Ortho-Photogrammetry for Prefabricated Energy-Efficiency Retrofits

by

Logan Gilmour

A thesis submitted in partial fulfillment of the requirements for the degree of

Master of Science

Department of Computing Science

University of Alberta

Abstract

Panel-based prefabricated retrofits of existing building envelopes are considered a promising strategy for improving energy performance and reducing greenhouse gas emissions. These projects typically require some form of 3D model of the as-is building geometry, in order to construct the panels offsite, and, currently they tend to use laser scanning for this purpose, because of their accuracy requirements. However, laser scanners must be operated from a tripod, and producing drawings from scans is a largely manual process involving unwieldy amounts of data. We propose a line-photogrammetric approach that requires a small number of photographs (typically 8), that we call 'Ortho-Photogrammetry'. Ortho-Photogrammetry is parametrised by axis-aligned planes and produces rectified geometry suitable for panel construction. It does not require a calibrated camera or targets. We demonstrate that it is sufficiently accurate to support prefabricated retrofits, even when used with photos from a consumer unmanned aerial vehicle (UAV), with on average $\pm 3.3mm$ 1σ error in our experiments.

Preface

This thesis was published in an abridged form as: L. Gilmour and E. Stroulia, “Ortho-photogrammetry for prefabricated energy-efficiency retrofits,” *Automation in Construction*, vol. 134, p. 104082, 2022

E. Stroulia was the supervisory author and was involved with concept formation and manuscript composition.

For Grandpa

All models are wrong but some are useful.

– George E.P. Box, 1979.

Acknowledgements

I'd like to thank Peter for giving me this problem to solve, and Eleni for setting me up to solve it. I'd like to thank Sticks & Stones for letting me use their drone, and Mark Carver from Canmet Energy for the data for Building 4. Finally, I'd like to thank Hero for helping keep me sane while doing this work during a pandemic.

This work was funded by the DITA NSERC CREATE program and the Mitacs Accelerate program.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Prefabricated retrofits of existing building envelopes are considered a promising strategy for improving energy performance and reducing greenhouse gas emissions [21], [29], [39]. As the materials are prefabricated offsite, these projects require some form of 3D model of the as-is building geometry. In general, existing building geometry is captured with one of four methods: tape measurement, tacheometry, terrestrial laser scanning, and photogrammetry [17], [51].

Tape measurements and tacheometric surveys only capture explicitly selected measurements and require significant manual effort for complex projects [17]. Laser scanning captures a series of dense point-clouds. These point-clouds are registered together via targets placed in the scene or feature-matching and then traced by an architectural technologist to produce a 3D model [53]. If additional geometry is deemed necessary, or existing geometry is suspect, it should be possible to refine the 3D model by revisiting the point-cloud, rather than the site. Though automatic semantic 3D model production (scan-to-BIM) is an active research area, this is currently a largely manual process involving unwieldy amounts of data [17], [47], [51].

A photogrammetric site survey comprises many photographs with overlapping contents such that any point on the building is present in at least two photographs. This is sufficient for 3D reconstruction [16] and also provides photographic documentation. Modern structure-from-motion photogrammetric processing ultimately also results in a point-cloud that requires significant manual labour to extract useful geometry [17]. Because a photogrammetric point-cloud is based on triangulation of individual points using multiple views, large areas with little detail (such as featureless walls) offer poor depth resolution, often resulting in a lower quality point-cloud than laser scanning. Structure from motion is typically considered insufficient for accurate as-is documentation [4], [16], [29]. However, this process can be accomplished with an inexpensive consumer camera, or even the camera built in to a photographic unmanned aerial vehicle (UAV) [10], [41].

Very high accuracy photogrammetry is possible with more manual labour. Rather than relying on computer vision techniques to automatically identify and match points across photographs, traditional close-range photogrammetry uses sub-pixel recognizable coded targets placed on the to-be-reconstructed object. It is possible to accurately reconstruct the locations of those targets with as low as 1:500,000 1σ root-mean-square (RMS) error to largest side-length [33], corresponding to 0.1mm 1σ error for a 10 meter building. This technique assumes, however, very careful camera calibration and operation, as well as significant labour for placing the coded targets.

As traditional close-range photogrammetry is very labour intensive and feature-based structure-from-motion photogrammetry is considered insufficiently accurate, current panel-based prefabricated retrofit projects tend to use laser scanning [5], [29], [40]. Ultimately, accuracy tolerances for geometry extraction depend on the system used for panel prefabrication. Borodinecs *et al*. recommend accuracy better than $\pm 5mm$ 1σ error for all geometry. The Competence Center Energy and Mobility (CCEM) Advanced Energy-Efficient Renovation of Buildings project (or CCEM Retrofit Project), specifies better than $\pm 4mm$ 1σ error around windows and $\pm 7mm$ 1σ error for the roof and various facade elements [22]. The system of our industrial partner is most sensitive to the locations of windows and doors, and is also compatible with a $\pm 4mm$ 1σ accuracy for those elements.

A photographic survey is typically quite expedient [51], as photographs can be taken quickly without a significant setup process. Given that photographs may also be taken by a UAV, an option currently unavailable to survey-grade laser scanners, a viable photogrammetric process could be useful, particularly when significant occlusions or tight spaces are involved.

The contribution of our work is a method for photogrammetric reconstruction of mostly-orthogonal planar structures, which we call ‘Ortho-Photogrammetry’. Though our work was motivated specifically by the high accuracy requirements of the panel-based retrofit application, the underlying algorithm is novel and may have applications outside of panel-based retrofits. The key novelty lies in the parametrisation of the building reconstruction problem by orthogonally-constrained planes, which then necessitates a series of modifications to the normal reconstruction process. This method has four main advantages over other existing methods. First, it requires a small number of photographs (usually 8) for a successful reconstruction. Second, it fits image lines to point markings, rather than fitting image points to markings. Lines are abundant in images of buildings, which make the method robust to occlusions. Third, geometry reconstructed with ortho-photogrammetry is hard-constrained to be orthogonal, and so is simple to use as a basis for panel design for prefabricated retrofits, as any features assuming orthogonal geometry in the panel designer’s chosen CAD (Computer Aided Design) software can be used. Finally, the strong geometric assumptions of the method alleviate the need for an extra mechanism for initialization of the optimization process, and allow the use of an uncalibrated camera (preferably of reasonable quality, equivalent roughly to a modern prosumer Digital Single-Lens Reflex Camera (DSLR)). This reduces the complexity of the software as well as the reconstruction process itself.

We have validated the effectiveness of our method with two real-world projects, in total involving four buildings undergoing panel-based prefabricated retrofits (three reconstructed from images taken by a consumer UAV); our data demonstrate that it is possible to reconstruct the building geometry with the necessary accuracy.

The thesis is laid out as follows. In Chapter 2, we review the accuracy achieved using existing photogrammetry systems applicable to the reconstruction of building geometry, and build up the motivations for our parametrisation and method, placing it in the literature. In Chapter 3, we explain the process of reconstruction using our method. In Chapter 4, we formalize the proposed parametrisation and optimization. In Chapter 5, we detail our two experiments we conducted to analyze the accuracy of the reconstructions resulting from our method, and discuss the results and the method's viability for panel-based prefabricated retrofits.

Chapter 2

Background Material

Photogrammetry is the reconstruction of 3D geometry from photographs, usually in the form of a set of 3D points [33]. Typically, the position and orientation of the camera for each photograph is also reconstructed. The internal parameters of the camera may be calibrated separately or may be reconstructed along with everything else, though typically “on-the-job” calibration necessitates more redundancy in photographs. This redundancy is easily achieved if coded targets allowing automatic measurements are used [18].

However, for measurement of buildings, placing coded targets to mark desired geometry is a time-consuming process requiring ladders and a variety of options for temporarily affixing targets to different facade materials. We begin by reviewing the accuracy of existing uses of photogrammetry in building reconstruction.

2.1 Accuracy

We will use “relative error” as the metric to evaluate accuracy [32, Section 7.2.1.7]. Relative error is the ratio of the standard deviation of the error to the largest dimension of the object being captured, typically expressed in the form 1:k. If we assume a conservatively small building with a largest dimension of approximately 10m, this means that to reconstruct the openings with $4mm$ 1σ error we are targeting an accuracy of 1:2500 or higher.

Sapirstein *et al.* [43] survey a variety of uses of structure-from-motion photogrammetry, primarily from automatically matched feature-descriptors but also including some with manually marked points, such as [12]. They find that when measuring the error of the reconstructed model vs. real-world control points shot with a total-station, the best methods have a relative accuracy of 1:1000. For our particular problem, this would correspond to approximately 10mm 1σ error, which is insufficiently accurate. They also review reports of higher accuracy on smaller objects when benchmarked against a laser scanner, with the best of these having an accuracy 1:5000. However, that particular subject was rough stone artefact with a texture ideal for structure-from-motion [31], photographed indoors in controlled conditions. None of the large-scale subjects (greater than 10m largest dimension) had accuracy better than 1:1000, whether compared with total-station control points or a laser-scanner point-cloud.

Ordóñez *et al.* [38] report an accuracy of approximately 1:760 with a single-image system for planar facade measurements that uses a laser-rangefinder to establish scale.

Murtiyoso *et al.* [36] describe the reconstruction of the facade of a 20m stone heritage building to approximately 1:2000 accuracy, with structure-from-motion photogrammetry using photos from a UAV. However, this is a task simpler than the problem we aim to address with our method in two ways. First, it involves the reconstruction of a single face of a building with no occlusions, and second, the textured stone building facade is highly amenable to automatic feature-matching.

Faltýnová *et al.* [16] report that using structure from motion photogrammetry, they could reconstruct a point-cloud with accuracy approaching the accuracy of a laser scanner using a high-end medium-format Pentax 645D with a 40 megapixel sensor, 56 images, and 20 hours of processing. They reconstructed a single elevation and use 6 control points shot with a total station. They do not provide the statistics, but from a heat-map of errors it appears that most points are within 5mm of the laser scan, though there are a non-trivial number of errors greater than 10mm.

Daftry *et al.* [10] reconstruct a single, unobstructed face of a the facade of a moderate-sized multi-family using photos from a UAV. They do not specify the size of the building, and only report the average absolute error of 9mm from control-points established with a total station achieved with their method.

In summary, 1:2500 accuracy reconstruction of buildings with low-detail planar surfaces and challenging occlusions has not been demonstrated with state-of-the-art photogrammetric methods.

2.2 Bundle Adjustment

Most photogrammetric approaches to 3D reconstruction use Bundle Adjustment to compute the final reconstructed geometry [27]. Bundle Adjustment is the simultaneous optimization of the estimated geometry of the object to be reconstructed, locations and orientations of the camera for each photo, and the internal parameters of the camera (such as focal length, photo sensor size, etc.), such that reprojection error is minimized [49]. This is a very general formulation: as long as the camera and object geometry are modeled such that it is possible to compute the 2D projection of the estimated geometry from the perspective of the camera, it is possible to project the estimated geometry back through the estimated camera, and superimpose it on the 2D geometry in the photos, then minimize the error between the estimated geometry and the measurements. It is important to note that this optimization is in general not convex, and so typically a separate procedure is required to initialize the parametric model to an estimate that is reasonably close to the optimal solution [23, Section 18.1].

2.3 Parametrisation

The choice of parametrisation can make a very significant difference in the performance of the optimization. A minimal parametrisation has the potential to work significantly better with off-the-shelf nonlinear optimization software [49]. Furthermore, when posed as minimizing the sum of squared errors of reprojection errors, bundle-adjustment is essentially a maximum-likelihood estimation of the underlying model (assuming that the error is normally-distributed). [49]. This suggests that the bias-variance trade-off should hold [52], which implies that by adding constraints on the model, it is possible to fit a given model with less data (i.e. fewer photographs and marked features), at the potential expense of introducing bias in the model.

We, therefore, tailor our parametrisation to the specific types of buildings that can be retrofitted using a panel-based approach. These buildings are highly planar and reasonably rectangular, so that the rectangular panels can practically be affixed to the facade. A minimal parametrisation of building models restricted to planar, rectified geometry, should enable bundle adjustment with fewer photographs, at the cost of only being able to model this idealized geometry typical of our target buildings. The panels being installed are rectified and planar, so finding the closest rectified and planar model of the building that fits the photographic data is likely a useful approximation of the true geometry for panel-design purposes.

2.4 Manhattan World Assumption

Perhaps the most simplifying assumption we can make for building geometry is that all planes are aligned to either the X, Y, or Z axis of the world coordinate system – this is the Manhattan World Assumption [8], so named for the tendency for city buildings and street grids to have such an alignment. This assumption has seen significant use in building reconstruction [17], in part because it provides a constrained setting for reasoning about the content of large, texture-less planes that automatic structure-from-motion photogrammetry cannot handle well [20]. It is also a useful assumption for constructing geometry from laser-scan point-clouds [35].

The Manhattan world assumption also significantly simplifies estimation of the camera orientation [7], [8], as well as the camera calibration [15], [55], such that both can be computed from a single image. This is not true of images with unconstrained geometry, which require at least two images to recover the camera orientation [23, Chapter 10] and camera calibration, even when assuming the camera calibration is known excepting its focal length [23, Section 19.1].

This assumption is too constrained to model many buildings – the simple introduction of steeped roofs defeats it. However, we may note that these buildings are still ‘nearly-Manhattan’, in that most planes are world-frame aligned, and the few that are not can be described by simply adding very few additional frames at different orientations that allow the description of the out-of-alignment geometry. A first version of this was introduced as the Atlanta World assumption [44], which added support for a number of frames with arbitrary rotation about the vertical axis, allowing urban environments with street plans beyond simple grids. A further generalization was introduced as a 'mixture of Manhattan-frames' [46], which allow frames with arbitrary 3D rotations. The basic insight is that man-made objects like buildings tend to organize geometry in rectified forms.

Such an assumption can significantly reduce the number of parameters required to model our geometry. The minimal parametrisation for a plane requires three parameters [23, Section 3.2.1] (a vector with direction normal to the plane and length encoding the closest from the plane to the origin). This can be reduced to a single parameter (distance to the origin) in the case that the plane direction is fixed to one of the three possible vectors permitted by the Manhattan frame prior to adjustment.

2.5 Line Photogrammetry

A disadvantage to the use of planes as photogrammetric primitives is that an individual infinite 3D plane does not project to an identifiable feature in 2D. This can be overcome through use of the plane’s textural content [19], [54] or by first reconstructing 3D lines [25], [45] or points [37], [42] and then fitting planes to them. However, two planes that are not parallel must necessarily intersect in a unique 3D line [23, Section 3.2.1]. As a 3D line simply projects to a 2D line [23, Section 8.1.2], plane-plane intersections provide a feature that may be projected to 2D.

It can be advantageous to use lines instead of points for marking photographs, as lines are heavily present in the urban environment and are robust to occlusion [17]. This latter point stems from the fact that a line has evidence spread across an image along its length, while a point corresponds to a single pixel of the image, and hence is more likely to be covered up by vegetation, clutter, or other parts of the building to be reconstructed. This should reduce the number of photographs required to capture all geometry with sufficient redundancy. Though our method is parametrised by planes, it is simple to calculate 3D lines from our underlying planes [23, Section 3.2.1].

Van den Huevel [50] formulates a line photogrammetry approach partially using planes in the underlying parametrisation, though they use planes in conjunction with points in order to express soft constraints effectively. In our method, geometric constraints are purely "hard" constraints; rather than adding additional residuals that would require weight tuning and could potentially slow down the optimization process, the constraints reduce the number of parameters required to describe a building, likely making the optimization faster [3].

2.6 Image-Based Modelling

Identifying a Manhattan frame within a single image is equivalent to discovering the orthogonal vanishing points corresponding to that frame [23, Section 8.6], which is sufficient for taking certain measurements from that single photograph even without knowledge of the camera calibration [9]. Additionally, it becomes possible to model details in this frame directly on top of a single image [24]. This is very promising; the ability to model plausible geometry directly on a single photograph alleviates the need for a mechanism to find an initialization for the bundle adjustment, as it becomes possible to simply begin the optimization process from the modelled geometry.

Our method is similar to Façade [14], in that it also models over photographs with highly constrained geometry in combination with line-based photogrammetry. However, where Façade provides a heterogeneous set of parametric primitives as the underlying model, our method parametrises all geometry with only planes set within nested restricted Manhattan frames. In spite of its simplicity, our method supports the modelling of arbitrary piece-wise planar geometry. Additionally, because the geometry is so well constrained, there is no need for camera calibration beyond the EXIF data embedded in a standard JPEG.

Chapter 3

The Modelling Process

Our method is implemented in a software tool that integrates functionalities of a CAD program and a photogrammetry software and guides the user to model the geometry of a building, based on a collection of images. A high-level overview of the process is as follows: The user begins by choosing a 'good' photograph from the a photosurvey, and models a small amount of the geometry visible in the photograph (typically a single rectangle), directly on top of the photograph. Next, they manually mark in the photo points that lie on specific edges of the model, triggering the software to fit the overlayed geometry to the photo. Continued modelling should now be easier, as new orthogonal geometry will naturally align with the photo. This process continues in a loop, with the user modeling geometry and then improving the overall fit of the model to the photos with markings. When all desired geometry is modelled in sufficient photos, the user can establish the scale of the geometry by importing external measurements, then export the final geometry for use in panel-design.

3.1 Photo Survey

Our process begins with the acquisition of the photographs to be used in reconstruction. Typically, a small subset of the photos taken will actually suffice for the reconstruction, making it unlikely that additional visits to the site will be required. Onsite, it is worth considering roughly what subset of photographs are desirable so that the photo survey will likely include them. This

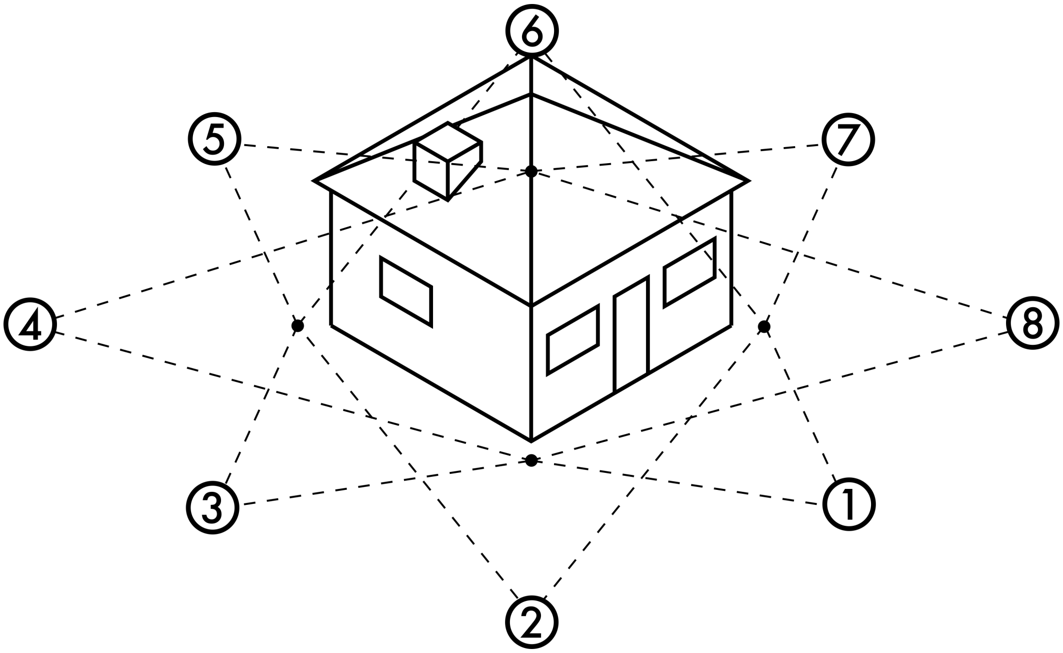

Figure 3.1: A generic network often used for roughly cube-shaped objects. Each number represents a camera position, and the dotted lines are the horizontal field of view.

is especially true when flying a UAV, as many camera locations are possible.

A prosumer DSLR or equivalent is recommended, as higher resolution will help resolve individual edges more accurately, and the quality of the optics and the stability of the interior orientation will improve the reconstruction [18]. For drone-based photography especially, a larger sensor is recommended, as the aperture needs to be small enough to capture the whole building in focus without changing the focus during shooting. The drone may move slightly (especially in the presence of wind), so faster exposures are likely desirable.

The accuracy of a photogrammetric reconstruction depends significantly on the 'strength' of the network of photographs used [2]. Intuitively, a strong network captures images of many geometric points at orthogonal camera angles, so that the rays implied by the various views of an individual point constrain its location, as well as possible. Ideally, each image should contain as much geometry as possible with little to no extraneous content, so that the visible geometry just fills the frame of the camera.

Network design is a complicated topic, but fortunately most small buildings are (to a first approximation) cubes, so that we can use a suitable 'generic network' [34] shown in figure 3.1. The essential configuration is 8 photographs, each point roughly at the center of the building, each at a 45 degree rotation relative to the last. As each photograph can be taken from approximately the same height at ground-level, this is a practical network to capture with a human camera operator.

However, single-family-homes may have very significant clutter obscuring some views of the building in the form of fences, adjacent buildings, and vegetation, and so in general the network will end up being the closest approximation possible given the possible vantages from which the majority of important geometry on the building is visible. UAV-based photographs provide a particularly helpful departure from the basic generic network, as it is possible to capture multiple orbits at different elevations, which can provide vantages that avoid clutter obstructing important geometry.

In addition to photographs, some measurements or control-points are required to establish the building scale. We typically use two total-station setups on two different elevations, and shoot three distinct points with each, which serve to adequately constrain the geometry as well as scaling it. It is also possible to use tape-measurements to establish the scale.

3.2 Overview of the Modelling System

Having conducted a photo survey, the acquired photos are then loaded into the software we have developed. The 3D user interface of our modelling software was developed using Unity3D[1], and the bundle-adjustment functionality of the tool uses the non-linear optimization software Google Ceres [1]. The modelling workflow relies on incremental bundle adjustment to add photographs and geometry to the solution in a step-wise fashion.

To start, the user chooses a 'good' photograph (informed by the photo survey guidelines in Section 3.1), and begins roughly modelling a 3D reconstruction directly on on top of the photo, by specifying axis-aligned planes that

[1] https://unity.com/

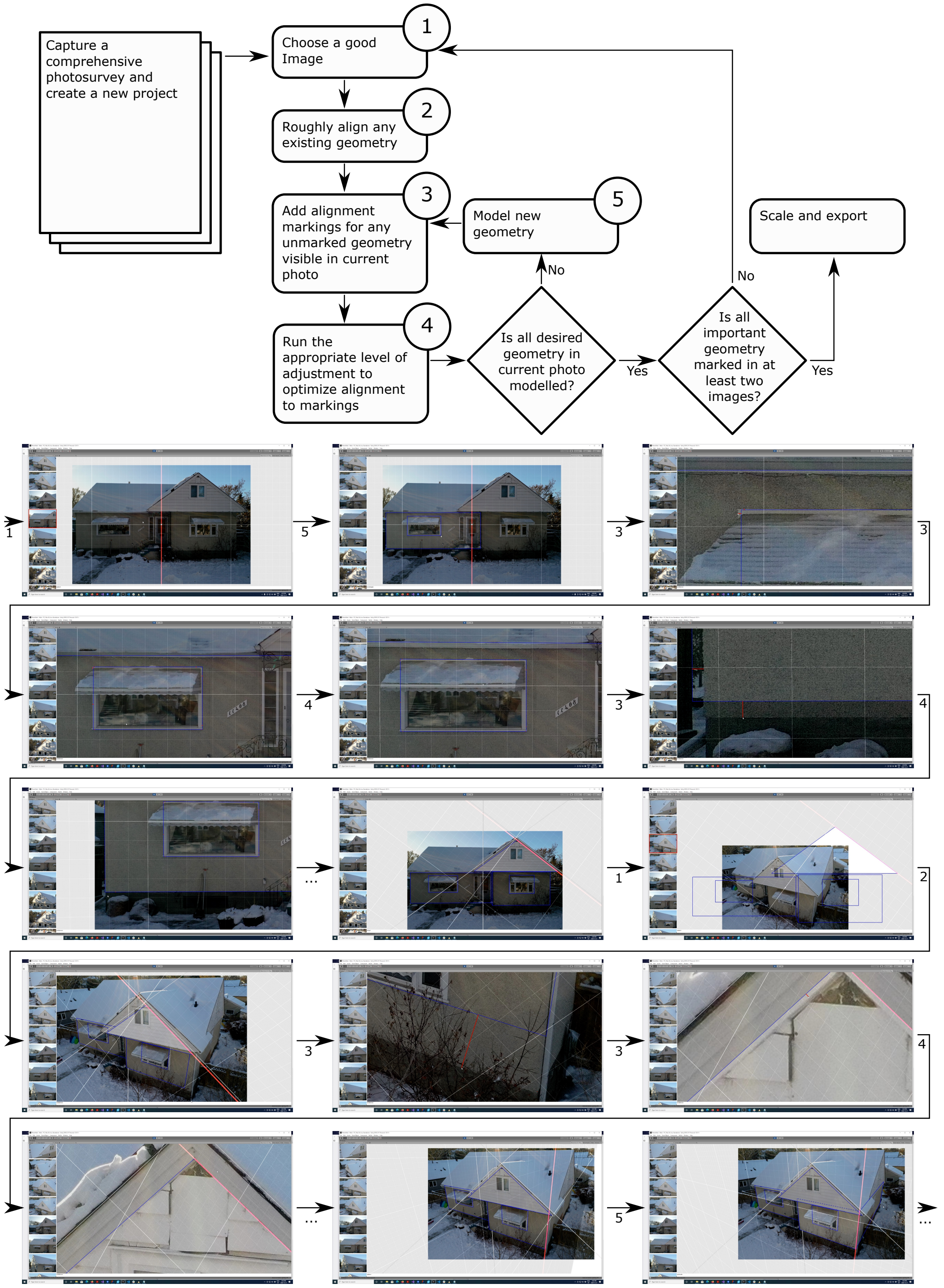
Capture a comprehensive photosurvey and create a new project
Choose a good Image
1
Roughly align any existing geometry
2
Add alignment markings for any unmarked geometry visible in current photo
3
Run the appropriate level of adjustment to optimize alignment to markings
4
Model new geometry
5
Is all desired geometry in current photo modelled?
No
Yes
Is all important geometry marked in at least two images?
No
Yes
Scale and export
1
5
3
3
4
3
4
...
1
2
3
3
4
...
5
...

Figure 3.2: The modelling process workflow and an example illustrated with screenshots.

intersect at edges in the photo. A plane is added by drawing an intersection line on top of a selected existing plane, beginning with a default plane that exists when modelling begins. Simply put, the user draws in the lines where two walls, a window frame and a wall, a wall and the soffit, etc., meet. In general, the first thing modelled should be a rectangle, which is sufficient to align the first camera, as it composed of two coplanar orthogonal pairs of parallel lines, which are sufficient for metric rectification of the plane of the rectangle [23, Section 8.8]. To bind these traced intersection lines to the photograph, the user then adds markings in the photo that specify positions on the image that model edges should ideally pass through. To optimize the model geometry to best fit these image markings, the user invokes the bundle-adjustment functionality of the tool.

The user may then continue modelling by drawing new geometry, extruding existing geometry, intersecting existing planes into new lines, or rotating existing planes about a single axis. All of these operations are automatically generate constraints (as they are implicit in the underlying parameterization), so no additional process is required to constrain the geometry. When the user has modelled all the geometry that it is practical to model in that first photo, they can proceed to roughly align a new photograph (chosen to most closely follow the guidelines in Section 3.1 while also providing as much visible geometry both new and overlapping as possible) to the existing model by moving and rotating the model. Then, the user should identify geometry elements modelled in previously examined photographs, and add alignment markings relating the existing model to the corresponding lines in that new photograph. After invoking the bundle-adjustment functionality to find a good alignment of the photo to the geometry, the user may then model any new geometry not visible in previously processed photographs. This process repeats until a set of photos that best approximate the generic network from Section 3.1 are included in the reconstruction. See figure 3.2 for the high-level workflow.

Once the model is complete, its scale must be established. The tool allows the user to add measurements collected from the site between parallel faces. It also allows the user to import a set of coordinates shot by a total-station and, after roughly aligning the stations rotation and position relative to the model, assign each coordinate to lie in one or more planes.

3.2.1 Bundle Adjustment

The tool supports multiple levels of adjustment, which helps ensure stable results throughout the process. By marking various parameters to be held fixed by the optimizer, the modelling tool enables the user to adjust

1. Only the camera positions and rotations for each image;
2. The camera positions and rotations and the model geometry;
3. The camera positions and rotations, the model geometry, and just the focal length f and first radial distortion term k_1; or
4. The camera position and rotation, the model geometry, and all of the intrinsic camera parameters.

We assume that all photographs were captured with the same camera, in the same configuration. As discussed above, our tool enables the user to align the model to new photographs, by continuing to add new geometry elements and further constraining existing geometry with additional markings. The user should employ only adjustment levels 1 and 2 until substantial geometry across at least two photos have been modeled. Level 1 is most useful for adding a new photograph to the adjustment, while level 2 is most commonly used, as it fits new and existing geometry without adjusting the camera's internal parameters In theory the camera calibration may be refined using a single image that identifies three orthogonal vanishing points (for example by modelling two adjacent rectangles of a cube) [23, Section 8.8]. Periodic use of adjustment 3 becomes useful once multiple photographs are included, as the focal length significantly affects the fit of the axis-aligned planes to the photographs, and the first term of the lens distortion will allow straight lines to bend to fit the images. In our experience, adjustment level 4 is only necessary on the final step of the method before exporting the model, as the calibration provided by level 3 should be sufficient to keep the adjustments stable and make drawing geometry easy.

Because of the small number of parameters used in Ortho-Photogrammetry, even a complete model with a full set of photographs required only 1-2 seconds for each adjustment in the experiments we ran, which used an Intel i7-7700K processor (4 cores at 4.2 GHz). If an adjustment returns poor results (typically because the geometry is under-constrained so that the model deviates very obviously from the images), the user can just undo the adjustment, add a few more markings to better constrain the geometry, and then adjust again. To help provide intuition for what additional markings are needed, the tool provides a real-time view of the optimization, showing each step as it occurs.

3.3 Planar Mesh Modelling

The tool ultimately relies on infinite-length 3D lines produced by plane-plane intersections in order to represent and optimize the residuals. This is fine for the optimization, as the lines are simply aligned to a finite number of markings with a natural extent stemming from the fact that the image line-segment being marked has finite length. However, a 3D model comprised exclusively of infinite 3D lines is very difficult to interpret and understand, both for the modeller as well as the subsequent users of the model. Therefore, the tool models geometry as a set of 3D polygonal faces. Each face is defined by a single base plane B in which the face lies, and an ordered list of planes $E_0, E_1, \ldots, E_n$ that intersect B to define each edge of the face. A plane may be referenced by multiple faces, just as a vertex may be referenced by multiple faces in a conventional indexed-face-set mesh [6].

We have developed a number of operations to enable the user to model a planar mesh. In terms of the overall workflow (figure 3.2), these operations belong in box five: "Model new geometry". These operations make use of the underlying parametrisation that will be defined in chapter 4, so that the planes are all axis aligned, and new nested single-axis rotation frames are created automatically as needed. Roughly speaking, the current geometry is rendered as either unbound edges, specified by two planes that uniquely intersect in that line, and rendered as infinite-length white lines, or as face edges, rendered as coloured line-segments. Additionally, faces are triangulated and written to a depth buffer so that they may occlude each-other.

For clarity in the figures in this section, we have used images of very simple objects rendered by a 3D modelling tool, rather than photographs.

3.3.1 Edge/Plane Selection

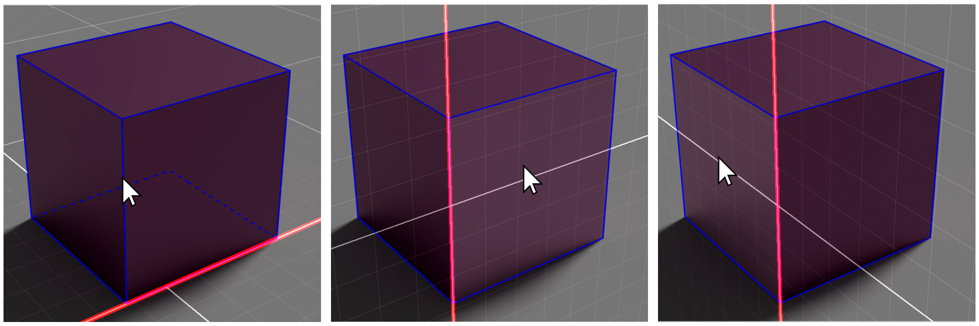

Figure 3.3: By clicking on the vertical edge and dragging left or right, the selected plane for the vertical edge is chosen.

The trouble associated with modelling with 3D planes is in some ways worse than working with 3D lines, as unbounded 3D planes can only be visualized by surface detail such as a synthetic grid. Drawing multiple grids simultaneously becomes quickly unintelligible, so we only visualize one plane at a time; specifically, the ‘selected plane’, which is chosen in tandem with a ‘selected edge’. Most of the other operations depend on the current selection state.

When using the ‘Select’ tool, an edge/plane selection is defined by first clicking on the edge to select, and then dragging away from the edge in either direction roughly perpendicular to it. When the mouse is released, the mouse position will be used to intersect a corresponding camera ray with both planes making up the edge, and the plane that is farther from the camera in that direction will be the selected plane. In this way, the selected edge is always ensured to contain the selected plane. The selected edge is indicated by a red highlight, and the selected plane is indicated by an overlayed grid. See figure 3.3. Note that because any edge is always defined by exactly two planes, this means that the current selected edge and plane always imply that other intersecting plane, which we will refer to as the 'selection-intersecting plane'. The selection-intersecting plane is used by a number of tools.

3.3.2 Draw

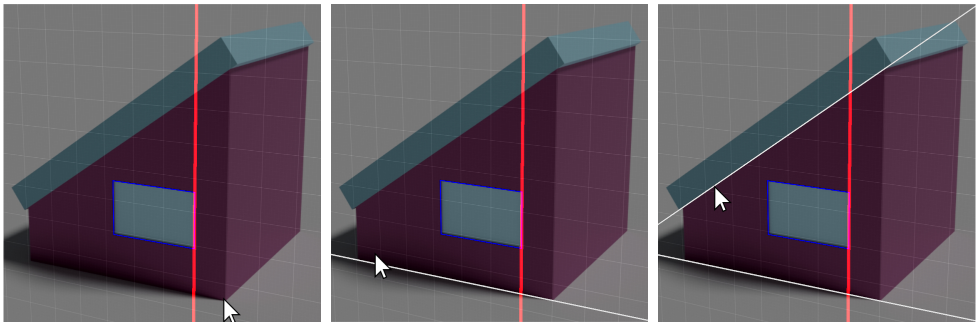

Figure 3.4: Clicking and dragging draws a grid-aligned line. In the third image, holding the control key allows the line to lie at any angle within the selected plane.

The user may press, drag, and release the mouse left button to create an edge in the selected plane. The mouse positions at press and release are used to calculate intersection coordinates of the camera ray corresponding to the mouse click and the selected plane. The 'press' coordinate identifies a point the edge must pass through, and the 'release' coordinate is used to choose which grid-aligned direction the line should take, by choosing the direction that minimizes the distance from the coordinate to the edge.

If the user holds the 'control' key while drawing, the edge is not constrained to be grid-aligned, and rather will pass through both the 'press' coordinate and the 'release' coordinate. This is accomplished by creating a new child frame of the selected plane's frame, that rotates about the selected plane's normal direction. See figure 3.4 for an example of drawing both grid-aligned and unconstrained edges.

Drawing an edge results in the creation of a new plane that defines that edge together with the selected plane. If the edge is axis-aligned, it is defined in the same frame as the selection-intersecting plane, assuming the selection-

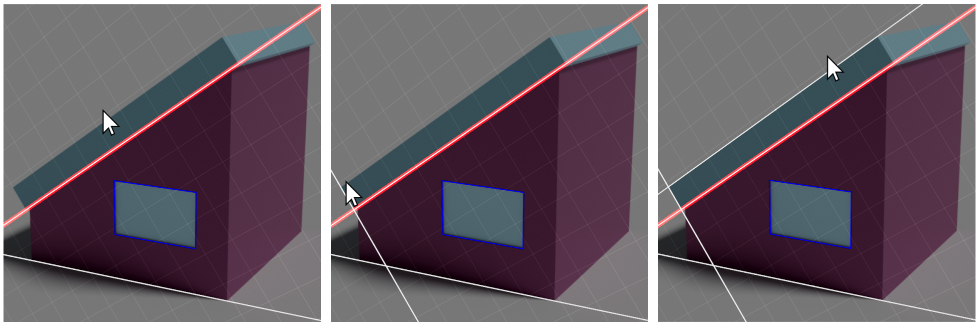

Figure 3.5: By selecting the rotated edge, we may draw grid-aligned edges in the frame defined by that rotated edge.

intersecting plane's frame is considered valid. The selection-intersecting plane's frame is valid if it is the same frame as the selected plane, or if it is in a direct child frame of the selected plane's frame, so that it rotates about the selected plane's normal. This validity check ensures that it is always trivial to calculate which frame axis are orthogonal to the selected plane's normal axis, so that horizontal and vertical edges are assigned the correct normal axis.

If the edge is not constrained to be grid aligned, then it is created in the newly defined frame that was created to allow its rotation. Note that because this new frame is considered 'valid' (it is a direct child of the selection-intersecting plane's frame that rotates about the selected plane's normal axis), it is then possible to re-select the current selected plane with the newly created edge as the selected edge. It is then possible to draw new grid-aligned lines, where the grid alignment is defined by the new frame, so that it is possible to draw edges that are constrained to be parallel or perpendicular to the rotated edge. (See figure 3.5)

3.3.3 Make Face

Note that the draw tool creates infinite length while lines that are not bound to any face. To bind these edges to a new face, we may first use the Select tool to select the plane in which the desired face will lie, and then use the Face tool to specify the edges that should make up the face. Only edges that lie in the selected plane may be chosen, and the edges should be specified in order proceeding around the edge of the face. The system does not depend

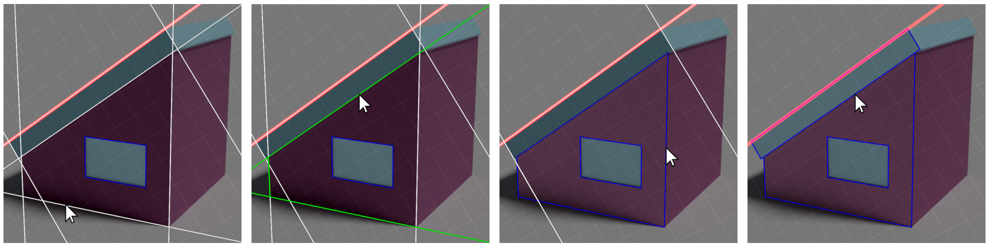

Figure 3.6: The user may click edges lying in the selected plane in sequence to create a polygonal face. The fourth image shows the result of creating a second face in the same plane that shares an edge with the first face.

on winding order, so either clockwise or counter-clockwise order is allowed. In figure 3.6, we see that the selected lines are visualized in green while the face is being constructed.

Once the face is fully specified, a new face is created. The selected plane is the base plane, and the intersecting planes are specified by the picked edges. This causes the edges are marked as 'bound', and are no longer rendered as infinite white lines. Rather, they are rendered as the line segments making up the face. However, note that in rightmost photo in figure 3.6, we show the result of creating a second face that reuses one of the edges from the first face; an edge may be used in multiple faces.

3.3.4 Intersect

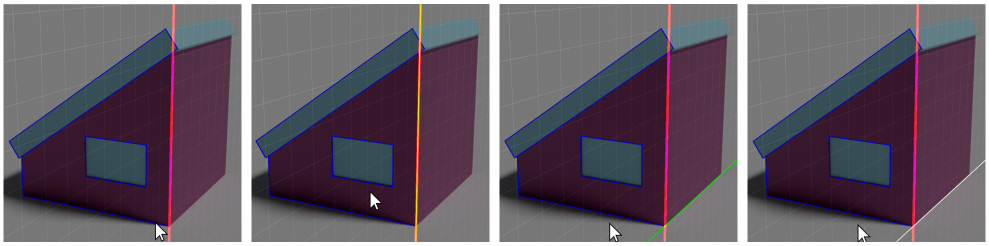

Figure 3.7: An intersection is created with the selected plane by clicking an edge and then dragging up or down, similar to how selection works. The fourth image shows the resulting intersection line on release of the mouse button.

The Intersect tool makes it possible to share planes across multiple different surfaces of a planar object. Note in the leftmost image in figure 3.7 that it is possible to select (and hence draw in) one of the intersecting planes of the original plane. However, if we were to simply ‘draw in’ the bottom edge of the object, the draw tool would create a new underlying plane specifying the intersection line. Specifically, it would be a different plane than the one specifying the bottom of the object in the original selected plane, which would be incorrect, as the bottom of the object is clearly a single plane.

To instead create a new edge in the new selected plane that is the intersection of the selected plane and an already existing plane, we may first use the Select tool to select the plane the edge will be created in, and then use the Intersect tool to chose the intersecting plane. The Intersect tool functions in the same way as the Select tool; to pick the intersecting plane, we may click on an existing edge that is composed from the desired plane, and then drag the mouse to the side of the edge that specifies that plane. While dragging, the resulting new edge is visualized with a green line, as seen in the middle two images in figure 3.7 (though note that one of the visualized lines in this case corresponds to the selected edge, so it appears yellow rather than green because of the highlight). Upon release of the mouse button, a new edge is created.

3.3.5 Extrude

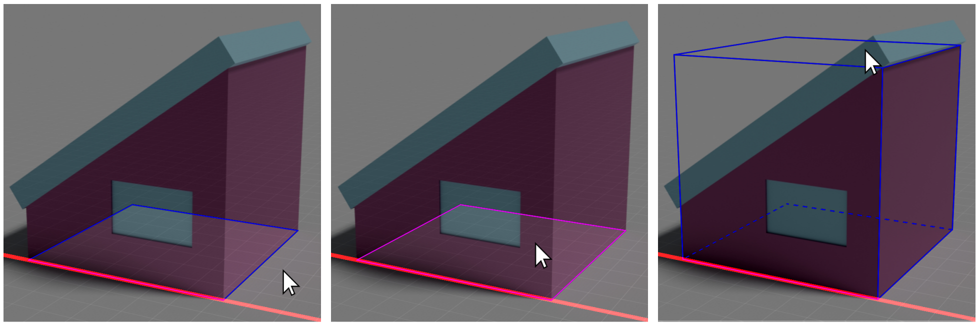

Figure 3.8: Clicking and dragging a face creates all of the faces required for the extrusion.

The Extrude tool is similar to the extrusion tools in conventional CAD packages, and simply expedites what could be done with a combination of drawing, filling, and intersecting. The user may click on a face in the selected plane and then drag, creating a fully bound volume as an extension of that face. This is accomplished by simply duplicating the active plane and then allowing the user to drag it to a new location, which only requires its distance from the origin d_A to vary. The new dragged face has the duplicated plane as its base plane and reuses the list of intersecting planes of the clicked plane. The side faces each have one of the original intersecting planes as a base plane, and its intersecting planes are the active plane, the dragged plane, and its two neighbours in the original intersection list. (See figure 3.8)

3.3.6 Rotate

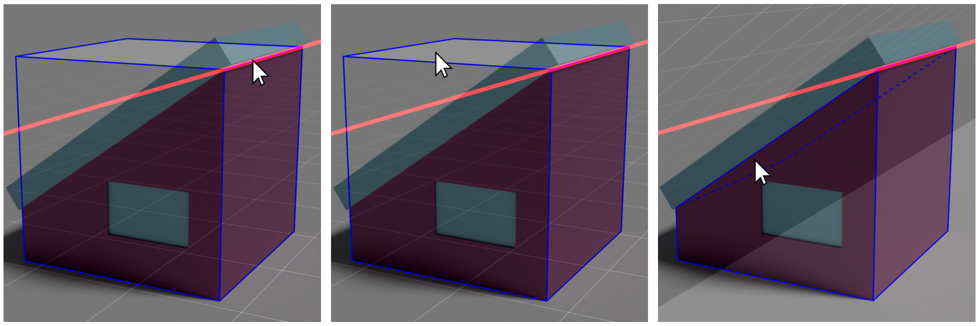

Figure 3.9: Clicking and dragging the selected plane rotates it about the selected edge. Any faces containing that plane will be adjusted accordingly.

The Rotate tool allows the user to rotate the active plane about the selected edge, if the selected plane is in the same frame as the selection-intersecting plane the edge. In order to rotate the plane about the axis defined by the edge, a new frame is created with the other plane's frame as its parent, and the axis of rotation set to the axis of intersection. As both planes are in the same frame, this will simply be the odd axis out: i.e. if the active plane's normal points down the Y axis and the other plane's normal points down the X axis, then the frame's rotation will be about the Z axis. The user may then drag the plane around the axis, which will simultaneously update the planes distance about the origin d_A and its frame's angle θ in order to maintain the selected edge position while rotating the plane.

The Rotate tool may also be used in the case that the above configuration already exists, because the selected plane was rotated before or because a compatible rotated frame was created with the draw tool. In that case, no new frame is created, and the parameters are simply adjusted. (See figure 3.9)

3.3.7 Rectangle

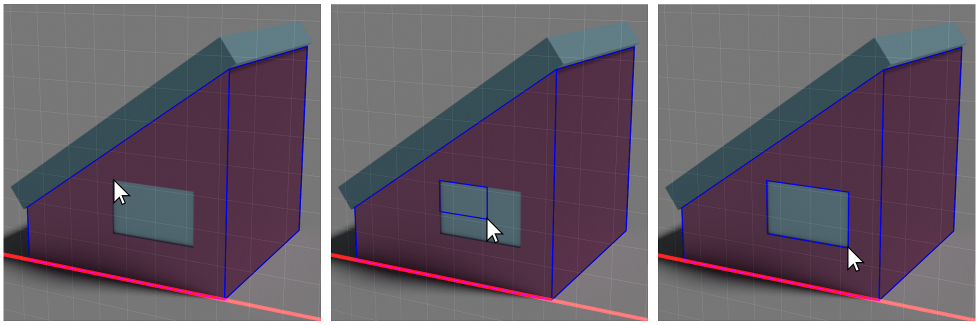

Figure 3.10: Clicking and dragging creates a grid-aligned rectangular face in the selected plane.

Similar to extrude, the rectangle tool is simply a convenient way to accomplish what could be done with the draw and face tools. The user may click and drag on the current active plane, and a new rectangular face will be created from four new intersecting planes defined in the frame implied by the active edge (i.e. the plane intersecting the active plane at the active edge). (See figure 3.10)

3.3.8 Delete

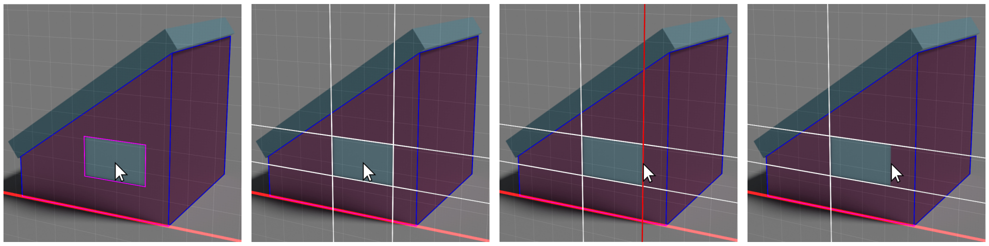

Figure 3.11: Image one and two show deleting a face, unbinding all of its edges. Images three and four show deleting one of those edges.

The user may delete both faces and edges. When a face is deleted, all of its edges return to the 'unbound' state. Only unbound edges may be deleted.

If an edge that is depended on by other faces is deleted, those faces are also deleted. (See figure 3.11).

3.4 Measurement Operations

The tool also provides a number of photogrammetric operations that allow the introduction of residuals from image markings, imported control points, and dimensions. These are integrated into the modelling process in a similar way to the core modelling operations, and rely on the selection state and current selected view. Aside from the marking tool, which is used continuously throughout the modelling process, these operations would typically be done at the end of the workflow shown in figure **??**, as part of the "Scale and export" step

3.4.1 Mark Image

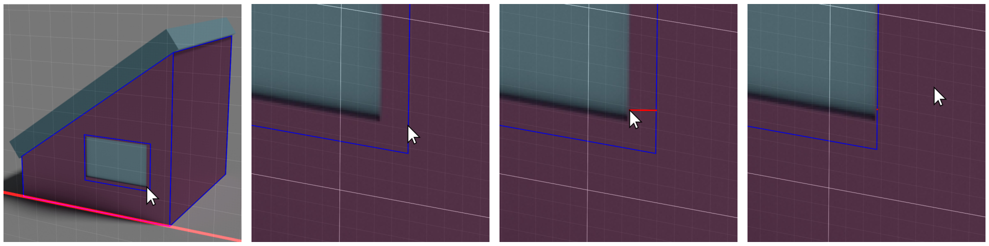

Figure 3.12: Clicking and dragging an edge creates a new marking on the current photograph that the edge should pass through. The fourth images shows the result of adjustment after adding that marking.

With the marking tool, the user may click on any visible bound edge in the scene, and then drag to an image location that edge should pass through and release. This marking is visualized as a red line showing the closest orthogonal distance from the marked point to the edge that has been marked, which is also the way that the residuals are constructed in the optimization. In the overall workflow, this operation is equivalent to box 3: "Add alignment markings for any unmarked geometry visible in the current photo". The user may then invoke the adjustment process in order to attempt to simultaneously minimize all residuals. (See figure 3.12).

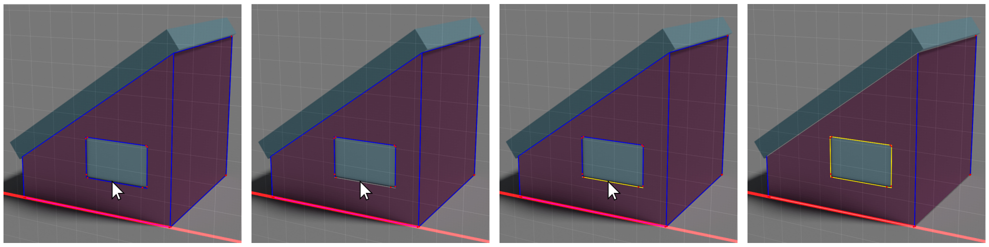

Figure 3.13: Right-clicking an edge cycles through the different weightings available, indicated by colour. The fourth image shows all of the lines assigned typical application-specific weightings.

The marking tool also allows the user to set the weighting given to any markings for a specific edge. The way this is done is currently tailored to the retrofit application, as our process requires the highest accuracy at the openings, moderate accuracy at the corners of walls, and low accuracy at the soffit and foundation lines. To this end, while using the marking tool, the user may cycle through the weights. The weighting for the residuals on a specific edge is shown as a colour: grey for $\frac{1}{3}$, blue for 1, and yellow for 3 (see figure 3.13). These correspond roughly to the relative accuracy requirements for the various features.

3.4.2 Dimension

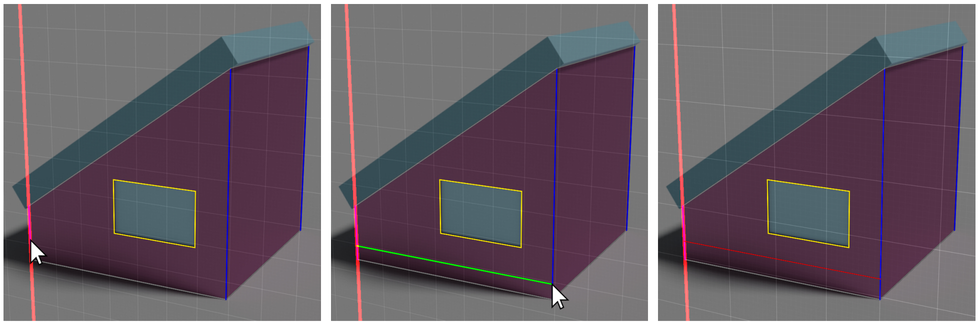

Figure 3.14: Clicking an eligible edge creates a dimension between the other plane of the selected edge and the plane contained in the clicked edge. Image four shows the result of adjustment after adding this dimension - note that the right edge is now grid aligned (as the dimension was a whole number).

The dimension tool allows the user to create dimensional constraints between planes on the same axis within the same frame. To do so, the user first selects an edge composed of one of the planes to be dimensioned, and the plane in which the dimension should be drawn. The selection-intersecting plane will then be one of the planes included in the dimension. The user then may then click on any edge that is composed of a plane with the same normal axis in the same frame (i.e. it is co-planar to the selection-intersecting plane), and a dimension line will be created lying along the selected plane (See figure 3.14). Initially, the dimension line is only a reference dimension. The user may create a constraint from that dimension line by clicking on it and typing a known measurement.

3.4.3 Control Point Tool

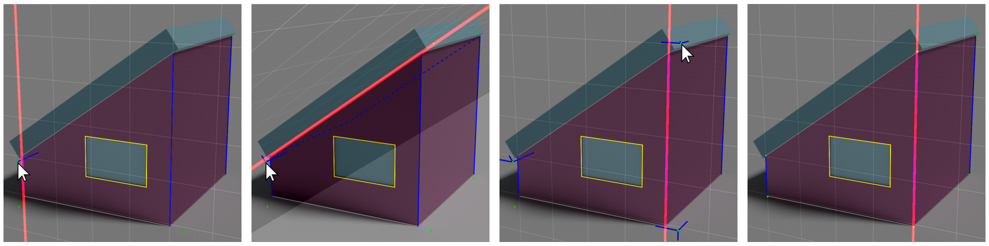

Figure 3.15: Clicking a control point adds a residual between it and the selected plane. Image four shows the result of adjustment.

The control-point tool allows the user to bind imported sets of coordinates to the geometry, so that the imported points may constrain the scale and relative locations of model planes. The user should first align the points to the model using standard rotation/translation controls. Then, the use may bind a coordinate to an individual plane by first selecting that plane, and then clicking on the coordinate. This will result in a blue 3D line orthogonal to the selected plane and terminating at the control point. This line indicates the current distance between the plane and the point, and is the residual that will be minimized in adjustment. Each point may constrain the location of multiple planes. (See figure 3.15)

For the purposes of the adjustment, the sets of control points are considered to be essentially cameras, so that when adjustment level 1 is invoked, all existing sets of control points will be rotated and translated to best fit the bindings created without yet affecting the modelled geometry. This helps stabilize the reconstruction process.

Chapter 4

Parametrisation

The key novelty in the bundle-adjustment functionality of our tool is that it models geometry using planes instead of points. The overall goal is to compute the parameters of a set of planes that best describe the surfaces of the building to be reconstructed. The positions, rotations, and internal parameters of the camera are also incidentally computed during this process, as they must be known to relate the images to the geometry.

The user specifies rough estimates for the locations of these planes as well as their relative orientations and how they intersect. The adjustment optimization attempts to align the intersection lines between planes with user-specified points in the image that visually contain the specific line, usually a building corner, window edge, etc. In this section we detail the parametrisation of planes and how they are intersected into lines so that the misalignment between the user-marked points in the photo survey images and those lines may be minimized across all images.

4.1 Plane Formulation

Our Ortho-Photogrammetry tool models a plane A as its normal $\vec{n}_A$ (a unit vector) combined with the signed distance d_A of the plane from the origin of the coordinate system. An arbitrary position vector $\vec{v}$ lies on the plane when the following vector equation holds [30, Section 5.2]:

$$\vec{v} \cdot \vec{n}_A + d_A = 0 \tag{4.1}$$

4.2 Axis-Aligned Planes

However, in order to reduce the total number of model parameters while enforcing a rectified reconstruction, our method models geometry with axis-aligned 3D planes. A plane A's normal $\vec{n}_A$ is selected from one of three basis vectors:

$$E = \{\vec{e}_x =< 1, 0, 0 >, \vec{e}_y =< 0, 1, 0 >, \vec{e}_z =< 0, 0, 1 >\}$$

$$\vec{n}_A \in E$$

The end result is that a plane A has only one varying parameter d_A that specifies the signed distance of a plane from the origin along the axis it has been assigned, and the normal of that plane is fixed.

4.3 Single-Axis Rotation Frames

In order to relax this constraint to allow rotations about a single axis (such as the planes making up a pitched roof), our method may optionally express the normal $\vec{n}_A$ of a plane A as the product of a rotation about a chosen axis of the coordinate system $R(\theta)$ and a basis vector $\vec{e}$:

$$R(\theta) \in \{R_x(\theta), R_y(\theta), R_z(\theta)\},\ \vec{e} \in E$$

$$\vec{n}_A = R(\theta)\vec{e}$$

This enables the description of common angles in planar buildings with the addition of only one varying parameter θ. This single-axis rotated coordinate frame can be shared by multiple axis-aligned planes. The frames may themselves be nested to allow the description of arbitrary 3D planes, for example:

$$\vec{n}_A = R_x(\theta_1)R_y(\theta_2)\vec{e}_x$$

In general, the elementary rotational frames are organized in a hierarchy, so that a given elementary rotation R_i is either the root rotation in the tree R_0, or has some parent rotation R_j under which it is nested. To calculate the normal $\vec{n}_A$ of a plane A, the process starts by first applying the leaf elementary rotation to A's basis vector $\vec{e}$, following with the parent rotation, and so on until the root is reached. The root rotation is defined as the identity (i.e., it is not rotated). This is analogous to standard hierarchical modeling [26, Section 6.6], but without the usual translation component.

This flexibility is necessary to allow specifying arbitrary planar geometry with nested axial rotation constraints. However, it is worth noting that in our experience with small buildings, most planes are either in the base frame with no rotation (e.g., a wall), or are within a single rotation frame about one of the base axis (e.g., a roof).

4.4 Plane-Plane Intersection Lines

On its own, a 3D plane A does not project to an identifiable feature in a 2D image, so the Ortho-Photogrammetry method first constructs lines in the 3D model by intersecting pairs of planes A and B. The topology of these intersections is specified by the user during modelling. A line $\mathbf{l}(t)$ is formulated as a parametric vector equation:

$$\mathbf{l}(t) = \vec{s} + t\vec{v} \tag{4.2}$$

Here, $\vec{s}$ is a point on the line and $\vec{v}$ is the direction of the line (a unit vector). Varying t (an arbitrary scalar parametrising the line) allows specification of any point along the line [30, Section 5.1]:

We calculate $\vec{s}_{AB}$ and $\vec{v}_{AB}$ of the intersection line $\mathbf{l}_{AB}(t)$ as follows. The direction $\vec{v_{AB}}$ of the line of intersection of planes A and B is:

$$\vec{v}_{AB} = \vec{n}_A \times \vec{n}_B \tag{4.3}$$

Consequently, a point $\vec{s}_{AB}$ on this line can be described as follows:

$$\vec{s}_{AB} = -\vec{n}_A d_A + (\vec{v}_{AB} \times \vec{n}_A)\frac{-(\vec{n_B} \cdot -\vec{n_A} d_A + d_B)}{\vec{n}_B \cdot (\vec{v}_{AB} \times \vec{n}_A)} \tag{4.4}$$

As this equation is not readily available in the literature, we provide a derivation in A.1.

4.5 Projection into an Image

A line in 3D may be projected into an image by projecting any two different 3D points on that line into the image using a standard camera transform. Subbing parameter values $t = 0$ and $t = 1$ into Equation 4.2, our method uses the following two points on the line of intersection of planes A and B:

$$\vec{p} = \mathbf{l}_{AB}(0) = \vec{s}_{AB}$$

$$\vec{q} = \mathbf{l}_{AB}(1) = \vec{s}_{AB} + \vec{v}_{AB}$$

Each point $\vec{p}$ and $\vec{q}$ is independently projected into the image by a standard pin-hole camera transformation. Each point (denoted here as $\vec{p}$, though the same equations are used for both points) is first transformed into camera-space:

$$\vec{p}_{cam} = -\mathbf{q}(\vec{p} - \vec{t})$$

where $\mathbf{q}$ is a quaternion [30, Section 4.6] specifying the orientation of the camera, and $\vec{t}$ is a vector representing the translation of the camera. The positions and rotations of the camera parameters are allowed to vary at every adjustment level.

Next, we project each point into the image. The camera calibration matrix, K, is defined as:

$$K = \begin{bmatrix} \frac{f}{s_x} I_x & & \frac{p_x}{s_x} I_x \\ & \frac{f}{s_y} I_y & \frac{p_y}{s_y} I_y \\ & & 1 \end{bmatrix}$$

where f is the focal length of the lens (in mm), p_x and p_y are the coordinates of the principle point on the sensor in mm, s_x and s_y are the dimensions of the camera sensor (also in mm), and I_x and I_y are the dimensions of the image in pixels. We assume that the sensor has square pixels and no skew (a reasonable assumption with modern image sensors [18]).

By applying K and then performing perspective division, the idealized pixel-space coordinates for each point are computed (more on this in the next Section) :

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = K\vec{p}_{cam}$$

$$\vec{p}_{ideal} = \begin{bmatrix} \frac{x}{z} \\ \frac{y}{z} \end{bmatrix}$$

A note for optimization: within the matrix K, regardless of the optimization level chosen, only the focal length f and the principle point coordinates p_x, and p_y are allowed to vary; the other parameters remain constant. Of s_x and s_y, only s_y is directly determined by the EXIF data; s_x is calculated from s_y using the aspect ratio determined by the image size in pixels (i.e. $s_x = \frac{s_y}{I_y} I_x$), which forces square pixels.

4.6 Lens Distortion

The points $\vec{p}_{ideal}$ and $\vec{q}_{ideal}$ then lie on the projection of the line $\mathbf{l}_{AB}(t)$ into an idealized image. From here, it is possible to compute the error (misalignment) of the projection of that line from an associated marking $\vec{m}_{pix}$ in the image, as the closest distance from the marked point to the projected line. However, assuming a pinhole camera, most lenses exhibit some amount of distortion. Rather than distort the lines and then compute the closest distance to the markings in the image, the tool *undistorts* the marking $\vec{m}_{pix}$ and computes the error in an idealized pixel space. Using a two parameter radial distortion model, its idealized counterpart is computed as follows.

First, the location of the marking on the image sensor is computed, in a coordinate system centered on the principle point. Even though this centre point is not necessarily the true center of radial distortion, it is typically close enough [23, Section 7.4]):

$$\vec{m}_{sensor} = \frac{\vec{m}_{pix}}{\vec{I}}\vec{s} - \vec{p}$$

Next, radial distortion correction is applied to the position of the marking on the sensor, based on the radial distance from the principle point:

$$r^2 = m_x^2 + m_y^2$$

$$\vec{m}_{corrected} = \vec{m}_{sensor} + k_1 r^2 + k_2 r^4$$

Finally, the marked point's location in the idealized pixel coordinates is computed:

$$\vec{m}_{ideal} = \frac{\vec{m}_{corrected} + \vec{p}}{\vec{s}}\vec{I}$$

4.7 Reprojection Error

Having two points $\vec{p}_{ideal}$ and $\vec{q}_{ideal}$ projected from the line $\mathbf{l}_{AB}(t)$ and an undistorted marking $\vec{m}_{ideal}$, all in idealized pixel space, the next step is to calculate the residual $\vec{x}_{error}$ between the line and the marking using the formula for finding the shortest vector from a 2D point to a 2D line (dropping the '*ideal*' subscript for readability):

$$\vec{x}_{error} = (\vec{m} - \vec{p}) - (\vec{q} - \vec{p})\frac{(\vec{q} - \vec{p}) \cdot (\vec{m} - \vec{p})}{(\vec{q} - \vec{p}) \cdot (\vec{q} - \vec{p})} \tag{4.5}$$

See A.2 for a derivation.

4.8 Optimization

Having formulated the model for bundle adjustment, our tool employs the nonlinear least-squares optimizer Google Ceres [1] to model and solve it. Ceres can automatically differentiate this problem using jets, so no derivative calculations are necessary.

The optimization is composed of a set of the residual computations just outlined (i.e., the vector difference between the marked point and the closest point on the corresponding intersection line), one per marking in each photograph. In terms of the actual optimization, this results in two residuals (each component of the 2D vector $\vec{x}_{error}$) being added to the optimization. They are automatically squared by the optimizer.

Our tool uses Ceres' Levenberg-Marquardt optimizer, and specifies the Schur Complement optimization option to take advantage of the sparsity of the bundle adjustment [49]. Also, for rotations specified as quaternions, the tool uses Ceres' built-in local parametrisation for quaternions, as quaternions are not a minimal parametrisation for rotations [1].

4.9 Calibration and Initialization

Typically, photogrammetry requires a careful calibration process for accurate results. Even with a calibrated camera, bundle adjustment is usually unlikely to converge unless it is initialized to a reasonable approximation of the correct solution. This can be accomplished via algebraic minimizations such as the Direct Linear Transform [49] that have closed-form solutions that yield a good enough starting point that the nonlinear optimization will converge.

However, by constraining the problem to rectilinear geometry, this objective becomes much more amenable to optimization. With moderately careful modelling and default camera parameter initialization, we have found that Ortho-Photogrammetry almost always converges.

The initialization for the geometry and extrinsic camera parameters is based on a very rough alignment built into the modelling process. The intrinsic camera parameters are initialized as follows: the lens distortion parameters k_1 and k_2 are set to zero; the principle point $\vec{p}$ is set to the to the sensor center; and the focal length and format size are initialized using the EXIF data from the photos.

4.10 Scaling

A fundamental limitation of photogrammetry is that it can only reconstruct geometry up to some unknown scale-factor s [23, Section 1.3]. To address this limitation, the tool allows the user to provide real-world dimensions in two different ways.

First, the user may specify a measured distance D between two parallel planes A and B in the reconstruction (i.e. two planes with the same basis $\vec{e}$

in the same rotational frame R). The potential deviation from this distance is added as an error term in the optimization with a configurable weight w. This is trivial to do, as the planes occupy the same coordinate system and are parallel, so the residual is:

$$d_{error} = w(|A_d - B_d| - D)$$

Alternatively, the user may import control points shot with a total station, and then specify that each point should lie in up to 3 different reconstructed planes. Each deviation is then added as an error term to the optimization. To do so, the system first transforms the specified plane's normal to model-space as set out in Section 4.3, yielding a plane A. Next, it transforms the specified total station coordinate $\vec{c}$ to model-space (the coordinate system of the building being reconstructed) using the current estimation of the station orientation $\mathbf{q}$ and position $\vec{t}$ for that coordinate:

$$\vec{c}_{model} = \mathbf{q}\vec{c} + \vec{t}$$

Finally, the distance from that coordinate to the plane A can be computed [30, Section 5.2]:

$$d_{error} = w(\vec{c}_{model} \cdot \vec{n}_A + d_A)$$

Note that specifying three planes in the same rotational frame, each with a different basis vector $\vec{e}$, is equivalent to specifying a 3D point [23, Section 3.2.1]. In this case, the component-wise error specified above is equivalent to the ℓ_2 error between that 3D point and the total station point, because the optimizer will square each residual in its construction of the least-squares optimization.

In theory a single scalar factor s is needed, and therefore either a single dimension or a pair of total-station coordinates should suffice to scale the geometry. In practice, a small number of extra measurements or control points can further constrain the geometry and provide a better solution, as found in other photogrammetric building reconstruction investigations [5], [16]. Additionally, some points may be impossible to photograph well, so it may be necessary to supplement the photogrammetric process with additional measurements.

To this end, the rotation and position parameters $\mathbf{q}$ and $\vec{t}$ for each total station are also adjusted during optimization, so that it is possible to use multiple setups that need not share coordinates or have a known transformation between them. The tool uses the photogrammetric geometry to reconstruct each total-station's location and orientation in the same way that it computes the camera location and orientation for each photograph. Just as in the photographic case, the tool enables the user to provide a rough alignment of the total-station coordinates with the model in order to initialize the optimization. It is also possible to combine measured dimensions with total-station coordinates.

4.11 Conversion to a Conventional Mesh

To render a view during modelling or export a standard points-based model, each vertex $\vec{x}_i$ of the face can be calculated by the intersection of the 3 planes that uniquely define it. A stable way of doing this is to first calculate the line of intersection of the planes E_i and E_{i+1}, which are the planes that define (in conjunction with the base plane) the edges adjacent to $\vec{x}_i$. Using Equation 4.4 given in Section 4.4, the line of intersection $\vec{s}+t\vec{v}$ is computed. Next, the point at which that line intersects with the base plane B can be identified using the equation for the the intersection of a line and a plane [30, Section 5.2]:

$$\vec{x}_i = \vec{s} + \frac{-(\vec{n_B} \cdot \vec{s} + d_B)}{\vec{n_B} \cdot \vec{v}} \vec{v} \tag{4.6}$$

With a points-based polygonal face established, it is possible to use polygon triangulation [13, Section 3.1] to simplify each face into triangles, and then combine them to create a triangle mesh suitable for rendering or export.

Chapter 5

Experiments

We conducted four experiments using photos and ground-truth measurements of buildings that are currently in the design phase for prefabricated energy efficiency retrofits. Our analysis is focused on the openings (windows and doors), as they are the areas that require the most stringent reconstruction so that the openings on the prefabricated panels can be successfully aligned with them. Other areas, such as soffit lines and building corners, are less important to the reconstruction, as the panels have significant adjustment capability at their borders. Additionally, because of factors like stucco build-up at the corners, significant variability in corner boards, and sag in the soffit, the actual building geometry at the corners is difficult to define by any means of non-destructive measurement. Given these difficulties combined with the fact these measurements are also significantly less important, we chose to focus on the openings. We report 1σ error in millimeters in order to assess whether our accuracy is suitable for prefabricated energy efficiency retrofits.

5.1 Buildings 1, 2, & 3

Buildings 1 - 3 are single family homes. Each building was photographed with a DJI Mavic 2 Pro UAV, with an integrated 20 megapixel camera with a 1" sensor and 10mm focal length (full-frame equivalent 28mm) lens. We shot in aperture-priority mode with f5.6 and ISO set to 400. We established the focal plane using auto-focus for a shot in which the building just filled the image, then locked the focus at that setting. Changing the focus changes the

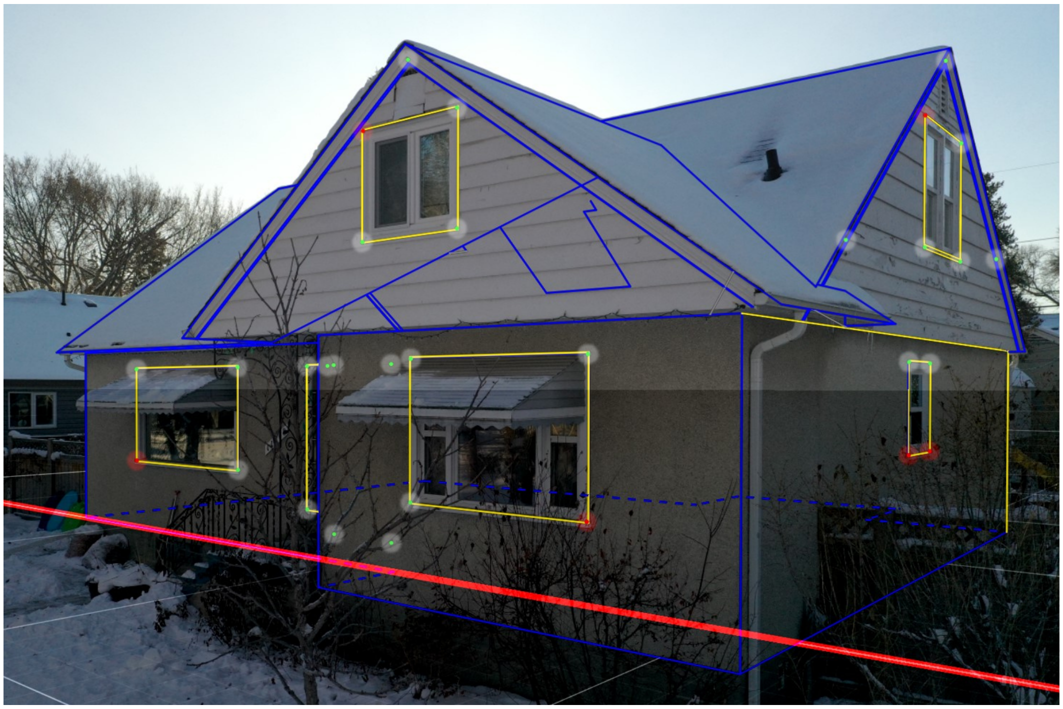

Figure 5.1: The total station points from the two visible elevations of building 1 (one station setup per side). The 6 red points are used for establishing scale.

calibration slightly [11], so it is best to take all photos at the same focus. In total, we took 56 photos of building 1, 77 photos of building 2, and 137 photos of building 3.

We established the ground truth for both buildings using a Nikon 322+ total station with an angular accuracy of 5" and a range accuracy of $3mm{+}3ppm$. Edges are the natural feature to reconstruct with photogrammetry, but because of beam divergence [28], electronic distance measurement is typically less accurate for edges and corners, so we placed four 20mm adhesive retroreflective targets at a known distance from the edges of each opening to be reconstructed (one for each corner). There is no overlap between the coordinates captured across station setups. However, the geometry that would be present within a single panel is always captured within a single station setup, so the relative geometry we need to fit for manufacturing purposes is captured. We captured 85 coordinates for building 1, 66 coordinates for building 2, and 68 coordinates for building 3.

The geometry was reconstructed using 7 of the 56 photos for building 1, 8 of the 77 photos for building 2, and 7 of the 137 photos for building 3. These photos were chosen to approximate the generic network discussed in Section 3.1 while avoiding occlusions. For building 1, each edge was marked on average 1.59 times, and had markings in on average 0.83 photos. This number can be less than one because both planes defining some edges may be sufficiently constrained by other edges, so that no markings end up being necessary or desirable for that particular edge (if it is highly occluded or otherwise difficult to identify in the photographs). For building two, each edge was marked on average 3.15 times and had markings in on average 1.47 photographs. The higher averages here are due to the fact that building 2 had several long edges with some slightly crooked sections, meaning that the easiest way to locate the overall trend of the line was to quickly mark many points along the edge in the photograph. Each edge of building three was marked on average 1.44 times, and was marked in on average 0.87 images.

For building 1, we found it best to use 6 points total to constrain the scale of the reconstruction. Using more coordinates than this resulted in quickly diminishing returns. Using the scaling method detailed above, we used 3 coordinates each from two total-station setups facing the east and south elevations respectively (Stn. 1 and Stn. 5 in Table 5.1), for a total of 6 coordinates (see figure 5.1).

For building 2, we used a similar strategy, scaling the reconstruction with 3 coordinates from the south elevation and 4 coordinates from the west elevation. The extra coordinate on the west elevation was used to help constrain part of the roof line that was difficult to photograph.

For building 3, an adjacent building on the east elevation only allowed for photos at significantly oblique angles, so we used an extra six coordinates in order to constrain the locations of the openings on that elevation. In total, we scaled the building with 12 coordinates from three setups: 3 on a setup on the north elevation, 3 from a setup on the west elevation, and six on a setup at the corner of the south and east elevations, with 4 coordinates on the east elevation (one on each opening) and 2 coordinates on the south elevation to tie the difficult-to-photograph openings into the rest of the reconstruction geometrically.

We used the remaining coordinates to verify the accuracy of the reconstructions. After freezing the reconstructed geometry, we computed the best fit for each of the total station setups' position and rotation to align them with their respective reconstruction via least-squares adjustment, then found the 1σ error between the total station coordinates and each plane the coordinate should lie in from the reconstruction.

5.2 Building 4

Building 4 is a four-unit multifamily, with a photosurvey and rectified drawings provided by CanmetENERGY. The building was captured with a Leica P40 laser scanner with a range accuracy of $1.2mm+10ppm$ and an angular accuracy of 8". Rectified drawings were produced by an architectural technologist.

The photos were taken with a NIKON D7200 (which has a 24.2 megapixel APS-C sensor) with a 20mm lens (full-frame equivalent 30mm). The photos were taken in manual mode, with 1/500 shutter speed, f-stop 8, and the focus ring locked after establishing focus. The ISO was allowed to vary.

A total of 135 photos were available, 7 of which we used for the reconstruction. Each edge was marked on average 1.24 times, and was marked in on average 0.95 photos. We used a total of 8 dimensions (the width and height of the left-most and right-most bottom floor windows on each side) to scale the model.

After investigating the provided rectified geometry overlayed on the pointcloud, we determined that the human rectification had favoured opening geometry accuracy over the building corners and foundation and soffit lines, as they had visually apparent large discrepancies. For this reason, we again restrict our analysis to the openings, including opening widths and heights as well as the center-to-center distances between horizontally and vertically adjacent openings.

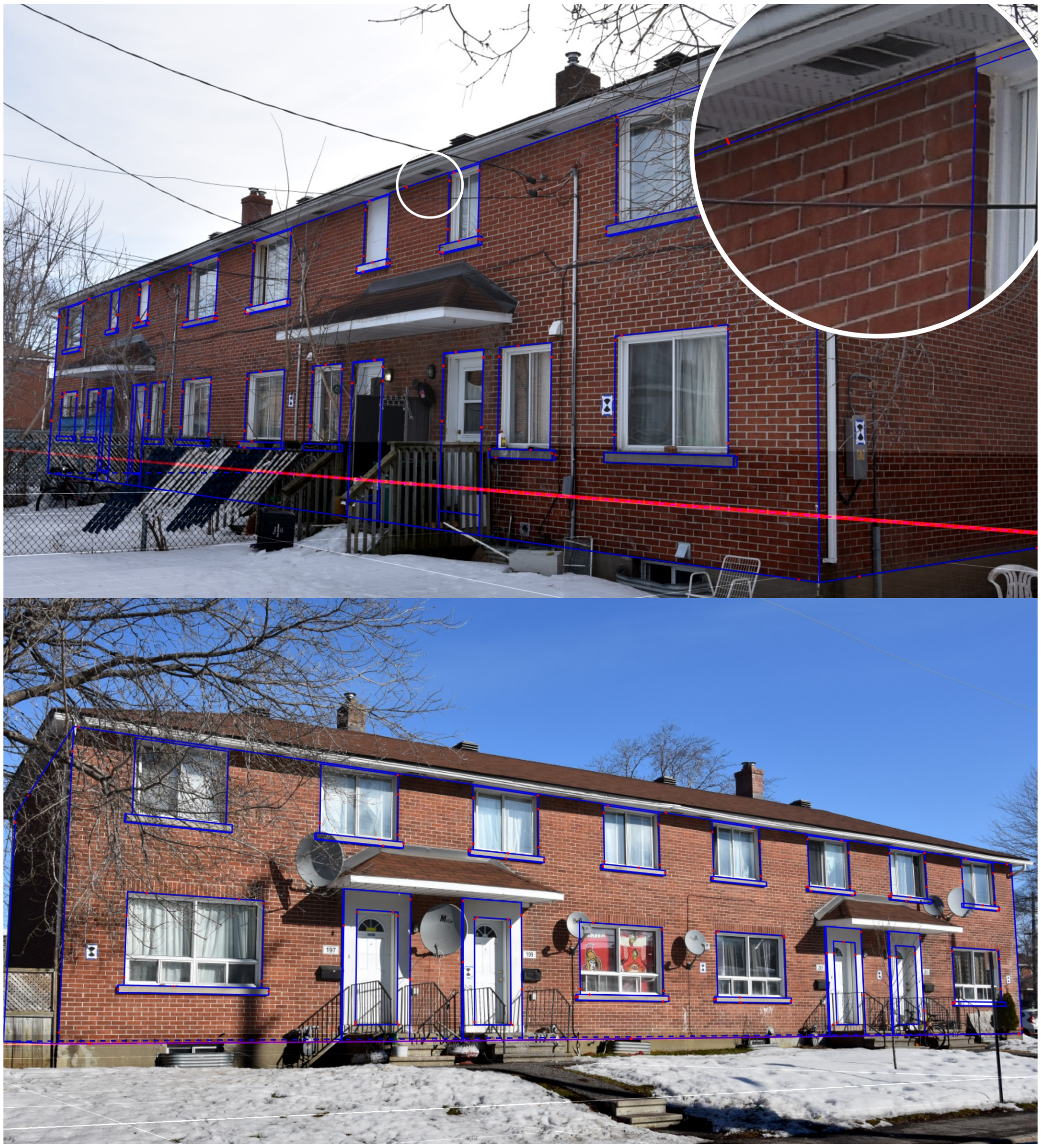

Figure 5.2: **Top:** The cluttered side of building 4. Also shown is a magnified view of several marked points (in red) demonstrating that they are actually lines showing the distance between the marked point (the edge in the image) and the model intersection line (blue). **Bottom:** The uncluttered side of building 4, which was reconstructed from two photos.

Table 5.1: Results. N is the number of measurements and $1\sigma(mm)$ is the standard deviation of the error of the measurements in millimetres

Dataset	**Subset**	N	$1\sigma(mm)$
Building 1	Stn. 1	54	2.83
2 pt. scale	Stn. 2	60	5.93
	Stn. 3	36	3.42
	Stn. 4	48	3.60
	Stn. 5	27	3.44
	All	225	**4.13**
Building 1	Stn. 1	54	2.87
6 pt. scale	Stn. 2	60	3.87
	Stn. 3	36	3.49
	Stn. 4	48	3.10
	Stn. 5	27	2.54
	All	225	**3.25**
Building 2	Stn. 1	57	4.18
7 pt. scale	Stn. 2	41	2.98
	Stn. 3	21	2.20
	Stn. 4	31	3.55
	All	150	**3.55**
Building 3	Stn. 1	20	2.24
12 pt. scale	Stn. 2	66	3.07
	Stn. 3	49	4.15
	Stn. 4	55	3.02
	All	190	**3.27**
Building 4	**All** uncalibrated	120	3.20
	All calibrated	120	3.30
	Photomodeler	98	7.58

5.3 Discussion

5.3.1 Building 1

We found that using six points to constrain the scale of the geometry, we have a 1σ error of $\pm 3.24mm$. If we only use two coordinates to scale the model, our 1σ error is approximately 4.13mm, which is on the edge of acceptable for our retrofit method, so collecting a few additional points while establishing scale is worth doing. See Table 5.1 for the results broken down by scaling method and total-station setup. In particular, it is interesting to note the improvement of the fit to Station 2's geometry, which is all on the south elevation of the building, where no control points were used. Three of the coordinates added for the 6 pt. scaling are on the opposite (north) elevation, captured by station 5. Because the geometry is constrained to be rectilinear, some constraints on the opposite elevation can be expected to propagate through to the opposite side of the building, and we seem to be seeing that here, as the accuracy of the reconstruction on the south elevation increases substantially with 6 pt scaling.

It is worth noting that this error **includes** the error resulting from rectification. Our method automates fitting orthogonal geometry to the underlying data, and does not require human judgement of how to best rectify slight deviations from square and plumb. This means that panel design can be accomplished easily using the resulting geometry, as it is easily used in CAD software.

Additionally, this experiment is likely reaching the limitations of the Nikon 322+ total station's range measurement, as noted in [43] and [48].

5.3.2 Building 2

Building 2 is a useful case study because it is a fairly challenging subject due to occlusions. In figure 5.3.2, we see that the backyard has a number of items obscuring the corners of the openings as well as the whole elevation. In the top photo, the deck, barbeque, table, storage box, rainwater collector, and tree all obscure corners of different openings. However, the lines that locate these corners are sufficiently visible to be marked. Additionally, though the

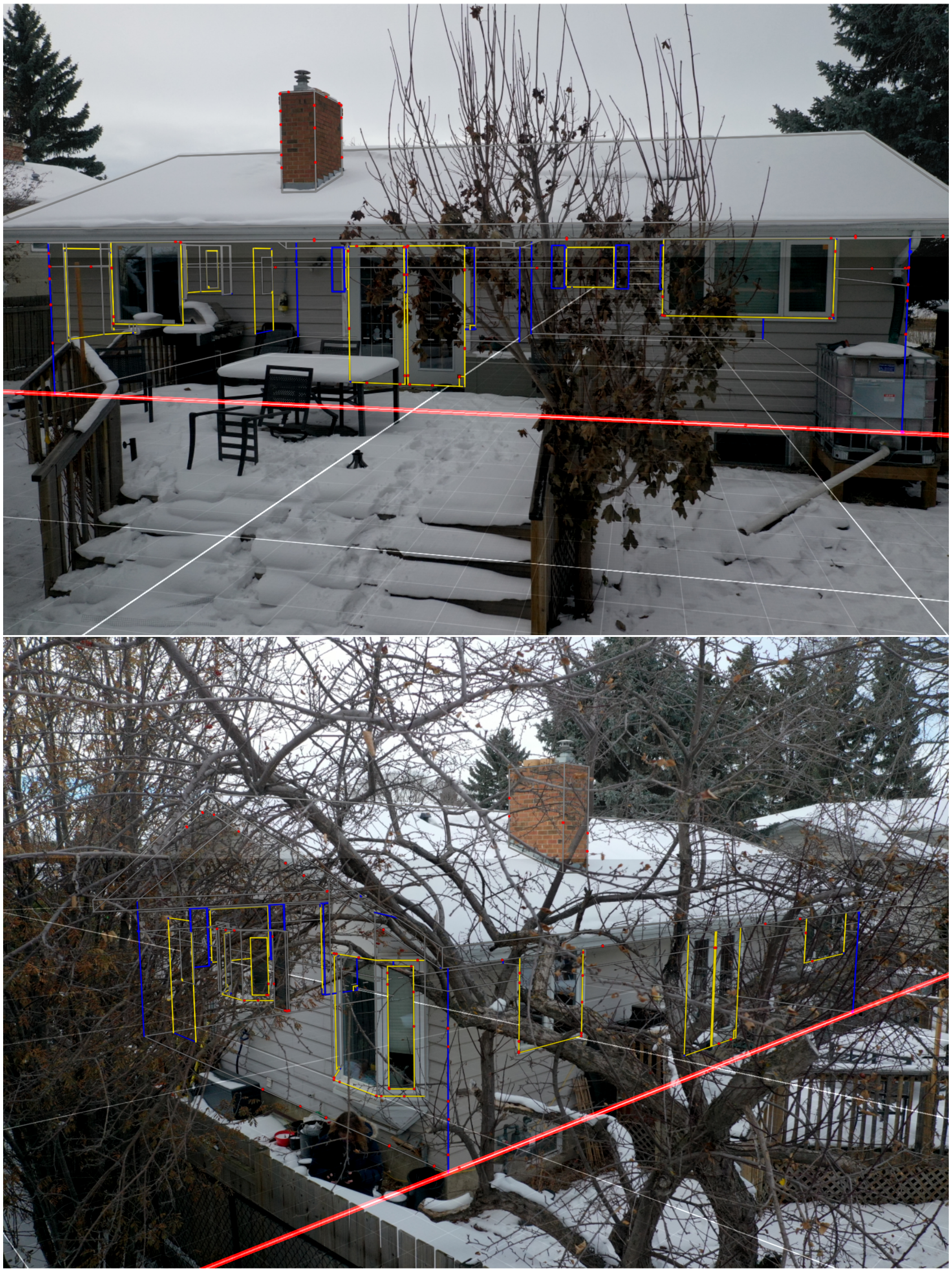

Figure 5.3: Two photos demonstrating the challenging occlusions for building 2.

top corners of both of rightmost and leftmost window are obscured by the soffit in this photo, one of two lines (the verticals) that locate each of these corners is visible, meaning this photo still partially constrains them.

The bottom photo in figure 5.3.2 is a more extreme example of difficult occlusions. The vegetation surrounding this side of the building makes it very difficult to see the geometry at all. Here again, however, the inherent occlusion resistance of line features makes it possible to add quite a number of markings through gaps in the branches.

Additionally, though we cannot mark all geometry on both elevations, the orthogonality constraints imply that even when we only are able to mark part of an opening, we are in some sense improving the geometry of the whole opening, as typically all of the lines making up a rectangular opening lie in one plane, and any one line can improve that plane's parameter estimation. Despite these occlusion difficulties, the reconstruction's 1σ error is $\pm 3.55mm$, which is very similar to building 1, and sufficient for panel-based retrofitting.

5.3.3 Building 3

Building 3 demonstrates how poor photographs can be augmented by external measurements. This building is closely adjacent to a larger building to the east that made a straight-on shot impossible, so we are limited to the two oblique angles shown in figure 5.3.3. In the bottom image in figure 5.3.3, we note that one of the openings has three of four corners obscured, so that it is primarily visible in only one highly oblique photo. We might suspect that this would hurt the reconstruction accuracy of this elevation, and indeed, if we remove the six additional constraint coordinates, the accuracy of the coordinates shot from station 4 worsens dramatically from 3.02mm 1σ error to 12.21mm 1σ error.

We would argue that this ability to simply intermingle coordinate measurements and photographic data is quite useful. We still benefit from the photographic documentation of the photogrammetric process, wherein we may see directly what lines correspond to what photographic features by overlaying the reconstruction. This means we can have reasonable confidence that

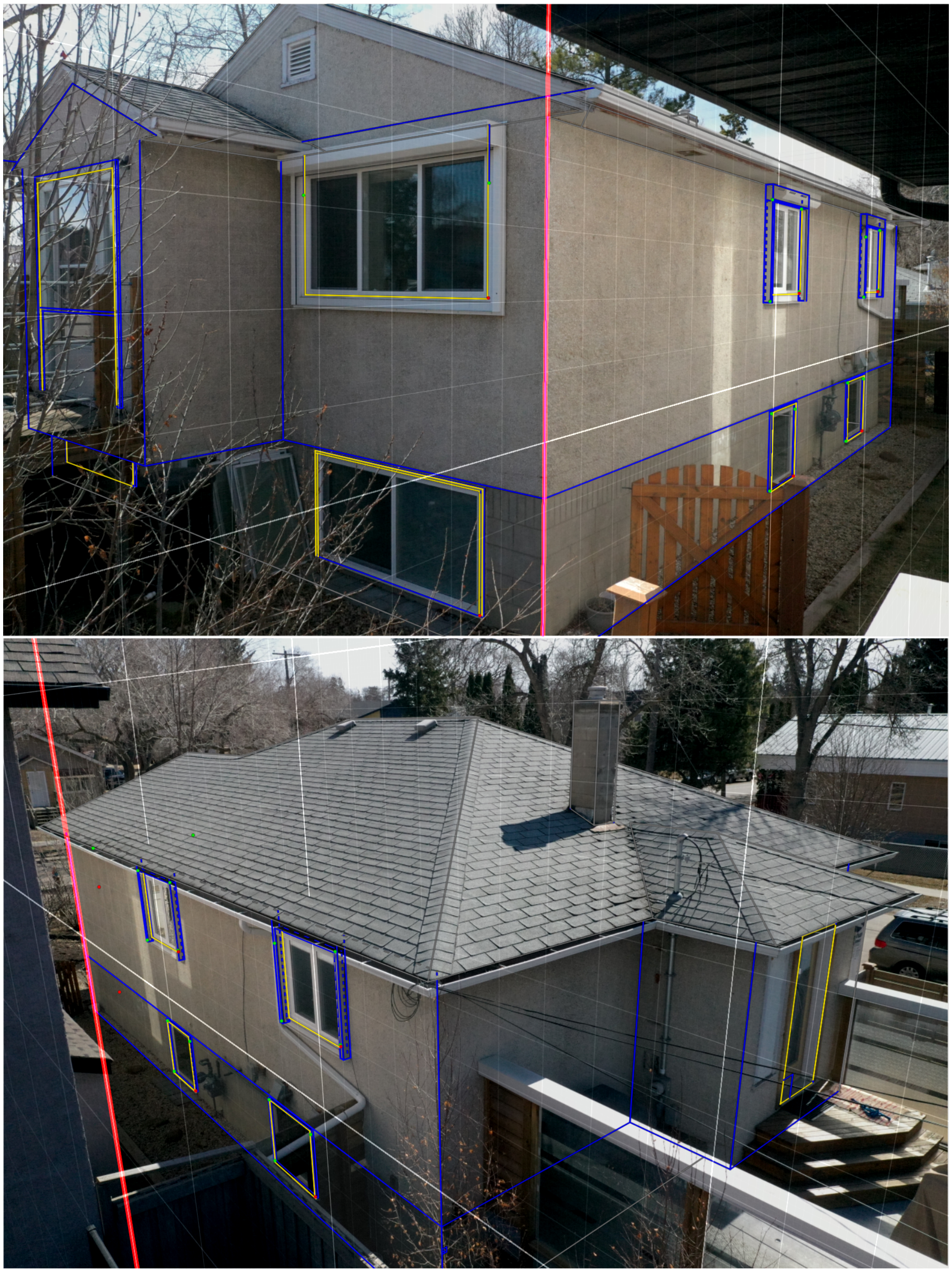

Figure 5.4: The two photos used to reconstruct building 3's east elevation. The red dots are the total station coordinates used to constrain the geometry, while the green dots are used to check the reconstruction.

the correct topology was captured. Furthermore, only a single coordinate is required per window to sufficiently constrain the four coordinates of the window corners, implying that though the photogrammetric data is far from ideal, it is still useful if it can be sufficiently constrained, and manual collection of measurements can be kept to a minimum.

5.3.4 Building 4

It is useful that we have independently human-rectified drawings for building 4, as we can compare our rectification with a human rectification. Our results here are particularly promising, as we achieve a similar error despite reconstructing a significantly larger building (building 1 and 2's longest sides are 12.2m, 12.5m, and 12.2m respectively, while building 4's longest side is 25.9m.)

We also have a manually-marked standard points-based 3D reconstruction created by CanmetENERGY in order to evaluate the commercial photogrammetry software Photomodeler for panel-based retrofitting. The reconstruction was accomplished using the photo survey provided, and used 29 out of the 135 photos. It was missing some geometry necessary to compare all dimensions, but we were able to compare 98 out of 120 dimensions. The error of the Photomodeler reconstruction is 1σ error was $7.58mm$, which is worse than the threshold we have identified ($\pm 4mm$ 1σ error), and CanmetENERGY has not opted to further evaluate a points-based approach. In contrast, our reconstruction has a $3.49mm$ 1σ error, which is significantly closer to the rectified drawings, despite using only 7 photos (less than a quarter used in the conventional reconstruction).

We also ran an adjustment using the same camera calibration used by CanmetENERGY for their Photomodeler reconstruction, created in Photomodeler with from multiple images of coded targets on a planar surface. We found that our accuracy actually decreased very slightly using this calibration, suggesting that our on-the-job calibration created during adjustment is reasonably good.

If the photos had been taken by a UAV, even fewer photographs would likely have been necessary. One entire side of building two was relatively free of clutter (see figure 5.2), and was reconstructed entirely from two photographs. We expect that the other side of the building, could have been reconstructed to a similar accuracy using only 2 or 3 photographs, as the UAV could have achieved unobstructed line-of-site to much of the geometry hidden behind fences for a human camera operator.

It is also worth noting that while the Leica P40 used to establish the ground-truth costs (at time of writing) approximately $90,000 USD, the Nikon D7200 camera and 20mm lens used to capture the photos costs approximately $2,000 USD. The Mavic 2 Pro drone we used for buildings 1 and 2 costs approximately $1,500 USD. These significant cost savings in equipment purchase could enable even small contractors to do such reconstructions themselves.

Chapter 6

Conclusion

In this thesis, we have presented Ortho-Photogrammetry, a novel method that reparametrises the photogrammetric process in terms of orthogonal planes so that the user may encode the natural priors of building geometry into a 3D reconstruction. The method is implemented in a tool that guides users to model geometry on top of a set of photographs and then adjusts the model correspondingly, with no initialization process required. Ortho-Photogrammetry assumes that buildings can be modelled as axis-aligned planes, as detailed in Section 4, and the workflow of our tool supports a process for adjusting these planes based on lines in photographs.

We have showed in Section 5.3 that our method achieves the accuracy of 4mm 1σ error for window and door dimensions, as required for panel-based prefabricated retrofits using Ortho-Photogrammetry. Additionally, we have showed in Section 5.3.4 that the method is also sufficiently accurate when benchmarked against a human-rectified laser scan, which provides additional support for the validity of the automatic rectification process. As only 7 or 8 photographs were used in each reconstruction, and each line was marked in on average approximately one photograph, the total user-interaction theoretically required seems similar to what would be required for tracing a laser scan, in which case each line should be inspected independently at least once in order to confirm that the reconstruction is viable. Furthermore, the results were achieved without any separate camera calibration procedure, and Section 5.3.1 demonstrates that it is possible to use from photographs taken with consumer

UAV.

We are currently assuming that a small number of control points will be established with a total-station to constrain and verify the geometry, especially on more complex buildings. However, the good results possible on Building 2 by simply scaling the model with a few window dimensions suggest that only a tape-measure or laser-distance meter is sufficient to scale and constrain simpler buildings.

6.1 Future Work

There are some limitations to the current work. Though marking is quick and efficient, modelling is currently somewhat arduous, when compared to a commercial modelling tool. Capturing a photo-survey with a drone as well as scaling measurements takes about two hours. The whole modelling process takes approximately eight hours currently, and that time is dominated by modelling. This is approximately the same amount of time our industry partner requires to create a model suitable for panel-design using a workflow they developed for tracing laser-scan point-clouds, so currently labour costs are similar between our method a laser-scanner-based method. Additionally, issues like sagging and bowing of the existing building geometry are not captured by the resulting 3D model (though they tend to be somewhat visible when the rectified geometry is superimposed over the photographs), and there is the potential for incorrectly modelled geometry or incorrectly marked edges to undermine the accuracy achievable.

In the future, we plan to integrate our method with an existing 3D modelling tool, allowing users to model in a familiar setting, which would likely improve their efficiency. It should be possible to implement a bi-directional transformation between a normal points-based mesh and our constrained planar mesh, so that the optimization of the geometry can still be accomplished using our well-suited parametrisation.

We also plan to investigate incorporating laser scan data directly into the optimization. The underlying plane-based parametrisation of our method should in theory allow for a workflow where point-cloud points from a given laser scan are assigned to specific planes, and then included in the optimization just as our total station coordinates currently are. This would provide a way to register and combine laser scans together in an optimization to find the best fit rectified geometry, and would allow for drone-based photographs to supplement areas that were inaccessible to scan from a tripod.

References

[1] S. Agarwal, K. Mierle, *et al.*, *Ceres solver*. [Online]. Available: `http://ceres-solver.org` (visited on 08/25/2021).

[2] K. B. Atkinson, Ed., *Close Range Photogrammetry and Machine Vision*. Caithness: Whittles Publ, 1996, ISBN: 978-1-870325-46-2.

[3] A. Bartoli and P. Sturm, "Constrained Structure and Motion From Multiple Uncalibrated Views of a Piecewise Planar Scene," *International Journal of Computer Vision*, vol. 52, no. 1, pp. 45–64, Apr. 2003, ISSN: 1573-1405. DOI: `10.1023/A:1022318524906`.

[4] A. Bhatla, S. Y. Choe, O. Fierro, and F. Leite, "Evaluation of accuracy of as-built 3D modeling from photos taken by handheld digital cameras," *Automation in Construction*, vol. 28, pp. 116–127, Dec. 2012, ISSN: 09265805. DOI: `10.1016/j.autcon.2012.06.003`.

[5] A. Borodinecs, J. Zemitis, M. Dobelis, M. Kalinka, and A. Geikins, "Development of Prefabricated Modular Retrofitting Solution forPost-World War II Buildings," in *Proccedings of 10th International Conference "Environmental Engineering"*, Vilnius Gediminas Technical University, Lithuania: VGTU Technika, Aug. 2017, ISBN: 978-609-476-044-0. DOI: `10.3846/enviro.2017.252`.

[6] M. Botsch, Ed., *Polygon Mesh Processing*. Natick, Mass: A K Peters, 2010, ISBN: 978-1-56881-426-1.

[7] J. Coughlan and A. Yuille, "Manhattan World: Compass direction from a single image by Bayesian inference," in *Proceedings of the Seventh IEEE International Conference on Computer Vision*, Kerkyra, Greece: IEEE, 1999, 941–947 vol.2, ISBN: 978-0-7695-0164-2. DOI: `10.1109/ICCV.1999.790349`.

[8] J. M. Coughlan and A. L. Yuille, "The manhattan world assumption: Regularities in scene statistics which enable bayesian inference," in *Proceedings of the 13th International Conference on Neural Information Processing Systems*, ser. NIPS'00, Cambridge, MA, USA: MIT Press, 2000, pp. 809–815. DOI: `10.5555/3008751.3008869`.

[9] A. Criminisi, I. Reid, and A. Zisserman, "Single View Metrology," *International Journal of Computer Vision*, vol. 40, no. 2, pp. 123–148, Nov. 2000, ISSN: 1573-1405. DOI: `10.1023/A:1026598000963`.

[10] S. Daftry, C. Hoppe, and H. Bischof, "Building with Drones: Accurate 3D Facade Reconstruction using MAVs," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, May 2015, pp. 3487–3494. DOI: 10.1109/ICRA.2015.7139681.

[11] F. Dai, Y. Feng, and R. Hough, "Photogrammetric error sources and impacts on modeling and surveying in construction engineering applications," *Visualization in Engineering*, vol. 2, no. 1, Dec. 2014, ISSN: 2213-7459. DOI: 10.1186/2213-7459-2-2.

[12] F. Dai and M. Lu, "Assessing the Accuracy of Applying Photogrammetry to Take Geometric Measurements on Building Products," *Journal of Construction Engineering and Management*, vol. 136, no. 2, pp. 242–250, Feb. 2010, ISSN: 0733-9364, 1943-7862. DOI: 10.1061/(ASCE)CO.1943-7862.0000114.

[13] M. de Berg, O. Cheong, M. van Kreveld, and M. Overmars, *Computational Geometry: Algorithms and Applications*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, ISBN: 978-3-540-77973-5 978-3-540-77974-2. DOI: 10.1007/978-3-540-77974-2.

[14] P. E. Debevec, C. J. Taylor, and J. Malik, "Modeling and rendering architecture from photographs: A hybrid geometry-and image-based approach," in *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, 1996, pp. 11–20. DOI: 10.5555/894876.

[15] J. Deutscher, M. Isard, and J. MacCormick, "Automatic Camera Calibration from a Single Manhattan Image," in *Computer Vision — ECCV 2002*, G. Goos, J. Hartmanis, J. van Leeuwen, A. Heyden, G. Sparr, M. Nielsen, and P. Johansen, Eds., vol. 2353, Berlin, Heidelberg: Springer Berlin Heidelberg, 2002, pp. 175–188, ISBN: 978-3-540-43748-2 978-3-540-47979-6. DOI: 10.1007/3-540-47979-1_12.

[16] M. Faltýnová, E. Matoušková, J. Šedina, and K. Pavelka, "BUILDING FACADE DOCUMENTATION USING LASER SCANNING AND PHOTOGRAMMETRY AND DATA IMPLEMENTATION INTO BIM," *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. XLI-B3, pp. 215–220, Jun. 2016, ISSN: 2194-9034. DOI: 10.5194/isprsarchives-XLI-B3-215-2016.

[17] H. Fathi, F. Dai, and M. Lourakis, "Automated as-built 3D reconstruction of civil infrastructure using computer vision: Achievements, opportunities, and challenges," *Advanced Engineering Informatics*, vol. 29, no. 2, pp. 149–161, Apr. 2015, ISSN: 14740346. DOI: 10.1016/j.aei.2015.01.012.

[18] C. S. Fraser, "Automatic Camera Calibration in Close Range Photogrammetry," *Photogrammetric Engineering & Remote Sensing*, vol. 79, no. 4, pp. 381–388, Apr. 2013, ISSN: 00991112. DOI: 10.14358/PERS.79.4.381.

[19] F. Fraundorfer, K. Schindler, and H. Bischof, "Piecewise planar scene reconstruction from sparse correspondences," *Image and Vision Computing*, p. 12, 2006.

[20] Y. Furukawa, B. Curless, S. M. Seitz, and R. Szeliski, "Manhattan-world stereo," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 1422–1429. DOI: `10.1109/CVPR.2009.5206867`.

[21] R. Garay Martinez, J. Benito Ayucar, and B. Arregi Goikolea, "Full scale experimental performance assessment of a prefabricated timber panel for the energy retrofitting of multi-rise buildings," *Energy Procedia*, vol. 122, pp. 3–8, Sep. 2017, ISSN: 18766102. DOI: `10.1016/j.egypro.2017.07.288`.

[22] R. Gottwald and T. Knabl, "3D Measuring for Building Refurbishment," *Reporter - The Global Magazine of Leica Geosystems*, vol. 59, pp. 28–31, 2008. [Online]. Available: `https://w3.leica-geosystems.com/media/new/product_solution/Leica_Geosystems_TruStory_3D_Measuring_for_Building_Refurbishment.pdf` (visited on 08/25/2021).

[23] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, Second. New York: Cambridge University Press, 2004, ISBN: 978-0-511-18711-7 978-0-511-18618-9 978-0-511-18895-4 978-0-511-18535-9 978-0-511-18451-2 978-0-511-81168-5 978-1-280-45812-5. [Online]. Available: `http://dx.doi.org/10.1017/CBO9780511811685` (visited on 04/24/2019).

[24] Heung-Yeung Shum, Mei Han, and R. Szeliski, "Interactive construction of 3D models from panoramic mosaics," in *Proceedings. 1998 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No.98CB36231)*, Santa Barbara, CA, USA: IEEE Comput. Soc, 1998, pp. 427–433, ISBN: 978-0-8186-8497-5. DOI: `10.1109/CVPR.1998.698641`.

[25] T. Holzmann, M. Maurer, F. Fraundorfer, and H. Bischof, "Semantically Aware Urban 3D Reconstruction with Plane-Based Regularization," in *Computer Vision – ECCV 2018*, V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds., vol. 11218, Cham: Springer International Publishing, 2018, pp. 487–503, ISBN: 978-3-030-01263-2 978-3-030-01264-9. DOI: `10.1007/978-3-030-01264-9_29`.

[26] J. F. Hughes, A. Van Dam, J. D. Foley, M. McGuire, S. K. Feiner, and D. F. Sklar, *Computer Graphics: Principles and Practice*. Pearson Education, 2014, ISBN: 978-0-321-39952-6.

[27] D. Hutchison, T. Kanade, J. Kittler, J. M. Kleinberg, F. Mattern, J. C. Mitchell, M. Naor, O. Nierstrasz, C. Pandu Rangan, B. Steffen, M. Sudan, D. Terzopoulos, D. Tygar, M. Y. Vardi, G. Weikum, S. Agarwal, N. Snavely, S. M. Seitz, and R. Szeliski, "Bundle Adjustment in the Large," in *Computer Vision – ECCV 2010*, K. Daniilidis, P. Maragos, and N. Paragios, Eds., vol. 6312, Berlin, Heidelberg: Springer Berlin

Heidelberg, 2010, pp. 29–42, ISBN: 978-3-642-15551-2 978-3-642-15552-9. DOI: 10.1007/978-3-642-15552-9_3.

[28] K. Kowalczyk and J. Rapinski, "Investigating the Error Sources in Reflectorless EDM," *Journal of Surveying Engineering*, vol. 140, no. 4, p. 06 014 002, Nov. 2014, ISSN: 0733-9453, 1943-5428. DOI: 10.1061/(ASCE)SU.1943-5428.0000130.

[29] K. E. Larsen, F. Lattke, S. Ott, and S. Winter, "Surveying and digital workflow in energy performance retrofit projects using prefabricated elements," *Automation in Construction*, vol. 20, no. 8, pp. 999–1011, Dec. 2011, ISSN: 09265805. DOI: 10.1016/j.autcon.2011.04.001.

[30] E. Lengyel, *Mathematics for 3D Game Programming and Computer Graphics*, 3rd ed. Boston, MA: Course Technology, Cengage Learning, 2012, ISBN: 978-1-4354-5886-4.

[31] J. L. Lerma and C. Muir, "Evaluating the 3D documentation of an early Christian upright stone with carvings from Scotland with multiples images," *Journal of Archaeological Science*, vol. 46, pp. 311–318, Jun. 2014, ISSN: 03054403. DOI: 10.1016/j.jas.2014.02.026.

[32] T. Luhmann, S. Robson, S. Kyle, and J. Boehm, *Close-Range Photogrammetry and 3D Imaging*, 2nd edition, ser. De Gruyter Textbook. Berlin: De Gruyter, 2014, ISBN: 978-3-11-030269-1.

[33] T. Luhmann, "Close range photogrammetry for industrial applications," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 65, no. 6, pp. 558–569, Nov. 2010, ISSN: 09242716. DOI: 10.1016/j.isprsjprs.2010.06.003.

[34] S. O. Mason, "Conceptual Model Of The Convergent Multistation Network Configuration Task," *The Photogrammetric Record*, vol. 15, no. 86, pp. 277–299, Oct. 1995, ISSN: 0031-868X, 1477-9730. DOI: 10.1111/0031-868X.00032.

[35] T. Mizoguchi, T. Kuma, Y. Kobayashi, and K. Shirai, "Manhattan-World Assumption for As-Built Modeling Industrial Plant," *Key Engineering Materials*, vol. 523–524, pp. 350–355, Nov. 2012, ISSN: 1662-9795. DOI: 10.4028/www.scientific.net/KEM.523-524.350.

[36] A. Murtiyoso, P. Grussenmeyer, M. Koehl, and T. Freville, "Acquisition and Processing Experiences of Close Range UAV Images for the 3D Modeling of Heritage Buildings," in *Digital Heritage. Progress in Cultural Heritage: Documentation, Preservation, and Protection*, M. Ioannides, E. Fink, A. Moropoulou, M. Hagedorn-Saupe, A. Fresa, G. Liestøl, V. Rajcic, and P. Grussenmeyer, Eds., vol. 10058, Cham: Springer International Publishing, 2016, pp. 420–431, ISBN: 978-3-319-48495-2 978-3-319-48496-9. DOI: 10.1007/978-3-319-48496-9_34.

[37] H. F. Nilsson and D. Grundberg, "Plane-Based Close Range Photogrammetric Reconstruction of Buildings," Ph.D. dissertation, Umea University, Mar. 2009.

[38] C. Ordóñez, J. Martínez, P. Arias, and J. Armesto, "Measuring building façades with a low-cost close-range photogrammetry system," *Automation in Construction*, vol. 19, no. 6, pp. 742–749, Oct. 2010, ISSN: 09265805. DOI: 10.1016/j.autcon.2010.03.002.

[39] S. Paiho, I. P. Seppä, and C. Jimenez, "An energetic analysis of a multifunctional façade system for energy efficient retrofitting of residential buildings in cold climates of Finland and Russia," *Sustainable Cities and Society*, vol. 15, pp. 75–85, Jul. 2015, ISSN: 22106707. DOI: 10.1016/j.scs.2014.12.005.

[40] P. Pihelo, T. Kalamees, and K. Kuusk, "nZEB Renovation with Prefabricated Modular Panels," *Energy Procedia*, vol. 132, pp. 1006–1011, Oct. 2017, ISSN: 18766102. DOI: 10.1016/j.egypro.2017.09.708.

[41] P. Rodriguez-Gonzalvez, D. Gonzalez-Aguilera, G. Lopez-Jimenez, and I. Picon-Cabrera, "Image-based modeling of built environment from an unmanned aerial system," *Automation in Construction*, vol. 48, pp. 44–52, Dec. 2014, ISSN: 09265805. DOI: 10.1016/j.autcon.2014.08.010.

[42] V. Sanchez and A. Zakhor, "Planar 3D modeling of building interiors from point cloud data," in *2012 19th IEEE International Conference on Image Processing*, Orlando, FL, USA: IEEE, Sep. 2012, pp. 1777–1780, ISBN: 978-1-4673-2533-2 978-1-4673-2534-9 978-1-4673-2532-5. DOI: 10.1109/ICIP.2012.6467225.

[43] P. Sapirstein, "Accurate measurement with photogrammetry at large sites," *Journal of Archaeological Science*, vol. 66, pp. 137–145, Feb. 2016, ISSN: 03054403. DOI: 10.1016/j.jas.2016.01.002.

[44] G. Schindler and F. Dellaert, "Atlanta world: An expectation maximization framework for simultaneous low-level edge grouping and camera calibration in complex man-made environments," in *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, vol. 1, Washington, DC, USA: IEEE, 2004, pp. 203–209, ISBN: 978-0-7695-2158-9. DOI: 10.1109/CVPR.2004.1315033.

[45] S. N. Sinha, D. Steedly, and R. Szeliski, "Piecewise planar stereo for image-based rendering," in *2009 IEEE 12th International Conference on Computer Vision*, Kyoto: IEEE, Sep. 2009, pp. 1881–1888, ISBN: 978-1-4244-4420-5. DOI: 10.1109/ICCV.2009.5459417.

[46] J. Straub, G. Rosman, O. Freifeld, J. J. Leonard, and J. W. Fisher, "A Mixture of Manhattan Frames: Beyond the Manhattan World," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, Columbus, OH, USA: IEEE, Jun. 2014, pp. 3770–3777, ISBN: 978-1-4799-5118-5. DOI: 10.1109/CVPR.2014.488.

[47] P. Tang, "Automatic reconstruction of as-built building information models from laser-scanned point clouds: A review of related techniques," *Automation in Construction*, vol. 19, no. 7, pp. 829–843, 2010, ISSN: 0926-5805. DOI: 10.1016/j.autcon.2010.06.007.

[48] I. Toschi, A. Capra, L. De Luca, J.-A. Beraldin, and L. Cournoyer, "On the evaluation of photogrammetric methods for dense 3D surface reconstruction in a metrological context," *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. II-5, pp. 371–378, May 2014, ISSN: 2194-9050. DOI: 10.5194/isprsannals-II-5-371-2014.

[49] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon, "Bundle Adjustment — A Modern Synthesis," in *Vision Algorithms: Theory and Practice*, G. Goos, J. Hartmanis, J. van Leeuwen, B. Triggs, A. Zisserman, and R. Szeliski, Eds., vol. 1883, Berlin, Heidelberg: Springer Berlin Heidelberg, 2000, pp. 298–372, ISBN: 978-3-540-67973-8 978-3-540-44480-0. DOI: 10.1007/3-540-44480-7_21.

[50] F. A. Van den Heuvel, "Line-photogrammetric mathematical model for the reconstruction of polyhedral objects," in *Electronic Imaging '99*, S. F. El-Hakim and A. Gruen, Eds., San Jose, CA, Dec. 1998, pp. 60–71. DOI: 10.1117/12.333798.

[51] R. Volk, J. Stengel, and F. Schultmann, "Building Information Modeling (BIM) for existing buildings — Literature review and future needs," *Automation in Construction*, vol. 38, pp. 109–127, Mar. 2014, ISSN: 09265805. DOI: 10.1016/j.autcon.2013.10.023.

[52] U. von Luxburg and B. Schölkopf, "Statistical learning theory: Models, concepts, and results," in *Inductive Logic*, ser. Handbook of the History of Logic, D. M. Gabbay, S. Hartmann, and J. Woods, Eds., vol. 10, North-Holland, 2011, pp. 651–706. DOI: 10.1016/B978-0-444-52936-7.50016-1.

[53] Q. Wang and M.-K. Kim, "Applications of 3D point cloud data in the construction industry: A fifteen-year review from 2004 to 2018," *Advanced Engineering Informatics*, vol. 39, pp. 306–319, Jan. 2019, ISSN: 14740346. DOI: 10.1016/j.aei.2019.02.007.

[54] T. Werner and A. Zisserman, "New Techniques for Automated Architectural Reconstruction from Photographs," in *Computer Vision — ECCV 2002*, G. Goos, J. Hartmanis, J. van Leeuwen, A. Heyden, G. Sparr, M. Nielsen, and P. Johansen, Eds., vol. 2351, Berlin, Heidelberg: Springer

Berlin Heidelberg, 2002, pp. 541–555, ISBN: 978-3-540-43744-4 978-3-540-47967-3. DOI: `10.1007/3-540-47967-8_36`.

[55] H. Wildenauer and A. Hanbury, “Robust camera self-calibration from monocular images of Manhattan worlds,” in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, Providence, RI: IEEE, Jun. 2012, pp. 2831–2838, ISBN: 978-1-4673-1228-8 978-1-4673-1226-4 978-1-4673-1227-1. DOI: `10.1109/CVPR.2012.6248008`.

Appendix A

Derivations

A.1 Plane-Plane Intersection

It is possible to find the intersection of two 3D planes A and B as follows. We assume that the planes A and B are not parallel. Let d_A and d_B the closest distance from planes A and B to the origin, respectively. Let $\vec{n}_A$ and $\vec{n}_B$ be their normal vectors. The direction of the line of their intersection $\vec{v}$ is:

$$\vec{v} = \vec{n}_A \times \vec{n}_B$$

This makes intuitive sense, as $\vec{v}$, being a direction orthogonal to each plane's normal vector, must be a direction upon which traveling will not change the distance (i.e. projection onto $\vec{n}_A$ or $\vec{n}_B$) to either plane.

To find a point $\vec{s}$ that lies on both planes (i.e. a point on their intersection line) as follows, first construct a line with a point on A (specifically $-\vec{n}_A d_A$), and a direction ($\vec{v} \times \vec{n}_A$). Because this line's direction is orthogonal to $\vec{n}_A$, and it has a point that lies on A, the whole line must lie in A. Because this line is orthogonal to $\vec{v}$, it must intersect the line $\vec{s} + t\vec{v}$ (the line intersection of A and B, which by definition lies in A). In particular, it must intersect the line at the same point it intersects B.

So, to find a valid $\vec{s}$, compute the point at which the line $-\vec{n}_A d_A + t(\vec{v} \times \vec{n}_A)$ intersects B by subbing in to the equation for the intersection point of a line and a plane (defined by Equation 4.6 in Section 4.11):

$$\vec{s} = -\vec{n}_A d_A \frac{-(\vec{n_B} \cdot -\vec{n_A} d_A + d_B)}{\vec{n}_B \cdot (\vec{v} \times \vec{n}_A)} (\vec{v} \times \vec{n}_A)$$

A.2 Closest Point to a 2D Line

For two points on a line $\vec{a}$ and $\vec{b}$, and a point $\vec{x}$, the vector from $\vec{x}$ to the closest point on the line is:

$$\vec{x}_{error} = (\vec{x} - \vec{a}) - (\vec{b} - \vec{a}) \frac{(\vec{b} - \vec{a}) \cdot (\vec{x} - \vec{a})}{(\vec{b} - \vec{a}) \cdot (\vec{b} - \vec{a})}$$

To see why this is true, note that this equation can be rewritten as:

$$\vec{x}_{error} = \vec{AX} - \vec{AB} \frac{cos(\angle BAX)||\vec{AB}|| \cdot ||\vec{AX}||}{||\vec{AB}||^2}$$

Which simplifies to:

$$\vec{x}_{error} = \vec{AX} - \frac{\vec{AB}}{||\vec{AB}||} cos(\angle BAX)||\vec{AX}||$$

Note that the second term here is exactly the vector projection of $\vec{AX}$ on to $\vec{AB}$, and so the whole equation represents the vector rejection of $\vec{AX}$ from $\vec{AB}$. As this is the vector representing only the perpendicular component of $\vec{AX}$ from $\vec{AB}$, it must be the shortest vector between $\vec{x}$ and the line through $\vec{a}$ and $\vec{b}$.