

# AUTOMATIC HERITAGE BUILDING POINT CLOUD SEGMENTATION AND CLASSIFICATION USING GEOMETRICAL RULES

A. Murtiyoso\* and P. Grussenmeyer

Photogrammetry and Geomatics Group, ICube Laboratory UMR 7357, INSA Strasbourg, France –  
(arnadi.murtiyoso, pierre.grussenmeyer)@insa-strasbourg.fr

Commission II, WG II/8

**KEY WORDS:** Point Cloud, Segmentation, Heritage, Automation, Classification

## ABSTRACT:

The segmentation of a point cloud presents an important step in the 3D modelling process of heritage structures. This is true in many scale levels, including the segmentation, identification, and classification of architectural elements from the point cloud of a building. In this regard, historical buildings often present complex elements which render the 3D modelling process longer when performed manually. The aim of this paper is to explore approaches based on certain common geometric rules in order to segment, identify, and classify point clouds into architectural elements. In particular, the detection of attics and structural supports (i.e. columns and piers) will be addressed. Results show that the developed algorithm manages to detect supports in three separate data sets representing three different types of architecture. The algorithm also managed to identify the type of support and divide them into two groups: columns and piers. Overall, the developed method provides a fast and simple approach to classify point clouds automatically into several classes, with a mean success rate of 81.61% and median success rate of 85.61% for three tested data sets.

## 1. INTRODUCTION

The segmentation of 3D point cloud data is a much discussed problem in the geomatics community. Recent advances in photogrammetry and laser scanning, coupled with the advent of the democratisation of drones, meant that 3D point cloud can now be obtained with higher levels of detail (Murtiyoso et al., 2018). While this presents a great advantage to heritage recording, it also adds to the complexity of the point cloud segmentation. This is more so in the case of complex heritage buildings, which translates further into more effort and time required to perform it manually (Barsanti et al., 2017).

There are two often distinct parts in the pre-processing of point clouds within the 3D modelling workflow. The raw point cloud is usually segmented into smaller clusters representing certain elements of the object, and then classified into object classes (e.g. pillar, arch, floor, etc.). Unfortunately, the 3D modelling process still retains a large part which requires manual intervention (Macher et al., 2017). Attempts to automate any part of this process will greatly save both time and resources in the overall workflow.

Segmentation is an important part of the 3D modelling process, from which we may derive semantic-rich models such as 3D GIS or BIM (Building Information Model) (Campanaro et al., 2016; Wang et al., 2015). Depending on the level of complexity, the process of manual segmentation may take a lot of time. Furthermore, point cloud automatic classification has also become more and more important in this regard. Classification will confer classes into the point cloud clusters resulting from the segmentation step.

This paper will describe an approach to automatically segment a historical building point cloud into architectural elements. In doing this, several simple Euclidean geometry-based rules are used in order to first distinguish the different building element

units e.g. roofs, structural support, floors, etc. Afterwards, by using a slicing approach to the point cloud, structural supports will be identified as point cloud clusters. Geometric rules will then be employed to help classify the segmented results into either the "column" or "pier" class automatically.

## 2. RELATED WORK

Structural supports such as columns present a particular interest for the heritage community, as often times they present a valuable example of historical engineering and architectural design. Much study has been done in the field of structural support automatic 3D modelling (Luo and Wang, 2008; Riveiro et al., 2016), but most focuses on simple pillars or supports. In this regard, automation for heritage-related structural support remains difficult due to the many different types linked to the architectural style. Murphy et al. (2013) focused on the creation of a library of parametric objects, following some common rules found in historical structural supports, while Antonopoulos and Antonopoulou (2017) opted for manual drawing by combining existing libraries and creating new parametric models. However, some common geometric rules can still be identified, e.g. the cross-section of a column is mostly circular.

Many approaches to point cloud segmentation automation are described in the literature. As pointed out by Nguyen and Le (2013), two general approaches exist in this regard: reliance on geometric axioms and mathematical functions (e.g. Macher et al., 2015) and the use of machine learning techniques (e.g. Bassier et al., 2017). While machine learning approaches are more robust against noise and occlusions, its main disadvantage is the time required to train the programme. In simpler cases, geometrical rules are often enough and may provide a faster result. In terms of point cloud classification, the process can be performed in a supervised (data-training), unsupervised, or

\* Corresponding author

interactive manner (Grilli et al., 2017). In this paper, the use of geometrical rules ensures a simple implementation and quicker results to segment and classify the point cloud. The use of these rules also enables a classification based on geometric characteristics of some typical architectural elements. A similar approach was also demonstrated in several other works concerning as-built BIM (Macher et al., 2017) and engineering applications (Riveiro et al., 2016). Another precedent to this study is a similar research by Luo and Wang (2008), which was nevertheless performed on modern columns and did not take into account point cloud classification.

### 3. METHODOLOGY

The proposed method employs several geometrical characteristics of typical historical buildings in performing the segmentation and classification process. The algorithm uses as a starting point the point cloud of a building. The first part of the developed approach was the identification and segmentation of the building's body and attic (Figure 1). The attic in this case is defined as the space between the roofs and ceilings of the uppermost story. This vertical segmentation is meant to facilitate further segmentation process and point cloud management. To this end, horizontal profiles of the object were extracted and their geometric properties were used to identify the attic from its main body. The area surface of the profiles was assessed, and a significant reduction of area surface was interpreted as the limit between the attic and the main body. In this manner, the algorithm was able to quickly and quite reliably determine these two parts of the building. In the absence of a tilted roof (such as the case with some modern buildings), the algorithm will simply determine that the building has no attic.

```
function [Struct] = atticsegment(ptCloud)
    Slices = slices(ptCloud);
    for i=1:numSlices
        ProjectedPts = project(ThisSlice, z_mean);
        Area = convhull(ProjectedPts);
        AreaList(i) = Area;
    end
    for i=1:size(AreaList)-1
        delta=abs(diff(AreaList(i);AreaList(i+1)));
        if (delta>tolerance)
            Zlimit = ThisSlice(z);
        end
    end
    Struct.Body = find(ptCloud(z)<=Zlimit);
    Struct.Attic = find(ptCloud(z)>Zlimit);
end
```

Figure 1. Pseudocode for the attic detection function.

```
function [Struct,SupportType] = supportdetect(Body)
    Slices = slices(ptCloud);
    SliceUsed = size(Slices)/2;
    Clusters = pcsegdist(SliceUsed);
    for i=1:numClusters
        ProjectedPts = project(ThisSlice, z_mean);
        Area = convhull(ProjectedPts);
        AreaList(i) = Area;
        if (Area<0.1*SliceArea && area>noise_thres)
            TypeList = "support";
        elseif area<noise_thres
            TypeList = "noise";
        else
            TypeList = "wall";
        end
    end
    for i=1:nbSupport
        circularity = (chull_perimeter.^2)/(4*pi*area);
        if (circularity<circularity_threshold)
            SupportType = "column";
        else
            SupportType = "pier";
        end
        CookieCutter=polybuffer(chull,bufferSize);
        index=isinterior(CookieCutter,Body);
        if (index=='TRUE')
            PtCloudIn=PtCloud(index,:);
        else
            PtCloudOut=PtCloud(index,:);
        end
    end
    Struct.PtCloud{i}=PtCloudIn;
end
```

Figure 2. Pseudocode for the support segmentation and classification function.

Further segmentation was performed to detect architectural elements from the building's body. Figure 2 shows a pseudocode of the developed approach to detect specifically structural supports. The `supportdetect` function consists of two parts. The first part concerns the detection of the structural supports, and successive segmentation into potential point cloud clusters. In this case, a 2D approach to a 3D problem was used to help with the process; a method similar to the one described by Macher et al. (2017). Consequently, a cross-section of the building's body (result of the previous `atticsegment` function) was extracted. From the cross-section, various "islands" represent different vertical elements of the building. In order to segment these elements into individual clusters, a region-growing segmentation based on Euclidean distance was performed. A preliminary filtering and immediate classification was then performed to distinguish between potential structural supports, walls, and point cloud noise. The filtering was done using the convex hull area criterion.

From the list of structural support clusters generated from this process, other geometrical rules were then used to determine if a structural support is a column or a pier. While there is no single agreed definition as to the distinction between a column and a pier, this study defined a column as a vertical support which mostly possesses a circular cross-section. On the other hand, a pier was defined as a support having a non-circular cross-section, mostly rectangular. This definition corresponds to the one taken from the UK-based Designing Buildings website ([https://www.designingbuildings.co.uk/wiki/Types\\_of\\_column](https://www.designingbuildings.co.uk/wiki/Types_of_column) accessed on 3 June 2019).

In order to distinguish between a circular and non-circular cross-section, again the convex hull is computed for each support's cross-section. For each structural support, the circularity parameter is computed from the convex hull parameters. This value follows the following formula, slightly modified from Takashimizu and Iiyoshi (2016):

$$circularity = \frac{Perimeter^2}{4\pi \cdot Area} \quad (1)$$

In this setup, a circularity value of a perfect circle is 1, while as the value increases the form of the object departs from a round form. While the circularity parameter is very easy to compute, it should be noted that it is not robust and is therefore prone to errors due to noises which may distort the form of the cross-section's convex hull.

In the final part of the code as shown by Figure 2, the segmentation and classification was extended back into the 3D space. Note that up to this point, only the building's cross-section's clusters of islands were segmented and classified. In order to do so, a similar approach to a previous research (Murtiyoso and Grussenmeyer, 2019) was used. In this approach, the convex hull of each support's cross-section is used as a "cookie-cutter" to obtain the 3D point cloud of all elevations corresponding to each island cluster. A buffering threshold was applied to the convex hull in order to give a tolerance to the process. A RANSAC plane fitting was then subsequently applied to remove the floor part of the segmented result. Finally, a last Euclidean distance-based region growing segmentation was performed in order to delete noises.

In this way, a form of automatic classification of the segmented point cloud clusters was conducted. The output of the general workflow consists of clusters of point clouds, segmented and classified into the attic and the main building body, which were then further classified into columns and piers.



Figure 3. The three cultural heritage sites used as test sites in this paper: (a) the Royal Pavilion of the Kasepuhan Palace in Cirebon, Indonesia; (b) the choir of the Catholic church of St-Pierre-le-jeune in Strasbourg, France; and (c) the Temple of Hera or colloquially the "Basilica" of the Paestum site in Italy.

#### 4. AVAILABLE DATA SETS

Case studies were conducted using three data sets (Figure 3). The first data set was taken at the Siti Inggil complex of the Kasepuhan Palace, Cirebon, Indonesia. This area dated to the 15<sup>th</sup> century and includes several historical pavilions within its 1,200 m<sup>2</sup> brick-walled perimeters. The site was digitised in 2018 using a combination of terrestrial laser scanner (TLS) and photogrammetry (both terrestrial and UAV), and was georeferenced to the Indonesian national projection system. The area was then segmented into individual building clusters which are automatically annotated with semantic information using pre-existing GIS data (Murtiyoso and Grussenmeyer, 2019). The Royal Pavilion located in these premises was thereafter used as case study for the developed method.

Another test was also conducted using a second data set obtained from the 19<sup>th</sup> century St-Pierre-le-jeune Catholic church located in Strasbourg, France. The data was acquired in 2016 and 2017 using UAV photogrammetry for the exterior and TLS scanning for the interior (Murtiyoso and Grussenmeyer, 2018). The resulting point cloud was then georeferenced to the French national projection system to facilitate future documentation of the site. The church was constructed in the Neo-Romanesque architectural style and is therefore characterised by columns and semi-circular arches. The church is part of the Neustadt UNESCO World Heritage site of the city of Strasbourg. For this study, the choir of the church which is characterised by twin pillars was used. The twin pillars provide a further challenge for the algorithm as they are conjoined. Furthermore, the posterior pillars are attached to iron fences.

The third data set was obtained from a laser scanning mission at the Paestum archaeological site in southern Italy. The data was acquired by the 3DOM-FBK Trento team (Fiorillo et al., 2013) and has been shared to the authors to be used as a case study in this paper. Specifically, the point cloud data of the "Basilica" or the Temple of Hera was used in this paper. The Basilica is the ruins of a Greek-style temple with Doric orders; Paestum itself being a Greek colony in the 7<sup>th</sup> century BC. The Paestum site is also a UNESCO World Heritage site.

The three available data sets provide very different styles of architecture. While Paestum's Doric columns provide a prime example of ancient Greek architecture, it is vastly different to the 15<sup>th</sup> century Javanese architecture of the Kasepuhan site. In the same manner, the 19<sup>th</sup> century St-Pierre church provides an example of Neo-Romanesque columns. The differences between the three available data set's architectural styles provide an interesting opportunity to assess the developed algorithm.

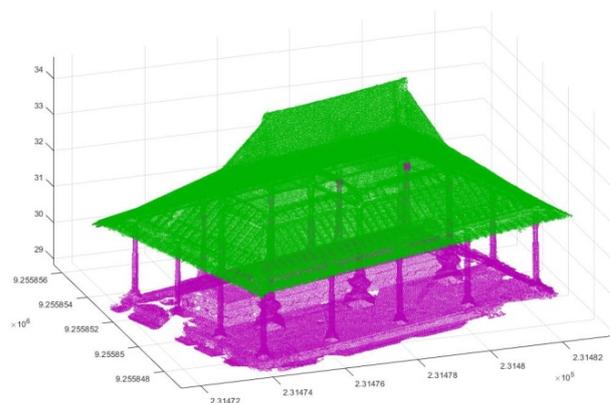


Figure 4. Automatically segmented building attic and body for the Kasepuhan Royal Pavilion dataset; the attic is here shown in green and the body in purple.

#### 5. RESULTS AND DISCUSSIONS

The first function as described in Figure 1 was applied to the Kasepuhan data set in order to separate the building's body from its attic, while the second function (Figure 2) was applied to all three data sets to detect, segment, and classify their respective structural supports.

The first result concerning the attic segmentation algorithm can be seen in Figure 4. The algorithm detects an abrupt change in overall cross-section convex hull area and determines automatically the upper part as the attic and the lower part as the building's body. In this regard, the programme managed to detect the attic automatically and quickly (about 5 seconds). This part of the algorithm is aimed as a sort of pre-processing for the point clouds of buildings which possess an attic, as a precursor to the structural support detection part of the developed algorithm. This preliminary processing will enable a fully automatic workflow which begins with the point cloud of the entire building as input, therefore minimising as much as possible human intervention during the process.

In the developed workflow, this step is followed immediately by the structural support detection as expressed in the pseudocode of Figure 2. For the Kasepuhan dataset, the building body part which was previously segmented was used as input, while for the St-Pierre and Paestum data sets the original point clouds were directly used as inputs. As has been described in section 3, there are two main parts of the algorithm which are performed simultaneously; namely the segmentation and the classification.

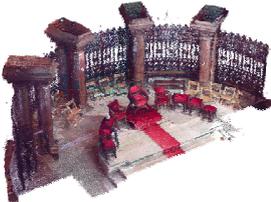
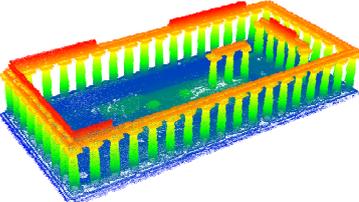
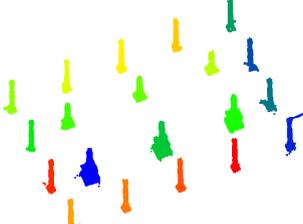
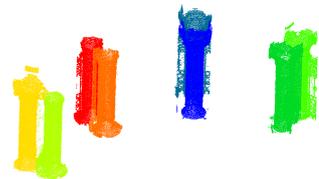
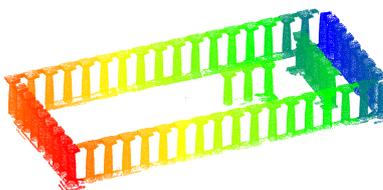
<i>Original point cloud</i>			
<i>Data set</i>	Kasepuhan	St-Pierre	Paestum
<i>Arch. style</i>	Javanese	Neo-Romanesque	Greek
<i>Object type</i>	Pavilion	Church choir	Temple ruins
<i>Point count</i>	155,115	1,852,162	1,181,025
<i>Detected</i>	20 supports	8 supports	58 supports
<i>Ground Truth</i>	20 supports	4 twin pillars	58 supports
<i>Result</i>			

Table 1. Comparison of the results of the segmentation part of the algorithm on the three available data sets. The resulting segmented clusters are displayed in random colours denoting one cluster for each colour.

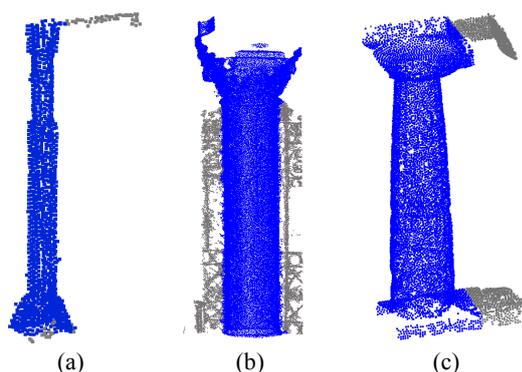


Figure 5. Sample support clusters of the resulting segmented point cloud from each data set: (a) Kasepuhan, (b) St-Pierre, and (c) Paestum. Blue colour denotes true positive points, while grey ones are false positive points.

Although both steps were integrally implemented in the same function, the following discussion will be divided into the segmentation and classification sections to help understand the consecutive steps of the workflow. All analysis was conducted using an Intel<sup>(R)</sup> Xeon<sup>(R)</sup> E5645 2.4 GHz CPU.

### 5.1 Point cloud detection and segmentation

As regards to the detection and segmentation of the input point cloud into potential building structural supports, a summary of the results from the three case studies is showcased in Table 1. As has been previously described, the three available data sets present a unique set of test data with three very different styles of architecture. That being said, the three shares a common characteristic in that columns within the three architectural types have circular cross-sections.

The Kasepuhan data set is the smallest of the three, consisting of only a little over 155k points, while Paestum consists of

1.1M points. The St-Pierre data set consists of 1.8M points for the choir part which is used in this study.

In total, the algorithm managed to detect 20 supports for Kasepuhan, 8 supports for St-Pierre, and 58 supports for Paestum. This corresponds well with the ground truth data, where identical numbers of structural supports are found in each data set. While the detection of the supports in Kasepuhan and Paestum are quite straight-forward due to the fact that in both data sets the supports are fairly apart from each other, the case of St-Pierre is more complex. In the St-Pierre choir data, the eight pillars are actually four pairs of twin pillars, each pair consisting of two columns conjoined at the plinth and capital levels. Furthermore, the posterior columns of each pair are attached to an iron fence which links the four pairs and forms a barrier between the choir and the ambulatory located behind it. Difficulties arose when applying the algorithm by default, because the function arbitrarily takes the middle altitude cross-section of the point cloud to perform the detection part. In this regard, the iron fence hinders a proper detection of the posterior columns as stand-alone supports. A tweak was necessary to be applied to the algorithm in order to properly detect each support, namely by setting the cross-section profile to be used in the detection part to the one just beneath the capitals where the iron fence ends. The buffering of the convex hull cookie cutter polygon also needed to be adjusted as to take into account the short space between two columns in a pair.

Figure 5 shows a sample of some structural supports detected by the algorithm on the three data sets, while showcasing some of the problems encountered with each particular case. For the Kasepuhan data set, some erroneous points were segmented together with a pier, effectively presenting a case of overclassification or false positive points. In this case, the overclassified points are those belonging to a sign post which was attached to the pier. This error typically manifests due to the use of the cookie cutter and Euclidean distance-based region

growing approaches. Although the RANSAC-based plane fitting filter managed to exclude the floor part of the segmented cluster, the use of distance-based region growing did not manage to exclude that of the sign post. This is because the sign post is attached to the pier, effectively telling the algorithm that these points belong to the same cluster.

A similar problem can be observed with the St-Pierre data set. As has been previously explained, the posterior columns are attached to an iron fence. Here the same problem with the combination of the cookie-cutter and distance-based region growing manifested itself. Indeed, the same reasoning can be followed to explain why a small part of the iron fence on each side of the support was included in the cluster. A similar, albeit more curious, problem can be seen in the Paestum data set. Here the same argument regarding the disadvantages of the cookie-cutter and distance-based region growing segmentation may explain the existence of the false positive points. However, the Paestum data set displays a systematic tendency to this, in that the same case happens to not only one support, but indeed many. This may be explained by the iterative nature of the algorithm, in which the previous iteration includes the left (in the relative directions of Figure 5) end of the cluster, while the current iteration takes what is rest. This happens consecutively, therefore creating this impression of systematic error. A possible solution would be to refine the algorithm's parameter, for example by fine tuning the buffering radius of the support's cross section convex hull.

Numerically speaking, some statistics related to this detection and segmentation part of the function can be seen in Table 2. In this table, the overclassified column describes the number of points considered as false positives, while the unclassified column denotes the true negative points. False negative points

are not showed since the values are negligible due to the cookie-cutter approach of taking all points of all elevations of a particular polygon shape. St-Pierre displays the highest percentage of unclassified points, amounting to 28.09%. This is easily explained by the condition surrounding the object of interest, namely the church choir. Ambiguities were evident due to the existence of the iron fence between the choir and the ambulatory, while the twin nature of the columns also generated errors. Furthermore, the input point cloud was not pre-processed or cleaned; several artefacts which can be considered as noise were present in the scene e.g. folded chairs, both open and folded (some are stored between the columns of a twin pair, thereby presenting significant uncleaned noise).

This problem can also be observed from the cloud-to-cloud analysis also presented in Table 2. The cloud-to-cloud analysis computed a Euclidean distance for each point to its nearest reference point, and was computed using the software CloudCompare (<https://www.danielgm.net/cc/> accessed on 4 June 2019). The standard deviation value for the St-Pierre data is quite high (40.6 mm).

As regards to the results for the Paestum and Kasepuhan data sets, both showed good results in terms of unclassified percentage, amounting to around 10-15%. The Kasepuhan data set also fared well in terms of the cloud-to-cloud analysis, although this may be a little biased due to the fact that Kasepuhan presents the lowest number of points and the supports are located fairly apart from each other, as well as the fact that it is the smallest object of the three in terms of scale. For Paestum, the higher order of values, both for the mean and standard deviation values reflected the larger scale of the object compared to the other two data sets. Indeed, Paestum's dimension is about 5 times that of Kasepuhan and St-Pierre.

Data set	Point Count		Mis-classified			C2C Analysis	
	Manual	Auto	Overclassified	Unclassified	% Unclassified	Mean (mm)	$\sigma$ (mm)
Kasepuhan	37 185	32 230	397	5 352	14.39	1.3	15.8
Paestum	616 766	594 030	55 434	78 170	12.67	24.2	89.6
St-Pierre	595 964	476 250	47 713	167 427	28.09	11.0	40.6
				<b>Mean</b>	<b>18.39</b>		
				<b>Median</b>	<b>14.39</b>		

Table 2. Segmentation statistics for the three data sets, also showing a cloud-to-cloud analysis which used the manually segmented clusters as reference.

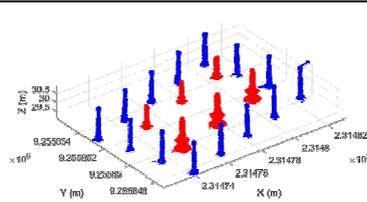
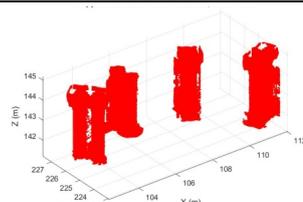
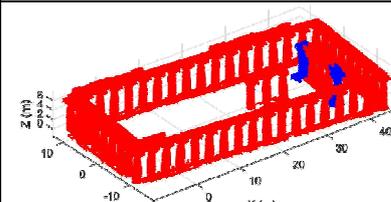
Data set	Kasepuhan	St-Pierre	Paestum
<b>Classified</b>	6 pillars, 14 piers	8 pillars	56 pillars, 2 piers
<b>Ground truth</b>	6 pillars, 14 piers	4 twin pillars	56 pillars, 2 piers
<b>Processing time</b>	38.28 seconds	83.56 seconds	341.16 seconds
<b>Result</b>			

Table 3. Comparison of the results of the classification part of the algorithm on the three data sets. Red denotes the column class while blue the pier class. Note that processing time is for both the segmentation and classification steps, which were integrated into one function.

However, Paestum's high standard deviation once again reflected the problem with the overclassification and its systematic nature, as has been described in the previous paragraph. Overall, the segmentation step presents an average success rate of 81.61% and median success rate of 85.61% for three tested data sets which are available.

## 5.2 Classification and processing time

The classification part of the algorithm is performed integrally with the detection and segmentation part within the function of Figure 2. This section will discuss the results strictly from the perspective of the classification part, in order to give a more systematic description in the paper.

Table 3 summarises the results obtained for the classification step. The algorithm utilises the circularity value for each support cluster's cross-section convex hull to determine if a cluster is attributed the class of column or pier; the column class being characterised by a more circular form denoted by a circularity value of around 1. An empiric value of 1.12, computed from the average circularity of ground truth columns, was used as the threshold between the "column" and "pier" classes. In this regard, results from all three data sets show promising results as the algorithm managed to correctly identify the form of all structural supports.

In the case of the Paestum data set, the ground truth data gave a total of 58 supports, comprising of 56 circular columns and 2 rectangular ones. Within the pre-defined context identified at section 3, these two rectangular columns can be considered as piers. The algorithm managed to detect the same number of support types, and correctly determined which support belonged to which class. The whole processing of the Paestum data, comprising the detection, segmentation, and classification, takes a total of 341.16 seconds.

For the Kasepuhan data set, the algorithm managed to detect the 6 columns located at the inner part of the pavilion, of which three are located on an elevated dais. The surrounding 14 wooden piers were also correctly identified. The algorithm took 38.28 seconds to generate this result.

The St-Pierre data also showed promising results, as the programme managed to identify the eight supports as columns, their twin nature notwithstanding. In this case, the processing time amounts to 83.56 seconds. Although the algorithm managed to perform the classification task well enough, it should be noted that fine tuning of some important parameters must be performed to address specific cases.

As may be observed from the processing time of the three data sets, it is not directly linked to the point count of their respective input data as much as to the number of supports detected and the density of the point cloud. The Paestum data took more time to process compared to the denser St-Pierre data due to the amount of supports detected (58). However, density also plays a role as can be seen in the comparison between St-Pierre and Kasepuhan. This can be explained by the fact that the algorithm works by relying on cross-sections; thus accelerating the classification part. The segmentation is therefore the part that takes more time depending on point cloud density and the number of identified structural supports. However, the overall processing time is still faster by at least a factor of 2 when roughly compared to the time it takes to perform the same task manually, without taking into account the time required to identify and classify each cluster into the appropriate classes.

## 6. CONCLUSIONS AND FURTHER WORK

This paper has attempted to describe an algorithm which enables the automation in heritage point cloud segmentation and classification using simple geometrical rules, such as convex hull areas and circularity values. The main focus of the algorithm is the detection of architectural elements; in this preliminary phase it is the structural support of the building. It has also showcased the results generated by said algorithm on three test sites with very different architectural styles with promising results. The method was implemented in the Matlab<sup>®</sup> language, but the main algorithm is open source and other implementations using other languages such as C++ and Python is perfectly possible.

Tests on the three available data sets showed that in terms of segmentation, the algorithm managed to correctly identify potential structural supports. The segmentation was then performed using the cookie-cutter approach, followed by additional filtering using RANSAC-based plane fitting segmentation to exclude floors and a Euclidean distance-based region growing segmentation to filter out the noises within the cluster. While this approach provided results in a fairly short processing time, it has its own downfalls as evidenced in section 5. Mainly, the approach is dependent on the particular condition of each case, with a deviation to the ideal condition (e.g. existence of attached sign posts or iron fence) resulting in overclassification. That being said, the algorithm managed to be quite reliable in some specific cases, such as the segmentation of the twin pillars in the St-Pierre data set. As has been stated previously, judging from the three available data sets, the algorithm attained mean and median success rates of 81.61% and 85.61% respectively.

In terms of the classification results, the use of the circularity value as a geometric parameter proved to be effective in differentiating between the column and pier classes. The developed algorithm is also fairly fast and contrary to machine learning-based techniques, it does not require training data; something which may be complex to implement due to the innumerable types of structural support architectural styles in the heritage domain. However, the algorithm which is based on geometrical rules is quite sensitive towards noise as has been previously hypothesised and shown in this paper.

Future studies will involve the refinement of the algorithm as well as its extension for other architectural elements (e.g. wooden beams, plinths, capitals, etc.). Later research will also address the generation of geometrical primitives from these segmented point cloud clusters. The final objective is to be able to generate the 3D model primitives for the segmented architectural elements. This will hopefully facilitate and save time in further processes down the 3D modelling workflow pipeline, such as the creation of 3D GIS and HBIM (Heritage Building Information Models). Furthermore, the (partial) automation of the classification process also aids in the semantic enrichment of these end products.

## ACKNOWLEDGEMENT

This research benefits from the Indonesian Endowment Fund for Education (LPDP), Republic of Indonesia. The authors also wish to thank Fabio Remondino of 3DOM-FBK Trento for agreeing to share the Paestum data set used in this paper. The codes developed for this paper are written in Matlab<sup>®</sup> but are open source and may be downloaded from the following link: [https://github.com/murtiad/M\\_HERACLES](https://github.com/murtiad/M_HERACLES).

## REFERENCES

- Antonopoulos, A., Antonopoulou, S., 2017. 3D survey and BIM-ready modelling of a Greek Orthodox Church in Athens. In: *IMEKO International Conference on Metrology for Archaeology and Cultural Heritage*, Lecce, Italy, October 23–25, 2017 3D. Lecce, Italy.
- Barsanti, S.G., Guidi, G., Luca, L. De, 2017. Segmentation of 3D models for cultural heritage structural analysis - some critical issues. In: *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*. Vol. IV-2/W2, pp. 115–122. doi:10.5194/isprs-annals-IV-2-W2-115-2017.
- Bassier, M., Vergauwen, M., Van Genechten, B., 2017. Automated Classification of Heritage Buildings for As-Built BIM using Machine Learning Techniques. In: *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Vol. IV-2/W2, 25–30. doi:10.5194/isprs-annals-IV-2-W2-25-2017.
- Campanaro, D.M., Landeschi, G., Dell'Unto, N., Leander Touati, A.M., 2016. 3D GIS for cultural heritage restoration: A "white box" workflow. *Journal of Cultural Heritage* 18, 321–332. doi:10.1016/j.culher.2015.09.006
- Fiorillo, F., Jiménez Fernández-Palacios, B., Remondino, F., Barba, S., 2013. 3D Surveying and modelling of the Archaeological Area of Paestum, Italy. *Virtual Archaeology Review* 4, 55–60.
- Grilli, E., Menna, F., Remondino, F., 2017. A Review of Point Clouds Segmentation and Classification Algorithms. In: *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*. Vol. XLII-2/W3, pp. 339–344. doi:10.5194/isprs-archives-XLII-2-W3-339-2017.
- Luo, D., Wang, Y., 2008. Rapid extracting pillars by slicing point clouds. In: *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*. Vol. XXXVII, pp. 215–218.
- Macher, H., Landes, T., Grussenmeyer, P., 2017. From Point Clouds to Building Information Models: 3D Semi-Automatic Reconstruction of Indoors of Existing Buildings. *Applied Sciences* 7, 1–30. doi:10.3390/app7101030.
- Macher, H., Landes, T., Grussenmeyer, P., 2015. Point clouds segmentation as base for as-built BIM creation. In: *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*. Vol. II-5/W3, pp. 191–197. doi:10.5194/isprsannals-II-5-W3-191-2015.
- Murphy, M., McGovern, E., Pavia, S., 2013. Historic Building Information Modelling - Adding intelligence to laser and image based surveys of European classical architecture. *ISPRS Journal of Photogrammetry and Remote Sensing* 76, 89–102. doi:10.1016/j.isprsjprs.2012.11.006.
- Murtiyoso, A., Grussenmeyer, P., 2019. Point cloud segmentation and semantic annotation aided by GIS data for heritage complexes. In: *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*. Vol. XLII-2/W9, pp. 523–528. doi:10.5194/isprs-archives-XLII-2-W9-523-2019
- Murtiyoso, A., Grussenmeyer, P., 2018. Comparison and assessment of 3D registration and georeferencing approaches of point clouds in the case of exterior and interior heritage building recording. In: *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*. Vol. XLII-2, pp. 745–751. doi: 10.5194/isprs-archives-XLII-2-745-2018.
- Murtiyoso, A., Grussenmeyer, P., Suwardhi, D., Awalludin, R., 2018. Multi-Scale and Multi-Sensor 3D Documentation of Heritage Complexes in Urban Areas. *ISPRS International Journal of Geo-Information* 7, 483. doi:10.3390/ijgi7120483.
- Nguyen, A., Le, B., 2013. 3D Point Cloud Segmentation : A survey. In: *2013 6th IEEE Conference on Robotics, Automation and Mechatronics (RAM)*. IEEE, pp. 225–230.
- Riveiro, B., Dejong, M.J., Conde, B., 2016. Automated processing of large point clouds for structural health monitoring of masonry arch bridges. *Automation in Construction* 72, 258–268. doi:10.1016/j.autcon.2016.02.009.
- Takashimizu, Y., Iiyoshi, M., 2016. New parameter of roundness R: circularity corrected by aspect ratio. *Progress in Earth and Planetary Science* 3(2). doi:10.1186/s40645-015-0078-x.
- Wang, C., Cho, Y.K., Kim, C., 2015. Automatic BIM component extraction from point clouds of existing buildings for sustainability applications. *Automation in Construction* 56, 1–13. doi: 10.1016/j.autcon.2015.04.001.