Geo-Accurate Model Extraction from Three-Dimensional Image-Derived Point Clouds

David Nilosek[a], Shaohui Sun[a], and Carl Salvaggio[a]

[a]Rochester Institute of Technology, 1 Lomb Memorial Dr, Rochester, NY, USA

ABSTRACT

A methodology is proposed for automatically extracting primitive models of buildings in a scene from a three-dimensional point cloud derived from multi-view depth extraction techniques. By exploring the information provided by the two-dimensional images and the three-dimensional point cloud and the relationship between the two, automated methods for extraction are presented. Using the inertial measurement unit (IMU) and global positioning system (GPS) data that accompanies the aerial imagery, the geometry is derived in a world-coordinate system so the model can be used with GIS software. This work uses imagery collected by the Rochester Institute of Technology's Digital Imaging and Remote Sensing Laboratory's WASP sensor platform. The data used was collected over downtown Rochester, New York. Multiple target buildings have their primitive three-dimensional model geometry extracted using modern point-cloud processing techniques.

Keywords: building extraction, three-dimensional point clouds, computer vision, geo-accurate, modeling

1. INTRODUCTION

Digital photogrammetry, the process of extracting geometry from multiple digital images, has recently experienced a surge in development with the introduction of many advanced computer vision algorithms to the field. A significant advantage to using modern computer vision techniques in the field of photogrammetry is the possibility of semi or complete automation of structure modeling. A number of computer vision researchers have explored the possibility of using aerial digital imagery to extract building models. Such as matching building edges across images to extract rooftop models.[1] Some researchers have looked at the possibility of using the three-dimensional information from a Light Detection and Ranging (LiDAR) system to find building boundaries and extract models in that fashion.[2] Others have looked at using both aerial imagery alongside LiDAR output to extract refined building models.[3] A study done by Leberl *et al.* (2010) demonstrated that with the current advances in computer vision algorithms and computation technology, the output from such processes can generate three-dimensional structure that is comparable with modern LiDAR systems. There are many advantages to using point clouds derived from digital imagery, the direct relationship between the three-dimensional structure and the imagery allows for easy point cloud-image processing. Leberl points out many of these advantages as well, such as the potential for good error checking and minimization due to the large amount of information redundancy across images.[4]

This paper looks at using a variation of a common computer vision structure from motion (SfM) work flow to extract geometrically accurate three-dimensional point clouds from aerial imagery, then processes the resulting point clouds to produce basic models of man-made structures within the scene. Some of the processing techniques are derived from LiDAR modeling algorithms, however, LiDAR information is not used in the work presented here. The basis for deriving geometrically accurate structure comes from aerial imagery being captured along with pointing information from an IMU/GPS system. While the information provided by the IMU/GPS system is highly accurate, the error within the system can still be reduced,[5] which leads to significantly better output from other computer vision techniques.

Further author information: (Send correspondence to D. Nilosek)
E-mail: drn2369@cis.rit.edu

Algorithms and Technologies for Multispectral, Hyperspectral, and Ultraspectral Imagery XVIII, edited by Sylvia S. Shen, Paul E. Lewis, Proc. of SPIE Vol. 8390, 83900J · © 2012 SPIE
CCC code: 0277-786X/12/$18 · doi: 10.1117/12.919148

2. APPROACH

A modified version of the well-known Bundler software written by Noah Snavley,[6] Geo-Bundler, is used to perform the bundle adjustments. The software was modified to use IMU/GPS information as the initial estimate in the bundle adjustment process, instead of using pose estimation techniques. This allows for the error within the whole system to be reduced in the coordinate system of the IMU/GPS information. The image correspondence information needed for bundle adjustment is generated using a GPU-accelerated version of the scale invariant feature transform (SIFT).[7] The optimized camera information is then passed to another well-known computer vision software packaged, PMVS.[13] This software is used unmodified to produce a dense three-dimensional point cloud in the coordinate system of the optimized camera information, which is geometrically accurate. The resulting point cloud is then processed by extracting man-made structures (e.g. buildings) from the point cloud, and then performing some noise removal processes. Once the point cloud has been refined, a meshing process called dual contouring is applied and a basic model is extracted from the point cloud. This model is also optimized by doing outline refinement and facet normal smoothing. The end-to-end process is laid out in Fig.1.

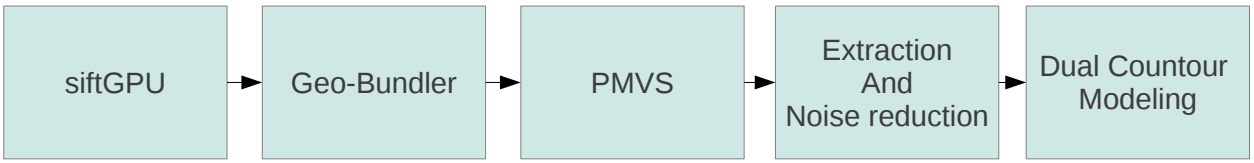


Figure 1. The work flow used in this paper following the order of processing

2.1 Bundle Adjustment with IMU/GPS input

Initial estimates for the bundle adjustment are given with the IMU/GPS information from the imagery, as well as from correspondences that are automatically found between each image. These correspondences are found using the SIFT feature extraction algorithm, one of the more popular feature detector and descriptors used in the computer vision community. SIFT will produce thousands of matches between each image, which will be refined using a random sample consensus (RANSAC) based approach to remove poor matches.[8] These points are used as the tie-points in the bundle adjustment process.

The work flow implemented in this research uses a GPU-accelerated version of SIFT based on the siftGPU libraries.[7] Given the large nature of aerial imagery, it was found that using the GPU gave significant speed improvements in the SIFT feature extraction and matching. Using a NVIDIA Tesla C1060 graphics card, approximately ten thousand SIFT features are detected in a 4000x2672 image in 0.5 seconds. Each set of features are also matched on the GPU, one image-to-image match takes approximately 0.17 seconds.

Each matched feature between each image is triangulated using a linear triangulation approach, based on the direct linear transform, that forms the triangulation problem as a linear systems optimization problem.[9] This requires solving an equation of the form A$\mathbf{X}$=0, Equation 1 shows the A matrix.

$$A = \begin{bmatrix} xp^{3T} - p^{1T} \\ yp^{3T} - p^{2T} \\ x'p'^{3T} - p'^{1T} \\ y'p'^{3T} - p'^{2T} \end{bmatrix} \tag{1}$$

where p and p' are the camera matrices derived from IMU/GPS information, and *x,y,x',y'* are the image point correspondences. Here, n in the form p^{nT} represents the nth row of that matrix. The three-dimensional point is triangulated by solving this system using a least squares approach, this is done using singular value decomposition (SVD). The solution is found as the vector corresponding to the smallest singular value of the A matrix.[9]

Bundle adjustment is the process of refining the initial estimates of a multi-view reconstruction system by minimization of the reprojection error of the three-dimensional structure. The initial estimates used for this system are the camera parameters, the image correspondences, and the three-dimensional point estimates for each correspondence. Traditionally, bundle adjustment was solved as a linear least square problem, however, as

the number of correspondences grows, the computational requirements grow as well. Researchers have looked at using other more efficient optimization algorithms for solving this problem. The method of optimization used here is the Levenberg-Marquart optimization algorithm.[10] Very efficient software exists for solving the bundle adjustment problem using the Levenberg-Marquart algorithm which is used in Bundler.[11]

By altering the Bundler software to take IMU/GPS input and using a GPU implementation of SIFT, the previously described steps necessary for bundle adjustment in a geo-accurate system are performed. An issue arises within the implementation of Bundler concerning the bundle adjustment. Bundler takes a unique approach to bundle adjustment, by iteratively adding cameras to the system then performing a bundle adjustment.[12] This will cause the whole system to drift, which is acceptable for a relative coordinate system, but not here. To counter this effect, large weights are placed on the GPS location points for the camera centers in the bundle adjustment, which prevents significant drifting within the system. The optimized camera matrices are then passed to the next process which will extract a dense geo-accurate three-dimensional point cloud.

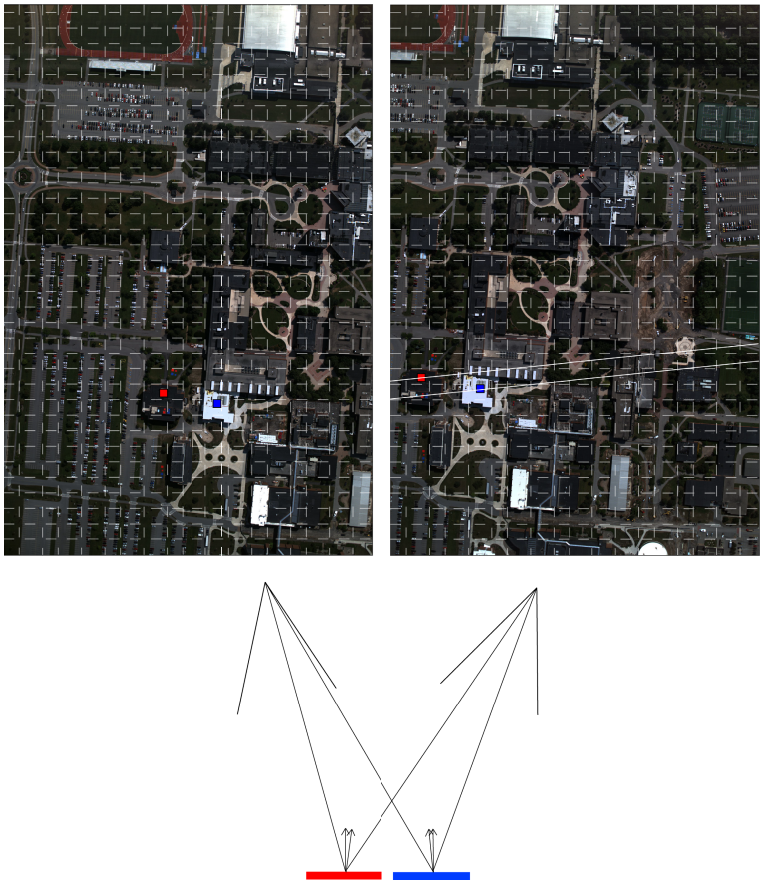


Figure 2. This shows a visualization of a patch being match between two gridded images using the epipolar line. Once the patches are matched the normals are calculated with respect to each camera and kept if matched within a tolerance[13]

2.2 Dense Point Cloud Reconstruction

The dense point reconstruction algorithm used in this work is patch-based multi-view stereo (PMVS). This algorithm uses properties of epipolar geometry to generate a dense set of correspondences across two images. Estimation of the epipolar relationship (fundamental matrix) comes from the input of the geo-accurate cameras provided from the previous steps. Two camera projection matrices are used to derive the fundamental matrix between those two cameras.[9] PMVS works by breaking each image up into small patches and doing a Harris feature detection to find points for matching. Here, when a feature is detected in a patch, that patch is used

for matching. Each patch is matched along it's corresponding epipolar line to search for a match. PMVS uses color and spatial consistency as it's initial matching criterion. Once a potential match has been found, a three-dimensional reconstruction of the patch is done, this results in a plane in which the normal to that plane can be calculated using the three-dimensional position of the plane and the viewpoint of the camera. The normal is reconstructed with the viewpoints used in the reconstruction, and checked for consistency. Fig.2 shows this process. Patches that produce normals that are non-consistent are removed[13]

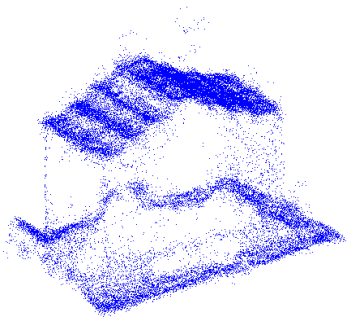
After initial matching, an expansion process is done to generate a dense correspondence. This is done by matching non-matched patches that are adjacent to matched patches and searching across images in the corresponding matched areas, essentially reducing the search area for each patch to a very small region around the seed patch. Each match is triangulated using the optimized IMU/GPS-based camera parameters to generate a dense geo-accurate point cloud.[13]

2.3 Model Extraction

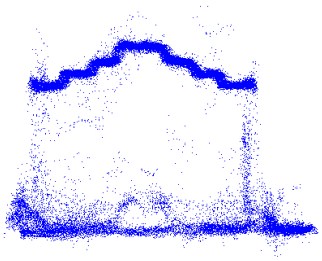
In this section, the image derived point cloud from the previous steps is utilized in further processing for the purpose of watertight building modeling.

2.3.1 Statistical Noise Reduction

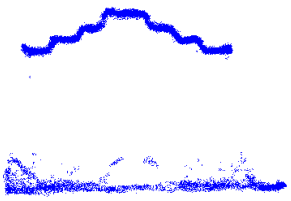
By selecting a large number of aerial images taken from multiple view angles with sufficient overlap, a dense point cloud can be generated. The work flow up to this point takes advantage of several optimization steps, which means error propagation is inevitable. Fig.3(a) and 3(b) show that a number of outliers exist in the point cloud data from an individual building. The output from the following modeling process can be corrupted even by a small number of outliers. Assuming the distances between each point and its neighbors follow a Gaussian distribution with a mean value and a standard deviation, a statistical outlier removal filter from the Point Cloud Library (PCL)[14] is adopted as the method of noise reduction. The noise removing result is shown in Fig.3(c).



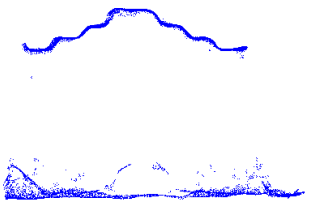
(a) Original points from a building with step-rooftop



(b) Profile view of the original points



(c) Profile view after statistical noise removal



(d) Profile view after surface smoothing

Figure 3. Example of necessary point cloud processing before mesh modeling

2.3.2 Smoothing

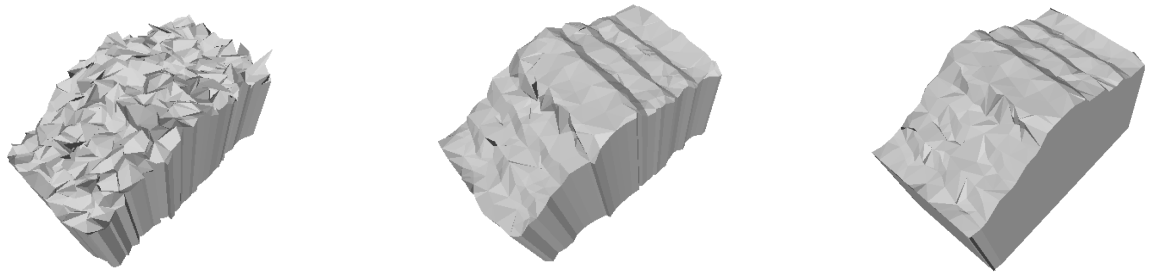
Image derived point clouds from methods like SfM have a lot of irregularities compared to the actual geometry. Some of these can be successfully removed by noise reduction, but others are very difficult to remove by applying just the noise filter. Ideally, the rooftop of a building should be represented by points which are planar in relation. In reality, the rooftop points are always a point cluster around where the roof is supposed to be. Direct surface reconstruction from these points will produce an inaccurate roof (eg. Fig.4(a)). Without having knowledge of the ground truth, the solution here is to utilize an implementation of the moving least squares (MLS) algorithm in PCL[14] which attempts to recreate the missing parts of the true surface by higher-order polynomial interpolations between the surrounding data points. Surface normal smoothing is achieved during this process (see Fig.3(d)).

2.3.3 Dual Contouring

Once the improved representation of the original point cloud is obtained, producing meshed models is desired in this research. Dual contouring is a method used to generate implicit surfaces from a point set. Compared to a similar technique called marching cubes one vital advantage of dual contouring is that it is able to reproduce sharp features, like the intersections of two adjacent surfaces. A robust 2.5D dual contouring technique[15] is used to create building models from the improved point cloud. The 2.5D method is an extended classic dual contouring method. The reason our image derived point cloud can be called 2.5D data is that the aerial imagery used in generating point clouds has very little information on building sides. The expected models consists of complex rooftops which are connected by vertical walls (eg. Fig.4). In order to conduct 2.5D contouring, normal estimation of the points on the rooftop is needed.

2.3.4 Outline Refinement

In most cases, the outlines of building rooftops are supposed to consist of connected straight lines. Meshed models from the last step tend to show wiggling borders. In order to achieve straight lines, the rooftops points are first projected onto a 2D plane. The estimated outline is detected by applying a recursive Douglas-Peucker (DP) polyline simplification,[16] and then points around the outline are moved onto it. The refined model is shown in Fig.4(c) .



(a) Before normal smoothing (b) After normal smoothing (c) After outline refinement

Figure 4. Differences between meshing before and after smoothing and outline refinement

3. DATA

The data used in testing the work flow presented in this work was collected over downtown Rochester, New York. This data was collected with the intention of use with multi-view reconstruction algorithms. Each image was collected with approximately 80 % foward lap and 90% sidelap. Flightlines were flown east-west, north south and in each cardinal direction over the city, providing a very dense collection of imagery over the center of Rochester. Each image is 4000x2672 pixels with a ground sample distance (GSD) of approximately 0.3 meters. Fig.5 shows

the camera centers for each image.

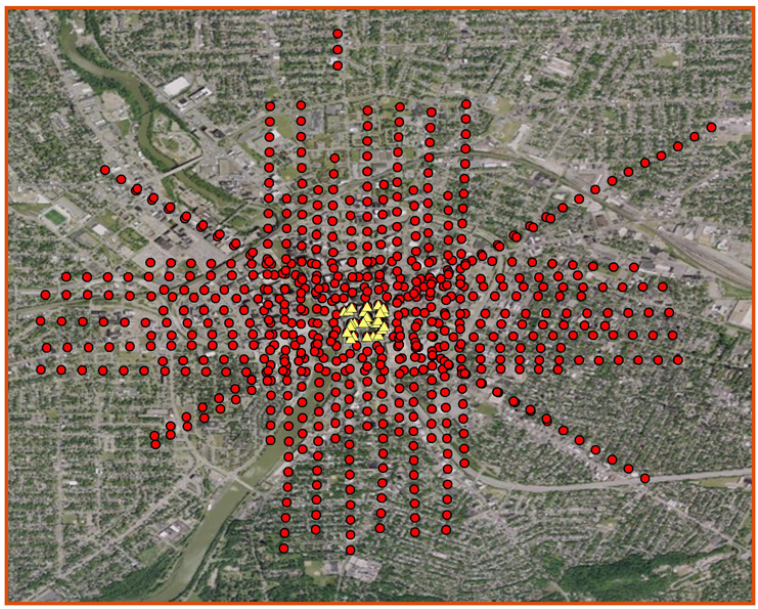


Figure 5. The camera centers for each image capture for an aerial collect over downtown Rochester, NY (approx. 2 sq. km.) This collect was specifically designed to have high overlap for use with three-dimensional reconstruction algorithms

4. RESULTS

A subset of the data presented in the previous section was used for testing with the work flow presented in this research. The center region of the collection was used here as it is the most dense region of the city. Fig.6 shows the dense geo-accurate point cloud generated from this process.

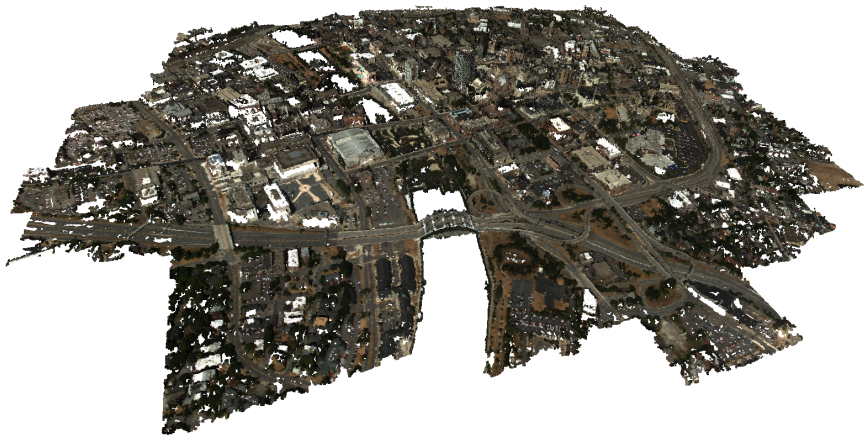


Figure 6. The geo-accurate dense point-cloud reconstruction of downtown Rochester NY using the densest collection area.

A manual verification of geo-accuracy was done by using point-to-point distance measurements compared to known physical measurements. It was found that the accuracy of the measurements was approximately 0.3

meters, which is very close to the GSD of the sensor used to collect the imagery. Four buildings were manually segmented from the dense point cloud for model extraction processing, Fig.7 shows the results of this processing. Due to the noisy nature of image-derived point clouds, even after noise reduction and smoothing, the image-derived models are still partially inaccurate, caused by the variability introduced by the remaining noise. The model is, however, a good representation of the real-world architecture as seen by the input imagery.

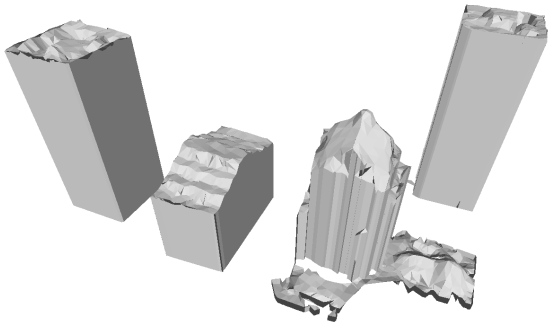


Figure 7. Four major buildings in downtown Rochester, NY modeled from the geo-accurate dense point-cloud

For further verification and visualization, the four building models derived from the dense point-cloud were textured with imagery and imported into Google Earth. Fig.8 shows a screen capture of this visualization. As stated in the previous sections, during the generation of the point cloud, the whole system tends to drift due to the nature of the algorithms being used. This manifests itself in the output dense point clouds as a translation in the whole system. Therefore, to geo-register the geo-accurate point clouds, a reverse translation was applied to the models to put them in a coordinate system that will work with GIS software such as Google Earth.

Figure 8. A screen capture of a textured and geo-referenced visualization of the geo-accurate models

5. CONCLUSIONS

The work shown here introduces a work flow that uses modified computer vision algorithms to produce a dense geo-accurate three-dimensional point cloud that, when segmented, can be used to produce representations of structures within a scene. This work flow is specifically designed to be used with aerial imagery flown with an accurate IMU/GPS system. By using a GPU-accelerated version of SIFT to generate point correspondences, a geo-accurate bundle adjustment can be run using a modified version of Bundler, which produces optimized camera projection matrices. This data can then be used with other computer vision algorithms to produce a dense point cloud in the geo-accurate three-dimensional coordinate system of the cameras, which in this work was PMVS. The output of PMVS yields a dense point cloud which can be meshed using the dual contour algorithm with some noise reduction post-possessing. The overall result is a basic representation of an individual target building.

Automated synthetic scene reconstruction from imagery has many applications in the remote sensing field, most of which fall under the category of rapid and user-friendly photogrammetric measurements. Given the geo-accuracy of this data it may be possible to merge this information along with GIS or other spectral data collected from the same scene. This would allow for further refinement of the extracted model, as well as possible attribution of model facets. This would increase the number of modeling applications which could use this type of data. Inclusion of oblique viewing data would also allow for the extraction and visualization of obstructed features (i.e the sides) of reconstructed structures.

The source code as well as the full image dataset used for the work presented in this paper can be found at http://dirs.cis.rit.edu/3d-workflow/ . This work flow has been compiled and tested under Ubuntu 11.04 and Gentoo 3.2.1 . A description of how to run the scripted work flow along with example data and expected output is also provided

ACKNOWLEDGMENTS

This work was carried out using funding received from the United States Department of Energy under BAA PDP08 Grant Number DE-AR52-07NA28115 and from Esri under Contract Number HM1572-10-C0002. Thanks also goes to Dr. Noah Snavely of the Department of Computer Science at Cornell University in Ithaca, NY, USA for his advisement during modifications made to Bundler.

REFERENCES

[1] C. Baillard, C. Schmid, A. Zisserman, A. Fitzgibbon, *et al.*, "Automatic line matching and 3d reconstruction of buildings from multiple views," in *IAP*, **32**, pp. 3–2W5, 1999.

[2] S. Lach and J. Kerekes, "Multisource data processing for semi-automated radiometrically-correct scene simulation," in *Urban Remote Sensing Joint Event, 2007*, pp. 1–10, IEEE, 2007.

[3] L. Cheng, J. Gong, M. Li, and Y. Liu, "3 d building model reconstruction from multi-view aerial imagery and lidar data," *Photogrammetric Engineering and Remote Sensing* **77**(2), pp. 125–139, 2011.

[4] F. Leberl, A. Irschara, T. Pock, P. Meixner, M. Gruber, S. Scholz, and A. Wiechert, "Point clouds: Lidar versus 3d vision," *Photogrammetric Engineering & Remote Sensing* **76**(10), pp. 1123–1134, 2010.

[5] M. Pollefeys, D. Nistér, J. Frahm, A. Akbarzadeh, P. Mordohai, B. Clipp, C. Engels, D. Gallup, S. Kim, P. Merrell, *et al.*, "Detailed real-time urban 3d reconstruction from video," *International Journal of Computer Vision* **78**(2), pp. 143–167, 2008.

[6] N. Snavely, "Bundler: Structure from motion (SfM) for unordered image collections." http://phototour.cs.washington.edu/bundler/, 2009.

[7] C. Wu, "SiftGPU: A GPU implementation of scale invariant feature transform (SIFT)." http://cs.unc.edu/ ccwu/siftgpu, 2007.

[8] D. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision* **60**(2), pp. 91–110, 2004.

[9] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*, Cambridge Univ Press, 2000.

[10] D. Marquardt, "An algorithm for least-squares estimation of nonlinear parameters," *Journal of the society for Industrial and Applied Mathematics* **11**(2), pp. 431–441, 1963.

[11] M. Lourakis and A. Argyros, "The design and implementation of a generic sparse bundle adjustment software package based on the levenberg-marquardt algorithm." www.ics.forth.gr/lourakis/sba, 2004.

[12] N. Snavely, S. Seitz, and R. Szeliski, "Photo tourism: exploring photo collections in 3d," in *ACM Transactions on Graphics (TOG)*, **25**(3), pp. 835–846, ACM, 2006.

[13] Y. Furukawa and J. Ponce, "Accurate, dense, and robust multiview stereopsis," *IEEE transactions on pattern analysis and machine intelligence* , pp. 1362–1376, 2009.

[14] "http://pointclouds.org/documentation/tutorials."

[15] Q.-Y. Zhou and U. Neumann, "2.5d dual contouring: A robust approach to creating building models from aerial lidar point clouds," in *Computer Vision ECCV 2010*, pp. 115–128, 2010.

[16] D. Douglas and T. Peucker, "Algorithms for the reduction of the number of points required to represent a digitized line or its caricature," *Cartographica: The International Journal for Geographic Information and Geovisualization* **10**(2), pp. 112–122, 1973.