EXPLORING EXISTING 3D RECONSTRUCTION TOOLS FOR THE GENERATION OF 3D CITY MODELS AT VARIOUS LOD FROM A SINGLE DATA SOURCE

E. Muñumer Herrero[1,*], C. Ellul[1], S. Cavazzi[2]

[1] Department of Civil, Environmental and Geomatic Engineering, University College London, UK
(estibaliz.herrero.17, c.ellul)@ucl.ac.uk
[2] Ordnance Survey, Southampton, UK – stefano.cavazzi@os.uk

Commission IV, WG IV/9

KEY WORDS: Point clouds, 3D reconstruction, 3D city model, 3D generalisation, level of detail, subsampling.

ABSTRACT:

The use of three-dimensional (3D) city models has increased in a wide range of applications beyond visualisation. However, generation and maintenance of 3D data comes at a high cost, time, and workload. The purpose of the generalisation where coarser versions are obtained from a source data is of great interest for National Mapping and Cadastral Agencies (NMCA), which would benefit obtaining multiple 3D versions of an area from a single source. The main aim of the exploration presented in this paper is to study the potential of downsizing point clouds as an approach to generate 3D city models at multiple levels of detail from a single source and evaluate the steps required to ensure the output is fit for real world applications from an NMCA context. While interesting results are obtained when testing with sample data, no software managed to semi-automatically reconstruct 3D model for buildings of rather complicated geometry.

1. INTRODUCTION

Digital representation of the Earth's surface and the spatial objects on it increased massively since the computerisation era, leading to 3D visualisation of the elements that compose a 3D city model such as buildings, vegetation, roads, containing not only spatial information but thematic too. Modelling cities in 3D may be carried out at different scales varying from city/neighbourhood scale to building-scale and at multiple resolutions or Levels of Detail (LoD) which determine the quantity of the content and detail in the model.

The usage of the resulting 3D city models has increased in numerous scenarios like in urban planning or disaster management and has enabled visual reasoning in certain use cases like virtual tours, navigation, or flight simulations. A broad range of applications benefit of the use of 3D city models for 3D spatial operations and estimations where the visualisation is not a requirement, for instance, energy demand or solar irradiation simulations to name a few (Biljecki et al., 2015).

For every specific application, the lifecycle of 3D city models consists of the generation, management, and usage of the 3D model. The first stage focuses on the design, conception, and data acquisition. Various acquisition sources are employed nowadays including photogrammetry, laser scanning, or surveying amongst others. The stages afterwards include the maintenance, storage and use of the 3D city models.

Given the high cost of data acquisition and processing, particularly at national scale - for instance, (Wong, 2018) estimates a cost of £31 to £389 million for the production of a national 3D mapping product for Great Britain depending on the density of buildings across it - it is considered good practice within National Mapping and Cadastral Agencies (NMCAs) to reuse existing data to meet the needs of different applications and user domains – i.e., to generate multiple maps and models from the same data source, each with a different target user group in mind. This process of deriving the information from a detailed source to a coarser one is known as generalisation. At the same time, different sources of data may provide information about the same features. These can be combined if necessary to generate the required output dataset.

Increasingly, one of the most common sources of data for 3D modelling applications is point cloud data, sourced from airborne Light Detection and Ranging - LiDAR (a method where a laser scanner on an airplane measures distances to obtain accurate 3D point clouds of a surface by measuring the time the reflected light signal takes to return), terrestrial (when the laser scanner is set on a station on the ground) or mobile laser scanning (when the laser scanners move along on a vehicle). For example, *The Environment Agency National LIDAR Programme*[1] data is now available for the United Kingdom (UK), *Actueel Hoogtebestand Nederland* for the Netherlands (see Section 3). However, although many approaches to generating 3D models from this data have been proposed (see Section 2) fewer attempts have been made to generate multiple models at different resolution from the same data source – i.e., to apply point clouds in a 3D generalisation context.

This paper presents a preliminary approach based on the existing 3D reconstruction tools with the aim of obtaining different 3D models depending on the point cloud density - a point cloud at its fullest density versus one of lower density for the same object.

The aim is to investigate the differences between the two representations, which have been obtained from a single source but by synthetically thinning the point cloud, new subsampled versions of point clouds have been obtained, i.e., re-using the

* Corresponding author

[1] data.gov.uk/dataset/f0db0249-f17b-4036-9e65-309148c97ce4/national-lidar-programme (10 March 2022)

data. In particular, we focus on automated reconstruction methods - automating the process of 3D reconstruction is important as to reduce or eliminate human bias.

Investigating the differences between the two (or more) reconstructed 3D models will enable users have a better understanding whether a point cloud based semi-/automated 3D reconstruction approach is suitable for the maintenance and use of those 3D city models, particularly in the case of the 3D generalisation where 3D models of coarser level of detail are created from a finer detailed source.

2. RELATED WORK

Reconstruction of 3D buildings has been a topic of great interest - researchers have been developing a variety of methods from different sources and data types, including point clouds. For instance, (Malihi et al., 2016) present a reconstruction method using dense point clouds obtained from photogrammetric images obtained by UAV (unmanned aerial vehicles).

Automation of the reconstruction process is essential in order to reduce time, costs, and subjectivity on the interpretation of the human operator. Many studies have attempted this automation, for both single building or city (or even national) scale. A method proposed by (Xiong et al., 2013) registers 3D point clouds obtained from laser scanner from different locations and automatically creates semantically rich 3D building model. Aside from techniques that generate results at a single LoD level, there are projects where results for multiple LoD are generated. For example, the collaboration project by (Dukai et al., 2020) proposes a workflow with the aim not only to automatically reconstruct 3D buildings in city models nationwide but to maintain, control and update the 3D database to make it of easier access to a larger number of users. Improvements to the method are published in (Peters et al., 2021) where the large areas are reconstructed automatically to CityGML[2] LoD1.2, LoD1.3 and LoD2.2[3] 3D building models based on 2D building polygons and LiDAR point clouds around the Netherlands, called 3D BAG. The quality of the reconstructed 3D data is assessed with the intention to offer the users with information for their applications as well as to improve the quality of the upcoming releases (Dukai et al., 2021).

For the indoor environment of the building, techniques to automate the 3D modelling have also been presented over the years such as (Budroni and Boehm, 2010) where point clouds from scans are processed to create CAD models for 3D architecture or the method presented by (Boeters et al., 2015) where it automatically creates LoD2 models with indoor details, naming it LoD2+. A first approach integrating exterior and interior data sources as CityGML/JSON and 2D floor plan images for the reconstruction of 3D city models is proposed by (Kippers et al., 2021), which is based on deep learning methods. While promising results are obtained for less detailed and geometrically complex floor plans, the authors suggest including more representative training data to the machine learning model to improve their method.

[2] CityGML is a conceptual model and exchange format for 3D City Modelling, which defines a number of Levels of Detail (LoD) for different levels of complexity of a 3D City model https://www.ogc.org/standards/citygml (22 April 2022).

[3] CityGML LoDs were extended by (Biljecki et al., 2016) where LoD1.2 depicts building recesses and extensions, LoD1.3 same as the previous but at multiple heights, and LoD2.2 includes roof structure and other details such as chimneys.

3. DATA

The aim of this research is to explore the possibilities of using a point cloud thinning approach to generate different resolutions of 3D city models. As noted in Section 2, a number of different approaches have been created to generate 3D city models from point clouds. A desktop review (see Section 4) was carried out to identify appropriate software for testing, and three point cloud datasets then selected for use in the resulting software. The first two are chosen after following the developers' suggestions and/or based on the sample data from the software package. The third was provided by Ordnance Survey (OS, the NMCA for Great Britain) in order ensure that a realistic dataset was also used in the testing.

3.1 Amsterdam

'AHN3' (*Actueel Hoogtebestand Nederland 3)*' is an open source, Netherlands-wide, geodata set containing detailed and precise 3D altitude information from airplanes and helicopters collected by laser technology, which include 3D point cloud datasets (LAS files)[4]. The centre of Amsterdam city is included in the *AHN3 '25GN1'* map sheet which is explored for its combination of buildings of different height (modern versus 17th century architecture style), the canals and roads (see Figure 1). Further details about the point cloud are displayed in Table 1.

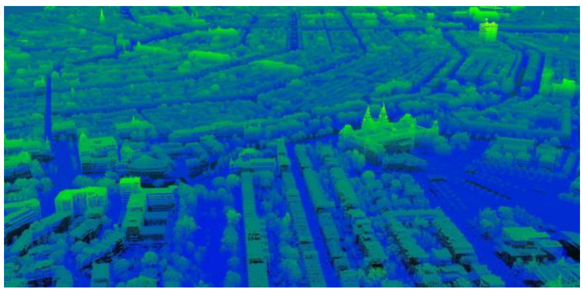

Figure 1. Point cloud of Amsterdam city obtained from AHN3.

Dataset	*# Points*	*Points per m^2*	*Points per m^3*
Amsterdam Original	527,162,888	16.869	0.106

Table 1. Number and density of points in the dataset.

3.2 PolyFit Sample data

One of the approaches for 3D reconstruction from point clouds found was 'PolyFit' -see Section 5.1.4- and in order to test it, their sample data was valuable for the experiment[5]. The downloaded folder contains a variety of buildings and other objects such as a sphere or a toy from which two buildings were selected for further testing: a detached house and a tall building for several reasons (see Figure 2):

- The façades include elements such as windows, doors, garage gates, balcony, etc.
- The roof of the detached house is irregular, with inclination and roof windows are well scanned.
- Both façades and roof have detailed information about the tiles (rather than a smooth flat surface).
- Information of the colour of the point, i.e., the intensity, is provided as real colours (RGB).

[4] app.pdok.nl/ahn3-downloadpage/ (31 March 2022)

[5] 3d.bk.tudelft.nl/liangliang/publications/2017/polyfit/polyfit.html (31 March 2022)

https://doi.org/10.5194/isprs-annals-X-4-W2-2022-209-2022

Further details about the point cloud are displayed in Table 2.

Dataset	***# Points***	***Points per m²***	***Points per m³***
Detached house	4,752,937	16,762.896	1,576.453
Tall building	72,821	569.521	52.634

Table 2. Number and density of points in each PolyFit dataset.

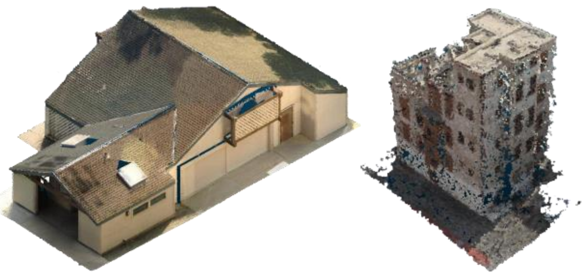

Figure 2. Point clouds from PolyFit sample data – Detached house on the left, tall building on the right.

3.3 Romsey Abbey

The Ordnance Survey Research department have carried out several acquisitions of the parish church in Romsey, Hampshire, England[6] and the experimental data acquired has been provided for this piece of research: on the one hand, point clouds obtained from various sensors in different terrestrial surveys are provided. These cover most of the façades but only a part of the roof. On the other hand, mesh outputs based on various aerial imagery acquisitions are given too, which help deal with the uncovered areas from the terrestrial scans. After transforming the meshes to point cloud format, all the clouds are set to the same coordinate system. The resulting registered (Section 4.2.1-F) point cloud (see Figure 3 and further detail in Table 3) covers the whole Romsey Abbey, including details on the roof, and it is ready to be used in coming steps (see Section 4.3.3).

Figure 3. Registered point cloud from Romsey Abbey.

Dataset	***# Points***	***Points per m²***	***Points per m³***
Romsey Abbey	39,840,832	10,393.800	154.874

Table 3. Number and density of points after the registration.

4. METHODOLOGY

In order to investigate the possibility of generating multiple 3D models from point clouds via thinning, the methodology aims to reconstruct 3D models from point clouds of various densities and compare the results obtained.

[6] romseyabbey.org.uk/contact-us/ (31 March 2022)

Four 3D reconstruction tools (3dfier, ArcGIS API for Python, Python Scripts and PolyFit) from both the commercial and academic fields are chosen for the exploration which is split in three main parts: An exploration and desktop review of the four selected tools (see Section 4.1), followed by testing the 3D reconstruction algorithm (see Section 4.2) and lastly, some further tests to evaluate the capabilities of the tool (see Section 4.3). The results obtained are explained in Section 5.

While it can be expected that software will produce useful results when evaluated using the provided sample data, the aim of the experiments described here are to explore the outcomes obtained when point density of the sample data is reduced, and when the algorithms are run against data from third parties. This will allow the evaluation, in a real world context of the potential of point clouds aa source for 3D models of different resolutions.

4.1 Methodology Part 1: Desktop review of software

The first step is a desk-based exploration of existing automated and semi-automated tools to reconstruct models. Candidate 3D reconstruction software was identified via an initial Internet search and for each selected tool documentation was reviewed in detail. Criteria forming part of the document review included:

- Source: consider the possible obstacles obtaining the tool, whether a software is free, if it runs under certain licence conditions, or if it is open source enabling modification of the algorithm if necessary.
- The background of the source (academia, commercial or other) can determine whether the software is well established and used by a wide number of people, contexts, or applications, and hence, well supported.
- The resolution of the reconstruction may vary from a simple extruded block to a detailed model (for instance, inc. information or the roof of façade elements)– A tool that obtains a more detailed solution from a dense point cloud may be able to generate less detailed models from thinned clouds
- Reconstruction algorithms that can be applied not only to individual but to group of buildings.
- An examination of the supported formats indicates the level of versatility - a wider number of formats (for both input and output) facilitate the usability of the tool and transferability of the resulting reconstructed 3D model.
- Additionally, checks were made to confirm whether the tools perform other additional processes on top of the reconstruction itself, such as thinning the point cloud or classifying the points are signs of a level higher in the expansion of the tool.

4.1.1 Reviewing the results: The results of the inspection for the criteria above (see Section 4.1) for each software are tabulated and compared so that the most appropriate package is selected. Special attention is paid to the LoD of the reconstructed model – the software is discarded if the 3D models are obtained at LoD1.1 or less (based on the LoD specifications by (Biljecki et al., 2016)). A point system is defined for the evaluation of each criterion and software, where the higher the total of points obtained enable further exploration of the tool:

0 points : unacceptable results.
1 point : acceptable results.
2 points : better than acceptable / positive results.
3 points : very good results.
4 points : the performance or the results are excellent.

https://doi.org/10.5194/isprs-annals-X-4-W2-2022-209-2022 |

4.2 Methodology Part 2: Testing and evaluation - Sample Data

The software packages selected in the first part of the methodology are tested in this stage with the objective of checking whether the tool performs as the vendors or developers promise. An analysis of the obtained results will enable a decision as to which tool is worth exploring further.

4.2.1 Preparing the software and data: The initial data for reconstruction testing will be the sample data provided by the developers which -it is assumed- will ensure the algorithm will perform as expected (see Section 3). It is possible that the data will require pre-processing in order to be completely ready for the algorithm, which could include:

A. *Decompression* – where the software packages require '.LAS' files, the downloaded '.LAZ' files need to be decompressed first.
B. *Conversion* – The downloaded data will be converted into another data type or file format the software requires.
C. *Noise removal* – Unwanted objects or artifacts might need to be removed from the cloud for a better reconstruction.
D. *Merging* – A given object or building might be segmented in different files which will need to be merged first in order to obtain the full coverage of the object.
E. *Reference system* – While point cloud data in a relative coordinate system could be sufficient for some projects, in others a coordinate reference system must be designated.
F. *Registration* – Somewhat linked to the previous two steps, different point clouds are unified by this process by geometric alignment from the same reference system (Bellekens et al., 2014).
G. *Surface normals* – Where the reconstruction software does not provide this option, a pre-process of calculating those is required for the creation of polygonal meshes (Remondino, 2003).
H. *Classification* – The process where a class is assigned to every point.
I. *Segmentation* – The practice where the data is split or divided, for instance when a file is too large, and a software cannot load it or process it further.

For each dataset, a process of *subsampling* is applied – various levels of point cloud density may offer 3D models at different LoDs and if those are not provided from source, they can be synthetically prepared by thinning the original point cloud at different densities. From the several randomly subsampled versions of the original cloud, this paper includes the significant results obtained for each dataset (see Table 4).

Dataset	***# Points***	***Points per m²***	***Points per m³***
Amsterdam - 75%	395,372,166	12.652	0.079
Amsterdam - 50%	263,581,444	8.435	0.053
Amsterdam - 25%	131,790,722	4.217	0.027
Detached house - 0.95m	734	2.651	0.250
Detached house - 1m	155	0.602	0.057
Tall building - 80%	58,257	455.972	42.176
Tall building - 75%	54,616	427.474	39.540
Tall building - 20%	14,564	113.991	10.544

Table 4. Details of the created subsampled datasets.

4.2.2 Testing the 3D reconstruction: This stage involves testing the reconstruction process, i.e., the input point cloud is transformed into one (or more) 3D building. A key point to be confirmed is whether 3D models of higher definition and detail are obtained from the point clouds of higher density compared to the ones obtained from less dense point clouds.

4.2.3 Reviewing the reconstruction results: Summarising the processes, the files or data used, and the results obtained will help understand which package offers a valid 3D reconstructed building in different LoDs for further explorations. A visual inspection of the results (where results were obtained) will allow the following evaluation:

- Q1: Without going into further checks such as orthogonality or volume changes analyses, a visual look to the reconstructed building can suggest whether it is watertight as long as there is not any noticeable hole in the mesh.
- Q2: Is the incline of the roof maintained, reduced or have they been flattened? This is directly linked to the LoD the tool is capable of handling, LoD2 onwards if incline and shape is preserved or LoD1 if the roof is flat and similar to a reconstruction by extrusion.
- Q3: Linked to the previous, the more features present (such as windows, doors, or balconies) the higher the LoD the package can manage, thus, more options to analyse the changes from a level to the other, i.e., the concurred generalisation.

These criteria will determine which package is selected for further testing.

4.3 Methodology Part 3: Testing beyond sample data

Only the software packages that have passed the previous steps of exploration, test and evaluation (see Section 4.1 and 4.2) are considered for further testing.

4.3.1 Test 1: 'Complex Buildings': The methodology presented in Section 4.2.2 is applied to point clouds of geometrically complex buildings.

4.3.2 Test 2: 'Mesh from a mesh': Where a mesh for a building already exists, it is dismantled and converted into a point cloud and used as input data for the reconstruction process (by following the methodology in Section 4.2.2).

4.3.3 Test 3: 'Merge meshes + Mesh from a mesh': In circumstances where the point cloud does not fully cover the building, the data from different sources need to be merged first. The list of data pre-processing steps explained in Section 4.2.1 (A-I) are the reference to follow for either merging the files of the same type or registering point clouds from difference coordinate reference systems. Once the clouds are ready, methodology proposed in Section 4.3.2 is applied.

5. RESULTS[7]

5.1 Results of Methodology Part 1: Desktop review

Four tools were identified from the initial search:

[7] Note that as the focus of this work was the fitness for purpose of the resulting buildings, processing time was not considered/measured. This would vary depending on the specification of the computer being used and the point cloud density. Given the single-building focus of the tests, parallelization of the task would also be possible.

	3dfier	ArcGIS API for Python	Python Scripts	PolyFit
Licence	Free under GNU General Public Licence[8]	(i) ArcGIS Pro & Extensions (ii) ArcGIS Runtime Advanced	Free under MIT Licence[9]	Free under GNU General Public Licence[7]
Background	Academia	Commercial	Academia	Academia
Reconstructed model LoD	LoD1.2	LoD1.2	LoD2.2	LoD2.2
Individual buildings	Yes	Yes	Yes	Yes
Group of buildings	Yes	Yes	No	No
Input - Supported formats	(i) Topologically connected polygons / Vector data supported by GDAL (ii) Point clouds / LAS, LAZ	Point clouds LAS	Point clouds XYZ, XYZN, XYZRGB, PTS, PLY, PCD	Point clouds VG, BCG, OBJ, PNG, JPG
Output (reconstruction)	- MultiSurfaces triangulated - Solids	Building multipatches	3D meshes	Polygonal surface models
Output formats	CityJSON, OBJ, STL, CityGML, IMGeo, CSV, Shapefile, PostGIS, GDAL	File Geodatabase Feature Classes (Multipatches)	OBJ, PLY,STL, OFF,GLTF / GLB	OBJ
Point cloud classification	No	Yes	No	No
Point cloud thinning	Yes	No	Yes	No

Table 5. Comparison of the key points for each software which help determine which one is worth for further testing.

5.1.1 3dfier: The 3D geoinformation research group at TU Delft released this software which automatically reconstructs 3D buildings from 2D geographical datasets such as topographic maps (Ledoux et al., 2021). Building polygons are lifted to the 3rd dimension, i.e., the elevation. Compared to the extrusion tools of commercial software, this package takes the semantics of the polygons in consideration.

5.1.2 ArcGIS API for Python: The ArcGIS Developers platform offers an API for Python for creating building models using point cloud classification (ESRI ArcGIS Developer, 2021). In other words, raw point clouds are first classified and then 3D buildings are generated.

5.1.3 Python Scripts automatically reconstruct 3D surfaces from point clouds. Scripts of two surface reconstruction processes presented by (Poux, 2020) are selected for the experiment: Ball-Pivoting Algorithm (BPA) and the Poisson Reconstruction. Libraries required are '*Numpy'* and '*Open3D'*.

5.1.4 PolyFit: Developed by (Nan and Wonka, 2017) it is a polygonal surface reconstruction framework from point clouds. RANSAC algorithm (Schnabel et al., 2007) is deployed to extract planes from the input point cloud, which get merged in pairs and fit into new planes iteratively in a refinement process. A bounding box clips these planes and pairwise intersections are determined. The optimal subset -candidates- of faces is chosen to reconstruct a watertight polygonal surface model.

5.1.5 Review of the results: Table 5 summarises the comparison results. From the perspective of 3D generalisation, the level of detail of the reconstructed models is of superior importance as the higher LoD the higher possibility of transforming it into coarser LoDs. All the tools can reconstruct models of LoD higher than LoD1.1 – 3dfier and ArcGIS API for Python create 3D models of LoD1.2 while Python Scripts and PolyFit reach LoD2.2.

ArcGIS API for Python is the only commercial package with a restricted licence - although exemptions apply for students. While this could be considered a disadvantage, it also suggests that the tool is well stablished and used by a wider number of users or applications, hence, well supported. The other three software are free and open source which permits an easier accessibility to the tool.

Linked to the degree of development of the package, all of them reconstruct single buildings but only 3dfier and ArcGIS API for Python can process more than one at the time. All except PolyFit offer additional processes such as classification or subsampling of the point clouds. The four tools support a wide range of formats which translates in a good transferability of the reconstructed 3D model.

Paying attention to the total of points assigned to each software per criterion (see Table 6), 3dfier and ArcGIS API for Python are the two software packages that scored the least points. Both reconstruct 3D models of LoD1.2 and have additional functionality. Although 3dfier offers many benefits like labelling the 3D models semantically, the watertightness and the many supported file formats, the resulting 3D buildings do not include finer details such as windows or doors and the roofs lose their incline.

	3dfier	*ArcGIS API for*	*Python Scripts*	*PolyFit*
Source	3	2	3	3
Background	2	3	2	2
LoD	1	1	4	4
Processing buildings	3	3	2	2
Supported formats	3	3	3	3
Additional functionalities	3	3	3	2
Total Points (Σ)	**15**	**15**	**17**	**16**

Table 6. Number of points awarded to each criterion.

[8] gnu.org/licenses/gpl-3.0.en.html (31 March 2022)
[9] opensource.org/licenses/MIT (31 March 2022)

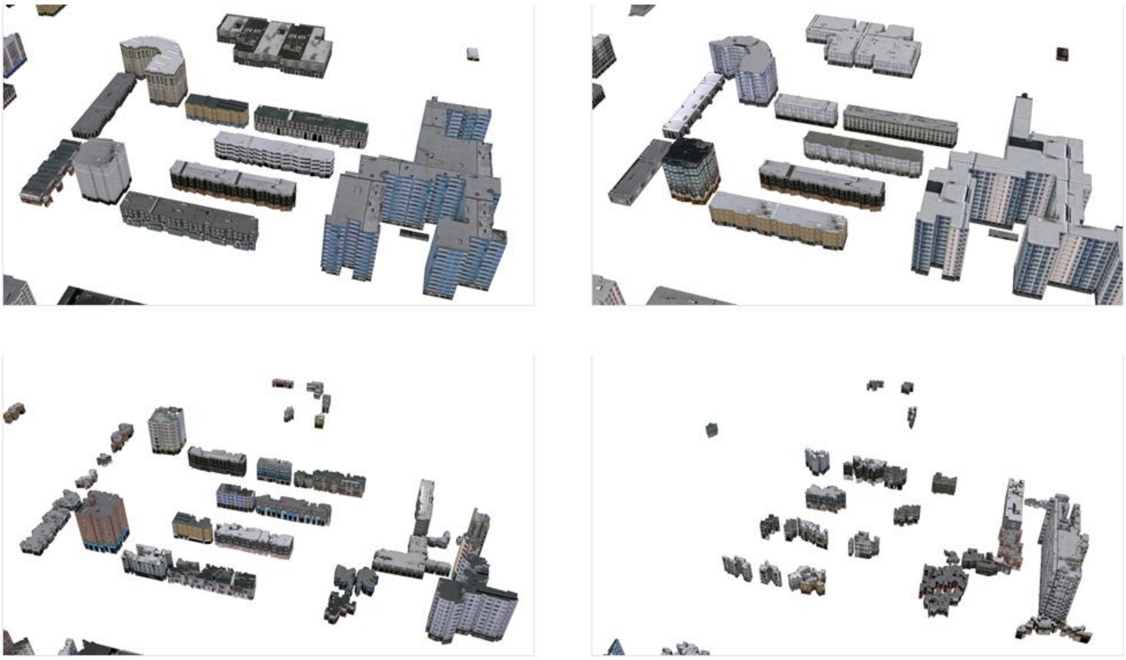

Figure 4: 3D building multipatches of Amsterdam reconstructed with ArcGIS API from the original point cloud (100% - top left) and its reduced versions (75% - top right, 50% - bottom left, 25% - bottom right).

ArcGIS API for Python carries the same disadvantage, however, if the tool is commercially well supported, potential future updates can obtain reconstructed models in higher LoD.

5.2 Results of Methodology Part 2: Exploration and desktop review

5.2.1 Preparing the software and the data – Results:

- ArcGIS API for Python series of steps proposed by the developers were replicated. The data used is the one from Amsterdam (Section 3.1) and in order to have be ready for the test it was decompressed (Section 4.2.1-A), the correct spatial reference system assigned (Section 4.2.1-E) and subsampled so that the package was tested for different point cloud densities. Following the developers guidelines, a classification of the point cloud is done too (Section 4.2.1-H). The package runs a deep learning model for the reconstruction process and data must be prepared beforehand segmented in separate sets for training, validation, and testing (Section 4.2.1-I).

- Python Scripts: Anaconda distribution package is set up with Python environment, including '*Numpy*', '*Matplotlib*' and '*Open3D*' libraries. The sample data suggested from the developer in the instructions correspond to a small figurine, hence, in order to get some comparable values in line with the results obtained from other packages, the data used here is the sample data for PolyFit. The data preparation steps are explained in the point below, however, in addition to those, the point clouds were converted from their original format -binary vertex group, BVG- to XYZ so that were ready to be used in the script (Section 4.2.1-B).

- PolyFit version 1.5 -the latest available at the time of the test- has been built from source code[10]. The sample data (see Section 3.2) downloaded from the developers' website is first decompressed (Section 4.2.1-A), and imported to *Mapple*[11] software so subsampled point clouds are computed. This process was carried out for both the detached house and the tall building.

- Summary of the results – With regard to the pre-processing of the data (Table 7) ArcGIS API for Python needed more steps than the others, a fact somewhat conditioned by the data itself rather than the package since raw data from a city contains lots of information as opposed to a noise-free point cloud of a single building.

Data pre-process	*ArcGIS API for Python*	*Python Scripts*	*PolyFit*
A. Decompression	✓	✓	✓
B. Conversion	×	✓	✓
C. Noise removal	×	×	×
D. Merging	×	×	×
E. Georeferencing	✓	×	×
F. Registration	×	×	×
G. Normals calc.	×	✓	×
H. Classification	✓	×	×
I. Segmentation	✓	×	×
Subsampling	✓	✓	✓

Table 7. Pre-processing steps applied to the data so that is ready for the reconstruction test (✓: Applied / ×: Not applied).

5.2.2 Testing the 3D reconstruction – Results:

Along with the reconstructed 3D building multipatches (Figure 4), the footprints of those were returned when processing ArcGIS API for Python. Considering that the point cloud has a density of 100%, subsampled versions of it were synthetically created (75%, 50% and 25%) and reconstructed.

The Python Scripts managed to generate 3D model of the detached house with the use of both BPA (Figure 5) and

[10] https://github.com/LiangliangNan/PolyFit/releases/tag/v1.5 (7 April 2022)

[11] https://3d.bk.tudelft.nl/liangliang/software.html (7 April 2022)

Poisson reconstruction algorithms. Tested with the original point cloud - of a density of a point every 0.01m - and a subsampled version of it - with a point every 1m.

Figure 5: BPA 3D reconstruction of the detached house for the original point cloud (left) and the subsampled one (right).

The detached house was processed in PolyFit (Figure 6), in its raw version – point cloud density of a point every 0.01m – and the subsamples – a point every 0.95m and every meter.

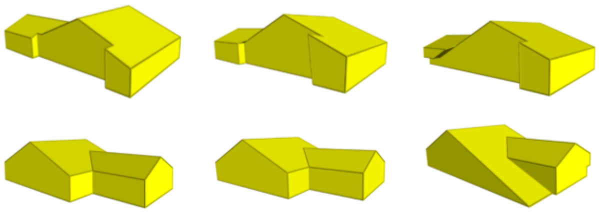

Figure 6: Reconstructed detached house in from the original point cloud (left) and the subsamples (0.95m middle, 1m right).

Similarly, for the tall building (Figure 7), subsampled version of the origianl point cloud are synthetically generated randomly. Where the original point cloud has 100% density, the thinned version are at 80%, 75% and 20%.

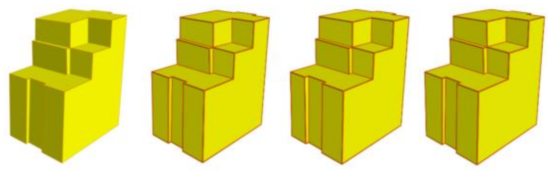

Figure 7: Reconstructed tall building (left to right: 100%, 80%, 75% and 20% point cloud density).

5.2.3 Review of the reconstructed results: The reconstruction packages chosen for the test managed to run the process from beginning to end in both original and down-sized version of the point cloud. The questions suggested in Section 4.2.3 are answered in Table 8. PolyFit has the highest score.

			Q1	*Q2*	*Q3*	*Σ*
ArcGIS API	Amsterdam	Raw	1	-1	-1	-1
		Subsample	1	-1	-1	-1
Python Scripts	Detached house	Raw	-1	1	0	0
		Subsample	-1	1	-1	-1
PolyFit	Detached house	Raw	1	1	-1	1
		Subsample	1	1	-1	1
	Tall building	Raw	1	1	-1	1
		Subsample	1	1	-1	1

Table 8. Points awarded in every question to every software on in each scenario. Scores given as '1' for '*Yes*', '-1' for '*No*' and '0' for '*Not completely*'.

5.3 Results of Methodology Part 3: Further testing beyond sample data

5.3.1 Test 1: PolyFit 'Complex Buildings': Earlier tests proved that PolyFit reconstructs meshes from sample data. Here, PolyFit was tested with point clouds of Romsey Abbey (Section 3.3) obtained from terrestrial and mobile surveys, which include a good example of a complex building.

Here, PolyFit was tested with point clouds of Romsey Abbey (Section 3.3) obtained from terrestrial and mobile surveys, which include a good example of a complex building. Prior to the reconstruction, the original point cloud was downsized so that both the raw point cloud and a subsampled version of it are tested in the reconstruction. In any case, the software was unable to finish the entire process and no 3D reconstructed model was obtained.

5.3.2 Test 2: PolyFit 'Mesh from a mesh': The OS not only provided data from the Romsey Abbey (Section 3.3) in point cloud form but mesh formats too, automatically generated from aerial imagery[12]. This test aims to experiment whether PolyFit can generate meshes for this dataset too. The given OBJ files needed a pre-treatment such as merging all the files to obtain a full coverage (Section 4.2.1-D) and a conversion to point clouds (Section 4.2.1-B).The normals of the points in the cloud were calculated beforehand (Section 4.2.1-G) too. Nevertheless, PolyFit was unable to process a successful 3D model for this point cloud.

5.3.3 Test 3: PolyFit 'Mesh meshes + Mesh from a mesh': The data used on the test above (Section 5.3.2) was generated from aerial imagery which results in certain parts of the building - mainly vertical information on the façades- not being covered. As this could have been an issue on the previous step, point clouds from terrestrial and mobile scanners were registered to the cloud (Section 4.2.1-F, Figure 3) to process this Test 3. A higher coverage comes with a higher computational cost and the software was unable to reconstruct a 3D surface.

5.3.4 Review of the results: The objective of the tests above is to explore if PolyFit reconstructs 3D buildings from increasingly thinned point clouds semi-automatically. No results were returned, therefore, PolyFit is not suitable to process point clouds of geometrically complex buildings like the cylindrical part of Romsey Abbey.

	Result(s)
Test 1 - 'Complex Buildings'	✗
Test 2 - 'Mesh from a mesh'	✗
Test 3 - 'Mesh meshes + Mesh from a mesh'	✗

Table 9. Results obtained for each test.

6. DISCUSSION

This paper set out to explore the potential of point thinning as an approach to generating multiple 3D city models at different level of resolution from the same data source, given the increasing availability of point cloud data within an NMCA context.

All the reconstruction algorithms returned some results when sample data was employed, however the results obtained for the subsampled versions of the point clouds were not as useful. This could be due to the downsizing of the cloud was done randomly, the density of the downsized cloud not being dense enough (this information was not clearly specified for each software package) or other quality and variability factors of the underlying data (e.g., for Romsey Abbey). Additionally, the Python algorithm (5.1.3) produced a mesh output which did not take orthogonality of walls into account (unlike the PolyFit algorithm). The results obtained with the downsized versions of the sample data suggest that further research is needed into what

[12] From personal communication with OS Innovation & Research team

is an expected outcome of a 3D, i.e., the level of detail expected in relation with the level of simplification. The lack of results when testing Romsey Abbey data contradict those cited in Section 2; Peters et al. (2021) managed to generate multiple LoD from the same cloud by partitioning the roof planes into lower complexity models.

Overall, the results demonstrate that while some reconstruction approaches perform successfully with their sample data no software is capable to semi-automatically reconstruct 3D models for buildings of rather complicated geometry.

7. CONCLUSION AND FURTHER WORK

This paper presents a methodology to compare existing approaches for the automatic (or semi-) transformation of point clouds into 3D building models. The methodology steps and software packages tested in every step are described (Section 4) and the result obtained (Section 5). From the perspective of reusing and repurposing 3D data there is room for further work from the point cloud front, such as analysing whether point cloud density variations influence what automated reconstruction approaches deliver in further examples of different shapes and complexity. Additionally, once it is possible to generate results that pass visual/manual inspection, metrics should be developed to automatically assess their quality. These might relate to measuring and documenting positional accuracy (e.g., against the original point cloud) but also to describing how values such as the area of individual components of the building - the roof, windows, doors, walls – vary depending on changing point cloud density.

As the use of detailed 3D data is increasing in a wide range of applications, generating different products from a single source is of great importance. The field of 3D generalisation is particularly concerned in that aspect and experiments where various 3D models of different LoD are reconstructed from the same source in different levels of point cloud density are of great interest.

ACKNOWLEDGEMENTS

The work presented in this paper has been conducted as part of a PhD studentship financed by the Engineering and Physical Sciences Research Council (EPRSC), Ordnance Survey (OS) and the University College London (UCL).

REFERENCES

Bellekens, B., Spruyt, V., Berkvens, R., Weyn, M., 2014. A Survey of Rigid 3d Pointcloud Registration Algorithms. *AMBIENT 2014: The Fourth Int. Conference on Ambient Computing, Applications, Services and Technologies, August 24-28, 2014, Rome, Italy*, 8–13.

Biljecki, F., Ledoux, H., Stoter, J., 2016. An improved LOD specification for 3D building models. *Computers, Environment and Urban Systems,* 59, 25–37. doi.org/10.1016/j.compenvurbsys.2016.04.005.

Biljecki, F., Stoter, J., Ledoux, H., Zlatanova, S., Çöltekin, A., 2015. Applications of 3D City Models: State of the Art Review. *ISPRS Int. J. Geo-Inf.*, 4(4), 2842–2889. doi.org/10.3390/ijgi4042842.

Boeters, R., Arroyo Ohori, K., Biljecki, F., Zlatanova, S., 2015. Automatically enhancing CityGML LOD2 models with a corresponding indoor geometry. *Int. J. Geogr. Inf. Sci.*, 29, 2248–2268. doi.org/10.1080/13658816.2015.1072201.

Budroni, A., Boehm, J., 2010. Automated 3D Reconstruction of Interiors from Point Clouds. *Int. J. Archit. Comput*. 8, 55–73. doi.org/10.1260/1478-0771.8.1.55.

Dukai, B., Peters, R., Vitalis, S., Van Liempt, J., Stoter, J., 2021. Quality assessment of a nationwide data set containing automatically reconstructed 3d building models. *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci,* 46-4/W4-2021, 17-24. doi.org/10.5194/isprs-archives-XLVI-4-W4-2021-17-2021.

Dukai, B., Peters, R., Wu, T., Commandeur, T., Ledoux, H., Baving, T., Post, M., Van Altena, V., Van Hinsbergh, W., Stoter, J., 2020. Generating, storing, updating and disseminating a countrywide 3D model. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* 44-4/W1-2020, 27–32. doi.org/10.5194/ISPRS-ARCHIVES-XLIV-4-W1-2020-27-2020.

ESRI ArcGIS Developer, 2021. Creating building models using point cloud classification. developers.arcgis.com/python/sample-notebooks/creating-building-models-using-point-cloud-classification/ (5 April 2022).

Kippers, R.G., Koeva, M., van Keulen, M., Oude Elberink, S.J., 2021. Automatic 3D Building Model Generation Using Deep Learning Methods Based On CityJSON and 2D Floor Plans. *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci*, 46-4/W4-2021),, 49–54. doi.org/10.5194/isprs-archives-XLVI-4-W4-2021-49-2021.

Ledoux, H., Biljecki, F., Dukai, B., Kumar, K., Peters, R., Stoter, J., Commandeur, T., 2021. 3dfier: automatic reconstruction of 3D city models. *Journal of Open Source Software,* 6(57), 2866. doi.org/10.21105/joss.02866.

Malihi, S., Valadan Zoej, M.J., Hahn, M., Mokhtarzade, M., Arefi, H., 2016. 3D Building Reconstruction Using Dense Photogrammetric Point Cloud. Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci. XLI-B3, 71–74. https://doi.org/10.5194/ISPRS-ARCHIVES-XLI-B3-71-2016.

Nan, L., Wonka, P., 2017. PolyFit: Polygonal Surface Reconstruction from Point Clouds. *IEEE International Conference on Computer Vision, Venice, Italy*, 2372–2380. doi.org/10.1109/ICCV.2017.258.

Peters, R., Dukai, B., Vitalis, S., van Liempt, J., Stoter, J., 2021. Automated 3D reconstruction of LoD2 and LoD1 models for all 10 million buildings of the Netherlands. *Photogramm. Eng. Remote Sens.*, 88, 165–170. doi.org/10.48550/arxiv.2201.01191.

Poux, F., 2020. Generate 3D meshes from point clouds with Python. *Towards Data Science*, towardsdatascience.com/5-step-guide-to-generate-3d-meshes-from-point-clouds-with-python-36bad397d8ba (5 April 2022).

Remondino, F., 2003. From point cloud to surface: the modeling and visualization problem. *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.,* XXXIV-5/W10. doi.org/10.3929/ethz-a-004655782.

Schnabel, R., Wahl, R., Klein, R., 2007. Efficient RANSAC for Point-Cloud Shape Detection. *Computer Graphics Forum*, 26, 214–226. doi.org/10.1111/J.1467-8659.2007.01016.X.

Wong, K.K.Y., 2018. Towards a National 3D Mapping Product for Great Britain. *Doctoral thesis*. University College London.

Xiong, X., Adan, A., Akinci, B., Huber, D., 2013. Automatic creation of semantically rich 3D building models from laser scanner data. *Autom. Constr.* 31, 325–337. doi.org/10.1016/J.AUTCON.2012.10.006.