Converting Elevation Contours to a Grid

Michael B Gousie

Dept of Computer Science
Rensselaer Polytechnic Institute, Troy, NY, 12180–3590
(518) 452–0678
`gousiem@cs.rpi.edu`

Wm Randolph Franklin

Electrical, Computer, and Systems Engineering Dept., 6026 JEC,
Rensselaer Polytechnic Institute, Troy, New York 12180–3590
(518) 276–6077, Fax: (518) 276–6261
`wrf@ecse.rpi.edu`, `http://www.ecse.rpi.edu/Homepages/wrf`

Abstract. We present two new methods for approximating elevation data from contours to a grid. The first repeatedly interpolates new contour lines between the original ones. The second starts with any interpolated or approximated surface, determines its gradient lines, and does a Catmull-Rom spline interpolation along them to improve the elevations. We compare the new methods to a more classical thin-plate approximation on various data sets. The new methods appear visually smoother, with the undesirable terracing effect much reduced.

Keywords: contours, interpolation, grids, elevation

1 Introduction

Hypsographic (elevation) data can be represented in various formats, including Triangulated Irregular Networks (TINs), grids or arrays, and contour lines. The various formats compete on factors such as complexity, size, and accuracy; it is not yet clear which format will eventually dominate. Since data may often be available only in one format, there is a need for conversion algorithms. This paper describes two new methods for converting from the contour format to an array, or grid, of elevations.

Since the grid will probably be spaced more closely than most of the contour lines, this is an interpolation problem. The ideal, but to-date impossible, criterion of interpolation quality is as follows. Start with a theoretical model of hypsography, which would map from random numbers to random terrains. This would assign a probability to each possible terrain. Just as human heights can be modeled as a normal random variable, without understanding physiology, it is conceivable that a model of hypsography might not necessarily require a deep understanding of geology. In any case, given such a theoretical model, and given a contour map, a Maximum Likelihood Estimator (MLE) would exist to find the "most likely" interpolated terrain that matched those contours. Unfortunately, until such a terrain model is available, this approach is infeasible.

Until then, the desirable characteristics of an interpolation or approximation algorithm are not obvious, since different attributes conflict with each other. For example, since geological features are stretched and distorted, an interpolation that is invariant with respect to nonuniform scaling in x and y might be desirable. However, this is not compatible with any algorithm using nearest points, such as a Voronoi method. Even allowing uniform scaling is incompatible with any algorithm containing an embedded constant distance, such as kriging. Again, second-degree continuity might be a desirable attribute, except that the real world is often not continuous at all. Forcing high-order continuity here will only create false ripples.

The conclusion is that visual inspection may be the best judge of an interpolation or approximation method. We may not be able to formalize it, but we know a good surface when we see one. In particular, we can easily see non-local artifacts, such as the terracing described below.

There are many methods for interpolating from isolated data points at isolated test points, available in packages at least since SYMAP. They could be repeated, albeit inefficiently, to interpolate from the sequence of points along a contour line to create a grid. However, that is certainly slow, and probably inappropriate, given the nonuniform distribution of contour points.

In the following, z represents elevation. A *test point* is a point whose z is unknown, but desired. A *data point* has a known elevation.

2 History

Interpolating from contour lines to an elevation array is a classic problem in computational cartography. First, there are the traditional heuristics, such as extending straight lines in eight directions from the test point until they intersect eight contour lines, then interpolating with a weighted average, as described in Jones, Hamilton & Johnson (1986). Similar inverse-distance weighting methods are shown in Watson (1992) and Heine III (1986), often using natural neighbors found by Sibson (1981). These methods can work if the contours are not kidney-shaped.

However, an artifact appears when interpolating a concentric set of circular contour lines, representing a cone. Since the outer contour line are longer than the inner ones, the averaging rule causes the surface between two contour lines to droop, as if pulled down by gravity, in a scalloping or terraced effect. For a thin-plate interpolation, this effect is shown in Figures 4 and 8. This is unacceptable.

Partial differential equations (PDEs) can be used to model a surface subject to certain constraints. Good cartographic introductions to PDEs for interpolation are Tobler (1979), Tobler (1996).

One simple PDE is the Laplacian, or *heat-flow* equation, $z_{xx} + z_{yy} = 0$, where $z_{xx} = \frac{\partial^2 z}{\partial x^2}$ etc. The relevance to heat-flow is that, if, we map elevations to temperatures, assume that the contour lines are fixed at known temperatures, and assume that the surface conducts heat uniformly, then, each point on the surface between the contour lines will eventually equilibrate to some temperature, which we map back to elevation. If this equation is solved by iteration on a grid, then the elevation of each point in the array, whose height is not already fixed, is the average of its four neighbors: $4z_{ij} = z_{i-1,j} + z_{i+1,j} + z_{i,j-1} + z_{i,j+1}$. However, the Laplacian also demonstrates the terracing artifact.

The more complicated *thin plate* model minimizes total curvature, similar to fitting a thin sheet of metal to fixed points, while minimizing the energy of bending. Gaussian curvature, commonly used in geometry, is inappropriate here, since it is zero on a "developable" surface, such as a bent sheet of paper. Instead we use the (scaled) divergence, or $z_{i-1,j} + z_{i+1,j} + z_{i,j-1} + z_{i,j+1} - 4z_{ij}$ for the curvature at a point.

The partial differential equation is: $z_{xx}^2 + 2z_{xy}^2 + z_{yy}^2 = 0$; the iterative equation on a grid is:

$$\begin{aligned} 20z_{ij} &= 8(z_{i-1,j} + z_{i+1,j} + z_{i,j-1} + z_{i,j+1}) \\ &\quad -2(z_{i-1,j-1} + z_{i-1,j+1} + z_{i+1,j-1} + z_{i+1,j+1}) \\ &\quad -(z_{i-2,j} + z_{i+2,j} + z_{i,j-2} + z_{i,j+2}) \end{aligned}$$

With this equation, information flows across the contour lines, which is desirable. This method produces less terracing, but instead demonstrates a *ringing* effect, similar to a Gibbs phenomenon when a square wave curve is being approximated by a Fourier series. Intuitively, the surface tries so hard to minimize the curvature, that, when the data is too nonsmooth, the surface has synthetic oscillations. For example, an interpolation of the desert floor next to a mesa would have this undesirable artifact. Figure 1 shows terracing and ringing schematically. The circles are the data points; the thin lines the interpolated surface.

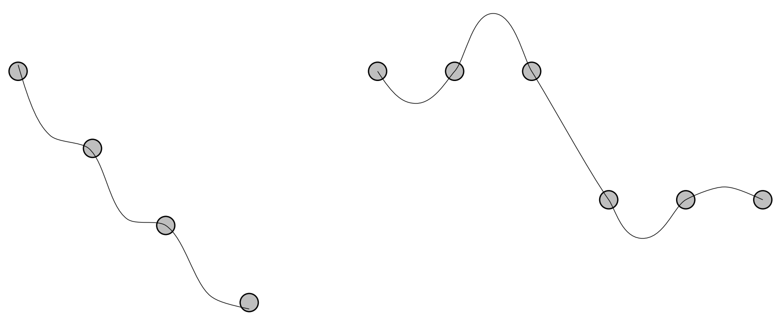

Figure 1: Terracing and Ringing in an Interpolated Surface

There are hundreds of works on thin-plate interpolation, mainly in the mathematical literature; we will merely sample them. The survey paper of Bolle & Vemuri (1991) discusses this, and other, methods. Powell (1997) discusses computational difficulties in interpolating many isolated points and presents an iterative method for up to 10^5 points. Briggs (1974) uses minimum curvature to interpolate contours; Gonzalez-Casanova & Alvarez (1985) improved it. Suter (1992) and Duchon (1976) mention how thin-plate spline interpolation can be expressed thus: $z = \sum_{i=1}^{N} c_i r_i^2 \log r_i + b_1 x + b_2 y + b_3$ where r_i is a distance from a known data point and (x, y) is the test point. Terzopoulos (1988) breaks the thin plate when a discontinuity is inferred. Smith & Wessel (1990) adds tension to the thin plate, to attempt to smooth the Gibbs phenomenon. Sinha & Schunck (1992) uses a 2-stage method: a moving least squares median, followed by a weighted bicubic spline. One program for thin-plate interpolation is *SphereKit* (1996), which can handle up to 500 points per section, using methods described in Franke (1982) and Sandwell (1987). One application of this was blending data sets for GTOPO30(Gesch & Greenlee 1996).

If the input data is not exact, then interpolating it, exactly, may be the wrong operation. Instead, an approximate surface, which passes near the data, might be preferable, since it can have other desirable properties, such as virtually eliminating ringing. This can be realized by assuming that there are springs between the data points and the surface, and minimizing the total energy of the springs and the curvature, as done by Terzopoulos (1988). Unfortunately, the terracing is only reduced.

Explicitly solving a PDE on an $N \times N$ grid, without utilizing the system's sparsity, takes time N^6, which is infeasible for large N. Sparse system solvers and iterative methods are more practical; their time is proportional to the desired accuracy. The best current iterative method is the *multigrid*. It finds an approximate solution on a coarse grid, then improves the solution on a finer grid, with periodic recourse back to the coarser grid for speed. The multigrid technique can solve systems with more than 10000×10000 cells. Terzopoulos (1983) used this method for solving the thin-plate equation. Douglas (1997) points to tutorials, bibliographies, and software.

Voronoi interpolation of the data points is a completely different method, (Gold & Roos 1994). Here, a Voronoi diagram is formed from the data points. The test point is inserted into the diagram, and the areas its Voronoi polygon steals from its neighbors are used to weight the neighbors' elevations. A Hermite transformation of the weights may be used to increase the continuity. Voronoi interpolation solves the problem of artifacts such as surface, slope, or curvatures discontinuities, slope or curvature discontinuities, and and local elevation extrema.

Triangulating the points into a Triangulated Irregular Network, perhaps with a Delauney triangulation, is a related method, (Franklin 1973), (Garcia 1992). Any interpolation method, either bilinear or a higher order spline, may be used inside each triangle. However, sometimes a triangle will have all three vertices on the same contour, causing it to be horizontal, which is undesirable. Christensen (1987) triangulates contours to create new elevation lines between existing contours. The implementation assumes input data properties such as closed contour lines. Rather than depicting the final surfaces, the paper shows the interpolated contours, on small, optimal, data sets.

Because linear interpolation may result in surfaces with flat areas, Watson (1992) blends the technique with a gradient estimation. Huber (1995) detects features such as ridges and valleys, interpolates new elevations on them, and then applies any other method to this enhanced set. This considerably reduces artifacts.

3 New Interpolation/Approximation Methods

We hypothesized and tested various new interpolation/approximation methods, designed with the following properties: the ability to handle realistic amounts of data, minimizing artifacts, such as terracing or ringing, and taking advantage of the data being generally arranged along contour lines. Two successful methods are *intermediate contours* and *gradient lines*.

3.1 Intermediate Contours

We wish to find a new contour in between existing successive contours to introduce new elevations for subsequent processing. It is desirable to find a midpoint between to contours that lies along the steepest slope, which is perpendicular to a given contour.

This method uses successive subdivisions. Between two adjacent contour lines, contour $\mathcal{A}$, at elevation a, and contour $\mathcal{B}$, at elevation b, we interpolate the intermediate contour, with elevation $(a + b)/2$, as follows. Pick a point on $\mathcal{A}$. Find the closest point on $\mathcal{B}$ to it; this approximates the direction of the steepest slope. The midpoint is one point on the intermediate contour.

Successive points on the intermediate contour may be determined. They may not always be adjacent, but that is not a problem because of how they will be used. Next, further, arbitrarily closely-spaced, contours may be computed. Figure 2 shows a section of Crater Lake, while Figure 3 shows the interpolated contours.

Figure 2: Crater Lake Original Contours

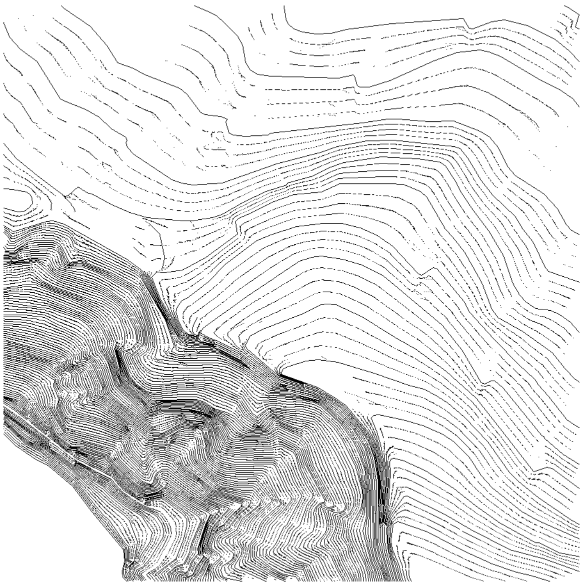

Figure 3: Crater Lake Interpolated Contours

The last phase of the procedure is to complete the surface between the new contours. To reduce the ringing effect, a thin-plate approximation is applied. The resulting surface is better than one produced by the thin-plate approximation alone because the terracing effects are reduced due to the additional intermediate contours.

3.2 Gradient Lines

This method was inspired by the concept of *lofting* in computer aided design. It is a 2-step process. First, we determine *gradient lines* from local maxima to minima on the surface. A gradient line should intersect a contour line perpendicularly. Gradient lines are calculated from the surface produced by another interpolation method, such as the thin-plate. The x-component of the gradient is this:

$$\frac{1}{2}(z_{i+2,j} - z_{i-2,j} + z_{i+1,j-1} - z_{i-1,j-1} + z_{i+1,j+1} - z_{i-1,j+1} - 4z_{i+1,j} - 4z_{i-1,j})$$

The idea is that, while the original interpolation is unacceptable, its gradient lines are good. At this point, each gradient is a 2-D line; we know its elevation only where is crosses contour lines.

The next step is to interpolate elevations along the whole gradient, from the known elevations at the contours. The 1-D interpolation method is a Catmull-Rom (aka Oberhauser) spline, (Faux & Pratt 1981), which has a continuous second derivative. The splines must be smoothed to create the final surface; the thin-plate approximation is applied once again.

4 Experiments

We tested our new approximation algorithms on various sets of USGS DLG contour data. To evaluate the fits, we calculated quantitative parameters such as total squared curvature, average absolute curvature, and root-mean-square error (RMSE). This table tests three approximation methods on the Crater Lake data.

Criterion	*Thin-plate*	*Intermediate Contours*	*Gradient Lines*
Total squared curvature, $\sum\sum(z_{i-1,j}+z_{i+1,j}+z_{i,j-1}+z_{i,j+1}-4z_{ij})^2$	72987	93170	72709
Average absolute curvature, $\frac{1}{(n-2)^2}\sum\sum(\lvert z_{i-1,j}+z_{i+1,j}+z_{i,j-1}+z_{i,j+1}-4z_{ij}\rvert)$	0.138	0.118	0.107
Root-mean-square error relative to DEM, $\sqrt{\frac{1}{n^2}\sum_{i=1}^{n^2}(z_i-w_i)^2}$, where w_i is a DEM point.	8.69	5.28	5.48

This shows that the intermediate contours surface has a larger squared curvature, but the gradient lines method has a slightly lower squared curvature. Both of our methods exhibit better average absolute curvature. The RMSEs of our methods are significantly better than the thin-plate approximation surfaces and are well within USGS Level 3 standards (the Crater Lake contour interval is 40). However, all these criteria are local properties of the surface, and do not measure artifacts such as terracing and ringing. For these, visual inspection seems the best method to date. Therefore, we present images of the surfaces interpolated by the various methods, for visual comparison.

The first test data set is a DLG of Crater Lake, Oregon, with contour lines as shown in Figure 2. Figure 4 shows a thin-plate approximation, while Figure 5 shows our interpolated contour algorithm, and Figure 6 our gradient method.

The second test case is Mount Washington, New Hampshire, Figure 7. Figure 8 shows a thin-plate approximated surface. Figure 9 shows some intermediate interpolated contours, while Figure 10 shows the resulting approximated surface. Figure 11 shows the approximated surface by the gradient method.

In both cases, our intermediate contour method is better than the thin-plate method, and our gradient method is the same or better still.

5 Untried Methods

One attractive method not yet implemented centers on the idea of a *conformal map*, a transformation of the plane that preserves angles. However, it bends straight lines into curves, and changes distances. Any closed curve, even a polygon, can be conformally mapped into a circle. A system of two closed curves, one inside the other, can be mapped into an annulus, or a ring between two concentric circles. We know how to interpolate a surface between an annulus. Therefore, the idea is to map two concentric contour lines to an annulus, interpolate, and inverse map the result. The problem is that calculating the desired conformal map is too hard. In addition, a new rule would be needed for a contour line containing *two* interior contour lines. Nevertheless, this method looks interesting. It might interpolate kidney-shaped contours better than many other methods.

A second untried method is the use of an MLE. We can reject an interpolated surface that has terraces at the contour. Although a real surface might have these features, it is very unlikely to have them exactly at the contours. Can this idea be generalized to select, among all the surfaces interpolating the contours, that one with the highest "probability?" This remains an open question.

6 Conclusion

Both the intermediate contours and gradient lines method produce surfaces with much reduced terracing compared to thin-plate techniques. The RMSE for both surfaces is also better using the new methods. In general, the intermediate contours method works well when contours are widely spaced. The gradient lines method proved better suited to contours that are moderately spaced, due to the step of finding an approximate surface from which to derive gradients.

References

Bolle, R. M. & Vemuri, B. C. (1991), 'On three-dimensional surface reconstruction methods', *IEEE Transactions on Pattern Analysis and Machine Intelligence* **13**(1), 1–13.

Briggs, I. (1974), 'Machine contouring using minimum curvature', *Geophysics* **39**(1), 39–48.

Christensen, A. H. J. (1987), Fitting a triangulation to contour lines, *in* N. R. Chrisman, ed., 'Auto–Carto 8:Eighth International Sumposium on Computer-Assisted Cartography', pp. 57–67.

Douglas, C. C. (1997), 'MGNet', `http://www.mgnet.org/`.

Duchon, J. (1976), Splines minimizing rotation invariant semi-norms in sobolev spaces, *in* W. Schempp & K. Zeller, eds, 'Constructive Theory of Functions of Several Variables', Springer-Verlag, pp. 85–100.

Faux, I. & Pratt, M. (1981), *Computational geometry for design and manufacture*, Mathematics and its applications, Halsted Press.

Franke, R. (1982), 'Smooth interpolation of scattered data by local thin plate splines', *Computing and Mathematics with Applications* **8**, 273–281.

Franklin, W. R. (1973), 'Triangulated irregular network program', `ftp://ftp.cs.rpi.edu/pub/franklin/tin73.tar.gz`.

Garcia, Antonio Bello, e. a. (1992), A contour line based triangulating algorithm, *in* P. Bresnahan, E. Corwin & D. Cowen, eds, 'Proceedings of the 5th International Symposium on Spatial Data Handling', Vol. 2, IGU Commission of GIS, Charleston SC USA, pp. 411–423.

Gesch, D. & Greenlee, S. (1996), 'GTOPO30 documentation', `http://edcwww.cr.usgs.gov/landdaac/gtopo30/README.html`.

Gold, C. M. & Roos, T. (1994), Surface modelling with guaranteed consistency — an object-based approach, *in* T. Roos, ed., 'IGIS'94: Geographic Information Systems', Lecture Notes in Computer Science No. 884, Springer-Verlag, pp. 70–87. `http://www.gmt.ulaval.ca/homepages/gold/papers/ascona.html`.

Gonzalez-Casanova, P. & Alvarez, R. (1985), 'Splines in geophysics', *Geophysics* **50**(12), 2831–2848.

Heine III, G. W. (1986), 'A controlled study of some two-dimensional interpolation methods', *Computer Oriented Geological Society Computer Contributions* **2**(2), 60–72.

Huber, M. (1995), Contour-to-dem a new algorithm for contour line interpolation, *in* 'Proceedings of the JEC '95 on GIS'. `http://lbdsun.epfl.ch/~huber/ctdjec.html`.

Jones, T. A., Hamilton, D. E. & Johnson, C. R. (1986), *Contouring Geologic Surfaces with the Computer*, Van Norstand Reinhold Company Inc., New York.

Powell, M. (1997), Thin plate spline interpolation in two dimensions, *in* 'CTAC97, The eighth biennial Computational Techniques and Applications Conference', Computational Mathematics Group of Australian and New Zealand Industrial and Applied Mathematics Society (ANZIAM), a Division of the Australian Mathematical Society.

Sandwell, D. T. (1987), 'Biharmonic spline interpolation of GEOS-3 and SEASAT altimeter data', *Geophys. Res. Lett.* **14**, 139–142.

Sibson, R. (1981), A brief description of natural neighbor interpolation, *in* Barnet, ed., 'Interpreting Multivariate Data', Wiley, pp. 21–36.

Sinha, S. S. & Schunck, B. G. (1992), 'A two-stage algorithm for discontinuity-preserving surface reconstruction', *IEEE Transactions on Pattern Analysis and Machine Intelligence* **14**(1), 36–57.

Smith, W. H. F. & Wessel, P. (1990), 'Gridding with continuous curvature splines in tension', *Geophysics* **55**(3), 293–305.

SphereKit (1996), `http://ncgia.ucsb.edu/pubs/spherekit`. National Center for Cartographic Information and Analysis.

Suter, D. (1992), Fast evaluation of multipoles without multipoles, Technical Report MCESE-92-1, Monash University, Clayton 3168, Australia.

Terzopoulos, D. (1983), 'Multilevel computational processes for visual surface reconstruction', *Computer Vision, Graphics, and Image Processing* **24**, 52–96.

Terzopoulos, D. (1988), 'The computation of visible-surface representations', *IEEE Transactions on Pattern Analysis and Machine Intelligence* **10**(4), 417–438.

Tobler, W. (1979), 'Smooth pycnophylatic interpolation for geographic regions', *J. Am. Stat. Assn.* **74**(367), 519–536.

Tobler, W. (1996), 'Converting administrative data to a continuous field on a sphere', http://www.sbg.ac.at/geo/idrisi/GIS_Environmental_Modeling/sf_papers/tobler_waldo/tobler_waldo.html.

Watson, D. (1992), *Contouring: A Guide to the Analysis and Display of Spatial Data*, Pergammon Press.

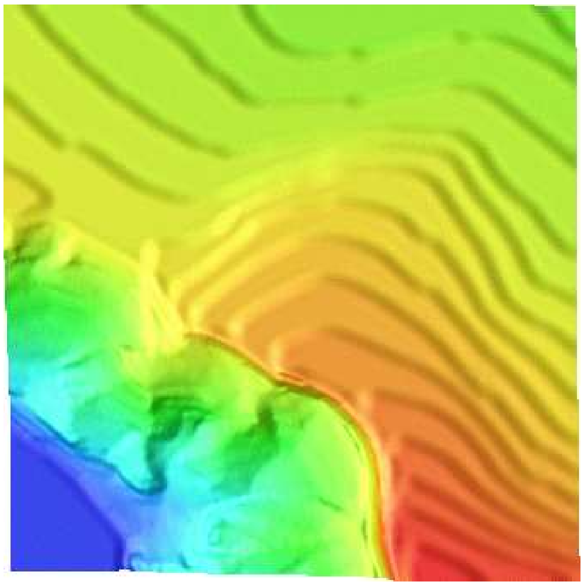

Figure 4: Crater Lake Thin-Plate Approximation

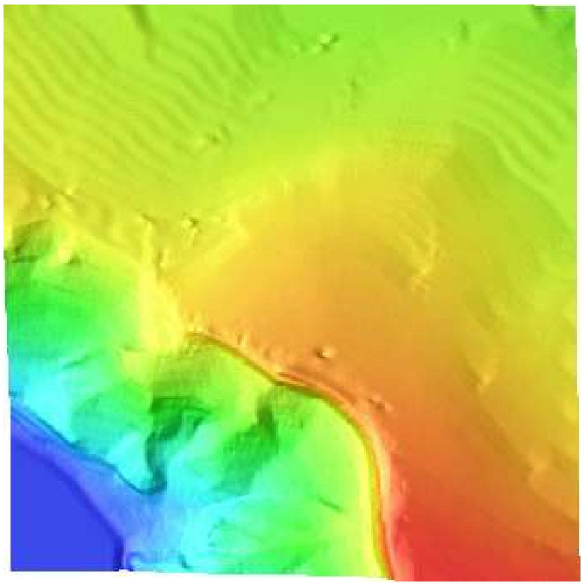

Figure 5: Crater Lake Intermediate Contour Approximation

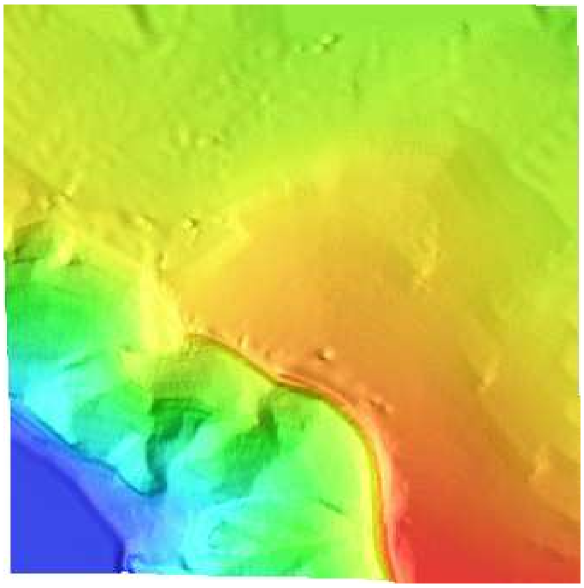

Figure 6: Crater Lake Gradient Line Approx.

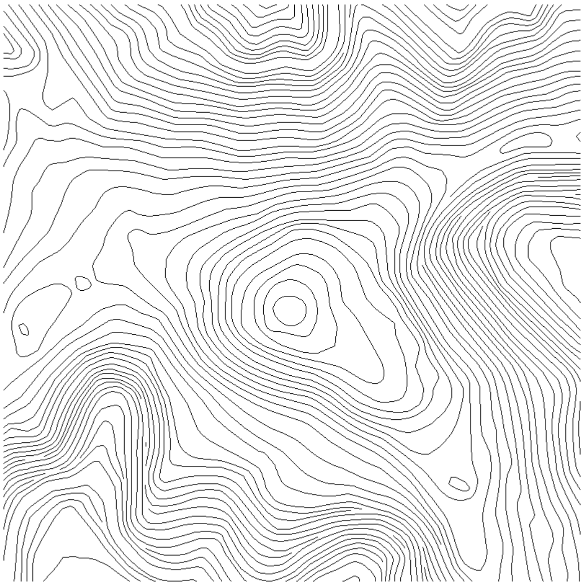

Figure 7: Mt. Washington Contours

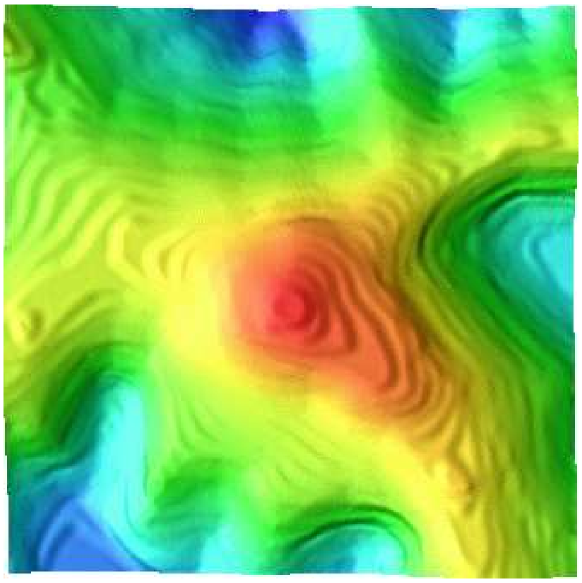

Figure 8: Mt. Washington Thin-Plate Approximation

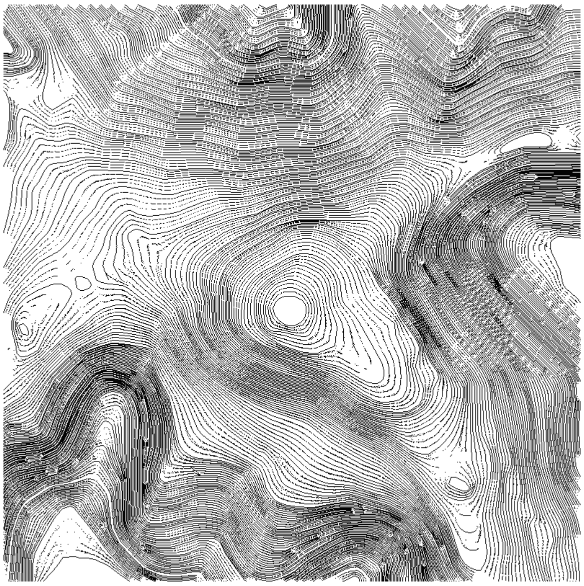

Figure 9: Mt. Washington Intermediate Contours

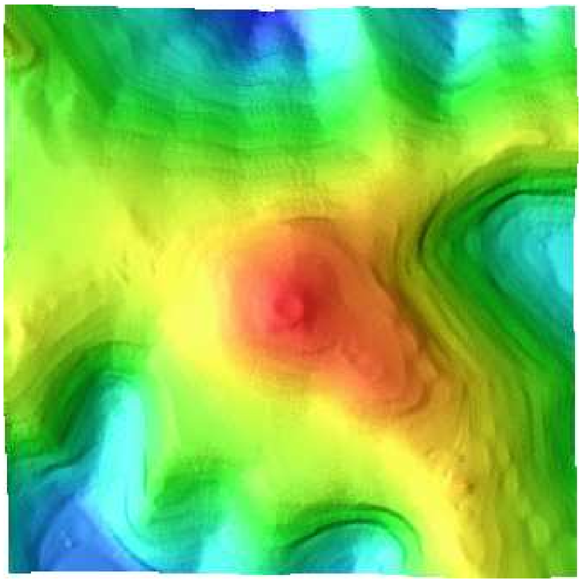

Figure 10: Mt. Washington Intermediate Contour Approximation

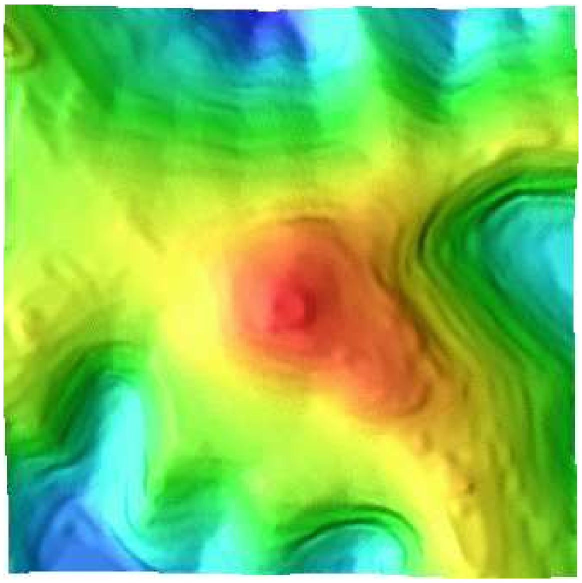

Figure 11: Mt. Washington Gradient Line Approximation