Automatic point Cloud Building Envelope Segmentation (Auto-CuBES) using Machine Learning

Bryan P. Maldonado, Nolan W. Hayes, Diana Hun

Buildings and Transportation Science Division, Oak Ridge National Laboratory*, United States of America

maldonadopbp@ornl.gov, hayesnw@ornl.gov, hunde@ornl.gov

Abstract -

Modern retrofit construction practices use 3D point cloud data of the building envelope to obtain the as-built dimensions. However, manual segmentation by a trained professional is required to identify and measure window openings, door openings, and other architectural features, making the use of 3D point clouds labor-intensive. In this study, the Automatic point Cloud Building Envelope Segmentation (Auto-CuBES) algorithm is described, which can significantly reduce the time spent during point cloud segmentation. The Auto-CuBES algorithm inputs a 3D point cloud generated by commonly available surveying equipment and outputs a wireframe model of the building envelope. Unsupervised machine learning methods were used to identify facades, windows, and doors while minimizing the number of calibration parameters. Additionally, Auto-CuBES generates a heat map of each facade indicating non-planar characteristics that are crucial for the optimization of connections used in overclad envelope retrofits. With a scan resolution of 3 mm**, the resulting window dimensions showed a mean absolute error of** 4.2 mm **compared to manual laser measurements.**

Keywords -

point cloud, segmentation, unsupervised learning

1 Introduction

Buildings are responsible for 30% of the total carbon dioxide emissions in the US [1]. To mitigate the impact of building operations on climate change, the application of energy codes in construction practices has reduced the energy use in buildings by more than 40% since their insertion in the 1980s. However, about 52% of residential and 46% of commercial buildings were built before energy codes [2]. Hence, large energy savings can be achieved by retrofitting older buildings and bringing them up to code.

Overclad envelope retrofits are an attractive solution since they minimize occupant disruption and shorten construction time at the job site. However, they require precise measurements of the existing envelope to adequately design and manufacture the retrofit panels. Current state-of-the-art retrofit panel design and sizing consists of three steps: 1) 3D point cloud data generation of the building envelope using commonly available surveying equipment, 2) manual segmentation of 3D point cloud data by a trained professional to identify and dimension window openings, door openings, wall protrusions, and other non-planar architectural features, and 3) modular panel layout optimization and sizing by an architect or engineer [3, 4]. The process of manually segmenting the point cloud data can be difficult and costly, often requiring third-party software and a trained professional to spend several man-hour-weeks depending on the size of the existing building. Additionally, after segmentation of the point cloud into different components of the envelope, the position and size of each component (window, door, etc.) must be extracted from the point cloud which often includes human-introduced errors due to the difficulty and tediousness of the process. Although commercially available software has been optimized to handle point clouds for manual segmentation, automated feature identification and measurement extraction are still challenging. Thus, the automation of these processes could save a significant amount of time and money while also minimizing errors.

Recent advances in machine learning have enabled the development of automatic segmentation algorithms for extracting building envelope features. Common practices include the use of photogrammetry (RGB cameras) data [5] or a combination of photogrammetry and light detection and ranging (LiDAR) data [6]. Although such algorithms can identify the locations and dimensions of windows and doors, they are limited to the resolution of the camera (~10s of mm) and might not achieve the millimetric accuracy needed for retrofit panel design. Deep learning techniques that analyze LiDAR data for point cloud segmentation are capable of automatically identifying the constituent components of common objects [7]. However, this requires a large amount of correctly labeled points for training neural networks. For the segmentation of large structures, the scarcity of training samples and inaccurate boundary segmentation have limited the scope of their usage [8]. For building envelope segmentation specifi-

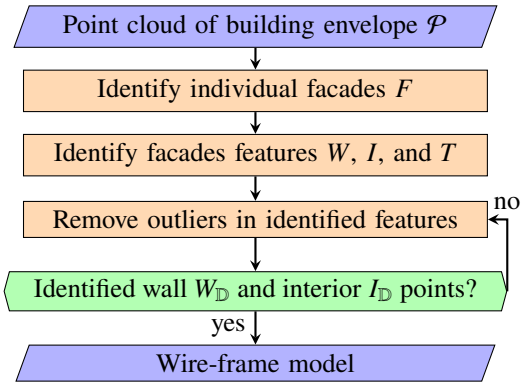

Figure 1. Flowchart of Auto-CuBES algorithm.

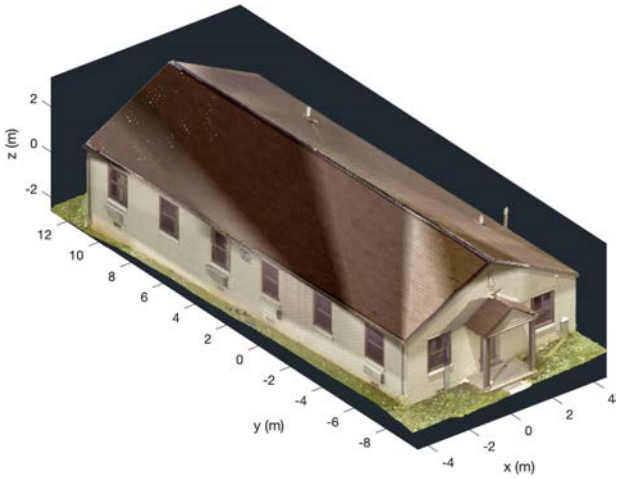

Figure 2. Point cloud $\mathcal{P}$ of a residential building.

cally, such a training set could require additional labor time and cost. Moreover, the variety and uniqueness of facade topologies exacerbate the real-world training sample scarcity problem. Unsupervised machine learning methods that do not need training data present a solution for the building envelope segmentation task. Even though unsupervised methods require the explicit definition of sequential tasks and task-dependent calibration parameters, they have been successful at generating architectural floor plans from point clouds of building interiors [9].

In this study, we introduce the Automatic point Cloud Building Envelope Segmentation (Auto-CuBES) algorithm based on unsupervised machine learning that can automatically label 3D point cloud data and reduce the time spent in manual segmentation. The algorithm can process high-resolution point clouds and generate a wire-frame building envelope model with a small set of calibration parameters. The remaining sections are as follows. Section 2 describes in detail each component of the Auto-CuBES algorithm. Section 3 shows the algorithm results. Finally, Section 4 presents conclusions and future research.

2 Auto-CuBES algorithm

Figure 1 shows the flowchart of the Auto-CuBES algorithm starting from the input point cloud file to the output wire-frame model. In contrast to deep learning methods where layers of neurons are placed between inputs and outputs, unsupervised methods are typically divided into sequential tasks. Each task of the Auto-CuBES algorithm plays an important role in segmentation and outlier removal. Although each task has its own set of calibration parameters, empirical observations suggest that most of them do not need to be recalibrated for different input files. However, the decision step in Fig. 1 might require some recalibration based on the level of accuracy defined by the user. The following subsections describe the unsupervised machine learning methods, calibration parameters, and preliminary results of each task.

2.1 Input point cloud file structure

The Leica Nova MS60 MultiStation was used to collect point cloud data with 7-dimensional elements of the form:

$$\mathcal{P} = \begin{bmatrix} P & L & C \end{bmatrix} = \begin{bmatrix} x & y & z & L & r & g & b \end{bmatrix} \quad (1)$$

where $P = \begin{bmatrix} x & y & z \end{bmatrix}$ are the rectangular coordinates, L is the LiDAR intensity recorded as the return strength of a laser beam, and $C = \begin{bmatrix} r & g & b \end{bmatrix}$ are the corresponding pixel colors obtained from the camera. A duplex residential house was scanned by placing the LiDAR scanner in front of each facade and stitching the scans together using four control points located near the corners of the building. Figure 2 shows the resulting point cloud used in this study. The point cloud encompasses a total of 32.2 million points with an average distance of 3 mm between points.

2.2 Identify and extract individual facades

The algorithm starts by identifying the four facades of the building and removing the roof points. To achieve this, and assuming that the LiDAR scanner was leveled, the point cloud was rotated about the z-axis to align the walls with the x, y canonical basis. To determine the angle of rotation, a plan view of the envelope was generated by selecting a subset of points corresponding to a section passing through the center of the building's height. In other words, the plan view of the envelope is the set:

$$E = \left\{ \begin{bmatrix} x & y & z \end{bmatrix} \in \mathcal{P} \,\middle|\, |z - \mathbb{E}[z]| < \delta \right\}. \quad (2)$$

Here, $\mathbb{E}[\cdot]$ is the expected value function and $\delta = 0.4$ m is a calibration parameter. The rotation angle was calculated by solving the minimum-volume oriented bounding box problem using the method described in Chang *et al.* [10].

The top-left plot of Fig. 3 shows the resulting plan view of the building envelope. By aligning the envelope E with

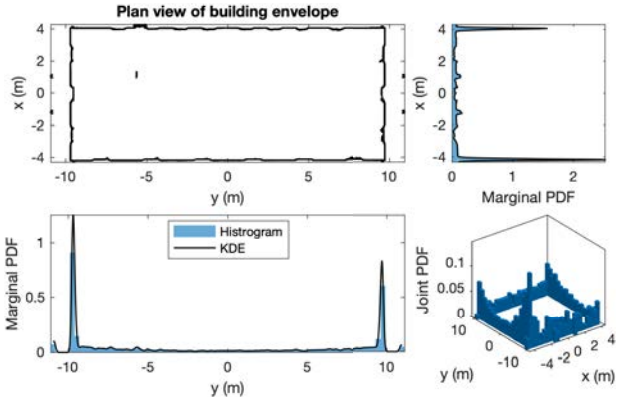

Figure 3. Plan view of building envelope E with joint and marginal PDFs of the x and y components.

the x- and y-axis, the individual facades can be identified by looking at the distribution of points over each axis. The histograms in Fig. 3 correspond to the joint and marginal distributions of $\begin{bmatrix} x & y \end{bmatrix} \subset E$. The marginal probability density functions (PDFs) $\text{pdf}(x)$, and $\text{pdf}(y)$ where approximated using kernel density estimators (KDEs). The Gaussian kernel was used in the KDE and the bandwidth was selected following Silverman's rule of thumb [11].

Note that the marginal PDFs have two main modes, each of which corresponds to one of the four facades of the building envelope. Let $\bar{x}_{>0} = \arg\max_{x>0} \text{pdf}(x)$ be the largest mode for the positive values in the x-axis. Similarly, define $\bar{x}_{<0}$ for the negative x-axis, and $\bar{y}_{>0}, \bar{y}_{<0}$ for the y-axis. Although each mode defines the average location of a facade, the facade thickness was obtained by selecting the interval between the inflection points around each mode. Let $\bar{x}^{\text{L}}_{>0} = \arg\max_{x>0} \frac{d}{dx}\text{pdf}(x)$ and $\bar{x}^{\text{R}}_{>0} = \arg\min_{x>0} \frac{d}{dx}\text{pdf}(x)$ be the inflection points around $\bar{x}_{>0}$. Note that the inflections points bound the corresponding mode on the left and right as follows: $\bar{x}^{\text{L}}_{>0} < \bar{x}_{>0} < \bar{x}^{\text{R}}_{>0}$. Using similar definitions for the remaining modes, the four facades of the building envelope can be extracted as:

$$F_x^{y<0} = \{p \in \mathcal{P} \mid \bar{x}^{\text{R}}_{<0} \leq x \leq \bar{x}^{\text{L}}_{>0}, \bar{y}^{\text{L}}_{<0} \leq y \leq \bar{y}^{\text{R}}_{<0}\} \quad (3a)$$

$$F_x^{y>0} = \{p \in \mathcal{P} \mid \bar{x}^{\text{R}}_{<0} \leq x \leq \bar{x}^{\text{L}}_{>0}, \bar{y}^{\text{L}}_{>0} \leq y \leq \bar{y}^{\text{R}}_{>0}\} \quad (3b)$$

$$F_y^{x<0} = \{p \in \mathcal{P} \mid \bar{x}^{\text{L}}_{<0} \leq x \leq \bar{x}^{\text{R}}_{<0}, \bar{y}^{\text{R}}_{<0} \leq y \leq \bar{y}^{\text{L}}_{>0}\} \quad (3c)$$

$$F_y^{x>0} = \{p \in \mathcal{P} \mid \bar{x}^{\text{L}}_{>0} \leq x \leq \bar{x}^{\text{R}}_{>0}, \bar{y}^{\text{R}}_{<0} \leq y \leq \bar{y}^{\text{L}}_{>0}\} \quad (3d)$$

Figure 4 shows the automatically identified facades. Note that the procedure described here is intended for a traditional envelope with four perpendicular facades. A more complex geometry will require a modified approach.

2.3 Identify and extract facade features

Each facade was individually analyzed to extract its main features (doors and windows). Based on traditional

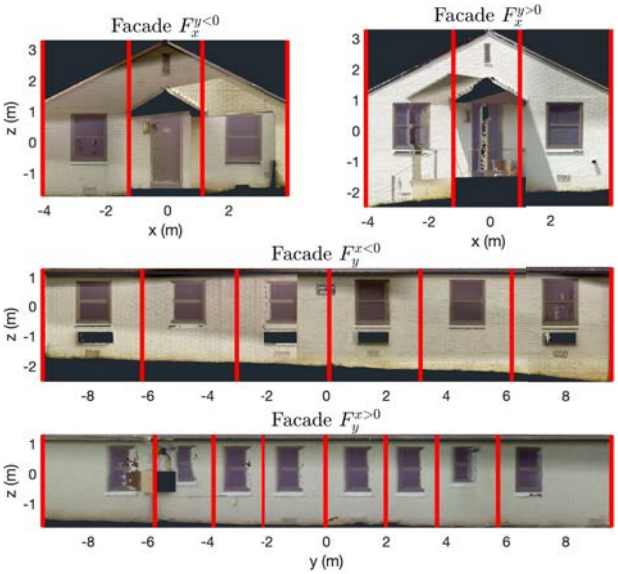

Figure 4. Four perpendicular facades from building envelope sectioned according to identified features.

residential construction practices, it was assumed that the window and door frames are recessed with respect to the exterior wall surface. Therefore, and thanks to the high resolution of the point cloud data, the facade thickness can be sliced into three main subsets: 1) interior points corresponding to windows and doors, 2) wall points, and 3) exterior points corresponding to features such as roof overhangs and window sills. In order to slice the facade, the point cloud distribution with respect to the thickness axis was considered. However, this method assumes that the facade is flat, which is not always the case. To remove flatness assumptions, the facade was decomposed into different sections depending on the number of features in the facade. The unsupervised k-means clustering algorithm was used to segment each facade into k sections [12]. Here, k is a parameter given by the user. Figure 4 shows how each facade can be automatically divided into sections corresponding to the number of features.

The algorithm assumes that each section is locally flat, even though the combined facade is not. Therefore, for each section within a facade, interior, wall, and exterior features were identified using the point cloud distribution of the thickness axis. As an example, consider dividing the first facade in Fig. 4 into the sections $\{F^{y<0}_{x,k}\}_{k\in\{1,2,3\}}$. Under the locally-flatness assumption, the facade section will be align with a plane of the form $y = a_k x + b_k z + c_k$. Moreover, consider the following coordinate transformation to align each section with the xz-plane:

$$y^k_{<0} = y - (a_k x + b_k z + c_k) + \mathbb{E}[y]. \quad (4)$$

The parameters of the linear function were found using

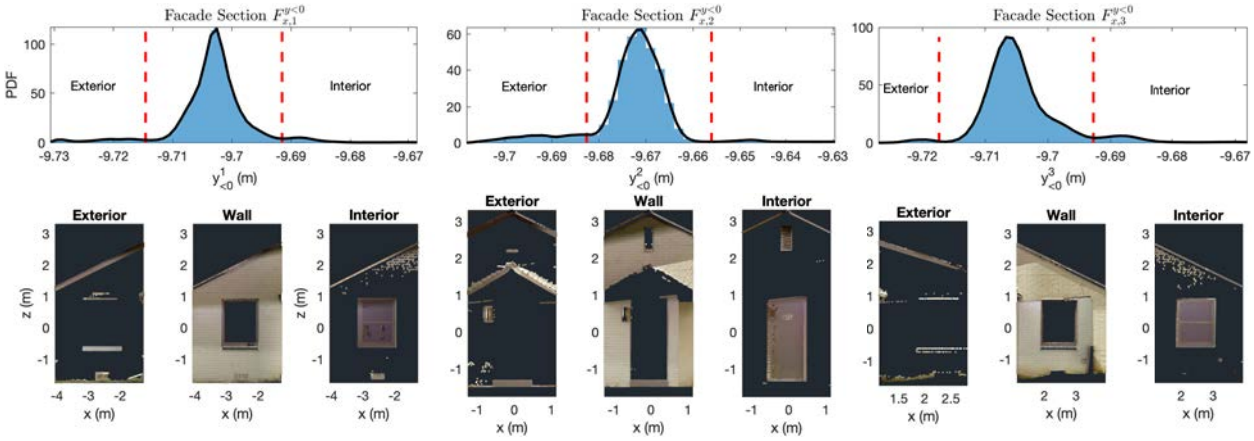

Figure 5. Segmentation of the facade $F_x^{y<0}$ into exterior, wall, and interior points by analysing individual sections.

robust least squares with bisquare weights due to the existence of outliers [11]. The top row of Fig. 5 shows the $\text{pdf}(y_{<0}^k)$ for each section. Although the resulting PDF is not unimodal, there is a clear dominant mode where the wall points are located. Let $\widehat{y}_{<0}^k = \arg\max \text{pdf}(y_{<0}^k)$ be the dominant mode, then the thickness interval corresponding to wall points was defined as $\widehat{y}_{<0}^{k,\text{L}} \leq y_{<0}^k \leq \widehat{y}_{<0}^{k,\text{R}}$, where:

$$\widehat{y}_{<0}^{k,\text{L}} = \max\left\{ y_{<0}^k < \widehat{y}_{<0}^k \;\middle|\; \frac{d}{dy}\text{pdf}(y_{<0}^k) = 0 \right\} \tag{5a}$$

$$\widehat{y}_{<0}^{k,\text{R}} = \min\left\{ y_{<0}^k > \widehat{y}_{<0}^k \;\middle|\; \frac{d}{dy}\text{pdf}(y_{<0}^k) = 0 \right\}. \tag{5b}$$

Note that the bounds in Eqn. (5) correspond to the first critical points of the PDF to the left and to the right of the dominant mode, respectively. This interval for each section is pictured in Fig. 5 by the red dashed lines. Consequently, the exterior points satisfy $y_{<0}^k < \widehat{y}_{<0}^{k,\text{L}}$ while the interior points correspond to $y_{<0}^k > \widehat{y}_{<0}^{k,\text{R}}$. This is also true for facade $F_y^{x<0}$, with the appropriate reformulation of the PDFs over the x-axis. However, keep in mind that the reverse is true for facades $F_x^{y>0}$ and $F_y^{x>0}$ because of the positive sign of the thickness component. Finally, let $W_{x,k}^{y<0}$, $I_{x,k}^{y<0}$, and $T_{x,k}^{y<0}$ be the point clouds corresponding to wall points, interior points, and exterior points, respectively. The second row of Fig. 5 shows the resulting point clouds. Note that the original assumption of recessed window/door placement holds and the interior points correspond mostly to the windows and door points. However, one can observe that roof and ground features are also present in the set of interior points. This highlights the need for outlier removal, where the roof, ground, and other facade characteristics are purposely removed in order to extract the dimensions of windows and doors.

2.4 Outlier removal

A distanced-based outlier removal was used to clean up the individual point cloud sections in order to extract accurate dimensions and locations of building features. A previous study by the authors used the Mahalanobis distance to determine outliers [13]. Although the properties of the Mahalanobis distance allowed for statistical hypothesis testing to determine outliers, the underlying Euclidean metric is not well suited for handling rectangular features such as windows and doors. Therefore, we present a new custom metric developed for rectangular point clouds.

Let $p \in \Pi$, where Π is a point cloud corresponding to wall or interior points. Consider the normalization:

$$\widetilde{p} = \frac{p - \mathbb{E}[p]}{\mathbb{M}\left[\left|p - \mathbb{M}[p]\right|\right]} \in \widetilde{\Pi}. \tag{6}$$

Here, $\mathbb{M}[\cdot]$ is the median function. The zero-mean and unit median absolute deviation transformation is a robust normalization approach to deal with outliers in each point cloud. Note that Eqn. (6) transforms the Cartesian coordinates of rectangular prisms, such as windows, doors, and rectangular walls, into cubes centered at the origin. In order to draw a boundary around the cube and remove outliers, consider using the Chebyshev metric applied to the rectangular coordinates of the normalized point cloud:

$$D_C(\widetilde{p}) = \max\left\{ |\tilde{x}|, |\tilde{y}|, |\tilde{z}| \;\middle|\; \begin{bmatrix}\tilde{x} & \tilde{y} & \tilde{z}\end{bmatrix} \subset \widetilde{p} \right\}. \tag{7}$$

Moreover, outliers are generally miss-classified points that are not only physically distant from the cluster centroid but also correspond to a different material. For example, note that point clouds of interior points $I_{x,k}^{y<0}$ in Fig. 5 have outliers corresponding to brick cladding and roof tiles. Such outliers are made of materials different from those of window frames. Therefore, the normalized light intensity

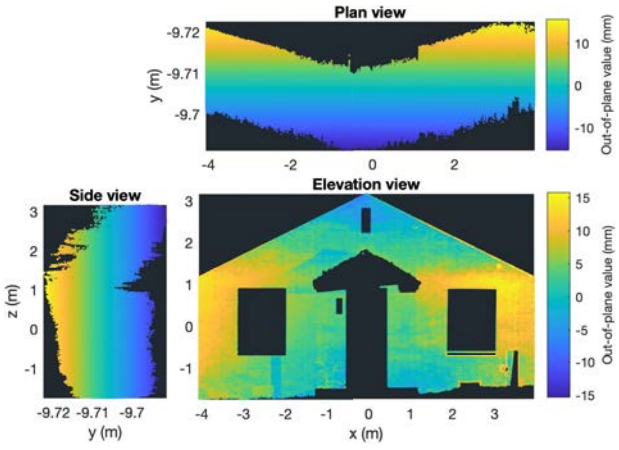

Figure 6. Plan, side, and elevation views of $W^{y<0}_{x,\mathbb{D}}$ with heat map describing flatness of outside surface.

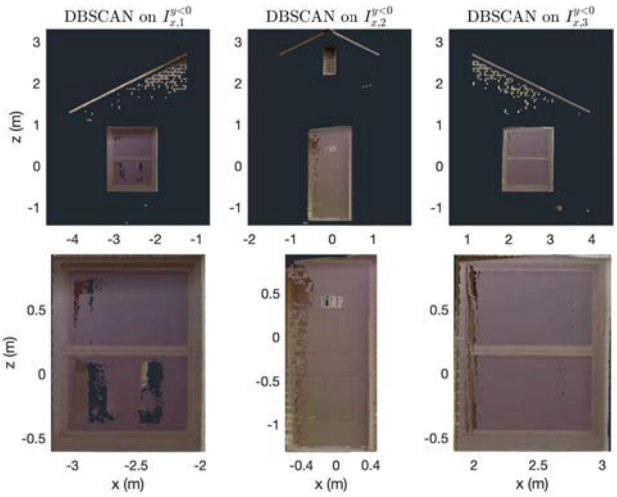

Figure 7. Outlier removal for $I^{y<0}_{x,k}$ using DBSCAN.

$\widetilde{L}$ can also be used to identify outliers since it correlates well with the reflectivity of the material. Finally, consider the following metric for identifying outlier points:

$$\mathbb{D}\colon \mathbb{R}^7 \mapsto \mathbb{R}^+, \quad \mathbb{D}(\widetilde{p}) = D_C(\widetilde{p}) + \alpha|\widetilde{L}|. \tag{8}$$

Here, $\alpha > 0$ is a calibration parameter that captures the relative importance of the light intensity component with respect to the rectangular coordinates within the point cloud.

The outlier removal task consists on discarding the elements of a point cloud with a distance $\mathbb{D}(\cdot)$ larger than a preset threshold D_{thres}. In other words, the reduced point cloud after the outlier removal task was defined as:

$$\Pi_{\mathbb{D}} = \{p \in \Pi \mid \mathbb{D}(\widetilde{p}) < D_{\text{thres}}, \widetilde{p} \in \widetilde{\Pi}\}. \tag{9}$$

2.4.1 Outlier removal for wall points

For this study, the calibration parameter $\alpha = 1$ worked well for all facades. On the other hand, the threshold D_{thres} was not uniform along all sections. As described by the flowchart in Fig. 1, the outlier removal task of the Auto-CuBES may require some iteration by the user. In this case, after a visual inspection of the resulting point cloud, the user can decide to pick a larger or smaller threshold D^{i+1}_{thres} at iteration i. It was empirically found that a value of $D^0_{\text{thres}} = 4$ provides a good starting point for the iteration.

Following the same example as before, consider the wall points $W^{y<0}_{x,k}$. After outliers were removed from each section, the wall points of the entire facade correspond to the union of all different sections, i.e., $W^{y<0}_{x,\mathbb{D}} = \bigcup_k W^{y<0}_{x,k,\mathbb{D}}$. Figure 6 shows the plan, side, and elevation views of the remaining elements after outlier removal. From the plan view, note that the cladding is not flat, but it rather bows in. The side view indicates that the wall is slightly leaning forward and is not perfectly plumb. The bowing, leaning, or bulging of external walls can be summarized in the elevation view by coloring the points according to the out-of-plane value, define as the position of each point with respect to the facade's average plane. The average plane was calculated with respect to the axis at the thickness of the wall, similar to Eqn. (4) but keeping a zero mean. The coloring not only helps to determine sections where wall stability could be a problem, but it also assists in the design of optimized connections for overclad panel retrofits.

2.4.2 Outlier removal for interior points

The top row of Fig. 7 shows the interior points $I^{y<0}_{x,k}$ for all three sections of the facade. Note that, in contrast to the wall points $W^{y<0}_{x,k}$, distinct clusters can be identified corresponding to window/door points, roof points, and other clusters such as vents. Hence, unsupervised clustering was performed. The algorithm chosen for this task was DBSCAN [14], which is a density-based clustering algorithm well suited to identify the cluster of each section corresponding to windows and doors. The DBSCAN algorithm determines if a point p belongs or not to a dense cluster by considering a neighborhood of radius $\epsilon = 12$ cm around p, if there are less than $m = 100$ points in the neighborhood, then the point is an outlier. Finally, the dense cluster corresponding to a window/door was decided based on its proximity to the centroid of $I^{y<0}_{x,k}$. The bottom row of Fig. 7 shows the resulting clusters after the DBSCAN algorithm was performed. Although the algorithm can correctly identify the dense clusters corresponding to the facade features, it cannot distinguish outliers within points at the edge of the clusters. It was observed that, in most cases, the subsequent use of the outlier removal method in Eqn. (9) was needed to obtain accurate dimensions.

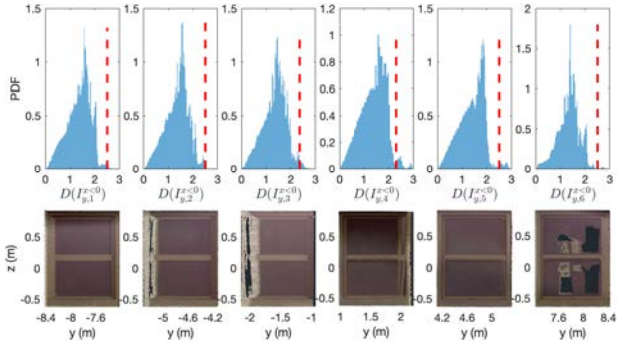

Figure 8. Clusters $I^{x<0}_{y,k,\mathbb{D}}$ corresponding to windows.

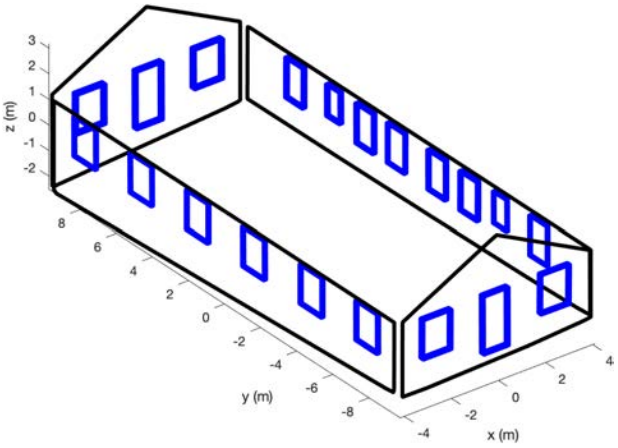

Figure 9. Wire-frame model of building envelope.

Figure 8 shows the resulting windows extracted from facade $F^{x<0}_{y}$ after the sequential application of DBSCAN and the outlier removal metric $\mathbb{D}(\cdot)$. The top row of Fig. 8 shows the histograms of the distance $\mathbb{D}(I^{x<0}_{y,k})$ calculated for each window after running the DBSCAN algorithm. For all the interior points, a value of $\alpha = 0.1$ was used to remove outliers mainly based on the cuboid shape of windows and doors. Even though the windows are physically similar, the histograms show some variability. This is due to the sensitivity of the point cloud scan with respect to scanner position, laser's angle of incidence, and window recessed distance from the wall. Nonetheless, we observed a general triangular distribution with a sharp drop after the main mode. Similar to the wall points, the distance threshold for removing outliers was not uniform for all windows and doors, and some iteration was needed. However, it was empirically found that a value of $D^0_{\text{thres}} = 2.5$ provided a good starting point. The individual values $D_{\text{thres},k}$ for each window are pictured by the red dashed lines over the histograms. Finally, the bottom row of Fig. 8 shows the resulting point clouds to be used for extracting the dimensions and relative positions of each window.

2.5 Wire-frame model generation

Once the original point cloud $\mathcal{P}$ has been segmented into wall points ($W^{y<0}_{x,\mathbb{D}}$, $W^{y>0}_{x,\mathbb{D}}$, $W^{x<0}_{y,\mathbb{D}}$, $W^{x>0}_{y,\mathbb{D}}$) and interior window/door points ($I^{y<0}_{x,k,\mathbb{D}}$, $I^{y>0}_{x,k,\mathbb{D}}$, $I^{x<0}_{y,k,\mathbb{D}}$, $I^{x>0}_{y,k,\mathbb{D}}$), the position and dimensions of the building envelope and its features can be extracted to create a simple wire-frame model. First, the exact dimension of each feature can be obtained by solving the minimum-volume oriented bounding box problem. This approach considers the possibility that windows and doors might not be aligned with the building facade. Moreover, as seen in Fig. 6, the facades are not perfectly flat, hence it should not be assumed that the windows are aligned. Thus, the bounding box not only provides the dimensions of each feature but also generates the rectangular prism needed for the wire-frame model. Given that the wall points do not have a rectangular shape, the facade dimensions and wire-frame model were obtained using the convex hull. Note that, in either case, the wire-frame generated is limited to convex point clouds. Non-convex facades, such as those with towers, will need a different approach to generate a wire-frame model.

Figure 9 shows the resulting wire-frame model when combining the convex hulls of wall points and the bounding boxes of windows and doors. Ultimately, the Auto-CuBES algorithm was able to reduce the highly detailed point cloud of the building envelope from 32.2 million points to a simplified wire-frame model with 309 points that summarized the essential information needed for retrofit panel design. Moreover, the wire-frame model combined with the out-of-plane coefficient in Fig. 6 can be a powerful tool to obtain necessary as-built dimensions for accurate overclad panel retrofits and for optimizing the position and dimensions of connections on existing facades previous to the retrofit process.

3 Results

Even though the Auto-CuBES algorithm has few calibration parameters for a single point cloud, the total number of parameters scales linearly with the number of facades and the number of features per facade in the building envelope. Ignoring the time needed to iterate over D^i_{thres} for windows and doors in each facade, the Auto-CuBES algorithm took a total of 12 m 22 s to run when implemented in MATLAB on a computer running on a quad-core Intel Core i7. Table 1 shows the breakdown of the time taken to run each task in the Auto-CuBES algorithm. Note that the DBSCAN algorithm takes the longest to run, taking 71% of the total running time. Although all other tasks in the Auto-CuBES algorithm have been optimized to some level using in-house developed code, the density-based

Table 1. Time taken to run Auto-CuBES algorithm

Identify and extract individual facades:	0 m 14 s
Identify wall and interior points:	2 m 03 s
DBSCAN only:	8 m 47 s
Outlier removal using metric $\mathbb{D}(\cdot)$:	1 m 05 s
Wire-frame model generation:	0 m 13 s
TOTAL TIME:	12 m 22 s

clustering for interior points was done using the single MATLAB command `dbscan`. Future work will explore alternatives to accelerate DBSCAN in order to reduce the overall Auto-CuBES execution time.

The accuracy of the Auto-CuBES was quantified by comparing the resulting dimensions for windows and doors versus manual laser measurements. Before discussing the results, it is important to mention two main caveats:

1. Manual measurements contain human errors: The person measuring needs to aim the laser at the exact interface between the window frame and the brick.
2. Only corner points were used to generate manual measurements: This assumes that the rough openings of windows and doors are square and straight.

Figure 10 shows the difference between the window dimensions obtained using the Auto-CuBES and the manual measurements for all facades. The calculated error is shown as a bar plot for each window's width (top row) and height (bottom row), and the scanner resolution (3 mm) is depicted with the dashed line. Facades $F_x^{y<0}$ and $F_x^{y>0}$ show errors comparable to the scanner resolution. They also correspond to the narrow section of the envelope, in which the scanner could maintain a high resolution throughout. Although not shown in the plot, the door dimensions for the facades showed comparable levels of accuracy. Facade $F_y^{x<0}$ shows a low error on window height, but some larger errors on window width. This is probably due to variable resolution across the facade. Although, on average, the scan has a 3 mm resolution, the actual controllable variable in the scanner is the angle increments for the robotic laser head. This means that surfaces that are scanned at an angle (e.g., edge of a wide facade) can have lower resolution than points directly in front of the scanner head. Finally, facade $F_y^{x>0}$ shows the largest errors among all. This was probably due to the scanner location with respect to the facade. In contrast to the other scans, the scanner position was not perpendicular to the facade due to the topographical constraints of the terrain around the building. Thus, the scan was taken at an angle and at a different altitude compared with the rest of the scans. This caused resolution and line-of-sight issues for the scanner.

To summarize the results, the mean absolute error (MAE) was calculated for all features in the building envelope. The width MAE was 4 mm while the height MAE was 4.4 mm when the four facades were included. This resulted in an overall MAE of 4.2 mm with an average scan resolution of 3 mm. If, however, facade $F_y^{x>0}$ is excluded due to the non-ideal scanning conditions, the overall MAE is closer to 3.2 mm. This indicates that, during ideal scanning conditions, the point cloud resolution is maintained and the error is minimized when the dimensions were automatically extracted from the point cloud. To cope with resolution and line-of-sight issues, future studies will look at a larger number of scans with overlapping areas.

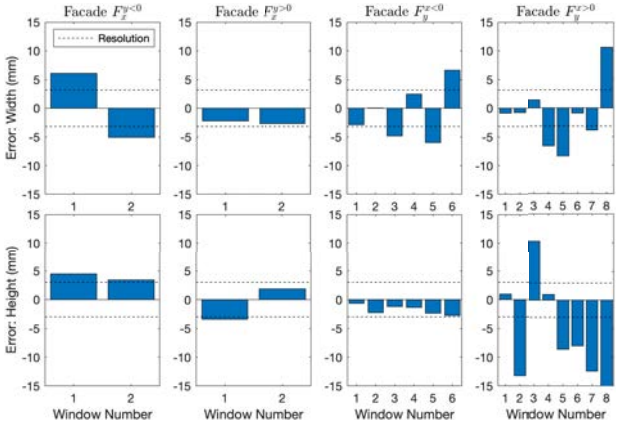

Figure 10. Error between bounding box dimensions and manual laser measurements for windows.

4 Conclusions

This study introduces the Auto-CuBES algorithm for extracting as-built dimensions of facades, windows, and doors from a 3D point could of a building envelope. In its current version, the algorithm is intended for simple one-floor structures comprising four main perpendicular convex facades, and rectangular windows and doors. The Auto-CuBES can process 32.2 million elements of a 3D point cloud and extract the minimum required points (309 in this study) to generate a wire-frame model of the building envelope. Additionally, the flatness of the external cladding is evaluated to optimize the design of connections for overclad panel retrofits. The individual tasks of the Auto-CuBES algorithm were based on unsupervised machine learning methods which do not require a training set with labeled data. This methodology was chosen due to the scarcity of labeled training samples for building envelopes and the inaccurate boundary segmentation of current supervised machine learning algorithms. The performance of the Auto-CuBES algorithm was evaluated in terms of computation time and accuracy compared to manual measurements. Without including the calibration time taken to find the optimal thresholds for distance-based

outlier removal, the Auto-CuBES processed 32.2 million points in 12 m 22 s. However, 71% of the computation time was taken by the DBSCAN task, which will be optimized in future versions. Manual laser measurements were compared against the automatically generated dimensions, resulting in an overall MAE of 4.2 mm for the point cloud in this study with 3 mm resolution. However, if only the facades that were scanned under ideal conditions are considered, then the MAE is comparable to the point cloud resolution. Therefore, the Auto-CuBES present a solution to expedite and reduce the cost of accurate overclad retrofits to bring old buildings up to energy codes.

Future research will focus on expanding the capabilities of the algorithm to include protruding and/or non-rectangular architectural features, as well as dealing with non-convex facades and non-rectangular envelopes. Current efforts on non-convex hull optimization and on random sample consensus (RANSAC) for arbitrary shape detection are being considered to expand the applicability of the Auto-CuBES. Additional development will include the integration with advanced construction methods such as the real-time evaluator (RTE) that is currently being developed by the authors.

5 Acknowledgements

This research was supported by the DOE Office of Energy Efficiency and Renewable Energy (EERE), Building Technologies Office, under contract DE-AC05-00OR22725, and used resources at the Building Technologies Research and Integration Center, a DOE-EERE User Facility at Oak Ridge National Laboratory.

References

[1] U.S. Energy Information Administration. Annual Energy Outlook 2022. Technical report, U.S. Department of Energy, 2022.

[2] The Alliance to Save Energy. Energy Efficiency Impact Report. Technical report, The American Council for an Energy-Efficient Economy, 2022.

[3] G. Iannaccone, G. Salvalai, M.M. Sesana, and R. Paolini. Integrated Approaches for Large Scale Energy Retrofitting of Existing Residential Building through Innovative External Insulation Prefabricated Panels. In *Sustainable Built Environment (SBE) Regional Conference*, pages 636–643, Zurich, 2016.

[4] G. Salvalai, M. M. Sesana, and G. Iannaccone. Deep renovation of multi-storey multi-owner existing residential buildings: A pilot case study in Italy. *Energy and Buildings*, 148:23–36, 2017.

[5] Norhan Bayomi, Mohanned El Kholy, John E. Fernandez, Senem Velipasalar, and Tarek Rakha. Building envelope object detection using yolo models. In *2022 Annual Modeling and Simulation Conference (ANNSIM)*, pages 617–630, 2022.

[6] E. Özdemir and F. Remondino. Segmentation of 3d Photogrammetric Point Cloud for 3d Building Modeling. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 6210:135–142, September 2018.

[7] R. Charles, H. Su, M. Kaichun, and L. J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 77–85, Los Alamitos, CA, USA, jul 2017. IEEE Computer Society.

[8] Xiaofei Yang, Enrique del Rey Castillo, Yang Zou, and Liam Wotherspoon. Semantic segmentation of bridge point clouds with a synthetic data augmentation strategy and graph-structured deep metric learning. *Automation in Construction*, 150:104838, 2023.

[9] Minju Kim, Dongmin Lee, Taehoon Kim, Sangmin Oh, and Hunhee Cho. Automated extraction of geometric primitives with solid lines from unstructured point clouds for creating digital buildings models. *Automation in Construction*, 145:104642, 2023.

[10] Chia-Tche Chang, Bastien Gorissen, and Samuel Melchior. Fast Oriented Bounding Box Optimization on the Rotation Group SO(3,R). *ACM Trans. Graph.*, 30(5), oct 2011.

[11] Kevin P. Murphy. *Machine Learning: A Probabilistic Perspective*. The MIT Press, 2012. ISBN 0262018020.

[12] D. Arthur and S. Vassilvitskii. K-means++: The advantages of careful seeding. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '07, page 1027–1035. Society for Industrial and Applied Mathematics, 2007.

[13] Bryan P. Maldonado, Nolan W. Hayes, Daniel Howard, and Diana Hun. Automatic Segmentation of Building Envelope Point Cloud Data Using Machine Learning. In *ASHRAE Topical Conference Proceedings*, 12 2022.

[14] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, KDD'96, page 226–231. AAAI Press, 1996.