VBUILT: Volume-based Automatic Building Extraction for As-Built Point Clouds

M.K.Masood[a], A.Pushkar[b], O.Seppänen[a], V.Singh[a], and A.Aikala[a]

[a]Department of Civil Engineering, Aalto University, Finland
[b]Department of Civil and Environmental Engineering, Carnegie Mellon University, U.S.A
E-mail: mustafa.khalidmasood@aalto.fi, apushkar@andrew.cmu.edu, olli.seppanen@aalto.fi, vishal.singh@aalto.fi, antti.aikala@aalto.fi

Abstract –
Monitoring the progress of a large construction site manually is a challenging task for managers. By collecting visual data of the site, many monitoring tasks can be automated using machine vision techniques. In this work, we study a new method of collecting site data, which is through crane camera images used to create 3D point clouds. The technology is cost-effective and enables automatic capturing and transmission of on-site data. To automatically extract buildings from the as-built point clouds, we present VBUILT, which uses 3D convex hull volumes to identify building clusters. Experimental results on 40 point clouds collected over four months on a large construction site show that the proposed algorithm can identify building clusters with 100% accuracy.

Keywords –
Building extraction, crane cameras, convex hull, progress monitoring

1 Introduction

Machine vision is revolutionizing the construction industry by allowing real-time and autonomous inspection of construction projects to replace laborious and error-prone manual methods. We can now capture as-built information in the form of 3D point clouds, which can be used for construction progress monitoring [1]-[3], quality control [4], structural damage assessment [5]-[6], heavy equipment planning [7] and safety management [8]-[9].

Point clouds are typically acquired using light detection and ranging (LiDAR) due to their high accuracy. However, laser scanners are not suited to capturing large spaces due to the high number of scans required. In fact, the time, cost and labour needed for accurately capturing an entire construction site using a laser scanner would be prohibitive. Large areas are surveyed using Airborne LiDAR, which includes the significant expense of a piloted airplane carrying specialist laser scanning equipment [10]. Besides, Airborne LiDAR produces a distant view of the site that is not suited for close-range analysis.

An alternative is the use of photogrammetric point clouds. For a large construction site, these can be created from images captured using cameras mounted on Unmanned Aerial Vehicles (UAVs) (a.k.a. drones). But UAVs have a few limitations, such as the need for flight permissions [11], image quality deterioration due to vibrations caused by weather and the difficulties in camera pose estimation with the continuous position changes of the UAV [12].

In this paper, we study a promising new approach to capture as-built information of a construction site, which is the use of photogrammetric point clouds created from crane camera images. Crane cameras offer the advantage that they are already present on construction sites and no flight permissions are required as in the case of UAVs. Their installation and operation is low-cost and convenient.

The crane camera solution used in this work was developed by Pix4D [13]. The 3D point clouds are created automatically every day and stored on the cloud. In this work, we analyse data collected on a large construction site in Finland.

We focus specifically on the problem of automatically extracting buildings from the construction site point clouds. Removing the non-building elements from a multi-building point cloud can enhance alignment with the BIM model for progress measurement, but this is beyond the scope of this paper. Our work builds on the workflow reported in the literature for automatic building extraction from airborne LiDAR data.

Our contributions are as follows:

- To the best of our knowledge, we are the first to study the potential of point clouds generated from crane cameras for automated progress monitoring.
- This is also the first work that addresses the

problem of automatic building extraction on a multi-building construction site.

- We introduce the use of 3D convex hull volumes to identify buildings on the point cloud.

The rest of the paper is organized as follows: Section 2 discusses related work on automatic building extraction, crane cameras and convex 3D hulls. Section 3 describes the crane camera data collection. Section 4 describes the VBUILT algorithm. Section 5 presents the results and discussion. Section 6 lays out the conclusions and future work.

2 Related Work

An illustrative example of the automatic building extraction workflow for airborne laser scanning (ALS) point clouds was presented by Sampath and Shan [14]. They performed ground extraction using a slope-based 1D bi-directional filter. Then, they applied a region-growing algorithm to cluster individual buildings. They set the dimensions of the moving window to incrementally more than twice the point spacing, which in a LiDAR point cloud differs in the across and along scan directions. They also set a minimum point count threshold to identify non-building clusters. Wu et al. [15] fused LiDAR data with image data to extract the boundary lines of buildings. They used a triangulated irregular network (TIN) progressive densification filtering method to extract ground points. Widyaningrum et al. [16] used a generalization of the convex hull called the alpha-concave hull [17] to do two-dimensional boundary tracing.

An excellent review of building extraction techniques for ALS point clouds is presented in [18] which categorizes all the reported techniques into three categories: i) 2D building outline extraction; ii) 3D model reconstruction of buildings and iii) 3D roof contour extraction.

Our problem differs from the reported works in a number of ways. First, we intend to extract the actual as-built point cloud of a specific building, rather than to construct a 3D model from 2D segments. Secondly, a region-growing algorithm such as the one used in [14] discards non-building clusters based on point count. That is, from the points that remain after ground extraction, those clusters that have less than a certain number of points are discarded as being possibly vehicles or trees. This method is not suitable for construction sites, where non-building clusters (corresponding to large equipment, for example) may also have a high point count. Thirdly, the reported works do not perform building identification. Our problem requires that after extraction, the buildings be labeled. Fourthly, our data is a photogrammetric point cloud obtained from crane camera images.

Additionally, we choose to use a variation of RANdom SAmple Consensus (RANSAC) called M-estimator SAmple and Consensus (MSAC) for separating ground and non-ground points. This algorithm was proposed by [19], who showed that changing the cost function of RANSAC to score inliers on how well they fit the data improves the robust estimation with no additional computational burden.

A recent work [20] used crane camera images for updating the 3D crane workspace on a construction site. However, the crane camera images were used to modify a laser-scanned point cloud, which had to be manually captured. Point clouds generated automatically from the crane camera images were not considered.

Another recent work [21] used the area of 3D convex hulls to automatically map discontinuity persistence on rock masses. However, to the best of our knowledge, the use of 3D convex hull volumes for building extraction has not been reported.

3 Data collection and description

The data was collected on the site of the Tripla project located in Helsinki, Finland, which includes a shopping centre, hotel, housing and offices. The total area of the site is 183,000 floor square meters. Our focus in this work is on three buildings of the site, whose as-designed model along with the building labels is shown in Figure 1(a). Two independent cameras were mounted on the jib of a tower crane. Images were taken automatically when the crane would begin operation and then transferred to the Pix4D Cloud, where they were converted to 2D maps and 3D models.

The dataset consists of 40 as-built point clouds corresponding to weekdays spanning August 17th, 2018 to November 23rd, 2018. The as-built point cloud for day 40 is shown in Figure 1(b). Twenty point clouds contain building A only (due to B and C not being captured by the crane cameras), while twenty others contain multiple buildings as described in Table 1.

Table 1. Description of point cloud data

Available data	No. of point clouds
Single-building data (A only)	20
Multi-building data	
A,B only	1
A,C only	10
A, B and C	9

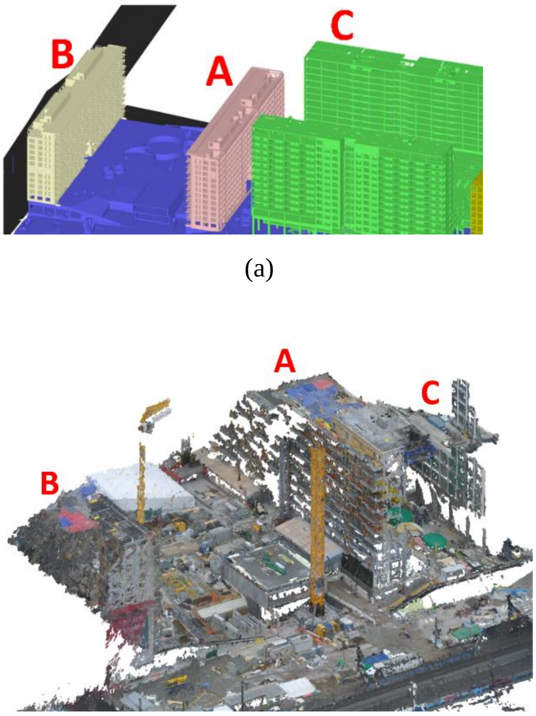
By default, the point clouds are georeferenced, with an average point density of the order of 1000 points/m^3, except for point cloud 40, which has an average point density of 37,233 points/m^3. However, for 13 point clouds between August 29th, 2018 and September 17th, 2018, the image geotags were not detected and thus the output coordinate system was arbitrarily set. Therefore, the point cloud could not be scaled and the resulting point density is of the order of 3 points/m^3. We subsequently refer to these point clouds as 'non-georeferenced'.

Figure 2(a) shows the front-view of Building A. The holes in the data are due to the portion of the building being out of the field-of-view of the camera. Figure 2(b) shows the top view of the building, which shows the formwork being laid. This is the clearest data of the building available.

4 Building extraction using VBUILT

In this section, we explain the VBUILT algorithm. Our objective is to extract the 3D point cloud of a specific building on the site to be used for further analysis, such as construction progress tracking.

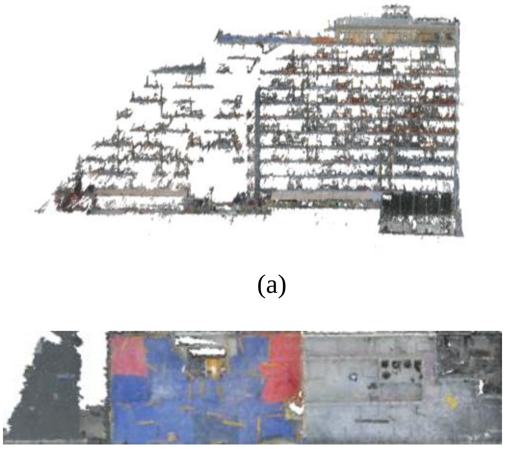
Our solution is to extract the ground plane using MSAC, then cluster the non-ground point cloud based on Euclidean distance, followed by an analysis of the 3D convex hull volumes of each cluster.



(a)

(b)

Figure 1. (a) As-designed model of the Tripla site with building labels A, B and C; (b) As-built point cloud of the Tripla site for day 40 (November 23rd, 2018)



(a)

(b)

Figure 2. Manually extracted point cloud of building A: (a) front-view and (b) top-view

4.1 MSAC-based ground extraction

Let the set of construction site point clouds be $S=\{S_1, S_2, \ldots S_D\}$, where $D=40$ and each point cloud $S_i=\{s_1, s_2, \ldots s_N\}$, $S_i \in \mathrm{R}^3$. First, we downsample the point cloud for faster processing using grid average downsampling, which merges points within a 3D grid by averaging their locations and colours.

Next, we need to separate the ground and non-ground points. For this, we implement the MSAC algorithm using the in-built MATLAB function *pcfitplane*, which we explain subsequently. We specify three parameters for the MSAC algorithm:

- the reference vector ($\mathbf{n_r}$) (the *orientation constraint*).
- the maximum absolute angular distance (denoted as $\theta_{\max}$) between the normal vector of the plane hypothesis ($\mathbf{n_h}$) and the reference vector.
- the threshold $d_{\max}$, which is the maximum distance between an inlier point and the fitted plane.

For ground plane extraction, $\mathbf{n_r}$ should be set normal to the xy plane. Then, the algorithm checks if the angular distance between $\mathbf{n_r}$ and $\mathbf{n_h}$ (denoted as θ) is less than $\theta_{\max}$. If not, the plane hypothesis is discarded and a new one is considered. Otherwise, the MSAC algorithm proceeds to calculate the following cost function:

$$C^j = \sum_{i=1}^{N} \rho_i \tag{1}$$

where j is the iteration number and the robust error term ρ is defined as follows:

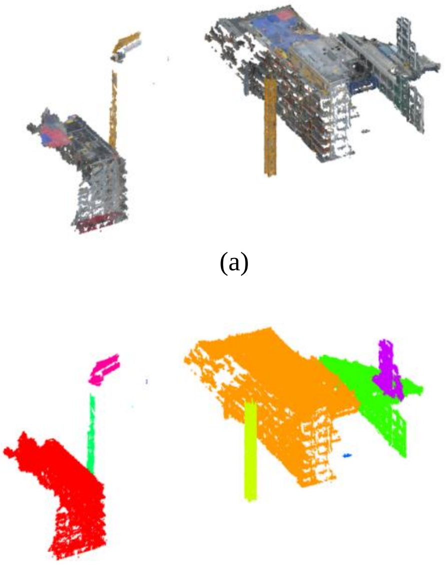
$$\rho = \begin{cases} |d| & if\ |d| < d_{max} \\ d_{max} & if\ |d| \geq d_{max} \end{cases} \tag{2}$$

If C_{j+1} is less than C_j, the $(j+1)_{th}$ model is taken as the best model found so far. Every time the best model is updated, the maximum number of iterations is set to k, which is defined as follows:

$$k = \frac{log(1-p)}{log(1-w^n)} \tag{3}$$

where p is the desired confidence level (set to 0.99 as per standard practice) that at least one point in the plane hypothesis is an inlier, w is the probability that a single point is an inlier and n is the number of points needed to define the model (which for a plane, as in our case, is 3).

For the georeferenced point clouds, θ_{max} and d_{max} were selected as 5° and 5 m, respectively, based on the intuition that the ground plane is only slightly inclined with respect to the xy plane. For the non-georeferenced point clouds, due to the smaller point cloud density, d_{max} was set to a higher value (40).


(a)

(b)

Figure 3. (a) Ground removed using MSAC; (b) Euclidean-distance based clusters

Note that the ground in the as-built point cloud contains many more points than the top floors of the buildings, thus yielding more inliers and a smaller cost (see Equation (1)). This is why the top floors are not mistaken for the ground by the algorithm.

Once the ground plane is identified, we remove the portion of the point cloud below it by checking the dot product of each point with the plane. If the dot product is negative, the point is below the plane. In other words, for a point s_i=(x_i, y_i, z_i) and the ground plane parameters a_g, b_g, c_g and d_g, if $a_g x_i + b_g y_i + c_g z_i + d_g < 0$, then s_i is a below-ground point and is removed. Finally, the ground plane is removed from the point cloud. Figure 3(a) illustrates the ground-removal for Day 40.

4.2 Euclidean distance-based clustering

With the ground and below-ground points removed, the task now is to cluster the point cloud. We do this using a Euclidean distance criterion. For every point s_i=(x_i, y_i, z_i) and s_j=(x_j, y_j, z_j), the Euclidean distance between the points is as follows:

$$d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2} \tag{4}$$

If d_{ij} is less than a distance threshold δ, then s_i and s_j are in the same cluster. We set δ =1 (and δ =5 for the non-georeferenced point clouds) as a reasonable estimate of the separation between intracluster and extracluster points. We refer to the clusters as the set C^d. The results of the clustering are shown in Figure 3(b).

4.3 Building identification using convex hull volumes

A convex hull of a set of points is the smallest convex set that contains all the points. In this paper, we describe the creation of a 3D convex hull based on the Quickhull algorithm, upon which the MATLAB function *convhull* is based. Note that *convhull* uses a more robust (albeit proprietary) strategy for coping with imprecision than the approach outlined in Section 4 of the original Quickhull paper [22], but otherwise the implementation is standard.

Starting with a 3D set of points, Quickhull first builds an initial hull from four points. Ideally, this tetrahedron should be maximal, which can be accomplished by first finding the maximum and minimum points along the X, Y and Z axes, then finding the furthest point from the line segment made by the first two points, then finding the furthest point from the plane made by the first three points. Once the initial hull is created, the point from the remaining set (not used in the initial hull) which is farthest from the hull is set as the *eye point*. Those faces of the hull that lie above the eye point (as determined by a simple plane test) are called visible faces, while those that lie below it are non-visible faces. The edges that connect visible with non-visible faces together constitute the horizon. To determine the horizon, a flood-fill approach is used. The algorithm starts from a visible face, and then travels across an edge to a neighbouring face. If the neighbouring face is also visible, the algorithm travels further across an edge until a non-visible face is encountered, at which point the offending edge is marked as being part of the horizon. Then, the algorithm returns to the last visible face and continues traveling across another edge. This process continues until the algorithm returns to the original visible face. Once all the horizon edges are found, the eye point is connected to it and thus the hull is extended. This process continues till no face remains with a point outside it.

Our use of convex hull volumes is based on the intuition that buildings are typically more voluminous than non-buildings on a construction site, especially a large site such as the one we are considering. As we shall demonstrate, this intuition is supported by our dataset. For example, Table 2 shows the convex hull volumes for C^{40}.

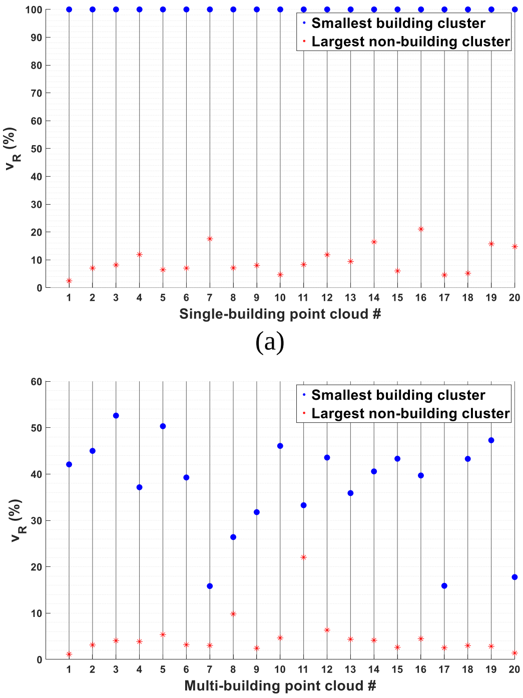
Table 2. Convex hull volumes of clusters

Cluster #	Convex hull Volume	Cluster-type (Manually labeled)
1	749.2316	Building
2	318.9671	Building
3	169.1460	Building
4	10.4517	Non-building
5	7.2664	Non-building
6	3.9378	Non-building
7	3.7009	Non-building
8	0.1131	Non-building
9	0.1000	Non-building
10	0.1000	Non-building

Note that the building clusters are much larger than the non-building clusters. This motivates us to define a quantity called the *relative volume* (v_R) as follows:

$$(v_R{}^i)_d = \frac{(v_{abs}{}^i)_d}{(v_{max})_d} * 100 \qquad (5)$$

where $v_{abs}{}^i$ is the convex hull volume for cluster i and v_{max} is the largest convex hull volume for the set of clusters under consideration (i.e. for day d). Figure 4 shows the smallest building cluster and the largest non-building cluster for each day based on the v_R values of the clusters. The figure shows a considerable separation between building and non-building clusters. As Figure 4(a) shows, for the single-building data, the building cluster is always the largest and maintains a significantly higher v_R value than the largest non-building cluster for each day. In fact, the mean of the relative volumes of the largest non-building clusters is 9.68%. For the case of multi-building point clouds (Figure 4(b)), the mean of the relative volumes of the largest non-building clusters is 4.72%, while that of the smallest building cluster is 37.35%.



(a)

(b)

Figure 4. Comparison of smallest building cluster and largest non-building cluster for (a) single-building point clouds and (b) multi-building point clouds

Based on these insights, we outline the following strategy for extracting the building clusters:

- For the single-building data, the cluster with the largest v_R value is the building.
- For the multi-building data, a cluster with v_R less than 15% is a non-building cluster. Additionally, we impose the maximum possible buildings on the site (3 in our case) as a constraint.

The algorithm finally divides the point cloud into two sets, *BuildingClusters* and *NonBuildingClusters*. The results are presented in the next section.

In order to label the extracted buildings, we consider the relative v_R values of the three buildings, but this approach is not fully demonstrable due to some limitations in our dataset, as is discussed in the next section.

5 Results and discussion

5.1 Classification: building vs non-building clusters

We define the classification accuracy as being 100% for a particular day if the set *BuildingClusters* contains the actual buildings available on the original construction site point cloud. The achieved accuracy is 100% for all 40 point clouds, i.e. building and non-building clusters are successfully separated for the entire dataset. Figure 5 shows the building-only point cloud obtained for day 40.

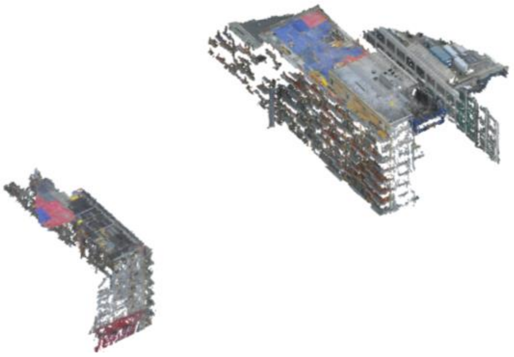


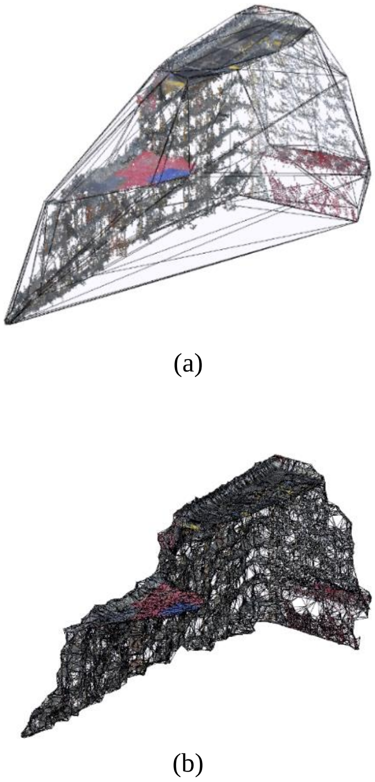
Figure 5. Building-only point cloud for day 40

5.2 Building labeling

In our dataset, building A always has the largest v_R value and it can therefore be labeled with 100% accuracy for the entire dataset. However, only a small segment of buildings B and C is captured by the crane cameras. The size of these segments varies inconsistently due to the varying positions of the crane cameras. Thus, a volume comparison cannot be consistently applied to these buildings until they are captured more completely. In other words, more complete data is required to label these buildings according to their relative v_R values.

5.3 Departure from convexity: effect of the *α-radius*

A convex hull is a specific case of an *alpha shape*, which is a polytope with a parameter called the *α-radius* that defines how tightly the boundary fits around the shape [23]. A convex hull results when the α-radius is infinity. In our specific case, using the convex hulls was more suitable due to the nature of a convex hull to “wrap around” the object, thus ignoring false cavities in the point cloud clusters resulting from the crane camera not capturing a portion of the site. An alpha shape, on the other hand, tends to “wrap into” the cavities of the point cloud clusters, resulting in a smaller boundary volume that does not approximate the actual volume of the building well. This is illustrated in Figure 6. We suggest this phenomenon might generalize to any point cloud with missing regions being subjected to boundary volume analysis. Additionally, using convex hull volumes obviates the need to find a suitable alpha radius, as was done in [24] for example.



(a)

(b)

Figure 6. Illustration of (a) convex hull (*α-radius*=∞) and (b) alpha shape with lower radius

5.4 Crane cluster outlier

Figure 4(b) shows that for multi-building point cloud

11, the largest non-building cluster has a v_R value of 22%, which is a clear outlier. Visualizing this cluster shows that the large volume comes from the presence of a crane in the cluster, as is shown in Figure 7. When we manually remove this crane from the cluster, the v_R value of the cluster reduces to 4.44%. Note that the maximum building number constraint we imposed deals with this outlier effectively, which is why this cluster is correctly classified as a non-building cluster despite being above the 15% v_R threshold.

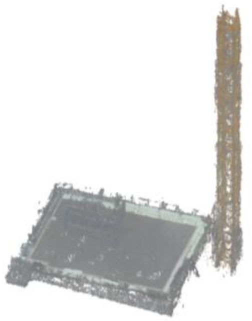


Figure 7. Outlier cluster for multi-building point cloud 11 (day 29/40) with v_R=22%

6 Conclusions and future work

Crane cameras are a convenient and cost-effective alternative to laser scanners and UAVs for visually capturing an entire construction site. In this work, we did a preliminary study on the potential of crane camera point clouds for automatic progress monitoring. We addressed the problem of automatically extracting buildings from the crane camera point clouds. We presented an algorithm called VBUILT which is based on the intuition that building clusters have larger 3D convex hull volumes than non-building clusters. We found that our dataset supports this intuition, showing a large separation between the relative volumes of building and non-building clusters. By utilizing this separation, VBUILT was able to correctly identify the building clusters for all 40 point clouds in the dataset. We also successfully labeled one building which is captured relatively well by the crane cameras, but would require better data of the other buildings to demonstrate the effectiveness of the relative volume strategy for multi-building labeling.

An important insight from our work was that convex hull volumes capture the volumes of building clusters with missing regions more effectively than alpha shapes with lower radii due to the tendency of the latter to wrap into the empty regions of the cloud. We thus recommend the consideration of the convex hull over alpha shapes for such volume-based analyses of incomplete point clouds.

The algorithm successfully dealt with an outlier non-building cluster by imposing a maximum building number constraint. However, in order to prevent such outliers altogether, an automatic crane removal algorithm can be developed in future work, possibly by leveraging on the high slenderness ratio and the uniform colour of the cranes.

The proposed algorithm worked successfully with both georeferenced and non-georeferenced point clouds, the latter constituting about 32% of our dataset. In future work, we will study why georeferencing fails for some point clouds and how we can address the problem.

The point clouds generated from crane cameras give a good overview of the site and high quality views of the top portions of the buildings where the laying of formwork and rebar can be observed. Thus, in future work, we intend to use the building point clouds obtained through VBUILT to infer the progress of formwork and rebar. Also, the crane camera point clouds contain holes in the building façade regions. More studies are needed to truly gauge the effectiveness of this portion of the point clouds for progress monitoring. Therefore, in the future, we intend to match the building-only point cloud with the as-designed Building Information Model (BIM) of the site to study the extent to which progress and deviations can be measured.

Acknowledgements

We thank YIT for providing the data for this study. The work was supported by the Reality Capture research project funded by Business Finland, Aalto University and a consortium of 5 companies.

References

[1] F. Bosché, M. Ahmed, Y. Turkan, C. T. Haas and R. Haas. 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*, vol. 49, pp. 201-213, 1 2015.

[2] C. Kim, H. Son and C. Kim. Automated construction progress measurement using a 4D building information model and 3D data. *Automation in Construction*, vol. 31, pp. 75-82, 5 2013.

[3] Y. Turkan, F. Bosche, C. T. Haas and R. Haas. Automated progress tracking using 4D schedule and 3D sensing technologies. *Automation in*

Construction, vol. 22, pp. 414-421, 3 2012.

[4] B. Akinci, F. Boukamp, C. Gordon, D. Huber, C. Lyons and K. Park. A formalism for utilization of sensor systems and integrated project models for active construction quality control. *Automation in construction,* vol. 15, pp. 124-138, 2006.

[5] J. Fernandez Galarreta, N. Kerle and M. Gerke. UAV-based urban structural damage assessment using object-based image analysis and semantic reasoning. *Natural hazards and earth system sciences,* vol. 15, pp. 1087-1101, 2015.

[6] C. Koch, S. G. Paal, A. Rashidi, Z. Zhu, M. König and I. Brilakis. Achievements and challenges in machine vision-based inspection of large concrete structures. *Advances in Structural Engineering,* vol. 17, pp. 303-318, 2014.

[7] D. Moon, S. Chung, S. Kwon, J. Seo and J. Shin. Comparison and utilization of point cloud generated from photogrammetry and laser scanning: 3D world model for smart heavy equipment planning. *Automation in Construction,* vol. 98, pp. 322-331, 2 2019.

[8] J. Seo, S. Han, S. Lee and H. Kim. Computer vision techniques for construction safety and health monitoring. *Advanced Engineering Informatics,* vol. 29, pp. 239-251, 4 2015.

[9] H. Kim, K. Kim and H. Kim. Vision-Based Object-Centric Safety Assessment Using Fuzzy Inference: Monitoring Struck-By Accidents with Moving Objects. *Journal of Computing in Civil Engineering,* vol. 30, p. 04015075, 7 2016.

[10] K. Johnson, E. Nissen, S. Saripalli, J. R. Arrowsmith, P. McGarey, K. Scharer, P. Williams and K. Blisniuk. Rapid mapping of ultrafine fault zone topography with structure from motion. *Geosphere,* vol. 10, pp. 969-986, 2014.

[11] S. Cardot. Crane Camera Site Surveying. *Gim International-the worldwide magazine for Geomatics,* vol. 31, pp. 31-33, 2017.

[12] S. Bang, H. Kim and H. Kim. UAV-based automatic generation of high-resolution panorama at a construction site with a focus on preprocessing for image stitching. *Automation in Construction,* vol. 84, pp. 70-80, 12 2017.

[13] [Anonymous]. Pix4D Launches Crane Camera Solution for Construction Industry. *Gim International-the worldwide magazine for Geomatics,* vol. 30, p. 7, 2016.

[14] A. Sampath and J. Shan. Building boundary tracing and regularization from airborne LiDAR point clouds. *Photogrammetric Engineering & Remote Sensing,* vol. 73, pp. 805-812, 2007.

[15] J. Wu, S. Jie, W. Yao and U. Stilla. Building boundary improvement for true orthophoto generation by fusing airborne LiDAR data. *Joint Urban Remote Sensing Event*, 2011.

[16] E. Widyaningrum, R. C. Lindenbergh, B. G. H. Gorte and K. Zhou. Extraction of building roof edges from lidar data to optimize the digital surface model for true orthophoto generation. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences,* vols. XLII-2, pp. 1199-1205, 5 2018.

[17] S. Asaeedi, F. Didehvar and A. Mohades. Concave hull, a generalization of convex hull. *Theoretical Computer Science,* vol. 702, pp. 48-59, 11 2017.

[18] I. Tomljenovic, B. Höfle, D. Tiede and T. Blaschke. Building Extraction from Airborne Laser Scanning Data: An Analysis of the State of the Art. *Remote Sensing,* vol. 7, pp. 3826-3862, 3 2015.

[19] P. H. S. Torr and A. Zisserman. MLESAC: A New Robust Estimator with Application to Estimating Image Geometry. *Computer Vision and Image Understanding,* vol. 78, pp. 138-156, 4 2000.

[20] J. Chen, Y. Fang and Y. K. Cho. Real-Time 3D Crane Workspace Update Using a Hybrid Visualization Approach. *Journal of Computing in Civil Engineering,* vol. 31, p. 04017049, 9 2017.

[21] A. Riquelme, R. Tomás, M. Cano, J. L. Pastor and A. Abellán. Automatic Mapping of Discontinuity Persistence on Rock Masses Using 3D Point Clouds. *Rock Mechanics and Rock Engineering,* vol. 51, pp. 3005-3028, 5 2018.

[22] C. B. Barber, D. P. Dobkin and H. Huhdanpaa, .The quickhull algorithm for convex hulls. *ACM Transactions on Mathematical Software,* vol. 22, pp. 469-483, 12 1996.

[23] H. Edelsbrunner and E. P. Mücke. Three-dimensional alpha shapes. *ACM Transactions on Graphics (TOG),* vol. 13, pp. 43-72, 1994.

[24] R. C. Santos, M. Galo and A. C. Carrilho, Building boundary extraction from lidar data using a local estimated parameter for alpha shape algorithm. *ISPRS-International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences,* vols. XLII-1, pp. 127-132, 9 2018.