Innovation in Lidar Processing Technology

by Mark E. Romano

In its infancy, lidar represented a paradigm shift in the remote sensing technology used to capture extremely high resolution elevation point data. While the benefits of this new technology were scientifically evident, putting the technology into realistic practice left much undone to fully and efficiently exploit source data. The photogrammetric industry was accustomed to a small number of elevation points and largely relied on hard and soft breaklines to sufficiently detail large scale topographic datasets. In addition, traditional CAD platforms became overwhelmed with the enormous amount of information yielded from this new sensor technology. As a result, many early implementations failed to produce efficient and effective solutions for data providers and end users. While computer technology advanced rapidly with respect to bandwidth, memory, and reduced cost, existing application architecture did not.

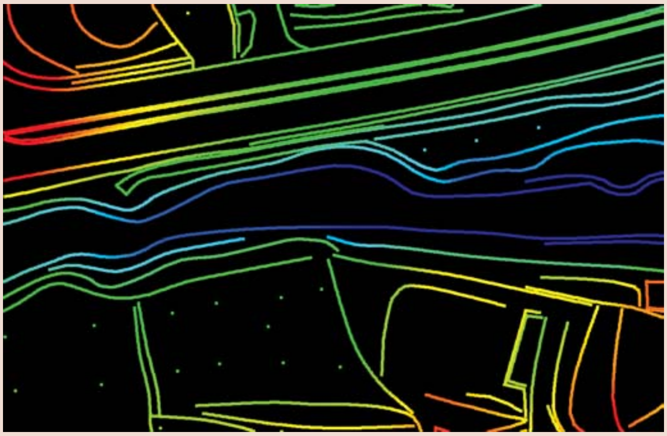

Figure 1. Traditional mass points and breaklines.

Over the last decade typical pulse rates for lidar sensors have risen from 4 KHz to 100 kHz. This has improved data resolution, accuracy, and further impacted processing rates and data volumes. Softcopy platforms are currently limited with respect to Triangulated Irregular Network (TIN) displays, which are orders of magnitude larger at equivalent scales. High resolution imagery, usually a major component of Digital Surface Model (DSM) data set generation, further exacerbates problems of surface edit and efficient compilation of breaklines and planimetric features.

For those who adopted lidar early, the only way to process data was to utilize numerous commercial solutions to piece together complicated workflows that eventually yielded the final data products necessary to satisfy user requirements and map specifications. This tedious and labor intensive process required multiple reformatting of text files as the only standard means of transporting data throughout the workflow. Many lidar companies have formed since inception of the technology, but commercial workflow solutions remain inefficient for people entering the industry.

Plato wrote that "necessity is the mother of invention." This was truly the case for Merrick & Company (Merrick) when we decided to purchase our lidar system in 2001. Having experience with lidar data since 1997, we also decided to commit significant programming resources to overcome the challenges encountered in lidar processing workflows. Our goal was to develop a C++ file-based application for the Windows operating system.

Initial development objectives and requirements for the application included the following:

- To overcome file size limitations of commercial software to manage and process large data volumes
- To develop a high performance graphic environment incorporating TINS, imagery, and contour displays

Figure 2. Ortho with mass points and breaklines superimposed.

- To reduce reliance on multiple software platforms
- To both automate and improve the efficiency of data classification
- To be compliant with common CAD and GIS platforms
- To generate breaklines more efficiently / accurately and to evaluate their appropriate use in lidar.

Performance Barriers

As illustrated in Figures 1 and 2, traditional photogrammetric users have defined surface models with data sources primarily composed of sparse mass points and numerous breaklines. Even at the largest map scales, this approach produced manageable file sizes.

In contrast, lidar data offered superior resolution but overwhelmed the TIN rendering and display limits of traditional CAD and softcopy workstation platforms. Lidar dramatically increased file sizes to unmanageable levels. This increase in data volume is evident when Figure 1 is compared to Figure 4. Figure 4 shows lidar data superimposed on an orthophoto. Although a much smaller project area than that used in Figure 1, this dataset contains many thousand more elevation points at equivalent scales. Hundreds of gigabytes of elevation data are not uncommon for lidar datasets.

continued on page 1202

continued from page 1201

Incorporation of all attribute fields of a single lidar elevation point requires 28 to 32 bytes per point in current binary formats. To fully utilize all available lidar data downstream, it was decided that all attributes should be carried throughout the workflow, but existing data formats made it impractical.

Development Considerations and Key Decision Points

Since it was the de-facto accepted hardware vendor industry standard for processing raw points, we adopted the .las binary format for display, storage, and manipulation of large elevation datasets. The application would also be compatible with our Leica-Geosystems ALS 40 lidar post-processor, eliminating the need for data reformatting. The .las binary format greatly reduced file size, and allowed us to carry all data fields. To overcome poor TIN performance and current commercial application file size limitations, development of a high performance TIN rendering engine was necessary to visualize, classify, and edit large volumes of surface data. Performance was made scalable based on machine resources. As a result, TIN rendering did not require a supercomputer or an expensive graphic card to run efficiently. Data can now be efficiently manipulated with computers much older by today's standards. Currently, demonstration and training are routinely accomplished with laptop computers, the primary restriction being storage space for large area projects. OpenGL was selected for graphic display since it is openly published, well documented, and supports the Windows OS. Data display performance has dramatically increased whereas file size limitations have largely been mitigated. This has provided a solid foundation on which to build additional tools. ESRI shapefile format was added as a visual means of managing, viewing, filtering, and parsing large datasets into manageable files, supporting internal workflow and end user tile schemes. Tile schemes can be transparently overlaid on the dataset for ease of use, fully supporting a multiple attribute table.

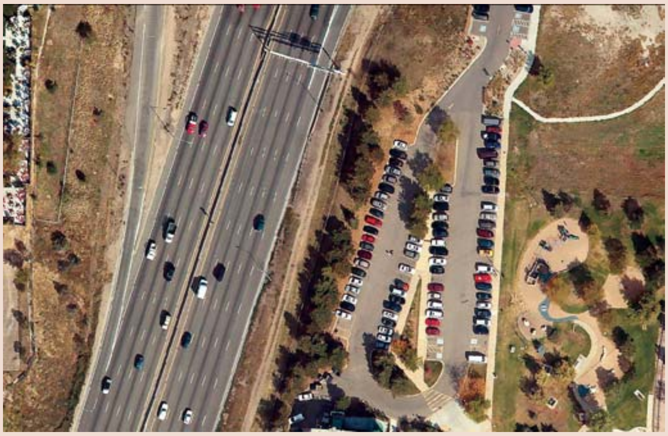

Figure 3. Ortho AOI.

Data Import/Export

Data export functionality was designed to reformat output data for common CAD and GIS software. The export functionality can be operated via a simple GUI or through a batch processing mechanism. Our philosophy was to keep the application CAD/GIS neutral, addressing the requirements of all potential software applications. Data for export could be selected from graphically delimited areas, complex polygons, or tile schemes, with selectable multi-class data, tile attributes, and buffers. For contour and grid utility, TIN's could be computed by any combination of data class. Grid algorithms were added to support text or ESRI binary formats, with user specified variables including Inverse Distance Weighting (IDW) functionality. Data import was designed so that lidar, existing photogrammetric, or other terrain data sources could be imported, tested, or manipulated. In designing the import functionality, we recognized that many projects have significant volumes of legacy data generated from years of surveying and mapping. This could be efficiently analyzed within the application to determine their accuracy and long-term value.

Imagery

Although lidar is a high resolution data source, it was evident that the imagery itself was an important part of the process and would be a necessary backdrop for intelligent editing of the DSM. Standard TIFF and wavelet image libraries, including JPEG2000 were developed to optimize performance. The image layer could be viewed as a backdrop for point clouds, contours, and shaded relief TINs. Transparency settings were user selectable.

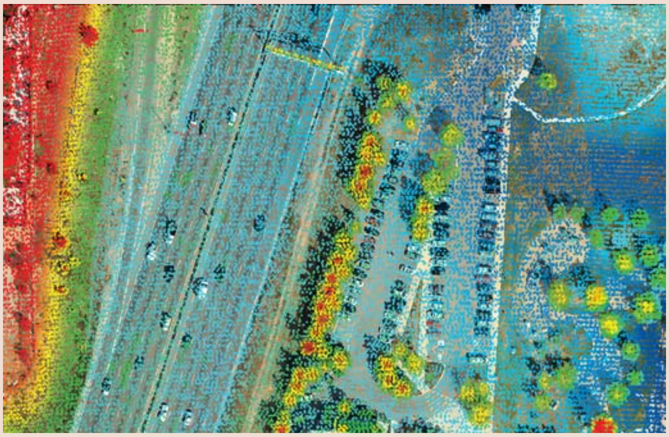

Figure 4. Ortho with lidar elevation data superimposed.

Contour Rendering

Contours are an important visualization tool for editing terrain data. However, like TINs, contour display within CAD applications is inefficient. In our software, the contour algorithms were developed to display on the TIN and/or image data layers. As a result of the high performance TIN engine adopted in the core application, the rendering performance is dramatic. Contours are immediately refreshed as TINs are reclassified and edited. This provides the editor or compiler immediate feedback. Contour batch export to CAD file functionality was incorporated to alleviate Digital Terrain Model (DTM) generation on outside platforms. Users can specify contour interval, tile schemes, buffers, or seamless generation of whole areas. The intent of the contour module was not to replace CAD or GIS platforms, rather the goal was to alleviate the most labor intensive steps in our workflow. Final contours are imported and attributed in commercially available software at the end of the process flow, as dictated by client deliverable formats and requirements.

The display, performance, and file size constraints for rendering TINs, contours, and imagery on the fly were greatly reduced. As a result, the seamless editing of large data blocks in a single cohesive application was realized. Users familiar with softcopy DTM editing techniques appreciate the value of this performance improvement and capability. Previously, in order to edit DTM data we were required to physically reload the numerous stereo models and adjoining surface data blocks, and then regenerate TIN and contour overlays. Finally, 3D rotational and profile windows were developed to provide the means for knowledgeable display and editing of point cloud and TIN data in a 3D environment (see Figure 5.)

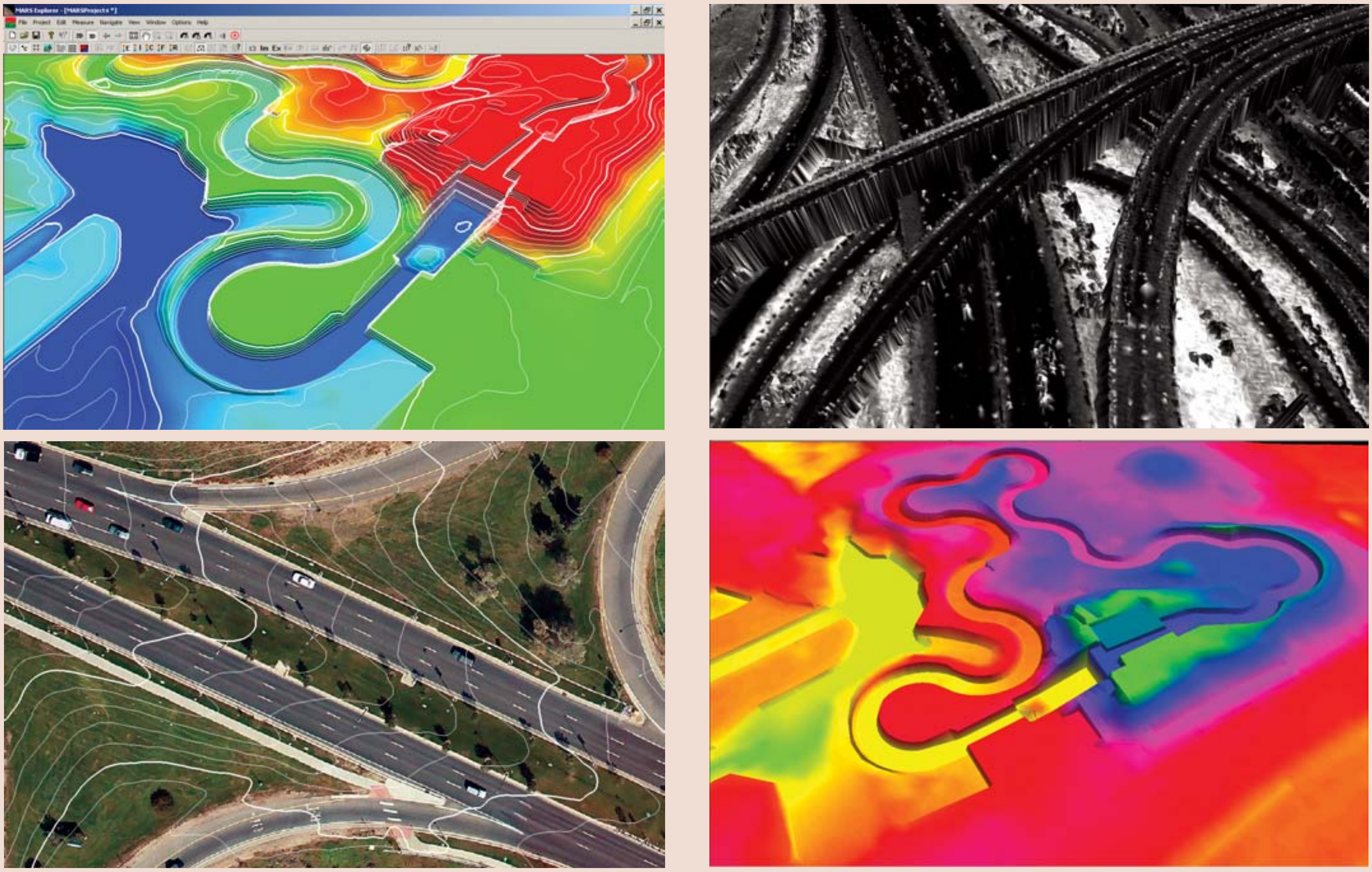

Figure 5. Illustrates; shaded relief with contours in 3D (ul), 3D lidar intensity (ur), ortho with real-time contour generation (ll), and 3D gradient color cycles (lr).

Quality Assurance

Quality assurance functionality was likewise incorporated into our software development plans. For quality assurance tests using GPS or other field verification methods, control points or cross sections could be imported and computed against the TIN at any time during the production process. Report statistics could be generated for all current data specifications in an easy to use GUI interface. For other analysis, data could be exported in Excel format. Supported specifications and standards include ASPRS, NMAS, FEMA, NSSDA, RMSE, and others. The user was given the option of using a contour wizard or directly specifying the vertical accuracy goal at any percentage level. In every case, all statistics are generated, with a clear indication of specified pass/fail criteria.

Data Classification (filtering)

Rigorous classification algorithms were developed to classify (separate) reflective surface and, ground (bald earth) data, to obtain an accurate topographic model. Automated batch filter routines were designed to yield 95%+ for all types of terrain and morphology. A flexible XML macro batch script was designed to iterate on datasets using any order and/or combination of filter algorithms, thereby maximizing the yield of the filter, and eliminating much of the manually intensive post-edit process. Figure 6 illustrates results from the automated filtering process with additional manual classification of discrete features. Buildings were classified automatically with developed algorithms. Additional discreet features were classified from orthographic and cross sectional views. In the software, feature data could be attributed into 265 discreet colors and numerical classes, and were definable in the project support file. Display class combinations were user selectable for display of, TIN rendering, contours, and export routines.

Post-Filter Edit

No automated filter process is 100% accurate. Since previous attempts to edit DSM data on softcopy platforms were highly inefficient, a complete set of editing tools were developed for DSM editing within the application environment. This allowed compilers to edit most surface model errors in a rapid display and reclassification

continued on page 1204

continued from page 1203

environment. Tools were designed to edit in the point cloud environment (in both orthographic and cross sectional views) with a simple pallet of paintbrush style edit tools. Dual screens, with multiple views and dockable tool pallets were supported. Viewing scale, color cycles, gradients, and overlays (contours, imagery, TINs, points) were independent and definable by the user, with selectable transparency levels.

Automated Feature Classification: The "Holy Grail?"

Interest has rapidly grown in methods designed to exploit lidar data beyond topographic needs. Traditional planimetric feature extraction using softcopy techniques is expensive due to the labor intensive nature of the compilation process. Reflective surface data contain all the elements to efficiently produce feature rich 3D models, but present significant challenges to both wire frame model generation and complex algorithms. Nevertheless, power companies were first to take advantage of this opportunity with danger zone (line clearance) applications. Other applications include true orthographic image surface generation, planimetric feature classification, and potential defense applications.

Derived planimetric accuracies are primarily driven by GPS accuracy, Ground Sample Distance (GSD) or lidar data resolution, Inertial Measurement Unit (IMU) system accuracy, and spot (beam) diameter on the ground. Realistic edge detection and its resultant horizontal planimetric accuracy is achievable in the range of 1-2m, but substantial research is still required. Many see this as the "Holy Grail" , but as lidar pulse rates (resolution) and accuracy continue to improve, so will planimetric accuracy. Automated building extraction is currently achievable, but accuracy and visual representation goals are undefined. Currently we are developing algorithms to automate wire frame generation of manmade features. The objective is to reduce file size while retaining realistic depictions of individual features. In parallel we are developing an integrated database to store all 2D and 3D vector information. When compiled, feature data are stored in two locations: 1) point data in .las format to affect the TIN in our application environment, and 2) vector files for export and visualization in GIS environments (see Figure 7.)

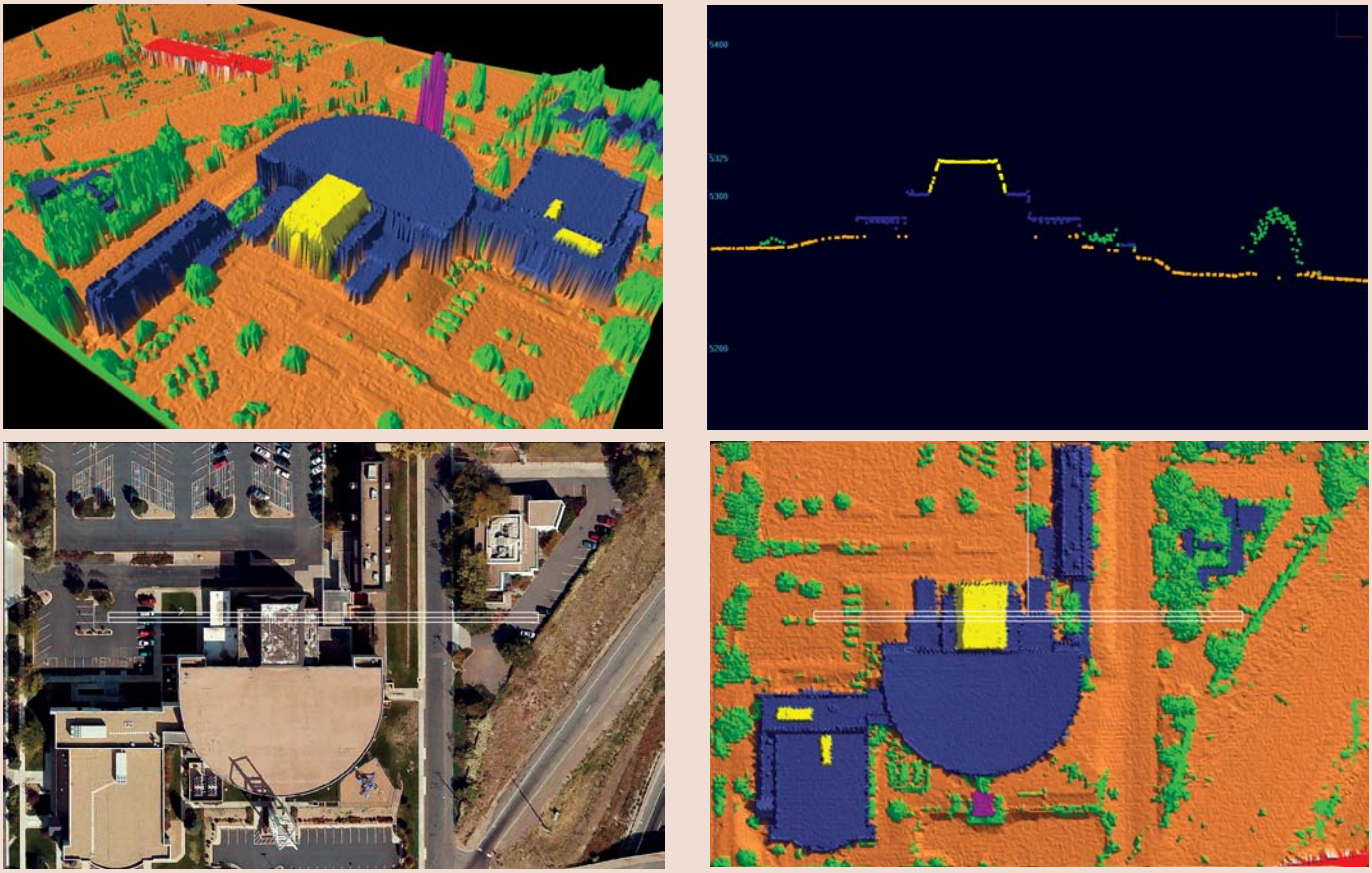

Figure 6. Cross sections (shown in white) are superimposed on the TIN (ul) and ortho image (ll). The resulting point cloud data (ur) is displayed by its classification color. Discrete classifications and color schemes for this data include: Ground Surface (8-brown), Canopy (5-green), Towers (7-Magenta), Cooling Infrastructure (10-Yellow), and Bridges (1-Red).

Lidar and Breaklines - When are They Necessary?

As previously noted in Figure 3, modern lidar systems generate enormously high-resolution data sets. The question of whether breaklines are necessary has spurred hot industry debate in recent years between traditionalists and lidar providers. In reality, the question is easily answered by the application and/or accuracy requirements of the data to be produced. FEMA has developed a lidar specification (technology appropriate) that is not driven by traditional photogrammetric based compilation techniques (a process driven specification) such as ASPRS or NMAS. Its premise tests for high data accuracy (95% RMSE), but does not have the traditional requirements for cartographic adherence. This does not discount either the technology appropriate or process driven approach for accuracy, rather it highlights that the two perspectives have distinctly different characteristics that warrant assessment by alternative methods. Additionally, the FEMA specification does not imply that breaklines are unnecessary for lidar data, only that there may be exceptions due to different target characteristics. An example of complex terrain is illustrative. With traditional techniques, soft breaklines and mass points must be compiled to reflect even minor surface grade changes. In contrast, in many instances lidar data support enough resolution to eliminate many soft breaklines.

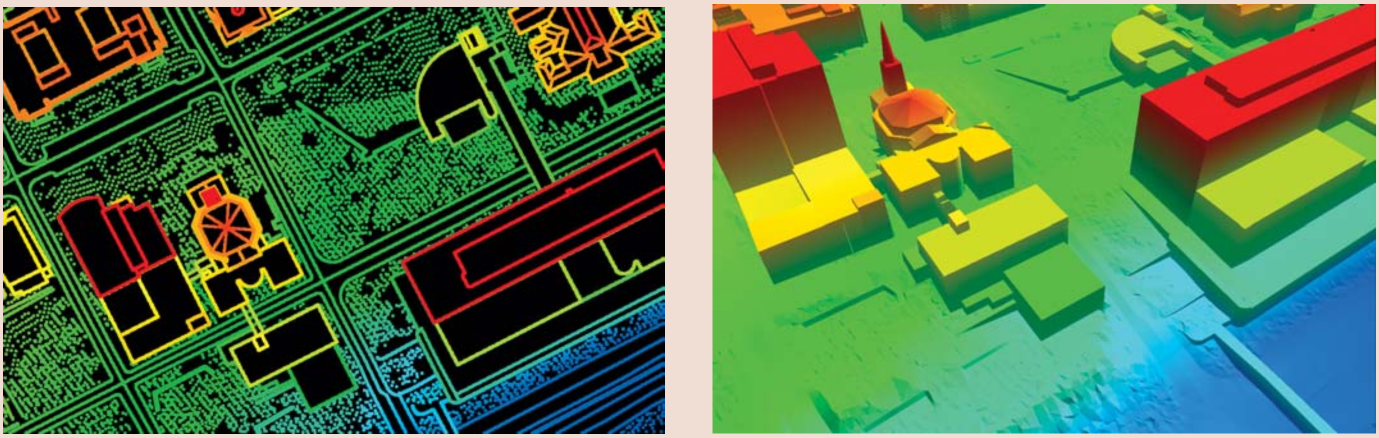

Figure 7. depicts lidar mass point data, breaklines, and compiled planimetric features, with resultant solid model rendering shaded by elevation in the application.

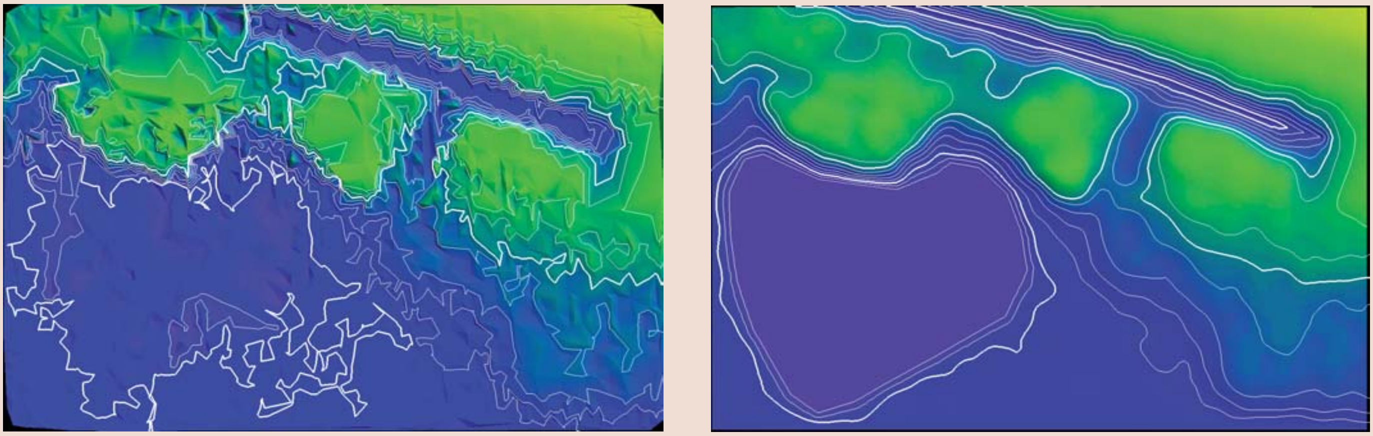

Figure 8. Raw lidar contours.

Contours as a result of the drape to surface routine.

Regardless of the method employed to generate a DSM, hard surface breakline requirements remain unchanged today. However, new techniques are being employed to compile breaklines outside the softcopy environment. There are positive and negative aspects on both sides of this discussion, all of which should be understood in order to make appropriate choices when writing scope of work and accuracy specifications for a particular data application. We have developed unique tools utilizing a 2D/3D draping approach for road and hydrographic features. For aboveground features such as buildings, careful consideration should be given to required planimetric horizontal accuracy requirements. While relief displacement is not relevant in a lidar dataset, it is affected directly by both the aforementioned constraints and data resolution (GSD). Our development goal is to eliminate as much of the softcopy edit environment as possible, thereby reducing man-hours.

A hydrographic feature cannot be accurately represented without supplemental breaklines. Two primary applications are cartographically correct contours and watershed analysis. Today's modern lidar systems are designed at a wavelength of 104.6nm and do not accurately produce elevation data on water surfaces. Secondly, water is not typically at a perfect static level and elevations interior to pond like features must be removed. Stream channels must flow downhill and ponds must 1) have a constant elevation and 2) approximate the top and bottom edge of the embankment.

Figure 8 depicts examples of lidar data with and without the addition of breaklines on pond and stream channel features. Elevation data are normalized (denoised) to produce a smooth cartographic representation. The lake elevation tool compiles average static level and automatically removes point data interior to the polygon, allowing only a single elevation to be permutated throughout the pond surface. The Stream Channel tool was designed to drape to the lidar surface and accept only the next lowest elevation point at a specified buffer width, while compiling downstream. During the draping stage the user specifies a buffer width to delete elevations close to the

continued on page 1206

continued from page 1205

breakline, producing a discreet break in the terrain. This process enhances cartographic representation and accuracy.

Summary

Over the last three years our application has become increasingly feature rich, alleviating over 90% of our reliance on commercial software applications and significantly increasing process efficiency and data accuracy. Our team continues to provide important feedback for tool development in a real-world production environment, enabling us to further improve our application. There seems no end in site for our development priority and wish lists, only questions of "which should come first?" and "how much will it cost?" We have learned a great deal from true software application development, rather than working with the patchwork of programs that new users must face. Looking forward, we will still use many of the tools commercially available along with our application. To remain competitive, it requires a never ending effort to keep pace with technology.

◆

Author

Mark Romano
Director of Lidar Operations
Merrick & Company
11906 Hangar Court, Broomfield, Colorado 80021
(303) 439-0803, Fax: (303) 439-0794, mark.romano@merrick.com

Mark Romano has over 27 years of experience in electro-optical systems development including 20 years in the aerospace and Remote Sensing industries. He has worked specifically in the lidar mapping and GIS industries since 1994, developing and commercializing remote sensing hardware, software applications, and workflows. Romano is currently the Director of Lidar Operations for Merrick & Company located in Aurora, Colorado, and has successfully implemented the lidar business plans and purchase of several lidar sensors over the past ten years. He leads the development and integration activities associated with lidar, digital camera systems (DACS™), and the Merrick Advanced Remote Sensing (MARS™) software application. He is a member of the American Society of Photogrammetry and Remote Sensing (ASPRS), GeoSpatial Information Technology Association (GITA), and, the Urban and Regional Information Systems Association (URISA).