/GeoStar>. URL
- [3] W Randolph Franklin, Metin Inanc, Zhongyi Xie, Daniel M. Tracy, Barbara Cutler, Marcus V A Andrade, and Franklin Luk. Smugglers and border guards: the GeoStar project at RPI. In 15th ACM International Symposium on Advances in Geographic Information Systems (ACM GIS 2007), Seattle, WA, USA, Nov 2007.
- [4] W. Randolph Franklin, Daniel M. Tracy, Marcus Andrade, Jonathan Muckell, Metin Inanc, Zhongyi Xie, and Barbara Cutler. Slope accuracy and path planning on compressed terrain. In Symposium on Spatial Data Handling, Montpellier FR, June 2008.
- [5] Wm. Randolph Franklin. Triangulated irregular network program, 1973. URL <http://www.ecse.rpi.edu/wrf/wiki/Research/tin73.tgz>. accessed 30 Oct 2008.
- [6] Michael Gousie and W. Randolph Franklin. Converting elevation contours to a grid. In Eighth International Symposium on Spatial Data Handling, pages 647-656, Vancouver BC Canada, July 1998. Dept of Geography, Simon Fraser University, Burnaby, BC, Canada.
- [7] Jared A. Gross, Christopher S. Stuetzle, Zhongxian Chen, Barbara Cutler, W. Randolph Franklin, and Thomas F. Zimmie. Simulating levee erosion with physical modeling validation. In ICSE-5 5th international conference on scour and erosion, San Francisco, 7-10 Nov 2010.
- [8] Metin Inanc. Compressing Terrain Elevation Datasets. Rensselaer Polytechnic Institute, 2008. PhD thesis,
- [9] Tsz-Yam Lau and W. Randolph Franklin. Completing fragmentary river networks via induced terrain. In Geospatial Data and Geovisualization: Environment, Security, and Society, a special joint symposium of ISPRS Technical Commission IV & AutoCarto 2010, Orlando, Florida, 15-18 Nov 2010.
- [10] Tsz-Yam Lau, You Li, Zhongyi Xie, and W. Randolph Franklin. Sea floor bathymetry trackline surface fitting without visible artifacts using ODETLAP. In 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (ACM SIGSPATIAL GIS 2009), Seattle WA USA, 4-6 Nov 2009.
- [11] You Li and W. Randolph Franklin. 4D-ODETLAP: A novel high-dimensional compression method on time-varying geospatial data. In Geospatial Data and Geovisualization: Environment, Security, and Society, a special joint symposium of ISPRS Technical Commission IV & AutoCarto 2010, Orlando, Florida, 15-18 Nov 2010.
- [12] Jon Muckell. Evaluating and compressing hydrology on simplified terrain. Master's thesis, Rensselaer Polytechnic Institute, 2008.
- [13] Jared Stookey. Parallel terrain compression and reconstruction. Master's thesis, Rensselaer Polytechnic Institute, 2008.
- [14] Jared Stookey, Zhongyi Xie, Barbara Cutler, W. Randolph Franklin, Daniel M. Tracy, and Marcus V.A. Andrade. Parallel ODETLAP for terrain compression and reconstruction. In Walid G. Aref et al., editors, 16th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (ACM GIS 2008), Irvine CA, 5-7 Nov 2008. URL <http://acmgis08.cs.umn.edu/>.
- [15] Christopher Stuetzle, W. Randolph Franklin, Barbara Cutler, Jonathan Muckell, Marcus Andrade, Jared Stookey, Metin Inanc, and Zhongyi Xie. Evaluating hydrology preservation of simplified terrain representations. In Walid G. Aref et al., editors, 16th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (ACM GIS 2008), Irvine CA, 5-7 Nov 2008. URL <http://acmgis08.cs.umn.edu/>. (PhD student poster and fast forward presentation).

- [16] Christopher S. Stuetzle, Zhongxian Chen, Barbara Cutler, W. Randolph Franklin, Jared Gross, Katrina Perez, and Thomas Zimmie. Computer simulations and physical modelling of erosion. In 7th International Conference on Physical Modelling in Geotechnics (ICPMG 2010), Zurich, 20-24 Jun 2010.
- [17] Daniel M. Tracy. Path Planning and Slope Representation on Compressed Terrain. PhD thesis, Rensselaer Polytechnic Institute, 2009.
- [18] Daniel M. Tracy, W. Randolph Franklin, Barbara Cutler, Franklin T. Luk, Marcus Andrade, and Jared Stookey. Path planning on a compressed terrain. In Walid G. Aref et al., editors, 16th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (ACM GIS 2008), Irvine CA, 5-7 Nov 2008. URL <http://acmgis08.cs.umn.edu/>. (poster and fast forward presentation).
- [19] Zhongyi Xie. Representation, compression and progressive transmission of digital terrain data using over-determined laplacian partial differential equations. Master's thesis, Rensselaer Polytechnic Institute, 2008.
- [20] Zhongyi Xie, W. Randolph Franklin, and Dan Tracy. Slope preserving lossy terrain compression. In 18th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (ACM SIGSPATIAL GIS 2010), San Jose, CA, USA, 2-5 Nov 2010. (PhD Dissertation Showcase).