

3-D OBJECT RECOGNITION FROM POINT CLOUD DATA

W. Smith, A.S. Walker, B. Zhang

BAE Systems, San Diego, CA 92127, USA
(william.smith4, stewart.walker2, bingcai.zhang)@baesystems.com

KEY WORDS: Automation, Extraction, Building, Feature, DEM/DTM, Point Cloud, LIDAR

ABSTRACT:

The market for real-time 3-D mapping includes not only traditional geospatial applications but also navigation of unmanned autonomous vehicles (UAVs). Massively parallel processes such as graphics processing unit (GPU) computing make real-time 3-D object recognition and mapping achievable. Geospatial technologies such as digital photogrammetry and GIS offer advanced capabilities to produce 2-D and 3-D static maps using UAV data. The goal is to develop real-time UAV navigation through increased automation. It is challenging for a computer to identify a 3-D object such as a car, a tree or a house, yet automatic 3-D object recognition is essential to increasing the productivity of geospatial data such as 3-D city site models. In the past three decades, researchers have used radiometric properties to identify objects in digital imagery with limited success, because these properties vary considerably from image to image. Consequently, our team has developed software that recognizes certain types of 3-D objects within 3-D point clouds. Although our software is developed for modeling, simulation and visualization, it has the potential to be valuable in robotics and UAV applications.

The locations and shapes of 3-D objects such as buildings and trees are easily recognizable by a human from a brief glance at a representation of a point cloud such as terrain-shaded relief. The algorithms to extract these objects have been developed and require only the point cloud and minimal human inputs such as a set of limits on building size and a request to turn on a squaring option. The algorithms use both digital surface model (DSM) and digital elevation model (DEM), so software has also been developed to derive the latter from the former. The process continues through the following steps: identify and group 3-D object points into regions; separate buildings and houses from trees; trace region boundaries; regularize and simplify boundary polygons; construct complex roofs. Several case studies have been conducted using a variety of point densities, terrain types and building densities. The results have been encouraging. More work is required for better processing of, for example, forested areas, buildings with sides that are not at right angles or are not straight, and single trees that impinge on buildings. Further work may also be required to ensure that the buildings extracted are of fully cartographic quality. A first version will be included in production software later in 2011.

In addition to the standard geospatial applications and the UAV navigation, the results have a further advantage: since LiDAR data tends to be accurately georeferenced, the building models extracted can be used to refine image metadata whenever the same buildings appear in imagery for which the GPS/IMU values are poorer than those for the LiDAR.

1. INTRODUCTION

In the past few decades, attempts to develop a system that can automatically recognize and extract 3-D objects (buildings, houses, single trees, etc.) from imagery have not been successful. The radiometric properties of 3-D objects are very complex and variable. Because of the different colors and patterns, it is very difficult for any algorithm to extract multiple buildings and houses automatically from imagery alone (Figure 1). Algorithms that work well with one set of images and 3-D objects may not work at all with a different set, because radiometric properties are often very different.

LIDAR data has unique properties for automatic extraction of 3-D objects. The most important and *invariant* property of a 3-D object in LIDAR data is 3-D. In other words, the very availability of Z distinguishes objects better than the 2-D image view. We can use this property to identify, extract, and label 3-D objects automatically. To identify an object in digital images, it is crucial to use an object property that does not change, i.e., is invariant. The 3-D properties of a 3-D object are ideal. As shown in Figure 2, the terrain shaded relief (TSR) makes manifest 3-D objects in a point cloud. In this case the point cloud was photogrammetrically derived from stereo imagery by means of NGATE software for extracting elevation automatically by matching multiple overlapping images (Zhang and Walter, 2009), but the algorithms in this paper are equally applicable to point clouds whether they come from LIDAR or

photogrammetry. All of the 3-D objects have one common property — they are above the ground. Modern stereo image matching algorithms and LIDAR provide very dense, accurate point clouds, which can then be used for automatic extraction of 3-D objects (Zhang and Smith, 2010).



Figure 1. Six different building colors and patterns from one image with a GSD 0.14': a supervised building region growing classification would need six signatures. Two (upper-right and lower-middle) cannot be used because they are inhomogeneous.

To identify an object in digital images, it is crucial to use an object property that does not change, i.e., is invariant. The 3-D properties of a 3-D object are ideal. As shown in Figure 2, the terrain shaded relief (TSR) makes manifest 3-D objects in a point cloud. Their locations and their approximate shapes are obvious. It is much easier to classify a TSR for 3-D objects than to classify digital images. The six buildings in Figure 1 are

shown in Figure 2. In this case the point cloud was photogrammetrically produced from stereo imagery by means of NGATE software for extracting elevation automatically by matching multiple overlapping images. It is available as an optional module for BAE Systems' commercial-off-the-shelf SOCET GXP[®] and SOCET SET[®] products (Zhang and Walter, 2009), but the algorithms in this paper are equally applicable to LIDAR or photogrammetrically derived point clouds. The 3-D objects have a common property — they are above the ground. Modern stereo image matching algorithms and LIDAR provide very dense, accurate point clouds, which can be used for automatic extraction of 3-D objects (Zhang and Smith, 2010).



Figure 2. TSR of a point cloud: 3-D objects are very apparent.

We have developed a system for 3-D object extraction called Automatic Feature Extraction (AFE). Although originally designed for modeling, simulation and visualization, the system has potential for use in robotic and UAV applications. The same algorithms could be used to extract and identify other types of 3-D objects such as vehicles, airplanes and people.

2. TECHNICAL APPROACH

2.1 Automatic transformation from a point cloud to a bare-earth model

The first key algorithm automatically transforms a LIDAR or photogrammetrically derived point cloud into a bare-earth model. The differences between a point cloud and a bare-earth model are the approximate shapes and locations of 3-D objects. In the past few years, we have developed several algorithms to transform a point cloud into a bare-earth model for specific cases. These have been used extensively by our customers with positive feedback. There are, however, no general-purpose algorithms that work for all types of terrain. Automatic extraction of 3-D objects without human interaction requires a generic bare-earth algorithm that works for most cases. We combined several specific algorithms to transform a point cloud into a bare-earth model for 3-D object extraction:

- Bare-Earth Profile: uses terrain profiles in different directions to identify non-ground points
- Bare-Earth Morphology: uses morphological operators to identify non-ground points
- Bare-Earth Histogram: uses elevation distribution or histograms to identify non-ground points
- Bare-Earth Dense Tree Canopy: uses local minimum elevation points to identify non-ground points

2.2 Automatic 3-D object extraction from point clouds using the bare-earth model

We have developed several key algorithms to extract 3-D objects from point clouds and bare-earth models automatically.

2.2.1 Identifying and grouping 3-D object points into regions:

Based on the difference between the DSM and the

DEM, we identify points with a height difference greater than the minimum 3-D object height, which is a parameter based on user input. We group these points such that points belonging to the same 3-D object have the same group ID and points belonging to different 3-D objects have different group IDs. This grouping algorithm is based on the height values and spatial relationships of these points.

2.2.2 Separating buildings and houses from trees:

Trees are generally found close to a house or a building. These trees may hang over or attach to a house or building. To extract the boundary of a house or building accurately, it is necessary to separate the trees. We assume that the roof tops of a house or building consist of a number of 3-D planes. Based on this assumption, we use dynamic programming and RANSAC (RANDOM SAMPLE CONSENSUS) algorithms to separate trees. In most cases, tree canopies do not form a 3-D plane. Points that do not belong to any 3-D plane are likely to be tree points. There are exceptions for points on air conditioners, TV antennae, etc. To overcome these exceptions, we have developed four region-growing algorithms to bring these points back based on their spatial relationships.

2.2.3 Tracing region boundaries:

We trace the outermost points of a region of points to form a polygon boundary.

2.2.4 Differentiating single trees from buildings and houses:

In most cases, LIDAR points on a single tree canopy will not form any accurate and sizable 3-D planes, nor will boundary segments of a single tree have a good dominant direction. We use these two criteria to differentiate single trees from buildings and houses.

2.2.5 Regularizing and simplifying boundary polygons:

Most houses and buildings have boundaries consisting of parallel and perpendicular segments. Based on this assumption, we have developed a RANSAC algorithm for 3-D lines to simplify boundary segments. We have developed another algorithm to estimate the dominant direction from the simplified boundary segments, which is used for regularizing. Once the dominant direction has been determined, we force the simplified 3-D line segments to be either parallel or perpendicular to the dominant direction. For 3-D objects with roofs consisting of multiple 3-D planes, we use the most reliable intersecting 3-D line from 3-D planes as the dominant direction. For 3-D objects with segments not parallel or perpendicular, such as curves, the estimation of dominant direction may fail. In this case, we extract 3-D line segments using a dynamic programming and least-squares fitting algorithm. We then link and intersect these 3-D line segments to form the boundary polygon of a 3-D object.

2.2.6 Constructing complex roofs:

A 3-D plane in a XYZ coordinate system has the equation $z = ax + by + c$. We cannot use a set of such equations to model a complex roof. We need to find the intersecting line between two 3-D planes. We need to intersect the boundary polygon with 3-D planes such that the boundary polygon has the corresponding segments and heights. We have developed an algorithm to deal with vertical façades on roof tops. The final complex roof is modelled by a regularized and simplified boundary polygon and a number of interior critical points as shown in Figure 3. With very dense and accurate LIDAR point clouds, AFE can extract 3-D side models accurately.



Figure 3. A complex building with more than 50 sides, extracted from a LIDAR point cloud: stereo images are used only to verify the accuracy of the extracted building.

3. CASE STUDIES

3.1 High-resolution LIDAR

In our first case study, we used a LIDAR data set with a post spacing of 0.2 meters or 25 points per square meter. The LIDAR data set was converted into a GRID format with a post spacing of 0.1 meters. We recommend using half of the original post spacing when converting from LIDAR LAS format into GRID format for our AFE software. AFE used the following set of parameters: minimum building height 2 meters; minimum building width 5 meters; maximum building width 200 meters; roof detail: 0.4 meters; enforce building squaring on.



Figure 4. AFE extracted 59 buildings and 13 trees from dense LIDAR. White lines are building boundaries.

AFE transforms a GRID DSM into a GRID DEM using parameters 1, 2 and 3 as the first step. As an alternative, interactive terrain editing tools can be used to transform a GRID DSM into a GRID DEM first, and then both DSM and DEM are used as inputs to start AFE. Parameters 1, 2 and 3 define the dimensions of 3-D objects that are of interest. Roof detail determines the number of triangles used to model a complex roof. With more triangles, we model a complex roof more accurately. On the other hand, more triangles take more processing power, memory and disk space. We recommend that the roof detail parameter should have a value close to twice the relative linear error of the DSM. We used a photogrammetric

project covering the same area to verify and compare the 59 buildings and 13 trees extracted by AFE, as shown in Figure 4. The root mean square error of building boundaries is about 0.2 meters or one post spacing.

3.2 Pennsylvania State LIDAR project

This is a LIDAR project with an average post spacing of 1.2 meters for the state of Pennsylvania. The post spacing along the scan lines is quite different from the post spacing perpendicular to the scan lines. One is about 0.9 meters and the other, about 1.5 meters. We converted the original LIDAR LAS files into our internal grid format with a post spacing of 0.46 meters (one half of the original smaller post spacing). There is a total of 266,933,400 (20,010 x 13,340) posts covering a mountainous area of 55.7 square kilometers in Allegheny County, Pennsylvania. AFE extracted 14,181 houses and buildings, and 79,067 individual trees in 9 hours and 27 minutes using 4 threads on 4 CPUs at 3 GHz each. Out of the 9 hours and 27 minutes, 2 hours and 20 minutes were for the 3-D buildings/houses and trees extraction, and 7 hours and 7 minutes (60%), for the DSM to DEM transformation. Figures 5-8 demonstrate the results.

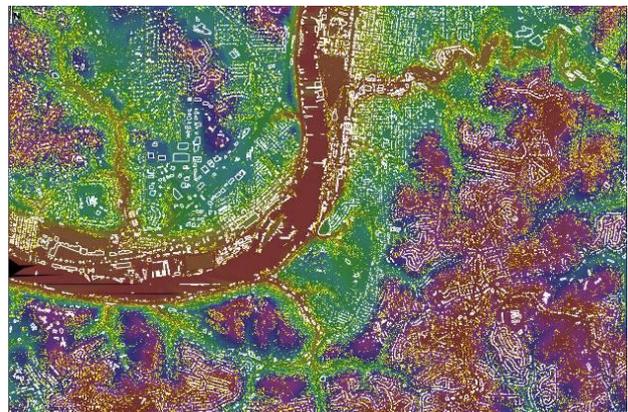


Figure 5. TSR covers 55.7 square kilometers in a mountainous area.

As shown in Figure 5, the steep terrain makes the DSM to DEM transformation challenging. One of the user-selected parameters that control the DSM to DEM transformation is the maximum building width. With a large value such as 300 to 500 meters, the transformation can detect and remove large buildings such as the one on the waterfront in Figure 5. This is only a problem in mountainous areas. This large value, however, can also chop off the top of a hill. As a result, we recommend using a smaller value in mountainous areas even if large buildings are not extracted. We used the following parameters for AFE: minimum building height 2.5 meters; maximum building width 300 meters; minimum building width 4 meters; roof detail 0.6 meters; enforce building squaring: on. It should be noted that LIDAR has blunders on water surfaces. As a result, the relief on the river (center) is not flat and false buildings have been extracted.

With 0.9-1.5 meters post spacing, small houses cannot be extracted by AFE. Most houses that are large enough have been extracted by AFE (Figure 6). For small houses, AFE needs at least 4 points per square meter. Even for the houses extracted, there is not enough detail due to the post spacing limitation. At this post spacing, only large houses such as the two on the lower portion are accurate enough for GIS mapping applications. AFE cannot extract forests. Dense tree canopy

areas are still an unsolved problem for our DSM to DEM transformation. It is a challenge to distinguish a tree from a house. There are cases when a tree is classified as a house.

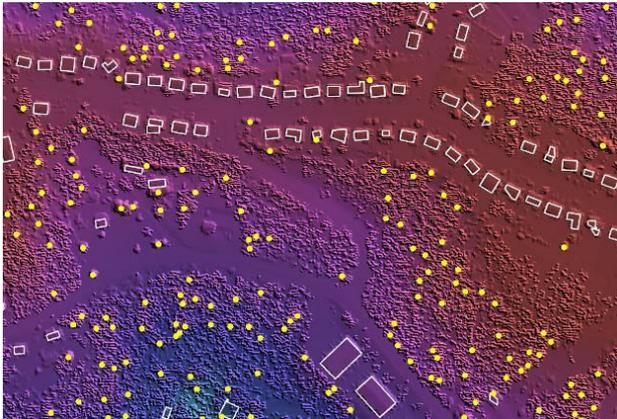


Figure 6. Buildings, houses, and trees at 1:1 resolution.

AFE can extract buildings, especially flat buildings with parallel or perpendicular sides, very accurately (Figure 7). In urban areas with flat terrain, AFE can perform much better than in mountainous areas. We recommend that users separate flat urban areas from mountainous areas when running AFE such that different parameters and strategies can be used.

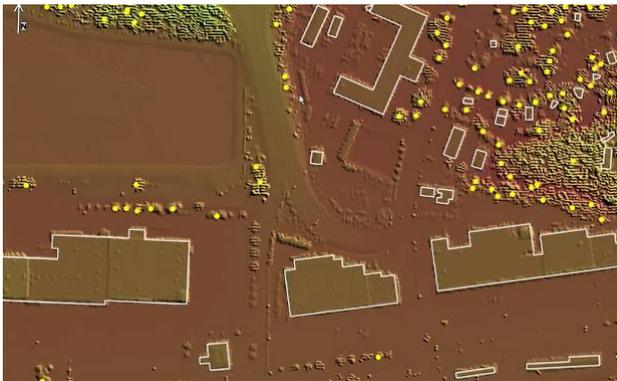


Figure 7. Extraction of buildings in flat areas

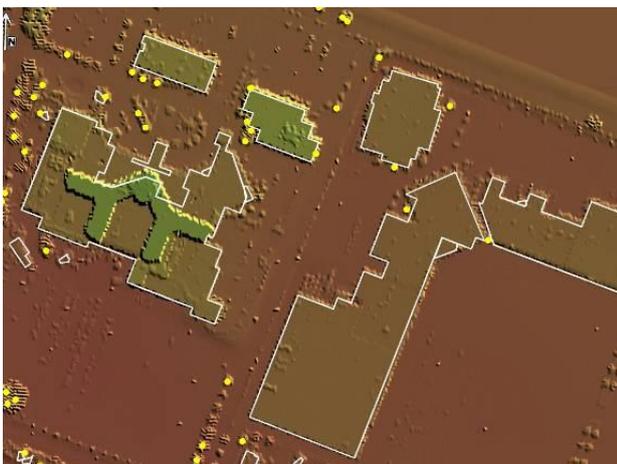


Figure 8. Complex buildings with irregular sides.

Complex buildings with irregular sides that are not parallel or perpendicular to each other still a challenge for AFE. In the AFE GUI, there is an option “Enforce Building Squaring.” When the vast majority of the buildings and houses have

parallel and perpendicular sides, users should turn this option on. The consequence is that the non-parallel sides may not be extracted correctly as shown in Figure 8.

3.3 Campus of the University of Southern California

This is a LIDAR project provided by USC’s Integrated Media Systems Center (IMSC) with an average post spacing of 0.4 meters for the USC campus (Figure 9). There is no LIDAR data in the lower right corner area, which is either black or uniformly red. The LIDAR point clouds were converted into a SOCET GXP internal grid format with a post spacing of 0.18 meters. There is a total of 138 million posts. The USC campus is rather flat, but there are many trees surrounding buildings. It is difficult when surrounding trees have similar heights to the building height.

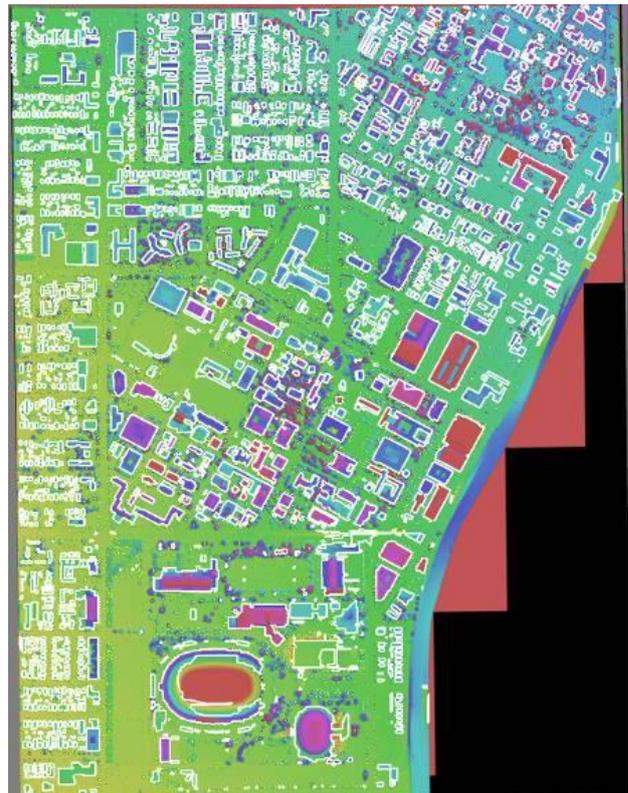


Figure 9. TSR of USC campus covers 24.8 square kilometers

AFE extracted 2464 buildings/houses and 5164 trees (Figure 10) in 1 hour 12 minutes with 4 CPUs at 3 GHz each. The time does not include transforming the DSM into a DEM. We used the following parameters for AFE: minimum building height 2 meters; maximum building width 300 meters; minimum building width 3 meters; roof detail: 0.4 meters; enforce building squaring on.

The area is relatively flat and the transformation from DSM to DEM is easier than the Allegheny County area. There are many trees in the center that are difficult to separate from buildings because they overhang the buildings or have similar heights to the buildings and are attached to them. AFE separated these trees reasonably well from the buildings.

AFE cannot extract buildings such as the football stadium and the track field (Figure 11), which are difficult for the DSM to DEM transformation. They do not have sides that are parallel or perpendicular to each other.



Figure 10. Trees are difficult to separate from buildings

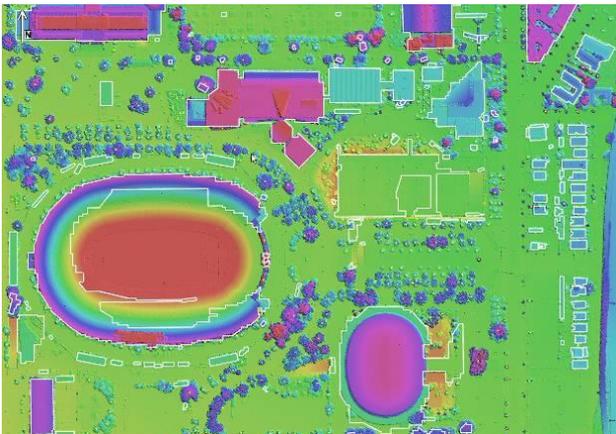


Figure 11. AFE cannot extract the football stadium and the track field

Buildings and houses with parallel or perpendicular sides are straightforward for AFE (Figure 12), but it may have difficulty when they less than 3 meters in height. Low 3-D features are difficult for the DSM to DEM transformation.



Figure 12. Rectangular buildings and houses easily extracted.

3.4 Luzern, Switzerland

This is a LIDAR project of Luzern, Switzerland with an average post spacing of 0.3 meters (Figure 13). The LIDAR point clouds were converted into a SOCET GXP internal grid format with a post spacing of 0.16 meters. There is a total of 475 million posts covering approximately 12 square kilometers. The LiDAR is very dense and high quality, but the buildings are

very complex due to the time period of the architecture. Many have interior holes in them which proved quite challenging. AFE extracted 2193 buildings/houses and 5225 trees. Figures 13-14 show the results. We used the following parameters for AFE: minimum building height 2 meters; maximum building width: 200 meters; minimum building width: 3 meters; roof detail: 0.5 meters; enforce building squaring on.

AFE allows user to define an area of interest (AOI) polygon such that only features within this AOI are extracted. This has proved especially useful for the Luzern project, where some building roofs are so complex that the elevation variations are very similar to trees. Since AFE uses elevation variation to differentiate buildings from trees, the threshold value which is determined by the roof detail parameter is difficult to set. With the AOI capability, users can divide the entire area into several regions and use the appropriate set of parameters within each region to run AFE most effectively. We expect that with this use of AOIs, the results could be much better.

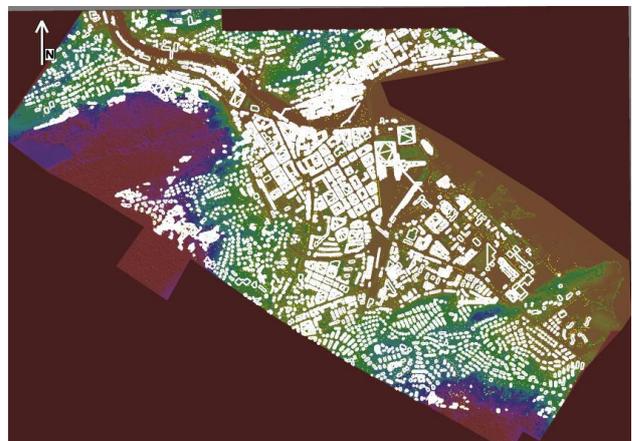


Figure 13. TSR covers 12.2 square kilometers in Luzern, Switzerland (data courtesy of Leica Geosystems)

Work is still needed to capture a complex building with interior open space accurately (Figure 14). AFE currently does not have logic to precisely extract buildings with holes.



Figure 14. Complex building with interior hole

4. ONGOING WORK

Estimating a figure of merit (FOM) for an automatically extracted feature is very important for production because AFE

is not and will not be perfect. In automatic terrain extraction, we use the cross-correlation coefficient as the FOM for each elevation post, which is a good indication of reliability. In AFE, we are investigating candidate FOMs. For each building/house, AFE computes 20 attributes, for example: maximum roof surface fitting error; average roof surface fitting error; maximum roof boundary fitting error; average roof boundary fitting error; percent of parallel or perpendicular sides; number of sides; maximum roof slope; average roof slope. We are hoping that a combination of these attributes can serve as a FOM for AFE. Once such a FOM is identified, it would provide a threshold, enabling a user to delete those features with questionable FOM values. This would result in a more reliable set of features that require little editing. Users could then manually extract the missing features. In production, editing a feature is more expensive than extracting a new feature.

While the results to date are very promising, therefore, additional work is being undertaken in several areas. Our experience underlines the value of being able to apply the algorithms to an area of interest in order to accommodate regions that would benefit from different parameters, such as the mountainous versus flat sections in the second case study. Secondly, efforts are being invested in improved modeling of complex rooftops and more accurate extraction of complex buildings, such as those typical in the Luzern data set. Thirdly, we have explained that it would be beneficial to have a quality assurance tool to identify potentially incorrect buildings. While the algorithms currently concentrate on building and tree extraction, Finally, other types of features could be extracted in the future such as power lines, dense tree canopies, and other volumetric objects: we are considering developments that would address these applications.

5. SUMMARY

Autonomous systems such as unmanned ground vehicles and unmanned airplanes are gaining traction for two reasons: they are in demand; and they are technically achievable. The geospatial community has been focusing on making “static” maps or non-real-time maps for decades. We anticipate that real-time 3-D mapping may have much wider applications than static maps. With massive parallel processing power such as GPU computing (a Tesla GPU card can have 448 processing cores with a double precision floating point capability of 515 Gflops) real-time 3-D mapping is technically achievable. Our study indicates that we can automatically recognize two types of 3-D features (buildings/houses and trees) from LIDAR point clouds. We expect AFE may recognize more types of 3-D objects or any 3-D objects that are above the ground and have certain sizes in the future. The core algorithms of AFE could be used to develop a real-time 3-D mapping system. Such a system could then be used to navigate unmanned ground vehicles.

The biggest challenge is the real-time requirement. We are not even close to real-time. Some of the algorithms are computationally intensive. Therefore, massively parallel processing may be required. Fortunately, the computing industry is moving rapidly toward GPU computing and parallel computing. For example, Microsoft Internet Explorer 9 was developed using GPU and parallel computing. The CUDA language from nVIDIA is gaining popularity in the software industry.

6. REFERENCES

- Douglas, J., Antone, M., Coggins, J., Rhodes, B.J., Sobel, E., Stolle, F., Vinciguerra, L., Zandipour, M., Zhong, Y., 2009. Contextual object understanding through geospatial analysis and reasoning (COUGAR). In: Sadjadi, F.A., Mahalanobis, A. (eds.), *Automatic Target Recognition XIX, Proceeding of the SPIE*, Vol. 7335, pp. 6-11.
- Zhang, B, 2008. Navigating a robot in complex terrain and urban environments with real-time 3-D mapping, *E&IS Fellows Periodical*, BAE Systems, September 2008.
- Zhang, B, Smith, W., 2010. From where to what – image understanding through 3-d geometric shapes, *ASPRS 2010 Annual Conference*, San Diego, California, USA, unpaginated CD.
- Zhang, B., Walter, M., 2009. Stereo image matching – breakthroughs in automated change detection, *Earth Imaging Journal*, 6(3), pp. 16-21.