

VisTRE: A Visualization Tool to Evaluate Errors in Terrain Representation

Christopher G. Healey
Computer Science Department
North Carolina State University
Raleigh, NC 27695–8206, USA

Jack Snoeyink
Computer Science Department
UNC Chapel Hill
Chapel Hill, NC 27599–3175, USA

Abstract

New data sources and sensors bring new possibilities for terrain representations, and new types of characteristic errors. We develop a system to visualize and compare terrain representations and the errors they produce.

1. Introduction

New sensors and improved computational capabilities allow us to collect huge amounts of sample data on which to build terrain models for a broad range of applications: photogrammetric correction, flood modeling, beach monitoring, visibility simulation, urban and natural resource management. In fact, new data comes so fast that it is hard to determine how its errors manifest. Daniel and Tennant [1] report that “technology has advanced and indeed outpaced quality assessment developments.”

In this paper we develop a system to visualize error on terrain models so that we can compare different terrain representations, different data sources, and different levels of detail, both numerically and visually. Our system uses concepts from perceptual visualization to choose mappings of data attributes to visual features in an attempt to help a developer answer the question of what is the best terrain representation, given resource limits, for a particular application. This first prototype has already been helpful in revealing correlations between errors and showing systematic error that would be hidden in a numeric assessment of error statistics.

1.1. Motivation for Visualizing Terrain Error

The methods used to evaluate error in terrain models are either indirect or coarse. Sophisticated engineering quality assessment is applied to bound the error introduced by the equipment that obtains the data, but the result is usually encoded in a single assessment number. Statistical analysis is applied to compare elevation data to “ground truth” at selected points, but this does not capture the correlation of error or its effects on different applications. Those who work with data from a particular sensor can tell interesting stories of its characteristic errors: e.g., how the old mechanical stereo plotters lead to washboard surfaces as operators undershoot and overshoot as the cursor flies back and forth

across the terrain, and how the recently-released data from the shuttle radar topography mission (SRTM) flattens ridges and valleys, and has data voids or 25 meter jumps in certain types of terrain.

These new sources of data also inspire new research into data representations, which bring more opportunities for error. In our case, we are particularly interested in new representations for which the resolution or level of detail may differ at different spatial locations. The TIN (Triangulated Irregular Network [2]) was proposed in 1977 for this purpose, and many other techniques (based quadtrees, splines, wavelets, ...) have been proposed since. Nevertheless, raster DEMs (Digital Elevation Models) remain the most common terrain representation because they make the data fit the computation. Raw data is never in raster DEM form, however, and it is difficult to determine what errors are introduced by the various processing and resampling operations that convert raw data into a raster DEM.

Given the errors that can be introduced by sensors, data processing, and data representations, what is the best type of terrain model for a given application? Unfortunately, we already encounter problems with the premise of this question; the error is there, but it is difficult to model and measure. One cannot get a single number, either from engineering quality assessment or from statistical analysis, that will show the effects of error on different applications.

And so much depends on the application. The capabilities of the application platform will determine memory limits, and the application itself will determine data resolution (since different geographical features are defined at different resolutions). Even the preference of the person running the application plays a role: Kumler’s monograph in 1994 addressed these questions for TINs and raster DEMs [3] and concluded that except for visualization DEMs were better, for a given amount of memory, but the author still liked TINs. (Actually, advances in compression [4, 5] mean that Kumler’s questions need to be asked again.)

1.2. System Foundations

Before we survey previous work, we take a moment lay the foundations for our system for perceptual-based visualization of terrain error. Our system has two types of modules: *surface modules* each represent a continuous terrain

surface derived from data. A surface module must report *surface attributes* when queried at *measurement locations*. Each surface module has attributes of elevation and resolution, and may register others such as slope, aspect, or confidence. Most surface modules have a level-of-detail control value that may be set by the user; this value may reflect the fraction of data used, the number of wavelet coefficients, or allowed deviation from data samples.

The *error visualization module* requests attributes at chosen measurement locations (from one or more surface modules, at one or more levels of detail). One set of returned attributes is deemed *ground truth*—often the user chooses the measurement locations as the most accurate data known for the ground truth surface. It chooses, with user interaction, a perceptual-based mapping from attributes, or their deviation from ground truth, to visual features (*e.g.* hue, luminance, size, orientation). It manages which measurement locations are shown to the user to give the correct impression of the error at the scale that the user is viewing, respecting the resolution of the underlying model. By simultaneously showing the results from two or more surface modules or levels of detail at the same measurement locations, the error visualization module permits the user to compare terrain representations and their errors.

Surface modules that are compared together must use the same coordinate system and earth datum. All images we show are using UTM coordinates on a WGS'84 datum [6], but lat/long or state plane coordinates could be used instead. Coordinate conversions and attribute derivation modules (such as slope from elevation samples) are available, but must be called for explicitly so that the user is aware that these conversions may be a possible source of error.

The application means that it is important to keep resolution in mind in this system: geographic features have a natural range of scales at which they can be defined—the difference between a pebble, a bolder and Yosemite's Half-Dome is a matter of scale.

Many of our visualization techniques apply not only to terrains, but also to general surfaces. We concentrate on terrains because our goal is to evaluate different representations' capabilities to support terrain analysis applications. For terrain users, surface topology is unimportant and registration is (almost) already solved by the standard coordinate systems provided with geocoded data. On the other hand, terrain data sets are huge, and come with interesting error characteristics from a wide variety of sensors. Our system can take advantage of the geocoding to unify and compare a diverse set of representations: including raster DEM, TIN, subdivision, spline, and wavelet surfaces.

A key goal of visualization is to present raw data in an *unambiguous* manner. It is critical that a visualization does not hide important details, or worse, introduce false information into an image. Consider a simple example: the rain-

bow color scale (purple, blue, cyan, green, yellow, orange, red) commonly used to encode continuous scalar data, for example, temperatures on a weather map. It is well known by visualization experts that this scale produces the illusion of visual discontinuities between neighboring color regions (*e.g.* between purple and blue, yellow and orange, and so on). When continuous data are visualized with the rainbow scale, users often conclude that values on either side of a visual boundary have a large difference. In fact, this is not true. The color scale has introduced false information into the resulting visualization¹. This is exactly the type of problem we want to avoid.

One way to manage and control how a visualization is interpreted is through an understanding of how the human visual system perceives basic properties of color, texture, and motion that appear in an image. We use results from psychophysical experiments to choose visual features that preserve patterns and relationships in the raw data, that avoid visual ambiguity, and that ensure an effective presentation of the data.

2. Previous Work

Our visualization prototype draws on previous work in uncertainty visualization, terrain visualization, multidimensional visualization, and properties of low-level human vision to build perceptually salient representations of multidimensional data.

2.1. Visualizing Uncertainty

As a first step towards understanding how best to visually represent uncertainty, confidence, or errors, researchers have studied different ways to model uncertainty. Gershon enumerated sources of error or uncertainty and described corresponding techniques to represent each, with special emphasis on examples of inappropriate presentation [7]. MacEachren et al. provide an excellent overview of recent research on uncertainty, demonstrating different methods to visualize uncertain data values. They propose seven interdisciplinary challenges that must be met to further advance our ability to manage, understand, and represent uncertainty [8].

Researchers have explored many methods to visualize the uncertainty associated with individual sample points or data values. For example, Wittenbrink et al. [9] used glyphs to display uncertainty in flow fields. Pang et al. demonstrated a comprehensive set of methods, including glyphs, geometry modification, sonification, animation, and psychophysical cues, in representative examples of applica-

¹Note that the visual boundaries in the rainbow scale are not problematic if they are managed correctly, for example, if they are used to subdivide continuous data into categories as is done in weather maps (*e.g.* regions of cold, cool, moderate, warm, hot, etc.)

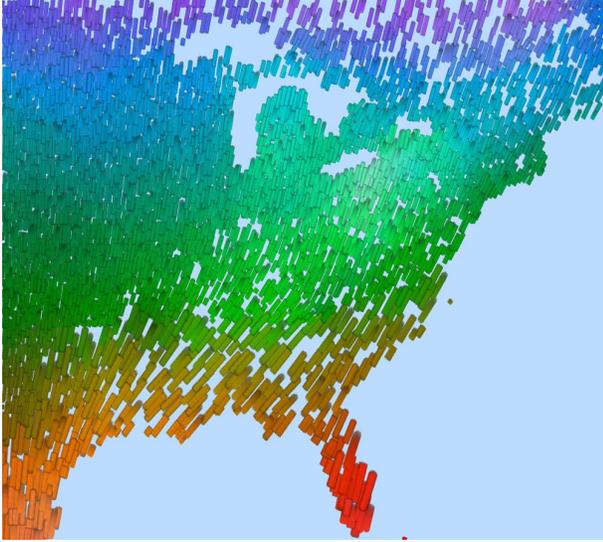


Figure 1: An example multidimensional visualization of historical weather conditions for January over the eastern United States; color represents temperature (purple and blue for cold to orange and red for hot), luminance represents cloud coverage (brighter for denser clouds), size represents surface pressure (larger strokes for higher pressure), orientation represents precipitation (rightward tilt for heavier rainfall), and density represents wind speed (denser packing of strokes for stronger winds)

tions [10]. Johnson and Sanderson emphasize the importance of including uncertainty in visualization, and propose a number of techniques to encode error and uncertainty values [11]. Grigoryan and Rheingans apply geometry modification to 3D surfaces—rendering surfaces as collections of points, each displaced by an amount proportional to its uncertainty in the surface normal direction [12].

Researchers have also studied ways to visualize multidimensional data, and since visualizing multidimensional data requires effective methods to present several attribute values at a common spatial location, the same methods can display both data and associated uncertainty values for individual sample points in a dataset. Methods to visualize multidimensional data include using glyphs that represent separate data attributes by varying separate properties of color, texture, and motion [13, 14], or using one visual feature (*e.g.* different orientations of a line glyph) to encode different values of an attribute, and a second visual feature (*e.g.* luminance) to represent the amount of the attribute present at any given spatial location [15]. Results from multidimensional visualization may offer important insights on how to best represent data and errors together in a common display.

2.2. Terrain Visualization

Terrain and height field visualization in particular, and mesh simplification and rendering in general, are important and well-researched areas in visualization and computer graph-

ics. We touch briefly on methods most related to our own goals and system design.

Terrain models are large, so much of the research on terrain visualization is to develop data structures that are efficient to render and methods to ensure that large, high-detail terrains display at interactive frame rates. Real-time simplification and visibility culling reduce the polygon count during rendering: Examples include the real-time adaptive meshes of Duchaineau *et al.* [16] and the nested grid caching scheme of Losasso and Hoppe [17]. Out-of-core visualization methods manage the large amount of data associated with high-detail terrains [18, 19].

Data can be visualized on top of the surface, once an underlying surface model is defined. Common ways to present data values include color coding (*e.g.* the well-known rainbow color scale used to represent temperature in weather maps), shading (*e.g.* surface normals and virtual lights produce a shaded relief of a terrain’s elevation features), and glyphs (*e.g.* contour lines for topography, or directed arrows to represent wind direction and speed). Again, the various methods used to map data to locations on the terrain are typically based on techniques for visualizing single-valued or multidimensional data (*e.g.* as demonstrated for meteorological data over maps [14]).

2.3. Perception

Our visualizations are constructed from psychophysical studies of how the human visual system “sees” fundamental visual properties in an image. The use of color, texture, and motion has a long history in the graphics, vision, and visualization literature (*e.g.*, in Treinish’s meteorological visualizations [20], see also Figure 1).

Examples of simple color scales include the rainbow spectrum, red-blue or red-green ramps, and the grey-red saturation scale. More sophisticated methods divide color along dimensions like luminance, hue, and saturation to better control the difference viewers perceive between different colors. Researchers in visualization have combined perceptually balanced color models with non-linear mappings to emphasize changes across specific parts of an attribute’s domain, and have also proposed automatic colormap selection algorithms based on an attribute’s spatial frequency, continuous or discrete nature, and the analysis tasks to be performed. Experiments have shown that color distance, linear separation, and color category must all be controlled to select discrete collections of distinguishable colors [13].

Texture is often viewed as a single visual feature. Like color, however, it can be decomposed into a collection of fundamental perceptual dimensions. One promising approach in visualization has been to use perceptual texture dimensions to represent multiple data attributes. Individual values of an attribute control its corresponding texture dimension. The result is a texture pattern that

changes its visual appearance based on data in the underlying dataset. Examples of perceptual dimensions include properties like size, density, orientation, and regularity of placement [13, 15].

Motion patterns are only beginning to be understood. They can be processed very rapidly by the low-level visual system [21]. Perceptual dimensions of motion like flicker, direction, and velocity have been studied in the psychophysical literature, and are now being used for notification in real-time systems [22], for cognitive grouping of elements [23], and for visualizing multiple independent data attributes.

Our visualization designs choose visual features that are highly salient, both in isolation and in combination. We map the features to individual data attributes in ways that draw a viewer’s focus of attention to important areas in a visualization. The ability to harness the low-level human visual system is attractive, since:

- high-level exploration and analysis tasks are rapid and accurate, usually requiring 200 milliseconds or less to complete,
- analysis is display size insensitive, so the time to perform a task is independent of the number of elements in the display, and
- different features can interact with one another to mask information; psychophysical experiments allow us to identify and avoid these visual interference patterns.

We have combined basic techniques from uncertainty, terrain, and multidimensional visualization with experimental results on perception to construct a software tool designed to visualize different types of error over an underlying terrain height field. Users control which visual feature to apply (hue, luminance, size, or velocity of motion) to represent different error types. We visualize error values in ways that maintain perceptual salience. For example, both the hue and luminance scales are built to ensure perceived balance, that is, equal-distance steps at different positions along the scale produce roughly equal perceived differences in the resulting hues or luminances. Sizes and velocities are also selected based on experimental results from visualization and psychophysics.

3. VisTRE

We designed a prototype visualization tool, VisTRE, to visualize terrain representations and their errors. Each terrain model records x , y , z -elevation and slope (stored as a unit-length surface normal) for a sequence of sample points. Two error measurements are also attached to each sample: elevation error, the difference in a sample point’s elevation compared to a ground truth terrain, and slope error, the ab-

solute angular difference between a sample point’s surface normal compared to a ground truth terrain.

We chose a simple point cloud visualization scheme for our initial system. Sample points are represented as square glyphs (*i.e.* variable-sized OpenGL points). Each glyph is positioned at its (x, y) location, with its z -position and surface normal defined by the sample’s elevation and slope. Simulated lighting is used to produce a shaded relief of the map’s underlying topology (Figure 2). Error values are displayed as variations in each sample point’s hue, luminance, size, or velocity of motion. Values for each visual feature were selected to maintain perceptual distinguishability and balance, specifically:

Hue: Hues range from blue for the lowest error values through green, yellow, orange, and red for highest error. Positions along the hue scale are mapped through a perceptually balanced color model [24] to ensure equal-distance steps produce roughly equal changes in perceived hue.

Luminance: Luminances range from approximately 40% of the monitor’s available brightness (to ensure glyphs are not too dark to distinguish differences in hue) to approximate 80% of available brightness (to ensure glyphs are not so bright that they produce highly unsaturated, pastel-like hues). Again, a perceptual color model is used to ensure balance between differences in error values and perceived differences in luminance.

Size: Sizes range from one unit to 2×2 , 3×3 , 4×4 , and 6×6 arrays. Experimental results from psychophysics suggest that area should double to generate a uniform perceived change in size. Array differences were therefore selected to approximately double successive areas, except for the change from one unit to 2×2 units, which is a four-fold increase. In practice, however, we have observed no negative effects from the over-emphasis of this size difference.

Velocity: Non-zero velocity moves a sample point’s glyph along a motion path parallel to the z -axis. The path extends up to nineteen units in length, centered about the sample’s initial position. All glyphs traverse their full path over a fixed time interval (*i.e.* glyphs with longer paths move with a higher velocity). Differences in path length were selected to approximately double successive velocities, again based on results that suggest this relationship is needed to produce uniform perceived differences in velocity [25, 26].

Some visual features (e.g. different values of hue and luminance) are selected using monitor properties like maximum luminance and triad chromaticity. Perceptually balanced color models and calibrated conversions to a monitor’s RGB space improve our ability to encode quantitative

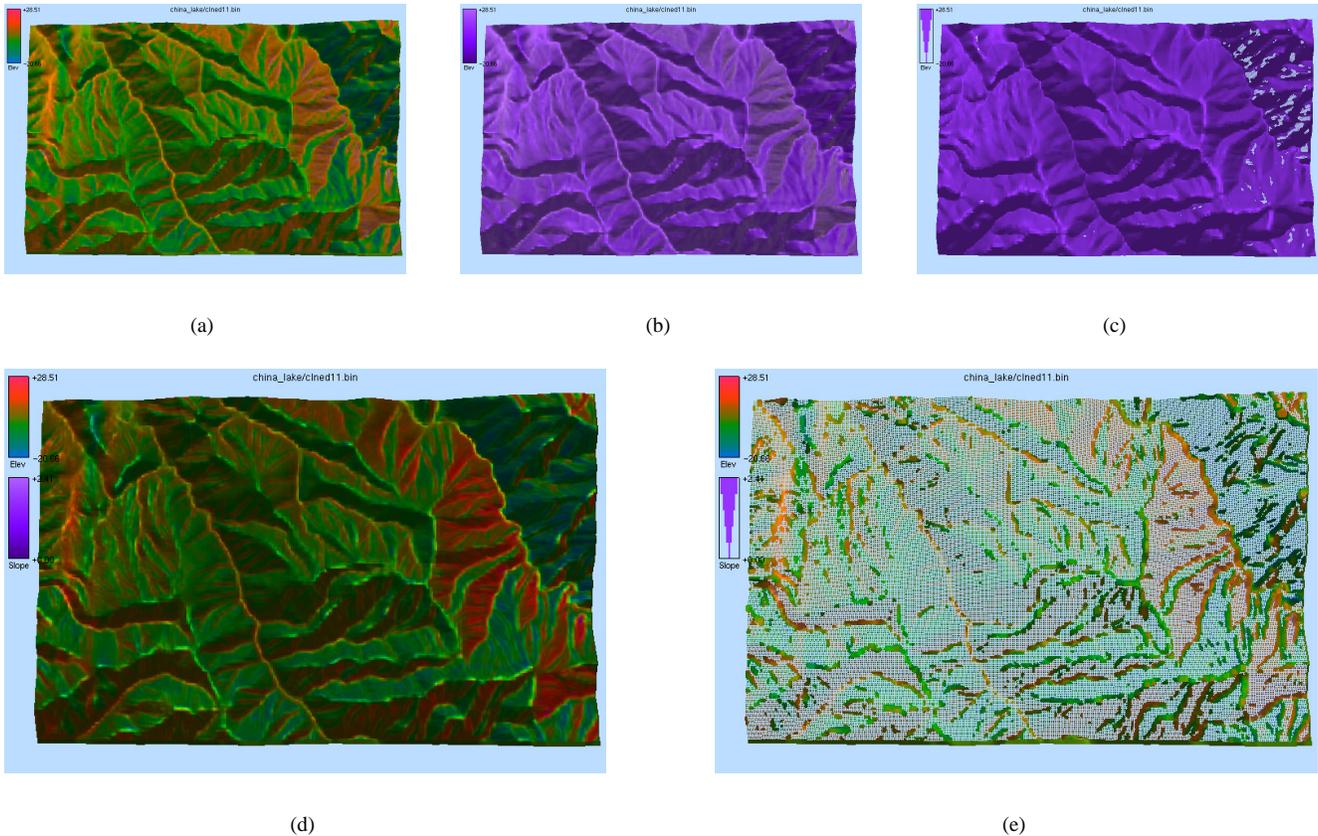


Figure 2: Examples of different visual features used to represent error values: (a) elevation error from the national elevation dataset (NED) terrain model at 1° resolution visualized using hue; (b) elevation error visualized using luminance; (c) elevation error visualized using size; (d) NED terrain model with elevation error visualized using hue and slope error visualized using luminance; (e) elevation error visualized using hue and slope error visualized using size

information using hue and luminance. Since the main task in our visualizations is error comparison, however, reasonable colours can still be selected even without knowing specific monitor properties.

Figure 2 shows examples of hue, luminance, and size used to visualize different error values. In Figure 2a, elevation error is visualized using hue, with green representing little or no error, red representing high positive error (*i.e.* z -heights above ground truth), and blue representing high negative error (*i.e.* z -heights below ground truth).

Figure 2b uses luminance to visualize the same elevation error: brighter regions for high positive error, and darker regions for high negative error. This image shows that luminance is less effective in a shaded relief environment, because variations in brightness due to error values are confounded by variations due to surface shading. Although we can turn off lighting effects, this then makes it difficult or impossible to judge surface orientations and elevations. Figure 3 demonstrates this by representing elevation error with luminance and disabling lighting. The underlying shading represents elevation error alone, with no direct cor-

respondence to surface direction (in spite of our natural tendency to interpret luminance changes as surface relief). A comparison with the true shaded surface (*e.g.* in Figure 2c) highlights this artifact.

Finally, Figure 2c uses size to visualize elevation error: large for high positive error, and small for high negative error. Although different positive errors are not immediately obvious at this resolution, negative errors are clearly visible as regions with background showing through the terrain. Compare this to the hue representation, where the small size of the negative error regions makes them more difficult to identify.

Figures 2d and 2e demonstrate visualizing elevation and slope error simultaneously with hue and luminance and hue and size, respectively. Because areas of high slope error are limited mainly to the peaks and valleys, both visualizations do a reasonable job of highlighting these locations. In the hue-luminance visualization, the areas with higher slope error are less obvious (again due to the confounding effect with surface shading), but the underlying elevation errors (visualized with hue) are easy to identify. The hue-size vi-



Figure 3: The terrain with elevation error visualized using luminance, and lighting disabled to remove the confounding effect of shading to represent surface orientation

ualization does a better job of highlighting regions of high slope error at the expense of presenting elevation error. Regions with low slope error produce smaller glyphs, which are not as effective at encoding distinguishable differences in hue. Again, this problem could be overcome by zooming in on specific regions of the map. This demonstrates how different combinations of visual features offer different strengths and weaknesses, and how these tradeoffs change as users interactively vary the way they view their data.

Our choice of a point cloud rather than a regular-grid mesh or triangulated irregular network (TIN) is based on a number of current and future requirements for our visualization tool. The digital elevation models we are asked to visualize can be irregular in their density and location of samples. This is because the acquisition of elevation and slope data varies depending on location and the sampling methodology (*e.g.* data from LIDAR, light detection and ranging, versus SRTM, the shuttle radar topography mission). Perhaps more importantly, we plan to investigate various methods for simplifying the terrain models, for example, a wavelet compression that can be applied to an irregular layout of samples. This representation provides point locations, but no connectivity information. Using point clouds frees us from having to add mesh edges in post-processing.

Users control which visual features to use to represent elevation or slope error. The domain of error values for a given terrain model is normalized and mapped to the range of displayable values for the given visual feature. Users can load multiple terrain models, and flip back and forth between them to compare slope and elevation error over the surface as a whole, or at specific spatial locations.

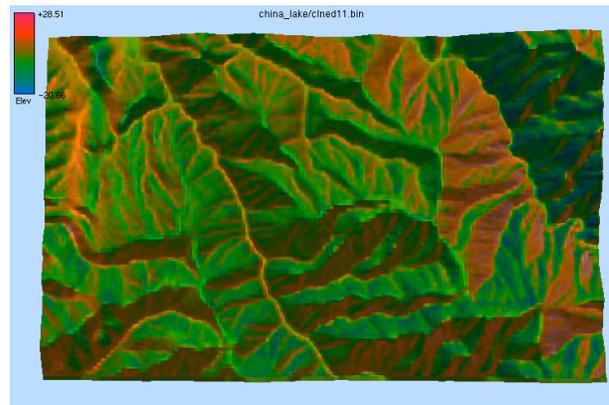
4. Conclusions and Future Work

We developed this perceptual-based visualization tool to help evaluate terrain representations and their errors, thus

we would like to conclude with a couple of vignettes for what aspects of this tool have already been helpful.



(a)



(b)

Figure 4: Effects of misregistered data: (a) the NED terrain in its original format, which is translated in y from corresponding positions in the ground truth terrain as indicated by the positive (red hue) errors on north-facing slopes and negative (blue hue) errors on south-facing. (b) the same NED terrain properly shifted to correct for the misregistration, producing significantly lower (green hue) elevation errors. Note how the ridge with error at left now stands out.

The tool shows correlations of error with terrain features and with other types of error. These have been important to see in several cases: for example, the tool showed us pretty quickly that the SRTM data flattened valleys and, to a lesser extent, ridges, a fact that was confirmed by Jim Slater, one of the data experts at NGA (personal

Correlation also shows where data is misregistered or warped, as the example of Figure 4 illustrates. These misregistrations can hide actual errors that become visible when they are removed.

Elevation and slope errors can also be used to highlight

changes in a terrain between data acquisition. Consider Figure 5, which shows SRTM data with elevation error visualized using hue and slope error visualized using luminance. A circular area in the upper-right corner of the terrain has high negative elevation error and high slope error. Google Earth imagery² reveals this as a large gravel pit, which apparently, was dug deeper after the ground truth data was acquired, but before the shuttle radar mission was conducted. This new topographical “feature” was therefore highlighted in the SRTM data as a large hole relative to ground truth.

To our surprise, our attempts to use velocity of motion to represent error were not as effective as we initially expected. Findings from both biological and computer vision suggest motion is one of the most salient visual cues. Psychophysical experiments have confirmed that various properties of motion like flicker, direction, and velocity can be used to encode data values in a visualization [25]. In our system, however, it is difficult to identify specific velocities, partly because small elevation and slope errors exist at almost all locations on many terrains, so every glyph is moving at some velocity. Moving glyphs break up the underlying surface, changing it from a shaded relief map into a collection of moving patches. This produces the impression of wave-like motion across the terrain, and confounds a user’s ability to pick out the velocity of any particular sample point and to compare velocities for different points. It was even difficult to determine if a sample point’s glyph is moving upward (for positive error) or downward (for negative error). Moreover, standard methods for viewing terrain do not favor the use of velocity. For example, when looking straight down on the terrain from above, velocity is difficult to detect because glyph motion paths are parallel to the viewing direction. When the terrain is rotated, the 3D perspective projection causes many of the glyphs to overlap with one another (*i.e.* groups of glyphs at different distances from the camera near to a common projection ray are rendered in similar regions on the screen). One possible improvement is to threshold velocity such that a minimum error is required before movement begins. Glyph motion could then be used to highlight regions of larger error values. Even with this change, it’s unclear how effective velocity can be in our visualization environment. Because of this, we plan to replace velocity with a visual feature better suited to our application environment.

The prototype is currently limited to small to moderate data sets of tens to hundreds of thousands of points. Some of the ways to raise this limitation are well-researched: the error visualization module needs to incorporate more culling techniques from Section 2.2, and the surface modules need to be engineered to accept large data sets. But there are other interesting questions for how to give the

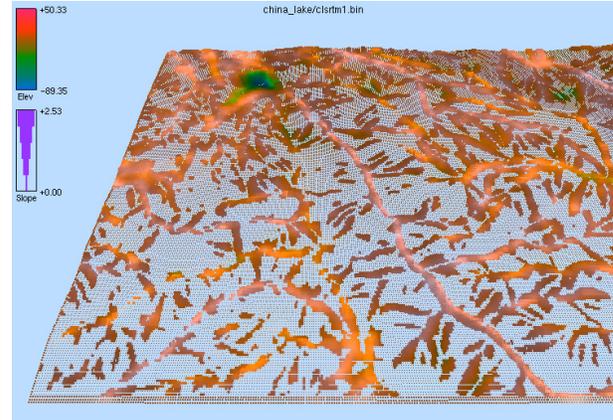


Figure 5: A visualization of shuttle radar topography mission (SRTM) elevation data; the region of high negative elevation error in the upper-right corner of the image is a gravel pit that was dug deeper after the ground truth terrain model was captured.

user the right perception of errors when much data is suppressed—if we give users a brush or zoom tool that allow them to express their level of interest in an area, how do we increase or decrease the resolution while keeping the perception of error consistent?

We would also like to enhance the ability to visualize error results for more two (or more) terrains together in a common display. For example, we might like to see how errors for SRTM terrains compare to errors for low-resolution DEM terrains. Currently, this requires flipping back-and-forth between two separate visualizations. Since the visual system is known to be highly inefficient at maintaining and comparing details, due to change blindness [27, 28]. A single combined display would allow us to directly analyze error values for the different surfaces at common terrain locations, and more importantly, see patterns in the error fields to answer questions such as: Where is one terrain better than another at estimating elevation or slope? Where is it worse? Are there particular properties of the terrain (valley, ridges, and so on) that drive these differences? A method to subdivide each glyph between several surfaces in an intelligent manner seems like a promising starting point towards achieving this goal.

Finally, we plan to validate our research with domain experts who study these types of datasets. This will test the basic capabilities of our design, and identify its strengths and weaknesses relative to existing tools currently used by the domain experts. To this end, we have initiated discussions with collaborators at the who study ways to store, represent, and simplify terrain, elevation, and gravity data at National Geospatial-Intelligence Agency. We are designing a prototype to provide to our collaborators, to allow them to visualize data at their organization. Feedback from these investigations will be used to further improve and enhance our visualization techniques.

²<http://earth.google.com>

Acknowledgments

This work was partially supported by NGA/Darpa HM1582-05-2-0003, and by National Science Foundation CISE-ACI-0092308.

References

- [1] Carlton Daniel and Keith Tennant, “DEM quality assessment”, in *Digital Elevation Model Technologies and Applications: The DEM Users Manual*, David F. Maune, Ed., pp. 395–440. ASPRS, Bethesda, MD, 2001.
- [2] T. K. Peucker, R. J. Fowler, J. J. Little, and D. M. Mark, “The triangulated irregular network”, in *Amer. Soc. Photogrammetry Proc. Digital Terrain Models Symposium*, 1978, pp. 516–532.
- [3] Mark P. Kumler, “An intensive comparison of triangulated irregular network (TINs) and digital elevation models (DEMs)”, *Cartographica*, vol. 31, no. 2, 1994, monograph 45.
- [4] C. Touma and C. Gotsman, “Triangle mesh compression”, in *Graphics Interface '98 Proceedings*, 1998, pp. 26–34.
- [5] D. Poulalhon and G. Schaeffer, “Optimal coding and sampling of triangulations”, in *30th International Colloquium on Automata, Languages and Programming (ICAZLP)*, 2003, pp. 1080–1094.
- [6] David F. Maune, Ed., *Digital Elevation Model Technologies and Applications: The DEM Users Manual*, ASPRS, Bethesda, MD, 2001.
- [7] N. D. Gershon, “Visualization in an imperfect world”, *IEEE Computer Graphics & Applications*, vol. 18, no. 4, pp. 43–45, 1998.
- [8] A. M. MacEachren, A. Robinson, S. Hopper, S. Gardner, R. Murray, M. Gahegan, and E. Hetzler, “Visualizing geospatial information uncertainty: What we know and what we need to know”, *Cartography and Geographic Information Science*, vol. 32, no. 3, pp. 139–160, 2005.
- [9] C. M. Wittenbrink, A. T. Pang, and S. K. Lodha, “Glyphs for visualizing uncertainty in vector fields”, *IEEE Transactions on Visualization and Computer Graphics*, vol. 2, no. 3, pp. 266–279, 1996.
- [10] A. T. Pang, C. M. Wittenbrink, and S. K. Lodha, “Approaches to uncertainty visualization”, *The Visual Computer*, vol. 13, no. 8, pp. 370–390, 1997.
- [11] C. R. Johnson and A. R. Sanderson, “A next step: Visualizing errors and uncertainty”, *IEEE Computer Graphics & Applications*, vol. 23, no. 5, pp. 6–10, 2003.
- [12] G. Grigoryan and P. Rheingans, “Point-based probabilistic surfaces to show surface uncertainty”, *IEEE Transactions on Visualization and Computer Graphics*, vol. 10, no. 5, pp. 564–573, 2004.
- [13] Christopher G. Healey and James T. Enns, “Large datasets at a glance: Combining textures and colors in scientific visualization”, *IEEE Transactions on Visualization and Computer Graphics*, vol. 5, no. 2, pp. 145–167, 1999.
- [14] Christopher G. Healey, James T. Enns, Laura G. Tateosian, and Mark Remple, “Perceptually-based brush strokes for nonphotorealistic visualization”, *ACM Transactions on Graphics*, vol. 23, no. 1, pp. 64–96, 2004.
- [15] C. Weigle, W. G. Emigh, G. Liu, R. M. Taylor, J. T. Enns, and C. G. Healey, “Oriented texture slivers: A technique for local value estimation of multiple scalar fields”, in *Proceedings Graphics Interface 2000*, Montréal, Canada, 2000, pp. 163–170.
- [16] M. A. Duchaineau, M. Wolinsky, D. E. Sigeti, M. C. Miller, C. Aldrich, and M. B. Mineev-Weinstein, “ROAMing terrain: Real-time optimally adapting meshes”, in *Proceedings Visualization '97*, Phoenix, Arizona, 1997, pp. 81–88.
- [17] F. Losasso and H. Hoppe, “Geometry clipmaps: Terrain rendering using nested regular grids”, in *SIGGRAPH 2004 Conference Proceedings*, Joe Marks, Ed., Los Angeles, California, 2004, pp. 769–776.
- [18] P. Cignoni, F. Ganovelli, E. Boggetti, F. Marton, F. Ponchio, and R. Scopigno, “Planet-sized batched dynamic adaptive meshes (P-BDAM)”, in *Proceedings Visualization 2003*, Seattle, Washington, 2003, pp. 147–154.
- [19] P. Lindstrom and V. Pascucci, “Terrain simplification simplified: A general framework for view-dependent out-of-core visualization”, *IEEE Transactions on Visualization and Computer Graphics*, vol. 8, no. 3, pp. 239–254, 2002.
- [20] L. A. Treinish, “Task-specific visualization design”, *IEEE Computer Graphics & Applications*, vol. 19, no. 5, pp. 72–77, 1999.
- [21] A. J. van Doorn and J. J. Koenderink, “Temporal properties of the visual detectability of moving spatial white noise”, *Experimental Brain Research*, vol. 45, pp. 179–188, 1982.
- [22] Lynn Bartram, Colin Ware, and Tom Calvert, “Moticons: Detection, distraction, and task”, *International Journal of Human-Computer Studies*, vol. 58, no. 5, pp. 515–545, 2003.
- [23] Colin Ware, *Information Visualization: Perception for Design*, Morgan Kaufmann Publishers, Inc., San Francisco, California, 2004.
- [24] CIE, *CIE Publication No. 15, Supplement Number 2 (E-1.3.1, 1971): Official Recommendations on Uniform Color Spaces, Color-Difference Equations, and Metric Color Terms*, Commission Internationale de L'Éclairage, 1978.
- [25] Daniel E. Huber and Christopher G. Healey, “Visualizing data with motion”, in *Proceedings Visualization 2005*, Minneapolis, Minnesota, 2005, pp. 527–534.
- [26] S. Mateeff, G. Dimitrov, and J. Hohnsbein, “Temporal thresholds and reaction time to changes in velocity of visual motion”, *Vision Research*, vol. 35, no. 3, pp. 355–363, 1995.
- [27] Ronald A. Rensink, “Seeing, sensing, and scrutinizing”, *Vision Research*, vol. 40, no. 10–12, pp. 1469–1487, 2000.
- [28] Daniel J. Simons, “Current approaches to change blindness”, *Visual Cognition*, vol. 7, no. 1/2/3, pp. 1–15, 2000.