3D Wireframe Reconstruction of Buildings from Point Cloud Data

Richard Wang
Avideh Zakhor, Ed.

Electrical Engineering and Computer Sciences
University of California, Berkeley

Technical Report No. UCB/EECS-2022-143
http://www2.eecs.berkeley.edu/Pubs/TechRpts/2022/EECS-2022-143.html

May 18, 2022

3D Wireframe Reconstruction of Buildings from Point Cloud Data

Richard Wang

Research Project

Submitted to the Department of Electrical Engineering and Computer Sciences, University of California at Berkeley, in partial satisfaction of the requirements for the degree of **Master of Science, Plan II**.

Approval for the Report and Comprehensive Examination:

Committee

Dr. Avideh Zakhor
Research Advisor

5/18/2022

(Date)

★ ★ ★ ★ ★ ★ ★

Dr. Jonathan Shewchuk
Second Reader

17 May 2022

(Date)

Abstract

Accurate 3D reconstruction of buildings has many applications including remodeling, renovation, or re-cladding. Existing software such as Revit and VectorWorks are primarily for designing new buildings, which by definition must be plumb, true, level, with 90 degree Manhattan geometry; as such they are not suitable for Scan-to-BIM applications where the building facades might not be vertical, flat, plumb, true, level or Manhattan. This is particularly true of old existing buildings in need of energy retrofit. In this paper, we develop a novel automatic pipeline for creating accurate 3D representation and modeling of a building beginning with a 3D point cloud. We use feature extraction and clustering methods in order to segment the original point cloud into its respective facades, roof, and ground, followed by the polygonization of each facade. Next we delineate the boundaries of fenestration such as doors and windows. We test our approach on two multifamily apartment buildings, two in Southern California and one in Canada. We characterize the accuracy of our method for one of the buildings in Southern California using hand measurements and tape, and show that our approach results in 33 to 50% less error than the corresponding Revit model.

Contents

List of Figures

List of Tables

1. Introduction

There is a need to create accurate 3D models of buildings in architecture applications such as renovation, remodeling, or re-cladding. In the re-cladding application, it is possible to cover the exterior of the building with insulation panels to improve the energy efficiency of buildings. To do so, one traditionally uses a 3D laser scanner to create a point cloud to be converted into Building Information Model (BIM) using Scan-to-BIM software such as Revit. Most existing architectural tools including Revit, are meant to be used for designing new buildings, which are necessarily vertical, plumb, true, level, and Manhattan. However, in remodeling or renovating existing buildings, one usually deals with older structures, which are not necessarily vertical, plumb, true or level with 90 degree Manhattan angles. One could argue that even for new buildings, the intended design does not always translate into reality once the construction is over. In this paper, we develop a new automatic method for accurate 3D model reconstruction of existing buildings from 3D point clouds that takes all of these non-idealities into consideration.

Our proposed workflow begins with a raw point cloud from a laser scan and through a process of feature extraction and clustering, segments the point cloud into facades from which a polygonal representation of the facades is computed. Our method has been designed to work without resorting to deep learning methods that require powerful GPUs and long training times. It allows for large point clouds 3-15 GB to be processed within 3-4 hours.

The outline of the thesis is as follows. Section 1.1 reviews related work. Section 1.2 outlines our proposed approach. Section 2 details the methods used in feature extraction and facade segmentation. Section 3 covers our approach on how to generate the polygonal representation of the segmented facades. Section 4 details result,s and Section 5 provides conclusions and future work.

1.1. Related Work

There are two main approaches to 3D building model reconstruction: image, and point cloud based. We start by reviewing related work on point cloud based reconstruction. Existing work in this area involving facade segmentation of raw unprocessed point clouds often utilizes off-the-shelf algorithm RANSAC [3, 19] to segment planes from point clouds. Typically, these methods employ successive iterations of RANSAC to segment each plane individually. However, as explained in Boulaassal et al. 2007 [3], plane segmentation from RANSAC can be inaccurate as it is being applied to the entire point cloud and established on 3 initial points. Zolanvari and Laefer [19] use RANSAC to roughly segment each facade of a building followed by their slicing method to extract boundaries. Adam et al. [1] introduce H-RANSAC, a hybrid modification of RANSAC, which combines 2D and 3D data to segment facades and windows in a point cloud.

Another class of methods utilize deep learning and convolutional neural networks from images to segment building facades [8]. Schmitz and Mayer [12] utilized the AlexNet architecture along with a deconvolutional layer to create an image classifier to classify pixels in building facade images. Fathalla and Vogiatzis [7] develop a pipeline utilizing convolutional neural networks (CNN) and restricted Boltzmann machine (RBM) to classify images of buildings. First a CNN is applied to the image for per-pixel classification and then is refined by a fine-grained classification model using RBMs to enforce contextual constraints. Rahmani and Mayer [11] introduce a pipeline using Structured Random Forest, CNN based Region Proposal Network, and rectangular fitting optimization for facade segmentation. Dai [5] uses a deep learning based semantic segmentation model with an ensemble learning strategy.

Another class of methods involve region growing, in which a region is grown from a seed based on a feature which groups together points of similar surfaces. An early study of region-growing algorithms by Woo et al. [16] simplified the point cloud into voxels via an oct-tree method, and merged voxels using region-growing based on similar normals. Carlberg [4] utilizes a region growing algorithm to classify landscape with 3D analysis to differentiate between planar, linear and scatter objects. Xu et. al [18] also use a region growing method to merge voxels to segment roofs from aerial LiDAR data. Vo et al. [15] use an oct-tree based region growing method for point cloud segmentation by voxelizing the point cloud and grouping together voxels with similar features computed by principle component analysis.

Most existing work regarding polygonization or generating polygonal representation of buildings feature some form of boundary detection algorithm. Liu et al.[10] compare linear and planar primitives to perform contour extraction and plane segmentation. Wudunn et al. [17] utilize Hough edge detection and RANSAC as well as connecting boundary lines to generate a polygonal building footprint from drone imagery. Tsenoglou [14] uses a region-based Hough transform to detect features in building facades. Zolanvari et al. [20] introduce an improved slicing method for facade and window detection.

1.2. Proposed Approach

Our approach to 3D representation of buildings consists of two parts: facade segmentation, followed by polygonization. Our approach to facade segmentation is heavily inspired by the region growing algorithm in Lalonde [9]. This algorithm grows a region from a seed point, by grouping all the points with similar normals. As such, each region corresponds to a planar facade of the building. We compute the normals in a similar fashion to Elseberg [6] and use positive cosine of the angle between normals to determine "similarity" between normals. The region growing stops when the search can no longer find any points with similar point normals. Once a surface is found, the points associated with that surface are removed from the point cloud and the algorithm continues with the remaining point cloud until all points are processed.

An important consideration in the above process is the stopping criteria. At what point do we decide to stop creating new planar facades? One possibility is to stop the process when the number of points in the resulting plane becomes a smaller than a pre-specified percentage of the total number of points in the point cloud. If this threshold is chosen to be small, we can be sure to find the facade for all planar surfaces in the building; however, this necessarily results in a large number of extraneous facades corresponding to pieces

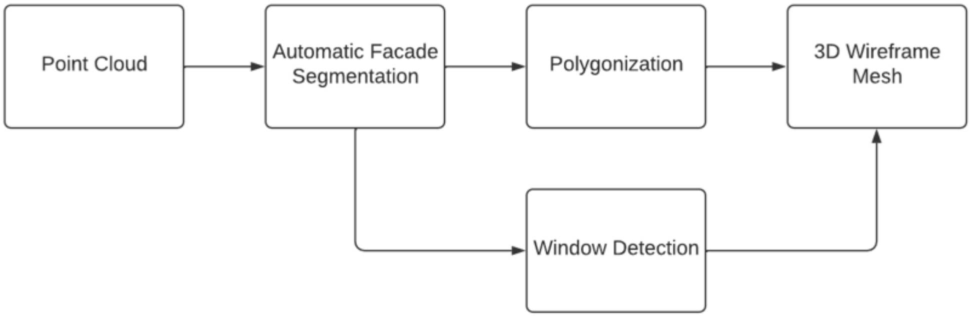


Figure 1.1.: Full Pipeline Flowchart

of ground, vegetation, fences and all kinds of other non-facade planes. On the other hand if the threshold is too large, we will have fewer extraneous planes, but might miss a small but important facade on the building. Our approach has been to choose a threshold which captures all the facades on the building, and prune away extraneous facades during the polygonization step.

Once the facades have been properly segmented, they are ready to be polygonized. There are three steps to polygonizing a given facade: first, calculate the neighboring facades; second, prune the neighbors for extraneous facades such as fences; third, intersect the planes of the remaining neighboring facades with the facade under consideration to produce lines; finally, intersect lines to obtain points which ultimately define the vertices of the polygon for a given facade. The full pipeline diagram is shown in Figure 1.1. In what follows, we describe each step in more detail.

2. Facade Segmentation

There are two steps to our proposed facade segmentation method: point cloud preprocessing followed by the feature extraction. First the original point cloud is voxel subsampled and point normals are calculated. Then, facades are iteratively segmented from the point cloud by a region growing algorithm on the condition of similar normals until all the points have been processed.

2.1. Voxel-Subsampling

Voxel subsampling is used in order to streamline and accelerate the segmentation procedure by the original high resolution point cloud before performing the rest of the analysis. Points from the original cloud are bucketed into a bunch of tiny uniform 3D cubes, or voxels and all the points in a voxel are averaged to create a single point. Note that this point is not the same as the center of the voxel. The purpose of this step is to create a smaller, more uniform point cloud that will lower compilation time in future steps.

2.2. Radius and K-Nearest Neighbor Search

Nearest neighbor searches are being used throughout our proposed method in both facade segmentation and polygonization. Nearest neighbor search queries a point to search for all the points in the point cloud nearest to that point. We utilize two different searches: radius search and hybrid radius/k-nearest neighbor search. Radius search returns all of the points within a specified distance from the query point whereas hybrid radius/k-nearest neighbor search returns k of the nearest neighboring points within a specified distance. We use hybrid radius/k-nearest neighbor search to calculate the surface normals and region growing algorithm for facade segmentation. In contrast, we use radius search is used in polygonization for point and edge validation.

Implementations of nearest neighbor searches utilize octree based methods in which the collection of points are represented in a k-d tree for faster searching (Elseberg, 2012).

2.3. Point Normal Calculation with Principle Component Analysis

Before segmenting the point cloud of the building into roof, facades, and ground, information about the surfaces in the point cloud must be extracted through the surface normals. We determine surface normals for each point with principle component analysis. For each point, the direction vector of the surface centered around that point within some region is computed. The method specifically uses a hybrid k-nearest neighbor/radius search in order to determine the neighborhood of points for the point for which the normal is computed. For this region of points, the covariance matrix is calculated followed by the eigen decomposition of that covariance matrix. The covariance measures the correlation between two random variables x and y. For n samples of each x and y, and the covariance between the random variables is as follows:

$$\sigma(x,y) = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y}) \tag{2.1}$$

In order to extend the concept of covariance to a set of points, we can calculate the sample covariance matrix for said set of points. In this case, the covariance between the features of each point is calculated as an element of the covariance matrix. Specifically for the covariance matrix C and set of points $X = [X_0, \ldots, X_n]$ with $\bar{X}$ representing the average, the covariance between the ith and jth point is given by:

$$C_{ij} = \sigma(X_i, X_j) \tag{2.2}$$

Therefore, the covariance matrix for the set of points is determined by:

$$C = \frac{1}{n-1}\sum_{i=1}^{n}(X_i - \bar{X})(X_i - \bar{X})^T \tag{2.3}$$

In order to extract the information from this covariance matrix, we determine the eigen decomposition of the covariance matrix. The eigen decomposition of any matrix is a representation of the matrix by its eigenvectors and eigenvalues. Matrices that are known to have eigen decompositions are diagonalizable and have the form

$$C = PDP^{-1}. \tag{2.4}$$

where the eigenvalues are listed as diagonal elements of matrix D and the eigenvectors are listed as the columns of P.

Given the eigenvectors, the eigenvector that is perpendicular to the most predominant plane is the direction of the normal of the surface. Note that with every surface there are two potential normal vectors.

2.4. Region Growing Method

Next we explain our proposed facade segmentation method. This algorithm is heavily inspired by the region growing algorithm in LaLonde [9]. The algorithm functions by growing a region from a seed point which captures an entire surface on the building. The hypothesis for this procedure is that if a seeded point lies on a surface, then all points on this surface should have similar normals. If this assumption holds true, then a region grown from a seeded point on the surface should capture the entire surface. A region is grown from the seed by adding points that have similar point normals as the seeded point. The similarity of normals is determined by the positive cosine of the angle between the normals. If this cosine is larger than 0.85, which we tuned empirically, then the normals are considered similar. The region growing stops when the search can no longer find any points with similar point normals. Once a surface is found, the algorithm continues with the remaining point cloud until all points within the cloud are processed. The algorithm is detailed in Algorithm 1. There are several parameters used in the facade segmentation algorithm.

Parameters:

- Point Normals Calculation:
  1. **voxel-size**: The size of voxels used for downsampling the point cloud
  2. **max-nn**: Maximum number of nearest neighbors for hybrid knn/radius search
  3. **r-seg**: segmentation radius for hybrid knn/radius search
- Region Growing:
  1. **k-nn**: k-nearest neighbors for k-nearest neighbor search
  2. **min-group-ratio**: Number of points in facade/number of points in total point cloud. Facades that have a ratio smaller than the user input threshold are not saved.
- RANSAC Plane Fitting:
  1. **num-iterations**: The total number of iterations that are performed in the Random Sample Consensus (RANSAC) algorithm. For accurate results, anywhere from 5,000 to 10,000 iterations is recommended.

Algorithm 1 Facade Segmentation Algorithm

```
Input: voxel-subsampled point cloud $P$ with point normals calculated
while points in $P$ unprocessed do
    create queue $Q$
    create set $F$ for facade points
    add unprocessed point $p_0$ in $P$ to queue $Q$ and set $F$
    let $n_0$ be $p_0$'s respective normal
    while $Q$ is not empty do
        pop the oldest point $p_i$ from queue $Q$
        add $p_i$ to set $F$ and mark as processed
        perform knn/radius hybrid search for points near $p_i$ in point cloud $P$
        add the neighboring points to set $S$
        for point $p_j$ in $S$ do
            if $|\cos\theta| = |\frac{n_0 \cdot n_j}{||n_0|||n_j||}| \leq 0.85$ then
                add $p_j$ to $Q$
            end if
        end for
    end while
    if size(F) $\geq$ min-group-ratio then
        create a new point cloud for the points in $F$
    end if
end while
```

The building we segmented is a fairly residential building in southern California which we refer to as Corona 205. The raw point cloud of Corona 205 is shown in Figure 2.1. The parameters used for segmentation are as follows:

- **voxel-size**: 0.02 m,
- **max-nn**: 150,
- **r-seg**: 0.05 m,
- **k-nn**: 10,
- **num-iterations**: 10000.

This set of parameters generated a total of 187 facades. The results of our facade segmentation algorithm can be seen in Figures 2.2, (a) showing well formed facades and (b) showing some "junk" facades. In section 3, we will cover the procedure for pruning these junk facades for the subsequent steps for creating a wireframe mesh.

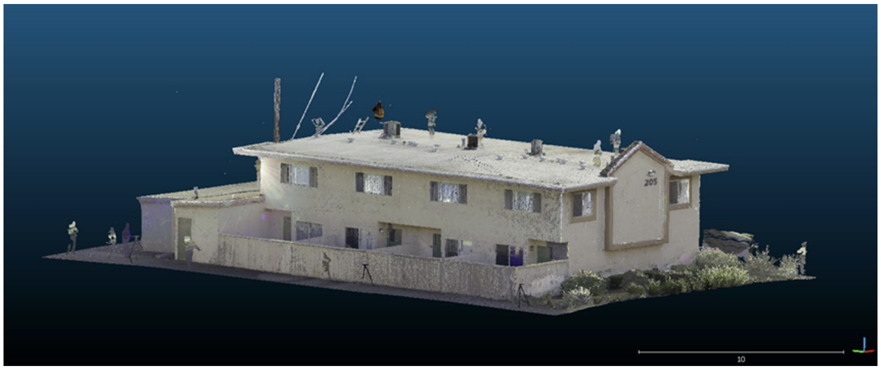
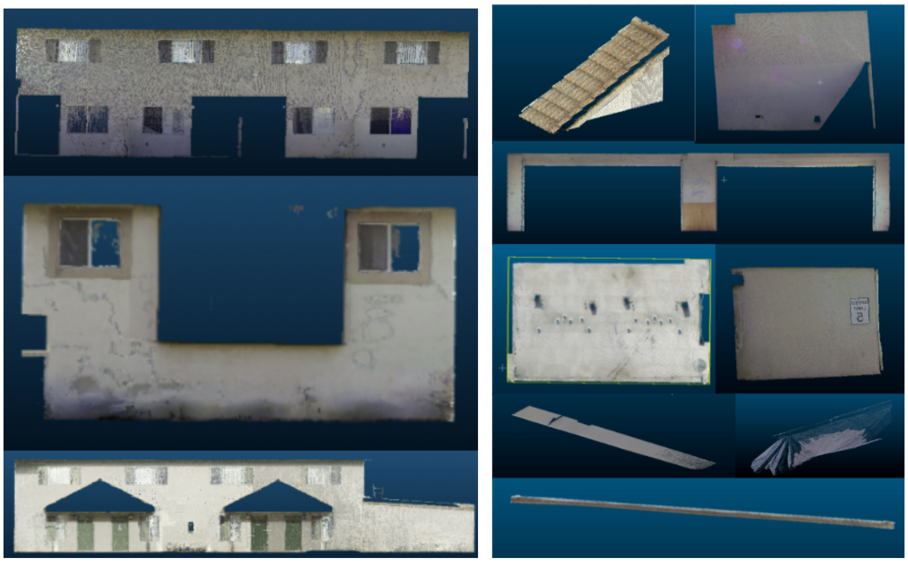


Figure 2.1.: Raw Point Cloud of Corona 205



(a)

(b)

Figure 2.2.: Example of segmented facades for Corona building 205 showing (a) well formed facades (b) "junk" facades

3. Creating a Wireframe Mesh

In this section, we review the polygonization step of the 3D modeling algorithm. Once the facades have been properly segmented, they are ready to be polygonized. To polygonize a single facade A, there are three steps: 1. Calculate the neighboring facades of facade A, 2. prune the neighbors to remove extraneous facades such as fences, and 3. intersect the planes of facades created in step 2 with plane A to compute the intersection points that define the final polygonal boundary for facade A. From now on, we will refer to facade A, the facade we want to polygonize, as the base facade.

3.1. Proximity Detection

This step involves creating a list of facades that are in the proximity of the base facade we wish to polygonize. To do this, we compute the intersection line between the base facade and every other facade Q obtained in the facade segmentation step and determine whether the intersection line is "within" the point cloud to check whether the facade Q is a neighbor. As seen in Figure 3.1, when intersecting the plane of the base facade (yellow) with the planes of the intersected facades (blue), the green line of intersection is considered to be a neighbor since there is a line segment that is "within" the base facade, whereas the red line intersection is not. To determine this, we sample along the intersection line to determine whether points on this line lie on the point set associated with each facade. This is determined by performing a search in a small radius around the sample point in both clouds. If the search reveals no neighboring points, then it indicates that the sample point does not lie within the point cloud indicating that the two facades are not neighbors. The procedure above is necessary because the facade segmentation step merely determines the planar equation of each facade and not their extent. Therefore, to determine proximity, we need to ensure that the intersection of the planes associated with two facades are inside the point cloud. The results of proximity detection are shown in Figure 3.2 where the base facade is outlined in red and neighboring facades are outlined in green.

3.2. Pruning

The purpose of the pruning step, which is carried out manually by the user, is to remove extraneous segments that could potentially interfere with the subsequent polygonization step. As seen in Figure 3.3, examples of such segments include fences, top and bottom portions of roofs, protrusions overhanging from buildings, vegetation, or extra sliver facades from the building. While one can limit the facade segmentation algorithm to find segments with sufficient number of points on them, as stated earlier, this could hinder finding all the needed facades for polygonization purposes. The user keeps the green facades and removes

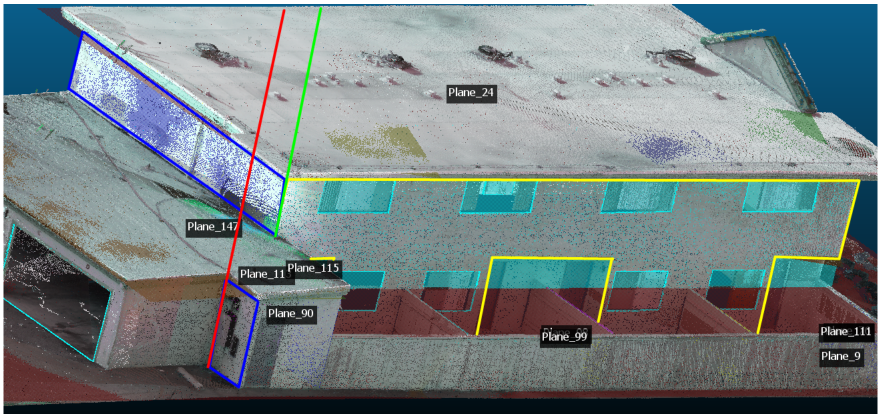


Figure 3.1.: Example of line intersections: base facade (yellow), intersected facades (blue), neighbor line intersection (green), non-neighbor line intersection (red)

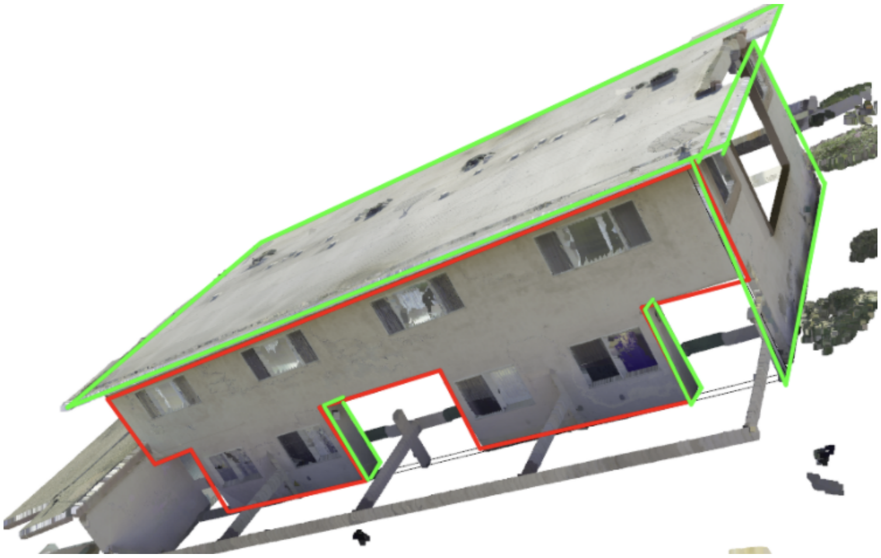


Figure 3.2.: Example of proximity detection for the base facade which is shown in red; the neighboring facades are shown in green. Some of the neighbors correspond to actual facades of the building and some are "junk" e.g. fences

the red ones. Interestingly, the facade segmentation algorithm in this case has found two planes for the roof: an upper one shown in red which must be removed, and a lower one shown in green which is kept to be intersected with the vertical facades in the polygonization step.

In practice, we encounter many situations in which the facade segmentation finds two or more facades corresponding to the two sides of the same wall. As seen in Figure 3.4. Most times, only one of the two needs to kept for polygonization.

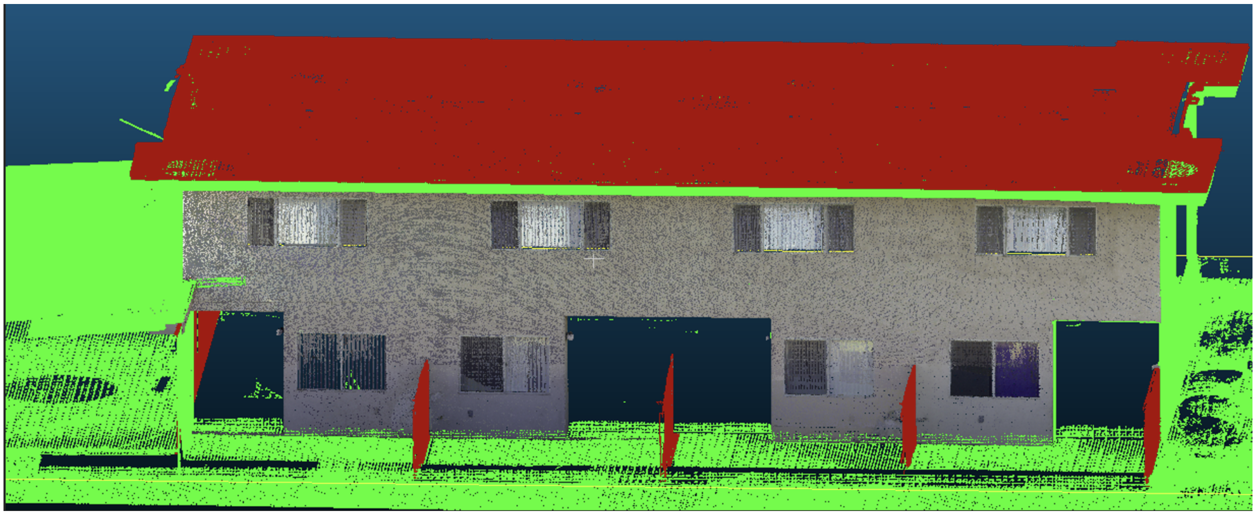


Figure 3.3.: Red facades are pruned away manually by the user, while the green ones are kept for polygonization

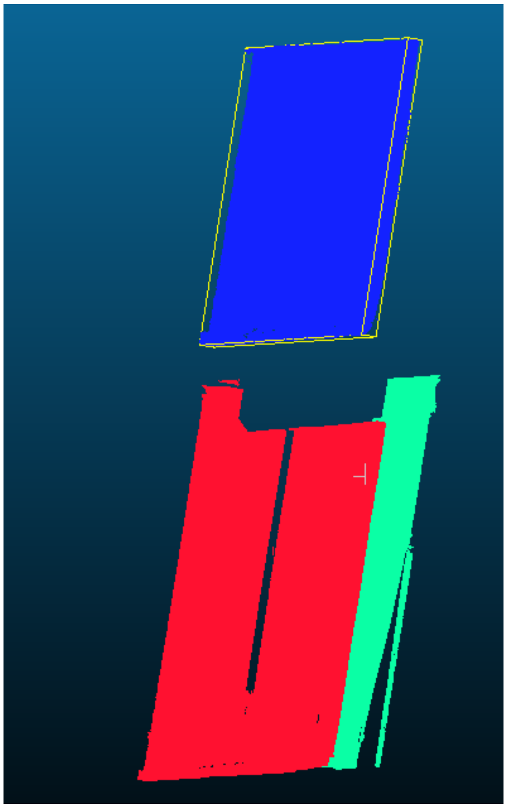


Figure 3.4.: Example of three facades with similar normals in red, blue, and green

3.3. Polygonization

The exact polygonal boundary of the base facade is computed after neighboring facades are determined and extraneous facades are manually removed. This is accomplished by using the point clouds of neighboring facades and their respective plane equations. The line intersections between neighboring facade's planes P_1 and P_2 and the base facade's plane B. We refer to the intersection between P_1 and B as l_1, the intersection between P_2 and B as l_2 and the intersection between P_1 and P_2 as l_3. Next we intersect l_1, l_2 and l_3 to compute their intersection point X which is the vertex of the polygon for all three planes P_1, P_2 and B which are illustrated in Figure 3.5. If these three lines do not form an intersection, i.e. two or more of the lines are parallel, then we know they do not form a facade corner. Next, we perform proximity analysis around point X using a radius nearest neighbor search to check whether the point is "valid", i.e. whether it is in the vicinity of P_1, P_2 and B. This reasoning is based on the assumption that corners of the polygon should be the intersection of three facades, as the intersection of three nonparallel planes is a point. Therefore, intersecting the three plane equations for the facades that make a corner of a facade should result in the coordinate of that corner.

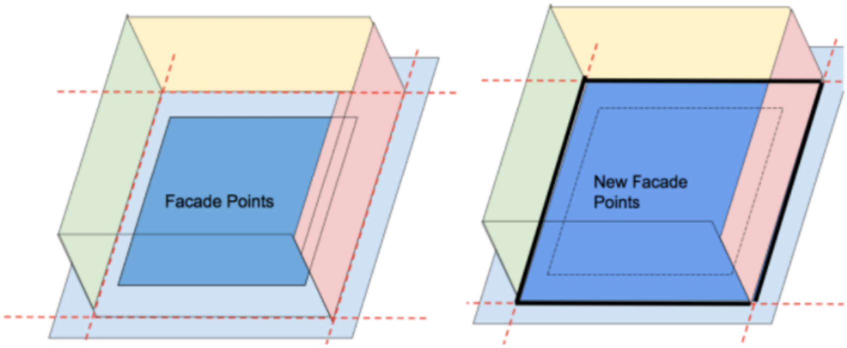


Figure 3.5.: Illustration of Intersections between planes of facades

3.4. Edge Validation

Once a set of candidate points have been found, the final step is to determine the valid edges for each set of points. Each pair of points generates a line segment, but not all line segments are valid. A valid line segment is one that is on the boundary of a facade, and when connected with other valid line segments for that facade, produces the polygonal outline for it. To determine the valid edges of a polygon, we proceed upon the assumption that each line segment should contain points within the point cloud on one side only. Line segments that either go through a facade or through empty space will not possess this property. Figure 3.6 illustrates an example of this concept, with valid line segments in green and invalid line segments in red. Each line segment is validated by taking points on the line and sampling for points in the facade point cloud on either side of the line segment with a radius nearest neighbor search. If we find points with the nearest neighbor search on one side but not the other, then the line segment is valid. Once edges between points are validated, the set of edges and points forms the final polygonal representation of the facade. Once the edges are determined, a simple traversal along each edge through each vertex generates the order of vertices that compose the polygonal representation for the base facade as illustrated in Figure 3.7.

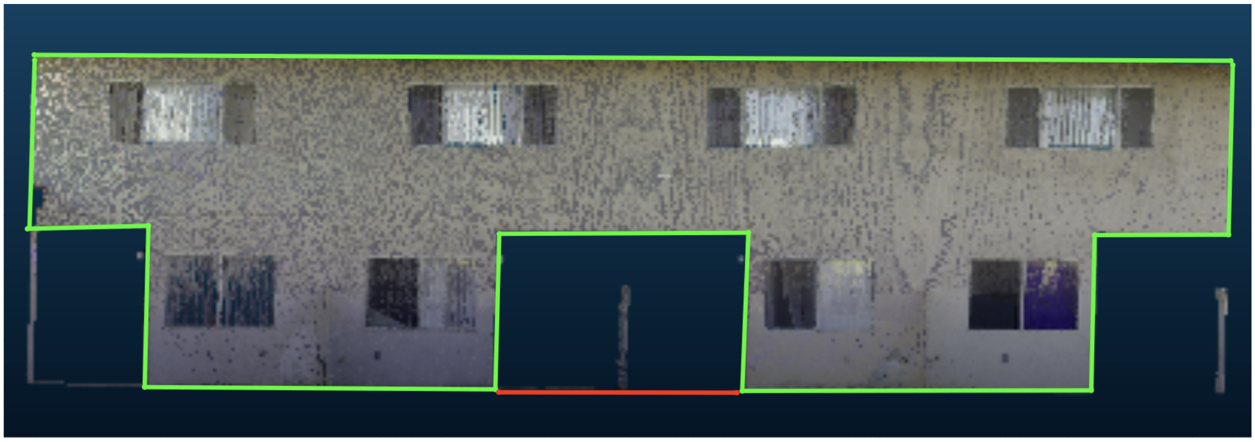


Figure 3.6.: Example of edge validation: red edges are removed and green ones are kept.

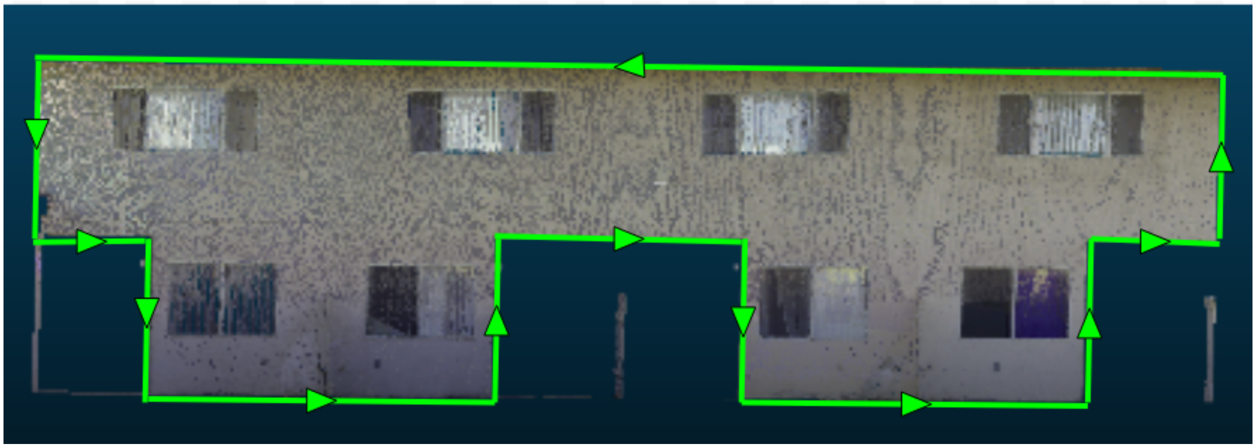


Figure 3.7.: Edge traversal example.

3.5. Fenestrations

Once the overall polygonal boundary has been determined, the next step is to determine the fenestrations of the facade such as windows and doors. Because the coloration and definition of fenestrations differs from building to building, this step in the pipeline is performed manually to obtain more accurate results. Initially, we considered selecting the corner points of each window in the point cloud. However, this method proved to be highly inaccurate as successively zooming makes it hard to visually discern the corner of the window and choosing a single point to represent the corner leads to inaccurate measurements as shown in Figure

3.8. The window corner is easily recognizable in the upper left image; after zooming in to get the lower right image, it becomes more difficult for the human eye to discern the true position of the corner of the window, hence to click on the "correct" point as the window corner. Instead we opted to determine the boundary of a fenestration by selecting several points on each side and interpolating the points with a straight line. Then, we intersect these lines to form the corners of the fenestration. Intuitively, the more points chosen the more accurate window measurements. Figure 3.9 demonstrates a selection of points to generate the window.

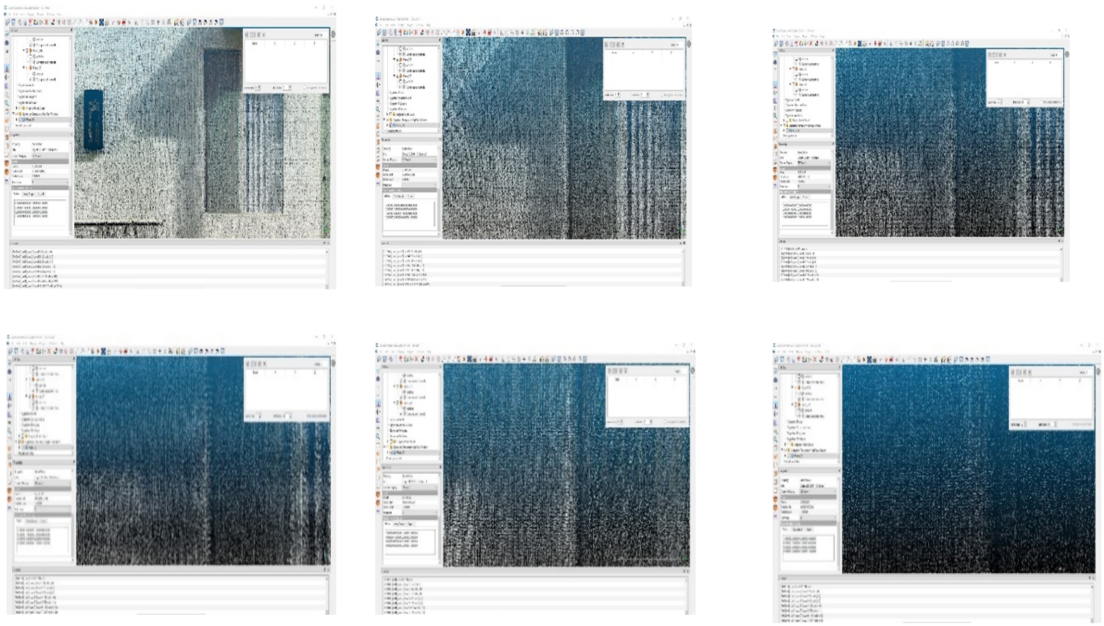


Figure 3.8.: Successive zooming of the side of a window for the point cloud of Corona 205 within Qpater tool. Zooming from upper right to lower left, it becomes harder to discern the "correct" point for the corner

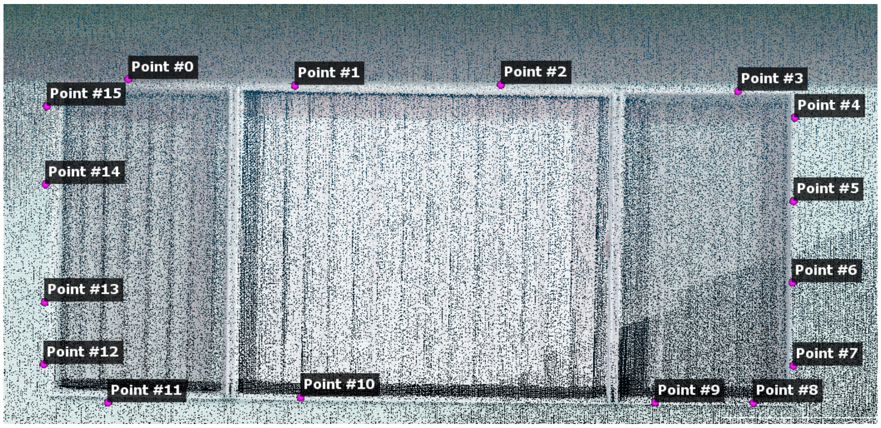


Figure 3.9.: Sample points used to delineate the boundaries of a window

4. Experimental Results

4.1. Datasets

We captured the point clouds for the Corona del Rey apartment for buildings 205 and 217 as well as a building named Presland in Canada. We tested our proposed approach on Corona Del Rey apartment 205 and validated our model by comparing it with Revit. The point cloud size for Corona 205 is 228 million points. In the e-57 format this results in a 2.4 GB file, which would require a computer with 64 GB or more memory to be able to visualize all the points. A screenshot of that point cloud visualized on a computer with 128 GB of disk space is shown in Figure 4.1. We created the wireframe model for Corona 205 shown in Figure 4.2 using the method described in previous sections. On a Windows machine with 1.86 TB of disk space, it took a total of 4 hours, 22 minutes, and 8 seconds to fully segment and compute planes for every facade in Corona 205. Once the facades and wireframe are computed, we generate .dxf files which are loaded into Rhino 7, an industrial design software, to take measurements.

We generated the wireframe models for Corona 217 and Presland as well. Corona 217 consists of 189 million points and can be seen in Figure 4.3. The Presland point cloud consists of 194 million points and can be visualized in Figure 4.4. The wireframe model for Corona 217, which took a total of 4 hours, 27 minutes, and 22 seconds, is shown in Figure 4.5. Likewise, the wireframe model for Presland took a total of 3 hours 56 minutes and 3 seconds and is shown in Figure 4.6. All three models used RANSAC num-iterations of 10,000 and min-ratio of 0.0002.

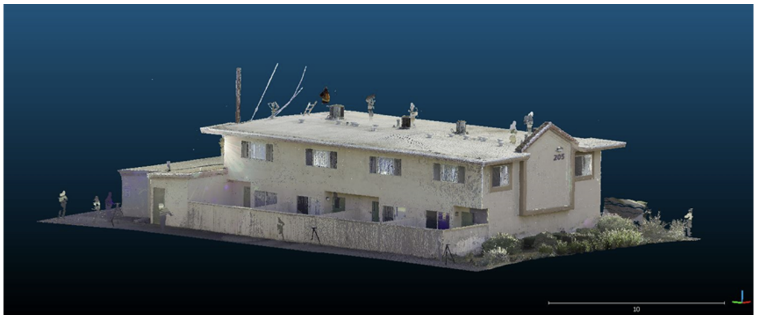


Figure 4.1.: Point cloud visualization of the Corona 205.

4.2. Deviation Files

Deviation maps illustrate the deviations on a facade with respect to the plane of that facade, which help visualize the non-uniformity of the facade. They are computed by projecting the points corresponding to a

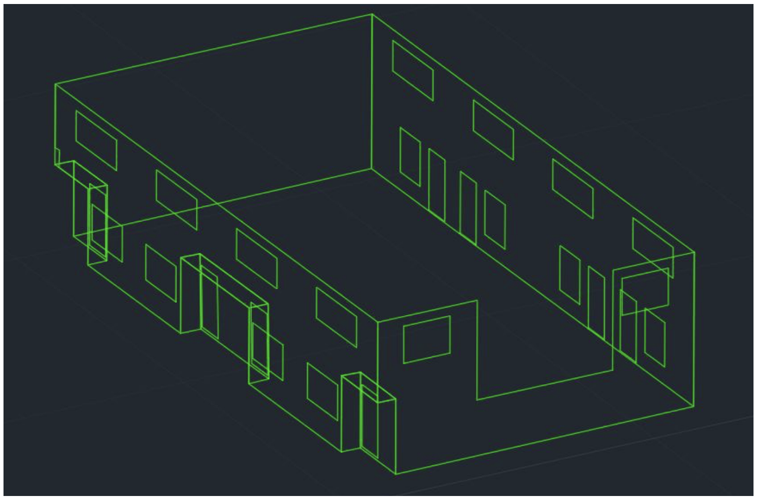


Figure 4.2.: 3D wireframe reconstruction of Corona 205 from point cloud.

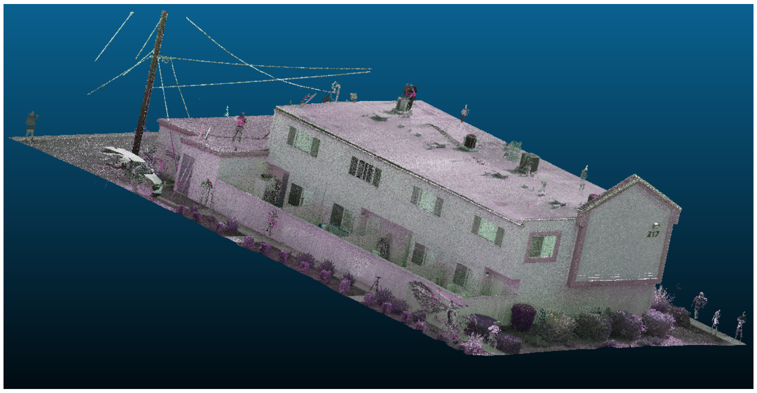


Figure 4.3.: Point cloud visualization of the Corona 217.

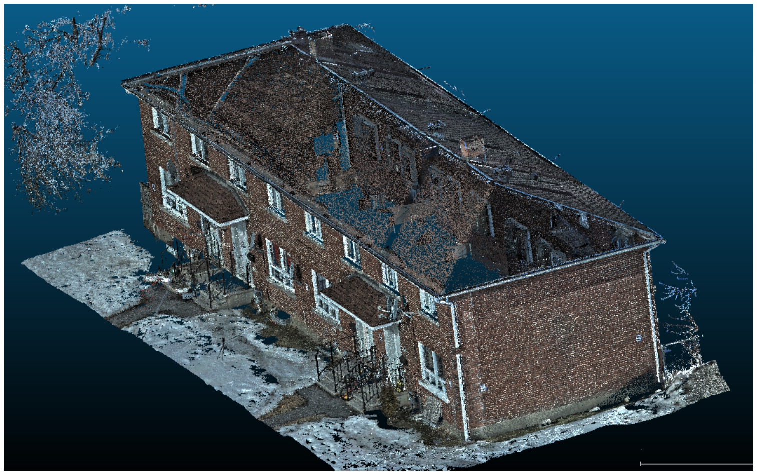


Figure 4.4.: Point cloud visualization of the Presland

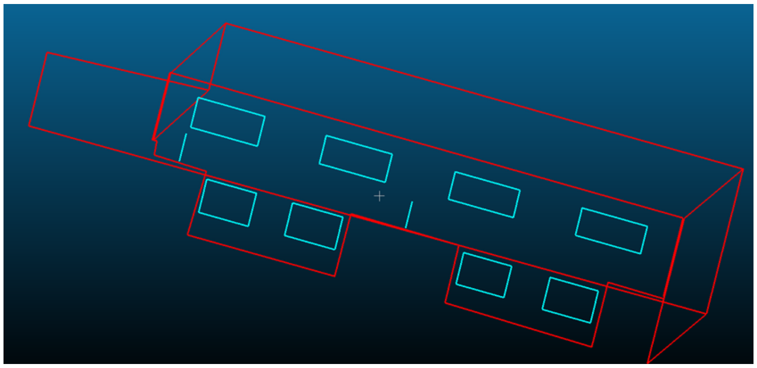


Figure 4.5.: 3D wireframe reconstruction of Corona 217 from point cloud.

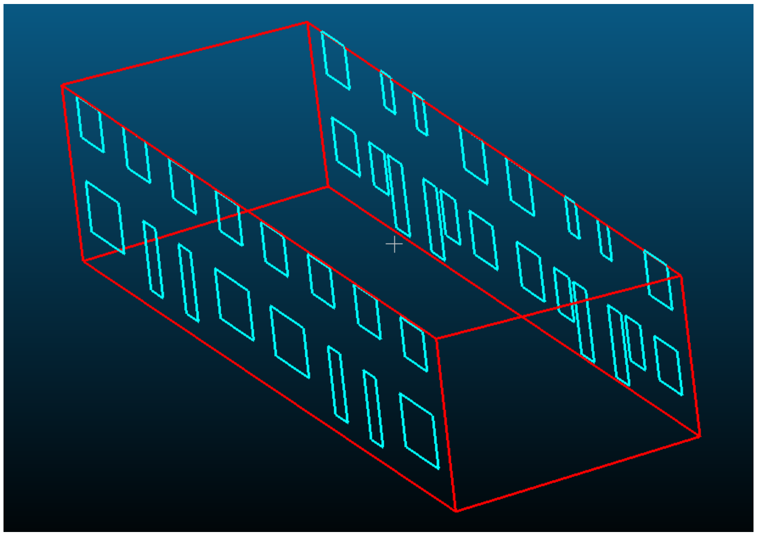
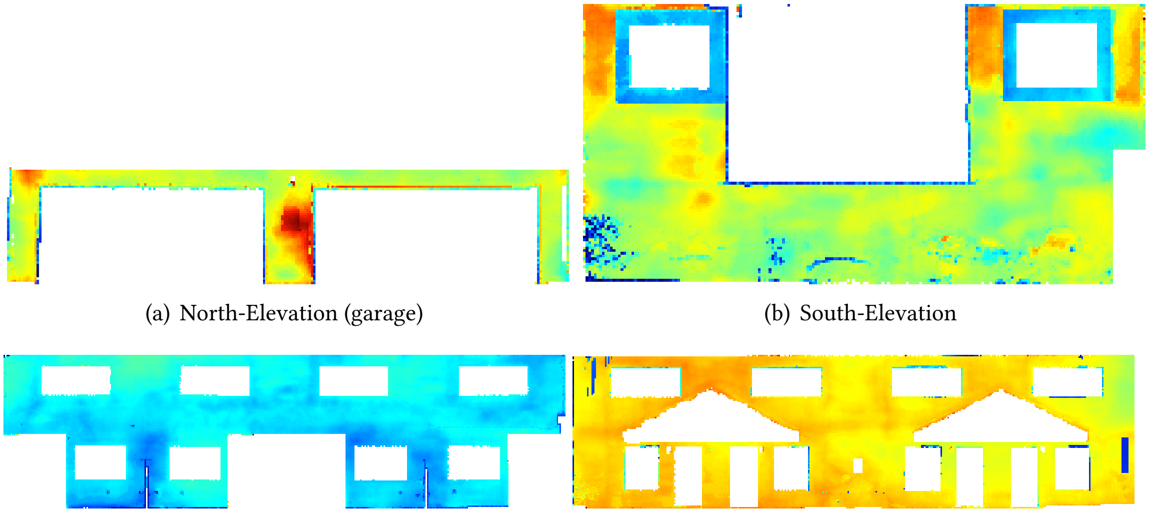


Figure 4.6.: 3D wireframe reconstruction of Presland from point cloud.

given facade onto the facade plane. The deviation maps are useful in the retrofit process such as installing insulation panels to the building exterior. The deviation maps for each of the four facades for Corona 205, one of which represents the garage, are shown in Figure 4.7.

4.3. Measurements of Corona 205

Thorough hand measurements were performed using tape and Disto-laser. We consider two different types of measurements in this report: "internal" and "overall". Internal measurements refer to those taken of horizontal and vertical distances between fenestration edges and/or facade edges. "Overall" measurements involve horizontal distances from a reference point usually at the edge of the facade to various facade features usually at the edge of fenestration. In Figure 4.8 internal measurements for the West-Elevation facade of Corona 205 are shown in red and orange and "overall" measurements are shown in green. The level of precision of the measurements is ±0.01 feet. Each fenestration is labeled with a number as seen in Figure 4.9. Once the wireframe model is computed, we use Rhino 7 to measure the dimensions of the



(a) North-Elevation (garage)

(b) South-Elevation

(c) East-Elevation

(d) West-Elevation

Figure 4.7.: Deviation maps for all facades of Corona 205.

wireframe model as seen in Figure 4.10. For the rest of this report, Table 4.1 shows the correspondence between the plots in the figures of this report and raw data used to generate them shown in the tables in Appendix A.

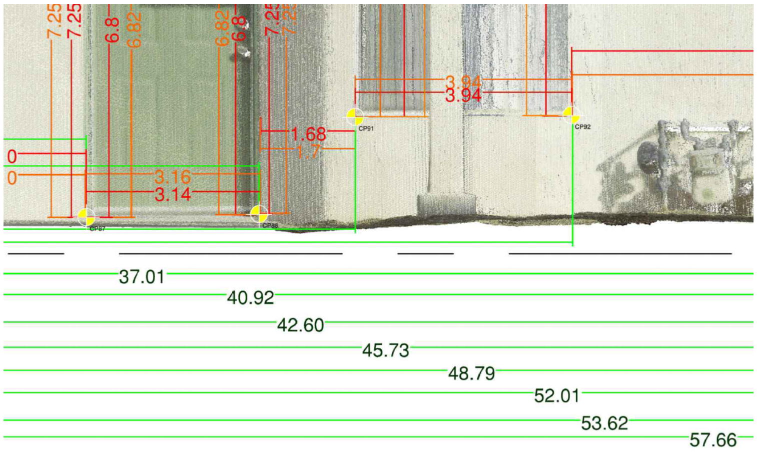


Figure 4.8.: Example Hand Measurements of Corona 205 taken in 2022, used to compare our method and existing models

4.4. Analysis of the Effect of Selecting Different Min-Group-Ratios

The min-group-ratio parameter is one of the instrumental parameters in the facade segmentation algorithm. If the min-group-ratio is too high, we might filter out facades that are required to form the polygonal boundary of a facade. This would require using the manual method to extract the required facade for polygonization. On the other hand, if the min-group ratio is too low, there will be an excess of facades,

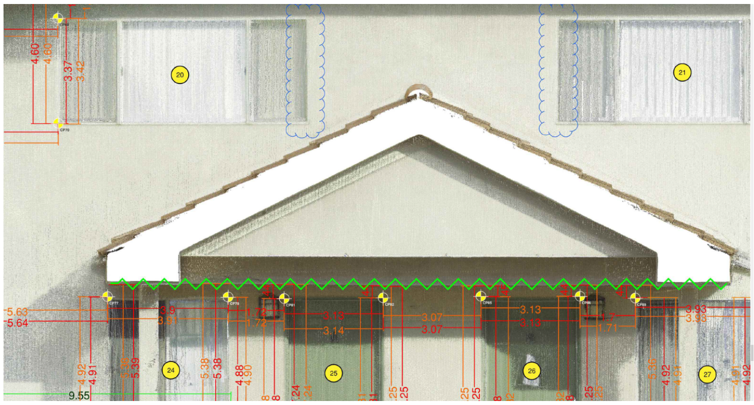


Figure 4.9.: Example numbering of fenestrations of West-Elevation Hand Measurements

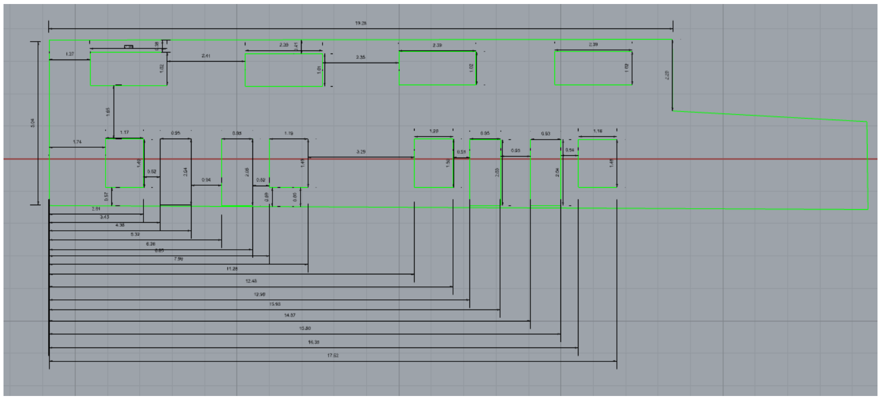


Figure 4.10.: Measurements taken of West-Elevation in Rhino 7

Figure	Table in Appendix A
4.11	A.3 and A.4
4.12	A.5 and A.6
4.13	A.8 and A.8
4.14	A.9 and A.10
4.21	A.11 through A.14
4.26	A.15 and A.16
4.27	A.17 and A.18
4.28	A.19 and A.20
4.29	A.21 and A.22
4.30	A.23 through A.26

Table 4.1.: Table of figures with their corresponding tables from the Appendix

which makes the pruning step more tedious and time-consuming. Table 4.2 illustrates the number of facades that are segmented for different values of min-group-ratio. The min-group-ratio resulting in the best user experience for Corona 205 is bolded at 0.0002 on a 5 million point downsampled version of Corona. For facades that are largely plain and rectangular filtering out extraneous smaller segmented pieces with a larger min-group-ratio is sufficient for polygonization. However, for buildings where the facades have more intricate geometries, the smaller 0.0002 min-group-ratio provides the most satisfactory user experience in that very few facades need to be manually created by the user. For a 0.0002 min-group-ratio, the segmentation produced a total of 145 facades. For Corona 205, min-group-ratios of 0.2, 0.02, and 0.002 do not segment enough facades for polygonization purposes.

The remaining parameters that remain constant for the results in Table 4.2.

- **voxel-size**:0.02 m,
- **max-nn**: 150,
- **r-seg**:0.05 m,
- **k-nn**: 10,
- **num-iterations**: 1000.

min-group-ratio	# of Facades Segmented	CPU Time to Segment
0.2	1	1 minute 56 seconds
0.02	8	2 minutes 26 seconds
0.002	31	2 minutes 53 seconds
0.0002	**145**	**3 minutes 13 seconds**
0.00002	1368	7 minutes 20 seconds

Table 4.2.: Comparison of numbers of facades segmented and run time for a 5 million point downsampled point cloud for Corona 205.

4.5. Analysis of Different Point Picking Numbers for Windows

Since windows and doors are processed manually by selecting several points per side of the window, more points selected per side typically yield more accurate measurements. Table 4.3 illustrates this concept, showing a 66% decrease in error when the number of sampled points along the window boundary is increased from 2 to 4. While selecting more points provides more accurate measurements, the trade-off is overall processing time.

Measurement Method	2 Points per Side	4 Points per Side
Tape (Feet)	0.06	0.02
Laser (Feet)	0.09	0.03

Table 4.3.: Comparison of average absolute error for different number of points picked per side for 4 West-Elevation windows

4.6. Analysis of Effect of Number of RANSAC Iterations on Accuracy

Since the accuracy of our pipeline relies on accurate plane intersection, it is important to determine the effect of the number of iterations RANSAC plane fitting has on the measurements. Table 4.4 illustrates the average absolute errors for each respective number of iterations of RANSAC for the East-Elevation. As one can see, 10,000 iterations of RANSAC iterations generated results with the least error.

num-iterations	Average Absolute Error (Feet)
1000	0.07
10000	0.03
100000	0.10

Table 4.4.: Average absolute "overall" error with differing number of iterations for RANSAC on West-Elevation

4.7. Comparison of Automatic Segmentation to Manual Segmentation

In this section, we compare the errors between using the automatic facade segmentation method and the manual segmentation method. In manual segmentation, the user segments each facade by hand using a clipping tool and adjusting a bounding box to isolate the points within a point cloud corresponding to a facade then introducing human error. A comparison of the average error produced between automatic and interactive facade segmentation can be seen below in Table 4.5. The table shows that the automatic method provides more accurate measurements by 0.01 to 0.02 feet.

Facade	Interactive Error (Feet)	Automatic Error (Feet)
West-Elevation	0.04	0.03
East-Elevation	0.04	0.04
South-Elevation	0.07	0.05

Table 4.5.: Comparison of average absolute "overall" error of measurements between QPater interactive and QPater automatic segmentation methods

4.8. Error Analysis against Hand Measurements

In this section, we analyze the internal and overall difference between our approach and hand measurements using laser and tape. We refer to this difference as "error" even though it is not always clear whether hand measurements are more or less accurate than the point cloud derived dimensions. In all plots and figures in the remainder of this paper, reference to average error indicates average absolute error.

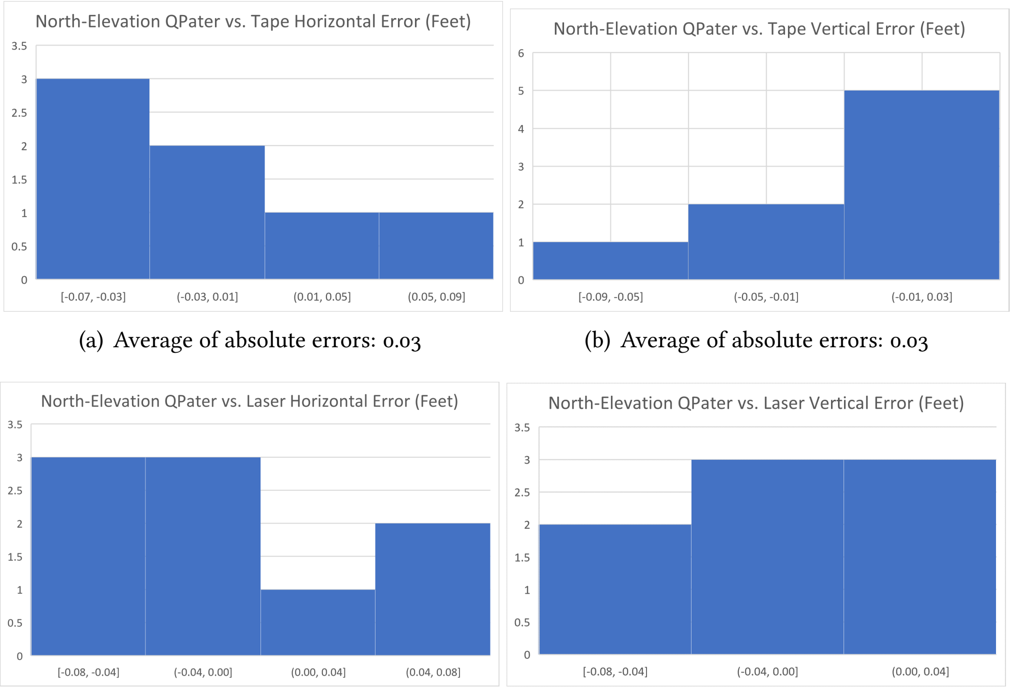
4.8.1. Internal Measurements: QPater

Histograms of internal vertical and horizontal errors for each facade can be found in Figures 4.11 through 4.14 for North, South, East, and West elevations, respectively. Figures 4.15 and 4.16 show the combined errors for laser and tape measurements, respectively. The summary of maximum and average absolute error for the plots shown in Figures 4.11 through 4.14 is shown in the second and forth columns of Table 4.6 from which a number of observations can be made. First, while the error between laser and tape are close to each other, the average absolute error corresponding to horizontal laser measurements tends to be slightly larger than that of tape measurements in two out of four facades in Table 4.6. This indicates that the horizontal laser measurements might be slightly less reliable than the tape ones. Second, the average absolute error in the vertical dimension tends to be higher than of horizontal dimension in three out of four facades. There can be multiple explanations for this observation. First, the "ground" plane in our method is almost certainly different from that chosen in hand measurements, thus adversely affecting vertical measurements. Second, objects protruding from the facades such as window frames result in slight occlusions or shadow right above the upper horizontal frames of windows particularly in higher stories. This shadow results from the laser data being captured from the ground. This leads to errors in manual measurements of the vertical distances involving upper horizontal boundary of window frames in the point cloud. For the horizontal direction however, the stitching of multiple scans collected laterally on the ground avoids any such occlusions or shadows along the vertical edge of window frames. Our third observation is that the maximum error in the vertical dimension is about twice as large as the horizontal maximum error in two out of four facades: East and West-Elevation. These large vertical errors occur predominantly with door height measurements in West-Elevation and East-Elevation, as seen in Figures 4.17 and 4.18, where yellow and blue lines denote errors of less and more than 0.06 feet respectively. The largest of these internal errors is 0.16 feet, corresponding to the left and right height measurements of the leftmost door in the East Elevation as shown in Figure 4.18. One hypothesis for the large door errors might be the threshold associated with a door which is at times hard to discern in a point cloud. The ground truth hand measurement defines the door lengths to include the threshold.

In South-Elevation, 3 of the 7 vertical measurements are outliers with errors between 0.10 and 0.13, corresponding to the window height measurements as shown in Figure 4.19. The error is larger around these features since unlike the polygonal boundaries, which are computed by intersecting planes, window and doors are determined manually which can introduce human and rounding errors. Specifically, as seen in Figure 3.8, it is not always easy for a user to discern and mark the boundary of the fenestration in the point cloud especially when the color of the fenestration is similar to that of the surrounding facade. Additionally, points on the interior side of the window, circled red in 4.20(a) and shown from a side view in 4.20(b) tend to blur the window boundary during the manual window extraction process.

4.8.2. "Overall" Measurements: QPater

The "overall" measurements for the four facades of Corona 205 are shown in Figure 4.21. As seen, most of these "overall" errors lie within ±0.05 feet, with a few errors greater than ±0.05. Figure 4.22 shows the combined error of all "overall" measurements into a single histogram. The maximum and average asbsolute error for the "overall" measurements for all four facades are shown in the second and fourth columns of Table 4.7, with averages ranging from 0.04 to 0.08 feet and maximums ranging from 0.07 to 0.18 feet.



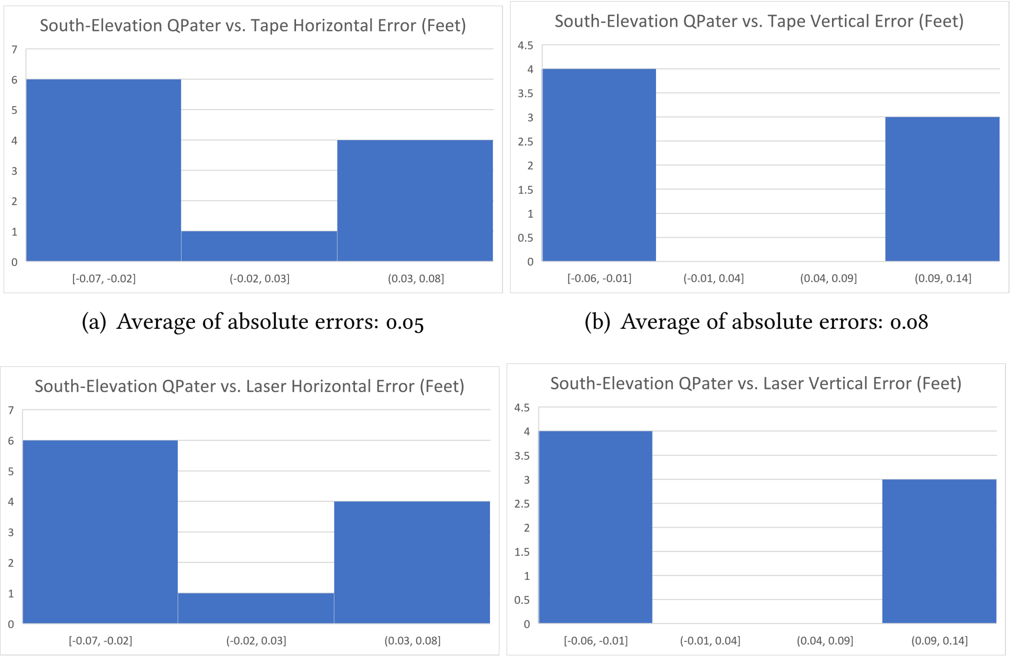
(a) Average of absolute errors: 0.03

(b) Average of absolute errors: 0.03

(c) Average of absolute errors: 0.04

(d) Average of absolute errors: 0.03

Figure 4.11.: Internal horizontal and vertical for North-Elevation facade (feet)



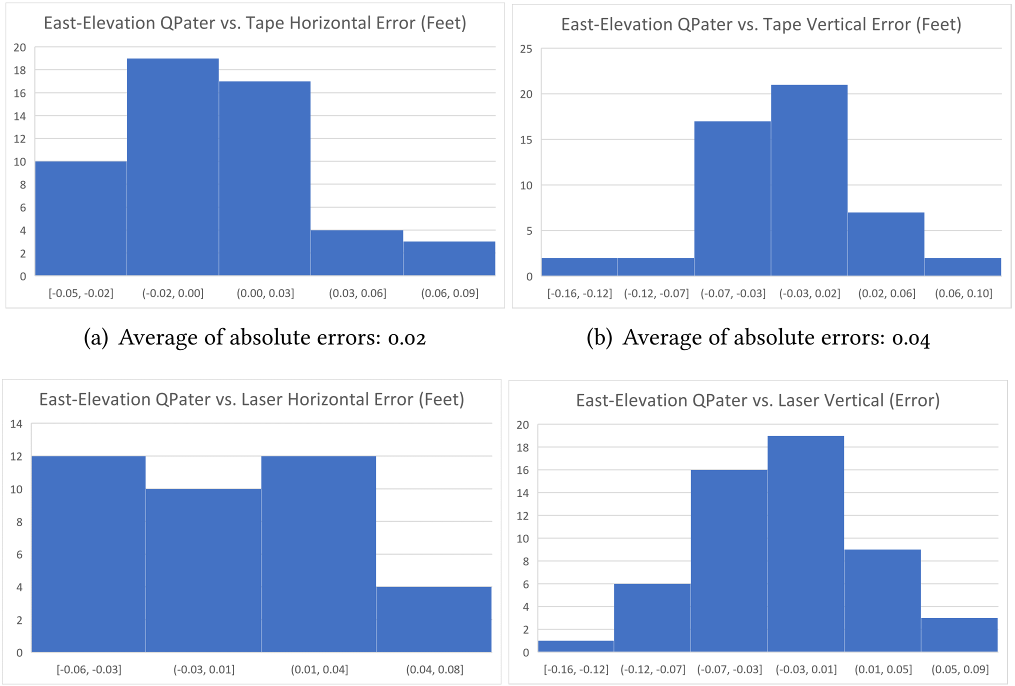
(a) Average of absolute errors: 0.05

(b) Average of absolute errors: 0.08

(c) Average of absolute errors: 0.05

(d) Average of absolute errors: 0.08

Figure 4.12.: Internal horizontal and vertical errors for South-Elevation facade (feet)



(a) Average of absolute errors: 0.02

(b) Average of absolute errors: 0.04

(c) Average of absolute errors: 0.03

(d) Average of absolute errors: 0.04

Figure 4.13.: Internal horizontal and vertical errors for East-Elevation facade (feet)

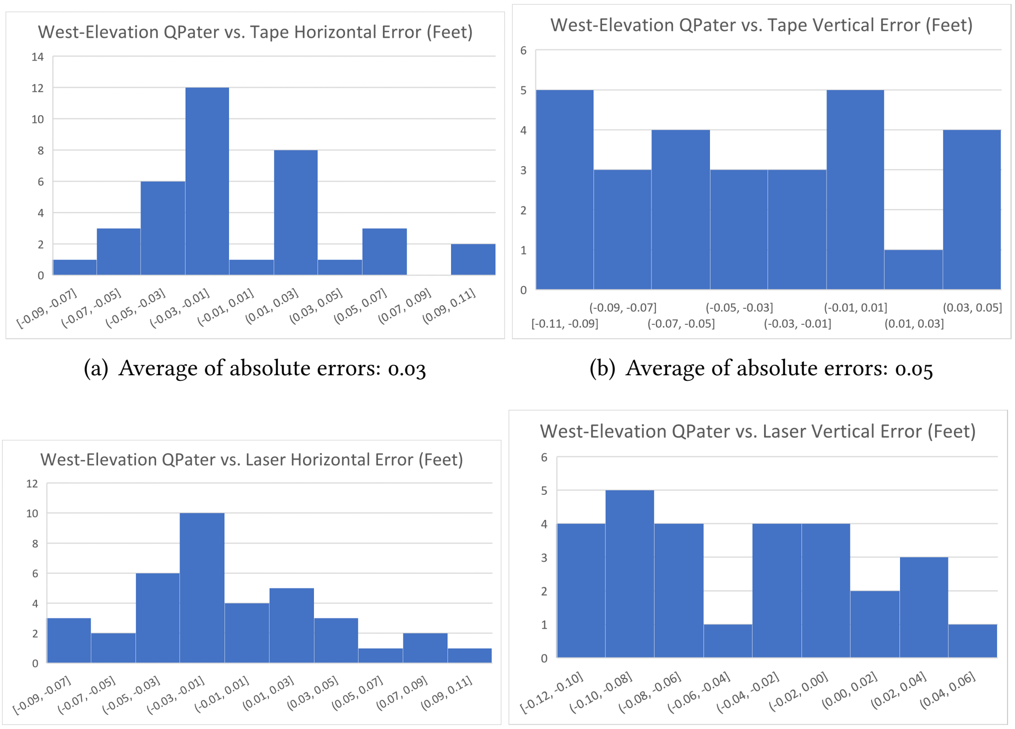
There are a few outliers with the "overall" measurements. In North-Elevation, there is one error of 0.17 feet shown in Figure 4.23. In West-Elevation, there are also two "overall" measurements with particularly large errors of 0.17 and 0.18 feet highlighted in Figure 4.24. These errors are also large for the Revit model which we will later use to compare our method with. This suggests that there may be a systematic error in some of the hand measurements and that the "digital" point cloud approach might in fact be more accurate.

4.9. Comparison of QPater Automatic Segmentation to Revit

Since Revit by Autodesk is considered to be the state of the art in scan-to-BIM conversion, in this section, we compare the accuracy of our method against Revit for Corona 205 using the hand measurements as ground truth. A Revit model of the Corona 205 dataset can be seen in Figure 4.25.

4.9.1. Internal Measurements: Revit

Internal horizontal and vertical errors for the Revit model are shown in Figures 4.26 through 4.29 for all four elevations. The third and fifth columns of Table4.6 show the maximum and absolute average error for the Revit model for the four facades. As seen, QPater has consistently lower maximum and absolute average error than Revit except for the South-Elevation vertical measurement, which was discussed earlier in the context of Figure 4.19.



(a) Average of absolute errors: 0.03

(b) Average of absolute errors: 0.05

(c) Average of absolute errors: 0.03

(d) Average of absolute errors: 0.05

Figure 4.14.: Internal horizontal and vertical errors for West-Elevation facade (feet)

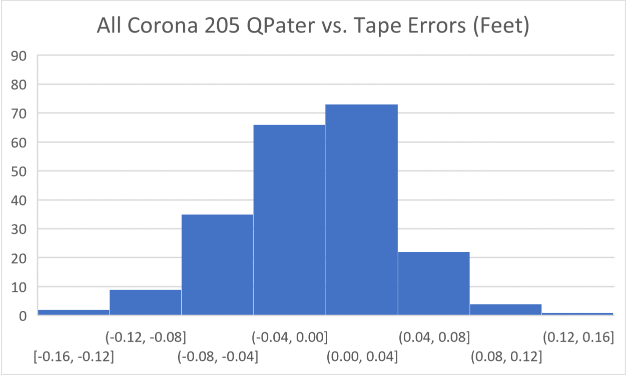


Figure 4.15.: Combined internal QPater tape errors (Average of absolute errors: 0.04)

4.9.2. "Overall" Measurements: Revit

Figure 4.30 illustrates the "overall" Revit errors and Figure 4.31 combines all of those values into a single graph. By inspection, the range of all Revit errors is larger than that of QPater, ranging from -0.21 to 0.26 feet for Revit and from -0.16 to 0.17 feet for QPater. The third and fifth columns of Table 4.7 show maximum and absolute average error in "overall" Revit measurements. As seen, the QPater average error is lower than that of Revit for all facades. Furthermore, the maximum error for QPater is lower than Revit in 3 of the 4 facades.

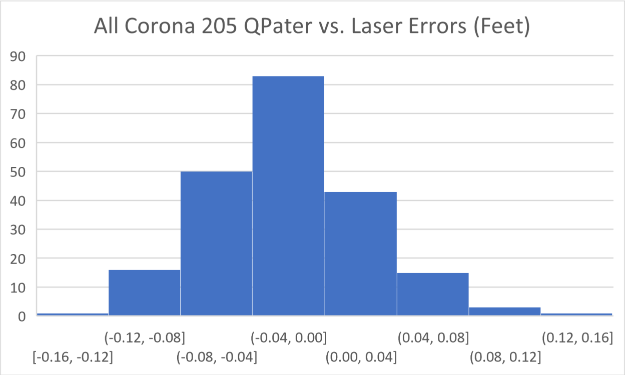


Figure 4.16.: Combined internal QPater laser errors (Average of absolute errors: 0.04)

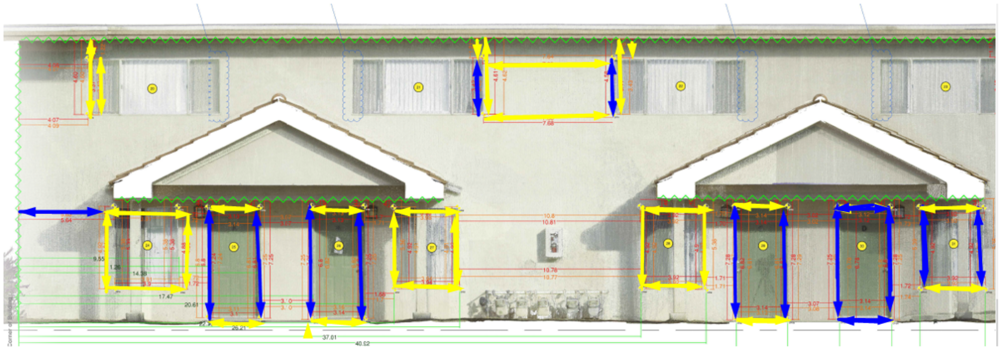


Figure 4.17.: Example of larger than 0.06 feet vertical errors in West-Elevation of Corona 205 shown in blue, less than 0.06 shown in yellow; total of 28 vertical measurements and 37 horizontal measurements

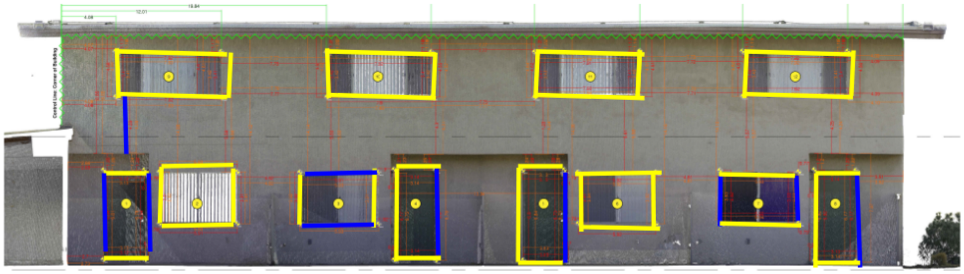
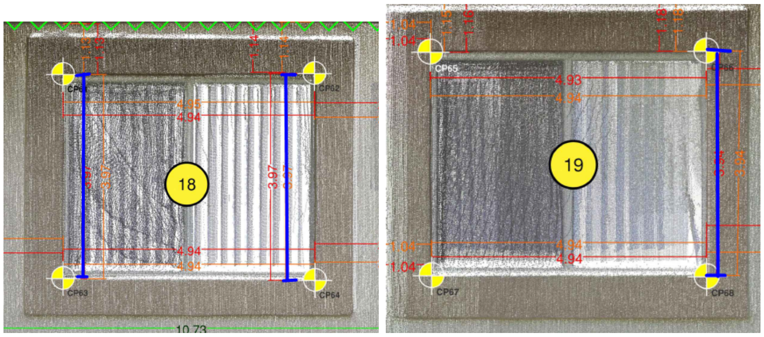


Figure 4.18.: Example of larger than 0.06 feet vertical errors in East-Elevation of Corona 205 shown in blue, less than 0.06 shown in yellow; total of 58 vertical measurements and 53 horizontal measurements

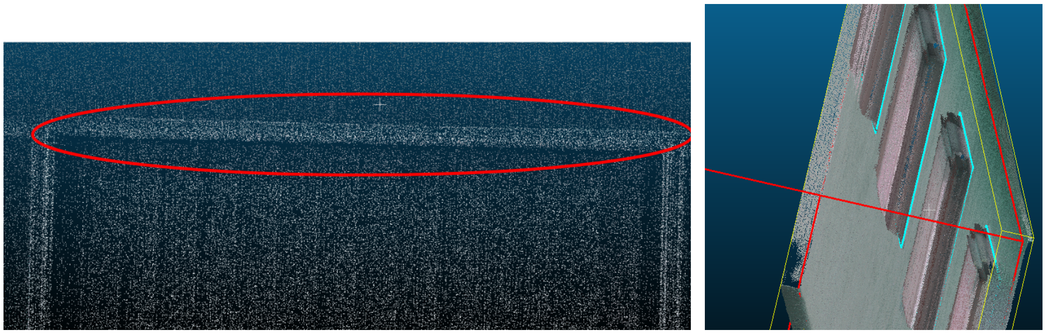
Figures 4.32 and 4.33 show the difference in internal errors for QPater vs. Revit for tape and laser measurement methods respectively. Figure 4.34 shows the same comparison for the "overall" measurements. As seen, QPater achieves overwhelmingly lower error in all three figures. Table 4.8 displays the maximum and average absolute errors, both internal and overall, for both tape and laser meaurements. As seen, QPater



(a)

(b)

Figure 4.19.: Measurements with particularly high error (blue) in South-Elevation

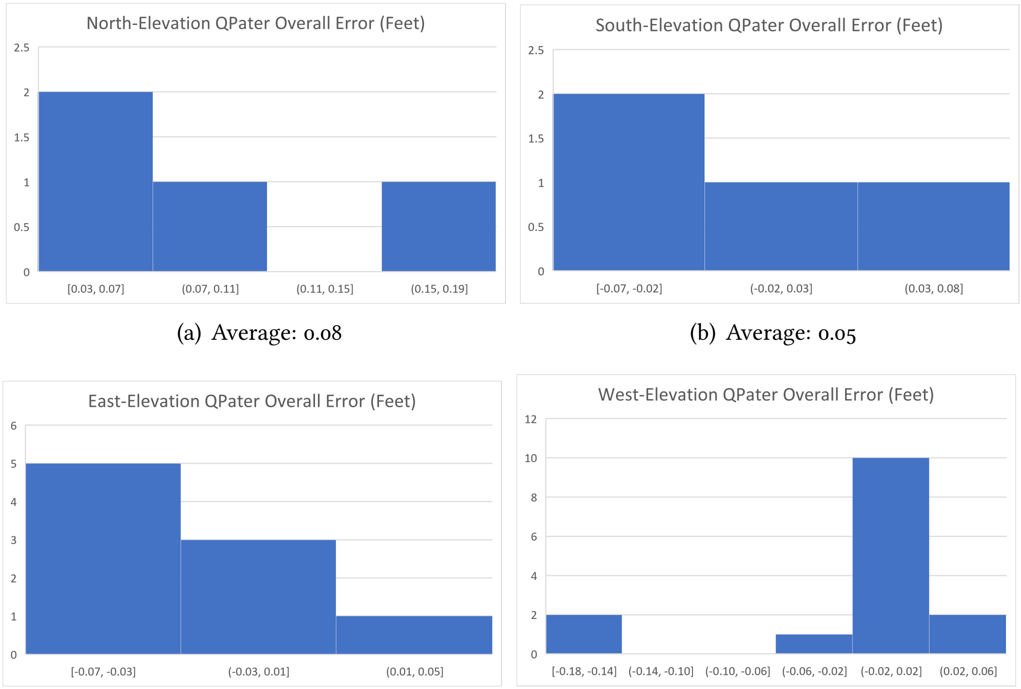


(a)

(b)

Figure 4.20.: Interior points behind a window can make the window boundary detection difficult in the point cloud domain. (a) frontal view of a window point cloud; (b) side view of the window point cloud indicating interior points

yields a decrease in average absolute error of about 33% to 50% and a decrease in maximum error of about 14% to 51%.



(a) Average: 0.08

(b) Average: 0.05

(c) Average: 0.04

(d) Average: 0.04

Figure 4.21.: "Overall" QPater errors for all four facades of Corona 205

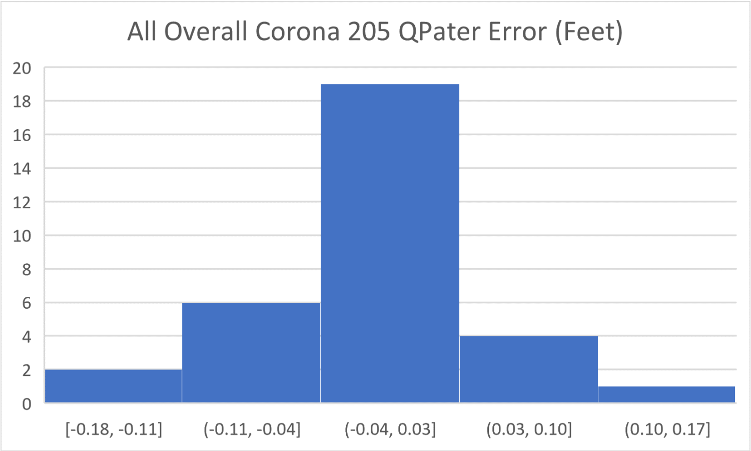


Figure 4.22.: Combined "overall" QPater errors of Corona 205 (average: 0.04 feet)

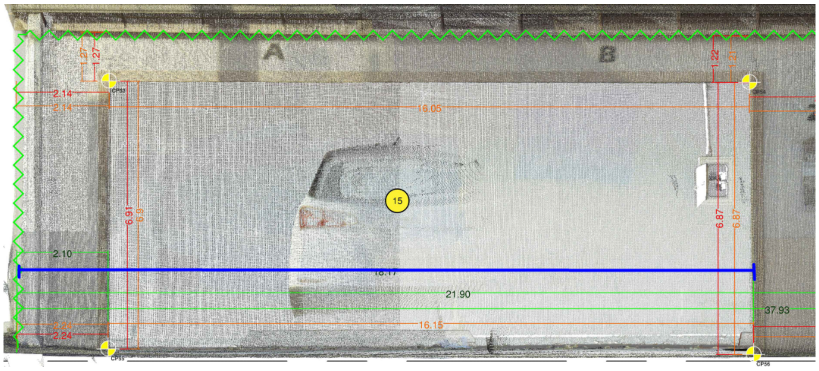


Figure 4.23.: "Overall" measurement with particularly high error (blue) in North-Elevation

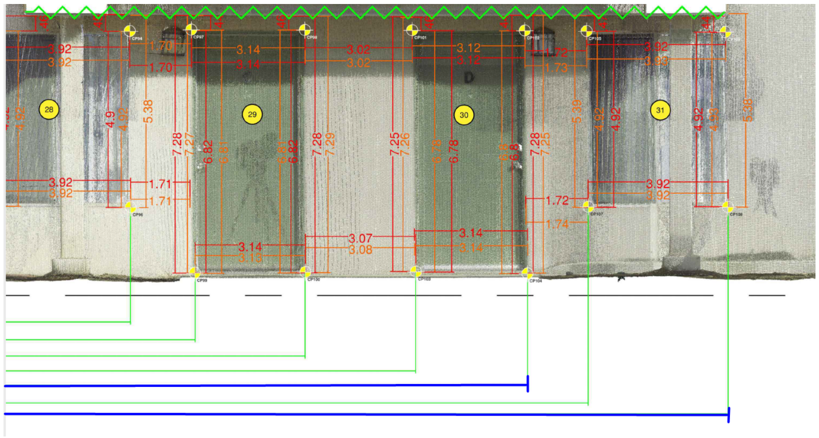


Figure 4.24.: "Overall" measurements with particularly high error (blue) in West-Elevation

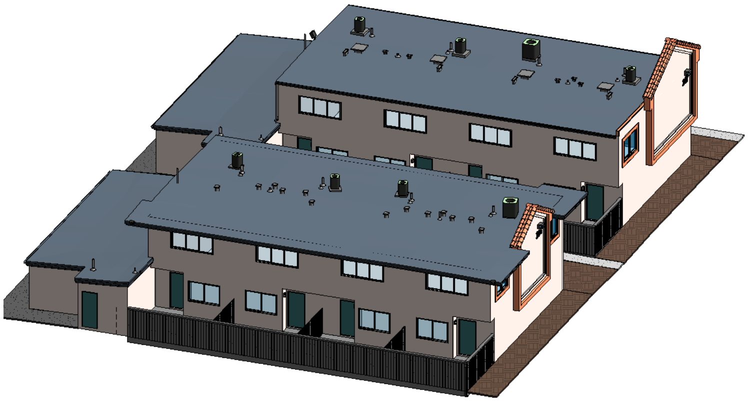
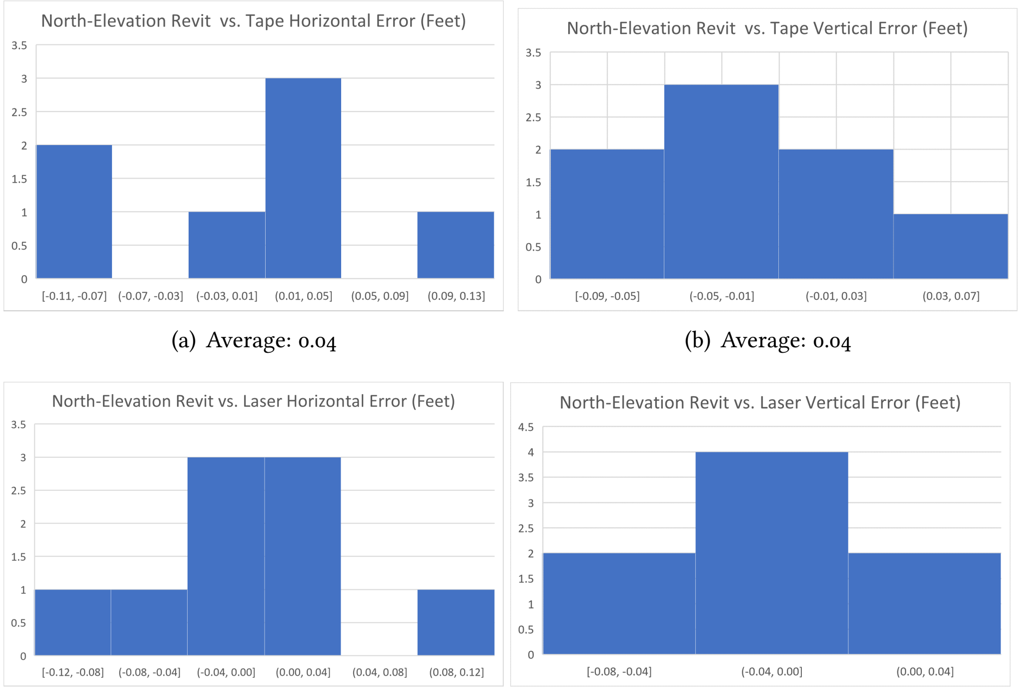


Figure 4.25.: Corona 205 and 217 Revit model



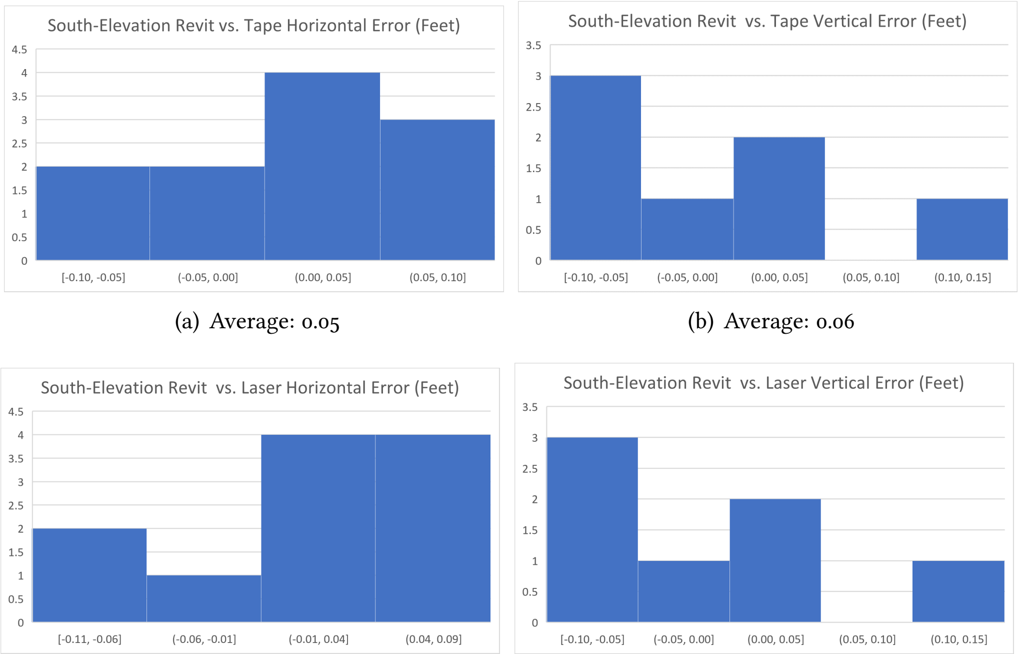
(a) Average: 0.04

(b) Average: 0.04

(c) Average: 0.04

(d) Average: 0.04

Figure 4.26.: Revit: internal horizontal and vertical errors for North-Elevation facade



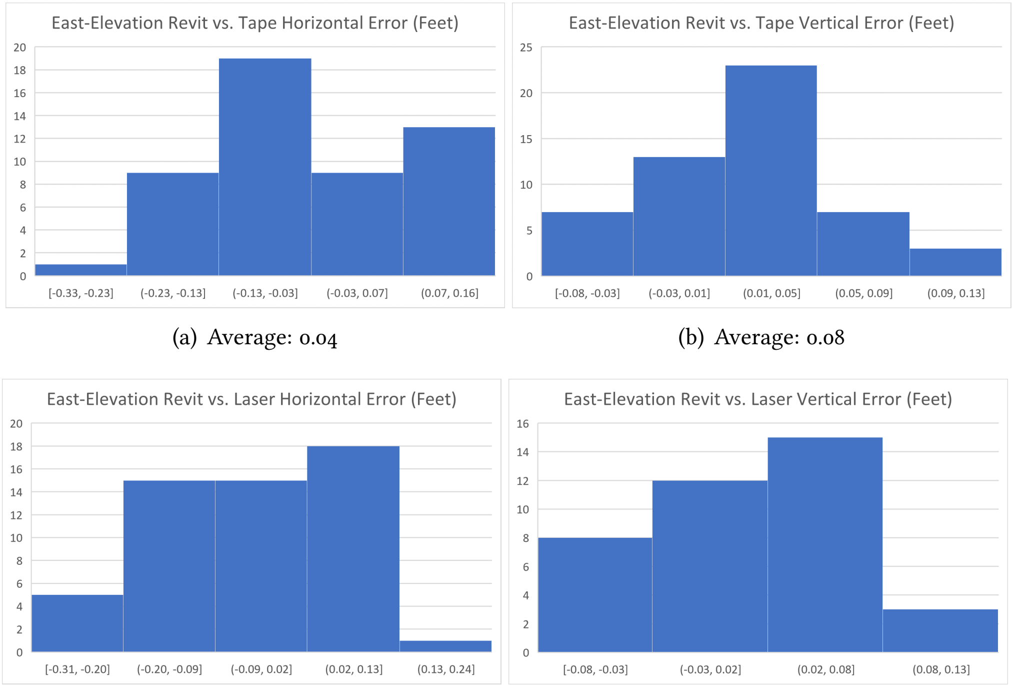
(a) Average: 0.05

(b) Average: 0.06

(c) Average: 0.05

(d) Average: 0.06

Figure 4.27.: Revit: internal horizontal and vertical errors for South-Elevation facade



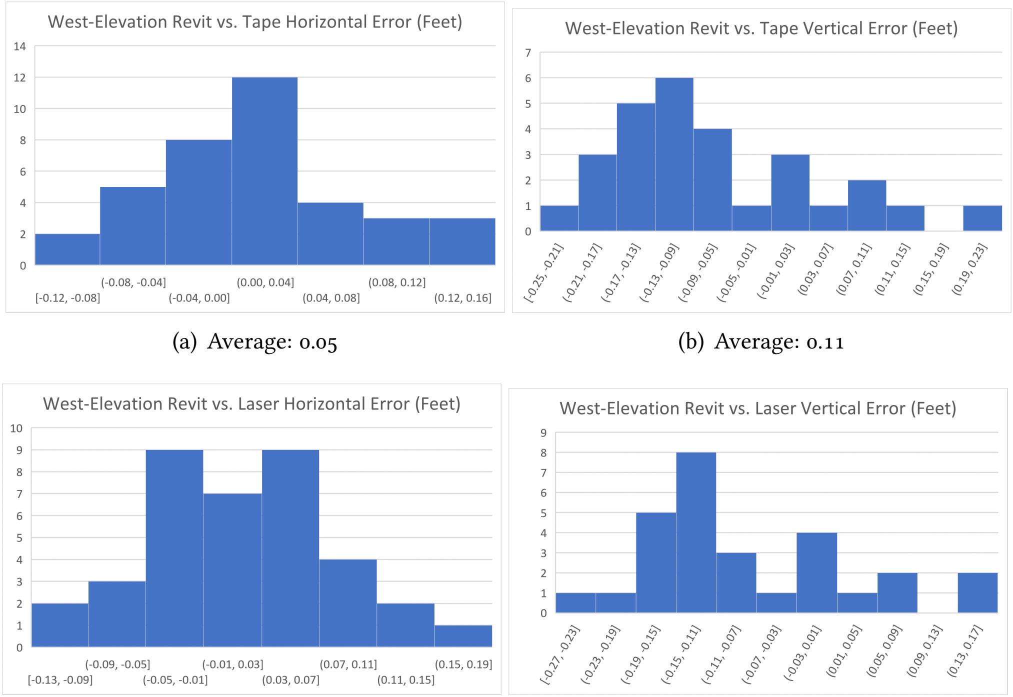
(a) Average: 0.04

(b) Average: 0.08

(c) Average: 0.03

(d) Average: 0.10

Figure 4.28.: Revit: internal horizontal and vertical errors for East-Elevation facade



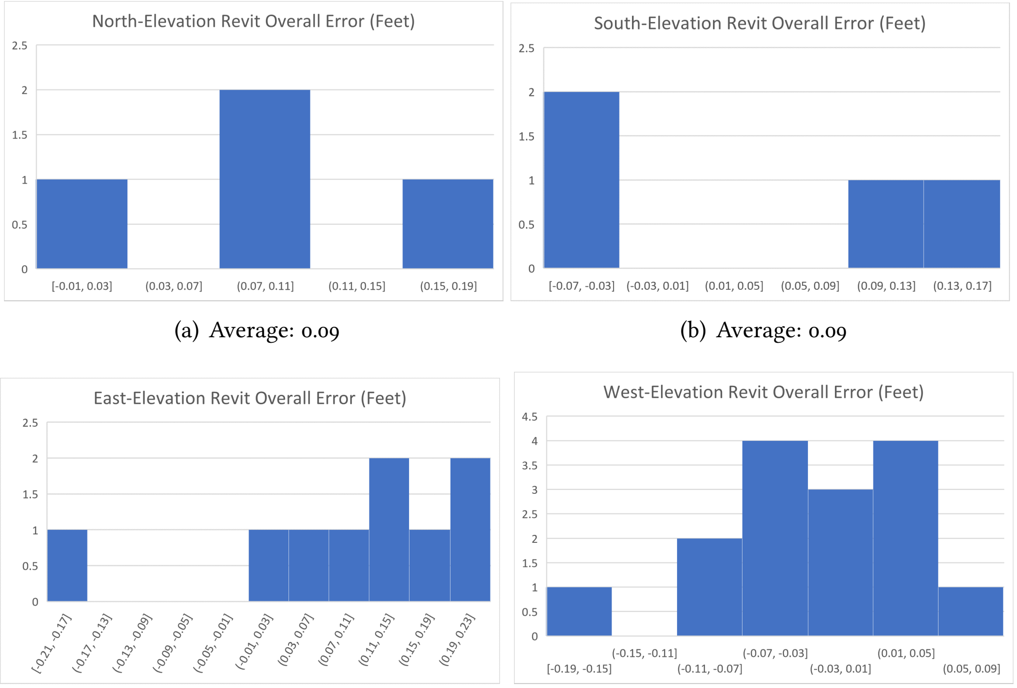
(a) Average: 0.05

(b) Average: 0.11

(c) Average: 0.05

(d) Average: 0.11

Figure 4.29.: Revit: internal horizontal and vertical errors for West-Elevation facade



(a) Average: 0.09

(b) Average: 0.09

(c) Average: 0.13

(d) Average: 0.05

Figure 4.30.: "Overall" Revit errors for all four facades of Corona 205

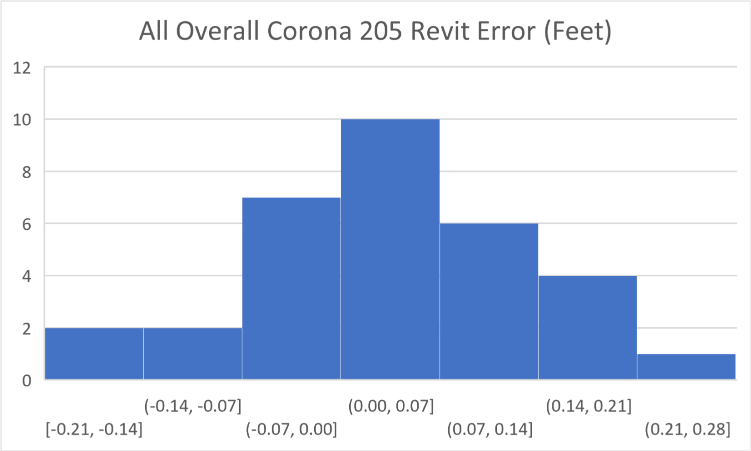


Figure 4.31.: Combined "Overall" Revit errors of Corona 205 (average: 0.06 feet)

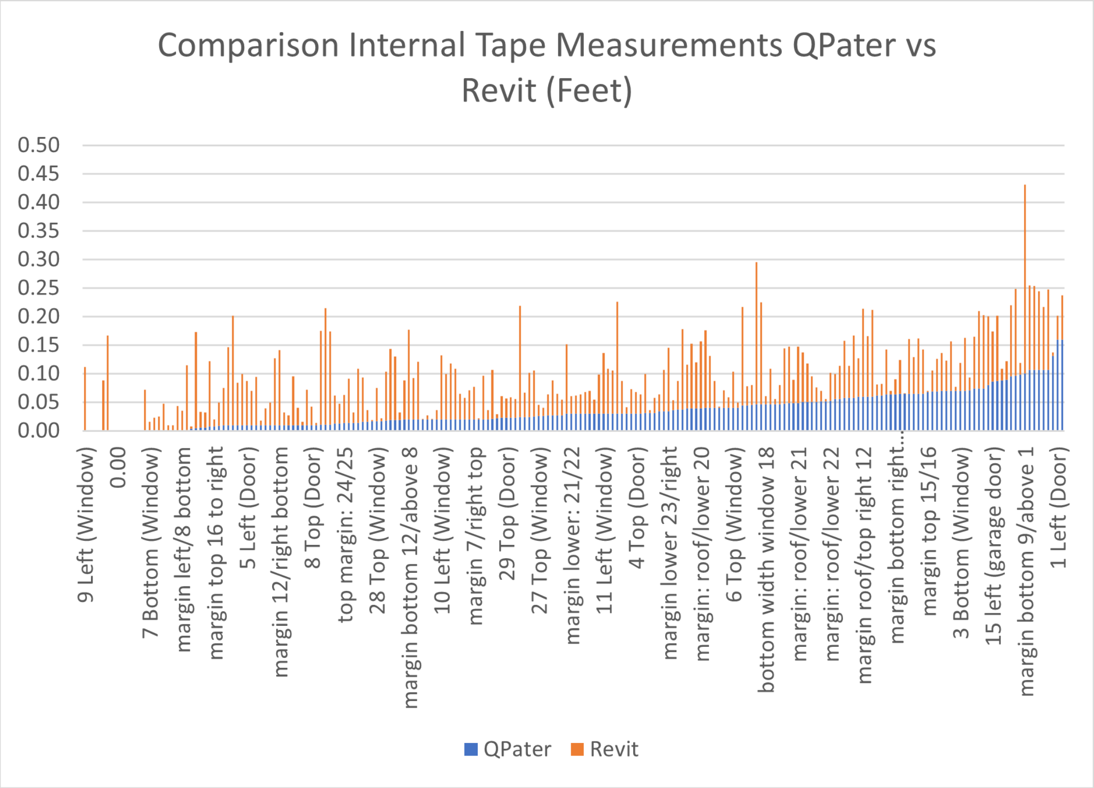


Figure 4.32.: Revit vs. QPater error for internal tape measurements (Feet)

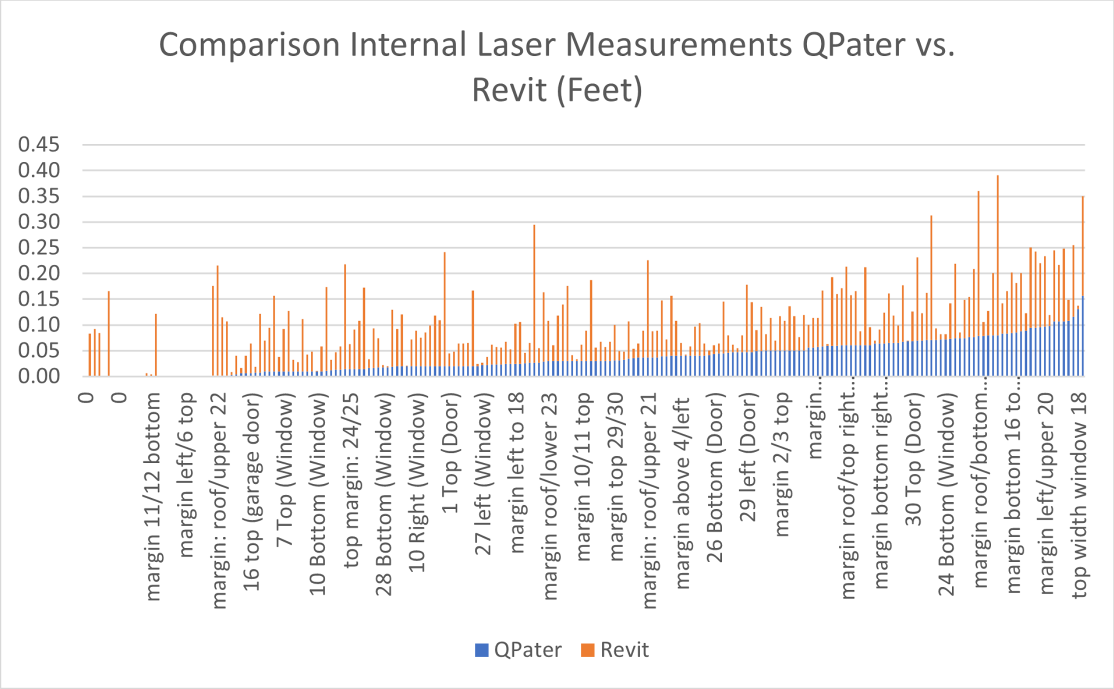


Figure 4.33.: Revit vs. QPater error for internal laser measurements (Feet)

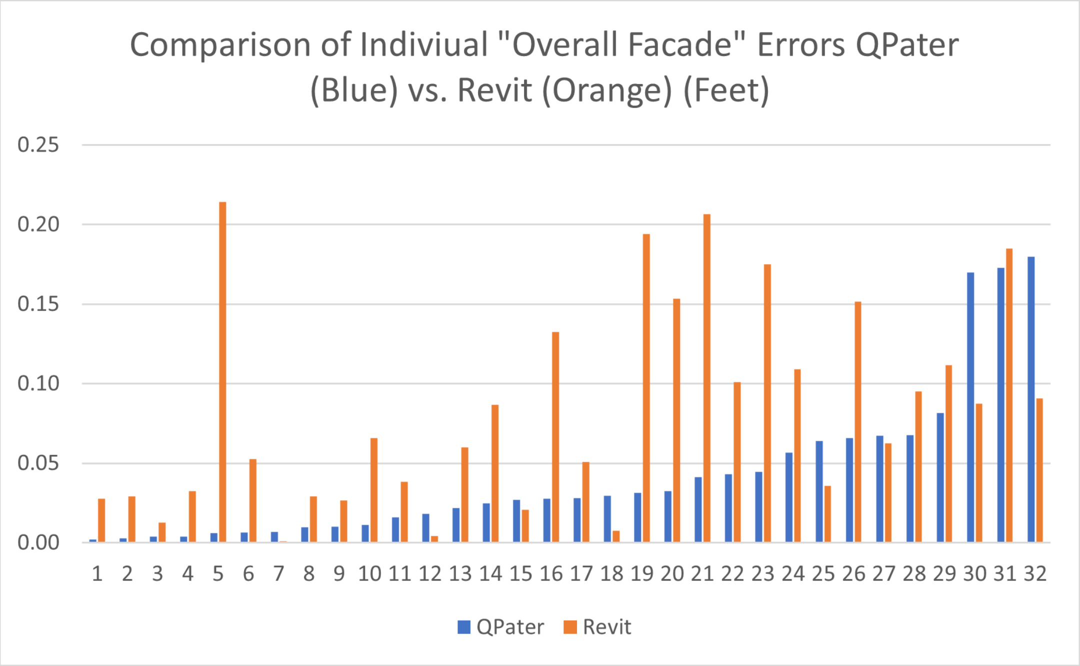


Figure 4.34.: Revit vs. QPater "overall" measurement error (Feet)

Measurement Type	Avg. QPater Error (Feet)	Avg. Revit. Error (Feet)	Max. QPater Error (Feet)	Max. Revit Error (Feet)
North Vertical Tape	0.03	0.04	0.09	0.09
North Vertical Laser	0.03	0.04	0.08	0.08
North Horizontal Tape	0.03	0.04	0.07	0.11
North Horizontal Laser	0.04	0.04	0.08	0.12
Combined North Tape	**0.03**	**0.04**	**0.09**	**0.11**
Combined North Laser	**0.04**	**0.04**	**0.08**	**0.12**
South Vertical Tape	0.08	0.06	0.13	0.12
South Vertical Laser	0.08	0.06	0.13	0.12
South Horizontal Tape	0.05	0.05	0.07	0.10
South Horizontal Laser	0.05	0.05	0.07	0.11
Combined South Tape	**0.06**	**0.06**	**0.13**	**0.12**
Combined South Laser	**0.06**	**0.06**	**0.13**	**0.12**
East Vertical Tape	0.04	0.08	0.16	0.33
East Vertical Laser	0.04	0.10	0.16	0.31
East Horizontal Tape	0.02	0.04	0.07	0.13
East Horizontal Laser	0.03	0.05	0.06	0.12
Combined East Tape	**0.03**	**0.07**	**0.16**	**0.33**
Combined East Laser	**0.04**	**0.08**	**0.16**	**0.31**
West Vertical Tape	0.05	0.11	0.11	0.25
West Vertical Laser	0.05	0.11	0.12	0.27
West Horizontal Tape	0.03	0.05	0.11	0.15
West Horizontal Laser	0.03	0.05	0.10	0.16
Combined West Tape	**0.04**	**0.08**	**0.11**	**0.25**
Combined West Laser	**0.04**	**0.08**	**0.12**	**0.27**

Table 4.6.: Comparison of maximum and average absolute internal errors of Revit and QPater with hand measurements

Facade	Average QPater (Feet)	Average Revit (Feet)	Max. QPater (Feet)	Max. Revit (Feet)
North-Elevation	0.08	0.09	0.17	0.15
South-Elevation	0.05	0.09	0.14	0.15
East-Elevation	0.04	0.13	0.07	0.21
West-Elevation	0.04	0.05	0.18	0.19

Table 4.7.: Comparison of average absolute and maximum "overall" error Revit vs. QPater

Measurement Type	Average QPater (Feet)	Average Revit (Feet)	Max. QPater (Feet)	Max. Revit (Feet)
"Overall"	0.04	0.08	0.18	0.21
Internal Tape	0.04	0.06	0.16	0.33
Internal Laser	0.04	0.06	0.16	0.31

Table 4.8.: Comparison of average absolute internal errors of Revit and QPater with Hand Measurements over all measurements

5. Conclusion and Future Work

We developed a detailed pipeline for 3D point cloud processing for converting a raw point cloud laser scan of a building into a 3D polygonal, wireframe representation. For this project, we were able to develop a region-growing based segmentation algorithm as well as novel methods for creating polygonal representations of point clouds. The wireframe representations were accurate to within an average error of 0.02 to 0.06 feet for both hand measurements with tape measures and laser, which is reasonable for margin of error provided ±0.01 feet). Our method outperformed the existing 3D modeling software Revit, showing decreases in error across all measurements and decreasing the average error by 33% to 50%

One shortcoming of our method is that it is highly dependent on input parameters and thresholds. Changes in the search radius, k-nearest neighbor number during facade segmentation, and the minimum size threshold for retaining facades all have different effects on the number and quality of the resulting facades and their polygonization. Therefore, there is room to improve our method to be less reliant on thresholds. We will also look to improve our method by adding the ability to polygonize overhanging roofs. Another direction to take our workflow in the future is adding the ability to handle facades with curved shapes, although such buildings are rare to come across. We hope to further improve our method by automating the pruning step. Since manual pruning can be tedious if the facade segmentation generates too many segments, we will look for methods to filter the so-called "junk" facades. Finally, we plan to investigate how to improve the representation of facades from planar models to more accurate non-planar representations such as spline or Bezier curves [2, 13].

Bibliography

1. A. Adam, E. Chatzilari, S. Nikolopoulos, and I. Kompatsiaris. “H-RANSAC: A hybrid point cloud segmentation combining 2D and 3D data”. In: *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.* 4:2, 2018, pp. 1–8.
2. N. M. Aziz, R. Bata, and S. Bhat. “Bezier surface/surface intersection”. In: *IEEE Computer Graphics and Applications* 10:1, 1990, pp. 50–58.
3. H. Boulaassal, T. Landes, P. Grussenmeyer, and F. Tarsha-Kurdi. “Automatic segmentation of building facades using terrestrial laser data”. In: *ISPRS Workshop on Laser Scanning 2007 and SilviLaser 2007.* 2007, pp. 65–70.
4. M. Carlberg, P. Gao, G. Chen, and A. Zakhor. “Classifying urban landscape in aerial LiDAR using 3D shape analysis”. In: *2009 16th IEEE International Conference on Image Processing (ICIP).* IEEE. 2009, pp. 1701–1704.
5. M. Dai, W. O. Ward, G. Meyers, D. D. Tingley, and M. Mayfield. “Residential building facade segmentation in the urban environment”. In: *Building and Environment* 199, 2021, pp. 107–921.
6. J. Elseberg, S. Magnenat, R. Siegwart, and A. Nüchter. “Comparison of nearest-neighbor-search strategies and implementations for efficient shape registration”. In: *Journal of Software Engineering for Robotics* 3:1, 2012, pp. 2–12.
7. R. Fathalla and G. Vogiatzis. “A deep learning pipeline for semantic facade segmentation”. In: *BMVC.* 2017.
8. J. Femiani, W. R. Para, N. Mitra, and P. Wonka. “Facade segmentation in the wild”. In: *arXiv preprint arXiv:1805.08634*, 2018.
9. J.-F. Lalonde, N. Vandapel, D. F. Huber, and M. Hebert. “Natural terrain classification using three-dimensional LiDAR data for ground robot mobility”. In: *Journal of Field Robotics* 23:10, 2006, pp. 839–861.
10. X. Liu, Y. Zhang, X. Ling, Y. Wan, L. Liu, and Q. Li. “TopoLAP: topology recovery for building reconstruction by deducing the relationships between linear and planar primitives”. In: *Remote Sensing* 11:11, 2019, pp. 1372–1390.
11. K. Rahmani and H. Mayer. “High quality facade segmentation based on structured random forest, region proposed network and rectangular Fitting”. In: *ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences* 4:2, 2018.

12. M. Schmitz and H. Mayer. "A convolutional network for semantic facade segmentation and interpretation". In: *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences* 41, 2016, pp. 709–715.
13. I. Schoenberg. "Cardinal interpolation and spline functions". In: *Journal of Approximation Theory* 2:2, 1969, pp. 167–206.
14. T. Tsenoglou, N. Vassilas, and D. Ghazanfarpour. "Robust line detection in images of building facades using region-based weighted hough transform". In: *2012 16th Panhellenic Conference on Informatics*. IEEE. 2012, pp. 333–338.
15. A.-V. Vo, L. Truong-Hong, D. F. Laefer, and M. Bertolotto. "Octree-based region growing for point cloud segmentation". In: *ISPRS Journal of Photogrammetry and Remote Sensing* 104, 2015, pp. 88–100.
16. H. Woo, E. Kang, S. Wang, and K. H. Lee. "A new segmentation method for point cloud data". In: *International Journal of Machine Tools and Manufacture* 42:2, 2002, pp. 167–178.
17. M. WuDunn, J. Dunn, and A. Zakhor. "Point cloud segmentation using RGB drone imagery". In: *2020 IEEE International Conference on Image Processing (ICIP)*. IEEE. 2020, pp. 2750–2754.
18. Y. Xu, W. Yao, L. Hoegner, and U. Stilla. "Segmentation of building roofs from airborne LiDAR point clouds using robust voxel-based region growing". In: *Remote Sensing Letters* 8:11, 2017, pp. 1062–1071.
19. S. I. Zolanvari and D. F. Laefer. "Slicing method for curved façade and window extraction from point clouds". In: *ISPRS Journal of Photogrammetry and Remote Sensing* 119, 2016, pp. 334–346.
20. S. I. Zolanvari, D. F. Laefer, and A. S. Natanzi. "Three-dimensional building façade segmentation and opening area detection from point clouds". In: *ISPRS Journal of Photogrammetry and Remote Sensing* 143, 2018, pp. 134–149.

A. Appendix

A.1. Comparison of Errors with Differing Number of RANSAC Iterations

The following section consists of a table of measurements comparing the errors between different iterations using different numbers of RANSAC iterations.

Hand Measurements	Errors (Feet) iterations: 100 0	Errors (Feet) iterations: 10000	Errors (Feet) num-iterations: 100000
9.55	9.61	9.55	9.61 Feet
11.26	11.29	11.25	11.35
14.38	14.40	14.37	14.50
17.47	17.49	17.45	17.59
20.61	20.60	20.60	20.70
22.30	22.38	22.31	22.44
26.21	26.28	26.21	26.35
37.01	37.11	37.01	37.11
40.92	41.04	40.94	41.01
42.60	42.72	42.62	42.72
45.73	45.83	45.70	45.83
48.79	48.85	48.79	48.92
52.01	51.94	51.84	51.94
53.62	53.74	53.64	53.71
57.66	57.61	57.48	57.58

Table A.1.: Overall Facade Measurements of West-Elevation for different # of RANSAC Iterations (Feet)

A.2. Tables for QPater "Internal" and "Overall" Measurements

The following section contains "internal error measurements for the QPater pipeline. Some of the measurements appear to be missing from the table and this is because they were not performed with the original hand measurements as some of the measurements would have been too dangerous or infeasible to perform.

Errors (Feet) for num-iterations:1000	Errors (Feet) for num-iterations:10000	Errors (Feet) for num-iterations:100000
-0.06	0.00	-0.06
-0.03	0.01	-0.09
-0.02	0.01	-0.12
-0.02	0.02	-0.12
0.01	0.01	-0.09
-0.08	-0.01	-0.14
-0.07	0.00	-0.14
-0.10	0.00	-0.10
-0.12	-0.02	-0.09
-0.12	-0.02	-0.12
-0.10	0.03	-0.10
-0.06	0.00	-0.13
0.07	0.17	0.07
-0.12	-0.02	-0.09
0.05	0.18	0.08

Table A.2.: "Overall" errors for different # of RANSAC iterations (Feet)

Type	Tape	Laser	Qpater	Tape Error (Feet)	Laser Error (Feet)
15 left (garage door)	6.91	6.9	6.82	-0.09	-0.08
15 right (garage door)	6.87	6.87	6.82	-0.05	-0.05
margin roof/15 top left	1.27	1.27	1.25	-0.02	-0.02
margin roof/15 top right	1.22	1.21	1.25	0.03	0.04
16 left (garage door)	6.83	6.84	6.82	-0.01	-0.02
16 right (garage door)	6.86	6.86	6.86	0.00	0.00
margin roof/16 top left	1.22	1.21	1.25	0.03	0.04
margin roof/16 top right	1.22	1.22	1.25	0.03	0.03

Table A.3.: North-Elevation QPater Vertical Errors (Feet)

Type	Tape	Laser	Qpater	Tape Error (Feet)	Laser Error (Feet)
Margin top left to 15	2.14	2.14	0.65	-0.01	-0.01
margin bottom left to 15	2.24	2.24	0.69	0.02	0.02
15 Top (garage door)		16.05	4.91		0.06
15 Bottom (garage door)	16.15		4.94	0.06	
margin top 15/16	3.71	3.71	1.11	-0.07	-0.07
margin bottom 15/16	3.68	3.68	1.1	-0.07	-0.07
16 top (garage door)		16.05	4.89		-0.01
16 bottom (garage door)		15.97	4.88		0.04
margin top 16 to right	2.14	2.15	0.65	-0.01	-0.02
margin bottom 16 to right	2.23	2.25	0.66	-0.06	-0.08

Table A.4.: North-Elevation QPater Horizontal Errors (Feet)

Type	Tape	Laser	Qpater	Tape Error (Feet)	Laser Error (Feet)
margin roof/left window18	1.13	1.13	1.08	-0.05	-0.05
margin roof/right window 18	1.14	1.14	1.08	-0.06	-0.06
roof/bottom left 18	3.97	3.97	4.10	0.13	0.13
roof/bottom right 18	3.97	3.97	4.07	0.10	0.10
left margin roof/window 19	1.15	1.16	1.12	-0.03	-0.04
right margin roof/window 19	1.18	1.18	1.12	-0.06	-0.06
height window 19	3.94	3.94	4.04	0.10	0.10

Table A.5.: South-Elevation QPater Vertical Errors (Feet)

Type	Tape	Laser	Qpater	Tape Error (Feet)	Laser Error (Feet)
margin left to 18	2.83	2.83	2.85	0.02	0.02
bottom width window 18	4.94	4.94	4.99	0.05	0.05
top width window 18	4.94	4.95	4.99	0.05	0.04
margin top 18/middle	2.99	2.99	2.92	-0.07	-0.07
margin bottom 18/middle	3.01	3.02	2.95	-0.06	-0.07
margin middle/top 19	1.04	1.04	0.98	-0.06	-0.06
margin middle/bottom 19	1.04	1.04	0.98	-0.06	-0.06
top width window 19	4.94	4.94	4.99	0.05	0.05
bottom width window 19	4.94	4.94	4.99	0.05	0.05
margin top right 19/right	2.82	2.82	2.76	-0.06	-0.06
margin bottom right 19/right	2.82	2.82	2.76	-0.06	-0.06

Table A.6.: South-Elevation QPater Horizontal Errors (Feet)

A.3. Tables for Revit "Internal" and "Overall" Measurements

The following section contains table for Revit "Internal" and "Overall" measurements.

Type	QPater	Tape	Laser	Tape Error (Feet)	Laser Error (Feet)
margin roof/above 1	8.83	8.85	8.85	-0.02	-0.02
margin bottom 9/above 1	4.18	4.28	4.26	-0.10	-0.08
9 Left (Window)	3.40	3.40	3.41	0.00	-0.01
9 Right (Window)	3.40	3.41	3.42	-0.01	-0.02
margin right roof/bottom 9	4.56	4.60	4.61	-0.04	-0.05
margin left roof/top 9	1.18	1.17	1.18	0.01	0.00
margin right roof/top 9	1.21	1.17	1.20	0.04	0.01
margin roof/bottom 2 left	14.01		14.08		-0.07
1 Left (Door)	6.78	6.94	6.81	-0.16	-0.03
1 Right (Door)	6.78	6.94	6.78	-0.16	0.00
margin roof/bottom 2 right	14.04		14.12		-0.08
margin roof/bottom 12	4.56	4.60	4.62	-0.04	-0.06
10 Left (Window)	3.42	3.40	3.42	0.02	0.00
10 Right (Window)	3.40	3.40	3.42	0.00	-0.02
11 Left (Window)	3.38	3.41	3.43	-0.03	-0.05
11 Right (Window)	3.38	3.41	3.42	-0.03	-0.04
12 Left (Window)	3.41	3.42	3.43	-0.01	-0.02
12 Right (Window)	3.43	3.40	3.41	0.03	0.02
2 Left Window)	7.94	7.90	7.93	0.04	0.01
2 Right (Window)	5.88	5.91	5.93	-0.03	-0.05
3 Left (Window)	5.88	5.91	5.93	-0.03	-0.05
3 Right (Window)	5.98	5.91	5.93	0.07	0.05
6 Left (Window)	5.98	5.91	5.93	0.07	0.05
6 Right (Window)	5.88	5.92	5.92	-0.04	-0.04
7 Left (Window)	5.88	5.92	5.93	-0.04	-0.05
7 Right (Window)	5.92	5.92	5.93	0.00	-0.01
4 Left (Door)	5.92	5.92	5.94	0.00	-0.02
4 Right (Door)	3.11	3.14	3.14	-0.03	-0.03
5 Left (Door)	3.11	3.14	3.13	-0.03	-0.02
5 Right (Door)	3.15	3.15	3.13	0.00	0.02
8 Left (Door)	3.15	3.15	3.13	0.00	0.02
8 Right (Door)	3.13	3.14	3.14	-0.01	-0.01
margin bottom 12/top 7	3.13	3.13	3.14	0.00	-0.01
margin roof/bottom 3	4.57	4.62	4.62	-0.05	-0.05
margin bottom 10/top 3	7.74	7.77	7.77	-0.03	-0.03
margin roof/10 bottom left	7.74	7.75	7.77	-0.01	-0.03
margin roof/10 top left	7.73	7.72	7.73	0.01	0.00
margin roof/11 top left	7.73	7.71	7.73	0.02	0.00

Table A.7.: East-Elevation QPater Vertical Errors (Feet)

Type	Qpater	Tape	Laser	Tape Error (Feet)	Laser Error (Feet)
margin left/9 top	4.09	4.07	4.07	0.02	0.02
margin left/9 bottom	4.10	4.08	4.08	0.02	0.02
9 Top (Window)	7.90	7.92	7.93	-0.02	-0.03
9 Bottom (Window)	7.90	7.92	7.93	-0.02	-0.03
margin left/1 top	2.63	2.65	2.66	-0.02	-0.03
margin bottom/1 bottom	2.67	2.70	2.73	-0.03	-0.06
1 Top (Door)	3.15	3.12	3.13	0.03	0.02
1 Botom (Door)	3.15	3.12	3.13	0.03	0.02
margin 1/right top	0.25	0.27		-0.02	
margin 1/right bottom	0.22	0.20		0.02	
margin 2/left top	0.81	0.78		0.03	
margin 2/left bottom	0.83	0.78		0.05	
margin 9/10 top	7.82	7.75	7.77	0.07	0.05
margin 9/10 bottom	7.82	7.77	7.76	0.05	0.06
10 Top (Window)	7.94	7.92	7.90	0.02	0.04
10 Bottom (Window)	7.94	7.90	7.93	0.04	0.01
11 Top (Window)	7.91	7.92	7.90	-0.01	0.01
11 Bottom (Window)	7.91	7.90	7.93	0.01	-0.02
12 Top (Window)	7.94	7.92	7.90	0.02	0.04
12 Bottom (Window)	7.94	7.90	7.93	0.04	0.01
2 Top (Window)	5.88	5.91	5.93	-0.03	-0.05
2 Bottom (Window)	5.88	5.91	5.93	-0.03	-0.05
3 Top (Window)	5.98	5.91	5.93	0.07	0.05
3 Bottom (Window)	5.98	5.91	5.93	0.07	0.05
6 Top (Window)	5.88	5.92	5.92	-0.04	-0.04
6 Bottom (Window)	5.88	5.92	5.93	-0.04	-0.05
7 Top (Window)	5.92	5.92	5.93	0.00	-0.01
7 Bottom (Window)	5.92	5.92	5.94	0.00	-0.02
4 Top (Door)	3.11	3.14	3.14	-0.03	-0.03
4 Bottom (Door)	3.11	3.14	3.13	-0.03	-0.02
5 Top (Door)	3.15	3.15	3.13	0.00	0.02
5 Bottom (Door)	3.15	3.15	3.13	0.00	0.02
8 Top (Door)	3.13	3.14	3.14	-0.01	-0.01
8 Bottom (Door)	3.13	3.13	3.14	0.00	-0.01
margin 2/3 top	4.57	4.62	4.62	-0.05	-0.05
margin 10/11 top	7.74	7.77	7.77	-0.03	-0.03
margin 10/11 bottom	7.74	7.75	7.77	-0.01	-0.03
margin 11/12 top	7.73	7.72	7.73	0.01	0.00
margin 11/12 bottom	7.73	7.71	7.73	0.02	0.00

Table A.8.: East-Elevation QPater Horizontal Errors (Feet)

Type	Tape	Laser	Qpater	Tape Error (Feet)	Laser Error (Feet)
margin: roof/lower 20	4.60	4.60	4.56	-0.04	-0.04
margin: roof/upper 20	1.20	1.22	1.25	0.05	0.03
20 height (Window)	3.37	3.42	3.35	-0.02	-0.07
21 height (Window)	3.42	3.43	3.31	-0.11	-0.12
margin: roof/upper 21	1.20	1.21	1.25	0.05	0.04
margin: roof/lower 21	4.61	4.62	4.66	0.05	0.04
margin: roof/lower 22	4.58	4.60	4.53	-0.05	-0.07
22 height (Window)	3.41	3.43	3.35	-0.06	-0.08
margin: roof/upper 22	1.17	1.18	1.18	0.01	0.00
23 height	3.42	3.43	3.35	-0.07	-0.08
margin roof/upper 23	1.17	1.17	1.18	0.01	0.01
margin roof/lower 23	4.58	4.59	4.56	-0.02	-0.03
24 left (Window)	4.91	4.92	4.89	-0.02	-0.03
24 right (Window)	4.88	4.90	4.89	0.01	-0.01
25 left (Door)	6.80	6.80	6.69	-0.11	-0.11
25 right (Door)	6.80	6.80	6.69	-0.11	-0.11
26 left (Door)	6.80	6.82	6.73	-0.07	-0.09
26 right (Door)	6.80	6.82	6.73	-0.07	-0.09
27 left (Window)	4.92	4.91	4.89	-0.03	-0.02
27 right (Window)	4.92	4.91	4.89	-0.03	-0.02
28 left (Window)	4.92	4.92	4.92	0.00	0.00
28 right (Window)	4.90	4.92	4.92	0.02	0.00
29 left (Door)	6.82	6.81	6.86	0.04	0.05
29 right (Door)	6.82	6.81	6.82	0.00	0.01
30 left (Door)	6.78	6.78	6.69	-0.09	-0.09
30 right (Door)	6.80	6.80	6.69	-0.11	-0.11
31 left (Window)	4.92	4.92	4.86	-0.06	-0.06
31 right (Window)	4.92	4.93	4.86	-0.06	-0.07

Table A.9.: West-Elevation QPater Vertical Errors (Feet)

Type	Tape	Laser	Qpater	Tape Error (Feet)	Laser Error (Feet)
margin left/upper 20	4.06	4.07	4.17	0.11	0.10
margin left/lower 20	4.07	4.09	4.17	0.10	0.08
margin upper: 20/21	7.68	7.64	7.71	0.03	0.07
margin lower: 20/21	7.68	7.68	7.71	0.03	0.03
margin upper 23/right	4.07	4.07	4.04	-0.03	-0.03
margin lower 23/right	4.07	4.08	4.04	-0.03	-0.04
margin left/24	5.64	5.63	5.71	0.07	0.08
24 top width (Window)	3.90	3.91	3.84	-0.06	-0.07
24 bottom width (Window)	3.90	3.91	3.84	-0.06	-0.07
top margin: 24/25	1.72	1.72	1.71	-0.01	-0.01
bottom margin: 24/25	1.72	1.72	1.71	-0.01	-0.01
25 Top (Door)	3.13	3.14	3.12	-0.01	-0.02
25 Bottom (Door)	3.10	3.11	3.12	0.02	0.01
top margin 25/26	3.07	3.07	3.08	0.01	0.01
bottom margin 25/26	3.10	3.10	3.08	-0.02	-0.02
26 Top (Door)	3.13	3.13	3.12	-0.01	-0.01
26 Bottom (Door)	3.14	3.16	3.12	-0.02	-0.04
margin top 26/27	1.70	1.71	1.71	0.01	0.00
margin bottom: 26/27	1.68	1.70	1.71	0.03	0.01
27 Top (Window)	3.93	3.93	3.90	-0.03	-0.03
27 Bottom (Window)	3.94	3.94	3.90	-0.04	-0.04
margin top 27/28	10.81	10.80	10.79	-0.02	-0.01
margin bottom 27/28	10.78	10.77	10.79	0.01	0.02
28 Top (Window)	3.92	3.92	3.94	0.02	0.02
28 Bottom (Window)	3.92	3.92	3.94	0.02	0.02
margin top 28/29	1.70	1.70	1.67	-0.03	-0.03
margin bottom 28/29	1.71	1.71	1.67	-0.04	-0.04
29 Top (Door)	3.14	3.14	3.12	-0.02	-0.02
29 Bottom (Door)	3.14	3.13	3.12	-0.02	-0.01
margin top 29/30	3.02	3.02	3.05	0.03	0.03

Table A.10.: West-Elevation QPater Horizontal Errors (Feet)

Hand Measurements (Feet)	QPater (Feet)	Error (Feet)
2.1	2.13	0.03
18.17	18.34	0.17
21.9	21.98	0.08
37.93	37.96	0.03

Table A.11.: Overall QPater Errors for North Elevation (Feet)

Hand Measurements (Feet)	QPater (Feet)	Error (Feet)
10.75	10.76	0.01
10.73	10.79	0.06
8.77	8.73	-0.04
8.76	8.69	-0.07

Table A.12.: Overall QPater Errors for South Elevation (Feet)

Hand Measurements (Feet)	Qpater (Feet)	Error (Feet)
4.08	4.04	-0.04
12.01	11.94	-0.07
19.84	19.78	-0.06
27.75	27.72	-0.03
35.53	35.50	-0.03
43.43	43.47	0.04
51.15	51.08	-0.07
59.05	59.02	-0.03
63.15	63.16	0.01

Table A.13.: Overall QPater Errors for East Elevation (Feet)

Hand Measurements (Feet)	Qpater (Feet)	Error (Feet)
9.55	9.55	0.00
11.26	11.25	-0.01
14.38	14.37	-0.01
17.47	17.45	-0.02
20.61	20.60	-0.01
22.30	22.31	0.01
26.21	26.21	0.00
37.01	37.01	0.00
40.92	40.94	0.02
42.60	42.62	0.02
45.73	45.70	-0.03
48.79	48.79	0.00
52.01	51.84	-0.17
53.62	53.64	0.02
57.66	57.48	-0.18

Table A.14.: Overall QPater Errors for West Elevation (Feet)

Type	Tape	Laser	Revit	Tape Error (Feet)	Laser Error (Feet)
Margin top left to 15	2.14	2.14	2.25	0.11	0.11
margin bottom left to 15	2.24	2.24	2.28	0.04	0.04
15 Top (garage door)		16.05	16.05		0.00
15 Bottom (garage door)	16.15		16.04	-0.11	
margin top 15/16	3.71	3.71	3.71	0.00	0.00
margin bottom 15/16	3.68	3.68	3.70	0.02	0.02
16 top (garage door)		16.05	15.99		-0.06
16 bottom (garage door)		15.97	15.99		0.02
margin top 16 to right	2.14	2.15	2.15	0.01	0.00
margin bottom 16 to right	2.23	2.25	2.13	-0.10	-0.12

Table A.15.: North-Elevation Revit Vertical Errors (Feet)

Type	Tape	Laser	Revit	Tape Error (Feet)	Laser Error (Feet)
15 left (garage door)	6.91	6.9	6.82	-0.09	-0.08
15 right (garage door)	6.87	6.87	6.84	-0.03	-0.03
margin roof/15 top left	1.27	1.27	1.24	-0.03	-0.03
margin roof/15 top right	1.22	1.21	1.18	-0.04	-0.03
16 left (garage door)	6.83	6.84	6.86	0.03	0.02
16 right (garage door)	6.86	6.86	6.86	0.00	0.00
margin roof/16 top left	1.22	1.21	1.16	-0.06	-0.05
margin roof/16 top right	1.22	1.22	1.26	0.04	0.04

Table A.16.: North-Elevation Revit Horizontal Errors (Feet)

Type	Tape	Laser	Revit	Tape Error (Feet)	Laser Error (Feet)
margin roof/left window18	1.13	1.13	1.03	-0.10	-0.10
margin roof/right window 18	1.14	1.14	1.08	-0.06	-0.06
roof/bottom left 18	3.97	3.97	3.96	-0.01	-0.01
roof/bottom right 18	3.97	3.97	3.99	0.02	0.02
left margin roof/window 19	1.15	1.16	1.18	0.03	0.02
right margin roof/window 19	1.18	1.18	1.08	-0.10	-0.10
height window 19	3.94	3.94	4.06	0.12	0.12

Table A.17.: South-Elevation Revit Vertical Errors (Feet)

Type	Tape	Laser	Revit	Tape Error (Feet)	Laser Error (Feet)
margin left to 18	2.83	2.83	2.91	0.08	0.08
bottom width window 18	4.94	4.94	4.95	0.01	0.01
top width window 18	4.94	4.95	5.00	0.06	0.05
margin top 18/middle	2.99	2.99	2.90	-0.09	-0.09
margin bottom 18/middle	3.01	3.02	2.91	-0.10	-0.11
margin middle/top 19	1.04	1.04	1.08	0.04	0.04
margin middle/bottom 19	1.04	1.04	1.10	0.06	0.06
top width window 19	4.94	4.94	4.93	-0.01	-0.01
bottom width window 19	4.94	4.94	4.97	0.03	0.03
margin top right 19/right	2.82	2.82	2.83	0.01	0.01
margin bottom right 19/right	2.82	2.82	2.79	-0.03	-0.03

Table A.18.: South-Elevation Revit Horizontal Errors (Feet)

Type	Tape	Laser	Revit	Tape Error (Feet)	Laser Error (Feet)
margin roof/above 1	8.85	8.85	8.93	0.08	0.08
margin bottom 9/above 1	4.28	4.26	3.95	-0.33	-0.31
9 Left (Window)	3.40	3.41	3.51	0.11	0.10
9 Right (Window)	3.41	3.42	3.47	0.06	0.05
margin right roof/bottom 9	4.60	4.61	4.52	-0.08	-0.09
margin left roof/top 9	1.17	1.18	1.01	-0.16	-0.17
margin right roof/top 9	1.17	1.20	1.00	-0.17	-0.20
margin roof/bottom 2 left		14.08	13.84		-0.24
1 Left (Door)	6.94	6.81	6.90	-0.04	0.09
1 Right (Door)	6.94	6.78	6.86	-0.08	0.08
margin roof/bottom 2 right		14.12	13.84		-0.28
margin roof/bottom 12	4.60	4.62	4.49	-0.11	-0.13
10 Left (Window)	3.40	3.42	3.51	0.11	0.09
10 Right (Window)	3.40	3.42	3.49	0.09	0.07
11 Left (Window)	3.41	3.43	3.52	0.11	0.09
11 Right (Window)	3.41	3.42	3.49	0.08	0.07
12 Left (Window)	3.42	3.43	3.49	0.06	0.05
12 Right (Window)	3.40	3.41	3.48	0.08	0.07
2 Left Window)	3.90	3.92	3.76	-0.14	-0.16
2 Right (Window)	3.91	3.91	3.83	-0.08	-0.08
3 Left (Window)	3.92	3.92	3.82	-0.10	-0.10
3 Right (Window)	3.92	3.92	3.83	-0.09	-0.09
6 Left (Window)	3.91	3.94	3.72	-0.19	-0.22
6 Right (Window)	3.91	3.92	3.84	-0.07	-0.08
7 Left (Window)	3.92	3.93	3.83	-0.09	-0.10
7 Right (Window)	3.93	3.93	3.82	-0.11	-0.11
4 Left (Door)	6.77	6.75	6.86	0.09	0.11
4 Right (Door)	6.80	6.78	6.89	0.09	0.11
5 Left (Door)	6.80	6.84	6.88	0.08	0.04
5 Right (Door)	6.73	6.73	6.85	0.12	0.12
8 Left (Door)	6.82	6.82	6.88	0.06	0.06
8 Right (Door)	6.77	6.77	6.92	0.15	0.15
margin bottom 12/top 7	5.58	5.56	5.41	-0.17	-0.15
margin roof/bottom 3		14.14	14.11		-0.03
margin bottom 10/top 3	5.60	5.60	5.52	-0.08	-0.08
margin roof/10 bottom left	4.60	4.62	4.52	-0.08	-0.10
margin roof/10 top left	1.20	1.20	1.09	-0.11	-0.11
margin roof/11 top left	1.20	1.22	1.19	-0.01	-0.03
margin roof/10 bottom right	4.55	4.57	4.53	-0.02	-0.04

Table A.19.: East-Elevation Revit Vertical Errors (Feet)

Type	Tape	Laser	Revit	Tape Error (Feet)	Laser Error (Feet)
margin left/9 top	4.07	4.07	4.14	0.07	0.07
margin left/9 bottom	4.08	4.08	4.18	0.10	0.10
9 Top (Window)	7.92	7.93	7.92	0.00	-0.01
9 Bottom (Window)	7.92	7.93	7.93	0.01	0.00
margin left/1 top	2.65	2.66	2.69	0.04	0.03
margin bottom/1 bottom	2.70	2.73	2.76	0.06	0.03
1 Top (Door)	3.12	3.13	3.15	0.03	0.02
1 Botom (Door)	3.12	3.13	3.16	0.04	0.03
margin 1/right top	0.27		0.22	-0.05	
margin 1/right bottom	0.20		0.20	0.00	
margin 2/left top	0.78		0.82	0.04	
margin 2/left bottom	0.78		0.83	0.05	
margin 9/10 top	7.75	7.77	7.84	0.09	0.07
margin 9/10 bottom	7.77	7.76	7.80	0.03	0.04
10 Top (Window)	7.92	7.90	7.92	0.00	0.02
10 Bottom (Window)	7.90	7.93	7.93	0.03	0.00
11 Top (Window)	7.92	7.90	7.93	0.01	0.03
11 Bottom (Window)	7.90	7.93	7.93	0.03	0.00
12 Top (Window)	7.92	7.90	7.84	-0.08	-0.06
12 Bottom (Window)	7.90	7.93	7.88	-0.02	-0.05
2 Top (Window)	5.88	5.91	5.90	-0.01	-0.03
2 Bottom (Window)	5.88	5.91	5.87	-0.04	-0.06
3 Top (Window)	5.98	5.91	5.90	-0.01	-0.03
3 Bottom (Window)	5.98	5.91	5.86	-0.05	-0.07
6 Top (Window)	5.88	5.92	5.86	-0.06	-0.06
6 Bottom (Window)	5.88	5.92	5.91	-0.01	-0.02
7 Top (Window)	5.92	5.92	5.85	-0.07	-0.08
7 Bottom (Window)	5.92	5.92	5.94	0.02	0.00
4 Top (Door)	3.11	3.14	3.18	0.04	0.04
4 Bottom (Door)	3.11	3.14	3.17	0.03	0.04
5 Top (Door)	3.15	3.15	3.17	0.02	0.04
5 Bottom (Door)	3.15	3.15	3.18	0.03	0.05
8 Top (Door)	3.13	3.14	3.17	0.03	0.03
8 Bottom (Door)	3.13	3.13	3.18	0.05	0.04
margin 2/3 top	4.57	4.62	4.69	0.07	0.07
margin 10/11 top	7.74	7.77	7.80	0.03	0.03
margin 10/11 bottom	7.74	7.75	7.71	-0.04	-0.06
margin 11/12 top	7.73	7.72	7.72	0.00	-0.01
margin 11/12 bottom	7.73	7.71	7.73	0.02	0.00
margin 12/right top	4.10	4.09	4.21	0.12	0.12

Table A.20.: East-Elevation Revit Horizontal Errors (Feet)

Type	Tape	Laser	Revit	Tape Error (Feet)	Laser Error (Feet)
margin: roof/lower 20	4.60	4.60	4.48	-0.12	-0.12
margin: roof/upper 20	1.20	1.22	0.95	-0.25	-0.27
20 (Window)	3.37	3.42	3.57	0.20	0.15
21 (Window)	3.42	3.43	3.57	0.15	0.14
margin: roof/upper 21	1.20	1.21	1.02	-0.18	-0.19
margin: roof/lower 21	4.61	4.62	4.51	-0.10	-0.11
margin: roof/lower 22	4.58	4.60	4.53	-0.05	-0.07
22 (Window)	3.41	3.43	3.49	0.08	0.06
margin: roof/upper 22	1.17	1.18	0.97	-0.20	-0.21
23 (Window)	3.42	3.43	3.51	0.09	0.08
margin roof/upper 23	1.17	1.17	1.01	-0.16	-0.16
margin roof/lower 23	4.58	4.59	4.51	-0.07	-0.08
24 left (Window)	4.91	4.92	4.90	-0.01	-0.02
24 right (Window)	4.88	4.90	4.92	0.04	0.02
25 left (Door)	6.80	6.80	6.66	-0.14	-0.14
25 right (Door)	6.80	6.80	6.69	-0.11	-0.11
26 left (Door)	6.80	6.82	6.66	-0.14	-0.16
26 right (Door)	6.80	6.82	6.67	-0.13	-0.15
27 left (Window)	4.92	4.91	4.92	0.00	0.00
27 right (Window)	4.92	4.91	4.89	-0.03	-0.02
28 left (Window)	4.92	4.92	4.81	-0.11	-0.11
28 right (Window)	4.90	4.92	4.82	-0.09	-0.11
29 left (Door)	6.82	6.81	6.68	-0.14	-0.13
29 right (Door)	6.82	6.81	6.65	-0.17	-0.16
30 left (Door)	6.78	6.78	6.67	-0.11	-0.11
30 right (Door)	6.80	6.80	6.66	-0.14	-0.14
31 left (Window)	4.92	4.92	4.86	-0.06	-0.06
31 right (Window)	4.92	4.93	4.92	0.00	-0.01

Table A.21.: West-Elevation Revit Vertical Errors (Feet)

Type	Tape	Laser	Revit	Tape Error (Feet)	Laser Error (Feet)
margin left/upper 20	4.06	4.07	4.21	0.15	0.14
margin left/lower 20	4.07	4.09	4.22	0.15	0.13
margin upper: 21/22	7.68	7.64	4.21	0.15	0.14
margin lower: 21/22	7.68	7.68	4.22	0.15	0.13
margin upper 23/right	4.07	4.07	7.80	0.12	0.16
margin lower 23/right	4.07	4.08	7.71	0.03	0.03
margin left/24	5.64	5.63	4.14	0.07	0.07
24 Top (Window)	3.90	3.91	4.18	0.11	0.10
24 Bottom (Window)	3.90	3.91	5.60	-0.04	-0.03
top margin: 24/25	1.72	1.72	3.92	0.02	0.01
bottom margin: 24/25	1.72	1.72	3.92	0.02	0.01
25 Top (Door)	3.13	3.14	1.67	-0.05	-0.05
25 Bottom (Door)	3.10	3.11	1.64	-0.08	-0.08
top margin 25/26	3.07	3.07	3.18	0.05	0.04
bottom margin 25/26	3.10	3.10	3.10	0.00	-0.01
26 Top (Door)	3.13	3.13	3.16	0.09	0.09
26 Bottom (Door)	3.14	3.16	3.18	0.08	0.08
margin top 26/27	1.70	1.71	3.16	0.03	0.03
margin bottom: 26/27	1.68	1.70	3.18	0.04	0.02
27 Top (Window)	3.93	3.93	1.67	-0.03	-0.04
27 Bottom (Window)	3.94	3.94	3.92	-0.02	-0.02
margin top 27/28	10.81	10.80	10.79	-0.02	-0.01
margin bottom 27/28	10.78	10.77	10.80	0.02	0.03
28 Top (Window)	3.92	3.92	3.86	-0.06	-0.06
28 Bottom (Window)	3.92	3.92	3.92	-0.01	-0.01
margin top 28/29	1.70	1.70	1.73	0.03	0.03
margin bottom 28/29	1.71	1.71	1.66	-0.05	-0.05
29 Top (Door)	3.14	3.14	3.17	0.03	0.03
29 Bottom (Door)	3.14	3.13	3.18	0.04	0.05
margin top 29/30	3.02	3.02	2.95	-0.07	-0.07
margin bottom 29/30	3.07	3.08	2.95	-0.12	-0.13
30 Top (Door)	3.12	3.12	3.18	0.06	0.06
30 Bottom (Door)	3.14	3.14	3.17	0.03	0.03
margin top 30/31	1.72	1.73	1.74	0.02	0.01
margin bottom 30/31	1.72	1.74	1.72	0.00	-0.02
31 Top (Window)	3.92	3.93	3.82	-0.10	-0.11
31 Bottom (Window)	3.92	3.92	3.88	-0.04	-0.04

Table A.22.: West-Elevation Revit Horizontal Errors (Feet)

Hand Measurements (Feet)	Revit (Feet)	Revit Error (Feet)
2.1	2.25	0.15
18.17	18.26	0.09
21.9	22.01	0.11
37.93	37.92	-0.01

Table A.23.: "Overall" Revit Errors for North Elevation (Feet)

Hand Measurements (Feet)	Revit (Feet)	Revit Error (Feet)
10.75	10.68	-0.07
10.73	10.69	-0.04
8.77	8.87	0.10
8.76	8.91	0.15

Table A.24.: Overall Revit Errors for South Elevation (Feet)

Hand Measurements (Feet)	Revit (Feet)	Revit Error (Feet)
4.08	4.26	0.18
12.01	12.11	0.09
19.84	19.95	0.11
27.75	27.77	0.02
35.53	35.72	0.19
43.43	43.64	0.21
51.15	51.21	0.06
59.05	59.18	0.13
63.15	62.94	-0.21

Table A.25.: Overall Revit Errors for East Elevation (Feet)

Hand Measurements (Feet)	Revit (Feet)	Revit Error (Feet)
9.55	9.52	-0.03
11.26	11.26	0.00
14.38	14.35	-0.03
17.47	17.51	0.04
20.61	20.66	0.05
22.30	22.33	0.03
26.21	26.18	-0.03
37.01	37.04	0.03
40.92	40.83	-0.09
42.60	42.60	0.00
45.73	45.68	-0.05
48.79	48.80	0.01
52.01	51.83	-0.19
53.62	53.68	0.06
57.66	57.57	-0.09

Table A.26.: "Overall" Revit Errors for West Elevation (Feet)