Smart Infrastructure and Construction
Pages 1–20

Keywords: Digital representation/ Point cloud object detection/ Temporary works/ Construction/ Artificial intelligence

ice | proceedings

Recognition of Temporary Vertical Objects in Large Point Clouds of Construction Sites

Miguel A. Vega Torres
Alexander Braun
Florian Noichl
André Borrmann
Chair of Computational Modeling and Simulation,
Technical University of Munich,
Arcisstr. 21, 80333 Munich, Germany

Heiko Bauer
Denis Wohlfeld
FARO EUROPE GmbH & Co.KG.,
Lingwiesenstraße 11/2, 70825 Korntal-Münchingen, Germany

Although adherence to project schedule is the most critical performance metric among project owners, still 53 % of typical construction projects exhibit schedule delays. While construction progress monitoring is key to allow effective project management, it is still a largely manual, error prone and inefficient process. To contribute to more efficient construction progress monitoring, this research proposes a method to automatically detect the most common temporary object classes in large-scale laser scanner point clouds of construction sites. Finding the position of these objects in the point cloud can help determine the current state of construction progress and verify compliance with safety regulations. The proposed workflow includes a combination of several techniques: image processing over vertical projections of point clouds, finding patterns in 3D detected contours, and performing checks over vertical cross-sections with deep learning methods. After applying and testing the method on three real-world point clouds and testing with three object categories (cranes, scaffolds, and formwork), the results reveal that our technique achieves rates above 88 % for precision and recall and outstanding computational performance. These metrics demonstrate the method's capability to support the automatic 3D object detection in point clouds of construction sites.

1. Introduction

Nowadays, inefficiencies, such as cost and time overruns, occur regularly within the construction industry. According to Mace and 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 and 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 and Golparvar-Fard, 2020).

Many approaches have emerged to address this problem. Recent 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 4D 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 as-built point cloud. Some of the most common temporary elements are:

scaffolds, formwork, cranes and reinforcement (Schach and Otto, 2017).

While recently there has been some effort to incorporate temporal structures into building information models (Jin and Gambatese, 2019; Pham *et al.*, 2020; Rodrigues *et al.*, 2021), usually these elements are not present in the model (Kim and Cho, 2015). Additionally, these temporary elements may occlude large portions of the permanent structures in the point cloud since they are adjacent to them. Notably, formwork and scaffolding occlude the direct view on permanent walls or slabs, making a reliable comparison with the 3D geometry of the model more challenging and hindering the detection of the current state of construction progress (Braun *et al.*, 2020).

To overcome this challenge, this study proposes a method to automatically detect cranes, scaffolds, and formwork in laser-scanned point clouds of construction sites. More specifically, this study tries to find an answer to the following research question: How is it possible to detect those three classes of objects efficiently, and accurately in large and complex point clouds?

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. Furthermore, the crane and its exact relative position to the building can support other methods that use cameras mounted on crane to track the construction progress (Braun *et al.*, 2015) or construction workers, such as the methods proposed by Neuhausen *et al.* (2020, 2018). One of the main ways this knowledge can be exploited is to enable automatic alignment of the point cloud with a reference BIM model, an issue that has been addressed by Masood *et al.* (2020).

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. These verifications can 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. After 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. Moreover, an automated dimensional quality assessment can also be performed as done by Kim *et al.* (2020), in which compliance with the structural plans can be ensured before pouring concrete.

After identifying the potential benefits of detecting temporary elements in point clouds of construction sites, this paper proposes an efficient method to automate this detection. The developed approach first rotates the point cloud to align it to the building axes; then, it locates crane masts and scaffolds in the point cloud. Subsequently, axis-aligned formwork elements are detected.

This paper is structured as follows. Section 2 reviews recent literature on construction progress monitoring and object detection in point clouds of construction sites, and identifies the research gaps. Then, the geometry of the target objects is described in detail according with the respective regulations in Section 3, which serve as a base for the developed object detection technique. Section 4 explains the proposed approach in this research. It illustrates the workflow of the implemented vertical object detection method. Section 5 reports results, analysis and validations of the proposed method. On top of that, computational performance analyses are presented. Section 6 drives the conclusions of this study and suggests possible future research directions.

2. Related Research

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 and Borrmann, 2019; Braun *et al.*, 2016; Amer and Golparvar-Fard, 2018), others use laser scanner point clouds (Bosché and Haas, 2008; Bosché, 2012; Turkan *et al.*, 2012; Kim *et al.*, 2013; Bosché *et al.*, 2015; Han *et al.*, 2018; Son *et al.*, 2017). Additionally there have been techniques developed that only use image information (Kropp *et al.*, 2018; Acharya *et al.*, 2019; Asadi *et al.*, 2019; Álvares and Costa, 2019).

To compare the acquired sensor information and the the prior BIM Model (also called Scan-vs-BIM), the existence of a 3D/4D building information model is a requirement. With a 4D model (in which every element has time information when it should be built) 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 (2014) made initial proposals to track temporary elements. However, their method is based on a Scan-vs-BIM approach that requires a BIM model and does not detect different temporary elements separately. 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; Fichtner, 2016; Macher *et al.*, 2017) (also call Scan-to-BIM). These methods focus mainly on the detection of floors, walls, ceilings, doors and windows in a point cloud. However, there is only limited research on the detection of cranes, scaffold or formwork elements in point clouds of construction sites.

While deep learning approaches for point cloud semantic segmentation seem to be very promising (Guo *et al.*, 2019), 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 and 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 prior 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.

This paper has two main differences in comparison with the methods proposed by Xu *et al.* (2018) and Wang (2019) to detect scaffold elements. First, it is applicable in large scale dense point clouds. That is because it filters the point cloud in regions of interest with image processing techniques in an efficient manner. Second, it is not restricted to a specific bay width distance (as the method proposed by Xu *et al.* (2018)) or specific geometric scaffold configuration (as the method proposed by Wang (2019)). Uniquely, our process allows the detection of almost all types of scaffolds that have a geometry following the corresponding regulations (as explain in 3).

Furthermore, in this paper, we focus not only on the detection of scaffold elements but also on a generalized method that can serve to efficiently detect the majority of vertical elements in point clouds. Our case studies focus on detecting cranes and formwork elements, but the the main steps of our pipeline can be used with slight modifications or extensions to detect for example walls, reinforcement, containers, fences, shoring and stacking pallets.

3. 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.

3.1. 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 paper focuses mainly on tower cranes because they are the most commonly used in the construction of tall buildings (Böttcher and 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 square, its width can vary between 1.2 m and 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.

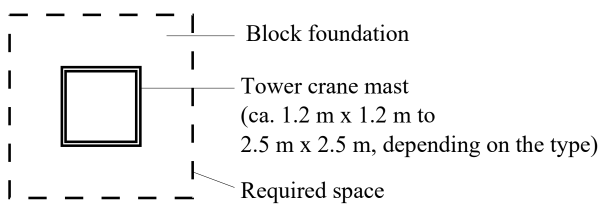


Figure 1. Top view of a tower crane mast with dimensions (Schach and Otto, 2017, p. 28).

3.2. 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, toe-boards, 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 paper 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 $2.4\,\mathrm{m}$, in this paper, only scaffold with a maximum width of $1.2\,\mathrm{m}$ 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 and Otto, 2017, p. 240), which have a width between the selected range ($0.6\,\mathrm{m}$ to $1.20\,\mathrm{m}$) in accordance with Table 1 of DIN EN 12 811-1. Similarly, the scaffold bay length could vary between $1.5\,\mathrm{m}$ to $3\,\mathrm{m}$ in line with DIN 4420-4. Figure 2 presents the main components of a scaffold, together with its standardized minimum and maximum dimensions.

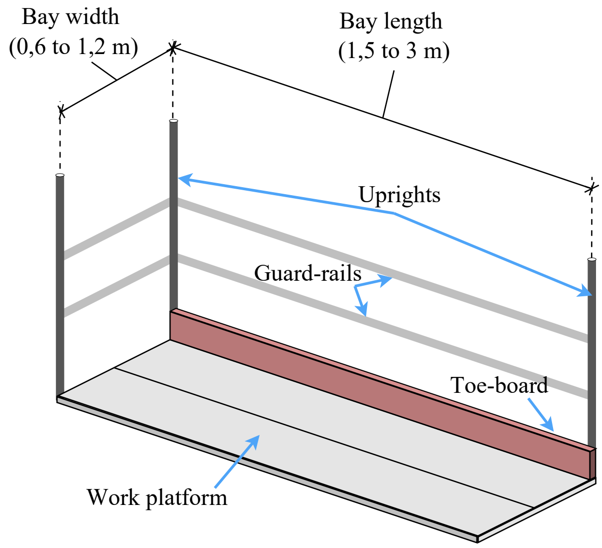
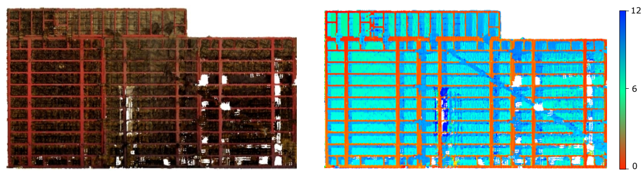


Figure 2. Main scaffold components and dimension ranges

3.3. 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 paper 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 3b) together with the wall panel will be exploited to detect formwork elements.



(a) (b)

Figure 3. Front view of a point cloud with formwork elements: (a) with the original RGB colors; (b) the color is in accordance with the depth of the points: red are the closest, blue the farthest from a front view perspective (units in centimeters)

4. Methodology

4.1. Overview

The workflow of the object detection method introduced in this paper is illustrated in Figure 4.

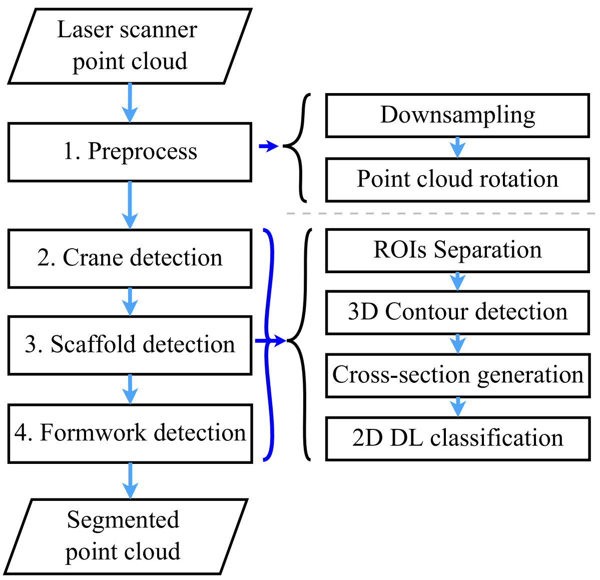


Figure 4. Workflow overview.

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 ROIs (Regions of interest) that may contain cranes are separated using image processing techniques over a vertical projection of the point cloud in the XY plane. Subsequently, an algorithm will search a pattern characteristic of a tower crane in detected 3D vertical lines (in the Z direction), 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).

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 algorithm is leveraged to determine the presence of formwork elements (see Formwork detection). It is worth mentioning that steps 2, 3, and 4 (i.e., crane, scaffold, and formwork detection) are independent of each other and the order does not influence the result. While they are execute one after each other in our pipeline, they could be executed in parallel.

4.2. Preprocessing

4.2.1. Downsampling

Filtering or downsampling the point cloud with the voxel grid method is vital for two reasons: First, it allows 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 contribution to reducing the computational cost as the number of points is reduced substantially, in all cases where 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 (Point Cloud Library) voxel grid method with a leaf size (VG_{ls}) of 5 mm in every leaf voxel of the octree. The VG method approximates the point cloud with the centroid in every voxel, it might not accurately represent the underlying surface in cases where there is a lot of noise in the data or the leaf size is large and the objects present curved surfaces.

4.2.2. Point cloud rotation

This step rotates 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* (Coughlan and Yuille, 1999)).

This rotation is done in two main steps (which will be explained more in detail later): First, Walls ROIs Separation with image

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

Table 1. Parameter Summary. Here R stands for rectangular structuring element and E for elliptical, MAX means that there is no upper limit for the blob area.

processing in a vertical projection, and second, 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. While for now this process is done manually, this could be automated by detecting the peaks of the histogram of the points projected in the Z-axes, as done by (Fichtner *et al.*, 2018; Turner and Zakhor, 2014; Oesau *et al.*, 2014). This separation is a requirement for the process to be able to filter objects by their minimum height. Figure 6 illustrates a building's first floor.

4.2.3. Walls ROIs separation

To find the building's structural axes, we first separate large load-bearing walls from the rest of the point cloud. The rationale for that is based on the assumption that large load-bearing walls are aligned with the building's structural axes, as it is usually the case.

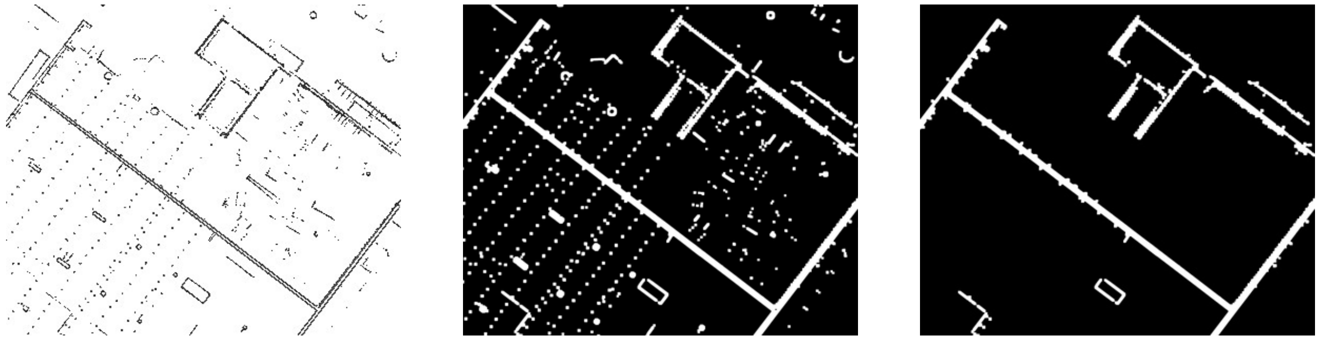
To filter load-bearing walls from the rest of the point cloud a vertical projection is generated in a gray scale image. As the point cloud was already downsampled, it is known that the minimum distance between two points is $5\,\mathrm{mm}$ (considering the usage of the voxel grid method with a (VG_{ls}) of $5\,\mathrm{mm}$).

Therefore, a point cloud vertical projection in a 2D 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 (e.g., possible presence of formwork covering the walls), it is assumed that vertical walls may have at least $1.2\,\mathrm{m}$ (h_{min}) of projected vertical height, which is around half of the height of an average wall.

To make this point clear, consider a vertical line of $1\,\mathrm{m}$ length. If the line is formed by points every $5\,\mathrm{mm}$, it implies that the line is actually a column of 200 points. If these points are projected in the XY plane in a grayscale accumulation image, they will be represented as a pixel with value 200. In this way, it is possible to separate objects of different heights using a vertical projection, as long as they have a vertical non-occluded surface

Certainly, it would not be reliable to estimate the height of the wall without having a point could with a low resolution (i.e., very dense point cloud in which the minimum distance between two points is less than $5\,\mathrm{mm}$), this assumption is justified, considering that the sensor used for scan acquisition can scan up to $350\,\mathrm{m}$ with a ranging error of $1\,\mathrm{mm}$ and an accuracy of 19 arcsec



(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 from one single large blob; (c) final Wall ROIs (W_{regions}) after separation by blob size. Test dataset: Nr. 2.

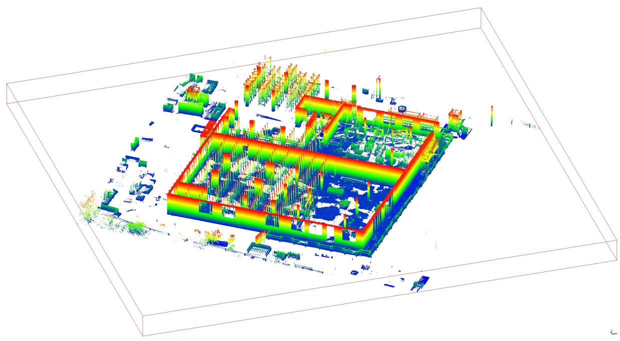


Figure 6. Clipped first floor of the test dataset Nr. 2.

for vertical/horizontal angles. In the case of point clouds with lower quality (e.g., photogrammetric or captured with mobile laser scanner) the presence of noise may not allow to have perfectly aligned points over a vertical surface.

Subsequently, to join small groups of connected white pixels (also called blobs) that are close to each other and may constitute more oversized objects, ten iterations of a morphological dilation with a *structural element* (S) with a rectangular shape of size 10 x 10 (S_{R10}) are applied giving the result illustrated in Figure 5b. Later the blobs can be separated by its number of white pixels, which is equivalent to its area.

For example, for walls, a minimum area of $A_{min} = 1.5\,\mathrm{m}^2$ was considered more appropriate, assuming that the minimum length of all walls is $5\,\mathrm{m}$ and its width is $0.3\,\mathrm{m}$. 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 of $1.2\,\mathrm{m}$.(as stated in Table 1).

4.2.4. 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 and Chaudhuri, 2015), with an angular resolution of $\pi/(180 \cdot 100)$ (i.e., two decimal precision in degrees), 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.

After point cloud downsampling and alignment with the axes of the coordinate system, the next step is the detection of the target objects, which is described in the following section.

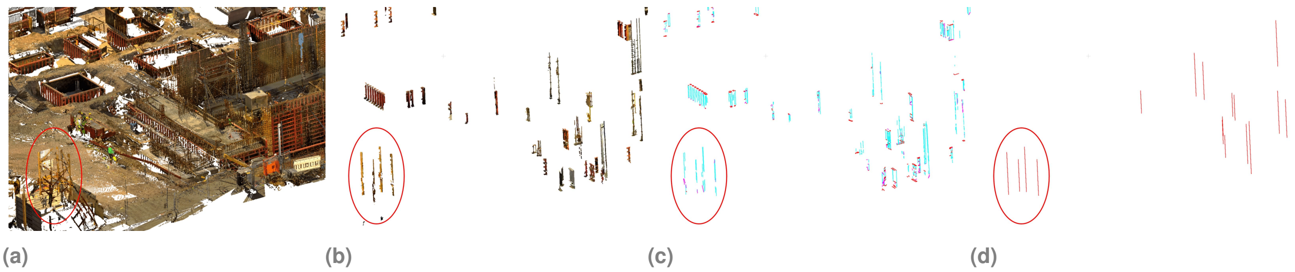


Figure 7. Detection of possible crane lines: (a) original point cloud with a red ellipse indicating the location of the crane; (b) Crane ROIs in (a), notice the presence of other thin and tall objects in addition to the crane; (c) detected 3D contours in (d) ;(d) filtered merged vertical lines from (c). Test dataset: Nr. 1.

4.3. Crane detection

The detection of cranes starts with a similar step as the one used to separate the wall ROIs but with adapted parameters of minimum height, dilation, and blob size (see Table 1). This step contributes to efficiently filter out points that are more likely to belong to a crane from the rest of the point cloud. In Figure 7b, all the elements that pass the filter are shown.

Since the detection of cranes is based on the search of the characteristic pattern of the four vertical steel profiles of a crane, the point cloud is reduced one more time to straight edges detected and then the pattern is searched on them.

Therefore, in the next step, 3D contours are efficiently detected with the algorithm provided by Lu *et al.* (2019). In this algorithm the process is divided into three main steps: First, the point cloud is segmented in regions based on the previous calculation of the Principal Component Analysis (PCA) information of every point; second, 3D planes are fitted in every region and lines are detected over a 2D projection of this planes which are then projected back to the 3D space; and finally, in a post-processing step, the detected 3D lines are passed through an outlier removal and a horizontal merging process.

The implementation of the algorithm of Lu *et al.* (2019) plays a crucial role in the proposed object recognition method, not only because it allows translating from unorganized points to 3D lines that delineate the objects, but also because it is fast. Therefore its implementation in large point clouds is very convenient. Figure 7c illustrates the 3D line detection results in a point cloud with the crane ROIs.

Subsequently, the center of the vertical lines are projected as points in the XY-plane and then merged in single lines if there is a maximum distance of $20\,\mathrm{cm}$ 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 $12.5\,\mathrm{cm}$ (Yasmin, 2019). These merged lines are then also filtered by their minimum length (see l_{min} in Table 1), resulting in the final long merged lines presented in Figure 7d.

Now that the vertical lines are detected, the pattern that characterizes a crane will be searched in these vertical lines.

As explained in Section 3.1, the mast of a tower crane always has a characteristic square section, with a lateral size between $1\,\mathrm{m}$ and $2.5\,\mathrm{m}$. Therefore, the main goal of this step is to find four vertical lines, which follow this geometric pattern. Figure 8a illustrates the possible regions where the steel profiles could be present.

As shown in Algorithm 1, the method will first search for pairs of vertical lines that are between $0.8\,\mathrm{m}$ and $2.7\,\mathrm{m}$ apart ($\pm\ 0.2\,\mathrm{m}$ of the original range). Then, to ensure that the selected lines are in similar height ranges, we verify the presence of an overlap of the Z value ranges.

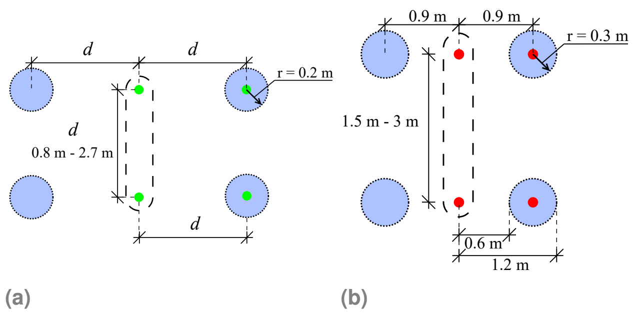


Figure 8. Location of possible lines: top view of the vertical lines (a) in green the crane lines, (b) in red scaffold lines. A pair of vertical lines are indicated with a dashed line. The other pair could be in the blue regions. These regions result from an offset to the left and the right from the first detected pair in the middle. In both examples, the other couples were successfully found, since they are in the blue regions.

Algorithm 1: Find pattern in vertical lines

Input : A vector with the merged vertical lines $M = \{L_0, L_1, ..., L_n\}$

Output: Vector of vectors of line indices $P \leftarrow \varnothing$ revealing possible crane locations

```
for ∀(L_i, L_j) ∈ M : i < j do
    p_i ← L_i^0, p_j ← L_j^0
    d ← ||p_i − p_j||
    u ← (p_j − p_i)/d
    u^⊥ ← (−u_y, u_x)
    if 0.8 < d < 2.7 and overlap(L_i, L_j) then
        C ← {p_i + du^⊥, p_j + du^⊥, p_i − du^⊥,
             p_j − du^⊥}
        R ← ∅
        for ∀c ∈ C do
            for ∀L_k ∈ M : i < k and k ≠ j do
                p_k ← L_k^0
                t ← ||p_k − c||
                if t < 0.2 and overlap(L_i, L_k) then
                    R ← R ∪ k
                else
                    R ← R ∪ 0
                end
            end
        end
        saveLineIndices(i, j, R^0, R^1)
        saveLineIndices(i, j, R^2, R^3)
    end
end
```

Afterwards, to determine whether the four lines indeed represent a crane or not, three examinations are carried out. First, if there is a crane, points should be present between every two continuous vertical lines. Secondly, the presence of a horizontal line between these vertical lines, with a length of at least $80\,\%$ of the distance between them is corroborated. The reason not to select $100\,\%$ of the total distance is to take into consideration the presence of possible occlusions in the scan. 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 less than $10\,\mathrm{m}$ below the point cloud's maximum Z value. Objects with a similar pattern in vertical lines and cross-section but lower than this height are disregarded as cranes. This last test serves to differentiate the cranes from similar but lower elements such as shoring.

4.4. 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 accomplished with the distances shown in Figure 8b, in accordance to the regulations regarding scaffold dimensions, as shown in Figure 2.

4.5. 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 filtered in blobs that are aligned to the X and Y-axes. Secondly, in every aligned blob point cloud, vertical cross-sections are generated from the downsampled point cloud and subsequently classified with a Deep Learning (DL) algorithm, revealing the location of the formwork elements.

As the point cloud is already rotated, it is possible to filter out formwork elements that are aligned to the building axes in an efficient manner. To do so, a morphological dilation operation with custom vertical and horizontal kernels is applied over the formwork ROIs. This operation results in two separate binary masks with vertical and horizontal blobs, which are shown in the left of Figure 9.

Subsequently, vertical cross-sections or facade view projections will be generated. 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 horizontal 2D lines with the maximum distance in between,

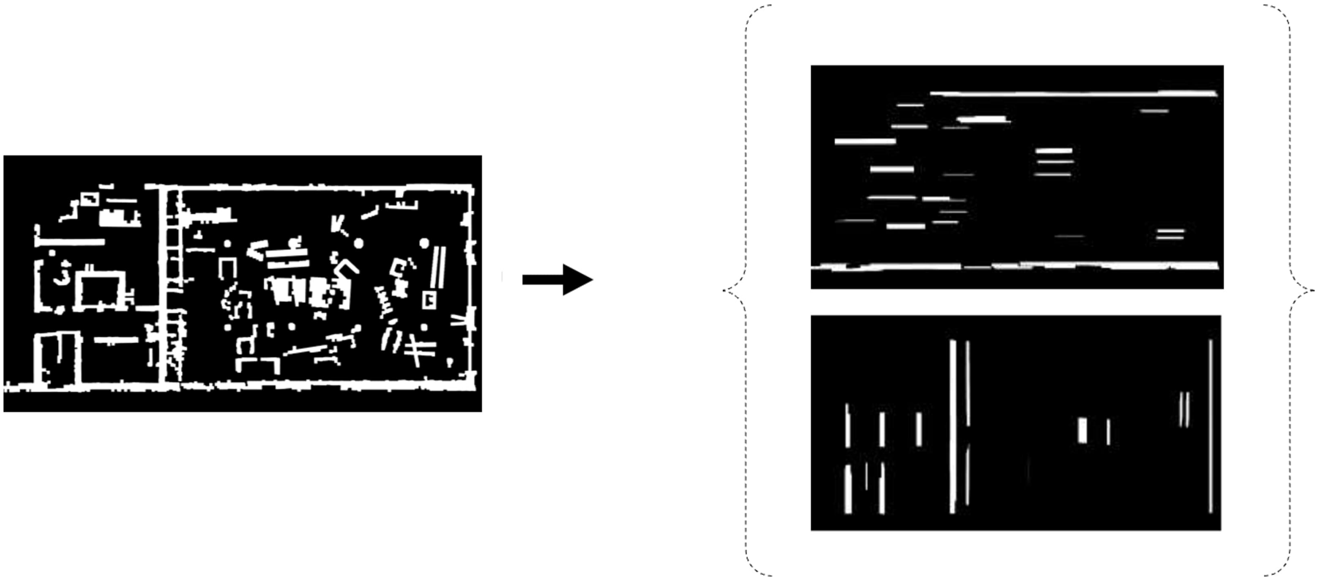
If the difference between them is larger than $11\,\mathrm{cm}$ (the minimum width of industry standard formwork (PERI, 2014, p. 42)), then there might be a formwork element. To allow processing of vertical blobs without major changes to the algorithm, these are rotated by 90° to treat them like the horizontal ones. 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 of the point cloud that is inside the blob.

Subsequently, a DL algorithm classifies these cross-sections as formwork or non-formwork. Compared to object detection in large point clouds, image classification with DL is a research area that has been studied for longer time (Qi *et al.*, 2017; Dai *et al.*, 2021) and which has demonstrated to over-perform even human experts in certain fields (Buetti-Dinh *et al.*, 2019). In contrast to cranes and scaffold's cross-sections, these cross-sections 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 implemented DL algorithm. For the network the AlexNet architecture (Krizhevsky *et al.*, 2012) was used. The training set consists of 244 depth images generated automatically from the point cloud in dataset number 1 and 3 (which will be described in more detail in Section 5). These datasets contain diverse types of formwork elements positioned in different configurations. Figure 10 shows a subset of these images. The data set was augmented with mirror flips over the x, y axes. The dataset was divided into 60 % training set and 40 % testing set. Two Dropouts were used to prevent overfitting, one located after the convolutional layers and the other after the first fully connected layer. Within 90 epochs, the algorithm achieved a maximum accuracy of 93.3 % over the testing set, demonstrating to be suitable for this classification task.

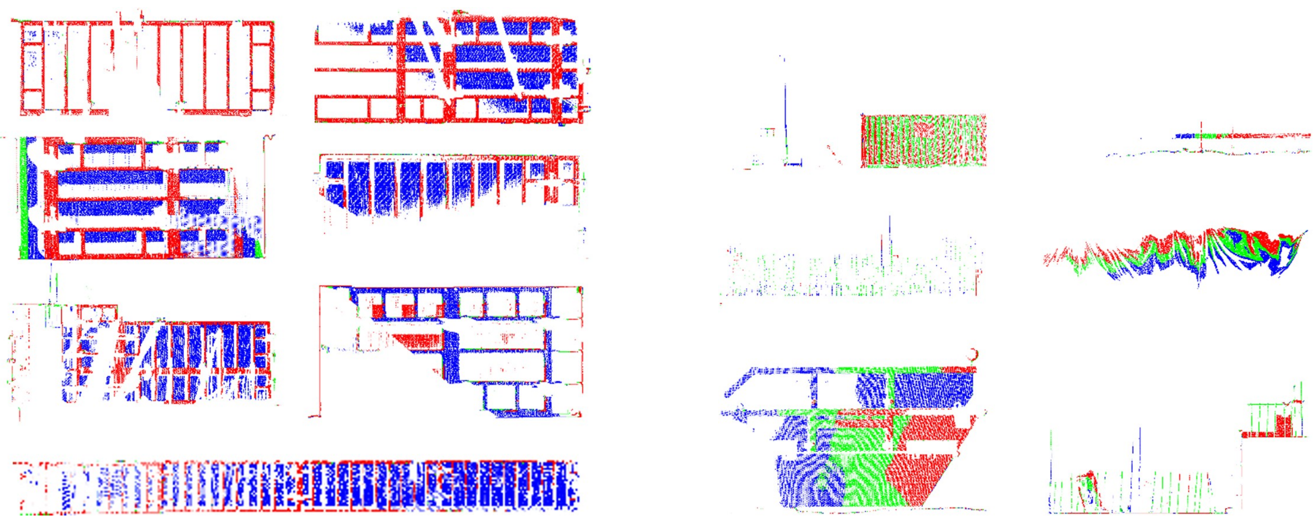
5. Results and Discussion

The proposed method's performance was validated on five different point clouds from construction sites in Germany acquired at different stages of the construction progress with a terrestrial laser scanner.



(a) (b)

Figure 9. Formwork regions of interest (ROIs): (a) original ROIs, (b) ROIs separated into horizontal and vertical formwork blobs.



(a) (b)

Figure 10. Subset of vertical cross-section used to train a CNN to classify formwork: (a) formwork; (b) non-formwork examples.

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 11 presents the segmentation results of the five data sets. While datasets Nr. 1, 2, and 3 belong to the same construction site, datasets Nr. 4 and 5 come from two different construction sites. Dataset Nr. 4 originates form the open-source dataset provided by Eickeler *et al.* (2021).

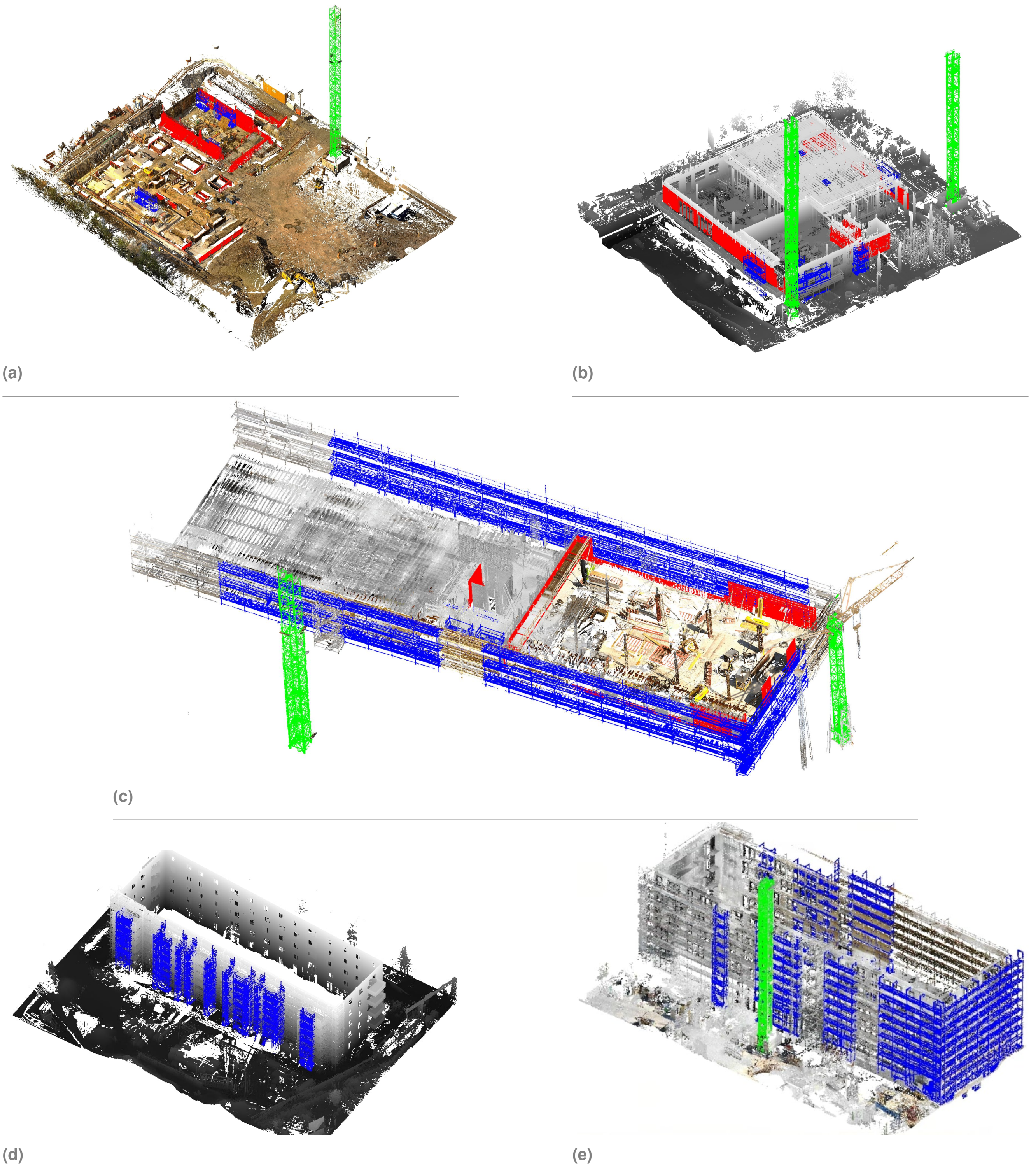


Figure 11. Automatically segmented point clouds: (a) dataset Nr. 1; (b) dataset Nr. 2 (as it is originally colorless, it is shown here with height ramp gray-scale colors); (c) dataset Nr. 3; (d) dataset Nr. 4 (as it is originally colorless, it is shown here with height ramp gray-scale colors); (d) dataset Nr. 5;. In green detected cranes masts, in blue detected scaffolds, and in red detected formwork elements.

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 for formwork elements, and based on the number of detected instances for crane masts and scaffold elements.

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
4	52, 70, 18	3,640	81,774,908
5	69, 38, 38	2,622	132,353,940

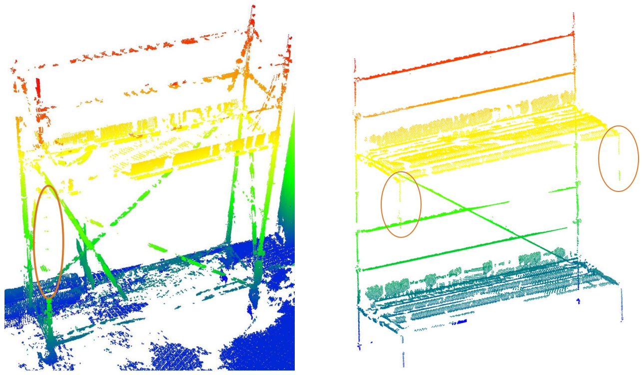
Table 3. Validation Results for each dataset.

Dataset Nr.	Object	Precision [%]	Recall [%]
1	Crane Mast	100.0	100.0
	Scaffold	100.0	100.0
	Formwork	85.1	68.1
2	Crane Mast	100.0	100.0
	Scaffold	89.1	95.1
	Formwork	36.4	90.3
3	Crane Mast	100.0	100.0
	Scaffold	100.0	82.6
	Formwork	85.1	100.0
4	Crane Mast	-	-
	Scaffold	92.9	43.3
	Formwork	-	-
5	Crane Mast	100.0	100.0
	Scaffold	100.0	92.6
	Formwork	-	-
	Overall	90.7	89.3

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

Table 4. Computational time in seconds for each dataset.

Step		Dataset Number				
		1	2	3	4	5
Preprocessing		67	103	34	35	96
Crane detection		51	381	95	22	79
Scaffold det.		168	2245	726	82	266
Formwork det.		153	148	72	50	62
Total time	[s]	439	2877	927	189	503
	[min]	7.3	48.0	15.5	3.2	8.4



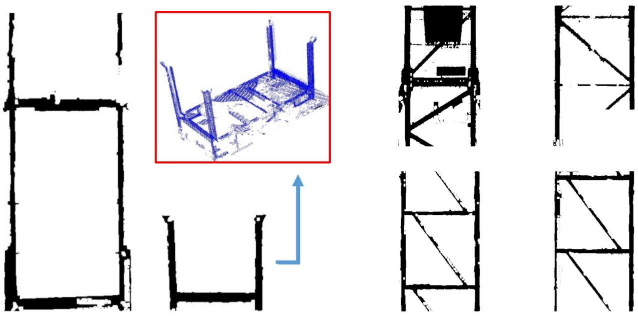
(a) (b)

Figure 12. 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.

5.1. Discussion

The results produced by the proposed technique are promising. While cranes and scaffold detection achieve precision and recall above 89.1 %, 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 13. 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 13a. Occlusions were again the main cause why not all scaffolds were detected. As shown in Figure 12, even if only one up-right was occluded, the method is not able to detect the scaffold.

In dataset Nr. 4 the recall of $43.3\,\%$ corresponds to the high number of false-negative scaffold elements caused by the small distance between the scaffold and the building, which is less than $200\,\mathrm{mm}$. The fact that the scaffold is too close to the building does not allow the algorithm to filter the scaffold ROIs cleanly since the dilation operation joins the projected uprights blobs with the near walls blobs. Subsequently the blob size separation step leaves behind these large blobs, while looking for small uprights projections.

While the crane detection results are impressive, there are cases when the method will not work, in particular when the underlying assumptions are violated. 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, which is a very similar case to when the scaffold is too close to the building. Another interesting finding in this research is that shoring elements and cranes have very similar cross-sections, as shown in Figure 13b. 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. A possible solution to delete the jib of the crane might be deleting the points that are 3 m below the maximum z value, this will delete also the jib, assuming the crane is the highest object in the point cloud.

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 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 or Nr. 5 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 $0.8\,\mathrm{m}$. Considering more possible scaffold dimensions makes the technique proposed here 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 $1\,\mathrm{s}$ 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

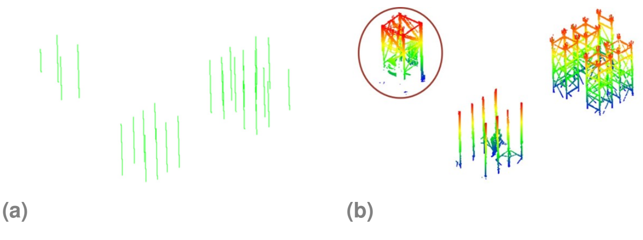


Figure 14. Detected groups of vertical elements for cranes only using the vertical lines: (a) detected vertical lines; (b) the corresponding point cloud inside the regions delimited from the groups of vertical lines. Note that even when there is only a single crane, the code detected more elements with the same pattern in vertical lines. Most of them are props (in the middle of the image b) and shoring (in the right).

is the presence of shoring and props that support slab formwork. As illustrated in Figure 14, 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 computation 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 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. In turn, their technique would be more appropriate to recognize objects with more complex geometries.

6. 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 taking advantage mainly of the objects' verticality to achieve a fast detection. In a preprocessing step, a downsampling method was proposed applicable to large-scale dense point clouds. Then a method that takes the raw points and finds the principal axes of the building is implemented. These axes enable the rotation of the point cloud and alignment with the XY axes.

Cranes and scaffold elements are then detected after efficiently filtering the point cloud vertical elements and transforming them into a 3D delineated representation. These features allow the search of high-level patterns characteristic in temporary elements on construction sites (such as cranes, scaffold, and shoring).

Once this pattern is found in vertical lines, specific features in its vertical cross-section reveal the final position of the target objects. Subsequently, The Manhattan Wold assumption in an aligned point cloud in conjunction with image processing techniques is leveraged to efficiently extract wall and formwork instances. Finally, the unique pattern in the depth of its vertical cross-section allows a deep learning classifier to distinguish between formwork and not-formwork elements.

Our main contributions are:

1. A detailed description of the geometry of the target objects (cranes, scaffold, and formwork) as they are defined by the corresponding regulations, norms, or manufacturers.
2. A method that uses domain knowledge to accurately, reliably, and understandably detect an extensive range of types of specific target objects in large-scale point clouds of construction sites. The method handles target objects with different dimensions (according to domain knowledge). Additionally, the technique is independent of the spatial

configuration of the target objects (for example, scaffolds do not need to be isolated or next to a facade).

3. An efficient technique to filter vertical objects in large-scale point clouds and classify them using deep learning techniques applied to vertical cross-sections. A similar approach can be leveraged to filter Manhattan-World wall instances (as demonstrated by Collins *et al.* (2021)) or to detect, for example, doors and windows in a point cloud of an interior of a facility; in a similar way as done by Quintana *et al.* (2018).

In conclusion, the authors argue that as long as there is a way to infer geometrical constraints on the target objects, it is possible to achieve outstanding 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 play a crucial role in detecting them. Like genetic algorithms, the successful implementation of such a method requires careful engineering of the objects' representation. In this case, it means a precise knowledge of the target objects' geometry. Such a technique would not apply to all objects. Nonetheless, the process is not limited to a few given examples or object size restrictions.

The domain knowledge compiled in this paper regarding the geometry of the target objects might be leveraged as input for the novel rule supported deep learning algorithms, such as Deep Neural Network with Controllable Rule Representations (DEEPCTRL) (Seo *et al.*, 2021) or Deep Learning Inspired Belief Rule-Based Expert System (BRB-DL) (Islam *et al.*, 2020). These algorithms have demonstrated to be more accurate, understandable and reliable than traditional deep learning algorithms which only learn from labeled data without considering domain knowledge.

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.

7. 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 detection of placed reinforcement would complete the primary set of nonpermanently-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 guard-rails, toe-boars, and working platforms.

Acknowledgements

We want to thank FARO Europe GmbH for funding this project and supporting and provisioning computing infrastructure and data essential to this publication.

REFERENCES

Acharya D, Ramezani M, Khoshelham K and Winter S (2019) Bim-tracker: A model-based visual tracking approach for indoor localisation using a 3d building model. *ISPRS Journal of Photogrammetry and Remote Sensing* **150**: 157–171, 10.1016/j.isprsjprs.2019.02.014.

Ahmed M, Seraj R and Islam SMS (2020) The k-means algorithm: A comprehensive survey and performance evaluation. *Electronics (Switzerland)* **9(8)**: 1–12, 10.3390/electronics9081295.

Álvares JS and Costa DB (2019) Construction Progress Monitoring Using Unmanned Aerial System and 4D {BIM}. In *Proc. 27th Annual Conference of the International Group for Lean Construction ({IGLC})*, International Group for Lean Construction, pp. 1445–1456, 10.24928/2019/0165.

Amer F and Golparvar-Fard M (2018) Decentralized Visual 3D Mapping of Scattered Work Locations for High-Frequency Tracking of Indoor Construction Activities. In *Construction Research Congress 2018*, American Society of Civil Engineers, pp. 491–500, 10.1061/9780784481264.048.

Armeni I, Sener O, Zamir AR, Jiang H, Brilakis I, Fischer M and 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, 10.1109/CVPR.2016.170.

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

Asadi K, Ramshankar H, Noghabaei M and Han K (2019) Real-time image localization and registration with bim using perspective alignment for indoor monitoring of construction. *Journal of Computing in Civil Engineering* **33(5)**: 04019031, 10.1061/(ASCE)CP.1943-5487.0000847.

Bosché F (2012) Plane-based registration of construction laser scans with 3D/4D building models. *Advanced Engineering Informatics* **26(1)**: 90–102, 10.1016/j.aei.2011.08.009.

Bosché F, Ahmed M, Turkan Y, Haas CT and 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, 10.1016/j.autcon.2014.05.014.

Bosché F and Haas CT (2008) Automated retrieval of 3D {CAD} model objects in construction range images. *Automation in Construction* **17(4)**: 499–512, 10.1016/j.autcon.2007.09.001.

Böttcher PDP and Neuenhagen H (1997) *Baustelleneinrichtung: betriebliche Organisation, Geräte, Kosten, Checklisten.* Bau-Verlag.

Braun A and Borrmann A (2019) Combining inverse photogrammetry and BIM for automated labeling of construction site images for machine learning. *Automation in Construction* **106**: 102879, 10.1016/j.autcon.2019.102879.

Braun A, Tuttas S, Borrmann A and Stilla U (2015) Automated progress monitoring based on photogrammetric point clouds and precedence relationship graphs. In *Proceedings of the 32nd International Symposium on Automation and Robotics in Construction and Mining (ISARC 2015)* (Malaska M and Heikkilä R, eds), International Association for Automation and Robotics in Construction (IAARC), Oulu, Finland, pp. 1–7, 10.22260/ISARC2015/0034.

Braun A, Tuttas S, Borrmann A and Stilla U (2020) Improving progress monitoring by fusing point clouds, semantic data and computer vision. *Automation in Construction* **116**: 103210, 10.1016/j.autcon.2020.103210.

Braun A, Tuttas S, Stilla U, Borrmann A and Center LO (2016) Incorporating knowledge on construction methods into automated progress monitoring techniques.

Buetti-Dinh A, Galli V, Bellenberg S, Ilie O, Herold M, Christel S, Boretska M, Pivkin IV, Wilmes P, Sand W, Vera M and Dopson M (2019) Deep neural networks outperform human expert's capacity in characterizing bioleaching bacterial biofilm composition. *Biotechnology Reports* **22**: e00321, https://doi.org/10.1016/j.btre.2019.e00321.

Collins F, Mafipour M, Noichl F, Pan Y and Vega M (2021) Towards applicable scan-to-bim and scan-to-floorplan: An end-to-end experiment. In *Proc. of the 32th Forum Bauinformatik*, Darmstadt, Germany.

Coughlan JM and Yuille AL (1999) Manhattan world: Compass direction from a single image by bayesian inference. In *Proceedings of the seventh IEEE international conference on computer vision*, vol. 2, IEEE, pp. 941–947.

Dai Z, Liu H, Le QV and Tan M (2021) Coatnet: Marrying convolution and attention for all data sizes. *arXiv preprint arXiv:2106.04803* .

Eickeler F, Sánchez-Rodríguez A and Borrmann A (2021) Adaptive feature-conserving compression for large scale point clouds. *Advanced Engineering Informatics* **48**: 101236.

Fichtner FW (2016) *Semantic enrichment of a point cloud based on an octree for multi-storey pathfinding*. Master's thesis, TU Delft, https://doi.org/10.1111/tgis.12308.

Fichtner FW, Diakité AA, Zlatanova S and Voûte R (2018) Semantic enrichment of octree structured point clouds for multi-story 3d pathfinding. *Transactions in GIS* **22(1)**: 233–248.

Golparvar-Fard M, Pena-Mora F and 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, pp. 249–256.

Golparvar-Fard M, Peña-Mora F and 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, 10.1061/(asce)cp.1943-5487.0000205.

Guo Y, Wang H, Hu Q, Liu H, Liu L and Bennamoun M (2019) Deep learning for 3D point clouds: A survey. *arXiv* : 1–2710.1109/tpami.2020.3005434, 1912.12033.

Han K, Degol J and Golparvar-Fard M (2018) Geometry- and Appearance-Based Reasoning of Construction Progress Monitoring. *Journal of Construction Engineering and Management* **144(2)**: 4017110, 10.1061/(asce)co.1943-7862.0001428.

Hu Q, Yang B, Xie L, Rosa S, Guo Y, Wang Z, Trigoni N and 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* : 11105–1111410.1109/CVPR42600.2020.01112, arXiv:1911.11236v3.

Islam RU, Hossain MS and Andersson K (2020) A deep learning inspired belief rule-based expert system. *IEEE Access* **8**: 190637–190651.

Jin Z and Gambatese J (2019) Bim for temporary structures: Development of a revit api plug-in for concrete formwork.

Kim C, Son H and Kim C (2013) Automated construction progress measurement using a 4D building information model and 3D data. *Automation in Construction* **31**: 75–82, 10.1016/j.autcon.2012.11.041.

Kim K and Cho Y (2015) Bim-based planning of temporary structures for construction safety. In *Computing in Civil Engineering 2015*, pp. 436–444.

Kim MK, Thedja JPP and 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, 10.1016/j.autcon.2020.103077.

Krizhevsky A, Sutskever I and Hinton GE (2012) Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems* **25**: 1097–1105.

Kropp C, Koch C and König M (2018) Interior construction state recognition with 4D {BIM} registered image sequences. *Automation in Construction* **86**: 11–32, 10.1016/j.autcon.2017.10.027.

Landrieu L and 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* : 4558–456710.1109/CVPR.2018.00479, 1711.09869.

Lin JJ and 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 and Li K (2019) Fast 3D Line Segment Detection From Unorganized Point Cloud. *arXiv:1901.02532* .

Maalek R, Lichti DD and Ruwanpura JY (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, 10.3390/rs11091102.

Mace B and Jones S (2016) How satisfied, really satisfied, are Owners?

Macher H, Landes T and Grussenmeyer P (2017) From Point Clouds to Building Information Models: 3D Semi-Automatic Reconstruction of Indoors of Existing Buildings. *Applied Sciences* **7(10)**: 1030, 10.3390/app7101030.

Masood MK, Aikala A, Seppaenen O and Singh V (2020) Multi-building extraction and alignment for as-built point clouds: A case study with crane cameras. In *Frontiers in Built Environment*.

Mukhopadhyay P and Chaudhuri BB (2015) A survey of Hough Transform. *Pattern Recognition* **48(3)**: 993–1010, 10.1016/j.patcog.2014.08.027.

Neuhausen M, Herbers P and König M (2020) Synthetic data for evaluating the visual tracking of construction workers. In *Construction Research Congress 2020: Computer Applications*, American Society of Civil Engineers Reston, VA, pp. 354–361.

Neuhausen M, Teizer J and König M (2018) Construction worker detection and tracking in bird's-eye view camera images. In *ISARC. Proceedings of the International Symposium on Automation and Robotics in Construction*, vol. 35, IAARC Publications, pp. 1–8.

Nikoohemat S, Diakité AA, Zlatanova S and Vosselman G (2020) Indoor 3D reconstruction from point clouds for optimal routing in complex buildings to support disaster management. *Automation in Construction* **113**: 103109, https://doi.org/10.1016/j.autcon.2020.103109.

Oesau S, Lafarge F and Alliez P (2014) Indoor scene reconstruction using feature sensitive primitive extraction and graph-cut. *ISPRS Journal of Photogrammetry and Remote Sensing* **90**: 68–82, 10.1016/j.isprsjprs.2014.02.004.

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

Pham KT, Vu DN, Hong PLH and Park C (2020) 4d-bim-based workspace planning for temporary safety facilities in construction smes. *International Journal of Environmental*

Research and Public Health **17(10)**, 10.3390/ijerph17103403.

Qi CR, Su H, Mo K and Guibas LJ (2017) Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 652–660.

Quintana B, Prieto SA, Adan A and Bosché F (2018) Door detection in 3d coloured point clouds of indoor environments. *Automation in Construction* **85**: 146–166.

Rodrigues F, Baptista JS and Pinto D (2021) Bim approach in construction safety–a case study .

Schach R and Otto J (2017) *Baustelleneinrichtung*. Springer Fachmedien Wiesbaden, 10.1007/978-3-658-16066-1.

Seo S, Arik S, Yoon J, Zhang X, Sohn K and Pfister T (2021) Controlling neural networks with rule representations. *Advances in Neural Information Processing Systems* **34**.

Son H, Kim C and Cho YK (2017) Automated Schedule Updates Using As-Built Data and a 4D Building Information Model. *Journal of Management in Engineering* **33(4)**: 4017012, 10.1061/(asce)me.1943-5479.0000528.

Turkan Y (2014) Tracking of secondary and temporary objects in structural concrete work. *Construction Innovation* 10.1108/ci-12-2012-0063.

Turkan Y, Bosche F, Haas CT and Haas R (2012) Automated progress tracking using 4D schedule and 3D sensing technologies. *Automation in Construction* **22**: 414–421, 10.1016/j.autcon.2011.10.003.

Turner E and Zakhor A (2014) Floor plan generation and room labeling of indoor environments from laser range data. In *Proceedings of the 9th International Conference on Computer Graphics Theory and Applications*, SCITEPRESS - Science and and Technology Publications, pp. 22–33, 10.5220/0004680300220033.

Wang Q (2019) Automatic checks from 3D point cloud data for safety regulation compliance for scaffold work platforms. *Automation in Construction* **104**: 38–51, 10.1016/j.autcon.2019.04.008.

Xu Y, Tuttas S, Hoegner L and 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, 10.1016/j.autcon.2017.09.014.

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

Zeng S, Chen J and Cho YK (2020) User exemplar-based building element retrieval from raw point clouds using deep point-level features. *Automation in Construction* **114**: 103159, https://doi.org/10.1016/j.autcon.2020.103159.