2021 European Conference on Computing in Construction
Online eConference
July 26-28, 2021

EFFICIENT VERTICAL OBJECT DETECTION IN LARGE HIGH-QUALITY POINT CLOUDS OF CONSTRUCTION SITES

Miguel A. Vega[1], Alexander Braun[1], Heiko Bauer[2], Florian Noichl[1], and André Borrmann[1]
[1]Chair of Computational Modeling and Simulation, Technical University of Munich, Germany
[2]FARO EUROPE GmbH & Co.KG., Korntal-Münchingen, Germany

ABSTRACT

Even when adherence to project schedule is the most critical performance metric among project owners, still 53 % of typical construction projects exhibit schedule delays. To contribute to more efficient construction progress monitoring, this research proposes a method to detect the most common temporary object classes in large-scale laser scanner point clouds of construction sites. The proposed workflow includes a combination of several techniques: image processing over vertical projections, finding patterns in 3D detected contours, and performing checks over vertical cross-sections. A deep learning algorithm was leveraged to classify these cross-sections for the purpose of formwork detection. After applying the method on three real-world point clouds and testing with three object categories (cranes, scaffolds, and formwork), the results reveal that the process achieves average rates above 88 % for precision and recall and outstanding computational performance (1 s to process 10^5 points). These metrics demonstrate the method's capability to support the automatic segmentation of point clouds of construction sites.

INTRODUCTION

Nowadays, inefficiencies, such as cost and time overruns, occur regularly within the construction industry. According to Mace & Jones (2016) 53% and 66% of typical construction projects record schedule delays and cost overruns, respectively. Moreover, KPMG revealed in its Global Construction Survey that adherence to the project schedule is not only the most essential performance measure in construction industry contracts but also the central issue in the execution of projects (Armstrong & Gilge 2017).

One of the root causes of these issues is that the monitoring process is still mostly performed manually in the construction industry. This practice is expensive, labor-intensive, and not comprehensive (Lin & Golparvar-Fard 2020).

Many approaches have emerged to address this problem. Current research proposes to compare a 4D building information model with a point cloud of a construction site, allowing to track progress (Braun et al. 2020, Bosché 2012). This tracking is possible because in a BIM model, all construction elements, besides having 3D geometry, are linked with process information, enabling them to report the planned state of construction at any given time. However, one of the preeminent challenges with this approach is the presence of temporary construction elements in the point cloud. These temporary elements are usually not present in the building information model, and even worse, may occlude large portions of the permanent structures in the point cloud. Therefore, the presence of these elements makes a reliable comparison with the 3D geometry of the model more challenging. Some of the most common temporary elements are: scaffolds, formwork, cranes, reinforcement, and machinery.

To overcome this challenge, the goal of this research is to detect cranes, scaffolds, and formwork in laser-scanned point clouds of construction sites. Besides the fact that these objects are prevalent on a construction site, detecting them is useful for the following reasons:

Since the number of cranes and their height varies depending on the construction phase, this information gives a rough idea about the state of the construction progress. Moreover, knowing the exact position of cranes would allow the verification of compliance with safety regulations, like the distance from the crane to the building or to other cranes.

Detecting scaffolding components is useful to track the construction site's progress and perform precise safety regulation checks regarding the minimum requirements that scaffold should fulfill, such as the presence of toe-boards and guard-rails in the right position. This checks could be done by implementing corroborated methods such as those introduced by Wang (2019). This last step is crucial because, as Wang identified, falling from scaffolds is one of the leading causes of fatal accidents on construction sites.

Identifying the location of the formwork gives crucial information about the exact current state of construction progress. A placed formwork does not exclusively represent a building element that is currently under construction, it also indirectly gives vital information about other completed tasks on the construction site. For example, the previous construction of a concrete slab on which the formwork is placed, or the placed rebars inside two wall formworks. Af-

ter the detection of formwork elements, the quality of the construction can also be evaluated. Beyond the correct position of the formwork itself (relative to the corresponding wall), the presence of openings and special elements can automatically be verified.

RELATED WORK

There has been a lot of improvement in automatic construction progress monitoring in the past decade. While some researchers based their methods on photogrammetric point clouds (Golparvar-Fard et al. 2011, 2015, Braun et al. 2020, Braun & Borrmann 2019, Braun et al. 2016), others use laser scanner point clouds (Bosché & Haas 2008, Bosché 2012, Kim et al. 2013, Bosché et al. 2015, Han et al. 2018). The existence of a 3D/4D building information model is implicit for these Scan-vs-BIM approaches. With a 4D model and a point cloud, an as-built vs. as-planned comparison is possible, allowing the automatic monitoring of the progress (Braun et al. 2020). However, the presence of temporary building elements hinders automatic progress tracking. Besides that, these temporary elements should be detectable, even without having a BIM model.

Turkan et al. (2014) made initial proposals to track temporary elements. However, their method is based on a Scan-vs-BIM approach that requires a BIM model. Only using point clouds, most of the related work focuses on the reconstruction of a building information model from scans (Maalek et al. 2019, Nikoohemat et al. 2020, Armeni et al. 2016). However, there is only limited research on the detection of cranes, scaffold or formwork using point clouds.

While deep learning approaches for point cloud segmentation seem to be very promising, they still have three critical shortcomings. One limitation is the maximum number of points that an algorithm can process simultaneously (e.g., 1m×1m with 4096 points) (Guo et al. 2019), making the method not very suitable to detect large objects in large-scale point clouds directly. A second drawback is the non-rotational invariant constraint of some techniques, like the one implemented by Zeng et al. (2020), which restricts the practice to only find items with known XYZ-orientation. A third and final drawback is that extracting the deep point features is usually very time-consuming and memory-costly (Zeng et al. 2020, Landrieu & Simonovsky 2018, Hu et al. 2020). Besides that, the successful implementation of a deep learning algorithm requires a large database of real labeled data to train the algorithms. Such a database is at this moment not available for temporary objects on construction sites.

Other state-of-the-art methods, that do not require labeled data, like the ones proposed by Xu et al. (2018) or Wang (2019), take advantage of the verticality of the objects to detect scaffold elements, as well as in-depth knowledge of the underlying geometry of the objects, like dimensions of the uprights or possible bay width distances. While still having some drawbacks, these methods showed promising results for the specific case of scaffold detection in point clouds of construction sites.

Further work can be done to detect scaffolds more efficiently, as well as to recognise additional objects, such as cranes and formwork elements. These enhancements are specifically the goal of this research.

GEOMETRY OF TARGET OBJECTS

This section summarizes necessary specifications about the target objects' usual geometry, which is crucial to detect these objects in a point cloud. Additional justification for the selection of certain types of target objects is also given.

Cranes

Some of the most common types of cranes in the construction industry are the crawler crane, self-erecting crane, telescopic crane, and tower crane. This research focuses mainly on tower cranes because they are the most commonly used in the construction of tall buildings (Böttcher & Neuenhagen 1997, p. 58).

The main components of a tower crane are the base, mast, slewing unit, operating cabin, jib, and counter-jib. The mast is generally made of individual steel trussed sections that are connected. The number of sections will determine the overall height of the crane.

While a mast section is always squared, its width can vary between 1.2 m to 2.5 m depending on the crane's type (see Figure 1). To allow the detection of self-erecting cranes that usually have a smaller mast width than tower cranes, we use a minimum mast width of 1 m instead of 1.2 m for crane detection.

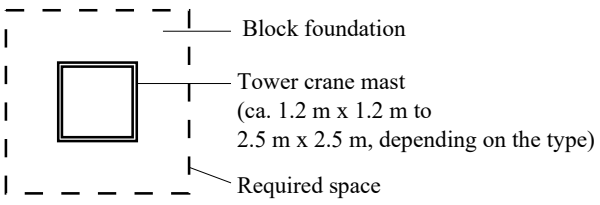


Figure 1: Top view of a tower crane mast with dimensions (Schach & Otto 2017, p. 28).

Scaffold

Opposite to sections of a tower cranes mast, scaffold elements consist of different smaller pieces that are usually manually assembled on the construction site. These are mainly: uprights, guard-rails, toeboards, and work platforms. Additionally, there are special sections of the scaffold system with diagonal braces, stairs, or additional accessories that enable the scaffold to adapt to different needs, such as bridges or extensions, to make the scaffold wider. This research will focus on detecting faced scaffold elements.

Depending on the manufacturer, a scaffold's exact geometry can vary, but standardized norms establish some minimum dimensions. Following DIN EN 12 811-1, the minimum scaffold bay width is 0.6 m, and while there could be a scaffold bay width of more than 2.4 m, in this research, only scaffold with a maximum width of 1.2 m will be considered. This consideration is based on the fact that cost-effective scaffold systems are mainly made in the width classes W06 and W09 (Schach & Otto 2017, p. 240), which have a width between the selected range (0.6 m to 1.20 m) in accordance with Table 1 of DIN EN 12 811-1. Similarly, the scaffold bay length could vary between 1.5 m to 3 m in line with DIN 4420-4. Figure 2 presents the main components of a scaffold, together with its standardized minimum and maximum dimensions.

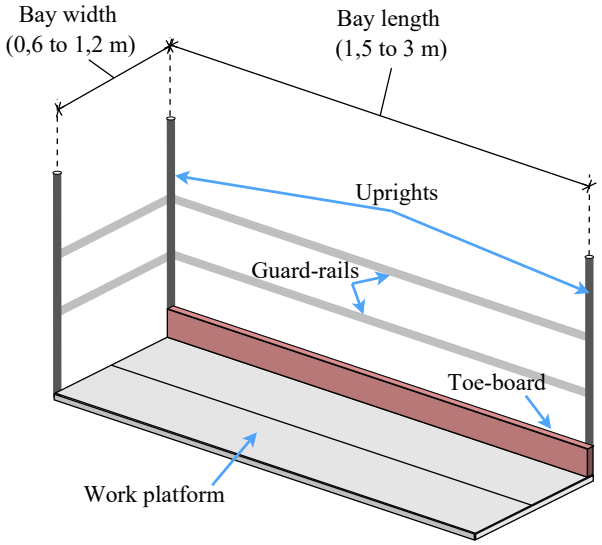


Figure 2: Main scaffold components and dimension ranges

Formwork

Among the many types of formwork, the most common are wall, column, and slab formwork. Similar to scaffold elements, there could be specialized types of formwork, and they could also have additional accessories, for example, working platforms. However, this research will concentrate on standard wall formwork.

Whereas the exact geometry of a formwork element depends on the manufacturer, the basic idea of vertical studs and horizontal walings in front of an interior wall panel always remains constant. The orthogonality between studs and walings (see red elements in Figure 8a) together with the wall panel will be exploited to detect formwork elements.

METHODOLOGY

Overview

The workflow of this research is illustrated in Figure 3. The first step is a preprocessing of the raw laser-scanned point cloud, in which down-sampling is applied, followed by a rotation of the point cloud that will align it to the building axes.

The second step is the detection of cranes, in which Regions of Interest (ROIs) that may contain cranes are separated using image processing techniques over a vertical projection of the point cloud. Later, an algorithm will search a pattern characteristic of a tower crane in detected 3D vertical lines, which will reveal the cranes' possible positions. Then, the final location of cranes is determined by applying checks over vertical cross-section projections. Subsequently, scaffold elements are detected with a very similar procedure as with cranes (see Scaffold detection).

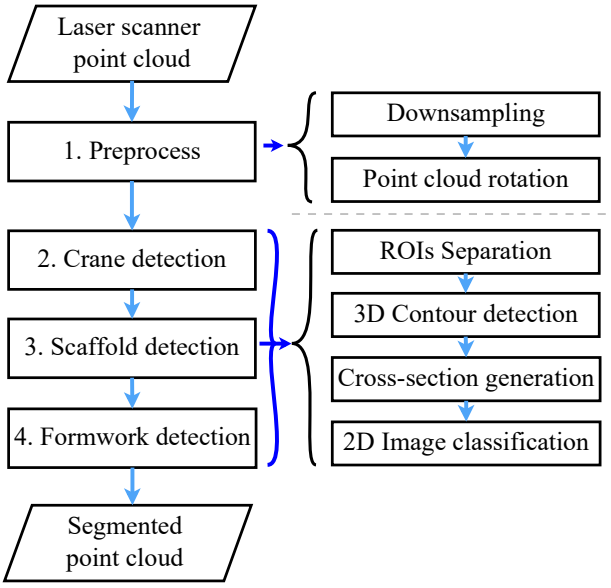


Figure 3: Workflow overview.

As the last step, formwork elements are detected. Here again, the ROIs that might contain formwork elements are prefiltered, vertical cross-sections projections are generated, and a Machine Learning (ML) algorithm is leveraged to determine the presence of formwork elements (see Formwork detection).

Preprocessing

Downsampling

Filtering or downsampling the point cloud is vital for two reasons: First, it will allow the method to take advantage of the fact that the point cloud has a relatively uniform density by assuring a certain average data spacing; and second, it is the first step that will reduce the computational cost as the number of points is reduced substantially, always when the original resolution is higher than the used leaf size.

To fast sub-sample the point cloud, it is first organized into an octree with a resolution of 5 m. The creation of this octree allows the implementation of the PCL voxel grid method with a leaf size (VG_{ls}) of 5 mm in every leaf voxel of the octree.

Table 1: Parameter Summary

Parameter	Description	Wall	Crane	Scaffold	Formwork
h_{min} [m]	Minimum object height	1.2	0.7	0.2	0.075
S	Structural element with its size	R10x10	R10x10	E5x5	R10x10
D_i	Number of dilation iterations	5	3	6	6
A_{min} [m^2]	Minimum blob area	1.5	0.0075	0.002	0.25
A_{max} [m^2]	Maximum blob area	MAX	0.3	0.075	MAX
l_{min} [m]	Minimum merged lines length	N/A	1.5	0.4	N/A

Point cloud rotation

This step aims to rotate the point cloud so that it is aligned with the building's principal axes. This alignment will allow taking advantage of the rectangular grid that usually the building's floor plans follow (also known as Manhattan World).

This rotation is done in two main steps:
1) Walls ROIs Separation with image processing in a vertical projection, and 2) determination of the final angle of rotation with 2D detected lines.

Before applying this method, the point cloud has to be divided into different building floors, for this, the user has to enter manually the minimum and maximum Z values of the corresponding floor to be analyzed. This separation is a requirement for the process to be able to filter objects by their minimum height. Figure 4 illustrates a building's first floor.

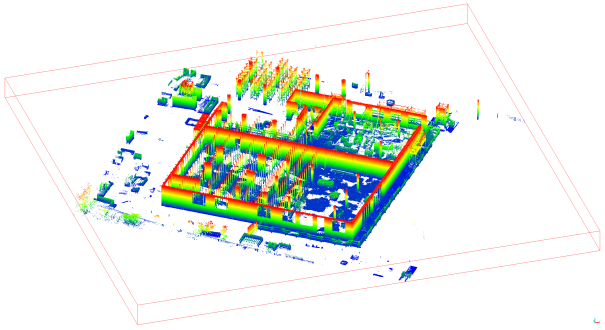


Figure 4: Clipped first floor of the Test dataset Nr. 2.

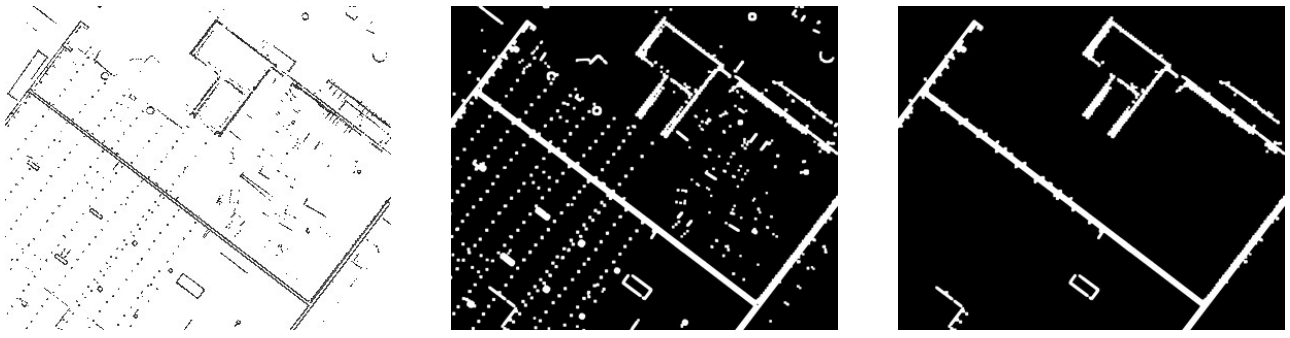
Walls ROIs separation

As usually all large load-bearing walls are aligned with the building's structural axes, they are first separated from the rest of the point cloud.

As the point cloud was already downsampled, it is known that the minimum distance between two points is 5 mm. Therefore, the point cloud projection in a grayscale accumulation image, which stores the number of points projected on each pixel, allows the differentiation of the objects by their minimum height. For example, considering the presence of occlusions in the point cloud and the possible presence of formwork covering the walls, it is assumed that vertical walls may have at least 1.2 m (h_{min}) of height, which is around half of the height of an average wall.

Subsequently, ten iterations of a morphological dilation with a structural element (S) with a rectangular shape of size 10 x 10 (S_{R10}), will join small blobs that are close to each other and may conform more oversized objects (as shown in Figure 5b). Later the blobs can be separated by its number of white pixels. Since the grid side length used to create the vertical projection has a value of 5 mm, one square meter in the point cloud is then represented by a region containing 40.000 pixels in the image.

For example, for walls a minimum area of $A_{min} = 1.5\,\text{m}^2$ was considered to be more appropriate (see in Table 1 all parameters). Figure 5c shows the final wall ROIs, which are the result of filtering the blobs by size in a dilated vertical projection after passing a height threshold.



(a) (b) (c)

Figure 5: Wall ROIs in a vertical projection: (a) original vertical projection (for better visibility, the inverted binary version is shown here); (b) binary image after threshold and dilation, notice here that the two surfaces of the walls now form one single large blob; (c) final Wall ROIs ($W_{regions}$) after separation by blob size. Test dataset: Nr. 2.

(a) (b) (c) (d)

Figure 6: Detection of possible crane lines: (a) original point cloud with a red ellipse indicating the location of the crane; (b) Crane ROIs in (a), notice the presence of other thin and tall objects in addition to the crane; (c) detected 3D contours in (d) ;(d) filtered merged vertical lines from (c). Test dataset: Nr. 1.

Angle of rotation with 2D lines

Once the ROIs of large walls are isolated in W_{regions}, this image is used as a mask to filter the original vertical projection. Using the probabilistic Hough transform algorithm (Mukhopadhyay & Chaudhuri 2015), with an angular resolution of $\pi/(180 \cdot 100)$, 2D lines are fitted in this filtered vertical projection. Finally, the angle of rotation is determined using the k-means algorithm (Ahmed et al. 2020) over a 1D histogram of the slopes of the previously detected 2D lines.

Once the point cloud is downsampled and aligned with the axes of the coordinate system, the next step is the detection of the target objects.

Crane detection

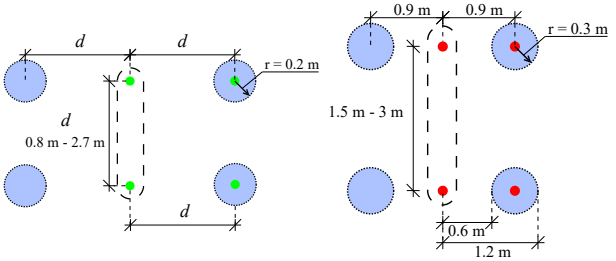
The detection of cranes starts with a similar step as the one used to separate the wall ROIs but with different parameters of minimum height, dilation, and blob size (see Table 1). This step will efficiently filter out points that are more likely to belong to a crane from the rest of the point cloud. In Figure 6b, all the elements that pass the filter are shown.

In the next step, 3D contours are efficiently detected with the algorithm provided by Lu et al. (2019). Figure 6c illustrates the 3D line detection results in a point cloud with the crane ROIs.

Subsequently, the vertical lines are projected in the XY-plane and then merged in single lines if there is a maximum distance of 20 cm between them, considering that the detected lines could be in any of the four borders of the steel profiles, which have a width of around 12.5 cm (Yasmin 2019). These merged lines are then also filtered by their length (see l_{min} in Table 1), resulting in the lines presented in Figure 6d.

Now that the vertical lines are detected, the pattern that characterizes a crane will be searched in these vertical lines. As explained before, the mast of tower cranes always has a characteristic square section, with a lateral size between 1 m and 2.5 m. Therefore, the main goal of this step is to find four vertical lines, which follow this geometric pattern. To do so, the algorithm will first search for pairs of vertical lines that are between 0.8 m and 2.7 m apart ($\pm$ 0.2 m of the original range). Then, to ensure that the selected lines are in similar height ranges, the algorithm checks that the Z value ranges overlap.

Figure 7a illustrates the four possible regions where the other two steel profiles could be present. These are according to the standard dimensions of cranes, which are illustrated in Figure 1.



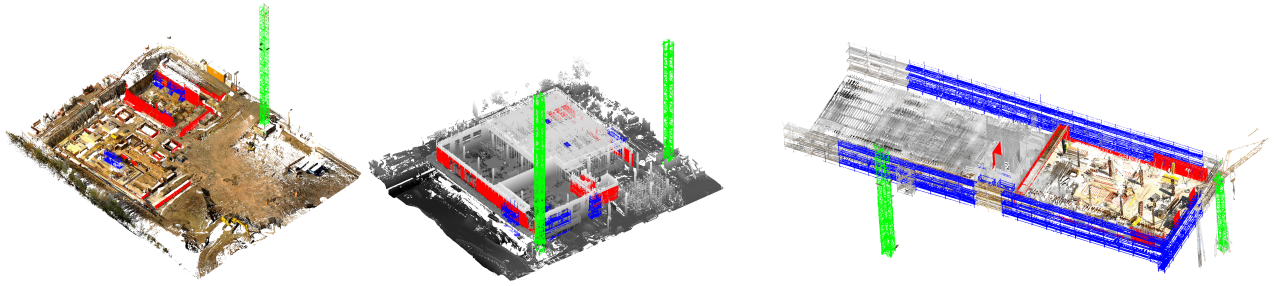
(a) (b)

Figure 7: Location of possible lines: top view of the vertical lines (a) in green the crane lines, (b) in red scaffold lines. A pair of vertical lines are indicated with a dashed line. The other pair could be in the blue regions. These regions result from an offset to the left and the right from the first detected pair in the middle. In both examples, the other couples were successfully found, since they are in the blue regions.

Afterwards, to determine whether the four lines indeed represent a crane or not, three checks are carried out. First, if there is a crane, points should be present between every two continuous vertical lines. The second check examines the presence of a horizontal line between these vertical lines, with a length of at least 80 % of the distance between them. Finally, and exclusively for cranes, a total height check reveals the ultimate location of the detected cranes. As cranes are usually the highest objects in a construction site, their height should not be lower than 10 m below the point cloud's maximum Z value. This last check serves to differentiate the cranes from similar but lower elements such as shoring.

Scaffold detection

The scaffold detection process follows very similar steps as the crane detection, with two main differences: Firstly, the threshold values of the ROIs separation phase are different (see Table 1). Secondly, detecting the pattern on vertical lines is also adjusted to detect not only square but also rectangular patterns that are characteristic for a scaffold. This adjustment is done with the distances shown in Figure 7b, in accordance to the regulations regarding scaffold dimensions, as shown in Figure 2.



(a) (b) (c)

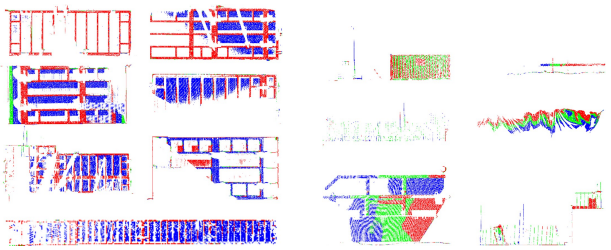
Figure 9: Automatically segmented point clouds: (a) dataset Nr. 1; (b) dataset Nr. 2 (as it is originally colorless, it is shown here with height ramp gray-scale colors); (c) dataset Nr. 3. In green detected cranes, in blue detected scaffolds, and in red detected formwork elements.

Formwork detection

The formwork detection procedure differs from the other two presented detection processes in two aspects: Firstly, while the threshold values are very similar to those used for wall ROIs separation, once the ROIs with formwork are separated from the whole point cloud, they are then filtered in blobs that are aligned to the X and Y-axes. Secondly, in every aligned blob point cloud, vertical cross-sections are generated and classified with a Deep Learning (DL) algorithm, revealing the location of the formwork elements.

To find the right location where these cross-sections must be created, 2D lines are detected in a vertical projection of the point cloud in every blob. For horizontal blobs, the algorithm search for the lowest and the highest horizontal lines. If the difference between them is larger than 11 cm (the minimum width of formwork (PERI 2014, p. 42)), then there might be a formwork element. To finally identify which blobs contain formwork elements, two vertical cross-sections are generated for every blob, one from the top and another from the bottom. Then a DL algorithm classifies these cross-sections as formwork or non-formwork. Something unique about these cross-sections is that they contain depth information; this enables the DL algorithm to consider the exterior studs and walings as well as the interior wall plane surface.

The PyTorch C++ frontend was used to train and test the used DL algorithm. The neural network used consists of five convolutional layers with max-pooling and ReLU activation and three fully connected layers. 244 images were used to train the model; these were generated with dataset Nr. 1, and a data augmentation step. Figure 8 shows a subset of these images.



(a) (b)

Figure 8: Set of vertical cross-section to classify formwork: (a) formwork; (b) non-formwork.

RESULTS AND DISCUSSION

The proposed method's performance was validated on three different point clouds from a construction site in Germany acquired at different stages of the construction progress with a terrestrial laser scanner, specifically with the FARO Laser Scanner Focus S 350 Plus.

Table 2 enumerates the different datasets, providing additional information about their aligned dimensions, the area they cover, and the number of points they contain. Figure 9 presents the segmentation results of the three data sets.

Table 3 shows the validation results for every dataset, giving every target object precision and recall. These were calculated based on the number of points on the respective segmented point cloud.

Table 2: Point cloud Datasets.

Nr.	Δx, Δy, Δz [m]	Area [m^2]	Nr. of points
1	71, 58, 46	4,118	127,121,272
2	53, 60, 46	3,180	223,272,813
3	39, 78, 25	3,042	67,213,140

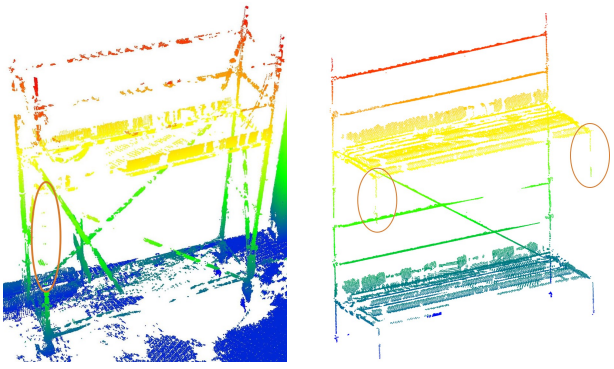
The proposed algorithms were all developed in C++ and tested on a laptop with a 2.80 GHz CPU and GTX 1050 GPU. Table 4 presents the times in seconds of the main steps for each dataset.

Table 3: Validation Results for each dataset.

Nr.	Object	Precision	Recall
1	Crane	100.0 %	100.0 %
	Scaffold	100.0 %	100.0 %
	Formwork	85.1 %	68.1 %
2	Crane	100.0 %	100.0 %
	Scaffold	89.1 %	95.1 %
	Formwork	36.4 %	90.3 %
3	Crane	100.0 %	100.0 %
	Scaffold	100.0 %	82.6 %
	Formwork	85.1 %	100.0 %
	Overall	88.4 %	92.9 %

Table 4: Computational time in seconds for each dataset.

Step		Dataset Number		
		Nr.1	Nr.2	Nr.3
Preprocessing		67	103	34
Crane detection		51	381	95
Scaffold det.		168	2245	726
Formwork det.		153	148	72
Total time	[s]	439	2877	927
	[min]	7.3	48.0	15.5

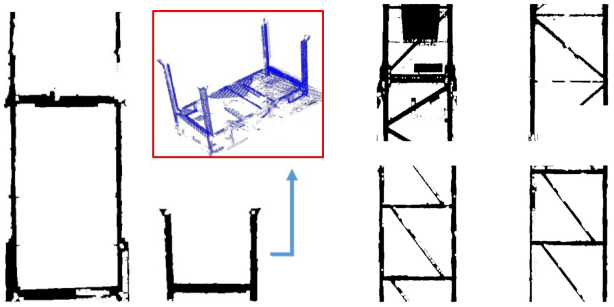


(a) (b)

Figure 10: False negative scaffolds: (a) non-detected scaffold in dataset Nr. 2; (b) one instances of a non-detected scaffold in dataset Nr. 3. The colors in this figure are according to the height of the points.

Discussion

The results produced by the proposed technique are promising. While cranes and scaffold detection achieve precision and recall above 82.6 %, there is more room for improvement regarding formwork detection, where the minimum rates were 36.4 % and 68.1 %. There are two main reasons for these low metrics: Firstly, the method classifies sections of point clouds as formwork or non-formwork. This fact result in low precision in cases when, e.g., only half of a large wall is covered by formwork. Secondly, the low recall in dataset Nr. 1 is due to the presence of occlusion in foundation formwork. This dataset was acquired with only 11 scans, leaving several foundation formworks, located in their respective excavation pits, very occluded.



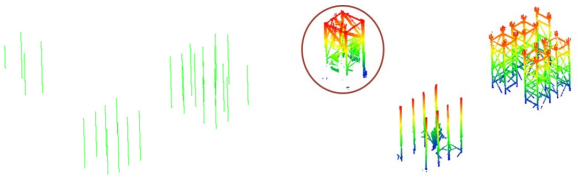
(a) (b)

Figure 11: Similar objects (a) cross-sections of scaffold (left) stacking pallets (right), the latter are wrongly classified as scaffolds; (b) cross-sections of cranes (up) shoring (down), the latter have similar cross-sections as cranes.

The precision of scaffold detection was affected by stacking pallets for props, which were wrongly classified as scaffold elements. This misclassification is caused by the fact that those elements show four vertical lines in the scaffold ranges and their cross-section also has a horizontal line, as illustrated in Figure 11a. Occlusions were again the cause why the recall was not perfect. As shown in Figure 10, even if only one up-right was occluded, the method is not able to detect the scaffold.

While the crane detection results are impressive, there are cases when the method will not work. For example, when banners are hanging on the side of the tower crane. With these elements, the proposed technique will prefilter the crane as a wall in the ROIs separation step. This issue is also present in the case of scaffolds covered with safety screens. Another interesting finding in this research is that shoring elements and cranes have very similar cross-sections, as shown in Figure 11b. To avoid this problem, the total height of the elements relative to the maximum point cloud height is compared. However, this solution implies the manual deletion of the jib of the crane.

The technique proposed by Wang (2019) relies on a first manual point cloud clipping of a small region where scaffolds are present. Since it takes the convex



(a) (b)

Figure 12: Detected groups of vertical elements for cranes only using the vertical lines: (a) detected vertical lines; (b) the corresponding point cloud inside the regions delimited from the groups of vertical lines. Note that even when there is only a single crane, the code detected more elements with the same pattern in vertical lines. Most of them are props (in the middle of the image (b)) and shoring (in the right).

hull of the detected uprights in a 2D projection, it will not filter successfully only scaffold elements in cases when many of them are present, like in the Test dataset Nr. 3 of this paper. On the contrary, the technique proposed here can be applied directly on large datasets, without restrictions on the amount or position of the scaffold instances.
Xu et al. (2018) limited their approach to detecting scaffolds next to a facade and with a particular bay width of 0.8 m. Considering more possible scaffold dimensions makes the proposed technique more robust. However, it will give lower performance than Xu et al. (2018) in low-quality point clouds.

Regarding the computational time, the method requires in average 1 s to process 10^5 points. However, it takes much more time in dataset Nr. 2 compared to the other two datasets. The reason for that is the presence of shoring and props that support slab formwork. As illustrated in Figure **??**, these elements have the same pattern in vertical lines as cranes. Therefore the method has to generate many cross-sections and perform the occupancy and the horizontal line check, demanding more time.

Nonetheless, in comparison with Wang (2019), the technique does not generate horizontal slices every 0.05 m and fits circles in each of them, which certainly requires more time. Additionally, in comparison with the deep learning method proposed by Zeng et al. (2020), their approach would require 15 s only to extract the deep features from a point cloud with 10^5 points. This is 15 times more than the average time that the proposed technique requires to detect the three target objects. However, their technique would be more appropriate to recognize objects with more complex geometries.

CONCLUSIONS

This paper investigated the detection of temporary elements in a construction site's point cloud, without the need of a previous integration with a BIM model and exploding mainly the objects' verticality to achieve a fast detection. In conclusion, one could argue that as long as there is a way to infer primary geometrical constraints on the target objects, it is possible to achieve awe-inspiring performance on a 3D object detection problem. This achievement is not only in terms of accuracy but also in computational time.

In this research, the target objects' vertical orientation and their minimum height, and other geometrical features played a crucial role in detecting them. Such a technique would not apply to all objects (e.g., deformable objects). Nonetheless, the process is not limited to a few given examples or object size restrictions.

Furthermore, using 2D and 2.5D projections allows the implementation of a very efficient method to filter and detect objects on massive point clouds. Finally, implementing a deep learning algorithm to classify 2.5D vertical cross-section projections proved to be very suitable for formwork classification, facilitating also a future possible extension of the method to detect other elements, e.g., reinforcement, containers, fences, etc.

Future work

Additional validation on more datasets, with temporary objects from different manufacturers will serve to test and improve the robustness of the method. Moreover, the recognition of vertical and horizontal placed reinforcement would complete the primary set of not permanently-visible objects that determine the current state of the construction progress.

Later, to achieve a fully automated construction monitoring, the integration with a detailed 4D building information model containing the permanent structures' geometry and time information is required, as done by Braun et al. (2020). This integration should be easier after the detection of the temporary objects and would also enable identifying and verifying openings and essential building elements in the right location on the construction site.

Subsequently, and as done by Kim et al. (2020), an automated dimensional quality assessment can also be performed to ensure compliance with the structural plans.

Safety regulations can also be verified in cranes and scaffold elements, for the latter Wang (2019) already proposed a method that requires the detection of every component of the scaffolds, such as guardrails, toe-boars, and working platforms.

ACKNOWLEDGEMENTS

We want to thank FARO EUROPE GmbH & Co. KG. for funding this project and supporting and provisioning computing infrastructure and data essential to this publication.

References

Ahmed, M., Seraj, R. & Islam, S. M. S. (2020), 'The k-means algorithm: A comprehensive survey and

performance evaluation', Electronics (Switzerland) **9**(8), 1–12.

Armeni, I., Sener, O., Zamir, A. R., Jiang, H., Brilakis, I., Fischer, M. & Savarese, S. (2016), '3D semantic parsing of large-scale indoor spaces', Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition **2016-Decem**, 1534–1543.

Armstrong, G. & Gilge, C. (2017), 'Global Construction Survey: Make it, or break it–reimagining governance, people and technology in the construction industry'.

Bosché, F. (2012), 'Plane-based registration of construction laser scans with 3D/4D building models', Advanced Engineering Informatics **26**(1), 90–102.

Bosché, F., Ahmed, M., Turkan, Y., Haas, C. T. & Haas, R. (2015), 'The value of integrating Scan-to-BIM and Scan-vs-BIM techniques for construction monitoring using laser scanning and BIM: The case of cylindrical MEP components', Automation in Construction **49**, 201–213.

Bosché, F. & Haas, C. T. (2008), 'Automated retrieval of 3D {CAD} model objects in construction range images', Automation in Construction **17**(4), 499–512.

Böttcher, P. D. P. & Neuenhagen, H. (1997), Baustelleneinrichtung: betriebliche Organisation, Geräte, Kosten, Checklisten, Bau-Verlag.

Braun, A. & Borrmann, A. (2019), 'Combining inverse photogrammetry and BIM for automated labeling of construction site images for machine learning', Automation in Construction **106**, 102879.

Braun, A., Tuttas, S., Borrmann, A. & Stilla, U. (2020), 'Improving progress monitoring by fusing point clouds, semantic data and computer vision', Automation in Construction **116**, 103210.

Braun, A., Tuttas, S., Stilla, U., Borrmann, A. & Center, L. O. (2016), 'Incorporating knowledge on construction methods into automated progress monitoring techniques'.

Golparvar-Fard, M., Pena-Mora, F. & Savarese, S. (2011), Monitoring changes of 3D building elements from unordered photo collections, in '2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)', IEEE.

Golparvar-Fard, M., Peña-Mora, F. & Savarese, S. (2015), 'Automated Progress Monitoring Using Unordered Daily Construction Photographs and IFC-Based Building Information Models', Journal of Computing in Civil Engineering **29**(1), 4014025.

Guo, Y., Wang, H., Hu, Q., Liu, H., Liu, L. & Bennamoun, M. (2019), 'Deep learning for 3D point clouds: A survey', arXiv pp. 1–27.

Han, K., Degol, J. & Golparvar-Fard, M. (2018), 'Geometry- and Appearance-Based Reasoning of Construction Progress Monitoring', Journal of Construction Engineering and Management **144**(2), 4017110.

Hu, Q., Yang, B., Xie, L., Rosa, S., Guo, Y., Wang, Z., Trigoni, N. & Markham, A. (2020), 'RandLA-Net: Efficient semantic segmentation of large-scale point clouds', Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition pp. 11105–11114.

Kim, C., Son, H. & Kim, C. (2013), 'Automated construction progress measurement using a 4D building information model and 3D data', Automation in Construction **31**, 75–82.

Kim, M.-K., Thedja, J. P. P. & Wang, Q. (2020), 'Automated dimensional quality assessment for formwork and rebar of reinforced concrete components using 3D point cloud data', Automation in Construction **112**, 103077.

Landrieu, L. & Simonovsky, M. (2018), 'Large-Scale Point Cloud Semantic Segmentation with Superpoint Graphs', Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition pp. 4558–4567.

Lin, J. J. & Golparvar-Fard, M. (2020), Construction Progress Monitoring Using Cyber-Physical Systems, in 'Cyber-Physical Systems in the Built Environment', Springer, pp. 63–87.

Lu, X., Liu, Y. & Li, K. (2019), 'Fast 3D Line Segment Detection From Unorganized Point Cloud', arXiv:1901.02532 .

Maalek, R., Lichti, D. D. & Ruwanpura, J. Y. (2019), 'Automatic Recognition of Common Structural Elements from Point Clouds for Automated Progress Monitoring and Dimensional Quality Control in Reinforced Concrete Construction', Remote Sensing **11**(9), 1102.

Mace, B. & Jones, S. (2016), 'How satisfied, really satisfied, are Owners?'.

Mukhopadhyay, P. & Chaudhuri, B. B. (2015), 'A survey of Hough Transform', Pattern Recognition **48**(3), 993–1010.

Nikoohemat, S., Diakité, A. A., Zlatanova, S. & Vosselman, G. (2020), 'Indoor 3D reconstruction from point clouds for optimal routing in complex buildings to support disaster management', Automation in Construction **113**, 103109.

PERI (2014), 'DOMINO Panel Formwork: Instructions for Assembly and Use – Standard Configuration'.

Schach, R. & Otto, J. (2017), Baustelleneinrichtung, Springer Fachmedien Wiesbaden.

Turkan, Y., Bosché, F., Haas, C. T. & Haas, R. (2014), 'Tracking of secondary and temporary objects in structural concrete work', Construction Innovation **14**(2), 145–167.

Wang, Q. (2019), 'Automatic checks from 3D point cloud data for safety regulation compliance for scaffold work platforms', Automation in Construction **104**, 38–51.

Xu, Y., Tuttas, S., Hoegner, L. & Stilla, U. (2018), 'Reconstruction of scaffolds from a photogrammetric point cloud of construction sites using a novel 3D local feature descriptor', Automation in Construction **85**, 76–95.

Yasmin (2019), 'Liebherr Tower Crane Parts Mast Section 180HC/256HC/290HC'.

Zeng, S., Chen, J. & Cho, Y. K. (2020), 'User exemplar-based building element retrieval from raw point clouds using deep point-level features', Automation in Construction **114**, 103159.