3-D Object recognition from point clouds

Dr. Bingcai Zhang, Engineering Fellow
William Smith, Principal Engineer
Dr. Stewart Walker, Director
BAE Systems Geospatial eXploitation Products™
10920 Technology Place
San Diego, CA 92127-1874
bingcai.zhang@baesystems.com
william.smith4@baesystems.com
stewart.walker2@baesystems.com

Abstract

The future market for real-time 3-D mapping extends far beyond traditional geospatial applications and includes the navigation of an unmanned autonomous vehicle (UAV). Extensive parallel processes such as graphics processing unit (GPU) computing make real-time 3-D object recognition and mapping achievable. Geospatial technology 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. 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 the radiometric properties of an object vary considerably from image to image.

Geospatial information technology such as digital photogrammetry provides location. We believe the next breakthrough may be automatically identifying 3-D objects. 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 applications, it has the potential to be valuable in robotics and UAV applications.

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.

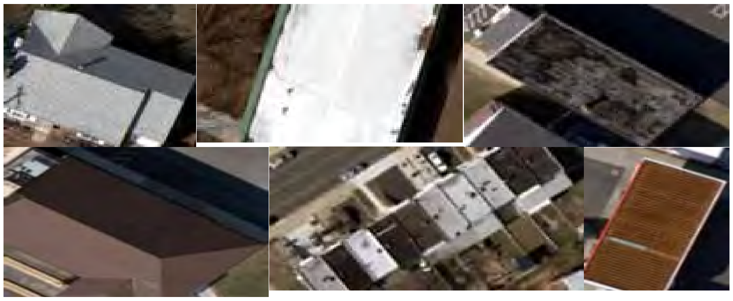


Figure 1. Six different building colors and patterns
For example, the supervised building region growing classification would need six signatures. Two (upper-right and lower-middle) of the signatures cannot be used because they are not very homogeneous. These six buildings are from one image (7500 x 11500) with a GSD of 0.14 feet.

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[1] software (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).

[1] NGATE is software used to extract 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.

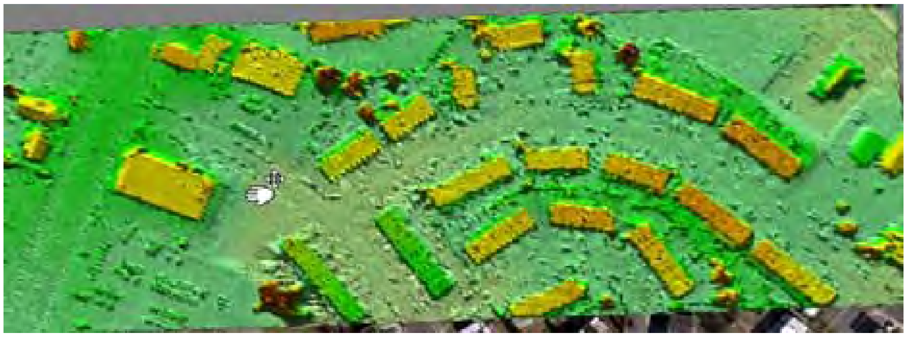


Figure 2. Terrain shaded relief of a point cloud
3-D objects are immediately apparent. Their locations and their approximate shapes are obvious. It is much easier to classify a TSR for 3-D objects than to classify digital images. The six buildings in Figure 1 are shown in this figure.

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 combine several specific algorithms to achieve a bare-earth model for 3-D object extraction:

Bare-Earth Algorithm	Description
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

A combination of the four algorithms above can transform a point cloud into a bare-earth model.

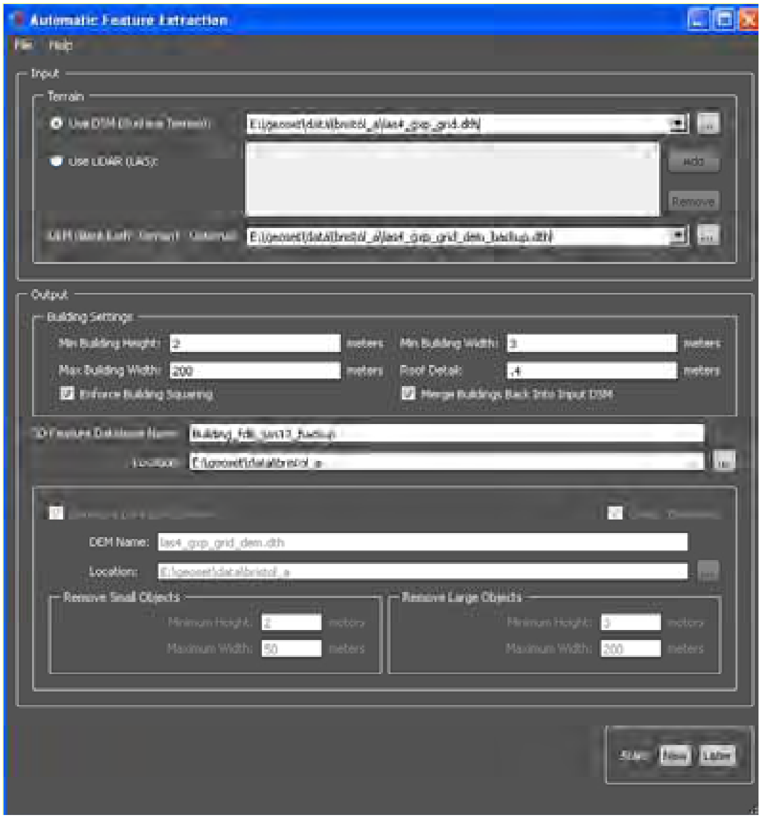
2.2 Automatic 3-D object extraction from point clouds using the bare-earth model

Over the past two years we have developed several key algorithms to extract 3-D objects from point clouds and bare-earth models automatically. Figure 3 shows the graphical user interface.

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. 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 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.
3. Tracing region boundaries: We trace the outermost points of a region of points to form a polygon boundary.
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.
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.

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 modeled by a regularized and simplified boundary polygon and a number of interior critical points as shown in Figure 5.



● Use DSM (Surface Terrain): E:\geoset\data\bristol_a\las4_gxp_grid.dth

○ Use LIDAR (LAS):

DEM (Bare Earth Terrain) External: E:\geoset\data\bristol_a\las4_gxp_grid_dem_backup.dth

Min Building Height: 2 meters Min Building Width: 3 meters

Max Building Width: 200 meters Roof Detail: .4 meters

☑ Enforce Building Squaring ☑ Merge Buildings Back Into Input DSM

3D Feature Database Name: Building_fdb_tes12_backup

Location: E:\geoset\data\bristol_a

DEM Name: las4_gxp_grid_dem.dth

Location: E:\geoset\data\bristol_a

Minimum Height: 2 meters

Maximum Width: 50 meters

Minimum Height: 3 meters

Maximum Width: 200 meters

Figure 3. AFE user interface

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

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.

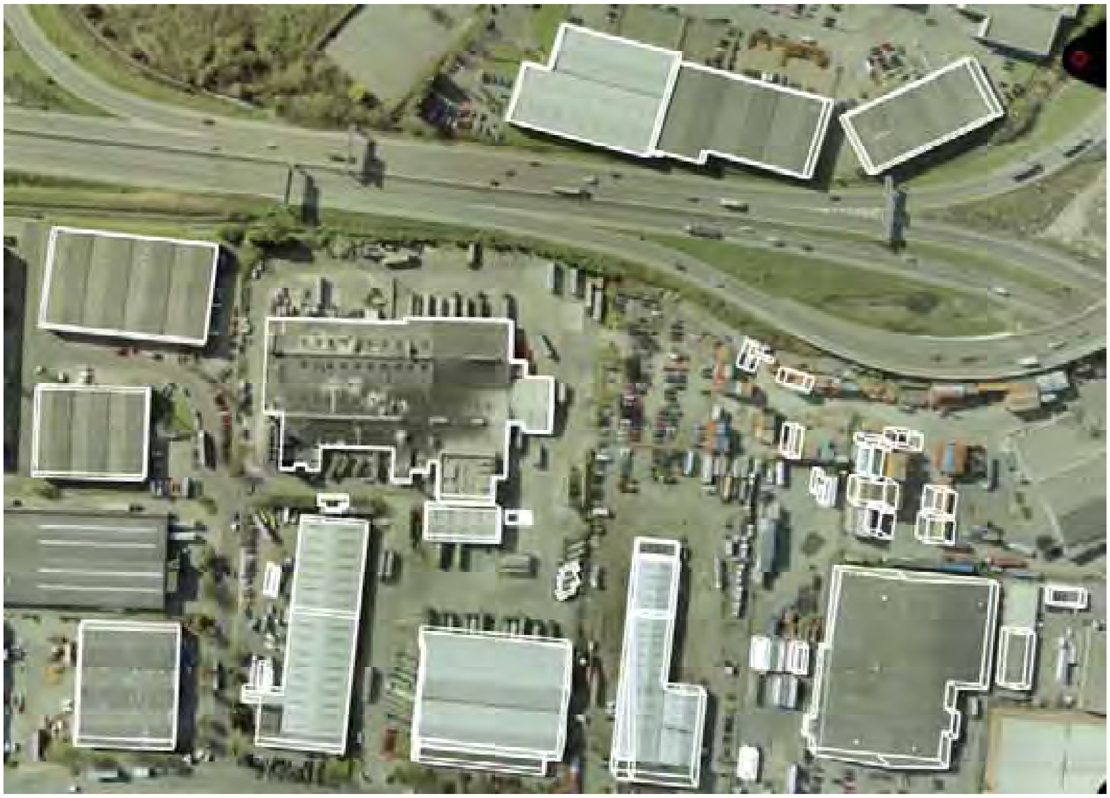


Figure 4. Case study one
AFE extracted 59 buildings and 13 trees. White lines are building boundaries. Yellow dots are trees (yellow dots are very small to see). The root mean square error of building boundaries is about 0.2 meters or one post spacing.

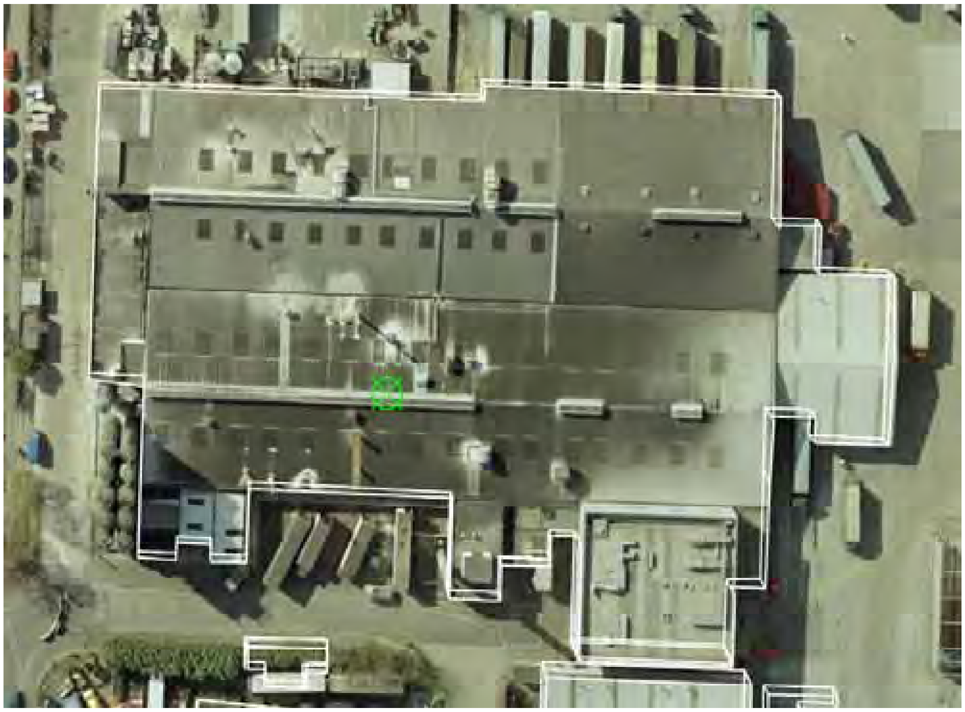


Figure 5. A complex building with more than 50 sides
Only LIDAR point cloud was used to extract this complex building. Stereo images are used only to verify the accuracy of the extracted building. With very dense and accurate LIDAR point clouds like this, AFE can extract 3-D side models accurately.

3.2 Pennsylvania State LIDAR project

This is a LIDAR project with an average post spacing of 4 feet 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 3 feet and the other, about 5 feet. We converted the original LIDAR LAS files into GXP internal grid format with a post spacing of 1.5 feet (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 21.5 square miles 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 or 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, for the DSM to DEM transformation. Figures 6 through 11 demonstrate the results. As shown in Figure 6, the area is not flat and this makes the DSM to DEM transformation challenging. One of the 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 at the cursor location in Figure 6. However, this large value 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

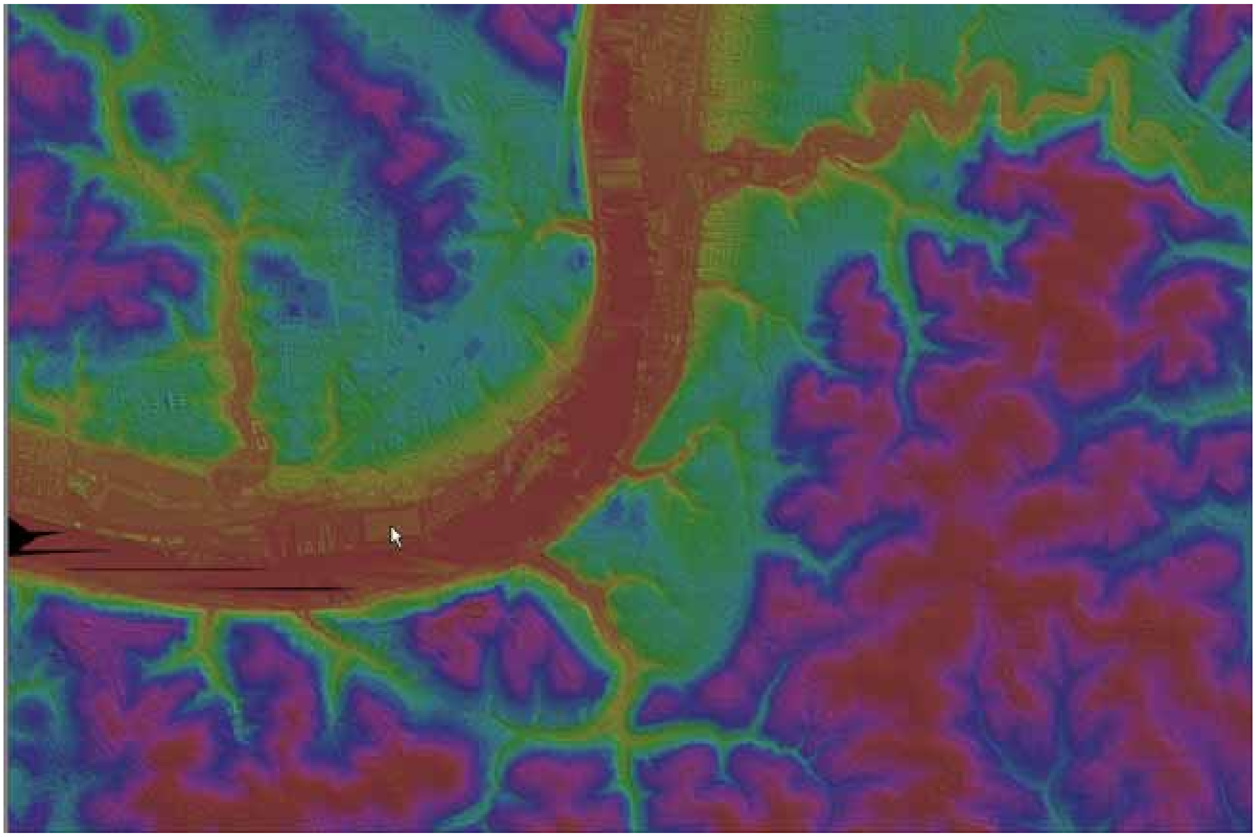


Figure 6. Terrain shaded relief covers 21.5 square miles in a mountainous area
The area is challenging for DSM to DEM transformation because of significant terrain slopes. The DSM to DEM transformation took more than 60% of the total computing time of 9 hours and 27 minutes. It should be noted that LIDAR has blunders on water surfaces. As a result, the terrain shaded relief on the river in the middle is not flat. Thus, false buildings have been extracted by AFE on the river as shown in Figure 7.

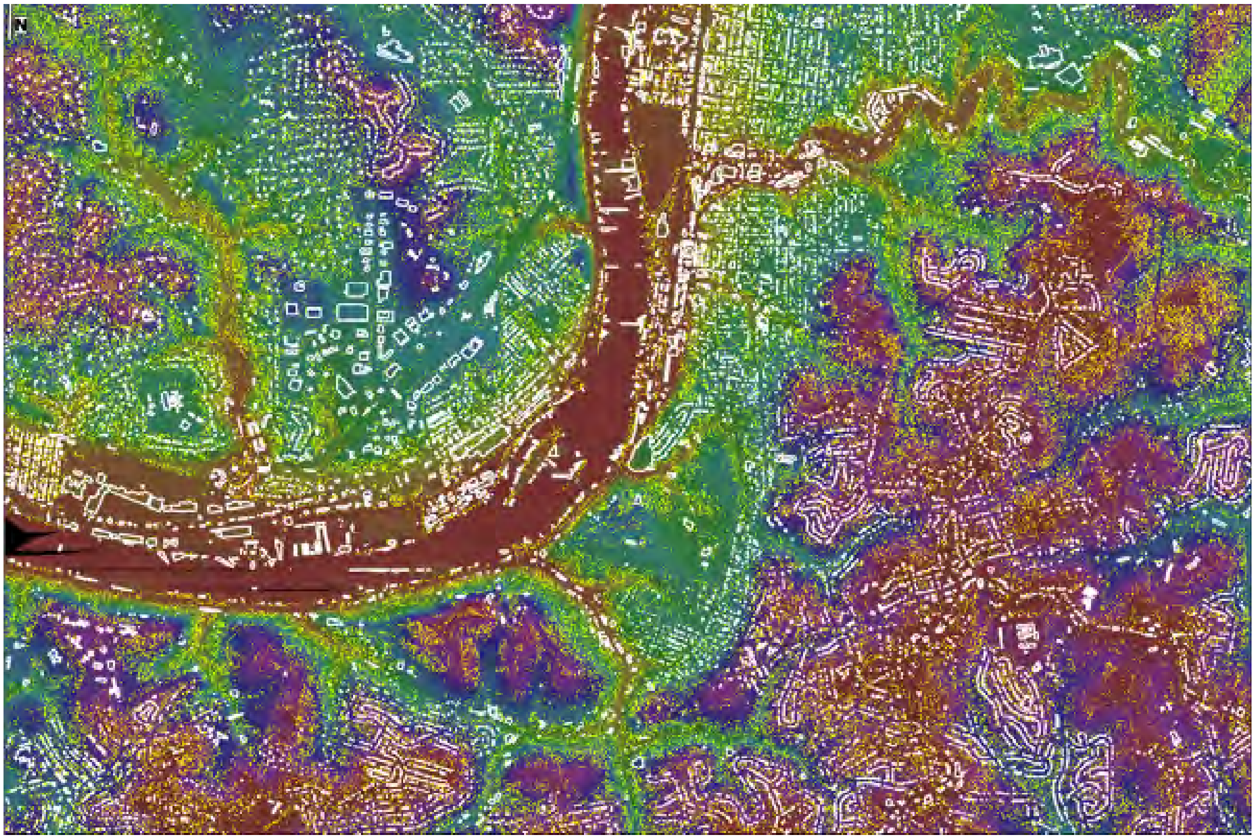


Figure 7. AFE extracted 14,181 buildings (white lines) and 79,067 trees (yellow dots) There are false buildings on water surfaces due to blunders in LIDAR point clouds. With 3 to 5 feet post spacing, small houses cannot be extracted by AFE. DSM to DEM transformation may miss extremely large buildings because of the user-selected maximum building width parameter. This is only a problem in mountainous areas. In relatively flat urban areas, users can always select a large enough maximum building width parameter to extract extremely large buildings without chopping off the top of a mountain.

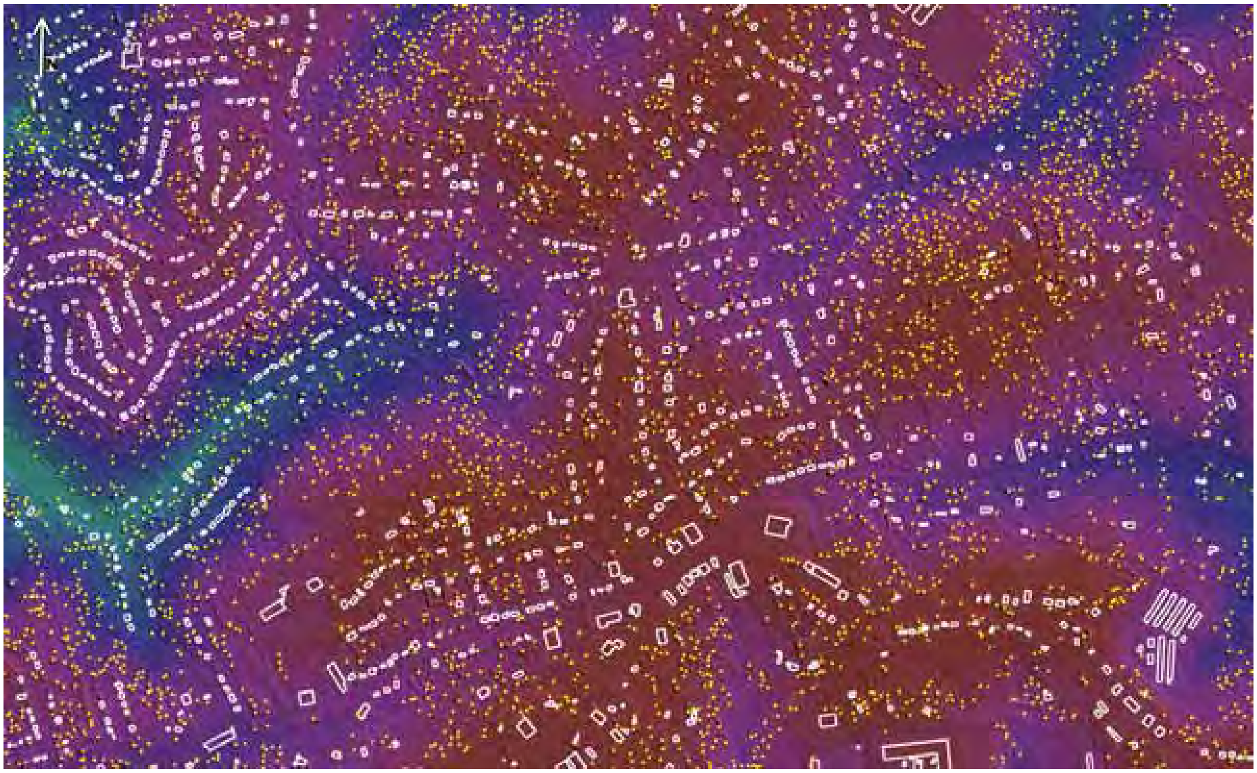


Figure 8. Buildings, houses, and trees at 4:1 resolution
Most houses that are large enough have been extracted by AFE. For small houses, AFE needs at least 4 points per square meter density. Even for houses extracted, there are not enough details because of the 3 to 5 feet post spacing limitation.

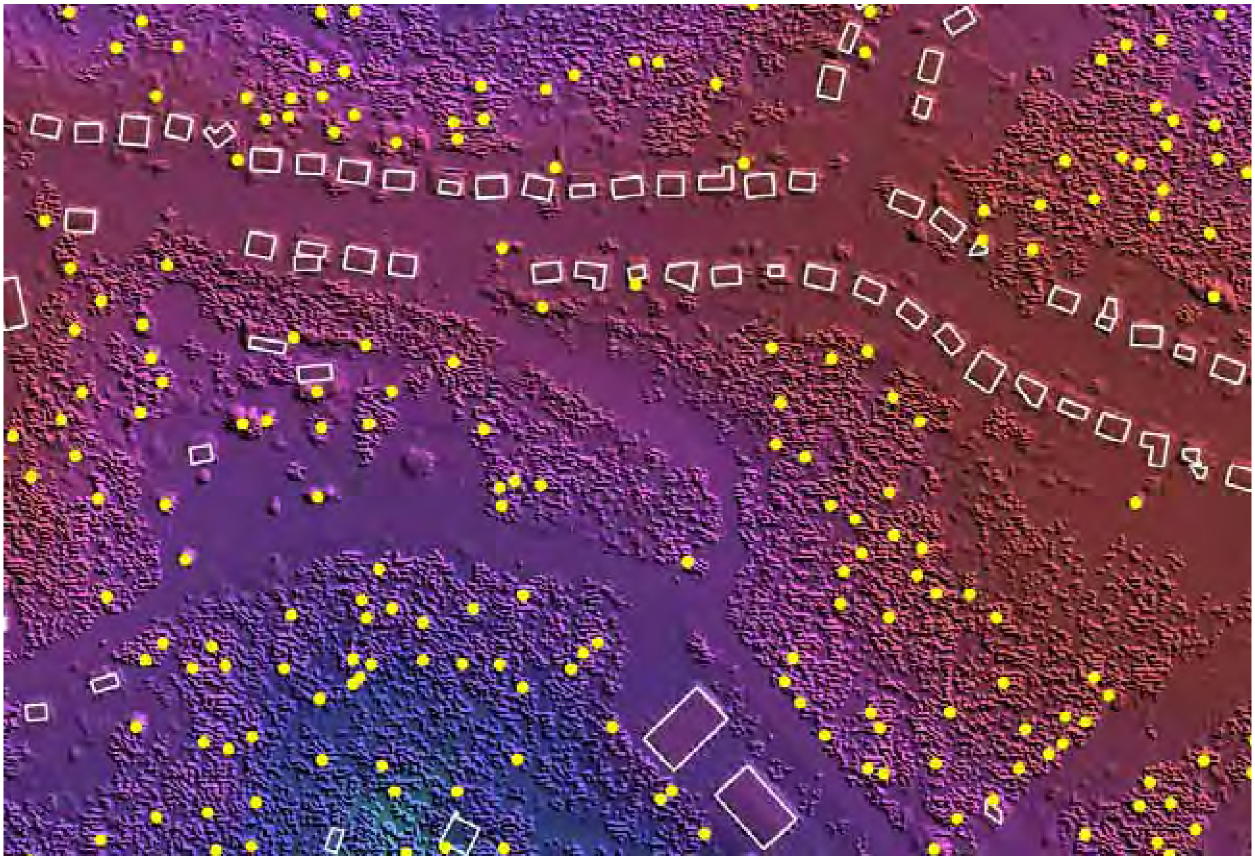


Figure 9. Buildings, houses and trees at 1:1 resolution
AFE cannot extract forests. Dense tree canopy areas are still an unsolved problem for our DSM to DEM transformation. For accurate house extraction, 3 to 5 feet post spacing is not dense enough. At this post spacing, only large houses such as the two on the lower portion are accurate enough for GIS mapping applications. It is still a challenge to distinguish a tree from a house. There are cases when a tree is classified as a house.

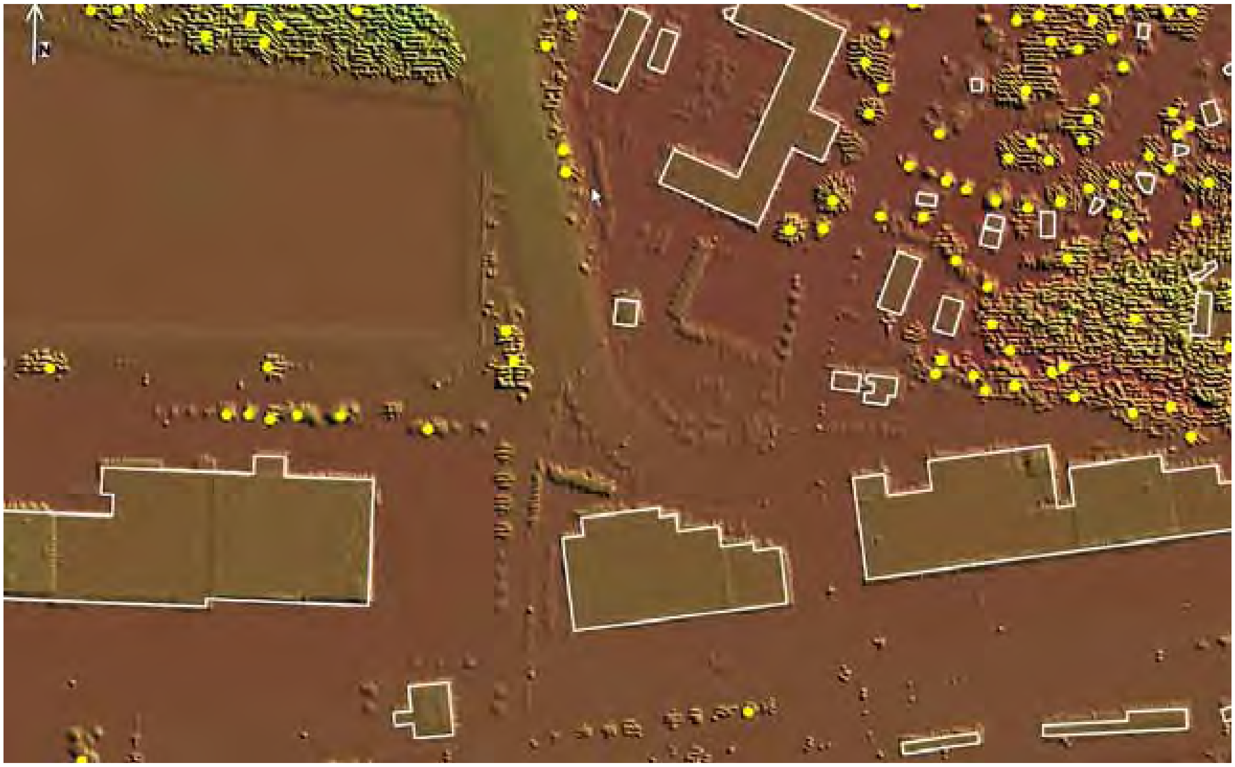


Figure 10. Extraction of buildings in flat areas
AFE can extract buildings, especially flat buildings with parallel or perpendicular sides, very accurately. In urban areas with flat terrain such as this, AFE can perform much better than in mountainous areas. We recommend that users separate flat urban areas from mountainous areas when running AFE such that different parameters and strategies can be used.

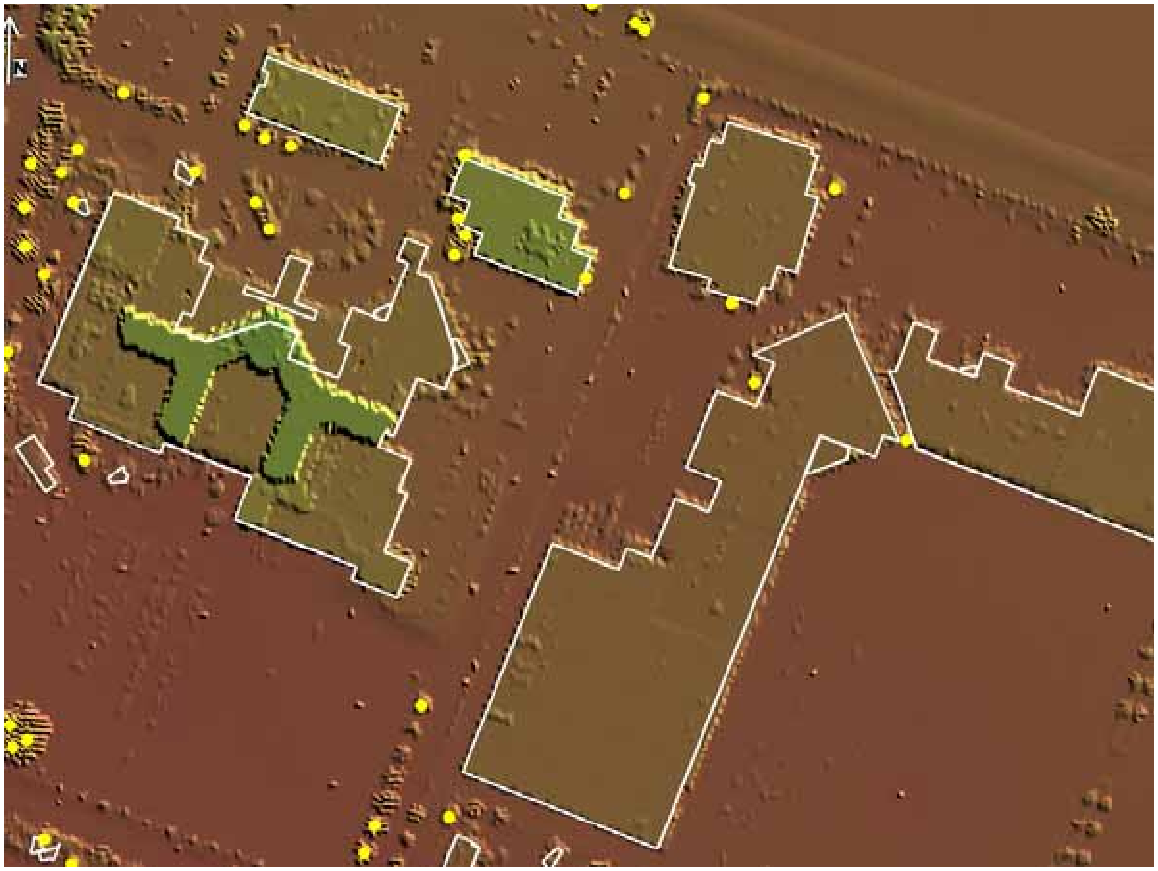


Figure 11. Complex buildings with irregular sides are still a challenge for AFE
Irregular sides are not parallel or perpendicular to each other. In the AFE GUI, there is an option "Enforce Building Squaring." When the vast majority of the buildings and houses have parallel and perpendicular sides, users should turn this option on. The consequence is that the non-parallel sides may not be extracted correctly as shown here.

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. The LIDAR point clouds were converted into a SOCET GXP internal grid format with a post spacing of 0.18 meters. There are a total of 138 million posts. The USC campus is rather flat, but there are lots of trees surrounding buildings. It is a difficult issue when surrounding trees have similar heights to the building height. AFE extracted 2464 buildings/houses and 5164 trees in 1 hour and 12 minutes with 4 CPUs at 3 GHz each. The time does not include transforming the DSM into a DEM. Figures 12 through 16 show the results achieved. 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

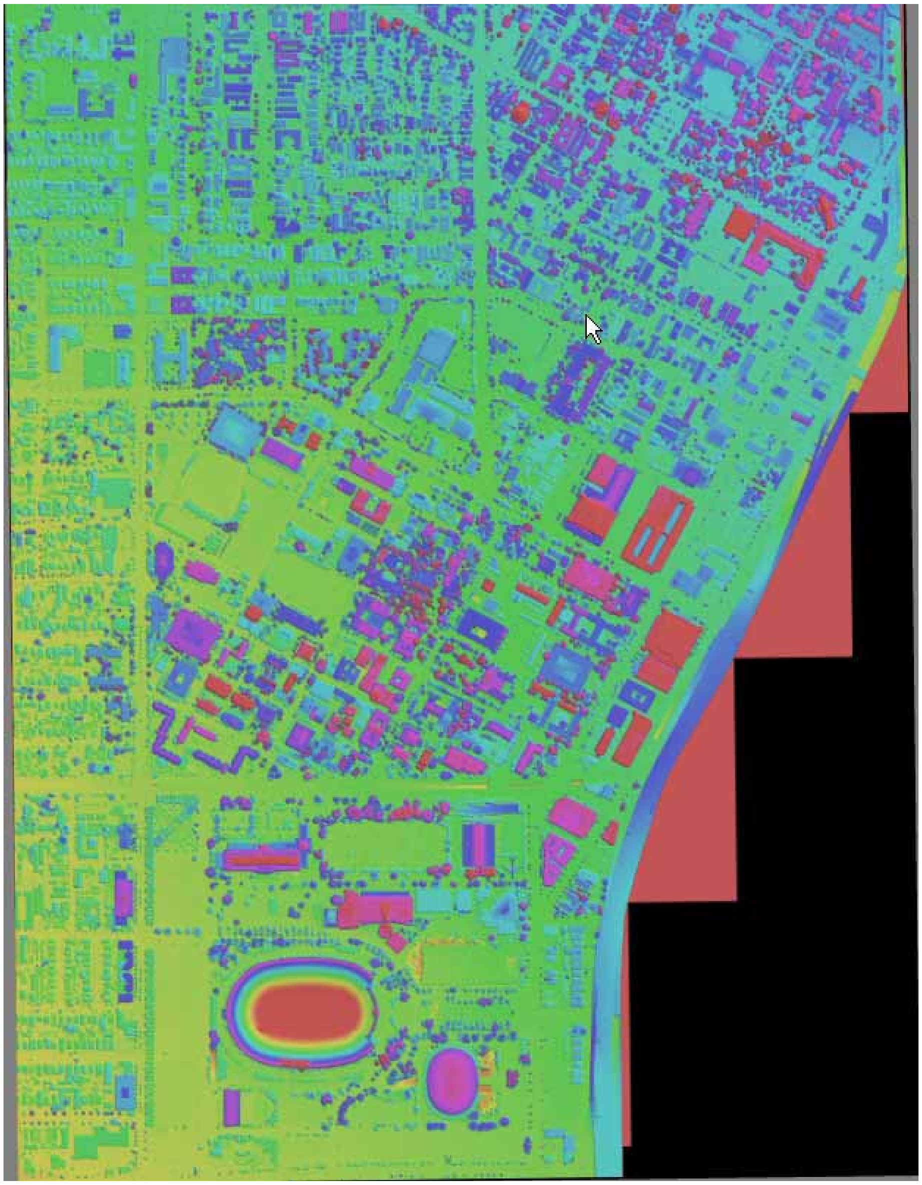


Figure 12. Terrain shaded relief covers 24.8 square kilometers of the USC campus The area is relatively flat and the transformation from DSM to DEM is easier than the Allegheny County area. AFE cannot extract buildings such as the football stadium and the track field. There are many trees. It is a challenge when trees overhang a building or are attached to a building with similar height. There is no LIDAR data in the lower right corner area, which is either black or uniformly red.

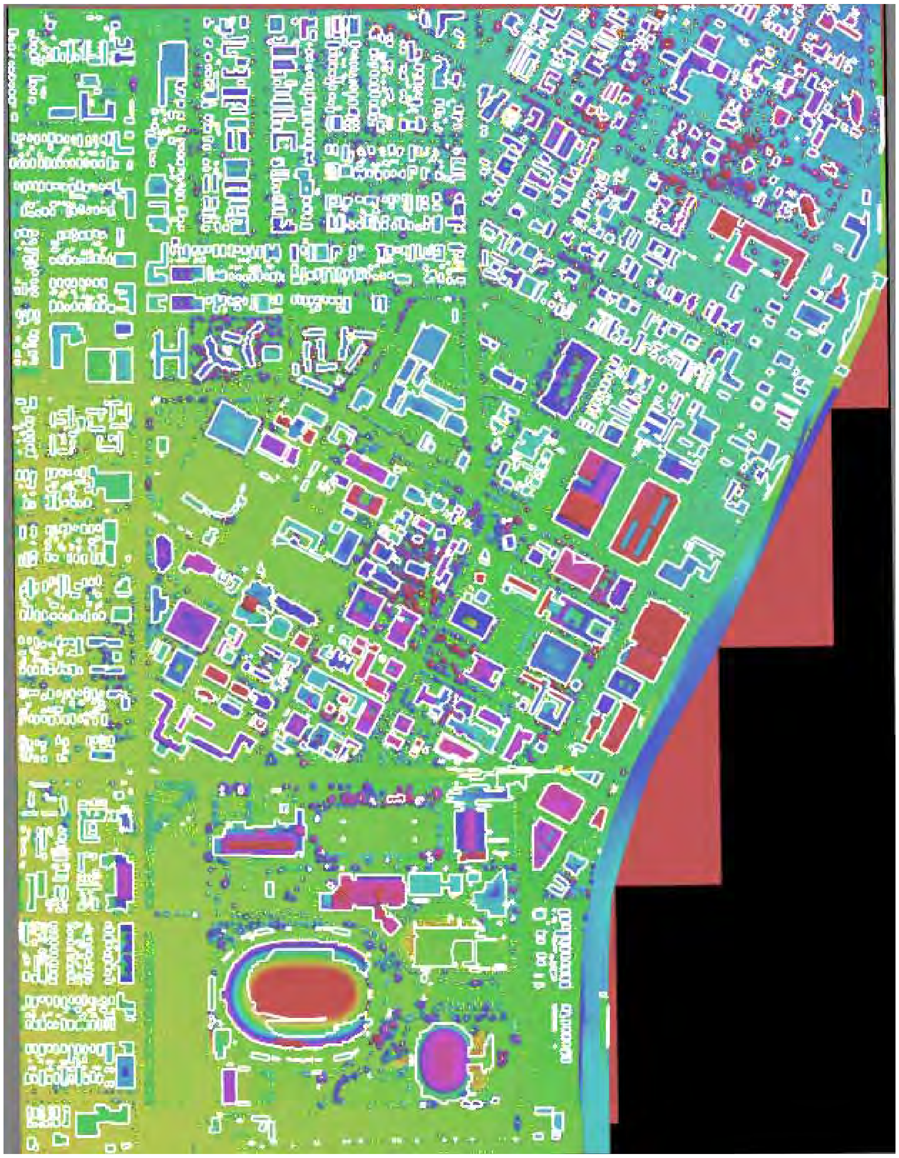


Figure 13. AFE extracted 2464 buildings/houses (white lines) and 5164 trees (yellow dots).

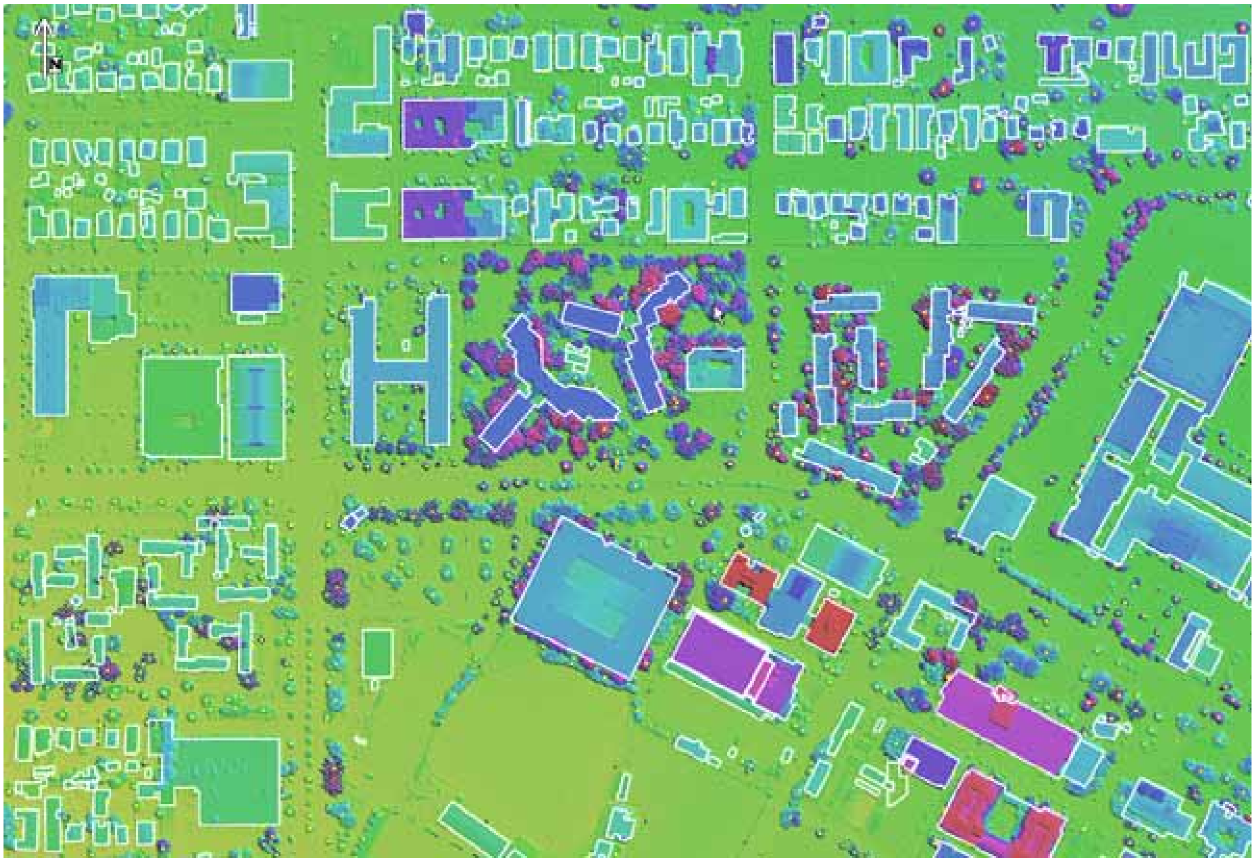


Figure 14. There are many trees in the center
These trees are difficult to separate from buildings because they have similar heights to the buildings and are attached to them. AFE separated these trees reasonably well from the buildings.

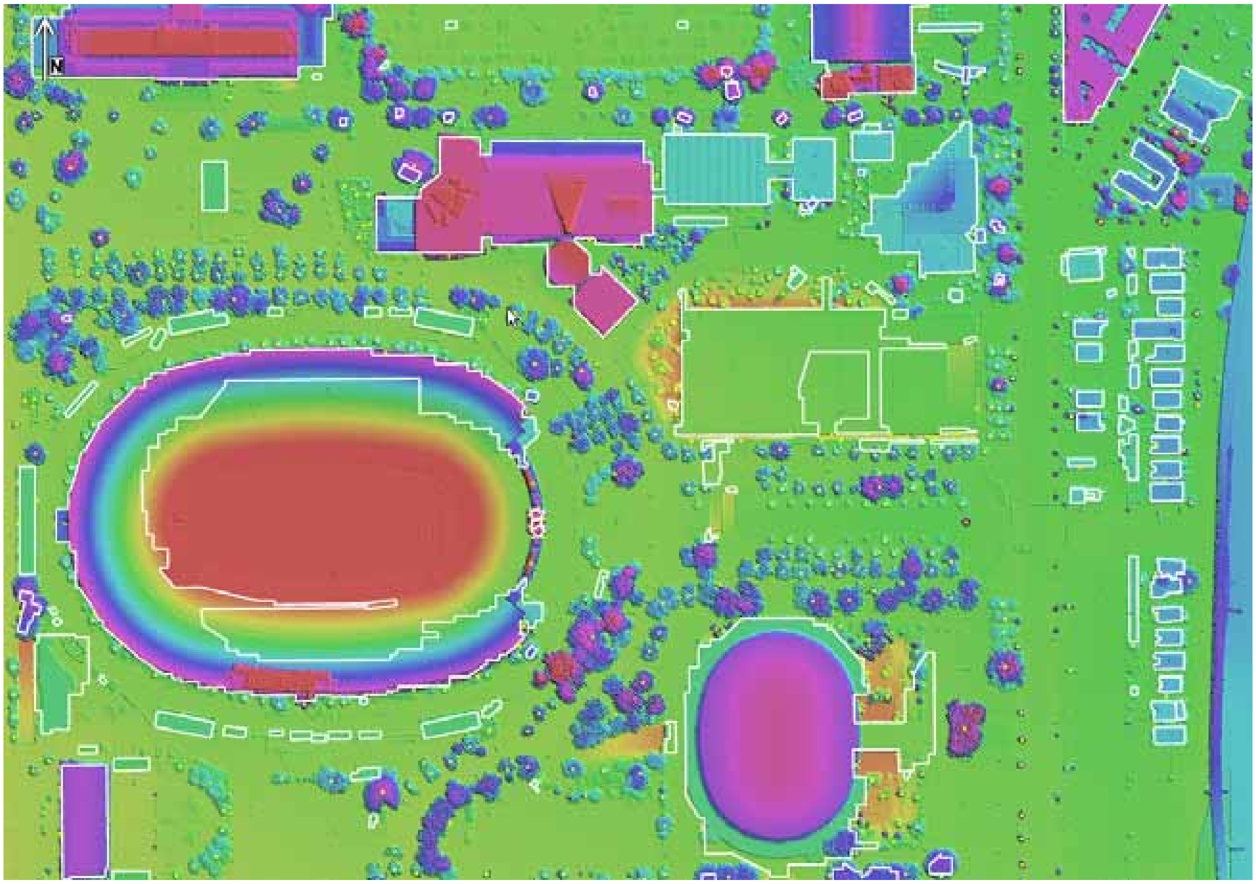


Figure 15. AFE cannot extract the football stadium and the track field
The football stadium and track field are difficult for the DSM to DEM transformation. They do not have sides that are parallel or perpendicular to each other. As a result, AFE did not extract them correctly.

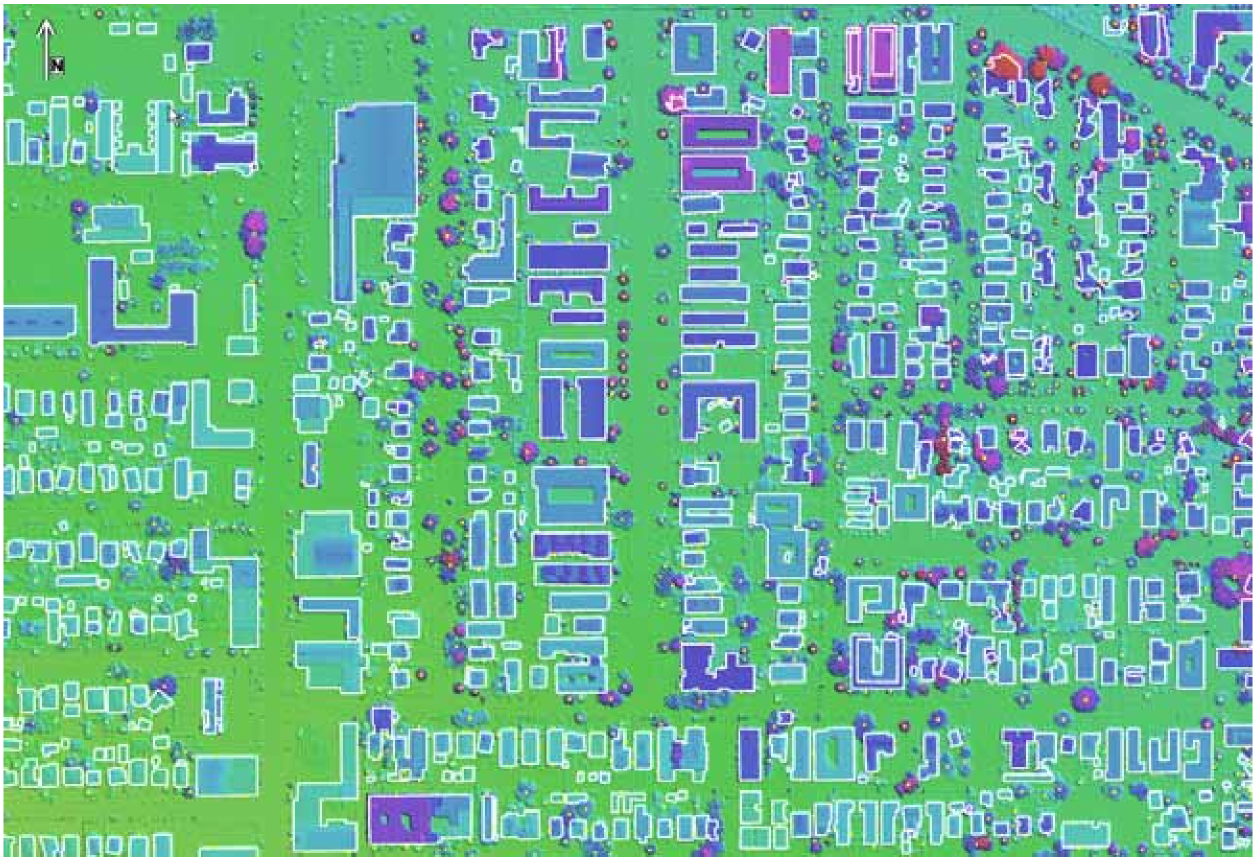


Figure 16. Buildings and houses with parallel or perpendicular sides
Buildings and houses with parallel or perpendicular sides are easy 3-D features for AFE. However, AFE may still have difficulty when the buildings and houses are less than 3 meters in height. Low 3-D features are difficult for the DSM to DEM transformation.

4 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.

5 Acknowledgement

Our thanks go to Carolyn Gordon for reviewing and editing this paper.

6 References

Zhang, B, and W. Smith, 2010. “From Where to What – Image Understanding Through 3-D Geometric Shapes”, ASPRS 2010 Annual Conference, San Diego, CA, unpaginated CD.

Zhang, B. and M. Walter, 2009. “Stereo Image Matching – Breakthroughs in Automated Change Detection”, *Earth Imaging Journal*, May/June, pp. 16-21.

Zhang, B, 2008. “Navigating a Robot in Complex Terrain and Urban Environments with Real-Time 3-D Mapping”, *E&IS Fellows Periodical*, September 2008.

Douglas, J., M. Antone, J. Coggins, B.J. Rhodes, E. Sobel, F. Stolle, L. Vinciguerra, M. Zandipour, and Y. Zhong, 2009. “Contextual Object Understanding through Geospatial Analysis and Reasoning (COUGAR)”, *Proceeding of SPIE*, Vol. 7335.