

Article

Automatic Workflow for Roof Extraction and Generation of 3D CityGML Models from Low-Cost UAV Image-Derived Point Clouds

Arnadi Murtiyoso * , Mirza Veriandi, Deni Suwardhi , Budhy Soeksmantono and Agung Budi Harto

Remote Sensing and GIS Group, Bandung Institute of Technology (ITB), Jalan Ganesha No. 10, Bandung 40132, Indonesia; veriandi@students.itb.ac.id (M.V.); deni@gd.itb.ac.id (D.S.); soeksmantono@gd.itb.ac.id (B.S.); agung@gd.itb.ac.id (A.B.H.)

* Correspondence: arnadi_ad@itb.itb.ac.id

Received: 6 November 2020; Accepted: 11 December 2020; Published: 12 December 2020



Abstract: Developments in UAV sensors and platforms in recent decades have stimulated an upsurge in its application for 3D mapping. The relatively low-cost nature of UAVs combined with the use of revolutionary photogrammetric algorithms, such as dense image matching, has made it a strong competitor to aerial lidar mapping. However, in the context of 3D city mapping, further 3D modeling is required to generate 3D city models which is often performed manually using, e.g., photogrammetric stereoplottling. The aim of the paper was to try to implement an algorithmic approach to building point cloud segmentation, from which an automated workflow for the generation of roof planes will also be presented. 3D models of buildings are then created using the roofs' planes as a base, therefore satisfying the requirements for a Level of Detail (LoD) 2 in the CityGML paradigm. Consequently, the paper attempts to create an automated workflow starting from UAV-derived point clouds to LoD 2-compatible 3D model. Results show that the rule-based segmentation approach presented in this paper works well with the additional advantage of instance segmentation and automatic semantic attribute annotation, while the 3D modeling algorithm performs well for low to medium complexity roofs. The proposed workflow can therefore be implemented for simple roofs with a relatively low number of planar surfaces. Furthermore, the automated approach to the 3D modeling process also helps to maintain the geometric requirements of CityGML such as 3D polygon coplanarity vis-à-vis manual stereoplottling.

Keywords: UAV; photogrammetry; point cloud; roof; 3D model; CityGML; low-cost; automation

1. Introduction

Photogrammetry has a long history of applications, beginning with military use in the early 20th century [1]. It has since been used extensively for small to medium-scale mapping during the course of the 20th century due to its natural advantage in covering large areas in a short time period. With the advent of laser scanning technology, photogrammetry experienced a short lull before being brought back to prominence with the development of high density image matching [2]. On the other hand, the increasing trend of Unmanned Aerial Vehicles (UAVs) or drones has also boosted interest in larger-scale aerial photogrammetry, including for mapping purposes [3,4].

The decreasing cost of UAV platforms and camera sensors, coupled with its capability to cover large areas, has arguably created one of the more cost-friendly combinations for mapping. Although recent

developments in miniaturized lidar has seen significant progress [5], as far as the overall costs are concerned, UAV-embarked aerial images remain a strong contender. Naturally the UAV photogrammetry technique also possesses caveats when considering a high-precision mapping, as is traditionally the case with classical photogrammetry. Most of these issues stem from the fact that modern digital photogrammetry benefits very much from developments in the computer vision domain. These issues might include camera calibration [6], bundle adjustment problems [7], up to dense image matching behavior [8]. Nevertheless, it remains an interesting solution to mapping in general. This is even more so when faced with budgetary constraints and demands. In the specific case of Indonesia, a recent major governmental push for a complete map coverage of the country's territories has regenerated interest in low-cost mapping solutions.

One of the principal results of UAV photogrammetry is a point cloud representing the mapped terrain derived from the dense image matching process. This point cloud is in many ways similar to the main product of lidar albeit with notoriously larger error in terms of the altitude due to the inherent low base-to-height ratio in photogrammetry. Photogrammetric point cloud also lacks useful information such as intensity values which lidar provides. Photogrammetric quality would therefore highly depend on the quality of the sensors as well as the metric considerations during data acquisition and processing. Within the context of urban city mapping, a point cloud must be processed further to create either vector maps or 3D representations of different object classes (e.g., buildings, vegetation, roads, etc.) consisting of geometric primitives, preferably with annotated attributes for 3D Geographical Information System (GIS) purposes. This "digitization" process is often performed using manual 3D stereoplottling in classical aerial photogrammetry. In the case of UAV photogrammetry, stereoplottling may be problematic due to the non-metric nature of the sensor.

The aim of the paper was therefore to introduce an automated workflow from UAV photogrammetry point cloud to the generation of 3D models of building roofs, of which a further extension would be its conversion into a CityGML-compatible 3D building model. The developed algorithm aims to generate 3D models complying to the Level of Detail (LoD) 2 as defined by the CityGML standard. The method of UAV photogrammetry is selected for this research in order to provide a low-cost solution to the aforementioned national mapping problem. The research also limits itself to the extraction of roofs and therefore by extension the buildings class, since we argue that in urban settings this class represents the most important aspect. For the purposes of the experiments, a point cloud of the Jatinangor campus of the Bandung Institute Technology was used. The point cloud was acquired using UAV photogrammetry from nadir images with no oblique images involved. A type of rule-based instance segmentation was implemented for the point cloud segmentation phase with the possibility of automatic object attribute annotation.

The paper is divided as follows: the next section will talk about some related work in the subject of aerial point cloud segmentation and the extraction of building roofs as well as its 3D modeling. The third section will thereafter describe the proposed workflow, by integrating rule-based point cloud instance segmentation and roof plane extraction algorithms. The following section will then present the results as well as some discussions, followed by conclusions to the paper.

2. Related Work

Aerial mapping has long been employed for topographic operations covering large areas [9]. Photogrammetry has a long history in this role, starting with classical nadir image stereo observations [1] and augmented by dense image matching [2] and structure-from-motion algorithms [10,11]. The development of laser scanning sensors or more precisely Aerial Laser Scanning (ALS) [12] added another alternative for aerial mapping [13]. In recent times, photogrammetry has been further improved with the (re-)implementation of oblique images [14,15] and most importantly the use of UAVs [16,17].

Indeed, UAVs enabled the democratization of photogrammetry, making it accessible to different types of stakeholders thanks to its relatively low cost when compared to traditional small scale aerial mapping [4]. Another interesting development is the miniaturization of ALS sensors with the objective of placing them on UAV platforms [5].

When compared to ALS, photogrammetry may seem to have a few more disadvantages. This includes a longer workflow involved to process the images compared to ALS' relatively simple workflow, thus introducing more human intervention and consequently more room for error. Nevertheless, proper photogrammetric processing is essential to ensure its geometric qualities [9,17,18]. That being said, photogrammetry, especially when employed with UAVs, remains a very interesting solution due its lower overall cost. Further developments such as improvements in GNSS technology and acquisition methods enable the reduction of field surveys [19,20], which traditionally present the most costly part in a photogrammetric project.

As has been previously mentioned, a recent push towards nation-wide large-scale mapping (up to 1:5000 scale) in Indonesia has sparked interest in cost-feasible solutions [21]. Some of the considered methods include high-resolution satellite imagery [22] and UAVs [23]. The use of UAV is interesting in this regard as it provides a low-cost solution. However as has been previously mentioned, various technical and non-technical issues such as limitations in terms of flight autonomy, payload, and regulations are also important to take into account in any UAV projects. Indeed, an increasing number of countries have enacted UAV-specific regulations which govern amongst others the permitted flying height, specific no-fly zones, or privacy issues [24,25]. In Indonesia, UAV mapping is at the moment of writing regulated by a ministerial decree as no nation-wide law is available yet. In addition to certain limitations on flying height and no-fly zones, pilot certification is required and a special permit by the Ministry of Defense for UAV mapping purposes is required [26]. A thorough project planning must therefore take into account not only geometric but also legal issues [8]. Furthermore, while the government's required outcome remains a 2D product, we argue that in the middle to long term 3D information systems will be needed especially for urban areas in the form of 3D GIS and/or smart city systems [27].

2.1. Aerial Point Cloud Classification

Point cloud segmentation is a logical follow-up process after its generation. This is because raw point clouds represent mainly geometric properties [28], whereas the understanding of any 3D scene requires semantic information [29]. Approaches to segmentation methods can be divided according to the approach taken; a typical classification of segmentation approaches include the distinction between rules-based approaches [30] and machine learning approaches [31,32]. The rule-based approach is also called the heuristic method in [32] or the algorithmic method in [28], and basically involves hard-coded prior knowledge (i.e., "rules") introduced to the algorithm. Machine learning techniques and its subset deep learning on the other hand, involves prior training using previously labeled or annotated datasets to predict the class of each point in the point cloud [33,34]. Deep learning, in particular, has seen much use in performing task automation, including in the field of point cloud segmentation [35,36].

Another way to classify segmentation methods may take inspiration from 2D image segmentation [37], mainly the distinction between semantic, instance, and panoptic segmentations. In semantic segmentation, a class is attributed for each point in the scene. Using the algorithmic approach, the most prominent example of this process is the segmentation of ground point clouds [38]. Region-growing algorithms may also be used to distinguish other classes [39]. Deep learning algorithms have also performed well in this category, as evidenced by several studies described in [36,40]. On the other hand, in instance segmentation, each object is classified as an individual entity. In view of information systems involving particular entities (e.g., a university campus or governmental complexes), this approach can be advantageous. In [41],

an algorithmic approach was presented to the problem of instance segmentation by using pre-existing GIS files, an approach similarly explored in a different form in [42,43]. Panoptic segmentation involves a combination of instance and semantic segmentations in the image processing terminology, which [41] also tried to perform by integrating ground classification.

2.2. 3D Modeling of Buildings

In view of application in 3D city model, a point cloud presents a redundancy of information [44]. Indeed, the simplification of point cloud structure into 3D primitives has been the main aim of many studies focusing in data-driven approaches. In [45], the authors proposed another approach based on 3D mesh, but ultimately still generate geometric primitives. Similarly, in [46] the authors mentioned two approaches based on the data-driven method to be used ad hoc, one using geometric constraints to create the building 3D models and another based on 3D mesh simplification.

Generally the process of 3D modeling of buildings from aerial point clouds involves a prior segmentation of the point cloud into the roof class [47–49]. The segmented roofs were then modeled using different methods such as PCA analysis [48], roof topology [50], label constraints [51], or the more classical Hough Transform and RANSAC [52,53]. Extrusion of the modeled roof into the available DEM should then, in theory, satisfy the requirements of LoD2 in the CityGML paradigm for 3D city modeling.

A similar approach to the one presented in this paper was presented in [54] in which the authors implemented Euclidean distance-based region growing and RANSAC to perform a similar task. Both [54] and [55] also used UAV data as input. Another research used heterogeneous sources, including a combination of terrestrial and aerial data, to create LoD3 compatible models [56]. In our approach, we implement a region growing method based on the value of point normals and Gaussian curvature to perform the segmentation into roof patches.

2.3. CityGML Paradigm

When working with 3D city models, users often face the problem of data heterogeneity resulting from a variety of data source, acquisition time, method, and required levels of detail. CityGML was created to address these problems [57]. CityGML is a standard that was established in 2008 by the Open Geospatial Consortium (OGC). It is one of the most prominent international standard which encompasses the geometric, semantic, and visual aspects of 3D city models [58].

One of the classes defined in CityGML is that of buildings which can be represented in various LoDs. Out of all the levels of detail for buildings defined within CityGML, LoD2 is particularly important. This is mainly because buildings in LoD2 introduce roofs into the model; a crucial requirement for a wide range of applications such as estimating solar potential of rooftops [59].

3. Data and Employed Methods

In this paper, the proposed workflow was tested using a point cloud generated from nadir UAV photogrammetry acquired from the Jatinangor campus of the Bandung Institute of Technology (Indonesia). A fixed-wing UAV was used to acquire a total of 187 nadir images, taken with a 20 mm Sony A6000 camera. With an average flying height of 200 m, this gave an average theoretical Ground Sampling Distance (GSD) of 4 cm. The overlap rate between the images was set at 80% with a sidelap of 60% between flight lines. In total, an area of about 60 hectares was covered by the images. Although the use of oblique images may increase the quality of the result in terms of data completeness [60], practically speaking this would involve more flights and thus increase the necessary resources [8]. In an attempt to formulate a possible rapid and low-cost solution, only nadir images were used for the generation of the point cloud. The resulting point cloud was then subsampled to a regular grid of 10 cm. The 10 cm value was selected to

fulfill the requirements of a 1:1000 scale map [61], even though the national requirement at the moment is set at 1:5000. As a consequence of this sole nadiral acquisition, building façades were mostly missing from the point cloud. This actually helps with the roof segmentation process as the remaining “wall” points may be considered as noise.

A general illustration of the workflow can be seen in Figure 1. The segmentation of the building roofs from the raw point cloud utilizes the M_HERACLES toolbox (link available in the supplementary materials section) with a workflow adapted from [41]. M_HERACLES uses 2D GIS shapefile geometries as prior knowledge in guiding the point cloud segmentation, while providing the possibility to annotate the segmented clusters directly with the GIS attributes. In this regard, the segmentation phase presents a type of instance segmentation, in which individual clusters of point clouds were acquired at the end of the process. Section 4 will discuss the results of the segmentation process and a quick comparison against a similar process performed using a commercial software.

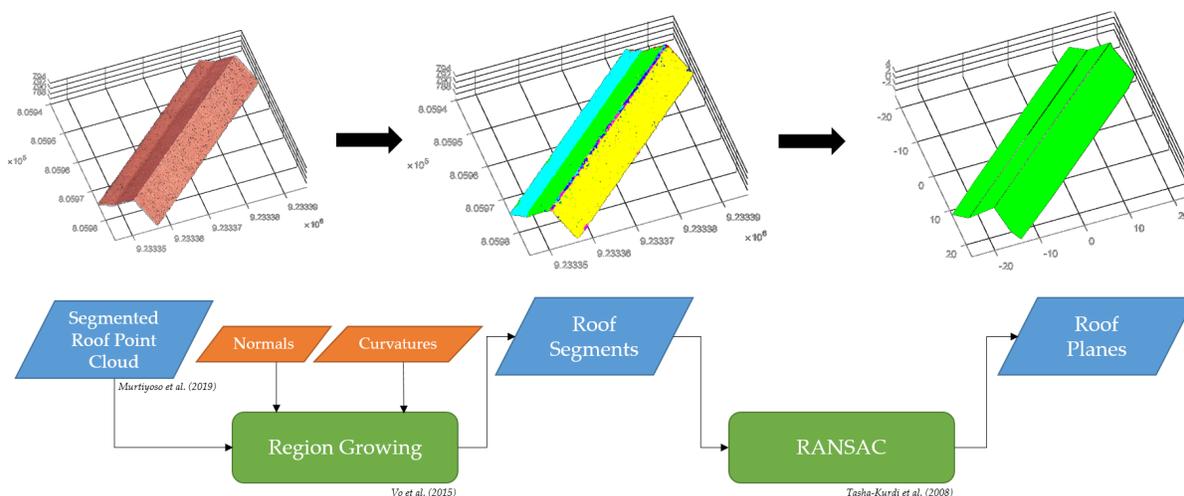


Figure 1. General illustration of the developed approach to extract roof 3D primitives from the input Unmanned Aerial Vehicle (UAV) images-derived point cloud.

The segmented roof point clouds were thereafter processed individually to extract 3D polygons representing the different roof surfaces. A region growing approach inspired by [39] was implemented using point cloud normal and curvature values as the main constraint. Several parameters for this phase needed to be adjusted, of which the main defining variable is the threshold value for the difference of normal angles between any neighboring points. This value was obtained empirically and was set at one degree. This process enabled a subdivision of the roof point cloud into planar segments (cf. Figure 1). Furthermore, vertical elements such as the point cloud of walls were excluded from the result. For each roof segment, a RANSAC-based algorithm then fits a planar surface and extracts the surface equation [53]. The tolerance for the RANSAC plane-fitting was set depending on the resolution of the input point cloud. In this case, a value of 20 cm was chosen (two times the point cloud spacing).

In view of the final objective of conversion to CityGML-compatible formats, two main geometric constraints were considered in the modeling process:

1. Coplanarity of points belonging to the same segment: in order to enforce this coplanarity, all points belonging to the same segment were projected to the computed surface. Furthermore, during the computation of the 3D polygon border and junction points, the RANSAC-derived surface equations play an important role. For example, junction points between two segments were computed along the 3D line resulting from the intersection of the two neighboring segment surfaces. In the case of

three or more neighboring segments, a simple least-squares solution was performed to obtain the junction point.

2. Absence of gaps between junction points (“snapping”): a specific function was developed to ensure as best as possible that no spatial gaps exist between the junction points of neighboring segments. Algorithmically, in order to enforce this constraint, the average coordinates of junction points located near to each other were averaged by taking into account the global/merged segments. At this point, the segment’s vertices were presented in a triangular mesh form (Figure 2). Here again, the coplanarity constraint was redeployed in order to keep the points coplanar with regards to each respective segment surface.

Several other supporting functions were developed to refine the resulting roof segment 3D polygon. This includes amongst others a mesh simplification function to reduce the number of vertices belonging to the same parallel lines in any given 3D surface and a function to convert the original triangular mesh into 3D multipatch polygon as required by the CityGML format. Most of the functions used in the described workflow with the notable exception of the conversion to CityGML were integrated into the updated version of M_HERACLES.

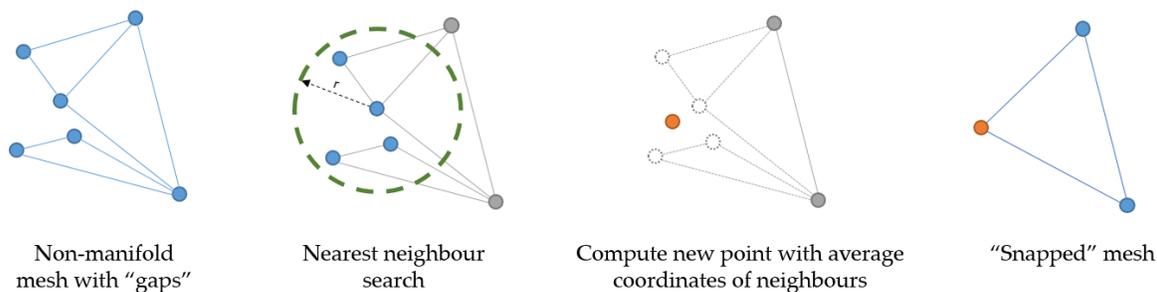


Figure 2. General illustration of the developed approach to extract roof 3D primitives from the input UAV images-derived point cloud.

4. Results and Discussions

The result section will be divided into three sections covering the building/roof segmentation, the generation of 3D roof models, and the conversion of said 3D models into CityGML format. Several metrics will be used to assess the obtained results. For the segmentation part, a measure of precision, recall, and F1 score will be used to determine the quality of the segmentation process. These scores are computed as follows:

$$\%P = \frac{TruePositive}{TruePositive + FalsePositive} \quad (1)$$

$$\%R = \frac{TruePositive}{TruePositive + FalseNegative} \quad (2)$$

$$\%F1 = 2 * \frac{\%P * \%R}{\%P + \%R} \quad (3)$$

where $\%P$ represents the precision, $\%R$ the recall, and $\%F1$ the harmonized scores used in the assessment. For the generated 3D roof models, assessment was two-fold: first a geometric analysis by computing the Euclidean distance between each 3D surface and the input point cloud was performed. This was done using the mesh-to-point function available in the open source software CloudCompare (<https://www.danielgm.net/cc/>, accessed 6 October 2020), whereas a Gaussian normal curve was fitted

into the resulting histogram of deviations in order to compute the average error and standard deviation values. A second assessment involves the verification of each surface vertices' coplanarity as required by the 3D polygonal/multipatch surface in CityGML models. The check was done by a simple operation involving singular value decomposition.

4.1. Building and Roof Point Cloud Segmentation

In order to perform the segmentation using M_HERACLES, two inputs were necessary. The point cloud generated by UAV photogrammetry was subsampled into a regular spatial grid of 10 cm. Aside from the input point cloud, a GIS shapefile of the test site was acquired. This shapefile was digitized from photogrammetric orthophotos for a separate GIS project of the university campus; we benefit therefore from this previously available data to help the segmentation process. Within the shapefile, several semantic attributes were also present. For test purposes, only a total of twenty buildings representing all the architectural types in the campus were used with similar buildings represented by one or several objects. Problematic cases such as the existence of heavy vegetation above the building or complex roof architectures were also included within this 20-object dataset.

A sample of six building point clouds segmented using the M_HERACLES toolbox is presented in Figure 3. A global view of the result is also shown in Figure 4. It is worth noting that as can be seen in Figure 3, the resulting clusters of segmented point clouds were automatically annotated with the corresponding semantic attributes from the 2D shapefile.

Following a precedent from [28], a comparison was performed between the results of M_HERACLES and a commercial software capable of performing point cloud classification. Agisoft Metashape version 1.6.5 (build 11249) was chosen to perform this task. However, it should be noted that while this comparison is useful in showcasing the advantages and disadvantages of M_HERACLES, the two solutions offer a different approach to the problem of segmentation.

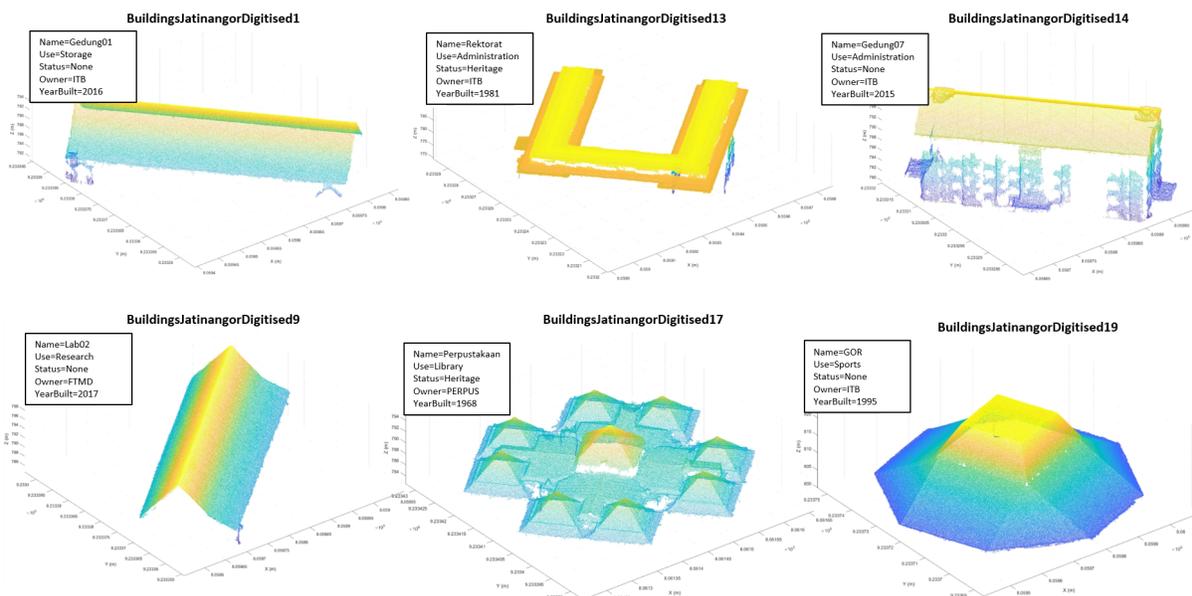


Figure 3. Result of the M_HERACLES instance segmentation, showing six of the 20 processed objects. Note the information box to the upper left of each individual figure, showing semantic attributes derived from the input 2D GIS shapefile.

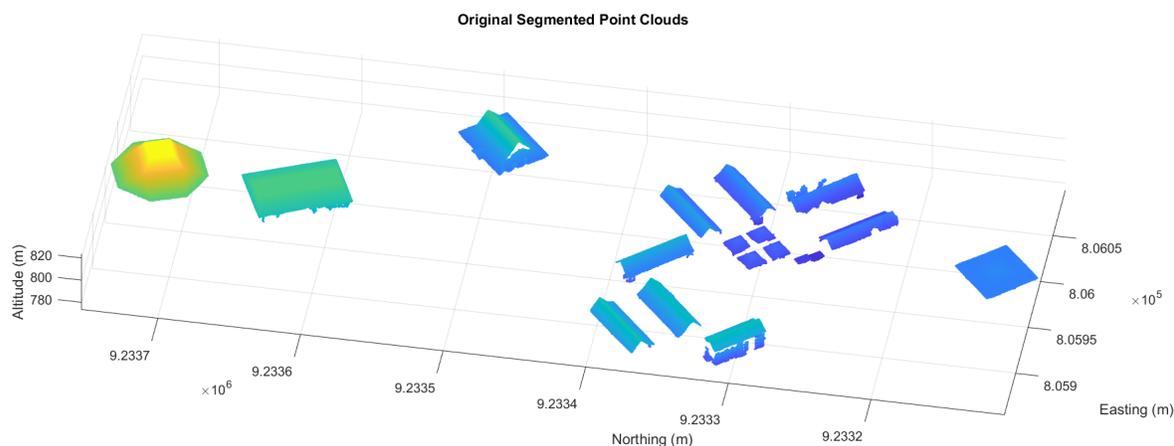


Figure 4. A global view of the segmented building roofs, before surface extraction.

Indeed, while M_HERACLES performs instance segmentation as has been previously established, Metashape provides instead a type of semantic segmentation. In these results, no individual object cluster were detected; rather the classification rests more general in nature. In this regard, M_HERACLES provides an advantage as it not only segments individual object clusters, but it also annotates the GIS-derived semantic information automatically to each instance. This is interesting in light of using the results as part of a 3D GIS. This of course comes with the caveat that a GIS shapefile is required beforehand, although the various filtering features of M_HERACLES reduces the necessity for precisely digitized shapefile. This opens therefore the possibility to quickly create the necessary GIS shapefiles from any source, e.g., available satellite or aerial orthophotos for the purposes of the instance segmentation.

Overall, M_HERACLES managed to produce better result than Metashape, in which the algorithm used during the classification is not divulged but is most probably based on a machine learning method. Figure 5 presents three samples of the segmented building from the Jatinangor dataset. As can be seen from this figure, Metashape encountered problems of mis-classification: several roof objects were labeled as the ground or vegetation classes. This result would evidently influence further results down the processing pipeline; M_HERACLES therefore gives another advantage in this regard. Interestingly enough, M_HERACLES managed to avoid the problem of mis-classification despite in some cases encountering heavy vegetation. In tackling this particular problem, M_HERACLES relies on a simple majority-takes-all classification approach as described in detail in [41].

Using the aforementioned statistical parameters, a numerical comparison was performed between the results of M_HERACLES and Metashape as can be seen in Figure 6. Concurrent with results from [28], Metashape presented a very high precision score (99.91%) while scoring lower in terms of recall (89.03%). M_HERACLES performed well in both scores (99.51% precision and 99.62% recall), although this is somewhat biased by the fact that it used a winner-takes-all approach in segmenting the point cloud. This means that the quality of M_HERACLES depends on the digitization quality of the input shapefile, even though measures were taken to minimize the effect of digitization error. This includes, among others, a buffer zone around the shapefile vector and the use of region growing algorithm to filter the result from noises [41]. That being said, the results of Metashape improved quite significantly when compared to previous experiments reported in [28]. Indeed, the machine learning approach to classification has the potential of becoming better as more training data are available.

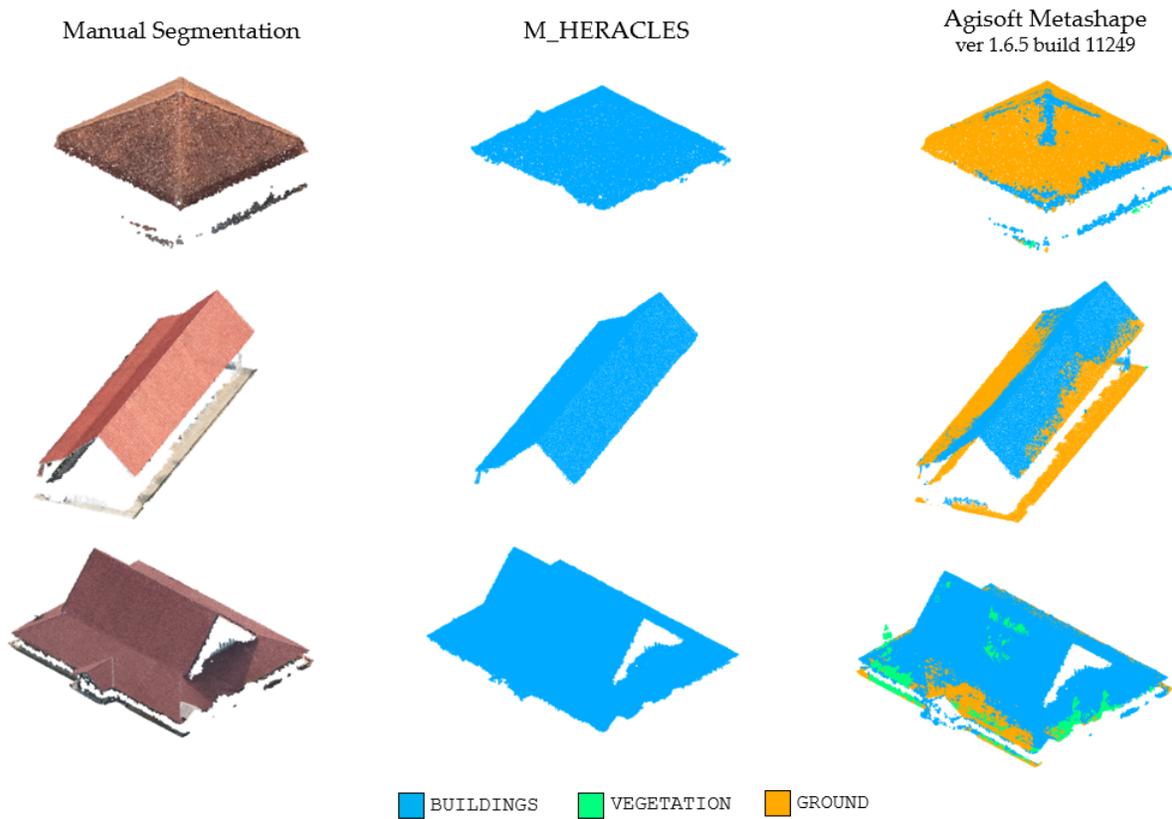


Figure 5. Three examples of the segmented building/roof clusters in ascending level of roof complexity. Note that M_HERACLES employs a winner-takes-all approach, hence the homogeneous semantic annotation. On the contrary, Metashape displays a case of mis-classification mostly between the “ground” and “building” classes.

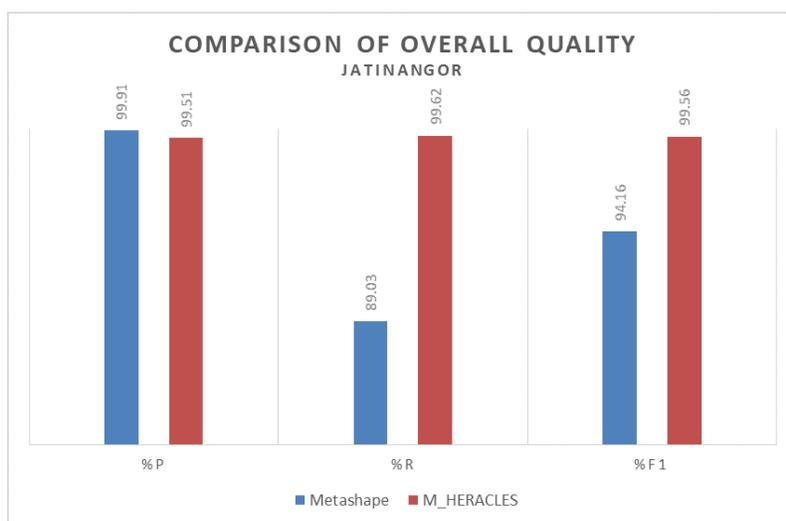


Figure 6. Global comparison of the precision, recall, and F1 scores of the classification results from M_HERACLES and Metashape.

4.2. 3D Building Model Generation

Out of the twenty buildings segmented in the previous section, seventeen were successfully used in the 3D roof generation. The remaining three presented more complex roof types and automatic 3D roof reconstruction was not possible without manual intervention. The seventeen building point clouds were processed using the proposed method, from which roof surfaces were extracted. This section will describe some assessments as regards to the quality of the roof segments, and not the building 3D models as a whole.

The seventeen roof models were able to be extruded to the linked Digital Elevation Model (DEM) to create a 3D mesh of the building (cf. Figure 7). At this point, the 3D mesh of the building walls serves a limited purpose since they are not yet compatible to the CityGML definition. However, an interesting output of the workflow proposed in this paper is the possibility to automatically annotate the resulting roof (and by extension the buildings also) 3D models with semantic information extracted from the input GIS shapefile (Figure 8).

In order to assess the quality of the generated roof models, two types of analysis were performed. Note here that the assessment only applies to the roof models and not the buildings, since M_HERACLES uses a data-driven method to reconstruct the roofs while the generation of the wall surfaces follows a more model-driven or parametric approach. Since the original point cloud also lacks building façades, the analysis was focused on the roofs instead.

The first analysis compares the generated roof models against the original point cloud. For this assessment, the perpendicular Euclidean distance between the point cloud and the generated roof surfaces were computed for each detected roof segment. This was done using the mesh-to-cloud functionality in CloudCompare, yielding results as shown in Figure 9. As can be seen from said figure, the overall average error amounts to 2.5 cm with a standard deviation of 6.0 cm. The error is defined as the deviation between the points in the input point cloud and their respective roof surface primitives. This result is quite satisfactory considering that the original point cloud's resolution is of the order of 10 cm. However, a closer look at the individual buildings' statistics (as shown in the left-hand side of Figure 9) demonstrates a more heterogeneous distribution of error.

One remark is that the overall error as well as the standard deviation tend to be better for buildings having less joints, i.e., intersection between the different surfaces constituting the building roof. This is shown by B1, B2, B7, B8, and B9 in Figure 9, which each consists of three roof segments, as opposed to, e.g., B3, B4, B5, and B6 with four roof segments each. This, however, may be explained by the fact that the M_HERACLES function contains features which computes adjusted coordinates of joint points in order to satisfy the "snapping" requirement. As may already be inferred logically, more noisy input data, e.g., B15 generated worse standard deviation values, although the mean deviation remains low.

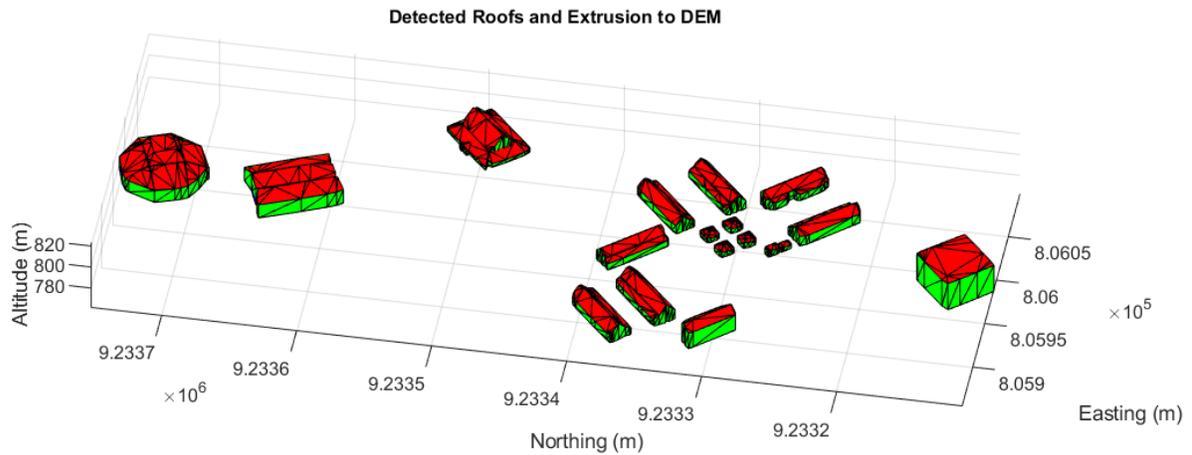


Figure 7. The detected roof 3D polygons with extrusion towards the Digital Elevation Model (DEM).

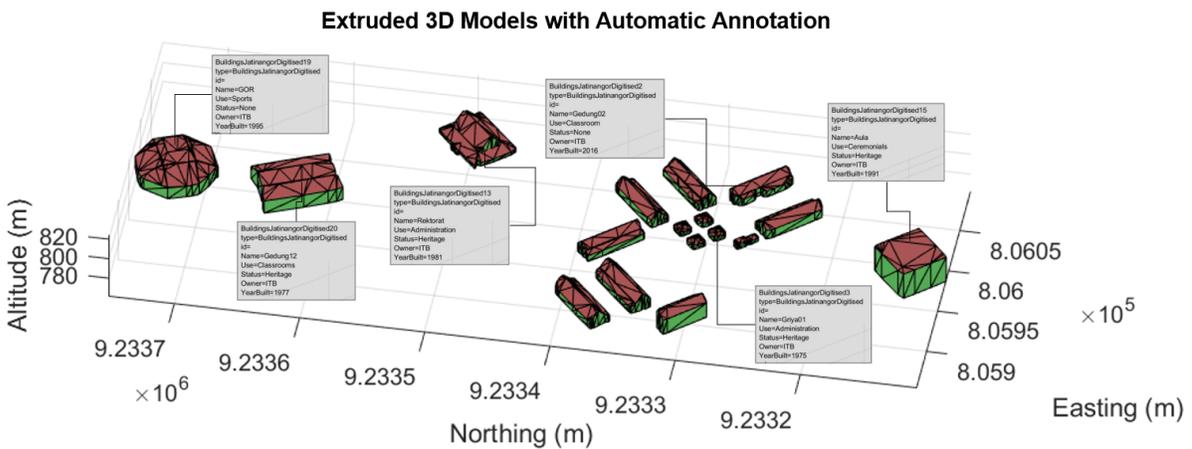


Figure 8. Illustration of the automatically annotated GIS attributes on the resulting building 3D models.

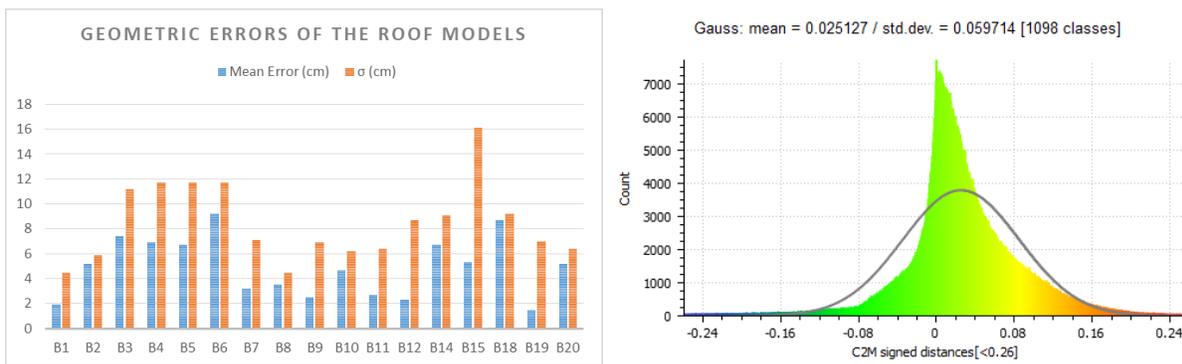


Figure 9. Geometric mean deviations between each generated roof 3D model and the original point cloud and their respective standard deviation values (left) and a histogram fitted to a Gauss curve representing the global values of the same error (right).

A second analysis was conducted by comparing the results of M_HERACLES to manually digitized roof surfaces from a stereoplotting operation. The main problem with manual stereoplotting measurements

is the necessity to force coplanarity for each roof segment. While this is possible to perform (cf. Figure 10), manual intervention is also prone to human error. M_HERACLES on the other hand, automatically respects this restraint due to its integrated coplanarity feature. Figure 10 shows that up to 92.54% of the generated roof surfaces respect the coplanarity constraint with a tolerance of 1 cm, slightly better than the results of manual stereoplotting (89.41%).

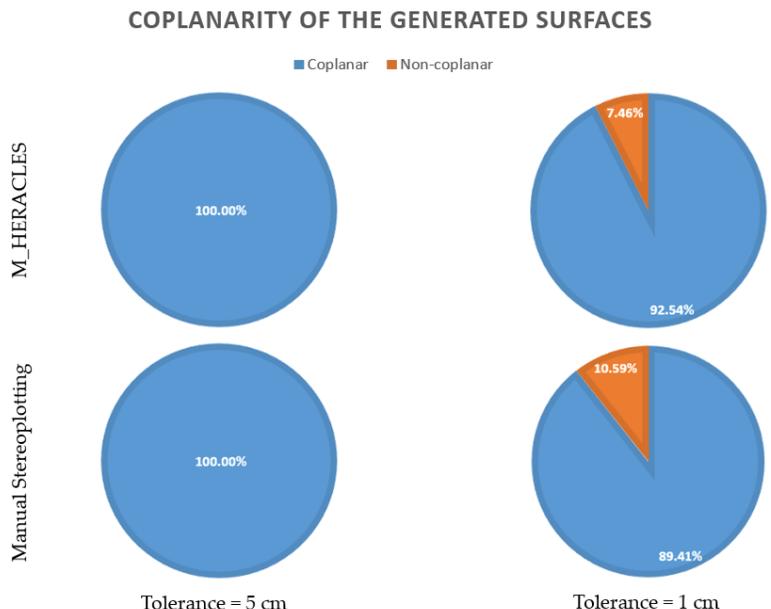


Figure 10. Coplanarity check for each set of vertices constituting the roof surfaces, with a tolerance of 5 cm (left) and 1 cm (right). The same parameters for roof surfaces generated by manual 3D stereoplotting are also given on the lower row as a comparison. Note that the original point cloud’s resolution is 10 cm.

Another issue with the automatic M_HERACLES involves the number of the generated vertices. Using stereoplotting, human intervention ensures an additional layer of qualitative interpretation to the 3D data. On the contrary, the proposed method is a purely data-driven one and the result depends strongly from the input data. On average, M_HERACLES generated 1.7 times more vertices than purely manual plotting.

4.3. CityGML Conversion

A separate conversion program was written in Python, enabling an automatic passage between M_HERACLES and CityGML. The program took roof surfaces created by M_HERACLES as input and generated wall surfaces using a downward extruding process and automatically writes the result in a CityGML file. Moreover, the building base was also created by merging roof surfaces into a roof base surface and then changing its height consistent to the building ground elevation extracted automatically from the DEM. In this regard, the CityGML representation falls into the pre-determined LoD2. Contrary to the mesh representation in Figure 8, both the roof segments and the derived wall surfaces were converted into a multipatch or 3D polygon in order to conform to CityGML requirements.

The conversion essentially involves a simple rearrangement of vertices to follow a clockwise rotation from the barycenter’s perspective. The sorting function first projects the vertices into a 2D plane determined by a Principal Component Analysis (PCA) computation. The barycenter of the vertices was then computed, from which bearing angles were calculated for each vertex with regards to the barycenter point and

arbitrary vector. The vertices were thereafter sorted according to these bearing angles to generate the 3D multipatch in the XML format. The resulting CityGML models are shown in Figure 11.

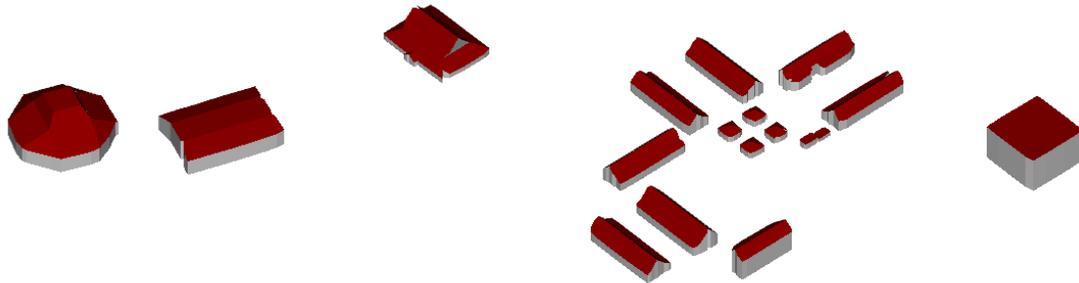


Figure 11. Resulting 3D CityGML models of the buildings generated by M_HERACLES and automatically converted using the Python function.

The main benefit of the M_HERACLES-based procedure is the reduction of time and resources required to produce the building model when compared to manually digitized roof models. Indeed, using manual stereoplottling techniques, not only does the process take longer, but certain skills are also required of the operator. Furthermore, the conversion program also creates perfectly vertical walls in every side of the building. The proposed program also enforces coplanarity of each surface, reducing the requirement for manual verification. This is crucial in order to create CityGML-compatible 3D models. As regards to the overarching objective of developing a low-cost solution while maintaining geometric quality, the use of nadiral UAV photogrammetry has been shown to be adequate at least for an overall point cloud resolution of 10 cm.

Other than these stated advantages, the program also has several shortcomings. Although M_HERACLES provided a feature to try as best as possible to prevent gaps between roof surfaces, the result is still limited mostly by the quality of the input data and the region growing process. Indeed, manual verification and in some instances several manual clean-up of the resulting models is still required even though the amount of manual intervention is greatly reduced. Another limitation of the conversion program as far as the CityGML models are concerned is that it does not at the moment accommodate the creation of surfaces in holes located in between roof surfaces. These holes represent vertical roof façades or voids caused by overhanging roofs (Figure 12). The program only creates wall surfaces that starts from the roof base to the ground, so if there is holes in the input data those holes will also be present in the output building model.

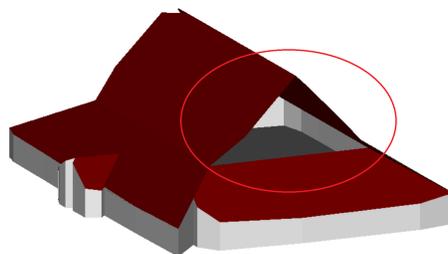


Figure 12. Example of a medium-complexity CityGML building model generated from M_HERACLES roof models. Notice the hole on the model as pointed by the red circle.

5. Conclusions

In this paper, an algorithmic and data-driven approach was proposed to address the 3D reconstruction problem of buildings in an aerial UAV-derived point cloud setting. The proposed method attempts to present an automated workflow from point cloud up to 3D building models which are compatible to the LoD2 definition of CityGML. The M_HERACLES toolbox was used for the segmentation process and up to the automatic creation of roof segments. Another independent function was then employed to convert these 3D roof segments into 3D building models by virtue of extrusion towards a pre-existing DEM.

Results show that the approach taken in this research managed to perform well for building segmentation, with an overall F1 score of 99.56% for the tested dataset. The generated roof segment geometric primitives in the form of 3D surface polygons provided satisfactory results for low to medium levels of roof complexity. However, this particular part of the algorithm was still unable to generate good results for complex roofs; in many instances of complicated forms, fine-tuning is still required to obtain acceptable results. However, of the low to medium complexity roofs that M_HERACLES managed to process correctly, a direct conversion to CityGML format was possible.

Three main advantages of the proposed approach can be retained, albeit not without several trade-offs. First, the use of M_HERACLES' approach to aerial point cloud segmentation permits an instance segmentation to be performed on individual building entities, as well as automatic attribute annotation from the input GIS shapefile.

A second advantage to the proposed method involves the use of the algorithmic approach in creating the 3D surfaces of the roof segments. A few constraints have been added with the specific objective of rendering the result as compatible as possible to the CityGML paradigm. Notably, the algorithm guarantees coplanarity between vertice points of every surface up to an acceptable tolerance. The snapping constraint also greatly reduces the required manual clean-up of 3D roof models; a crucial step that must also be performed when using manually stereoplotted 3D roof models.

The last advantage is straightforward and rings true for most automation attempts: the proposed method reduces processing time as well as resources. Traditionally, 3D city models are created from roof models digitized manually using the stereoplotting method; this means that a skilled operator is required to perform the task. The algorithm provides a way to reduce this requirement while also giving a generally faster solution.

There remains, however, much to be improved from the current algorithm presented in this paper. The GIS-based segmentation may provide its benefits, but may also pose an additional constraint depending on the project's demands. As we have mentioned briefly in Section 2, the use of machine or deep learning approaches may lessen the reliance on pre-existing, hard-coded algorithmic information such as shapefiles. In this regard, developments toward panoptic deep learning segmentation will be a very interesting approach for future work. Furthermore, the purely data-driven approach for the roof reconstruction may benefit from additional rules and/or geometric constraints in order to improve its overall results, especially when dealing with more complex roof forms. More tests with more varied datasets are also envisaged in order to further validate the performance of the proposed workflow.

Supplementary Materials: M_HERACLES is an open source toolbox written in Matlab[®] and available for download via GitHub in the following webpage: https://github.com/murtiad/M_HERACLES (last updated 15 October 2020). Additionally, the 3D polygon to CityGML conversion program can be accessed in the following link: <https://github.com/veriandi/roofshp2citygml> (last updated 26 October 2020).

Author Contributions: Conceptualization, Arnadi Murtiyoso and Deni Suwardhi; methodology, Arnadi Murtiyoso, Mirza Veriandi, and Deni Suwardhi; software, Arnadi Murtiyoso and Mirza Veriandi; validation, Arnadi Murtiyoso; formal analysis, Arnadi Murtiyoso and Mirza Veriandi; investigation, Arnadi Murtiyoso and Mirza Veriandi; resources, Deni Suwardhi; data curation, Deni Suwardhi, Budhy Soeksmantono, and Agung Budi Harto; writing—original draft preparation, Arnadi Murtiyoso and Mirza Veriandi; writing—review and editing, Deni Suwardhi, Budhy

Soeksmantono, and Agung Budi Harto; visualization, Arnadi Murtiyoso; supervision, Deni Suwardhi and Agung Budi Harto; project administration, Deni Suwardhi and Budhy Soeksmantono; funding acquisition, Deni Suwardhi, Budhy Soeksmantono, and Agung Budi Harto. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by LPPM-ITB under the Research, Community Services, and Innovation Program (*Program Penelitian, Pengabdian kepada Masyarakat, dan Inovasi—P3MI*) grant scheme attributed to the Remote Sensing and GIS Group (KK INSIG). The APC was likewise funded by the same grant.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Grussenmeyer, P.; Hanke, K.; Streilein, A. Architectural Photogrammetry. In *Digital Photogrammetry*; Kasser, M., Egels, Y., Eds.; Taylor & Francis: London, UK, 2002; pp. 300–339.
2. Remondino, F.; Spera, M.G.; Nocerino, E.; Menna, F.; Nex, F. State of the art in high density image matching. *Photogramm. Rec.* **2014**, *29*, 144–166. [[CrossRef](#)]
3. Murtiyoso, A.; Grussenmeyer, P.; Koehl, M.; Freville, T. Acquisition and Processing Experiences of Close Range UAV Images for the 3D Modeling of Heritage Buildings. In *Digital Heritage. Progress in Cultural Heritage: Documentation, Preservation, and Protection: 6th International Conference, EuroMed 2016, Nicosia, Cyprus, 31 October–5 November 2016, Proceedings, Part I*; Ioannides, M., Fink, E., Moropoulou, A., Hagedorn-Saupe, M., Fresa, A., Liestøl, G., Rajcic, V., Grussenmeyer, P., Eds.; Springer International Publishing: Cham, Switzerland, 2016; pp. 420–431.
4. Colomina, I.; Molina, P. Unmanned aerial systems for photogrammetry and remote sensing: A review. *ISPRS J. Photogramm. Remote Sens.* **2014**, *92*, 79–97. [[CrossRef](#)]
5. Corte, A.P.D.; Rex, F.E.; de Almeida, D.R.A.; Sanquetta, C.R.; Silva, C.A.; Moura, M.M.; Wilkinson, B.; Zambrano, A.M.A.; da Cunha Neto, E.M.; Veras, H.F.; et al. Measuring individual tree diameter and height using gatereye high-density UAV-lidar in an integrated crop-livestock-forest system. *Remote Sens.* **2020**, *12*, 863. [[CrossRef](#)]
6. Fraser, C.; Stamatopoulos, C. Automated Target-Free Camera Calibration. In Proceedings of the ASPRS 2014 Annual Conference, Louisville, KY, USA, 23–28 March 2014.
7. Börlin, N.; Grussenmeyer, P. External Verification of the Bundle Adjustment in Photogrammetric Software Using the Damped Bundle Adjustment Toolbox. In Proceedings of the International Society of Photogrammetry and Remote Sensing, Prague, Czech Republic, 12–19 July 2016; Volume XLI-B5, pp. 7–14.
8. Murtiyoso, A.; Grussenmeyer, P. Documentation of heritage buildings using close-range UAV images: dense matching issues, comparison and case studies. *Photogramm. Rec.* **2017**, *32*, 206–229. [[CrossRef](#)]
9. Lumban-Gaol, Y.; Murtiyoso, A.; Nugroho, B. Investigations on the bundle adjustment results from sfm-based software for mapping purposes. In Proceedings of the International Archives of the Photogrammetry, Remote Sensing 434 and Spatial Information Sciences-ISPRS Archives, Riva del Garda, Italy, 4–7 June 2018; Volume XLII-2.
10. Wu, C.; Agarwal, S.; Curless, B.; Seitz, S.M. Multicore bundle adjustment. In Proceedings of the CVPR 2011, Colorado Springs, CO, USA, 20–25 June 2011; pp. 3057–3064.
11. Rupnik, E.; Daakir, M.; Pierrot Deseilligny, M. MicMac—A free, open-source solution for photogrammetry. *Open Geospat. Data Softw. Stand.* **2017**, *2*, 14. [[CrossRef](#)]
12. Granshaw, S.I. Photogrammetric Terminology: Third Edition. *Photogramm. Rec.* **2016**, *31*, 210–252. [[CrossRef](#)]
13. Vosselman, G.; Maas, H.G. Airborne and Terrestrial Laser Scanning. In Proceedings of the 2nd FIG Regional Conference, Marrakech, Morocco, 2–5 December 2003; Whittles Publishing: Dunbeath, UK, 2003; pp. 1–10.
14. Murtiyoso, A.; Remondino, F.; Rupnik, E.; Nex, F.; Grussenmeyer, P. Oblique Aerial Photography Tool for Building Inspection and Damage Assessment. In Proceedings of the International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Denver, CO, USA, 17–20 November 2014; Volume XL-1, pp. 309–313.
15. Rupnik, E.; Nex, F.; Remondino, F. Oblique multi-camera systems-orientation and dense matching issues. In Proceedings of the International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Castelldefels, Spain, 12–14 February 2014; Volume XL-3/W1, pp. 107–114.

16. Nex, F.; Remondino, F. UAV: Platforms, regulations, data acquisition and processing. In *3D Recording and Modelling in Archaeology and Cultural Heritage: Theory and Best Practices*; Chapter Photogrammetry; Remondino, F., Campana, S., Eds.; Archaeopress: Oxford, UK, 2014; pp. 73–86.
17. Murtiyoso, A.; Grussenmeyer, P.; Börlin, N. Reprocessing Close Range Terrestrial and UAV Photogrammetric Projects with the DBAT Toolbox for Independent Verification and Quality Control. In Proceedings of the LowCost 3D 2017, Hamburg, Germany, 28–29 November 2017; Volume XLII-2 W8, pp. 171–177.
18. Jaud, M.; Passot, S.; Le Bivic, R.; Delacourt, C.; Grandjean, P.; Le Dantec, N. Assessing the accuracy of high resolution digital surface models computed by PhotoScan[®] and MicMac[®] in sub-optimal survey conditions. *Remote Sens.* **2016**, *8*, 501. [[CrossRef](#)]
19. Stöcker, C.; Nex, F.; Koeva, M.; Gerke, M. Quality assessment of combined IMU/GNSS data for direct georeferencing in the context of UAV-based mapping. In Proceedings of the The International Archives of the Photogrammetry, 459 Remote Sensing and Spatial Information Sciences, Bonn, Germany, 4–7 September 2017; Volume XLII-2W6, pp. 355–361.
20. Tampubolon, W.; Reinhardt, W. Direct georeferencing of UAV data based on simple building structures. In Proceedings of the International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences—ISPRS 462 Archives, Prague, Czech Republic, 12–19 July 2016; Volume XLI-B1, pp. 1043–1050.
21. Susetyo, D.B.; Hidayat, F. Specification of Map Generalization from Large Scale to Small Scale Based on Existing Data. *Iop Conf. Ser. Earth Environ. Sci.* **2019**, *280*, 012026. [[CrossRef](#)]
22. Putra, T.Y.D.; Sekimoto, Y.; Shibasaki, R. Toward the evolution of national spatial data infrastructure development in Indonesia. *ISPRS Int. J. Geo Inf.* **2019**, *8*, 263. [[CrossRef](#)]
23. Rokhmana, C.A. Percepatan Pemetaan Kadaster Memanfaatkan Teknologi Wahana Udara Tanpa Awak. *BHUMI J. Agrar. Dan Pertanah.* **2013**, *No 38*, 263–268.
24. Elsayed, M.; Mohamed, M. The impact of airspace regulations on unmanned aerial vehicles in last-mile operation. *Transp. Res. Part D Transp. Environ.* **2020**, *87*, 102480. [[CrossRef](#)]
25. Morales, A.C.; Paez, D.; Arango, C. Multi-Criteria Analysis of Uavs Regulations in 6 Countries Using the Analytical Hierarchical Process and Expert Knowledge. In Proceedings of the The International Archives of the Photogrammetry, 473 Remote Sensing and Spatial Information Sciences, Toronto, ON, Canada, 30 August–2 September 2015; Volume XL-1/W4, pp. 175–181.
26. Kemenhub. *Peraturan Keselamatan Penerbangan Sipil (PKPS)*; Kemenhub: Jakarta, Indonesia, 2019.
27. Winczek, M.; Zięba-kulawik, K.; Wężyk, P.; Strojczek-jaźwińska, P.; Bobrowski, R.; Szparadowska, M.; Warchoł, A.; Kiedos, D. LiDAR and image point clouds as a source of 3D information for a smart city—The case study for trees in Jordan Park in Kraków, Poland. In Proceedings of the GIS Ostrava 2020—UAV in Smart City and Smart Region, Ostrava, Czechia, 18–20 March 2020.
28. Murtiyoso, A.; Grussenmeyer, P. Virtual disassembling of historical edifices: Experiments and assessments of an automatic approach for classifying multi-scalar point clouds into architectural elements. *Sensors* **2020**, *20*, 2161. [[CrossRef](#)] [[PubMed](#)]
29. Poux, F.; Neuville, R.; Nys, G.A.; Billen, R. 3D point cloud semantic modelling: Integrated framework for indoor spaces and furniture. *Remote Sens.* **2018**, *10*, 1412. [[CrossRef](#)]
30. Maalek, R.; Lichti, D.D.; Ruwanpura, J.Y. Automatic recognition of common structural elements from point clouds for automated progress monitoring and dimensional quality control in reinforced concrete construction. *Remote Sens.* **2019**, *11*, 1102. [[CrossRef](#)]
31. Grilli, E.; Menna, F.; Remondino, F. A Review of Point Clouds Segmentation and Classification Algorithms. In Proceedings of the International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Nafplio, Greece, 1–3 March 2017; Volume XLII-2/W3, pp. 339–344.
32. Bassier, M.; Vergauwen, M.; Van Genechten, B. Automated Classification of Heritage Buildings for As-Built BIM using Machine Learning Techniques. In Proceedings of the ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Ottawa, ON, Canada, 28 August–1 September 2017; Volume IV-2/W2, pp. 25–30, [[CrossRef](#)]
33. Grilli, E.; Remondino, F. Classification of 3D digital heritage. *Remote Sens.* **2019**, *11*, 847. [[CrossRef](#)]

34. Matrone, F.; Grilli, E.; Martini, M.; Paolanti, M.; Pierdicca, R.; Remondino, F. Comparing machine and deep learning methods for large 3D heritage semantic segmentation. *ISPRS Int. J. Geo Inf.* **2020**, *9*, 535. [[CrossRef](#)]
35. Malinverni, E.S.; Pierdicca, R.; Paolanti, M.; Martini, M.; Morbidoni, C.; Matrone, F.; Lingua, A. Deep learning for semantic segmentation of point cloud. In Proceedings of the The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Ávila, Spain, 1–5 September 2019; Volume XLII-2/W15, pp. 735–742.
36. Rizaldy, A.; Persello, C.; Gevaert, C.M.; Oude Elberink, S.J. Fully Convolutional Networks for Ground Classification from LiDAR Point Clouds. In Proceedings of the ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences, Riva del Garda, Italy, 3–7 June 2018; Volume IV-2, pp. 231–238.
37. Kirillov, A.; He, K.; Girshick, R.; Rother, C.; Dollar, P. Panoptic segmentation. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 9396–9405.
38. Zhang, W.; Qi, J.; Wan, P.; Wang, H.; Xie, D.; Wang, X.; Yan, G. An easy-to-use airborne LiDAR data filtering method based on cloth simulation. *Remote Sens.* **2016**, *8*, 501. [[CrossRef](#)]
39. Vo, A.V.; Truong-hong, L.; Laefer, D.F.; Bertolotto, M. Octree-based region growing for point cloud segmentation. *ISPRS J. Photogramm. Remote Sens.* **2015**, *104*, 88–100. [[CrossRef](#)]
40. Hu, X.; Yuan, Y. Deep-learning-based classification for DTM extraction from ALS point cloud. *Remote Sens.* **2016**, *8*, 730. [[CrossRef](#)]
41. Murtiyoso, A.; Grussenmeyer, P. Point cloud segmentation and semantic annotation aided by GIS data for heritage complexes. *Int. Arch. Photogramm. Remote. Sens. Spat. Inf. Sci.* **2019**, *XLII*, 523–528. [[CrossRef](#)]
42. Liu, C.J.; Krylov, V.; Dahyot, R. 3D point cloud segmentation using GIS. In Proceedings of the 20th Irish Machine Vision and Image Processing Conference, Belfast, UK, 2018; pp. 41–48.
43. Kaiser, P.; Wegner, J.D.; Lucchi, A.; Jaggi, M.; Hofmann, T.; Schindler, K. Learning Aerial Image Segmentation from Online Maps. *IEEE Trans. Geosci. Remote Sens.* **2017**, *55*, 6054–6068. [[CrossRef](#)]
44. Biljecki, F.; Stoter, J.; Ledoux, H.; Zlatanova, S.; Çöltekin, A. Applications of 3D city models: State of the art review. *ISPRS Int. J. Geo Inf.* **2015**, *4*, 2842–2889. [[CrossRef](#)]
45. Verdie, Y.; Lafarge, F.; Alliez, P.; Verdie, Y.; Lafarge, F.; Alliez, P.; Generation, L.O.D.; Acm, S.; Verdie, Y.; Lafarge, F.; et al. LOD Generation for Urban Scenes. *ACM Trans. Graph. Assoc. Comput. Mach.* **2015**, *34*, 15. [[CrossRef](#)]
46. Toschi, I.; Nocerino, E.; Remondino, F.; Revolti, A.; Soria, G.; Piffer, S. Geospatial data processing for 3D city model generation, management and visualization. In Proceedings of the International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Hannover, Germany, 6–9 June 2017; Volume XLII-1/W1, pp. 527–534.
47. Awrangjeb, M.; Gilani, S.A.N.; Siddiqui, F.U. An effective data-driven method for 3-D building roof reconstruction and robust change detection. *Remote Sens.* **2018**, *10*, 1512. [[CrossRef](#)]
48. Gilani, S.A.N.; Awrangjeb, M.; Lu, G. Segmentation of airborne point cloud data for automatic building roof extraction. *Gisci. Remote Sens.* **2018**, *55*, 63–89. [[CrossRef](#)]
49. Alidoost, F.; Arefi, H. A CNN-Based Approach for Automatic Building Detection and Recognition of Roof Types Using a Single Aerial Image. *PFG J. Photogramm. Remote Sens. Geoinf. Sci.* **2018**, *86*, 235–248. [[CrossRef](#)]
50. Nys, G.A.; Poux, F.; Billen, R. City json building generation from airborne LiDAR 3D point clouds. *ISPRS Int. J. Geo Inf.* **2020**, *9*, 521. [[CrossRef](#)]
51. Yang, J.; Kang, Z.; Akwensi, P.H. A Label-Constraint Building Roof Detection Method From Airborne LiDAR Point Clouds. *IEEE Geosci. Remote Sens. Lett.* **2020**, 1–5. [[CrossRef](#)]
52. Tarsha-Kurdi, F.; Landes, T.; Grussenmeyer, P. Hough-Transform and Extended Ransac Algorithms for Automatic Detection of 3D Building Roof Planes From Lidar Data. In Proceedings of the ISPRS Workshop on Laser Scanning 2007 and SilviLaser 2007, Espoo, Finland, 12–14 September 2007; Volume XXXVI, pp. 407–412.
53. Tarsha-Kurdi, F.; Landes, F.; Grussemeier, P. Extended RANSAC algorithm for automatic detection of building roof planes from LiDAR data. *Photogramm. J. Finl.* **2008**, *21*, 97–109.

54. Drešček, U.; Fras, M.K.; Tekavec, J.; Lisec, A. Spatial ETL for 3D building modelling based on unmanned aerial vehicle data in semi-urban areas. *Remote Sens.* **2020**, *12*, 1972. [[CrossRef](#)]
55. Malihi, S.; Zoj, M.J.V.; Hahn, M. Large-scale accurate reconstruction of buildings employing point clouds generated from UAV imagery. *Remote Sens.* **2018**, *10*, 1148. [[CrossRef](#)]
56. Wen, X.; Xie, H.; Liu, H.; Yan, L. Accurate reconstruction of the LOD3 building model by integrating multi-source point clouds and oblique remote sensing imagery. *ISPRS Int. J. Geo Inf.* **2019**, *8*, 135. [[CrossRef](#)]
57. Gröger, G.; Plümer, L. CityGML—Interoperable semantic 3D city models. *ISPRS J. Photogramm. Remote Sens.* **2012**, *71*, 12–33. [[CrossRef](#)]
58. Ogori, K.A.; Biljecki, F.; Kumar, K.; Ledoux, H.; Stoter, J. Modeling cities and landscapes in 3D with CityGML. In *Building Information Modeling*; Springer: Berlin/Heidelberg, Germany, 2018; pp. 199–215.
59. Biljecki, F.; Dehbi, Y. Raise the roof: Towards generating LOD2 models without aerial surveys using machine learning. *ISPRS Ann. Photogramm. Remote. Sens. Spat. Inf. Sci.* **2019**, *IV-4/W8*, 27–34. [[CrossRef](#)]
60. Nex, F.; Gerke, M.; Remondino, F.; Przybilla, H.J.; Baumker, M.; Zurhorst, A. ISPRS benchmark for multi-platform photogrammetry. In Proceedings of the ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Munich, Germany, 25–27 March 2015; Volume II-3/W4, pp. 135–142.
61. Kulur, S.; Yildiz, F.; Selcuk, O.; Yildiz, M.A. The Effect of Pixel Size on the Accuracy of Orthophoto Production. In Proceedings of the ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences, Prague, Czech Republic, 12–19 July 2016; Volume III-4, pp. 53–57.

Publisher’s Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).