Evaluating 3D Building Extraction from Image-Derived Point Clouds

Katherine Grzedzicki
Chester F Carlson Center for Imaging Science
Rochester Institute of Technology
Ksg2511@rit.edu

20 May 2015

Abstract- Three-dimensional building models are essential for supporting a variety of applications such as urban planning, damage assessments, and for visualization purposes. In this paper, an automated 3D building extraction algorithm is evaluated using point clouds generated from airborne imagery and compared to algorithm-performance using lidar collected over the same area. Specifically, this algorithm identifies key features from rooftops, using those features to estimate building outlines. While originally created for use on lidar point cloud data, the algorithm was evaluated using point clouds generated from airborne RGB imagery. The image-derived point cloud was created using a structure from motion (SfM)-based software, Agisoft Photoscan, and georeferenced using co-located lidar points as ground truth. This point cloud data was then run through the algorithm after which, an extracted building file was created. This building file was compared to a building file created from lidar data over the same area. Preliminary results show that while many tall buildings were extracted from the imagery point cloud, several other buildings were missed due to horizontal normals affecting the Euclidean Clustering performance. By removing these horizontal normals, building extraction accuracy increased significantly. lidar data resulted in 100% building extraction for scenes tested. When horizontal normals were manually removed, imagery point clouds had almost duplicative building extraction results but manual error was introduced into the model affecting accuracy. For best performance, automatically removing horizontal normals in imagery-derived point clouds would significantly improve automated building extraction.

1. INTRODUCTION AND BACKGROUND THEORY

The use of 3D-derived imagery has increased significantly over the past decade. Though basic stereoscopic imaging has been around for over a century, more sophisticated models and approaches are being developed to optimize three-dimensional imaging. Specifically related to digital imaging, computer software engineers are researching the most current and effective approaches to create accurate 3D models and use those models to create intelligent, imagery-derived products.

Popular in fields such as forestry and urban planning, lidar (light detection and ranging) sensors are gaining notoriety for their high vertical accuracy and innate 3D exploitability. lidar point cloud scenes offer precise 3D representation, making modeling scenes easier and more cost-effective than manually digitizing stereo image pairs, as previously done. Several commercial imagery vendors, such as Google Earth, are eager to implement 3D imagery into their standard workflow. While lidar is designed to provide 3D data to its users, its limited global collection footprint makes it difficult to use exclusively. However, since optical sensors are widely available at a larger scale, using this data to create lidar-like point clouds allows for global 3D data to be accessible.

Currently there exist several fully-automated algorithms and software packages to create 3D images from a collection of 2D images. Using two or more images of an overlapping scene, optical imagery can geometrically be aligned to provide height information. While neither lidar point clouds nor image-derived point clouds are generally considered better than the other modeling, Leberl *et al.* did two tests to compare the two sources. Results showed that surface density is greater using image data to create point clouds, yet both data have comparable accuracy. The study goes on to show additional benefits of using imagery instead of lidar for modeling such as the ease of processing, the ability for error checking with redundant points, and larger area collects.

One common method of creating point clouds from imagery is using a structure from motion (SfM) process. This process typically involves detecting corresponding feature points between two overlapping areas in a scene and estimating three-dimensional positions. This paper looks at an evaluation of an automated building extraction from point cloud data, using imagery-derived point clouds created from SfM-based software. Two independent point cloud scenes were created from airborne imagery and run through the building extraction model to determine the algorithms effectiveness on image-derived-point clouds.

2. METHODOLOGY

2.1 Point Cloud Creation

The building extraction algorithm being tested was initially created using lidar point cloud data as input. To replicate results using aerial imagery, a 3D point cloud was created. Imagery was collected by the Wildfire Airborne Sensor Program (WASP), which is an imaging platform designed by the Rochester Institute of Technology (RIT) Center for Imaging Science and operated by Kucera International Inc. This sensor consists of a RGB, short-wave infrared (SWIR), mid-wave infrared (MWIR), and long-wave infrared (LWIR) cameras in addition to a high-resolution LIDAR. Two separate scenes were collected by WASP and used in this study to evaluate building extraction. The first scene of over four hundred RGB images was collected over an area of approximately 2.8km^2 in downtown Rochester, NY. This scene was co-collected with high-resolution lidar for direct comparison between the data sources. The second scene collected was 0.64km^2 over a key just south of Tampa, Florida. Only three images of this scene were collected and used to test success of the algorithm using minimum imagery for point cloud generation. All images were delivered prior to orthorectification in order for a point cloud to be generated.

2.1.1 Photoscan

The point cloud construction algorithm used in this study is a structure from motion (SfM)-based algorithm implemented by the Photoscan software package. This program implements several computer vision algorithms to generate an accurate and dense point cloud from two-dimensional imagery.

The workflow using Photoscan is straightforward and produces high-quality point clouds with minimal computational effort, depending on the input. This software loads images as input and if any EXIF information is retained in metadata, Photoscan automatically aligns the images. Since all the data used was obtained prior to orthorectification, the data required alignment. In order to align data, Photoscan uses SfM algorithms to determine the relative orientation of each point in order to calculate camera models of the scene. Once the camera positions are calculated, multi-view stereo algorithms are used for a 3D meshed surface computation. Since this step is most computationally intensive when using several high-resolution images, the user has the option of choosing a lower quality model. For all scenes used in this study, the highest quality metrics were chosen in Photoscan, despite computation time, in order to retain detailed rooftop features used in building extraction. After the 3D mesh is created, the model can be texturized based on existing input photographs. The final 3D model can be saved in a variety of community-accepted point cloud formats, such as LAS, PLY, or ASCII text.

At this point the 3D point cloud is expressed in a relative, local coordinate system. In order to establish both models into an absolute coordinate system, a minimum of three ground control points (GCPs) are needed for geo-referencing. These points were selected from ground points in the lidar model in the Rochester dataset, and from Google Earth for the Florida dataset. One helpful feature of Photoscan is that when a ground control point is entered into the software, it flags other images where that same GCP is located. These flagged areas can be manually edited if they are not pointing to the exact ground location that Photoscan approximated. It is recommended that GCPs be selected from several locations in the scene and that most images have a GCP associated with it for optimized geo-referencing. Coordinate information for these GCPs is entered in as either degrees minutes seconds or decimal degrees and using this information, Photoscan georeferences the scene to a Geodetic coordinate system, typically WGS84.

2.2 Data Preparation

Once the point cloud was created and georectified, some additional steps were needed before the algorithm could be run. The first step in data preparation was converting the Geodetic coordinates, created upon georectification in Photoscan, into UTM coordinates. This step is essential since the algorithm specifically reads UTM coordinates, and any other conversion would result in a skewed projection and poor results. Secondly, it is required that the data be saved in ASCII format with columns saving X, Y, Z information as well as any intensity information, if it was created. Once the data was in this format, it could be run through the algorithm successfully. It is recommended to check X, Y, Z point cloud data, regardless of source, to ensure that the ASCII columns as well as projection information is correct before running it through the algorithm.

While there are several approaches to converting point cloud models and changing the file type, commercial software, Quick Terrain Modeler, created by Applied Imagery was used. This software is commonly used as a 3D visualization program in addition to creating a variety of lidar-derived analytical products. However, Quick Terrain Modeler also has the unique features of quickly and efficiently projecting coordinate information, calculating point cloud statistics (such as density), as well as saving out point clouds into different file types.

2.2.1 Code Preparation

In order to be referenced correctly, the ASCII point cloud data generated for analysis needs to be placed in the correct folder structure located where the rest of the code is compiled. Additionally, based on compiler requirements, some of the java code needs to be compiled before a new user begins to run the code. This information is saved in an updated README text file located in the parent folder where the code is hosted (/cis/masters/ksg2511/sxs4643/Research).

2.3 Building Extraction Algorithm

The code being evaluated is hosted and compiled on a linux system with Matlab and C++ code being implemented. Upon loading, the point cloud is first passed through a noise filter calculated by a statistical outlier removal, followed by a calculation of point normals and curvatures for each remaining point. Then, the scene is classified by using a graph cuts optimization algorithm, which looks at the distribution of the normals previously calculated. To extract the base terrain from any above-ground features, a hierarchal Euclidean clustering approach is applied to the scene. After this step, buildings are isolated and classified. Since detecting and extracting complex rooftops, is a crucial step in building extraction, a region-ground segmentation method is applied to the extracted buildings to retain all unique rooftop features in modeling. Boundaries on the buildings are created through rectilinear fitting and building models are generated using a 2.5 dual contouring method.

The output of this method is four point clouds in ASCII text to represent different steps in the building extraction. One file represents the input scene input initially, while another represents the classified scene used after noise removal. The two remaining files generated are used in analyzing results. One file represents the terrain classification of all ground points in the scene. Ideally this scene will be exclusively ground points with voids representing buildings extracted. The last file isolates and displays building rooftops extracted in the code.

3 RESULTS AND DISCUSSION

3.1 Image-Derived Point Clouds

Successful point cloud generation from optical imagery in Photoscan is a function of the input images, overlapping features among co-located images, and preferences selected by the user during each step. Ideally the user would have hundreds of images, as we did in the Rochester case, which results in a high-density point cloud with several features. However it is notable that the Florida dataset, with only three images collected and used, still successfully created a dense point cloud. The most noticeable difference between the two models, was that the Florida scene had more voids in the point cloud than the Rochester scene did. This is partially due to the small quantity of input images as well as some missing areas in the overlap. Both datasets had good data density considering how many images were used in their creation (Figures 1-4). Both the Rochester and the Florida image-derived point clouds had an overall scene density of approximately 6 points per meter (ppm^2). This is surprisingly notable performance for the Florida scene where only three images were used in point cloud creation.

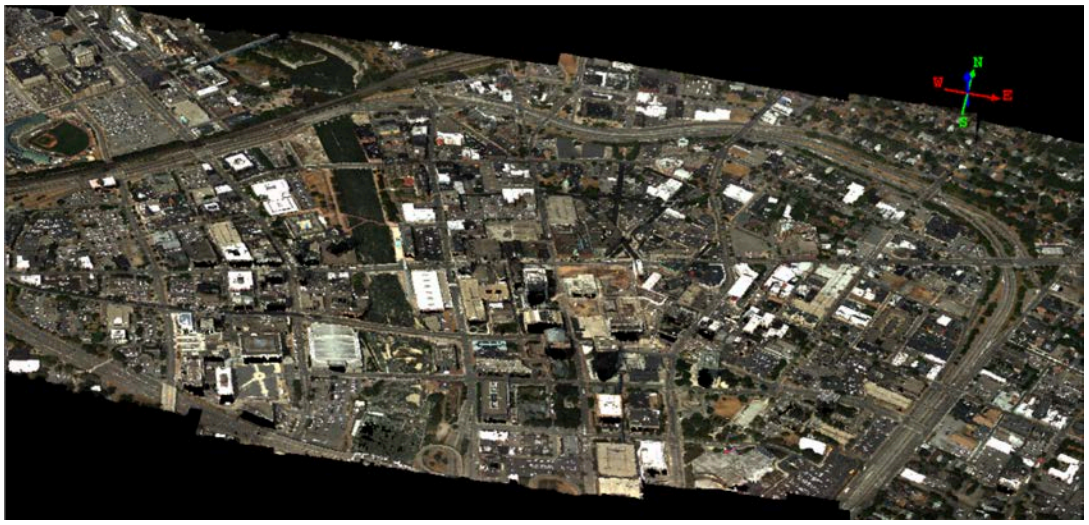


Figure 1: An overview of the downtown point cloud in Rochester, NY created with Photoscan.

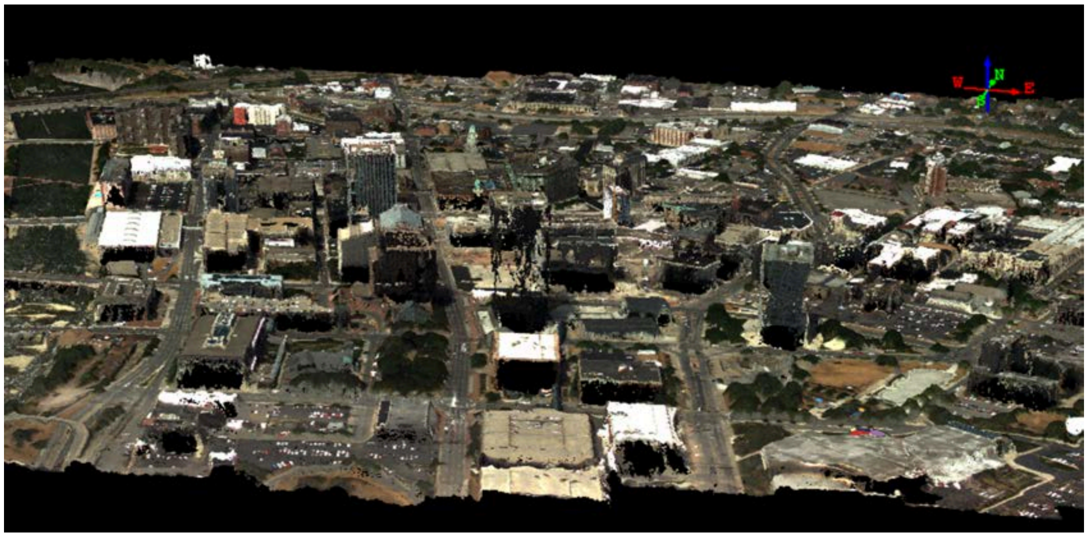


Figure 2: A more detailed, perspective view of skyscrapers in downtown Rochester. Some building sides can be perceived in this view, illustrating the detail retained in point cloud generation.

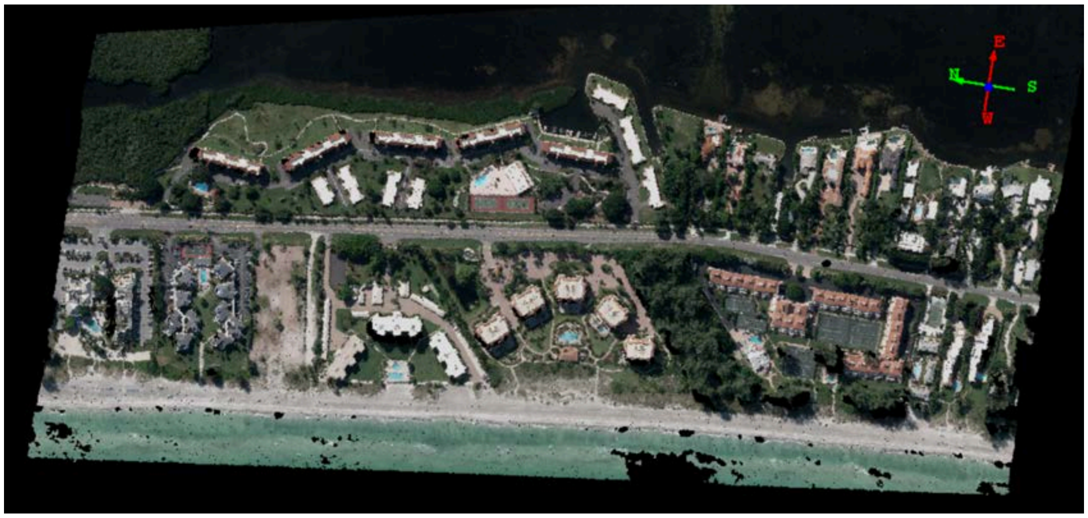


Figure 3: An overview of the Florida point cloud created with Photoscan. Several voids in this dataset occurred during point cloud processing over water points. This is mostly due to lack of overlapping imagery used in point cloud creation.

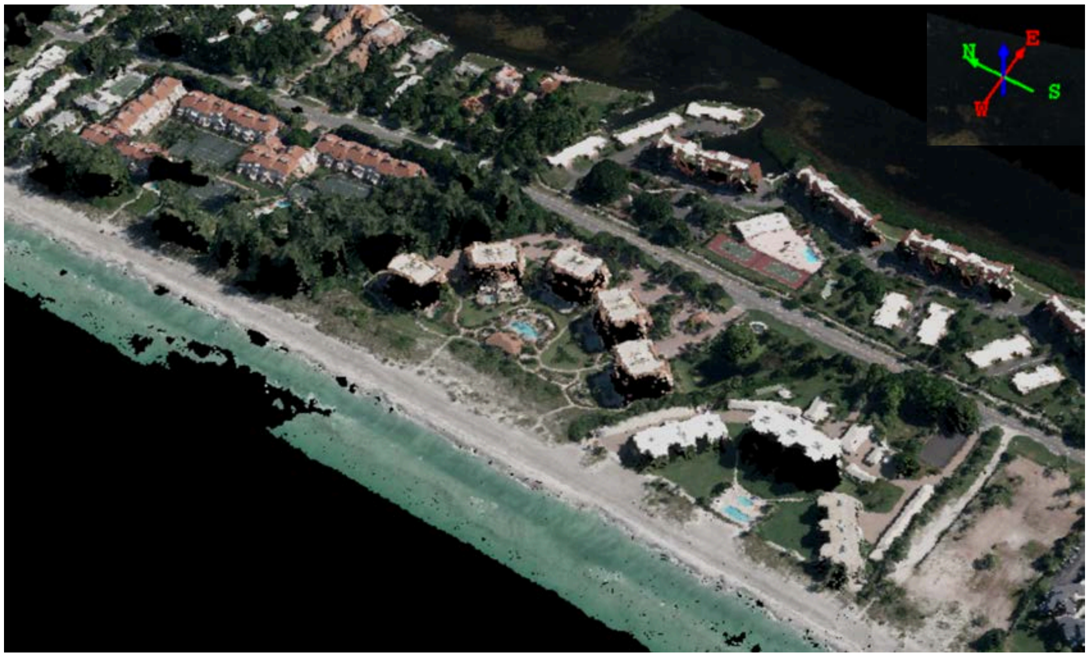


Figure 4: A zoomed in perspective view of the Florida image-derived point cloud. Most of this subset was used as input for the building extraction algorithm.

3.2 Building Extraction Analysis

Each scene used in analysis was analyzed by slightly varying metrics based on the data available. Since the Rochester dataset had both lidar and optical imagery, point density was measured for both sources post-classification to directly compare specific building extraction with each other. After the data is run through the algorithm, several output files were analyzed to qualitatively gauge performance. Displaying the lidar terrain file with image-derived point cloud building results overlaid, illustrates how well the optical imagery worked compared to lidar. Ideally the ground points would have perfectly linear voids where buildings exist. Taking these ground points with another output file of building points essentially fills in the voids, recreating the original model. A qualitative analysis of how the Rochester data performed used the terrain file created from the building extraction with lidar with the RGB building model overlaid. Wherever the voids still existed in the lidar terrain file, the image-derived point cloud failed.

Preliminary results from image-derived point clouds illustrated that several buildings were unable to be extracted in the scene. After evaluating the algorithm approach, it was discovered that based on the Euclidean clustering technique, sides of buildings were inhibiting successful building extraction. One attempt to solve this problem would be to remove the horizontal normals so that Euclidean clustering could isolate rooftops from the ground with no connecting points. Manually doing this for testing purposes provided promising results. (Figure 5). Automatically calculating and removing the horizontal normals is essential for successful building extraction using image-derived point clouds.

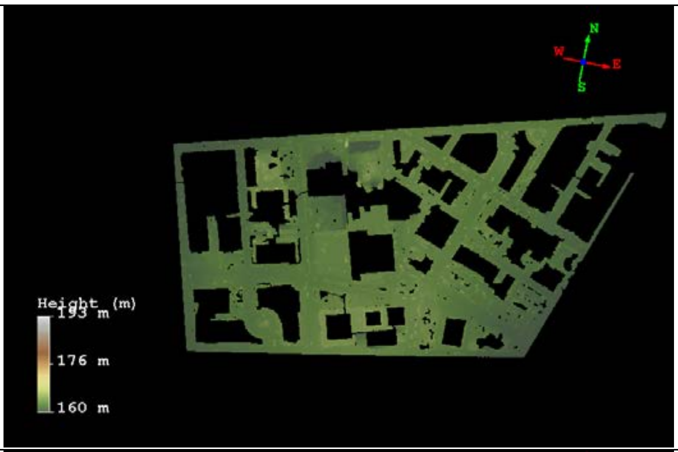
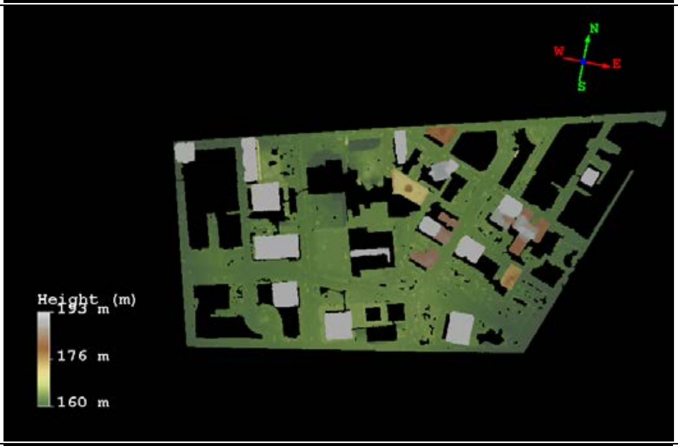
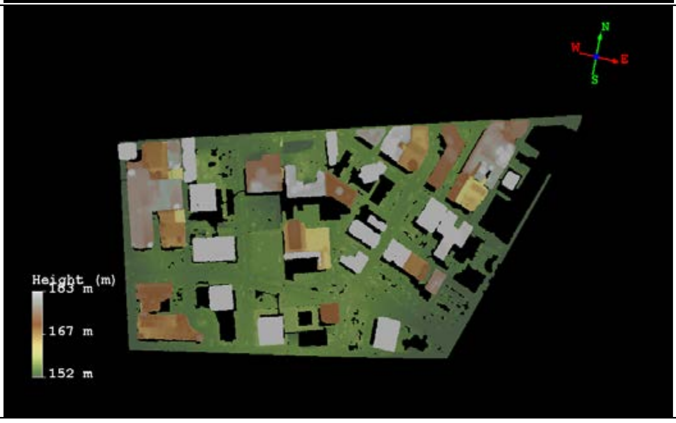
a) lidar terrain file post-building extraction algorithm. Voids in the data represent where buildings were extracted out.	
b) Image-derived point cloud results displayed over the lidar terrain file. Voids represent where this dataset failed to remove buildings. This is due to sides of buildings inhibiting successful Euclidean Clustering and resulting in buildings not being separated from the scene.	
c) Image-derived point cloud results displayed over the lidar terrain file after the horizontal normals were removed. This shows improved accuracy when compared with the original image-derived point cloud in (b). Small voids still indicate buildings that the image-derived point cloud missed but the lidar point cloud extracted.	

Figure 5: A qualitative comparison on how the image-derived point cloud performed against the lidar point cloud for building extraction in the Rochester scene. a) shows the lidar terrain file output from the algorithm where b) and c) show image-derived point cloud results for the data with and without horizontal normals (respectively).

Another evaluation technique was comparing the roof points and structures retained in three buildings extracted in both the lidar point cloud and the image-derived point cloud. (Figure 6)

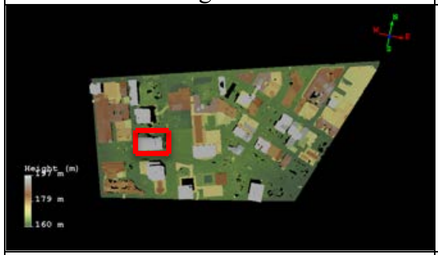
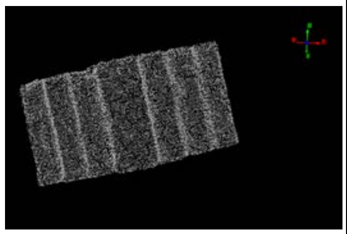
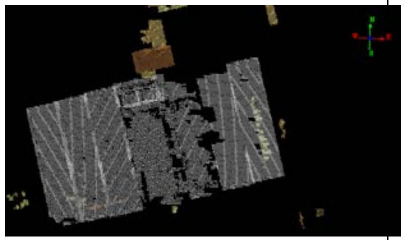
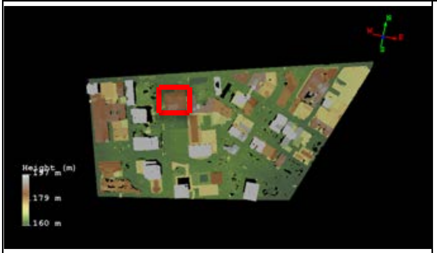
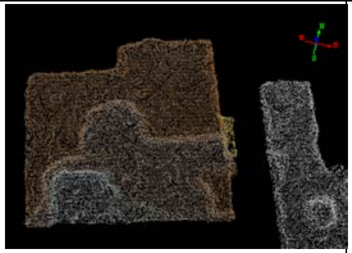
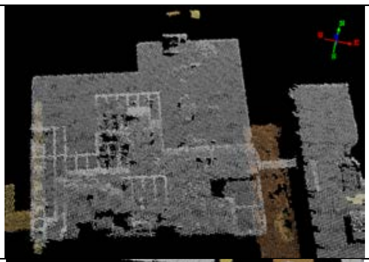
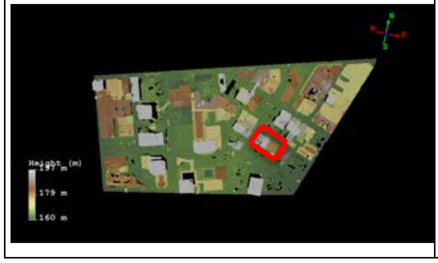
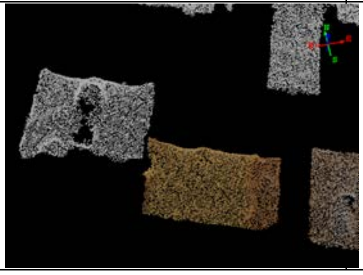
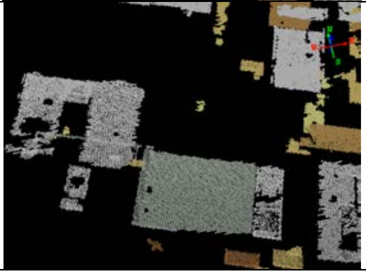
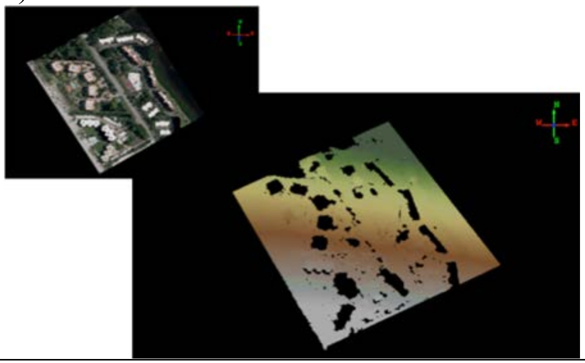
Building Extracted	Image-Derived Point Cloud	lidar Point Cloud
		
		
		

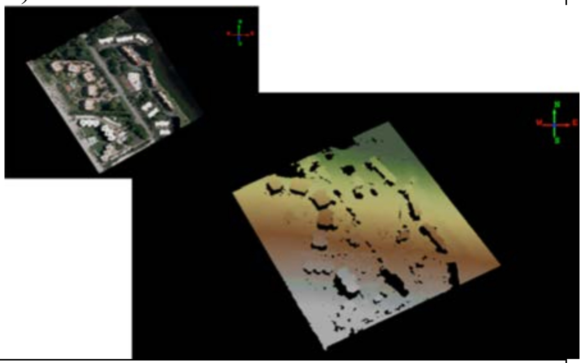
Figure 6: An overview of the three areas in the Rochester dataset used to quantify how many points were retained and the quality of rooftop structures modeled. This shows how the image-derived point clouds are unable to retain the detailed rooftop features that the lidar data generates. While this could be attributed to point density, there is still much detail lost in these point clouds.

While the lidar rooftop data shows significantly more voiding compared to the image-derived point clouds, it clearly retains much more structural information that the image-derived data fails to include. This is most obvious is the second building evaluated where the lidar rooftop has several unique, linear features at different heights. While the image-derived data generally extracts the different heights from the rooftop, all the features included in the lidar data are lost.

Determining quantitative metrics for success with the Florida dataset posed a challenging problem considering there was no co-collected lidar of the same scene. Qualitatively, results are consistent with the Rochester image-derived point cloud since sides of buildings posed the same problem. Overall, large buildings with complex rooftops were successfully isolated for modeling. However some buildings, at least seven in this subset of the scene, were either mistakenly identified as trees or the ground surface and not extracted. (Figure 7) a)



b)



c)

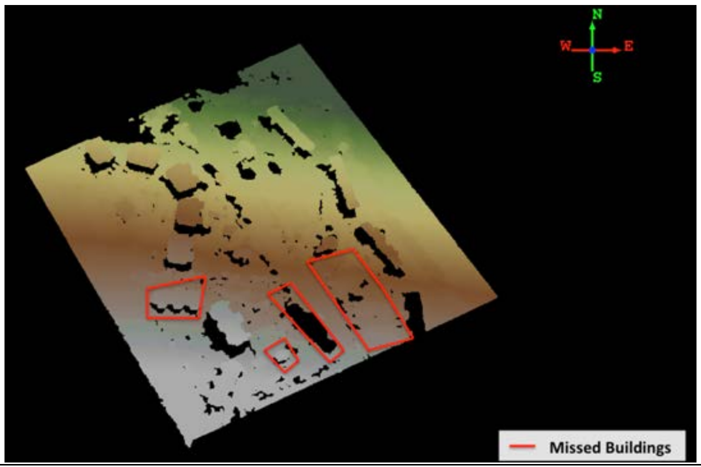


Figure 7: A qualitative comparison on how the Florida image-derived point cloud performed for building extraction. a) shows the terrain file output from the algorithm where voids should indicate all buildings in the scene. Though the terrain file isolates most of the buildings, there are still several that it considers terrain. b) shows the extracted building files displayed over the terrain file. Voids in this dataset show buildings or other features identified as trees. c) This image shows the extracted building file displayed over the terrain file. Polygons highlighted in red are buildings that were missed in extraction mostly due to horizontal normals.

The Florida scene is unique from the Rochester scene since there are many more one-story buildings. There is a correlation between taller buildings and improved building extraction since there are typically fewer consistent clustered points in taller buildings, thus impacting the Euclidean clustering. By removing the horizontal normals in the Florida scene, improved building extraction is expected.

3.3 Future Work

While this algorithm successfully extracted rough building models from 3D point cloud data using airborne imagery, several improvements could be looked at in additional studies. One improvement that could help improve building extraction is using RGB information retained in imagery for modified building segmentation. Furthermore, retaining RGB information improves visual context in the final point cloud product, which is typically desired by customers. An additional study could combine image-based point clouds and lidar point clouds to help provide more information where voids exist, or improve scene point density. Another area that needs to be looked at is retaining detail in roof structure from image-derived point clouds. Lastly, a more sophisticated approach to removing horizontal normals in a point cloud scene could be developed to improve building extraction for dense image-based point clouds.

4 CONCLUSION

Automated extraction of three-dimensional building models can provide analysts with precise geometric models to be used in a variety of applications such as urban planning, disaster assessments, or for visualization. In this study, an algorithm used to extract buildings from lidar data was evaluated using image-derived point clouds. Emphasis was placed on creating a robust point clouds from several optical images in Agisoft Photoscan to be used in analysis. The results from the data initially showed that sides of buildings inhibit successful Euclidean clustering during building classification. By removing horizontal normals in the point cloud, successful building extraction could be performed in image-derived point clouds using the proposed algorithm. Additionally, it can be concluded that while the image-derived data was able to extract most buildings, unique rooftop details were lost. This approach, however, is sufficient if approximate building models are needed in an urban environment.

5 REFERENCES

1. Sun, S. *Automatic 3D Building Detection and Modeling from Airborne lidar Point Clouds.* Rochester Institute of Technology, Ph.D. Dissertation. (2013)
2. Nilosek, D., Sun, S., and Salvaggio, C. *Geo-Accurate Model Extraction from Three-Dimensional Image-Derived Point Clouds*. Proc. of SPIE Vol. 8390 (2012)
3. Leberl, F., Irschara, A., Pock, T., Meixner, P., Gruber, M., Scholz, S., and Wiechert, A. *Point Clouds: lidar versus 3D Vision.* Photogrammetric Engineering & Remote Sensing. Vol. 76, No 10. pp. 1223-1134 (2010)
4. Nilosek, D., Walvoord, D., and Salvaggio, C. *Assessing Geoaccuracy of Structure from Motion Point Clouds from Long-Range Image Collections.* SPIE Optical Engineering. Vol 53(11) (2014)
5. Sirmacek, B., Taubenbock, H., Reinartz, P., and Ehlers, M. *Performance Evaluation for 3D City Model Generation of Six Different DSMs from Air- and Spaceborne Sensors.* IEEE Selected Topics in Applied Earth Observations and Remote Sensing. pp. 59- 70. (2012)
6. Verhoeven, G. *Archaeological Three-Dimensional Reconstructions From Aerial Photographs with Photoscan.* Archaeological Prospection. Vol. 18, Issue 1. pp. 67-73 (2011)
7. PhotoScan Agisoft, <http://www.agisoft.ru>
8. Hartley, R., and Zisserman, A. *Multiple View Geometry in Computer Vision.* Cambridge University Press. (2000)