Automated Quantification of Gradient Defined Features

T. Fedenczuk[1], P. Fryer[1], J. Escartin[2], A.J.L. Harris[1]

[1] SOEST - HIGP, University of Hawaii, Honolulu, USA.
1680 East-West Rd POST504, Honolulu HI 96822, USA (tmf@hawaii.edu)
[2] Equipe de Géosciences Marines, Institut de Physique du Globe, Paris, France.

***Abstract* - We present the summary and application of a new MATLAB/GIS technique for the quantification of gradient defined features in submarine environments. The technique utilizes MATLAB scripts to convert bathymetry data into a gradient dataset, produce gradient maps, and most importantly, automate the process of defining and characterizing gradient defined features such as flows, faults, landslide scarps, folds, valleys, and ridges. The features are defined according to strict gradient threshold criteria that are quantifiable and reproducible. The technique also calculates volumes of features with irregular surface boundaries, as well as other calculations such as vertical and horizontal lengths, underlying slopes, slope corrected distal edge thickness, yield strength, etc. The technique can also be used in non-topographic applications that have gradient data such as temperature gradients, nutrient densities, etc.**

I. INTRODUCTION

The forearc portion of the Izu-Bonin/Mariana (IBM) convergent plate margin hosts a series of large serpentinite mud volcanoes (Fig. 1). One of the largest of these active mud volcanoes is Big Blue seamount. Its summit and flanks were surveyed using high resolution bathymetric (EM300) and sidescan (DSL-120) data sets. The seamount lies at a depth of ~1240 m, with a relief of more than 2 km above the surrounding ocean floor, and a maximum width of ~40 km. Along the southwest flank, at least four flows (ranging from simple to compound) are well preserved and relatively undisturbed (ABCD in Fig. 2). We analyzed the three, top-most, flows. We used flow D (the lowest flow field along the flank), as well as other surrounding features, to help complete the underlying slope analysis.

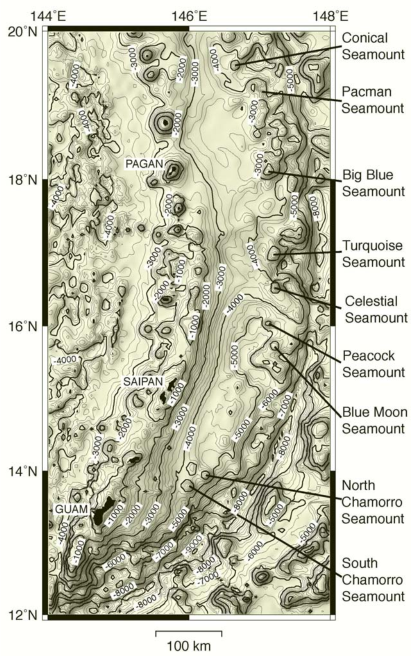

Fig. 1 Bathymetry map of Mariana Forearc, showing Big Blue Seamount relative to other seamounts, as well as the islands of Guam, Saipan, and Pagan.

II. METHODS

All transformations and computations were made using MATLAB scripts and functions written by the authors. Geographic information systems (GIS) were used for visualization and digitization.

In MATLAB, the algorithm imports XYZ bathymetry data and processes the bathymetry into a gradient data set (1) that includes latitude, longitude, slope in the *x* direction, slope in the *y* direction, and a combined slope magnitude (2).

The gradient (∇z) at point (x_0, y_0) is defined by:

$$\nabla z\,(x_0, y_0) = (z_x(x_0, y_0),\, z_y(x_0, y_0)) \qquad (1)$$

where *z* is the bathymetry value (or height for DEMs); z_x and z_y and denotes differentiation with respect to *x* (latitude) and *y* (longitude), respectively. The magnitude (M) of the gradient is then calculated at each point as:

$$M(x_0, y_0) = \| \nabla z(x_0, y_0) \| = \sqrt{z_x(x_0, y_0)^2 + z_y(x_0, y_0)^2} \qquad (2)$$

It then creates gradient maps using color to display the magnitude of the gradient (Fig. 2 and 3), and creates normalized vector arrows that can be overlaid to indicate the direction of the gradient (Fig. 3).

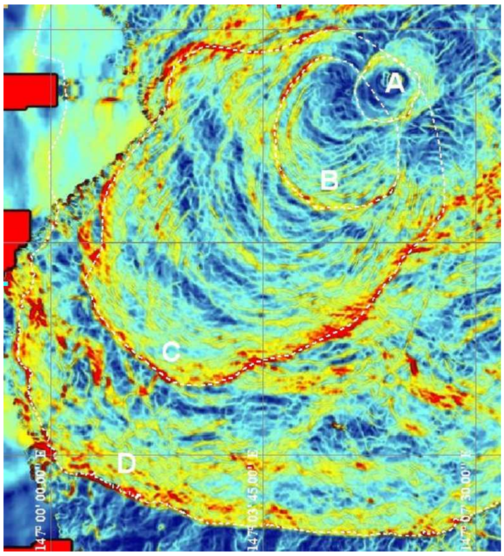

Fig. 2 Gradient map for Big Blue mud volcano, flow fields A through D. Color bar is optimized for relevant slope range. Dark blue represents a slope of 0° and red represent slopes > 20°, with green and yellow gradations in between.

At this stage, the generated gradient map is imported into GIS and gradient defined features are identified and digitized as shapefile lines, and exported back into MATLAB. In this application, we identified four main flow fields (A, B, C, D) (Fig. 2).

Next, the MATLAB algorithm automatically samples a gradient feature (such as a flow's distal edge) at discrete, user-defined intervals, and calculates the trend of the maximum gradient at each sample point to ensure that the profile direction is accurate. It creates profiles based on a user-defined, minimum-gradient threshold, and plots these profiles on the gradient map for error checking in GIS. Next, the algorithm calculates the horizontal and vertical length for each profile, and exports the profiles into MATLAB workspace files (.mat), and ASCII files (for import into GIS). The algorithm can be customized to measure other characteristics such as profile trend, slope, etc.. It then determines the underlying slope of each flow using two customized methods for two separate sets of calculations (volume and rheology), and calculates the slope corrected distal edge thickness for each point for use in yield strength calculations.

Assuming that the slope corrected thickness, t, at which the flow came to rest represents the critical thickness, t_c, (3) can be solved to calculate yield strength using density (ρ), gravity (g) and underlying slope (θ):

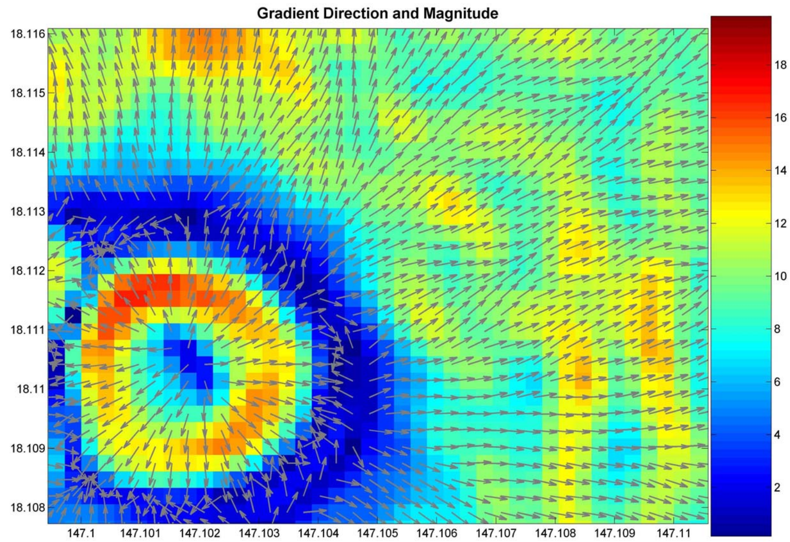

Fig. 3 Example of a gradient color map, overlaid by normalized, reverse-gradient vector arrows

$$Sy = t_c \rho g \sin\theta \qquad (3)$$

The algorithm then calculates the minimum, maximum, mean, and median of distal edge thickness for each feature, and creates a report with a summary of results. Next, it calculates the volume of each flow as defined by its perimeter thickness. To do this, we assume that the thickness within the boundaries of the flow satisfies Laplace's equation:

$$\nabla^2 t = \frac{\partial^2 t}{\partial x^2} + \frac{\partial^2 t}{\partial y^2} = 0 \qquad (4)$$

We use Jacobi iteration (5) to solve Laplace's equation numerically for thickness inside the flow, given values along the boundary. We create a mesh grid of an irregular boundary area, assign the boundary values to the closest grid points, and assign all of the points inside the boundary with an initial value of zero. Then, for each mesh-grid point inside of the boundary, the function successively solves:

$$t_{j,k}^{(n+1)} = \frac{1}{4}[t_{j-1,k}^{(n)} + t_{j+1,k}^{(n)} + t_{j,k-1}^{(n)} + t_{j,k+1}^{(n)}] \tag{5}$$

Where $t_{j,k}$ is thickness at longitude (j) and latitude (k), and n is the iteration number [1].

The automated quantification technique described above has several benefits. The user defined thresholds and criteria are very precise (e.g. a slope of 12 degrees at a sample interval of 10m) and can be calculated consistently throughout all of the samples. The number of samples can be increased by a magnitude of ten or more; this helps verify the measurements and give them greater resolution. Also, the entire extent of the feature is sampled evenly at the regular intervals and omits the human predisposition for rendering prominent landforms that are easier to digitize but are not proportionally representative of the feature as a whole. Automating this technique makes it much faster than the manual method, permitting thresholds to be changed quickly and the boundaries redrawn, thus allowing for fast comparison, error correction and optimization. The results are also reproducible, allowing other researchers to verify each other's work.

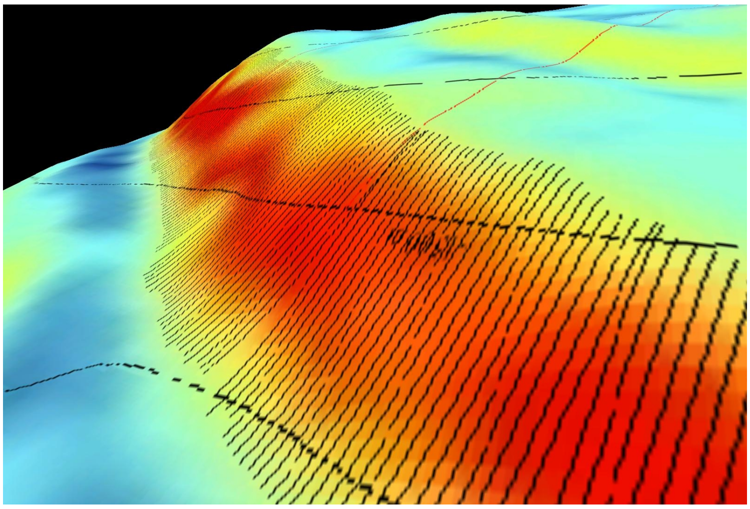

Fig. 4 A 3-D perspective of a distal edge with profile lines extended to a threshold of 13°.

III. RESULTS

At a sample interval of 10m, the technique produced 303 profiles for flow A, 982 for flow B, and 2931 profiles for flow C. The profile boundaries were set consistently at an interpolated gradient threshold of 13°. For each profile, horizontal and vertical distance, underlying slope, and slope corrected thickness were calculated.

Mean bulk density (mbr) data from previous work done on similar volcanoes in the area [2] [3], was used to constrain the upper and lower density limits in yield strength calculations. Density-constrained yield strength values were calculated.

Volumes were calculated using Jacobi iteration (a numerical solution to Laplace’s equation) with a grid spacing of 10m and 1000 iterations. The calculated volumes increased from flow A to B and again from B to C. We compared our results to a simpler volume estimation that calculates the mean and median distal edge thickness and then multiplies it by the surface area [4] [5].

IV. CONCLUSION

The automated morphology-quantification method presented in this paper has enabled us to calculate the thickness of the distal edges of three mudflows under three different underlying slope conditions using more than 5000 sample points each. All calculations are repeatable and use the same threshold criteria. The most important advantage to this quantification method is that we can conduct the same calculations using different parameters rapidly. We can also edit or append the MATLAB functions and scripts for other applications.

ACKNOWLEDGMENT

The application of this research has been funded by the **NSF-Margins** program, grant number OCE-0002584, and by the **Maui High Performance Computing Center.**

REFERENCES

[1] Strauss, W. A. , 1992, *Partial Differential Equations*, John Wiley & Sons, New York. pp. 208-210.
[2] Fryer, P., Pearce, J.A., Stokking, L.B., et al., 1992, *Proc. ODP, Sci. Results,* 125:593-614, College Station, TX, .
[3] Salisbury, M.H., Shinohara, M., Richter, C., et al., 2002. *Proc. ODP, Init. Repts.,* 195, College Station TX.
[4] Rowland, S. K., 1996, Slopes, lava flow volumes, and vent distributions on volcan Fernandina, Galapagos islands, J. Geophys. Res., 101, 27,657-27,672.
[5] Stevens, N.F., Wadge, G. & Murray, J.B., 1999, Lava flow volume and morphology from digitised contour maps. *Geomorphology*, 28, 251-261.