

# Sea Floor Bathymetry Trackline Surface Fitting Without Visible Artifacts Using ODETLAP

Tsz-Yam Lau  
Rensselaer Polytechnic  
Institute  
Troy, NY, USA  
laut@cs.rpi.edu

You Li  
Rensselaer Polytechnic  
Institute  
Troy, NY, USA  
liy13@cs.rpi.edu

Zhongyi Xie  
Rensselaer Polytechnic  
Institute  
Troy, NY, USA  
xie22@cs.rpi.edu

W. Randolph Franklin  
Rensselaer Polytechnic  
Institute  
Troy, NY, USA  
wrf@ecse.rpi.edu

## ABSTRACT

High quality, artifact-free fitting a bathymetry (sea-floor) surface to very unevenly spaced depth data from ship tracklines is possible with ODETLAP (Overdetermined Laplacian Partial Differential Equation). The problem is that many data points, but with imprecise values, can be collected along and near the ships, e.g., with Multibeam Bathymetry, but there are no data between the tracklines, which may be a considerable distance apart. The numerous artifacts of previous surface fitting algorithms were so totally unacceptable that even Photoshop has been used to smooth them. In contrast, the implementation of this version of ODETLAP generates surfaces that do not exhibit artifacts even when displayed with techniques that highlight minor slope irregularities, which is, in the authors' opinion, the most appropriate evaluation metric. Since the data is imprecise, ODETLAP's surface approximates, rather than interpolates, the data. ODETLAP can trade off smoothness and accuracy to capture the small features that are not artifacts. This extension to ODETLAP is a variable smoothness parameter so that points distant from a known depth are smoothed differently from points close to a known depth. The broader implication is that ODETLAP is a very powerful algorithm with many applications. The Matlab implementation is freely available for nonprofit research and education.

## Categories and Subject Descriptors

G.1.2 [Approximation]: Approximation of surfaces and contours; H.2.8 [Database Applications]: Spatial databases and GIS; I.3.5 [Computing Methodologies]: Computer Graphics Computational Geometry and Object Modeling

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage, and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. ACM GIS '09, November 4-6, 2009, Seattle, WA, USA (c) 2009 ACM ISBN 978-1-60558-649-6/09/11...\$10.00

## General Terms

Algorithms, Experimentation, Performance

## Keywords

GIS, bathymetry, terrain reconstruction, ODETLAP, overdetermined sparse linear system

## 1. INTRODUCTION

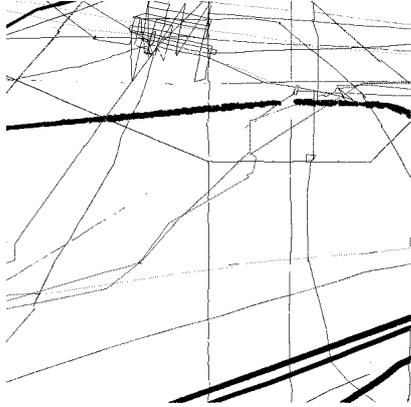
A bathymetric chart, the underwater equivalent of a topographic map, represents the depth and features of the ocean floor. A peculiar difficulty of obtaining raw data is that, unlike for terrain, there are no wide-area data capture techniques such as IFSAR or LIDAR. The popular Multibeam Bathymetry technique[13] produces many data points with 10m resolution in a swath up to 10km wide along a ship's trackline. Between the tracklines there is no data[9, 10]. The National Geophysical Data Center[5] and the Scripps Institute[7] are excellent portals into the available data and techniques. *Surface reconstruction* is required to compute a grid of  $n \times n$  predicted values  $\mathbf{z}$  from the set of known depths  $\mathbf{h}$  at  $k$  positions  $(x_1, y_1), (x_2, y_2), \dots, (x_k, y_k)$  inside the grid.

This paper will first consider how to evaluate the reconstructed surface. Then it will present several conventional methods, illustrating their weaknesses. Finally it will show how ODETLAP performs, and extend it to process bathymetric data, with significantly improved results compared to the existing methods.

Figure 1 shows the locations of the data points in the sample dataset used throughout this paper. Black pixels represent data points. In this  $601 \times 601$  pixel grid, which corresponds to a  $10 \times 10$  degree region in  $1' \times 1'$  resolution, there are 857 670 data points distributed among 23 630 different pixels that have at least one data point. We have multiple data points (mean = 36) for one pixel because the original data points have spatial position that are more precise than  $1'$ . We average all the data points in each pixel.

The resulting pixels' depths have a range of 5232m, from the deepest to the shallowest. The multiple points within each

pixel have a range up to 1926m in depth. 54 pixels have a range of at least 1000m in depth, while 7874 pixels each have a range of at least 100m in depth for the points inside them. This considerable range of depths within individual pixels indicates either the limited precision of the data or the existence of small steep underwater features.



**Figure 1: Demonstration Bathymetry Dataset: Known Points**

## 2. EVALUATION CRITERIA

Evaluating the reconstructed surface is nontrivial. There is insufficient information to extrapolate the limited spatially high-frequency data to the large regions between tracklines, which makes it difficult to tell if a predicted terrain feature reflects reality. What is left? At least, our generated surfaces should not contain features that are clearly artifacts.

Because of the data’s limited precision, approximating the data is preferable to interpolation. Nevertheless, elevation changes smaller than the known precision can still be unacceptable if the generated slopes have artifacts. Accurate slopes are important for identifying undersea landslides and for laying cables and pipelines.

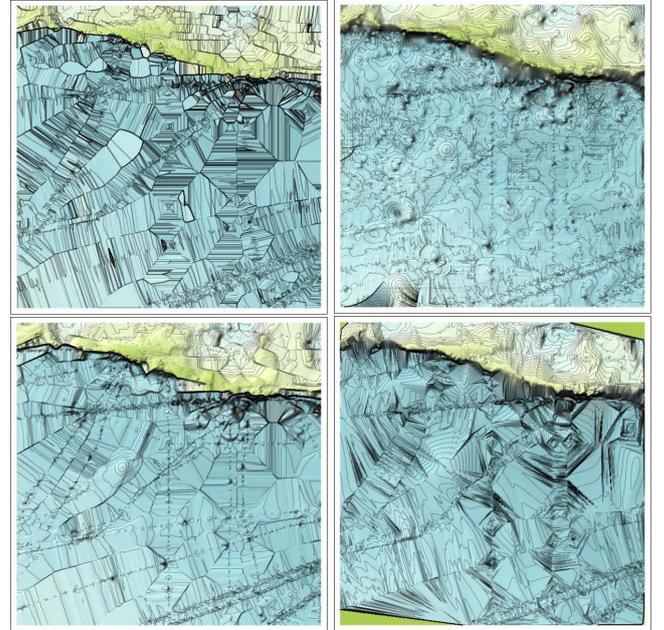
In this paper, we illustrate the different approaches with relief plots overlaid with contours using Mathematica’s *ReliefPlot* and *ListContourPlot*. That allows us to visualize both small elevation and small slope changes. Larger versions of the figures are available online[4].

## 3. PRIOR WORK

Figure 2 presents the results of some common *interpolation* methods. They include proximity polygons[12] (also known as Voronoi or nearest point polygons), inverse distance weighting[8] (using ArcGIS with power = 2, number of neighbors = 10 to 25, sector type = four), kriging[3] (using ArcGIS with type = ordinary, with a global first-order trend removed before Kriging, number of neighbors = 10 to 25, sector type = four, model: default spherical model, with nugget = 3744, range = 365, sill = 215139), and second-order spline interpolation (with Matlab’s *griddata*). These methods are all unsatisfactory because of numerous artifacts. In a satisfactory surface the known points must not be visible.

Because of the measurement imprecision described earlier, *approximation*, which allows a relaxation from the measured

values, is preferable. De-correlating depth fluctuations from the density of known positions is vital because the way data are collected does not imply anything about the roughness of the terrain. *Trend surface analysis*[14] can be useful for preprocessing the data before kriging. Finally, local sub-modeling may be preferable to a simple function for modeling the often complicated sea floor. However these involve human judgement, which is undesirable. Post-processing, manually smoothing the generated surfaces with Photoshop, has also been used to improve the surface’s appearance, CleanTOPO2[6].



**Figure 2: Top left: Proximity Polygon Interpolation; Bottom left: Inverse Distance Weighting Interpolation; Top right: Kriging Interpolation; Bottom right: Second-order Spline Interpolation**

## 4. ODETLAP

ODETLAP (Overdetermined Laplacian Partial Differential Equation) was earlier presented in the context of lossy terrain compression[1, 11, 15, 16]. ODETLAP, operating on an elevation grid where only a few points are known, computes a surface over the entire grid. It is an extension of the Laplacian Partial Differential Equation (PDE), which may be discretized as

$$z_{i,j} = \frac{z_{i-1,j} + z_{i+1,j} + z_{i,j-1} + z_{i,j+1}}{4} \quad (1)$$

In ODETLAP, each point whose depth is known induces two equations, an *averaging equation* and an *exact equation*. The averaging equation is Equation 1 above. The exact equation at some known position  $(i, j)$  is

$$z_{i,j} = h_{i,j} \quad (2)$$

For each unknown position, we create only an averaging equation. Therefore, if  $k$  of the  $n^2$  points have known depths, there are  $n^2 + k$  equations for the  $n^2$  unknown variables in  $\mathbf{z}$ . Since each equation involves only one to five of the  $n^2$  unknowns, the system is *sparse*.

When there are more equations than unknowns, and the system of equations is inconsistent, we optimize for a least squares solution. Expressing the overdetermined system as  $\mathbf{Az} = \mathbf{b}$ , the error is  $\mathbf{e} = \mathbf{Az} - \mathbf{b}$ . We want to solve for the  $\mathbf{z}$  that minimizes  $\|\mathbf{e}\|$ . Various algorithms, like Matlab’s *mtdivide* can solve this efficiently.

#### 4.1 Smoothness/accuracy factor $R$

In a least squares solution, multiplying both sides of an equation by a constant will scale that equation’s error and change its relative importance in the solution. We multiply all our averaging equations by a parameter  $R$ , which trades off accuracy and smoothness. A larger  $R$  causes the solution to be smoother, but to be farther from the known points. The basic ODETLAP has only this one parameter, which gives it great conceptual simplicity.

ODETLAP was originally created to interpolate from contour lines to an array of elevations. Its advantages include being able to work with contour lines (continuous or intermittently broken), inferring mountain tops inside a ring of contours, and enforcing continuity of slope across contours. All these are favorable features of natural-looking terrain.

However, ODETLAP is resource-intensive. For an  $n \times n$  dataset with  $k$  known points,  $T = \Theta(n^3 + k)$ . Solving our demo dataset takes about 13 CPU-minutes and 1.1GB of writable memory on a 2.8GHz Levono ThinkPad W-700 with 4GB of main memory running Ubuntu 9.04 Linux and 64-bit Matlab R2009a. For larger datasets, we can partition the domain, solve the pieces in parallel, and smoothly merge the solution pieces, Stookey et al[11].

#### 4.2 Results with fixed $R$

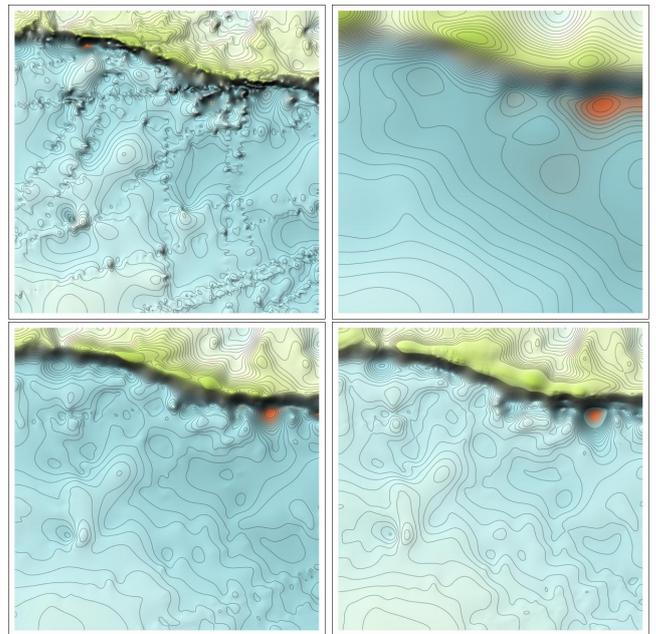
Figure 3 shows the results for various fixed  $R$ . With  $R = 1$ , the mean absolute error is 37m, or 0.7% of the data range. However, artifacts are apparent especially in the lower right corner because the surface tries too hard to track the fluctuating imprecise data points.

Using  $R = 10$  smooths the surface, reducing the artifacts but also possibly genuine features. Here, the mean absolute error is 101m or 2% of the data range, which is about the precision of the data.  $R = 100$  continues the progression, with almost no visible artifacts and a mean absolute error of 228m or 4.4% of the data range. However many sea floor details are lost. The unanswered question is whether these were real.

#### 5. EXTENSION TO VARIABLE $R$

The goal here is to spread the influence of a data point over a larger distance when the data points are more spread out. (However proximity polygons are unacceptable because of their bad results.) The new idea is to extend ODETLAP by varying  $R$  over the dataset, and using smaller values when the known points are sparser. That should reduce the mean absolute error, exhibiting fine details in the generated surface, while simultaneously reducing obvious artifacts.

Using a lower  $R$  where the known points are more spread out will better preserve the depth fluctuations. The higher  $R$  where the known points are denser will smooth the surface



**Figure 3: Surface computed by ODETLAP: Top left:  $R = 1$ ; Bottom left:  $R = 10$ ; Top right:  $R = 100$ ; Bottom right: variable  $R$  from 10 to 100**

more to reduce artifacts caused by noise in the data. That will de-correlate the trackline artifacts from how way data is collected.

#### 5.1 Result and Analysis

We prototyped the above algorithm, varying  $R$  from 10 to 100, with the result shown in the bottom right of Figure 3). That combines the best feature of the fixed  $R = 100$  case, which is an absence of artifacts, with the best feature of a fixed  $R = 10$ , which is a smaller error. Indeed, now the mean absolute error is 129m, or 2.4% of the data range, only slightly greater than for a fixed  $R = 10$ .

#### 6. CONCLUSION AND FUTURE WORK

This paper demonstrates the applicability of the ODETLAP framework in reconstructing sea floor surfaces from very unevenly distributed depth data. The key observation is that different regions now have their own unique needs for the smoothness/accuracy trade-off. A region that is densely populated with known positions will have each position’s averaging equation weighted higher to increase the surface’s smoothness there. In contrast, sparsely populated regions need a small  $R$  to reduce the error. Meanwhile, fluctuations in the generated surface are decorrelated from how the data is collected, as it should be.

We are now considering other methods and automating the determination of  $R$ , relating  $R$  to some quantifiable aspect of the localized sea floor (such as the random error of measurements), employing supplementary data such as the gravimetry data used by Smith and Sandwell[9], realizing the algorithm on GPUs[2], and generalizing the algorithm to predicting unknown values with similar input data characteristics.

## 7. REFERENCES

- [1] M. B. Gousie and W. R. Franklin. Augmenting Grid-based Contours to Improve Thin Plate DEM Generation. *Photogrammetric Engrg. & Remote Sensing*, 71(1):69–79, 2005.
- [2] GPGPU.org. GPGPU (General-Purpose computation on GPUs). <http://www.gpgpu.org>, (retr. 6/26/09).
- [3] D. G. Krige. A statistical approach to some mine valuations and allied problems at the Witwatersrand. Master's thesis, University of Witwatersrand, 1951.
- [4] T. Y. Lau, Y. Li, Z. Xie, and W. R. Franklin. ACMGIS09 bathymetry paper images. <http://wrfranklin.org/acmgis09-bathymetry-paper-images/>, (retr. 9/13/09), Sept. 2009.
- [5] National Geophysical Data Center. Multibeam bathymetry. <http://www.ngdc.noaa.gov/mgg/bathymetry/multibeam.html>, (retr. 6/8/09), Apr. 2009.
- [6] T. Patterson. CleanTOPO2: Edited SRTM30 plus world elevation data. <http://www.shadedrelief.com/cleantopo2/>, (retr. 6/8/09), Oct. 2007.
- [7] Scripps Inst. of Oceanography. Measured and estimated seafloor topography. [http://topex.ucsd.edu/WWW\\_html/mar\\_topo.html](http://topex.ucsd.edu/WWW_html/mar_topo.html), (retr. 6/8/09), Apr. 2009.
- [8] D. Shepard. A two-dimensional interpolation function for irregularly-spaced data. In *Proceedings of the 1968 ACM National Conference*, pages 517–524, 1968.
- [9] W. H. Smith and D. T. Sandwell. Global sea floor topography from satellite altimetry and ship depth soundings. *Science Magazine*, 277, issue 5334, 1997.
- [10] W. H. Smith and D. T. Sandwell. Predicted seafloor topography: NGDC data announcement number: 94-MGG-04. <http://www.ngdc.noaa.gov/mgg/fliers/94mgg04.html>, (retr. 6/8/09), July 2008.
- [11] J. Stookey, Z. Xie, B. Cutler, W. R. Franklin, D. Tracy, and M. V. Andrade. Parallel ODETLAP for terrain compression and reconstruction. In *16th ACM SIGSPATIAL Intl. Conference on Advances in Geographic Information Systems*, 2008.
- [12] A. H. Thiessen. Precipitation averages for large areas. *Monthly Weather Review*, 39(7):1082–1084, 1911.
- [13] University of Rhode Island, Office of Marine Programs. Discovery of sound in the sea: Echo sounder-multibeam. <http://www.dosits.org/gallery/tech/osf/esm1.html>, (retr. 6/8/09), Mar. 2008.
- [14] A. E. Wren. Precipitation averages for large areas. *Canadian Journal of Exploration Geophysics*, 9(1):39–44, 1973.
- [15] Z. Xie, M. A. Andrade, W. R. Franklin, B. Cutler, M. Inanc, D. M. Tracy, and J. Muckell. Approximating terrain with over-determined Laplacian PDEs. In *17th Fall Workshop on Computational Geometry*, IBM TJ Watson Research Center, Hawthorne NY, 2–3 Nov 2007. poster session, no formal proceedings.
- [16] Z. Xie, W. R. Franklin, B. Cutler, M. A. Andrade, M. Inanc, and D. M. Tracy. Surface compression using over-determined Laplacian approximation. In *Proceedings of SPIE Vol. 6697 Advanced Signal*

*Processing Algorithms, Architectures, and Implementations XVII*, San Diego CA, 27 August 2007. Intl. Soc. for Optical Engrg. paper 6697-15.

## 8. APPENDIX

The code for fixed  $R$  follows. Excluding comments and blanks, it has only 32 lines, which number could be reduced at the cost of decreased legibility. Most of the program sets up the overdetermined system of equations, which is then solved in one line. It illustrates both a major strength and weakness of Matlab. The former is that it embodies considerable numerical computational intelligence, so that we do not, e.g., need to know how to solve sparse overdetermined systems. The latter is that efficiently working with matrices requires knowing assorted obscure idioms. Explicit loops are very slow in Matlab; array notation is almost required.

```
function [W] = odelap(z,R)
% z: Square array of input elevations:
% 0: Unknown value; >0: Known value
% R: input parameter trading off accuracy on
% the known points (low R) vs smoothness (hi R).
% W: output matrix of elevations.
spparms('spumoni',0);
spparms('tight');
nzind=find(z);
z1=min(min(z(nzind)));
zh=max(max(z));
N=size(z,1);
N2=N*N;
np=size(nzind,1); % Number of nonzero elts.
leftbordind=[1:N];
rightbordind=[N2-N+1:N2];
topbordind=[1:N:N2-N+1];
bottombordind=[N:N:N2];
cornerind=[1,N,N2-N+1,N2];
R4=4*R;
maindiag=R4(ones(N2,1));
maindiag([leftbordind,rightbordind,topbordind,
          bottombordind])=3*R;
maindiag(cornerind)=2*R;
plus1diag = -R(ones(N2,1));
minus1diag = plus1diag;
plusNdiag = plus1diag;
minusNdiag = plusNdiag;
plus1diag(bottombordind)=0;
minus1diag(topbordind)=0;
B=sparse(N2,N2);
B=spdiags([maindiag,plus1diag,minus1diag,plusNdiag,
          minusNdiag],[0,-1,1,N,-N],B);
X=[1:np]';
sechalf=spconvert([X,nzind,ones(np,1);(np+1) N2 0]);
B=[B;sechalf];
C=[zeros(N2,1);z(nzind);0];
W1 = B \ C; % Solve the overdetermined linear system.
W = reshape(W1,N,N);
```

The rest of the Matlab implementation, e.g., for variable  $R$ , is freely available for nonprofit research and education.