Videogrammetric Roof Surveying Using a Hybrid Structure from Motion Approach

A Dissertation
Presented to
The Academic Faculty

By

Habib Fathi

In Partial Fulfillment
of the Requirements for the Degree
Ph.D. in the
School of Civil and Environmental Engineering

Georgia Institute of Technology
December 2013

Videogrammetric Roof Surveying Using a Hybrid Structure from Motion Approach

Approved by:

Dr. Reginald DesRoches, Advisor
School of Civil & Environmental Engineering
Georgia Institute of Technology

Dr. Nelson Baker
School of Civil & Environmental Engineering
Georgia Institute of Technology

Dr. Jochen Teizer
School of Civil & Environmental Engineering
Georgia Institute of Technology

Dr. Ioannis Brilakis
Department of Engineering
University of Cambridge

Dr. Frank Dellaert
School of Interactive Computing
Georgia Institute of Technology

Date Approved: November 14, 2013

ACKNOWLEDGEMENTS

First and foremost, I would like to thank my advisors, committee members, industry partners, fellow students, and family members. Without their guidance and support, this dissertation would not have been possible.

I would like to express my deepest sense of gratitude to my advisor, Dr. Ioannis Brilakis, for seeing my potential and showing me the path to my academic aspirations. Many times when I felt I had reached a dead-end, Dr. Brilakis has analyzed the problem from a broader perspective and presented new directions to follow. I am also grateful for my co-advisors, Dr. Reginald DesRoches and Dr. Patricio Vela, whose knowledge and determination have allowed me to achieve more than I thought was possible. Their advice and suggestions have been invaluable.

I was also fortunate to have Dr. Nelson Baker, Dr. Frank Dellaert, and Dr. Jochen Teizer on my thesis committee. They are truly world-class researchers, and their feedback helped me gain new understanding about my research.

I would like to thank the technical and financial support that I have received from Metalforming Inc. and National Science Foundation during my PhD study. I specially thank Geoffrey Stone and William Wilkins whom their support and knowledge have helped me to gain deeper understanding of the problem from an industry perspective.

I am also deeply grateful to my parents for their hard work and sacrifice to give me the gift of education. They have encouraged me to believe I can succeed in whatever path I choose, and they nurtured my curiosity from an early age. Thanks for their enthusiastic support at every step of the way.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Summary

In a roofing project, acquiring the underlying as-built 3D geometry and visualizing the roof structure is needed in different phases of the project life-cycle. A 3D representation of the roof structure is required from the architectural standpoint and the dimensions of roof plane boundaries are required from the construction standpoint. Architectural drawings, building information model (BIM) files, or aerial photogrammetry are used to estimate the roofing area in the bidding process. However, as a roof structure is never built to the exact drawing dimensions, as-built dimensions of boundaries of every roof plane after installing the underlayment have to be obtained.

There are a number of surveying methods that can be used for this purpose: tape measuring, total station surveying, aerial photogrammetry, and laser scanning. Tape measuring is the most common practice despite the fact that it is not safe and exposes roofing workers to safety hazards. Roof surveying using a total station alleviates some of the inefficiencies, but requires trained surveyors and a stable and flat location to shoot from, which is not always available. The existing aerial photogrammetric methods eliminate most of the on-site labor requirements, but cannot fulfill the industry requirements for measurement accuracy. A laser scanner can well serve the purpose but the equipment and labor costs are prohibitive. In summary, obtaining measurements using the exiting roof surveying methods could be costly in terms of equipment, labor, and/or worker exposure to safety hazards.

Aiming to address this limitation and provide roofing practitioners with an alternative roof reconstruction and surveying method that is simple to use, automated, inexpensive, and safe, a close-range videogrammetric roof reconstruction framework is presented in this research. When using this method, a roofing contractor will simply collect stereo video streams of a target roof once the roof underlayment has been installed. The captured data is processed to generate a 3D wire-diagram for every roof plane. In this process, distinctive visual features of the scene (e.g., 2D points and lines) are first automatically detected and matched between stereo video frames. Matched features and the camera calibration information are used to compute an initial estimation of the 3D structure. Then, a hybrid bundle adjustment algorithm is used to refine the result and acquire the geometry that has the maximum likelihood. Afterwards, different planes of the roof are found in the refined 3D result using a half-plane detection algorithm. Finally, based on the information from points, lines, and planes, a 3D wire-diagram is generated for every plane which includes as-built dimensions of the roof.

The main contribution of this study is to create a general framework for videogrammetric roof reconstruction by identifying specific characteristics of roof scenes and then designing the necessary steps/processes. If an available algorithm in the literature fulfills the requirements at each step, it is utilized directly; otherwise, a modified or new algorithm is created such that the expectations are met. Specific contributions in this framework are the following, in the order of importance:

- Formulate a hybrid structure from motion pipeline which combines information from point and line features. It involves formulating a sparse bundle adjustment process by representing 3D coordinates of point and lines as well as their projections into the image space with the same number parameters. This allows modifying the well-known and efficient Sparse Bundle Adjustment (SBA) package such that it is applicable for the given study.

- Validate a close-range videogrammetric roof reconstruction framework that is capable of producing a measureable 3D wire-diagram for every plane in a roof structure.
- Determine a dynamic support region for a line segment such that its descriptor vector is invariant with respect to changes in the following features: scale, rotation, viewpoint, coordinates of the end-points.
- Design a particular stereo camera calibration procedure that eliminates/alleviates the problems that conventional camera calibration algorithm encounter in far-range applications. The goal is not to provide new mathematical relationships for estimating the necessary parameters; instead, conventional camera calibration algorithms are used in a specific procedure.
- Design a multi-step candidate plane generation algorithm that minimizes the possibility of missing salient planar regions and allows dealing with a broader range of scenes (poorly to well-textured).

CHAPTER 1

ANALYSIS

1.1. Roofing Industry

The roofing industry is part of what is known as the “building envelope” or “building enclosure” industry. A building envelope includes all the components that make up the shell or skin of the building and separate the exterior from the interior (e.g., walls, roofing, foundations, windows, and doors). Roofing contractors deal with the covering on the uppermost part of the building. They, in general, install roofing, siding, sheet metal, and roof drainage systems. The roofing industry can be divided into residential and commercial sectors which are quite different, and most companies specialize in one or the other. In each sector, three primary types of work are typically considered: new construction, maintenance/repair, and roof replacement.

Despite this categorization, the ecosystem of a roofing project is almost the same for different types of works. In the bidding process, a roofing contractor uses architectural drawings, a Building Information Model (BIM) file, or aerial images to estimate the roofing area. However, a roof structure is never built to the exact drawing dimensions; even in roof replacement projects, the dimensions are typically altered due to intrinsic restrictions of the construction process. This creates a discrepancy between the design and as-built dimensions. Accordingly, after the project is awarded and the underlayment is installed, there is a need to acquire as-built dimensions of the roof structure in order to cut covering material such that different pieces perfectly fit together and the waste is minimized. The National Roofing Contractors Association (NRCA Roofing Manual, 2012) recommends limiting the measurement errors to ±¾in. (≈1.9cm) and in some cases ±1in. (≈2.54cm) in order to be able to address practical constraints;

however, this limitation could change based on the specifications of a project. Once the as-built dimensions are acquired, the covering material (e.g., sheet metal, plastic, and photovoltaic products) is cut and overlaid. The finished project is then handed over to the client and the same process is repeated when a roof replacement is needed. The current practice is to replace a roof based on decisions on fixed intervals, regular inspection results, and reported maintenance issues (Coffelt & Hendrickson, 2010). A housing roof cycle lasts about 15 years, and that commercial roofs average 20 years.

As can be inferred, 3D visualization of a roof structure and collecting accurate enough as-built geometry is an essential activity in every roofing project. Performing this task using an efficient method/device has been a long-standing challenge in the roofing industry.

1.2. State of Practice

Several surveying technologies have evolved over the years aiming to provide efficient, precise, and simple ways to visualize and geometrically document as-built condition of a roof structure. Contractors use these methods for a variety of purposes such as estimating the area or volume of a building component (Izquierdo, et al., 2008), digital roof modeling (Hashiba, et al., 2003), and etc.

A roofing contractor needs to survey a roof structure several times over the course of its build. A number of surveying methods are used by practitioners for this purpose: tape measuring, total station surveying, aerial photogrammetry, and laser scanning. They can be categorized into two groups based on the necessity for physical contact. These methods are discussed below and their advantages and limitations are analyzed.

1.2.1. Tape measuring

Manual measurement with a tape measure is the most common form of roof surveying methods and belongs to the category of methods that require physical contact

(NRCA Roofing Manual, 2012). When measuring with a tape measure, a team of roofing employees climb onto the roof with a sketch including an outline of the roof perimeter and take measurements manually. This process is simple and needs minimal expertise; however, it is time-consuming, expensive in terms of labor costs, and not always accurate (Wood, 2012) (Fredericks, et al., 2005). As an example and based on on-site interviews, a 30,000 square foot "cut up" (a roof with many intersecting planes) commercial building may require a team of two men for approximately 6hrs to do tape measuring and 2.5hrs to transfer collected data in an appropriate format. The most important disadvantage of the tape measuring method is the necessity for physical contact that results in higher cost of operation and exposure of the measuring crew to fall hazards (Fredericks, et al., 2005). Falling is one of the most critical safety hazards, particularly on sloped roofs (Figure 1.1), and contributes to the very high number of occupational injuries and fall deaths which occur in the roofing industry (7% of private construction fatalities in 2009 (OSHA, 2009)). Non-contact-based methods, which will be discussed below, try to alleviate these problems by providing the opportunity to measure as-built dimensions of a roof structure from a distance. However, tape measuring is still the standard practice in the industry despite its inherent deficiencies.

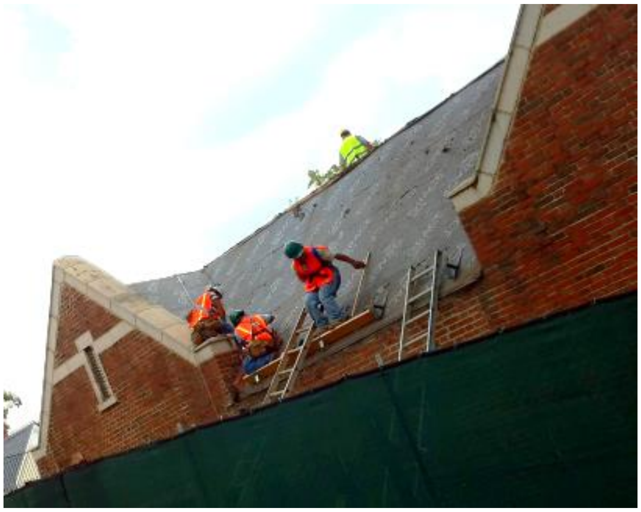


Figure 1.1: Worker exposure to safety hazards

1.2.2. Total station surveying

This roof surveying method provides accurate as-built measurements by using a total station which typically costs $3,000 to $8,000. Total station is an electrical/optical instrument that emits a single axis laser beam (Figure 1.2) and therefore, can only perform one measurement at a time. The process requires surveyors who are trained in surveying methods/techniques, hardware (survey instrument and computers), software (SDMS, DTM, etc.), data transfer, and metric conversion (Coaker, 2009). Moreover, having knowledge about different parts of a roof structure is necessary. Before taking measurements, a surveyor needs to prepare a sketch that outlines the roof perimeter. When collecting data, the surveyor locates a point on the ground to set the total station over. The point should be selected such that necessary surveying points (i.e., where roof planes come together, corners, and center points) are visible. This presents challenges in complex roofs with many intersecting planes where some important points are not visible from the ground. In this case, the surveyor would need to find a “stable” and somewhat “flat” location on the roof to place the instrument; the location should provide adequate stability which is required for taking precise measurements. The surveyor can then start recording X, Y, and Z coordinates of desired points and meanwhile marking them on the

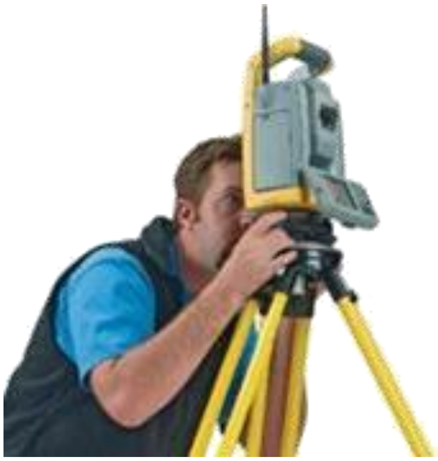


Figure 1.2: Total station surveying

sketch. Once the required data is recorded for all points, the surveyor can use a laptop at the jobsite or go back to the site office to transfer the collected data into an appropriate software program. Roof measurements are then extracted. In general, this surveying method is a safe practice because the surveyor stands on the ground or a fixed position on top of the roof, but it requires surveying expertise.

1.2.3. Aerial photogrammetry

In this process the address of the property is typed in an on-line request form and, using geocoding, a software program calculates longitude and latitude, enabling to extract the correct imagery from the available satellite/aerial images (in reality, the most accurate method used by leading roof measurement companies is the use of aerial photography and not the satellite images). CAD professionals then provide roof measurements using photogrammetric software programs. This technique is inexpensive, safe, and easy to use. It also does not require any on-site measurements and hence eliminates the on-site labor requirements. However, a case study that has been performed to evaluate the accuracy of aerial roof measurements on 1,291 roof structures indicates that measurement errors are expected to be in the range of ±4% of the actual length (EagleView Technologies, 2012). Another study shows that the acquired measurements can have errors up to 5% (Cory, 2009). This accuracy is not sufficient and reliable for applications in the construction phase of a roofing project life-cycle such as digital fabrication of sheet metal roof panels which requires accuracy within approximately ±2cm. Therefore, this method can only be used in the bidding process in order to estimate the roofing area. Another disadvantage of this method relates to the resolution of the available satellite/aerial imagery. In the US for example, commercial satellite images are limited to 18 in. per pixel (0.5 m per pixel) while aerial images are available at resolutions down to 4 to 6 in. per pixel for most of the populated areas in the US (EagleView Technologies, 2011). Moreover, the method is unusable if satellite images are not available for a specific property or if the satellite images do not include recent renovations that have changed the geometry of the roof.

1.2.4. Laser scanning

A laser scanner can provide a dense cloud of 3D points by measuring the distance of 10,000 to 100,000 points every second with mm level accuracy (Tang, et al., 2009). In general, the process is simple and does not expose the crew to safety hazards. The instrument is setup on a fixed location (sometimes it is required to put the instrument on top of the roof for proper visibility). A trained surveyor then collects the necessary data and performs post processing steps for extracting roof planes and perimeter of the roof. The main limitation of this method is the high hardware costs. A laser scanner may cost thousands of dollars (e.g., a Leica C10 laser scanner can be bought at $100,000 or be rented at $4,000 per job (Dai, et al., 2013)).

1.3. State of Research

Table 1.1 presents a summary of the state of practice. It illustrates advantages and limitations of the existing methods according to five factors, which are the features of an ideal roof surveying method from a roofing contractor perspective (presented values for each feature are based upon field evaluations that are performed under the NSF I-Corps project entitled “videogrammetric roof surveying system for digital fabrication of sheet metal roof panels”):

- **Cost:** The cost items that are involved in a roof surveying process are equipment, software/processing, and labor costs. Unlike software/processing and labor costs, the equipment costs are one-time expenses but the payment may be spread out over many surveying cases. Our field evaluation has shown that equipment costs of less than $5,000 are acceptable for contractors; on the other hand, a total of $100 per every 10,000 square feet of the roofing area is acceptable for total of software/processing and labor costs.

- **Accuracy:** According to roofing manuals, measurement errors for sensitive tasks (e.g., digital fabrication of roof panels) should not exceed ±1in. (~±2.5cm). Although for other tasks, this threshold can be up to 3in.
- **Simplicity:** In the roofing industry, a process is defined as simple if an average non-technical roofing employee can collect the necessary data. This data is going to be used directly or inputted into a software program.
- **Safety:** Data collection and its processing should not expose any of the data collectors to safety, and especially fall, hazards.
- **Efficiency:** A roof surveying method is considered to be efficient if the entire operation (i.e., data collection, processing, and post-processing) can be completed in less than 4hrs.

Table 1.1: A comparison among roof surveying methods (advantages and limitations)

	Tape Measuring	Total Station Surveying	Aerial Photogrammetry	Laser Scanning
Cost	✓		✓	
Accuracy	✓	✓		✓
Simplicity	✓		✓	
Safety		✓	✓	✓
Efficiency				

A comparison shows that aerial photogrammetry is the least expensive, simplest, and safest method; however, it cannot produce accurate enough measurements for most activities. The reason could be the limited resolution of commercially available satellite images (EagleView Technologies, 2011). Considering all of these issues and aiming to find a roof surveying solution that is not hindered by the limitations stated above, this research investigates the technical feasibility of using close-range machine vision-based methods.

The use of machine vision-based techniques for 3D reconstruction of built environments has been the subject of many research initiatives both in computer vision and civil infrastructure management applications (Pollefeys, et al., 2008) (Gallup, 2011) (Irschara, et al., 2012) (Podbreznik & Potocnik, 2010) (Brilakis, et al., 2011) (Jog, et al., 2011) (Golparvar-Fard, et al., 2013) (Rashidi, et al., 2013). Existing methods typically perceive the 3D shape of a structure by analyzing local motions of a camera. This process is commonly called Structure from Motion (SfM) and is the basic geometrical theory behind all of these methods. In general, there are three main SfM approaches: 1) feature point-based SfM; 2) line-based SfM; and 3) hybrid SfM which benefits from a combination of points and lines. A single camera, a stereo set of cameras, or a multi-camera system may be used to collect images/videos and then recover the 3D structure of the scene. The sensor system can also be calibrated or uncalibrated. Euclidean 3D reconstruction, however, necessitates using calibrated cameras. The following sub-sections will first discuss the camera calibration process and then analyze each category of SfM approaches in terms of their possible applicability for roof reconstruction and surveying. Hypothesizing/detecting and subsequently verifying planar surfaces in a roof structure are considered next.

1.3.1. Camera calibration

Camera calibration is the process of determining a set of camera parameters that describe the mapping between 3D world coordinates and 2D image coordinates. The parameters to be calibrated are categorized into intrinsic and extrinsic parameters. Intrinsic parameters represent internal geometry and optical characteristics of the lens while the camera position and orientation in the 3D world reference system are extrinsic parameters. The existing camera calibration methods are divided into two categories: a) explicit calibration (i.e., conventional approach) and b) self-calibration. Methods in the first category estimate the calibration parameters by establishing correspondences between reference points on an object with known 3D dimensions and their projection on the image. On the other hand, self-calibration automatically provides necessary parameters using the geometrical constraints in images, but is less accurate than the explicit methods (Furukawa & Ponce, 2009). Since the output accuracy is one of the main goals in this research, this section only focuses on the explicit approach.

The first step in calibrating a camera is to define a camera model. In computer vision, most practical cameras are represented by a pinhole camera model (Figure 1.3). In this model, each point in the world space $(X,Y,Z)^T$ is projected by a straight line into the image plane, through the camera center C. The intrinsic parameters in this model are focal length (f_x, f_y), principal point (u,v), and distortion coefficients (k_1,k_2,p_1,p_2,k_3). Also, the camera position (t) and orientation (R) in the 3D space are extrinsic parameters. If image points are represented by homogeneous vectors, a 3D point is projected on the image plane as

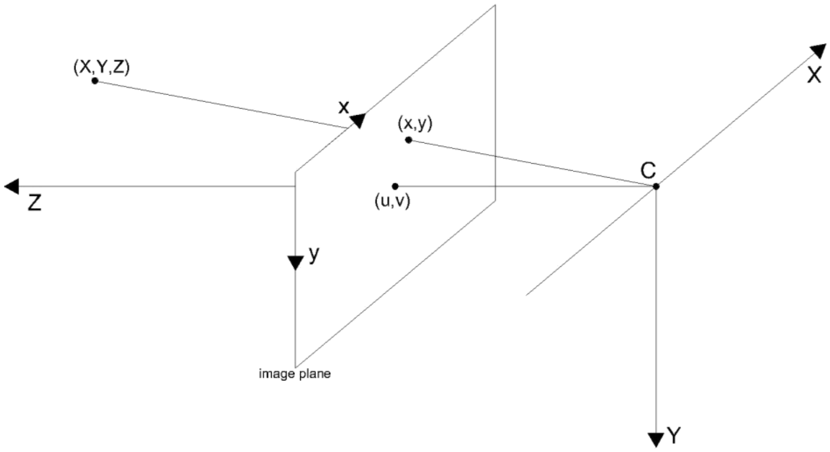


Figure 1.3: Pinhole camera model

$$\begin{pmatrix} x'' \\ y'' \\ w \end{pmatrix} = R \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} + t \tag{1-1}$$

$$x_d = x''/w \quad , \quad y_d = y''/w \quad , \quad r^2 = x_d^2 + y_d^2 \tag{1-2}$$

$$x' = x_d(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) + 2 p_1 x_d y_d + p_2 (r^2 + 2x_d^2) \tag{1-3}$$

$$y' = y_d(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) + p_1 (r^2 + 2 {y_d}^2) + 2 p_2 x_d y_d \tag{1-4}$$

$$x = f_x x' + u \quad , \quad y = f_y y' + v \tag{1-5}$$

For explicit camera calibration, a set of images is captured from different views of a checkerboard or an object with known dimensions. Then, correspondences between 3D coordinates of corner points and their 2D image projections are established. The abovementioned equations are finally used in a non-linear optimization problem to estimate the unknown camera parameters. This general process has been used in most of the existing camera calibration methods such as (Zhang, 2000), (Wang, et al., 2008), and (Furukawa & Ponce, 2009). Although these methods have been mainly proposed for a single camera, they can be expanded to the stereo camera calibration scenario. In this case, the relative position between two cameras (R_0, t_0) needs to be found beside the intrinsic and extrinsic parameters of each camera. The geometric relationship between the left (R_l, t_l) and right (R_r, t_r) cameras can be expressed as follows

$$R_0 = R_r R_l^{-1} \quad , \quad t_0 = t_r - R_r R_l^{-1} t_l \tag{1-6}$$

These conventional methods have been successfully used in close-range 3D reconstruction applications with spatial accuracies that rival laser scanning (Seitz, et al., 2006); however, the same level of accuracy has not been achieved in far-range applications even using cameras with multi megapixel resolution (Dai, et al., 2013).

Most of the existing image-based reconstruction methods in computer vision use a single camera for image acquisition. This imposes a constraint on the generated results because, by using a single camera, a scene can only be reconstructed up to an unknown scale factor (Pollefeys, et al., 2008). This limitation is of great importance in infrastructure applications that require spatial data collection in the Euclidean space. The use of a calibrated stereo camera setup eliminates this problem but at the cost of additional steps for camera calibration and more sensitivity of the results to the calibration parameters (Peng, 2011) (Xu, et al., 2012). Due to the fact that end-to-end dimensions of boundaries of roof planes need to be measured in the Euclidean space (practical constraints and safety concerns in a jobsite impede the possibility of taking a reference measurement for scaling up the results from a single camera), the use of a stereo camera setup is considered in this research and analyzed next.

In a stereo reconstruction problem, the accuracy of results could be very sensitive to the intrinsic and extrinsic stereo camera calibration parameters as well as the distance between the camera and the object of interest (House & Nickels, 2006) (Geiger, et al., 2011). This may be justified by the point that in such a problem, the estimated parameters are kept constant in all of the SfM process and errors can accumulate. (Dang, et al., 2009) have presented a thorough mathematical analysis for sensitivity of stereo 3D

Table 1.2: Sensitivity of 3D reconstruction to erroneous stereo camera calibration parameters (Dang et al., 2009)

Error Source	**Linear Sensitivity of 3D Reconstruction**	**Error Source**	**Linear Sensitivity of 3D Reconstruction**
yaw error $\Delta\Psi_L$	$\frac{\Delta Z}{\Delta\Psi_L} \approx -\frac{Z^2}{b}\left(1+\tilde{x}_L^2\right)$	baseline error Δb	$\frac{\Delta Z}{\Delta b} \approx -\frac{Z}{b}$
pitch error $\Delta\Phi_L$	$\frac{\Delta Z}{\Delta\Phi_L} \approx \frac{Z^2}{b}\tilde{x}_L\tilde{y}_L$	center offset ΔC_L	$\frac{\Delta Z}{\Delta C_L} \approx -\frac{Z^2}{bf}$
roll error $\Delta\Theta_L$	$\frac{\Delta Z}{\Delta\Theta_L} \approx \frac{Z^2}{b}\tilde{y}_L$	focal length error Δf_L	$\frac{\Delta Z}{\Delta f_L} \approx \frac{Z^2}{bf^2}\left(x_L - C_x\right)$

reconstruction to erroneous camera calibration parameters. The results of this study are summarized in Table 1.2, where Z is the depth of the point to be reconstructed; b is the baseline; f is the focal length in pixels; C_x is the x-coordinate of the camera center; x is the x-coordinate of the point in the image space; $(\tilde{x}, \tilde{y})$ are normalized coordinates of the point in the image space; and subscript L denotes the left camera in the stereo rig. From this table, it can be concluded that the sensitivity of the results is the highest for yaw, pitch, and roll; reconstruction errors also scale linearly with Δb and higher tolerances are acceptable in estimating b.

The existing camera calibration packages such as Camera Calibration Toolbox not only provide the best estimation for each parameter but also calculate the amount of uncertainties in the given estimation (in terms of a ± range). This range of uncertainty is another source of information that could be used to analyze the sensitivity of the process. An observation in (Peng, 2011) indicates that if the distance between the camera set and the calibration board is (more or less) kept constant, the range of the uncertainties decreases. On the other hand, if the distance keeps varying in a larger range, higher uncertainties in the estimated parameters could be seen. Another observation demonstrated that the uncertainties for distortion parameters could be as high as 300% which may not have a significant impact in close-range applications but certainly affects the accuracy of far-range 3D reconstruction. In general, the following reasons could be listed for such a behavior. First, the projection function including the distortion effect is a non-linear function; hence if the input data covers a broader range of distances, there is a higher chance to be trapped in local optima. Second, the cost function in the optimization process is the reprojection error which is more sensitive to the data related closer distances; hence, the estimated parameters could result in high spatial distance error for data from farther distances.

All these observations indicate a need for further research and experiments on stereo camera calibration with the aim of finding proper mathematical formulation or calibration procedure such that the abovementioned effects could be minimized.

1.3.2. Point-based SfM

The feature point-based SfM approach involves a strategy of concentrating on points in the scene whose matching point can be automatically found across multiple views. The output of this approach is a "cloud" of 3D points; in this cloud, each point represents 3D coordinates of an object point visible in two or more views (i.e., images or video frames) of the same scene. Therefore, the main challenge in this approach is to automatically find correct matching points across different views of a scene. The emergence of invariant local feature points and descriptors such as SIFT (Lowe, 2004) and SURF (Bay, et al., 2008) have addressed this problem. Robust feature point matching using these descriptors has led to several successful studies in close-range 3D reconstruction where the object distance to the camera is less than approximately a few meters e.g., (Hernandez & Schmitt, 2004) (Furukawa & Ponce, 2006) (Zhang, et al., 2010).

In the past years, a number of studies have also shown that this approach could be used for reconstruction of large scale environments. For example, (Goesele, et al., 2007) presented a multi-view stereo algorithm for 3D reconstruction of large scale environments from community photo collections. A comparison between the results of this algorithm and the data acquired with a time-of-flight laser scanning system demonstrated that 90% of the points in the generated point cloud are within 128mm of the laser scanned model of a 51m high building. (Pollefeys, et al., 2008) and (Frahm, et al., 2010) studied real-time 3D reconstruction of the urban environments, which mostly exhibit planar surfaces. (Pollefeys, et al., 2008) have reported the following error values for real-time 3D reconstruction of a Firestone store using a set of horizontal and upwardfacing video cameras: median error of 21.9mm, mean error of 47.9mm, and completeness of 66%; completeness measures how much of the building is reconstructed. In another study, (Sinha, et al., 2010) have proposed a hybrid approach that works based on feature point and vanishing point matches in images. Due to the use of vanishing points, this approach is particularly suited for man-made environments which typically consist of three main vanishing directions. They demonstrated that this approach is capable of generating better or same quality point clouds compared to BUNDLER, which is a standard pipeline for point-based SfM. (Gallup, 2011) recently demonstrated how to automatically create detailed 3D models of urban environments from street level imagery. The structure in urban scenes (e.g., planarity, orthogonality, verticality, and texture regularity) and a plane-sweep stereo method were used to achieve 3D reconstruction with greater efficiency in terms of running time and memory use. The main goal of this study, however, was to achieve high quality visualization rather than the Euclidean accuracy of the reconstruction.

In the construction community, (Golparvar-Fard, et al., 2009) used the point-based approach on uncalibrated daily progress photographs of construction sites for sparse 3D reconstruction of the jobsite and construction progress monitoring. The main limitation of this study is that it only provides 3D coordinates of feature points. Edge points, such as points on the perimeter of a roof, are deleted in the feature point detection process (Lowe, 2004) and hence they do not exist in the output. Dense point cloud generation algorithms such as (Furukawa & Ponce, 2010) could be used to overcome this limitation. The generation of a dense as-built point cloud for a construction site has been considered in (Golparvar-Fard, et al., 2013). The study presents an automated approach for progress monitoring of activities at a jobsite based on two sources of information: a dense point cloud and a BIM file. The dense as-built point cloud is generated from unordered daily construction photo collections. Although the metric accuracy of the reconstruction is not presented in those research efforts, a comparative study shows that errors in the order of ±6-8cm should be expected (Dai, et al., 2013). Additionally, when these methods are applied to reconstruct scenes from roof structures, necessary points for roof surveying (i.e., boundary points) may not be reconstructed. As an example for such a case, Figure 1.4 shows a dense point cloud that is generated for a roof model. To address this problem, one may consider identifying different roof planes from the 3D point cloud and then intersecting those planes to detect boundary lines; this could work for places that roof planes intersect; but not all boundary lines are located in those places.

Another important issue with using the point-based SfM approach in the construction industry is that in many of the built infrastructure scenes, most areas lack distinctive points due to the prevalence of poorly-textured surfaces (e.g., smooth surface of a concrete wall) which ultimately results in several holes (i.e., missed data) (Figure

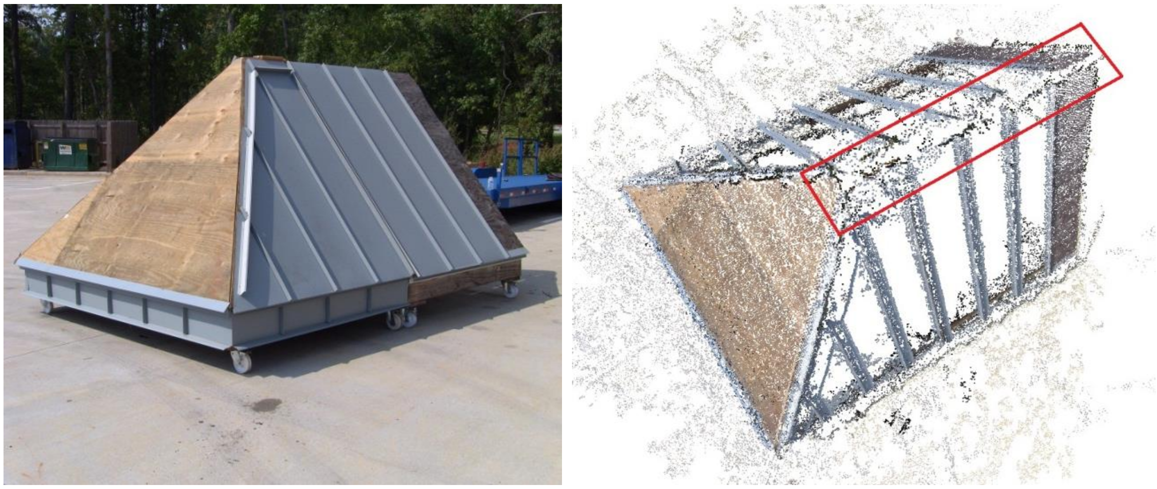


Figure 1.4: A dense 3D point cloud from a roof model. Notice the area marked by the red rectangle.

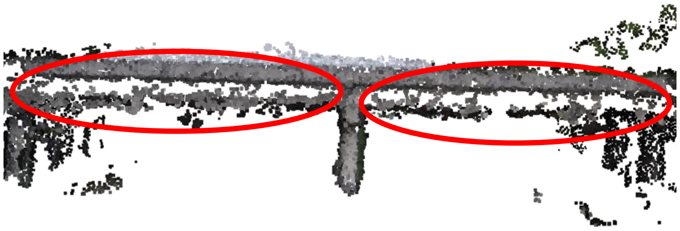


Figure 1.5: Point-based 3D reconstruction of a bridge

1.5) in the generated 3D point cloud and frequently causes instability that leads to failure of the process (Tomono, 2009).

1.3.3. Line-based SfM

The line-based SfM recovers the underlying 3D structure of a scene from line segments detected in multiple views of the scene. The output of this approach is a 3D line set. Similar to the feature point-based approach, the primary step here is to detect and match line features across views. The following paragraphs analyze the state-of-the-art algorithms for this purpose.

Many of the existing line segment detection algorithms use an edge detector followed by a Hough transform to extract all lines that contain a number of edge points exceeding a threshold (Fernandes & Oliveira, 2008) (Du, et al., 2010). Drawbacks of these methods are: a) textured regions that have a high edge density can cause many false detections; and b) setting thresholds. In another category of line detection methods, edge points are first detected and then chained into line segments. These methods are parameterless and usually give well-localized, accurate results. However, many of the small detected lines are false positive and there is also a need for several thresholds. The threshold question was thoroughly studied in (Desolneux, et al., 2002). Their method counts the number of aligned points (i.e., points with gradient direction approximately orthogonal to the line segment) and then finds the line segments as outliers in a non-structured, a contrario model. This method locates lines where alignments are present (no false negative) and hence has few false positives. But, it often misinterprets arrays of aligned line segments. (Akinlar & Topal, 2011) addressed this problem by using clean and contiguous chain of edge pixels produced by a new edge detection algorithm. The detector includes a line validation step due to the Helmholtz principle, which allows controlling the number of false detections. (Von Gioi, et al., 2010) also presented a linear-time algorithm that benefits from the advantages of previous methods. The key idea was to ignore gradient magnitudes and only use gradient orientations. This method requires no parameter tuning and gives accurate results. In general, these recent line detection algorithms have shown promising performance in terms of completeness, run-time efficiency, and the need for parameter tuning.

Once line feature are detected, they should be matched across different views of a scene. Compared to point and region matching, line matching is still a very challenging task due to several reasons: a) inaccuracy in the location of line endpoints; b) unavailability of strong disambiguating geometric constraints; and c) lack of rich textures in the local neighborhood of a line (Schmid & Zisserman, 1997). Because of these inherent difficulties, several approaches have been tried through the past years to achieve a robust line matching algorithm. (Schmid & Zisserman, 2000) used the epipolar geometry for line endpoints in short baseline matching, and one parameter family of plane homographies in wide baseline matching. The limitation of this method is the need for known geometrical relations between images in advance. Aiming to remove this limitation, (Bay, et al., 2005) proposed a method for line matching in color images, where an initial set of line correspondences are generated using color histogram; then, a topological filter is used to iteratively increase possible matches. This method heavily relies on color rather than the texture around the line and it may fail in the case where color is not distinctive. In another category of line matching methods, researchers have tried to construct a multi-dimensional descriptor vector for each line segment and use the vector difference for locating good matches. For example, (Wang, et al., 2009) clustered line segments into local groups according to their spatial proximity and assigned a descriptor to each group. The similarity measure of group pairs is based on the location of line end-points, orientation of the lines, and their intersection angle; this allows the method to be affine invariant. The methods, however, cannot handle general camera motions and relies on the availability of several lines in a close proximity. In another study, (Wang, et al., 2009) presented a SIFT-like descriptor called mean-standard deviation line descriptor (MSLD) which does not need any prior knowledge for line matching. It is purely image content-based and applicable to general scenes. However, this method provides poor matching results for line segments that are located in object boundaries, when the background of the object changes in two views. The reason is the following: the descriptor vector is built for a rectangular pixel support region around a line segment and therefore for a specific line segment at object boundaries, half of the information may completely change in two different views (Figure 1.6). There also exist some other studies that use line-point invariants (Fan, et al., 2012) or intersection context of coplanar line pairs (Kim & Lee, 2012) for robust line matching; but these methods heavily rely on the existence of some predefined structures which certainly limits their fields of application.

The line-based SfM is mostly used for man-made scenes like office interiors or urban cityscapes, where not enough point features can be reliably detected, while an abundant number of lines are visible (Chandraker, et al., 2009). (Werner & Zisserman, 2002) presented a method for automated architectural reconstruction across image triplets by classifying lines according to principal orthogonal directions. (Bartoli & Sturm, 2005) addressed the problem of algebraic representation of lines for camera motion estimation

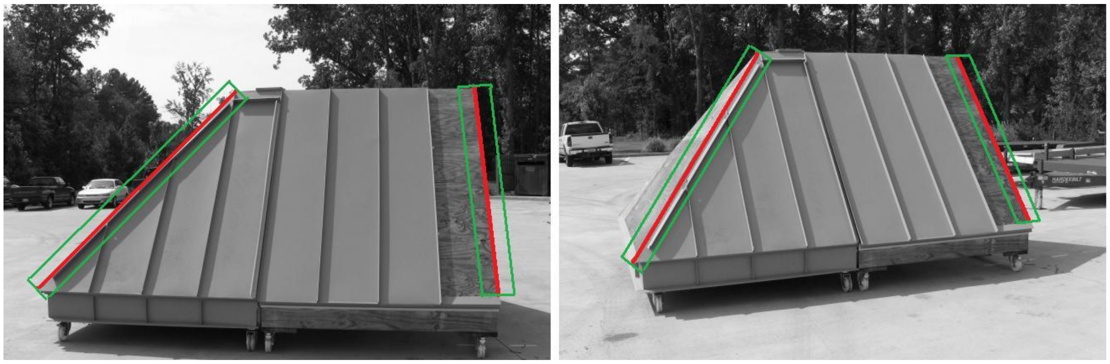


Figure 1.6: Rectangular pixel support region around line segments used in Wang et al. (2009). Notice completely different pixel information available in one side of the lines at object boundaries.

and 3D structure reconstruction from line correspondences across multiple views. They proposed orthogonal representation, which allows non-linear optimization of 3D lines using the minimum four parameters with an unconstrained optimization engine. This potentially solves the previous over-parameterizations of 3D lines that induce gauge freedoms and/or internal consistency constraints. (Schindler, et al., 2006) employed the knowledge of urban scene structure (i.e., three main directions for lines) for line-based SfM from multiple widely separated views. They proposed a new approach for 3D line representation which could ultimately allow reducing the number of optimization parameters for each line segment. More recently, (Chandraker, et al., 2009) proposed a hypothesize-and-test framework to estimate the motion of a stereo rig from line segments in real-time. This method avoids computationally extensive optimizations in order to increase speed and hence is only applicable in finding the approximate location of the camera set in a complex indoor environment.

This approach (i.e., line-based SfM) has been used in most of the existing aerial image-based 3D roof reconstruction methods (Moons, et al., 1998) (Baillard & Zisserman, 2000) (Scholze, et al., 2002) (Suveg & Vosselman, 2004) (Rau, 2012) for two reasons: 1) line features can be localized more accurately because they have more image support (i.e., several pixels construct a line) than point features; and 2) the main goal in 3D reconstruction of a roof is to extract 3D coordinates of plane boundaries which are typically straight lines. These methods generate nice visualizations; however, they can only be used for estimation purposes because errors in the order of several centimeters (e.g., 0.01% of the distance between the camera and the structure (Cui, et al., 2012)) are expected.

1.3.4. Hybrid SfM

A combination of point and line features is being used in this approach to broaden the use cases of image-based 3D reconstruction. Robust point matches are typically used to boost line matching accuracy. In practical settings, it has been shown that leveraging image lines in addition to points can lead to improved performance (Christy & Horaud, 1999). It is also well-known that constraints imposed by line correspondences on camera pose estimation are weaker than those provided by points (Hartley, 1997). Given this information, it seems that a visual odometry or SfM algorithm that incorporates any combination of point and line features is capable of generating better solution for the problem at hand.

Hybrid SfM has been mostly used for indoor mapping and Simultaneous Localization and Mapping (SLAM) problems and has shown significant performance improvement and robustness (Ramalingam, et al., 2011) (Pradeep & Lim, 2012). Point and line features are used in (Ramalingam, et al., 2011) to develop a general technique for the problem of geo-localization (i.e., camera pose in the world coordinate system). They reported significant cumulative error reduction in motion estimation. This method is applicable for any combination of features including 3 points, 2 points and 1 line, 1 point and 2 lines, and 3 lines. In a more recent study, (Pradeep & Lim, 2012) used hybrid SfM to generate a minimal solver for performing online visual odometry using a stereo camera setup. Independent of the feature type, this method computes the 6DoF motion from minimal sets over the available data. Using the constraints implied by two trifocal tensors, it builds a system of polynomial equations and then solves it using a quaternion-based direct solution approach. It is shown that this method generates more accurate and robust estimations.

Despite the improved performance because of using hybrid SfM in the existing studies, none of those methods can be used for large-scale 3D reconstruction. The reason is that the presented mathematical relationships are only applicable for two consecutive pairs of images and hence cannot be used to formulate the bundle adjustment step in a large 3D reconstruction problem.

In general, it may be concluded that hybrid SfM could facilitate the 3D reconstruction pipeline by benefiting from well-localized line features while retaining the well-known advantages of point features. Despite this significant potential, no studies have been performed to explore and quantify the amount of performance improvement or loss that can be achieved in 3D reconstruction of large-scale built environments if such an approach is used. Moreover, the necessary mathematical formulations have not been presented in the literature.

1.3.5. Hypothesizing and verifying planar surfaces

It can be argued that the most prevalent geometric entity in built environments is a planar surface. The planarity assumption has been therefore widely used in the literature to provide a more realistic representation of the real-world scenes. Existing methods use a 3D point cloud, a 3D line set, or a combination of 3D points and lines to identify planar regions.

(Bartoli, 2007) investigated the automatic modeling of planar scenes using a 3D point cloud. Random sampling was utilized to generate multiple plane hypotheses, select the most likely planes with respect to actual images, and finally segment the point cloud into multi-coplanar groups. The method outperforms existing algorithms, that are only based on geometric criterion or a disjoint data segmentation scheme, in terms of the total number of detected planes. However, the number of false positive cases is higher and the random sampling nature of the process demands higher computational power (Sinha, et al., 2009). Moreover, planar surfaces in non-textured regions can be easily missed due to the lack of reconstructed 3D points. In order to overcome the limitations of multi-view stereo algorithms that require textured surfaces and therefore work poorly for many architectural scenes, (Furukawa, et al., 2009) used a specific Manhattan-world model where all planes must be orthogonal. The method retains only high-confidence points in textured areas and uses the normal vectors to extract three dominant orthogonal directions for the scene. Although the orthogonality assumption holds true in reconstructing building façades and indoor scenes, it is not satisfied in general scenes from the built environment or roof structures.

A piecewise planar model was also presented in (Sinha, et al., 2009) that can recover a fairly exhaustive set of dominant scene planes based on robust plane-fitting of 3D points and lines while utilizing strong vanishing point cues to infer their orientations. The method uses a Markov Random Field formulation to generate piecewise planar depth maps. This necessitates some initial assumptions regarding the spatial likelihood distribution of each feature which reduces the generality of the method. On the other hand, the use of vanishing directions can limit the use-cases to scenes that contain well-defined vanishing points (typically urban scenes that have three orthogonal vanishing directions). The combined use of 3D points and lines for identifying planar regions in a scene was also studied in (Wang, et al., 2013). The method that is formulated based on a family of half-planes rotating around a single 3D line and inter-image homographies. Their main assumption, however, is that in the local neighborhood of every line segment, one can find a number of feature points that are coplanar with the given line. As can be inferred, the need for existence of 3D feature points in the neighborhood of each line imposes an important limitation on this method.

Despite the success of piecewise planar stereo methods in detecting planar regions, they typically suffer from one or both of the following issues: a) reliance on the availability of enough point features in the scene to generate plane hypotheses, and therefore high probability of failure in poorly textured scenes; and b) computationally expensive process for generating and verifying plane hypotheses. This indicates a need for further research on computationally efficient piecewise planar methods that use global scene information for hypothesizing plane equations while minimal assumptions are made. The importance of such a method is highlighted in the case of using video streams which requires processing of a significant number of frames in a reasonable amount of time.

1.3.6. Machine vision-based methods for 3D reconstruction of roof structures

Although there are several methods in the literature that are able to reconstruct complex buildings from images with minimal user intervention, automated building reconstruction is still a challenging task. 3D reconstruction of roof structures is an important part of this category. A number of studies have been performed to automate 3D reconstruction of roofs in urban areas (Moons, et al., 1998) (Baillard & Zisserman, 2000) (Scholze, et al., 2002). These methods only use aerial images as the input data and model a roof as a structured ensemble of planar polygonal faces. Moreover, they assume a reliable set of 3D line segments have been already derived from aerial images and are available as the input data, without any explanation for potential methods that are needed to provide such data. Some of these methods with some additional manual inputs have been used in commercial aerial roof measurement products offered by companies like EagleView or SkyTech Imaging. However, they can only be used for estimation purposes because of the limited resolution of aerial images (errors up to ±15cm are expected). Following paragraphs provide more details about two of these methods that have shown better performance.

(Baillard & Zisserman, 2000) presented a method for 3D reconstruction of roofs using multiple aerial images, with a novel approach for computing planar facets from a set of 3D lines. The key idea was to use both geometric and photometric constraints from all images (i.e., 3D planes were found by using both 3D lines and their image neighborhoods over multiple views). This was achieved through a plane-sweep strategy: first, a set of planar facets constrained by 3D lines were hypothesized in the 3D space; then, possible plane hypotheses were identified by checking the similarity over multiple images. The method defines a one-parameter family of planes $\pi(\theta)$ for each 3D line (Figure 1.7). This implicitly results in defining a one-parameter family of homographies $H(\theta)$ between any pair of images because each plane defines a planar homography between two images. It requires minimal image information since a plane is generated from only a line correspondence and its image neighborhood; in particular, two lines are not required to instantiate a plane. However, the method incurs a considerable computational cost since it is necessary to search a continuous range (0-180 degrees) in order to find the correct angle of rotation for each hypothesized plane. Also, the method only focuses on the visualization aspect of the problem and does not provide 3D measurements. (Cui, et al., 2012) has shown that if such a method is used for Euclidean 3D reconstruciton, measurement errors in the order of 0.01% of the distnace between the camera and the structure should be expected; given the amount of this distance in aerial imagery, the error could be as high as tens of centimeters.

In another study, (Scholze, et al., 2002) proposed a model-based algorithm for 3D reconstruction of complex polyhedral building roofs from aerial images using semantic labeling. The modeling process consists of two steps. a) 3D line segments were grouped into planes and further into faces using a Bayesian analysis. In this step, a roof was geometrically modeled as an ensemble of planar polygonal faces (i.e., patches). b) The

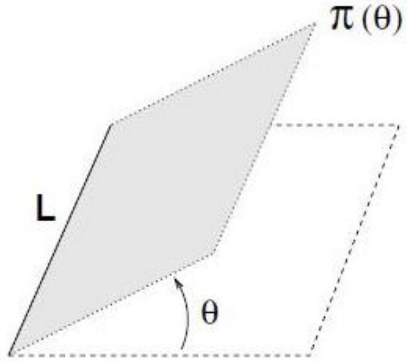


Figure 1.7: One parameter family of half-planes containing a 3D line L used in Baillard and Zisserman (2000)

preliminary geometric models were subject to a semantic interpretation. Five different semantic labels for patch segments were identified (ridge, gutter, gable, convex, and concave). Geometric measurements were then used to assign the semantic labels to the segments in a patch. One of the important shortcomings of this algorithm is the need for test datasets to learn the statistics of geometric measurements.

A number of studies have been also performed to alleviate the measurement problem by fusing aerial photogrammetry with airborne laser scanning data (Jaw & Cheng, 2008) (Sampath & Shan, 2010) (Cheng, et al., 2011). Although these methods solve all the aforementioned problems, the cost of airborne laser scanning is prohibitive.

1.4. Problem Statement and Research Objectives

3D modeling/visualization and extracting the underlying as-built geometry using an efficient method/device has been a long-standing challenge in the construction industry. Contractors require this information for a variety of purposes such as estimating the area or volume of a building component, controlling the quality of construction, and managing the operations and maintenance. In a roofing project, dimensions of a roof structure should be measured in different phases of the project life-cycle. In the construction phase and after installing the underlayment, end-to-end dimensions of boundaries of every roof plane have to be obtained with a certain level of accuracy that could change based on the project specifications (e.g., the measurement error should not exceed ±2cm for digital fabrication of sheet metal roof panels).

There are a number of methods that can be used for this purpose: tape measuring, total station surveying, and aerial photogrammetry. Tape measuring, as the most common practice, requires teams of men carrying tape measures and climbing all over the target roof. This is one of the most critical safety hazards, particularly on sloped roofs, and contributes to the very high number of occupational injuries and fall deaths which occur in the roofing industry. It is also laborious and not always accurate. Roof surveying using a total station requires trained surveyors and a stable location to shoot from, which is not always available. Aerial photogrammetry needs significant amount of manual post-processing work and can become labor intensive for complex roof structures. It can also be inaccurate to a degree of more than 5%. In summary, obtaining these measurements using the exiting methods could be costly in terms of equipment, labor, and/or worker exposure to safety hazards.

A safe, inexpensive, automatic, and accurate enough roof surveying method eliminates all the above-mentioned problems and brings the roofing trade up to the modern standards of CAD/CAM used in many other industries. The existing general purpose automatic or semi-automatic machine vision-based approaches theoretically have all the desired factors except acceptable level of accuracy. 3D reconstruction algorithms for building roof reconstruction from aerial images have the same deficiency. However, a close-range machine vision-based algorithm specialized for roof surveying can significantly reduce the amount of error if the understanding that roofs are composed of intersecting planes is incorporated in the process.

Close-range machine vision-based algorithms are typically categorized into three groups based on the type of visual feature that is used: points, lines, and a combination of points and lines. Feature point-based algorithms discard most of edge points (which are needed to acquire roof dimensions) because of the difficulties that exist in their matching process; hence, boundary points of roof planes may not be reconstructed with a high probability. Identifying different roof planes from the generated 3D point cloud and then intersecting those planes to detect boundary lines may alleviate this problem in some cases; but it is not a general solution because not all boundary lines are located in the intersection areas. Line-based algorithms can theoretically address this problem because roof plane boundaries are usually straight lines. However, this category of algorithms suffers from three issues: 1) line matching has been less successful than point matching because of its inherent difficulties and mismatches have resulted in incomplete and error-prone 3D reconstruction results; 2) line correspondences impose much weaker constraints than those provided by feature points on SfM steps such as camera motion estimation (Hartley, 1997); and 3) degeneracy for lines is far more severe than for points (Hartley & Zisserman, 2004). These issues could frequently cause instability that leads to failure of the process. The third category could solve all the previously mentioned problems by incorporating any combination of point and line features as available in the image so that the most accurate solution can be achieved. It allows using well-established mathematical equations that exist for point features while taking advantage of well-localized line features. Despite this advantage, existing methods in this category have mainly focused on solving the visual odometry problem, which is the process of determining the position and orientation of a camera in an environment, and have shown performance improvement in that area. The possibility of using the concepts of this category for 3D reconstruction of the built environment or roof structures and the subsequent performance improvement or loss have not been fully explored yet and still further studies are needed. An issue that arises with applying this category of SfM to large-scale 3D reconstruction problems is the need for proper mathematical formulation in order to perform the necessary nonlinear global optimization called bundle adjustment. (Lourakis & Argyros, 2009) have already presented a package that is being used as an efficient tool; but, the current version is only applicable to point-based monocular reconstruction. Therefore, new mathematical formulations are needed to be incorporated in this package such that it is capable of handling hybrid stereo reconstruction problems.

The primary objective of this research is to address the problem stated above by investigating the technical feasibility of designing a close-range video-based roof reconstruction framework, such that it achieves a measureable 3D wire-diagram for every roof plane considering the following metrics: completeness of the wire-diagram, Euclidean accuracy of the end-to-end dimensions of the edges, and the number of manual data that is needed to fix a missing edge.

1.5. Research Questions

To effectively address the defined research objective, several questions need to be answered. Below is a summary of these questions.

- What is the appropriate hardware system for a machine vision-based roof surveying algorithm?
- How can we use the known information about sensitivity of Euclidean accuracy of 3D points with respect to calibration parameters and design a camera calibration procedure that is capable of providing higher Euclidean accuracies?
- Which type of visual feature(s) should be detected and used through the process? Is it necessary to use a combination of different feature forms?
- If a combination of visual features is needed, what is the appropriate mathematical formulation?
- Is there a need for manual inputs in some steps of the process?
- How to deal with scenarios that some roof planes or plane boundaries are not detected or reconstructed?

CHAPTER 2

SYNTHESIS

2.1. Overview of the Framework

A close-range video-based framework for roof surveying is proposed in Figure 2.1. In this figure, rectangular boxes represent an algorithm/function and ellipses are inputs/outputs. The processes that are shown by solid lines have been well-addressed in the literature and hence there is no need to present detailed information about them in this section; while dashed lines indicate a modified or new algorithm.

When using this framework, a roofing contractor collects stereo video streams of a target roof plane once the roof underlayment is installed. The sensor system is a

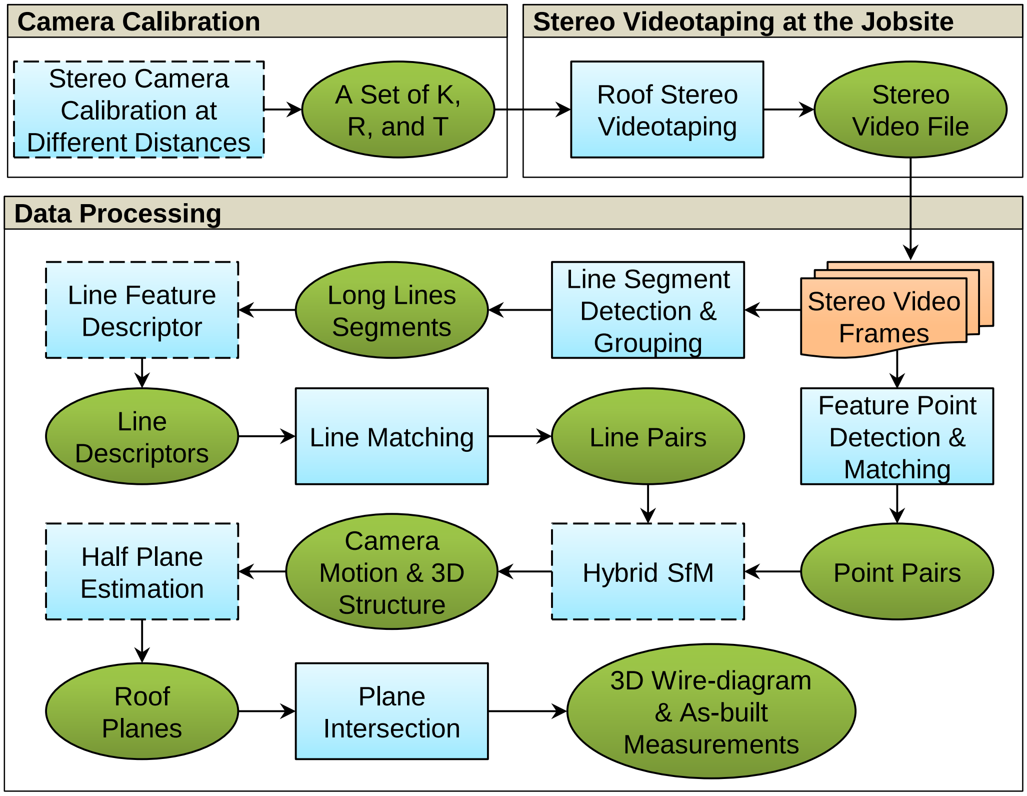


Figure 2.1: Overview of the proposed videogrammetric roof surveying framework

calibrated stereo camera set that has been attached to an extendible pole; the pole allows providing the necessary visibility for videotaping different roof planes. The captured video data is processed at the jobsite using a Tablet PC or laptop. Once a 3D wire-diagram of the target roof plane is generated, as-built dimensions are automatically extracted and saved in a digital file. Following steps summarize mechanics of the framework:

a. Calibrate a stereo camera set for several predefined *D* values (*D* is the distance between the calibration pattern and camera set). This results in a set of intrinsic and extrinsic camera parameters for each *D* value.

b. Detect distinctive feature points at each video frame, construct a descriptor vector for each detected point, and match them between left and right views of a stereo pair of video frames.

c. Build scale-space representation of each video frame, detect line features at local extrema of the scale-space, and group detected features to find long line segments (vanishing direction, scale of the detected segments, Euclidean distance between the end point of one segment and the start point of another segment, and slope of line segments are some of factors that are used in the line segment grouping process).

d. Construct two descriptor vectors for each line segment (a separate descriptor for each side of the line in order to address the problem of background change at object boundaries).

e. Match long line segments between left and right views of a stereo pair of video frames.

f. Track/match the detected visual features (i.e., feature points and lines) between consecutive stereo video frames and construct a feature connectivity graph.

g. Use a hybrid bundle adjustment algorithm, every time that a new stereo pair of video frames is added. The camera motion between the last pair of stereo video frames and previous one is initialized as zero.
h. Repeat the feature detection, matching, and hybrid bundle adjustment processes until all the stereo video frames are processed. The output is the 3D structure of the scene and motion of the camera set in the environment.
i. Apply a piecewise planar stereo strategy to estimate half-planes using the information from points and lines in order to find roof planes.
j. Find undetected roof plane boundaries by intersecting roof planes.
k. Construct a 3D wire-diagram for every roof plane which includes required as-built measurements for roof surveying.

2.2. Solution Constraints, Assumptions, and Research Hypothesis

The proposed roof surveying framework is subject to several constraints and limitations. In most cases, this is the consequence of assumptions that have been made in different steps of the framework. The nature of the surveying activity and characteristics of scenes from roof structures also contribute to these constraints.

Following control variables and delimitations are considered. A set of high-resolution video cameras capable of streaming raw video data are used along with fixed focal length lenses that have the minimal distortion (e.g., <0.05%); this is necessary to avoid information loss during image compression, change of focal length, and ability to detect straight lines in frame sets. Once the sensor system is setup, the following parameters should not change while collecting the necessary data: video resolution, focal length, and relative position of the two cameras. Videotaping should be relatively smooth, although the framework accounts for temporary jerking and corrects it. The weather and illumination conditions are also assumed to be those typically used is surveying (ranging from cloudy to bright sunlight but not rainy). Extreme cases such as very dark or shiny environments, direct sunlight towards the camera set, etc. may result in the failure of the framework.

The application of the framework is limited to slopped roofs with intersecting planes. The existence of well-defined and almost straight edges is an important factor for the success of this solution. If such edges do not exist and also those edges cannot be found by plane intersection, or if there are curved edges, the solution may fail. The height of the building should not be more than the height that is accessible using an extendible pole; otherwise, the required visibility cannot be met. A solution could be to use a crane or lift to provide the necessary elevation and visibility; however, these equipment may not be available in every roofing jobsite or the rental/purchase cost may be detrimental. Moreover, the proposed solution is based upon the assumption that a roof structure only consists of planar surfaces; so, it cannot be applied to roofs with curvy surfaces. Finally, only roof planes that are visible in the video frames could be reconstructed and the framework is not supposed to use probability theorems to inference the invisible sections or missing segments.

Consequently, the overall research hypothesis that is tested in this paper is: “if the proposed framework is used with the considerations explained above, a complete 3D wire-diagram including end-to-end dimensions of the edges can be generated for every roof plane.”

2.3. How to Select Appropriate Optics

In any computer vision application, the optics (i.e., camera and lens) should be selected such that the predefined goals can be achieved. The combination of the camera and lens is an important factor that affects the field of view, area that is covered by one pixel, resolution, and working distance. If these issues are not considered while selecting the optics, one should not expect desired outcomes no matter how robust the algorithms are. This section provides recommendations in order to select the most appropriate optics for the problem at hand.

Camera sensor resolution and lens focal length are the two unknown parameters that need to be determined. On the other hand, in a real life scenario, the range of values for the following parameters is expected to be known. 1) Field of view which is the width and height of the scene as viewed by the lens. Considering the previously explained challenges for line detection and matching, the minimum field of view for roof surveying is suggested to be half of the longest dimension in the roof structure. This should provide the texture information that is required for building a descriptor vector and also avoid detecting the target line in multiple segments. 2) Area that is covered by one pixel. Although most of the existing feature detection algorithms are capable of detecting the feature coordinates with sub-pixel accuracy (Lowe, 2004) (Bay, et al., 2008) (Von Gioi, et al., 2010), it is safe to assume that these coordinates could have errors up to one pixel. The covered area by this pixel can be selected based on the level of measurement accuracy that is needed (e.g., 2cm). 3) Working distance which is the distance from the lens to the roof structure.

Knowing the field of view and area that is covered by one pixel, the minimum resolution required for the sensor can be calculated. The camera sensor resolution translates 1 to 1 with the image resolution which is defined as the number of pixels in the image and is typically presented in two dimensions (e.g., 720×1280). Each dimension can be considered separately but it is often the case to reduce it to one dimension for simplicity.

$$\textit{sensor (image) resolution} = 2 \times (\textit{field of view / area covered by one pixel}) \qquad (2\text{-}1)$$

For example, if the field of view is 400cm and the area that is covered by one pixel is selected to be 1cm, the required image resolution is *2×(400/1)=800 pixels*. However, if we consider the possible rotations of the camera, the diagonal value needs to be used. Using the Pythagorean theorem the diagonal is about 1130. A sensor resolution of 1280×1024 could be used for this scenario. It needs to be mentions that this equation can be modified to solve for any of the other variables as long as there is only one unknown parameter.

Once the camera is selected, the sensor size can be used to calculate the focal length of the lens. In the given problem, it is typically the case that the working distance is flexible. This allows using a range of working distance options to get a focal length range. Once the focal length is selected, the thresholds for working distance can be calculated.

$$\textit{focal length} \times \textit{field of view} = \textit{sensor size} \times \textit{working distance} \qquad (2\text{-}2)$$

As a general rule, shorter focal length for a lens would result in wider field of view. This can introduce radial distortion, where an image looks curved and bulged out in the center. It has been shown that the radial and tangential lens distortion model, which is used in this study, can reasonably be used to compensate for the distortions if the field of view is approximately less than 90 degrees (Ricolfe-Viala & Sanchez-Salmeron, 2010). Therefore, such considerations should be made while selecting the appropriate optics.

$$\textit{horizontal field of view} = 2 \times \textit{arctan(width / }2\times\textit{focal length)} \qquad (2\text{-}3)$$

$$\textit{vertical field of view} = 2 \times \textit{arctan(height / }2\times\textit{focal length)} \qquad (2\text{-}4)$$

In addition to the camera optics, the synchronization of two cameras in a stereo setup should be considered. There are several methods, ranging from hardware to software solutions, that could be used to synchronize the shutter and exposure of the two cameras such that they take pictures exactly at the same time. An overview of these methods could be found in (Persson, 2009). However, these methods do not always provide 100% performance and therefore the issue of using slightly unsynchronized cameras and its impact on the 3D reconstruction results could be raised. This issue has been studied in (Fujiyoshi, et al., 2003) (Svedman, et al., 2005) (Svedman, 2005) (Wolf, 2006) (Bazargani, et al., 2012). It has been shown in (Svedman, 2005) that such errors in synchronization could results in errors up to 0.6% of the Z coordinate of a 3D point.

Since the goal of this research is not to be involved in hardware design, the research takes advantage of the existing high-performance digital cameras that are designed for computer vision applications and allow a user to ask for a frame at any given time using their API. However, this does not limit the application of the proposed framework because theoretically any two cameras could be synchronized in order to build a stereo camera system.

2.4. Multi-Step Stereo Camera Calibration for Improved Euclidean Accuracy

The accuracy of results in far-range stereo image-based 3D reconstruction is very sensitive to the intrinsic and extrinsic camera parameters determined during camera calibration. The existing camera calibration algorithms could induce a significant amount of error due to poor estimation accuracies and wide range of uncertainties in camera parameters, when they are used for far-range scenarios such as mapping civil infrastructure. This may lead to unusable results, and even failure of the whole reconstruction process.

Inspired by the results of previous studies ((House & Nickels, 2006) (Strecha, et al., 2008) (Peng, 2011) (Xu, et al., 2012)) and considering a constant size for the calibration board as the control variable, this research hypothesizes that the following procedure for a multi-step stereo camera calibration can enhance the final Euclidean accuracy of 3D points. If the distance between the sensor system and the calibration board is denoted by D, a conventional explicit stereo camera calibration procedure is repeated i times ($i = 1,...n$) for different D values. At each repetition, a different value is selected for D (e.g., $D_i = 10m$) and while it is kept constant, a set of stereo video streams are collected (Figure 2.2). During the video recording process, the camera system and the board move in a way that D_i does not change significantly. The best strategy could be keeping the camera set in fixed location and instead moving the calibration board in different directions and angles. As a requirement, the calibration board should be videotaped from different angles and the whole pattern should be visible in all video frames. The collected data is then used as the input in a conventional calibration algorithm to find the required parameters. The result corresponding to D_i is saved and the process is repeated for another D. The outcome is a multiple set of calibration parameters $\{P_i \mid i = 1,...,n\}$, each corresponding to a specific D. It is necessary to mention that while videotaping, it is preferred to move the camera set such that the calibration board appears at different areas of video frames. It is known that if the calibration pattern only appears at the central part of video frames, the estimations will behave poorly at peripheral areas (Zhang, 2000).

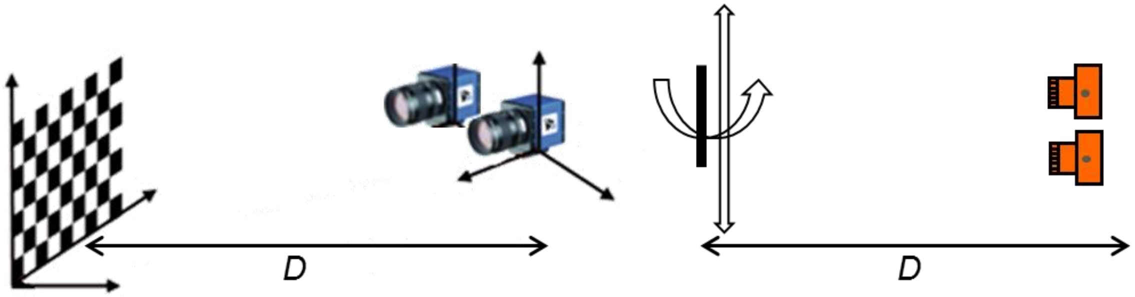


Figure 2.2: Data collection schematic for a D value (left: side view; right: top-down view)

The sensor system is then used to collect stereo videos from a target infrastructure scene. In the processing step (i.e., SfM), the results of the proposed multi-step stereo camera calibration procedure are used. First, the average of each calibration parameter is calculated from $\{\mathrm{P}_i \mid i = 1,...,n\}$. In the visual triangulation step, these average values are used to find an initial estimation of 3D coordinates of points. For point j in k-th stereo view (p_{jk}), the P_i that has the closest D to the Z coordinate of p_{jk} is found. The corresponding calibration parameters are then tied to p_{jk} and the 3D coordinates are recalculated. This new information is inputted in the bundle adjustment process to achieve the final estimation for 3D points.

As can be inferred, this section did not aim to provide new mathematical relationships for estimating the necessary parameters. Instead, conventional camera calibration algorithms (e.g., OpenCV or Bouguet's camera calibration toolbox) were used; but, a new procedure was proposed to perform separate conventional camera calibration for some predefined D values.

2.5. Improved Affine Invariant Descriptor Vector for Line Segments

Line segments cannot be described using the conventional correlation windows or any modified version of this approach because of the following reasons: unstable location of end-points, weak epipolar constraint (i.e., only epipolar beam and not epipolar line), and lack of distinctive texture in the local neighborhood. Previous studies have proposed using histogram-based descriptor vectors (e.g., image gradients, color, clustering according to spatial proximity) to address this issue; however, most of these algorithms fail to provide a highly distinctive descriptor for line matching under rotation, illumination change, image blur, viewpoint change, and partial occlusion. The primary reason is the following: these algorithms (such as MSLD (Wang, et al., 2009)) assign a rectangular pixel support region to every line segment with predefined dimensions disregarding the length of the line or the visual texture in the local neighborhood; therefore, the information inside the support region could be very different for a specific line in two views which makes the matching process very difficult if not impossible. This section hypothesizes that assigning a dynamic pixel support region for a line segment will increase the distinctiveness of the generated descriptor vector. The MSLD algorithm (Wang, et al., 2009) is considered here as the basis and the proposed hypothesis is applied to modify/improve this algorithm.

Straight lines are initially detected in an image using the exiting state-of-the-art methods that work based on the scale-space theory such as (Khaleghi, et al., 2009). Collinear line segments are then merged to generate long line segments. Since line segments are often not fully extracted and split into several smaller (more or less) collinear line fragments, the merging process is necessary to avoid mismatches because of gaps in the edge response. The output of these two primary steps is used as the input for the next step which is constructing descriptor vectors for a line segment.

A very important issue that should be dealt with in the beginning is the problem of occluding object boundaries (Figure 1.6). In such boundaries, a change in viewpoint will cause inconsistency with the image in one side of the occlusion and make the descriptor vector inaccurate. A very simple solution for this problem is to separate the pixel support region into two parts, one on either side of the line segment, and generate a descriptor for each of these sides (side 1 and side 2). Since only one of the two descriptors will be on the side of the occluding boundary, the information from the other one can be used for locating robust matches. An issue that arises with this strategy is the question that which descriptors should be used to compare line segment A with B (i.e., A-side-1 with B-side-1 or B-side-2 and vice versa). The use of the epipolar geometry is proposed here to address this problem. For this purpose, line segments in each view are converted into directed lines. The two end-points of a line segment are randomly labeled as s and e; the directed line is therefore $\overrightarrow{se}$. The side in the clockwise direction is called $side1$ and the other side is labeled as $side2$. Now, the epipolar lines corresponding to the points s_1 and e_1 in view 1 are found using the fundamental matrix. If the epipolar lines are not parallel to the line in view 2, two scenarios could happen:

- Epipolar line for s_1 is closer to s_2 and the one for e_1 is closer to e_2; in this case, side 1 in view 1 should be compared with side 1 in view 2 and side 2 in view 1 should be compared with side 2 in view 2
- Epipolar line for s_1 is closer to e_2 and the one for e_1 is closer to s_2; in this case, side 1 in view 1 should be compared with side 2 in view 2 and side 2 in view 1 should be compared with side 1 in view 2

However, if the epipolar lines are more or less parallel to the line in view 2, the abovementioned approach cannot be used. In this case, the unit normal vector of the directed line in view 1 (n) is calculated in the clockwise direction and added to e_1 to achieve point p. The fundamental matrix between two views is used to calculate the epipolar line corresponding to p. As shown in Figure 2.3, the dot product between the

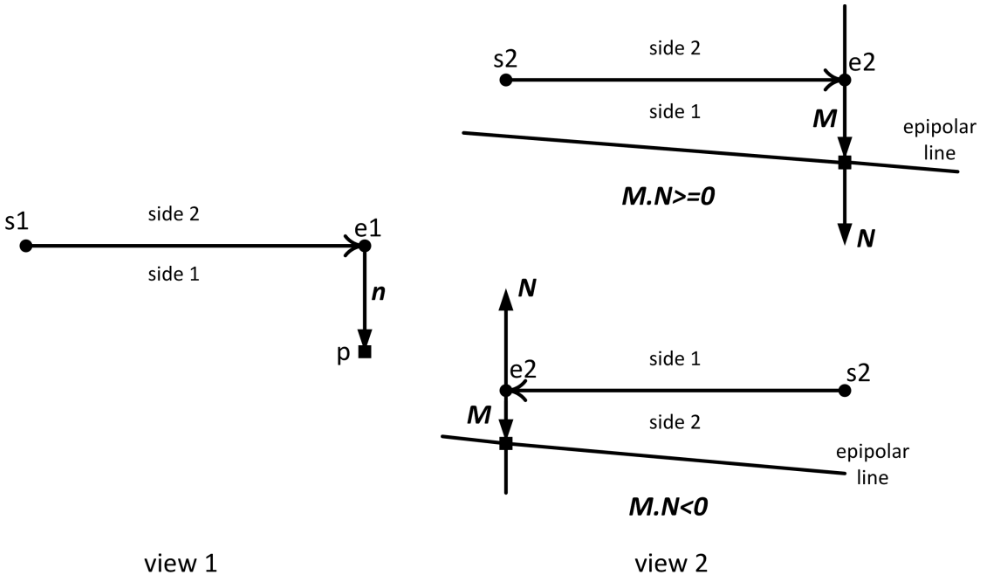


Figure 2.3: Determining the side correspondence in two views using epipolar geometry

normal vector of the line in view 2 (N) and the vector from e_2 to the intersection of the epipolar line and normal vector can determine the corresponding sides. If the dot product is greater than or equal zero, the sides with same numbers should be compared to each other and vice versa.

After addressing the problem of occluding boundaries, a support region needs to be determined for each side of a line segment. An algorithm which works based on locating zero-crossings in the Laplacian function is proposed here. It has been shown that a region enclosed by the zero-crossings of the Laplacian operator is scale-invariant and also insensitive to a wide range of viewpoint transformations (Lindeberg, 1998) (Tuytelaars & van Gool, 2000). In order to locate such a region, a seed point should to be given first. The proposed algorithm uses the two end-points of a line segment as well as its middle point for this purpose. Starting from each seed point, the Laplacian operator is used on the pixels along rays emanating from the seed point (Figure 2.4). The zero-crossing on each ray is marked as a boundary point (the zero-crossings are typically located at places that image gradient changes rapidly). The enclosed region that is generated by connecting these boundary points is the region of interest.

The generated support region has the following features: a) the shape and size of the region is more or less the same despite the potential inaccuracies in locating the end-points of a physical line segment in an image (Figure 2.5); b) in poorly-textured areas, the region expands until edge-like points are detected; hence, the region always consists of

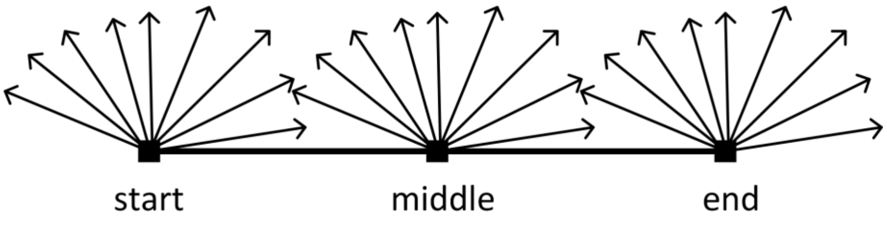


Figure 2.4: Laplacian along “rays” emanating from start, middle, and end points of a line segment

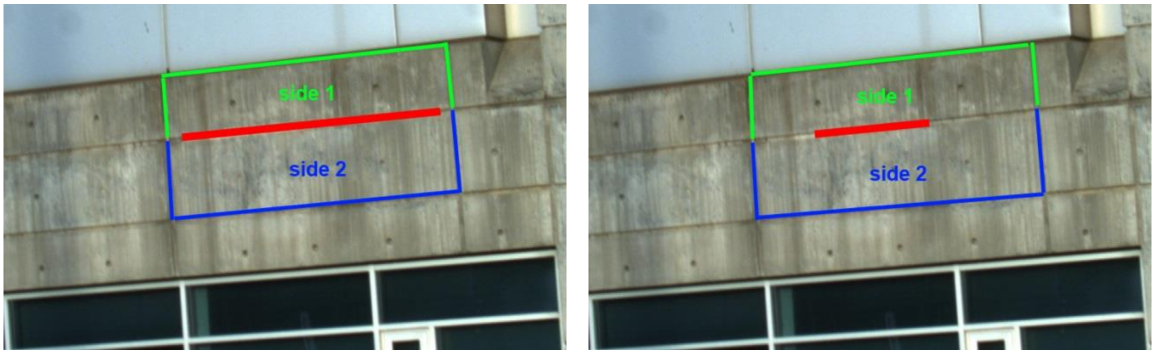


Figure 2.5: Shape and size of the support region despite inaccuracies in locating the end-points

points that are distinctive; c) the shape and size of the region is more or less the same in two views with viewpoint changes or distortions.

Once a pixel support region is assigned to each line segment, a multi-dimensional descriptor vector needs to be constructed for each region based on the local image gradients of the enclosed pixels. The direct use of the MSLD algorithm (Wang, et al., 2009) in this case would not result in desirable and distinctive descriptors because of the viewpoint changes, image rotations, and distortions. This research proposes using the canonical representation of the regions before assigning the descriptors. It is known that two directions are required for canonical representation. The direction of the line segment is proposed to be used as the first direction. In order to determine the second direction, the histogram of the angle of gradients for pixels inside the region should be found first. The peak of this histogram is proposed to be used as the second direction. An example is demonstrated in Figure 2.6. Finally, MSLD algorithm is applied on the canonical representation of the regions and its similarity measure is used to match line segments in different views.

2.6. Hybrid Bundle Adjustment

In a 3D reconstruction pipeline, the 3D structure of an environment and the motion of a camera set are initially estimated using linear methods (in a video-based

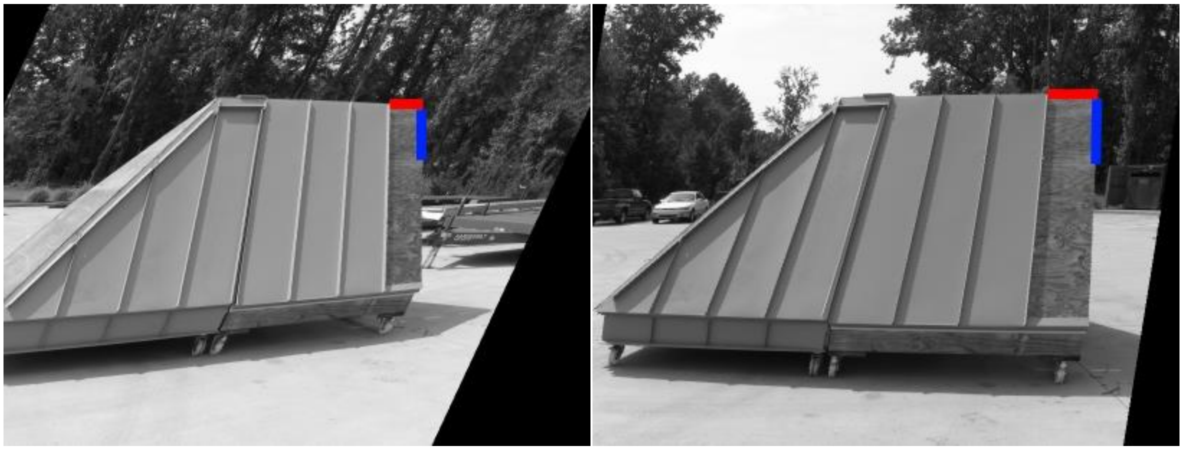


Figure 2.6: An example for canonical representation of support regions (red/horizontal line: line segment; blue/vertical line: direction of the peak of the histogram of angle of gradients

method, the local motion of the camera set between two frames could be simply initialized as zero); however, they are refined later in a non-linear optimization process called bundle adjustment. The Sparse Bundle Adjustment (SBA) package in (Lourakis & Argyros, 2009) is currently used for monocular point-based SfM as a well-known and efficient tool. The mathematical relationships used in the SBA package for solving a large but sparse optimization problem are generic and could be generalized for problems that include lines or a combination of points and lines. However, no such formulation exists in the literature. The focus of this section is therefore to provide the mathematical relationships needed in a stereo-based hybrid bundle adjustment process such that the SBA package can be modified and used for solving the given optimization problem. The main challenge here is to parameterize 3D points and lines as well as their reprojections in the 2D image space with the same number of parameters in order to be able to use the SBA package. These formulations are presented for two different scenarios: a) all stereo camera calibration parameters are known and fixed; and b) the extrinsic parameters (i.e., rotation and translation between the left and right cameras) are known and fixed but only an initial estimation is available for the internal parameters.

2.6.1. Hybrid bundle adjustment if all calibration parameters are known

Assuming all the internal and external parameters of the stereo camera setup are known through the calibration process and the captured video frames are undistorted, location of the stereo camera rig in the environment at time i is denoted by seven parameters (three for translation and a unit quaternion vector for rotation). Therefore, the camera projection matrices at time i (P_L^i and P_R^i) can be represented as follows

$$P_L^i = K_L \left[R_L^i \mid t_L^i \right] \qquad (2\text{-}5)$$

$$R_L^i = \frac{1}{e_i^2 + f_i^{\,2} + g_i^2 + h_i^2} \times \begin{bmatrix} e_i^2 + f_i^{\,2} - g_i^2 - h_i^2 & 2 f_i g_i - 2 e_i h_i & 2 f_i h_i + 2 e_i g_i \\ 2 f_i g_i + 2 e_i h_i & e_i^2 - f_i^{\,2} + g_i^2 - h_i^2 & 2 g_i h_i - 2 e_i f_i \\ 2 f_i h_i - 2eg & 2 g_i h_i + 2 e_i f_i & e_i^2 - f_i^{\,2} - g_i^2 + h_i^2 \end{bmatrix} \qquad (2\text{-}6)$$

$$P_R^i = K_R \left[R_L^i R \mid R_L^i t + t_L^i \right] \qquad (2\text{-}7)$$

where K_L and K_R are the left and right intrinsic camera parameters; R_L^i and t_L^i are the rotation matrix and translation vector of the left camera at time i with respect to a predefined coordinate system (typically left camera center at time $i = 0$); $e_i + f_i \vec{i} + g_i \vec{j} + h_i \vec{k}$ is the quaternion representation of R_L^i (the unit quaternion representation is used for R_L^i to reduce the number of optimization parameters from 9 to 4 for each camera view); R and t are the extrinsic (i.e., rotation and translation) calibration information of the stereo camera rig.

The set of camera calibration information, that is estimated based on the procedure presented in section 2.4, is used here. For a specific feature and based on the disparity in the left and right views, an initial depth value is estimated using the information corresponding to one of D values. Then, the closest D value to the initial depth is found and the corresponding calibration information is used afterwards.

Knowing left and right camera projection matrices at time i allows calculating 2D image coordinates of the projection of 3D points and lines in the stereo video frames at time i. If j-th 3D point is denoted by $N_j = \begin{bmatrix} X & Y & Z & W \end{bmatrix}^T$, homogeneous coordinates of its projection in left and right camera views at time i (n_L^{ij} and n_R^{ij}) are

$$n_L^{ij} = P_L^i N_j \tag{2-8}$$

$$n_R^{ij} = P_R^i N_j \tag{2-9}$$

On the other hand, a 3D line can be represented by a homogeneous Plucker 6-vector L. Given two 3D points $M^T \sim \left(\overline{M}^T \mid m\right)$ and $N^T \sim \left(\overline{N}^T \mid n\right)$:

$$L^T \sim \left(a^T \mid b^T\right) \tag{2-10}$$

$$a = \overline{M} \times \overline{N} \quad , \quad b = m\overline{N} - n\overline{M} \tag{2-11}$$

k-th 3D line in Plucker coordinates (L_k) can be parameterized by orthonormal representation proposed in (Bartoli & Sturm, 2005). Four optimization parameters are needed for each 3D line which is the same number of parameters used for 3D points; this satisfies the need for representing 3D points and lines with the same number of parameters. If the initial estimation for a 3D line is given by $L_0{}^T \sim \left(a_0{}^T \mid b_0{}^T\right)$, the orthonormal representation of L_0 is $\left(U, W\right) \in SO(3) \times SO(2)$ where

$$U = \begin{pmatrix} \frac{a_0}{\|a_0\|} & \frac{b_0}{\|b_0\|} & \frac{a_0 \times b_0}{\|a_0 \times b_0\|} \end{pmatrix} \tag{2-12}$$

$$W = \begin{pmatrix} \|a_0\| & -\|b_0\| \\ \|b_0\| & \|a_0\| \end{pmatrix} \tag{2-13}$$

The four optimization parameters are $\mathrm{P}^T = \left(\theta^T \mid \alpha\right)$ where the 3-vector θ and the scalar α are used to update U and W. Once P is computed in the minimization process, $U \leftarrow UR(\theta)$ and $W \leftarrow WR(\alpha)$. Using this representation, homogeneous coordinates of the projection of k-th 3D line in left and right camera views at time i (l_L^{ik} and l_R^{ik}) can be calculated by

$$l_L^{ik} = \tilde{P}_L L_k \tag{2-14}$$

$$l_R^{ik} = \tilde{P}_R L_k \tag{2-15}$$

where $\tilde{P}_L$ and $\tilde{P}_R$ are 3×6 matrices which can be found as follows. Given a 3×4 camera projection matrix $P = \left(\overline{P} \mid p\right)$, $\tilde{P}$ is defined as $\tilde{P} = \left(\det\left(\overline{P}\right)\overline{P}^{-T} \mid \left[p\right]_\times \overline{P}\right)$ to project Plucker line coordinates. Consequently and similar to points, the projection of lines in 2D image space needs three parameters.

Jacobian of the reprojection function $f\left(a,b,c\right) = \hat{x}$ is another important part of the hybrid bundle adjustment process presented here. The function f takes $a = \left(a_1^T, a_2^T, ..., a_i^T\right)^T$, $b = \left(b_1^T, b_2^T, ..., b_j^T\right)^T$, and $c = \left(c_1^T, c_2^T, ..., c_k^T\right)^T$ as parameters and returns $\hat{x} = \left(\hat{x}_{p11}^T, ..., \hat{x}_{pij}^T, \hat{x}_{l11}^T, ..., \hat{x}_{lik}^T\right)^T$. Here, a_i is the vector of estimated location of the left camera at time i, including a translation vector and a unit quaternion vector for rotation (7 parameters in total); b_j is the vector representing the 4 parameters of the j-th world point in the homogeneous coordinate system; c_k is the vector including the 4 parameters used for orthonormal representation of the k -th world line; $\hat{x}_{pij}$ is the projected homogeneous image coordinates of world point j in the i -th stereo frame (6 parameters in total); and $\hat{x}_{lik}$ is the homogeneous projected image coordinates of world line k in the i -th stereo frame (6 parameters in total). The Jacobian matrix, in this case, will have a sparse structure.

Once the coordinates of projected points and lines in the stereo video frames are found, total reprojection error is calculated by adding individual errors of all points and lines. For each point, the Euclidean distance between the original image point and its reprojection is considered as the error. The error for each line is the normalized area between the original line and its reprojection; the normalized error is the result of dividing the area by the length of the original line. SBA minimizes the summed squared reprojection error.

The output of the whole process is a sparse 3D point cloud and a set of infinite 3D lines that their Plucker coordinates are known. 3D points can be directly visualized and used in subsequent processes; however, infinite 3D lines need to be converted to 3D line segments with specific end-points. The following equations could be used to find 3D points that are on a 3D line with Plucker coordinates.

If $M = (m_1, m_2, m_3, 1)$ and $N = (n_1, n_2, n_3, 1)$ are two 3D points with homogeneous coordinates, the Plucker coordinates of the line that is connecting them are

$$L \sim (a_1, a_2, a_3, b_1, b_2, b_3) \tag{2-16}$$

$$a_1 = m_2 n_3 - m_3 n_2 \quad , \quad a_2 = m_3 n_1 - m_1 n_3 \quad , \quad a_3 = m_1 n_2 - m_2 n_1 \tag{2-17}$$

$$b_1 = n_1 - m_1 \quad , \quad b_2 = n_2 - m_2 \quad , \quad b_3 = n_3 - m_3 \tag{2-18}$$

If the moment part of the coordinates is denoted by $V = (v_1, v_2, v_3) = (a_1, a_2, a_3)$ and the direction part is denoted by $W = (w_1, w_2, w_3) = (b_1, b_2, b_3)$, a 3D point $Q = (q_1, q_2, q_3, 1)$ on L must satisfy the following equation

$$W \times Q = -V \text{ where } \times \text{ is the cross product} \tag{2-19}$$

$$w_2 q_3 - w_3 q_2 = -v_1 \tag{2-20}$$

$$w_3 q_1 - w_1 q_3 = -v_2 \tag{2-21}$$

$$w_1 q_2 - w_2 q_1 = -v_3 \tag{2-22}$$

In the given problem, the 3D coordinates of Q are unknown, but V , W , 2D coordinates of the line end-points in the image space, and the projection matrix are known. Therefore, a linear system of equations can be established and solved for Q .

2.6.2. Hybrid bundle adjustment if only extrinsic calibration parameters are known

When the exact values for internal camera parameters are not known and only an initial estimation of their values have been calculated, captured video frames cannot be reliably undistorted. This means that straight lines may appear as curved line segments and hence the algorithm presented in the previous section cannot be used. Preliminary findings are indicating that this scenario is probable due to the uncertainties that we face in estimating these parameters. Moreover, the Euclidean accuracy of 3D points and lines is sensitive to the distortion coefficients in far-range applications which is another reason that suggests to include internal camera parameters in the optimization process. Following paragraphs provide mathematical relationships needed for this purpose.

The required process for feature points is straight forward. Feature points are detected in distorted video frames and then initial estimation of camera parameters (information corresponding to an appropriate *D* value) are used to find an initial estimation for 3D points and camera motion. The projection model in this case is the following (only left camera is considered here for simplicity but the same formulae can be used for right camera by changing the camera matrix and other information)

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = P_L^i \begin{bmatrix} X \\ Y \\ Z \\ W \end{bmatrix} \tag{2-23}$$

$$x' = x/z \quad , \quad y' = y/z \tag{2-24}$$

$$x'' = x'(1 + k_1^L r^2 + k_2^L r^4 + k_3^L r^6) + 2 p_1^L x' y' + p_2^L (r^2 + 2x'^2) \tag{2-25}$$

$$y'' = y'(1 + k_1^L r^2 + k_2^L r^4 + k_3^L r^6) + p_1^L (r^2 + 2y'^2) + 2 p_2^L x' y' \tag{2-26}$$

$$u = f_x^L \times x'' + c_x^L \quad , \quad v = f_y^L \times y'' + c_y^L \quad , \quad w = 1 \tag{2-27}$$

where $r^2 = x'^2 + y'^2$, $\left(k_1^L, k_2^L, p_1^L, p_2^L, k_3^L\right)$ are distortion coefficients for the left camera, $\left(f_x^L, f_y^L\right)$ are the horizontal and vertical focal length of the left camera, and $\left(c_x^L, c_y^L\right)$ is the principal point. These formulations are used to calculate reprojection of 3D points and the corresponding Jacobian matrix. The reprojection error is the Euclidean distance between the detected point in the distorted view and its reprojection into the 2D image space.

In case of line features, the process is more complex. First, the initial estimations of internal camera parameters are used to undistort a video frame. Line segments are then detected in the undistorted view. On the other hand, an edge point detection algorithm

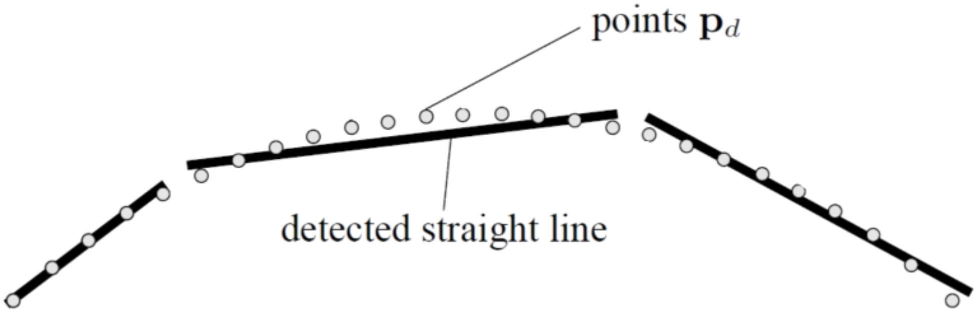


Figure 2.7: Detection of points P_d

with subpixel accuracy is used on the original (i.e., distorted) view to find the subpixel location of the edge pixels that correspond to the detected lines in the undistorted view. This way, a number of points on the curved line segment in the original view are acquired. These points (P_d) will be used in next steps to calculate reprojection errors (Figure 2.7).

Similar to the case of feature points, initial estimation of camera parameters are used to find an initial estimation for 3D lines and camera motion. The same projection process that is described in the previous section is applied to find the projection of 3D lines. This leads to coordinates of projected line in the undistorted view. To calculate the reprojection error, undistorted coordinates of P_d are first found using the internal camera parameters in the optimization process. Below is the algorithm that can be used for this purpose.

```
$x_0 = (u_d - c_x) / f_x;$
$y_0 = (v_d - c_y) / f_y;$
$x = x_0;$
$y = y_0;$
for (int iter = 0; iter < 5; iter++) {
$r2 = x * x + y * y;$
$icdist = 1.0 / (1 + ((k_3 * r2 + k_2) * r2 + k_1) * r2);$
deltaX = 2 * $p_1$ * x * y + $p_2$ * (r2 + 2 * x *x);
deltaY = $p_1$ * (r2 + 2 * y * y) + 2 * $p_2$ * x * y;
x = ($x_0$ – deltaX) * icdist;
y = ($y_0$ – deltaY) * icdist; }
u = (x * $f_x$) + $c_x$;
v = (y + $f_y$) + $c_y$;
```

where u_d and v_d are distorted coordinates of a point while u and v are its undistorted coordinates. The reprojection error is the sum of the Euclidean distance between undistorted coordinates of P_d and the projected line, divided by the number of points in P_d.

2.7. Hypothesizing and Verifying Planar Surfaces

The ultimate goal of this section is automatic identification of salient planar regions using global scene information as the input data (a sparse 3D point cloud, 3D line segments, multi-view correspondence of points and lines, and camera projection matrix for each view). The input data is the output of the proposed hybrid SfM algorithm. Two major steps need to be followed in order to identify salient planar regions: generating plane hypotheses and verifying candidate planes.

2.7.1. Generating plane hypotheses

A three-step strategy is proposed to formulate candidate planes. In the first step, half-planes are identified using a combination of 3D points and line segments while satisfying coplanar and region constraints. The region constraint restricts the range over which the search process is performed. It is simply a rectangle around the projection of a 3D line in one of the views. The length of this rectangle is the length of the 2D line in the view and the width can be selected as a predefined percentage of the line length. On the other hand, the coplanar constraint limits the selection of candidate half-planes to those in which all the feature points are coplanar in the 3D world scene. A successful formulation is achieved if coplanar feature points exist that are located in the local neighborhood of a 3D line segment. As can be inferred, this is a strong assumption which may not hold true in many scenarios. The second and third steps address this issue. These two steps only consider the points and lines that have not been used for hypothesized half-planes during the first step. The second step formulates a plane from two intersecting line segments and verifies their coplanarity using intersection context. On the other hand, a RANSAC-based approach is used in the third step to hypothesize a plane from at least three points.

Identification of Half-Planes: Assuming that each correct half-plane contains at least one corresponding feature point in multiple views, a half-plane is parameterized using a 3D line segment L and a 3D feature point X which is located within a restricted 3D region around L. This region is defined using a normal distribution centered at L with a standard deviation of σ. Let S_p be the set of all points in the 3D sparse point cloud that satisfy the region constraint for L. If the number of points in S_p is greater than one ($|S_p| \geq 1$), sufficient information exists to hypothesize a half-plane and then verify it. For each point X_i in S_p, the number of points that are coplanar with the half-plane parameterized by L and X_i is computed and the set S containing feature points whose associated half-plane corresponds to the highest number of coplanar points is identified.

$$S = \max S_j = \{X_i \mid d(X_i, \pi(L, X_j)) < \varepsilon, \forall X_i \in S_p\} \qquad (2\text{-}28)$$

where $\pi(L, X_j) = (\pi_1, \pi_2, \pi_3, \pi_4)$ is the homogeneous 4-vector π in the world coordinate system that represents the half-plane parameterized by the 3D line L and the feature point X_j. The same process is applied to all 3D line segments and the points and lines that have been used to parameterize half-planes are marked as “used”.

Coplanar Lines from Intersection Context: Line segments that at least one side of them have not been used in the half-plane formulation step are considered here. Given this line set, intersecting lines are paired when both are closely located (i.e., the distance between end points and the intersection point is less than a threshold). Although there is a high probability that two intersecting lines are coplanar, the final set could generally include both coplanar and non-coplanar pairs. The discrimination is therefore based on the following fact: when the intersection point of two intersecting lines is back-projected onto the 2D image space, a match of the intersection point can be found in multiple views. The matching similarity is estimated by assigning invariant region descriptors such as SIFT or SURF to back-projected intersection points and computing the Euclidean distance between them. If two lines are labeled as coplanar, a plane hypothesis is generated from homogeneous coordinates of end-points of the first line segment (X_1 and X_2) and one of the end points of the second line segment (X_3).

$$X_1 = \begin{pmatrix} \tilde{X}_1 \\ 1 \end{pmatrix} \quad X_2 = \begin{pmatrix} \tilde{X}_2 \\ 1 \end{pmatrix} \quad X_3 = \begin{pmatrix} \tilde{X}_3 \\ 1 \end{pmatrix} \text{ where } \tilde{X} = (X, Y, Z)^T \tag{2-29}$$

$$\pi = \left(\left(\tilde{X}_1 - \tilde{X}_3\right) \times \left(\tilde{X}_2 - \tilde{X}_3\right), \quad -\tilde{X}_3^{\,T}\left(\tilde{X}_1 \times \tilde{X}_2\right)\right) \tag{2-30}$$

RANSAC-Based Plane Hypothesis Generation from Points: The objective here is to find multiple locally fit models for the set of 3D points that have not been used in the half-plane identification step (S'_p) using a RANSAC-based approach (i.e., sampling, scoring, and contiguity). A plane equation can be computed from three points in S'_p that have been sampled randomly. The first point is selected while considering a uniform distribution over S'_p. However, the second and third points are sampled from normal distributions centered at the first point with a standard deviation of σ'. The plane equation (i.e., model) can be calculated using equations 2-29 and 2-30. Each model is then evaluated according to points that are at nearby distance of the original samples. The inlier set for each model is determined by computing the distance of each point to the plane. A new plane is obtained as the least-square fit to the inlier points.

2.7.2. Verifying candidate planes

The output of the described procedures in the previous section is a set of plane hypotheses that may include repetitive or incorrect candidates. This section proposes to use photo-consistency measures for identifying correct planes according to the geometry demonstrated in Figure 2.8. The figure shows that point to point correspondences can be calculated over a set of images using the homographies defined by a 3D plane equation. Given the plane π, a 3×3 homography matrix H_j^i is found between the i-th and j-th

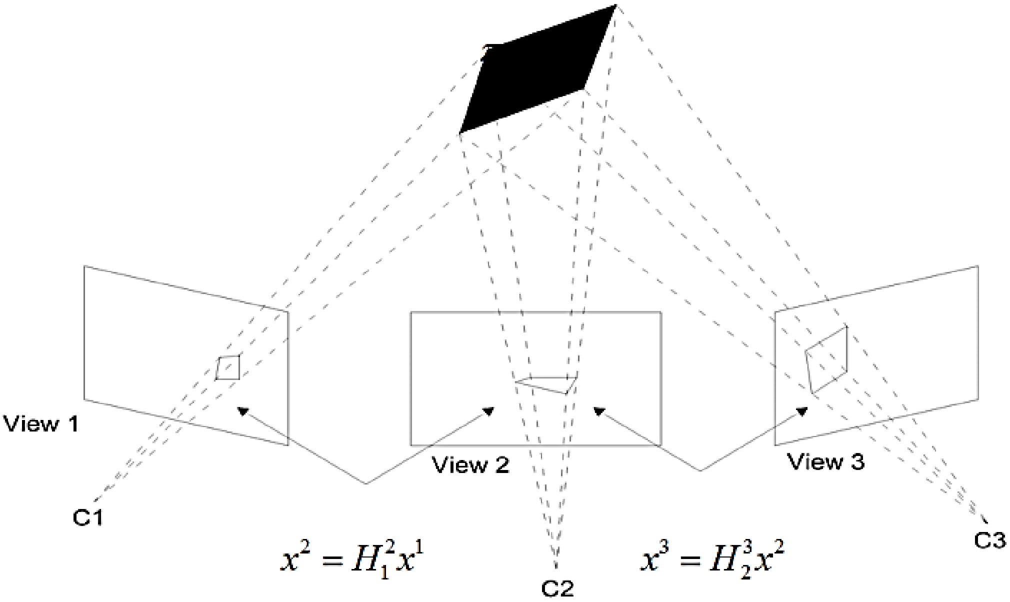


Figure 2.8: Geometric correspondence between views

views, so that a homogeneous point x^i in the i -th view is mapped to x^j in the j -th view

$$x^j = H_i^j x^i \tag{2-31}$$

The 3×4 camera projection matrices for each view are used to calculate the homography matrices. Considering two views, if the projection matrix for the first view is presented in the canonical form $P = [I \mid 0]$ and the projection matrix for the other view is denoted by $P' = [A \mid a]$ then

$$H = A + aV \text{ where } V = -\frac{1}{\pi_4}(\pi_1, \pi_2, \pi_3)^T \tag{2-32}$$

where π is the plane in the homogeneous 4-vector form and $\pi_4 \neq 0$.

Having calculated the homography matrices for each pair of views, the following image intensity similarity score function is used to assess the correlation of image patches mapped by the homographies

$$Sim(\pi) = \sum_{i \in ValidViews} \left(\sum_{x_j \in POI} Cor^2\left(x_j, x_j^i\right) \right) \text{ where } x_j^i = H_j^i(\pi) x_j \tag{2-33}$$

where POI represents the points of interest that are obtained using the canny edge detector; and $Cor\left(x_j, x_j^i\right)$ is the normalized cross correlation between points x_j and x_j^i within a local n×n window. Plane candidates that their corresponding similarity score is less than a predefined threshold are discarded. The remaining candidates are evaluated based on the equation, slope, and perpendicular distance to find nearly identical planes that overlap. These planes are merged together and the plane equation is recalculated using a least-square optimization approach.

2.8. Performance Metrics

Although the main goal of this research is to investigate the technical feasibility of the proposed close-range video-based roof reconstruction framework, there are several intermediate steps involved in the process which can be evaluated separately according to different performance metrics. These intermediate steps are multi-step stereo camera calibration, straight line matching, hybrid SfM, and identifying salient planar regions.

Multi-step stereo camera calibration: The performance of the proposed calibration procedure is assessed based on the following metrics. a) Spatial accuracy of the initial estimation for 3D coordinates of points with different range values. In a stereo reconstruction, 3D points can be achieved from only left and right views of a scene at a given time. In this research, this 3D information is regarded as the initial estimation for 3D coordinates of points. Therefore, this metric aims to evaluate the spatial accuracy of the 3D points that are reconstructed from only left and right views while different calibration information is used. b) Spatial accuracy of a dense 3D point cloud. Since the reconstruction from only left and right views at a given time is not complete and accurate enough for the purpose of this research, a complete 3D reconstruction of the scene is generated from multiple sets of stereo frames. This metric aims to evaluate the accuracy of this reconstruction.

Improved straight line descriptors: Two performance metrics are used for this algorithm: recall and precision for line matching in two views. Recall is the number of correct matches divided by the total number of straight lines that are visible in both views. Precision is the number of correct matches divided by the total number of matches that are generated by the algorithm.

Hybrid SfM: Average reprojection error for point and line features, total number of processed views, completeness of the reconstruction, and spatial accuracy of the reconstructed features are the metrics to be considered for this step. These evaluations are performed in three scenarios: 1) only points are used; 2) only lines are used; and 3) a combination of points and lines are used.

Identify salient planar regions: The performance metrics for the detection accuracy include true positive (TP), false positive (FP), and false negative (FN). TP represents the planar regions that are detected correctly; FP shows the non-planar regions that are detected as a planar region; and FN represents the planar regions that are not detected.

Close-range video-based roof reconstruction: The proposed framework is supposed to generate a measureable 3D wire-diagram for every roof plane. Following metrics are used to evaluate this framework: completeness of the wire-diagram, Euclidean accuracy of the end-to-end dimensions of the edges, and the number of manual data that is needed to fix a missing edge.

It also needs to be noted that the processing time is not considered as an important factor in this study and hence it is not measured in any of the above mentioned steps.

CHAPTER 3

SYNTHESIS

3.1. Solution Implementation and Prototype

A prototype is created using Microsoft Visual C# and Windows Presentation Foundation (WPF) to implement the proposed framework. OpenCV (Intel® Open Source C++ Computer Vision Library) and VXL are selected as its main image processing libraries. They both are free and open source. The C# prototype provides a base to connect to any number of cameras through Ethernet network, USB, and IEEE 1394 connection with real-time responsiveness. On the other hand, C++ dll files (dynamic link library) are generated for several algorithms (e.g., hybrid bundle adjustment) and then invoked from the C# platform. The reason is twofold: better run-time efficiency by avoiding managed code; and direct use of the existing open source C++ codes that are available online.

A step by step procedure is followed to implement different steps of the proposed framework and test their performance. Following paragraphs provide more details about these steps.

Multi-step stereo camera calibration: An automatic stereo camera calibration algorithm is developed using the functions available in OpenCV. The user runs the program while videotaping a calibration pattern at a predefined distance from the camera set. The program is real-time responsive and automatically detects the calibration pattern in every video frame. Once the pattern is successfully detected in a pair of stereo video frames using the OpenCV's cvFindChessboardCorners function, chessboard corners are automatically refined to their location with subpixel accuracy and also matched between the two views by invoking the cvFindCornerSubPix function (Figure 3.1). This process

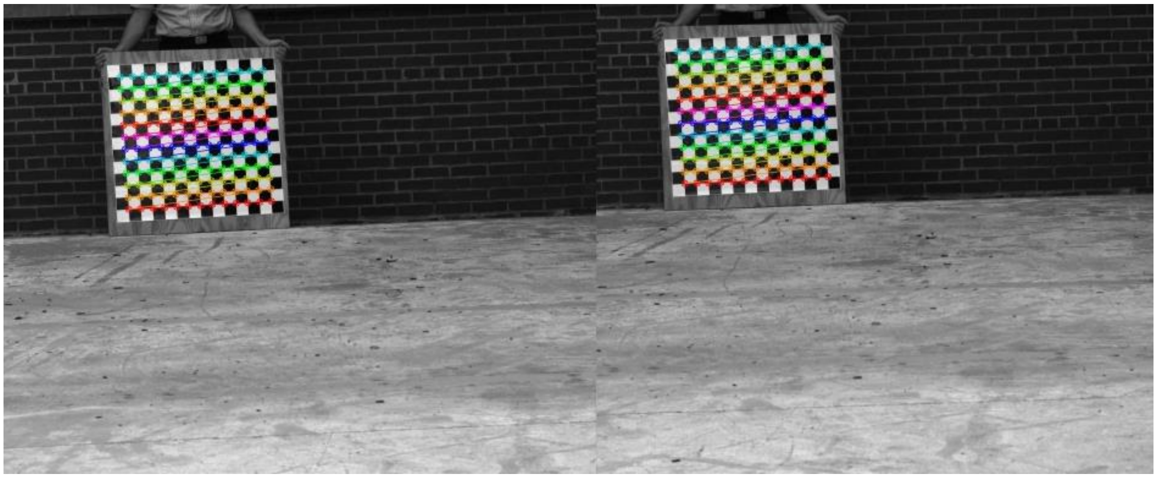


Figure 3.1: Automatically detected and matched corner points in the calibration process using OpenCV

continues until enough number of views are captured (typically between 30 to 40). Then, the calibration function (cvStereoCalibrate) is invoked and the necessary parameters are calculated. cvStereoCalibrate provides the possibility of calibrating a stereo camera set according to different constraints such as zero radial or tangential distortions, fixed principal point, fixed aspect ratio, and/or fixed focal length. The same process is repeated for different *D* values.

Feature point detection and matching: Among the various feature point detection algorithms available in the literature, the 64-dimensional version of SURF algorithm is selected. Once these feature points are detected, their corresponding descriptor vectors are computed. These points are then matched between the left and right views using the distance between descriptor vectors. A RANSAC algorithm is also used to further discard mismatches. All these steps are performed using the available functions in OpenCV library.

Line feature detection and matching: A combination of the algorithms presented by (Khaleghi, et al., 2009) and (Von Gioi, et al., 2010) is used to detect straight line segments. The detection algorithm aims at extracting lines that are repeatable and stable with respect to scale variations due to change of viewpoint. The algorithm takes a video frame $I(x, y)$ along with a set of Gaussian kernels $G(x, y, \sigma_i)$, where σ_i is the kernel width or scale. The first step is to generate the scale-space representation of the given video frame at k different scales through convolution with Gaussian kernels of different width. The DoG images are then computed and stored for each of the scales. Then, Line Segment Detector (LSD) proposed by (Von Gioi, et al., 2010) is applied to localize potential line segments and keep only the ones for which the DoG attains an extrema over scales. The algorithm initially computes the level-line angle at each pixel to generate a level-line field. The field is then segmented into connected regions of pixels that share the same level-line angle up to a certain tolerance (each connected region is a candidate for a line segment). The candidates are subject to a validation procedure which is based on the *a contrario* approach and the Helmholtz principle. The produced line segments may contain lines that are pieces of a longer line segment. To group these collinear segments, the prototype uses the HSV-based algorithm presented in (Bay, et al., 2005).

In order to build the HSV-based descriptor, an image is represented in the HSV color space which enables certain invariance towards illumination changes and provides enough distinctiveness between colors. The well-known quantization approach proposed in (Smith & Chang, 1995) is used to partition the HSV color space into 166 bins. This quantization approach places more importance on the hue channel than on saturation and value. For each line segment, a rectangular local neighborhood, perpendicular to the line, is selected and a color histogram ch is created for the pixels in the neighborhood. ch is the vector $\left[h_1, h_{2,} ..., h_{166}\right]$ in which each bin h_i contains the number of pixels having a certain color i, normalized by the total number of pixels in the selected neighborhood. Once a descriptor is generated for each line segment, the similarity between two segments is determined. This is measured using the distance between their descriptors

$$d = (ch_1 - ch_2)^T A (ch_1 - ch_2) \tag{3-1}$$

where $A = \lfloor a_{i,j} \rfloor$ is a 166×166 matrix and its elements are the Euclidean distance between c_i and c_j of the palette p in the quantized HSV color space (equation 3-2). If two line segments are collinear and have similar color histograms, they are merged into one longer segment.

$$a_{i,j} = \frac{1}{\sqrt{2}}\left[(V_i S_i \cos(H_i) - V_j S_j \cos(H_j))^2 + (V_i S_i \sin(H_i) - V_j S_j \sin(H_j))^2 + (V_i - V_j)^2\right]^{0.5} \tag{3-2}$$

After detecting line features and merging collinear ones, correspondences are established across multiple frame sets. For this purpose, multi-dimensional descriptor vectors are constructed for each line segment based on the improved affine invariant line descriptor proposed in section 2.5 and the MSLD algorithm (Wang, et al., 2009). The first step is to assign a dynamic pixel support region to each side of a line segment. As explained before, rays emanating from the start, middle, and end point of a line segment and zero-crossings of the Laplacian operator are used to locate an enclosed region. A gradient histogram is then constructed for the pixels inside each region according to the gradient magnitude m and orientation α at each pixel. If the location of a pixel in an image is shown by (x, y) and the intensity value of the pixel at (x, y) is represented by $I(x, y)$, the gradient magnitude $m(x, y)$ and orientation $\alpha(x, y)$ are calculated by

$$dx = I(x+1, y) - I(x-1, y) \quad , \quad dy = I(x, y+1) - I(x, y-1) \tag{3-3}$$

$$m(x, y) = \sqrt{dx^2 + dy^2} \quad , \quad \alpha(x, y) = \tan^{-1}\left(\frac{dy}{dx}\right) \tag{3-4}$$

where α is the angle of gradient (i.e., orientation); dy is the local intensity gradient in the vertical direction; and dx is the local intensity gradient in the horizontal direction. It can be seen that α is more or less equal to the orientation of the line segment in 2D image space.

The angle between a line segment and the gradient orientation of the pixels inside its support region could change significantly with viewpoint changes and hence adversely affect the matching process. To alleviate this problem, the perspective distortion of the support regions needs to be compensated. This is accomplished by rectifying a line segment and the peak of its support region's gradient orientation histogram into a special configuration in which the two directions are orthogonal. This normalized image region is called canonical representation. The rectification is performed by estimating a 2D homography matrix H_k from a region patch $P(x_k)$ to a canonical frame C_k, and the transformation is represented by

$$C_k \equiv C(x_k) \equiv P(H_k x_k) \tag{3-5}$$

Once the regions are transformed to the canonical frame, a modified version of the SIFT-like strategy proposed in (Wang, et al., 2009) is used to construct the line descriptor. As illustrated in Figure 3.2, a gradient histogram-based descriptor is constructed for support regions at each side of a line segment. The relative orientation values are found by subtracting the line orientation from pixel gradient orientations in the canonical frame; this helps to obtain rotation invariance. The relative orientation values at each support region are used to form orientation histograms that summarize the content over 8 bins covering the 360 degree range of orientations. Moreover, each relative orientation value is weighted by its gradient magnitude and Gaussian-weighted circular window with $\sigma = 1.5$, as suggested in (Lowe, 2004). The gradient description matrix

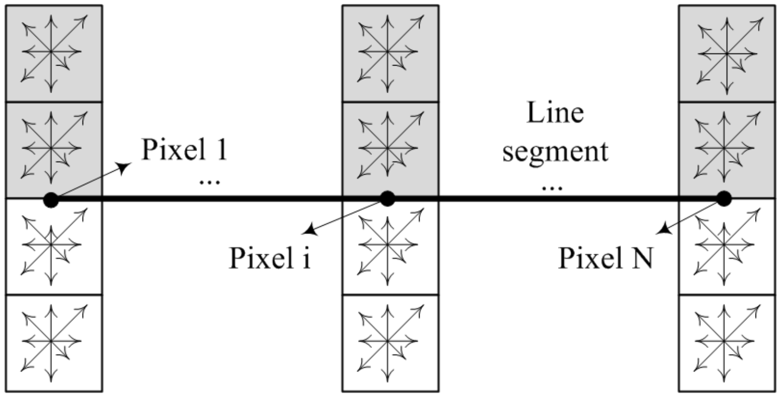


Figure 3.2: Gradient histogram for each side of a line segment

(GDM) concept proposed in (Wang, et al., 2009) is then used for each side of a line. Accordingly for each line segment L, two GDMs are formed. The mean and standard deviation of the vectors constructing a GDM is found and normalized to make the descriptor invariant to linear changes of illumination. The mean and standard deviation vectors are then concatenated to construct a 32-dimensional descriptor vector for each side of the line segment. Euclidean distance between descriptors is finally used to match line segment in different views.

Hybrid Structure from Motion: When a new pair of stereo video frames (s_i) is added to the processing pipeline, the visual features in the previous pair (s_{i-1}) are tracked/matched among the two consecutive frame sets. Matched visual features as well as the calibration information are used to calculate an initial estimation for camera motion between s_i and s_{i-1} (this can be also initialized to zero because of the use of video data). To calculate the camera motion, the trifocal tensor-based method proposed in (Pradeep & Lim, 2012) is used. Assuming a stereo setup in its canonical form and two subsequent stereo frame sets as input (four frames in total), two trifocal tensors are calculated based on $T^L = R_0 t^T - t_0 R^T$ and $T^R = R_0 (R_0 t + t_0)^T - t_0 (R_0 R)^T$; where R_0 and t_0 are the extrinsic calibration information; R and t encode the rigid camera motion between the two frame sets; T^L is a trifocal tensor that can be calculated from point and line correspondences between the left and right views before the motion and the left view after the motion; and T^R is a trifocal tensor that is calculated from feature correspondences in the left and right views before the motion and the right view after the motion. Using these relationships, one can write a linear system of equations in terms of the twelve unknown parameters of the motion.

The estimation of points, lines, and camera poses for the reconstructed scene is refined using the proposed hybrid bundle adjustment process. A modified version of the Sparse Bundle Adjustment package is developed to include the proposed mathematical formulation. The parameter vector p includes the parameters that have to be refined. The length of p is $7m + 4n_p + 4n_l$; 7 parameters for each camera pose, 4 parameters for homogeneous 3D coordinates of each point, and 4 parameters for orthonormal representation of every line. On the other hand, the measurement vector x includes the 2D coordinates of the detected features in the existing views. The length of x is $3n_{vp} + 3n_{vl}$; 3 parameters for homogeneous 2D coordinates of a world point that is visible in a view and 3 parameters for the homogeneous 2D coordinates of a world line that is visible in a view.

If camera poses are parameterized with a quaternion vector (q_1, q_2, q_3, q_4) as well as a translation vector (t_x, t_y, t_z) and a 3D point is parameterized with a 4-vector $(X, Y, X, 1)$, the point projection function can be defined as equation 3-6. The partial derivatives of this function with respect to the camera and point parameters are also needed in the optimization process. These derivatives are the Jacobians (equation 3-7).

$$(u, v, 1)^T = f_p(q_1, q_2, q_3, q_4, t_x, t_y, t_z, X, Y, Z, 1) \tag{3-6}$$

$$J_p = \frac{\partial \begin{pmatrix} u \\ v \\ 1 \end{pmatrix}}{\partial s_p} = \begin{bmatrix} \frac{\partial u}{\partial q_1} & \frac{\partial u}{\partial q_2} & \frac{\partial u}{\partial q_3} & \frac{\partial u}{\partial q_4} & \frac{\partial u}{\partial t_x} & \frac{\partial u}{\partial t_y} & \frac{\partial u}{\partial t_z} & \frac{\partial u}{\partial X} & \frac{\partial u}{\partial Y} & \frac{\partial u}{\partial Z} & 0 \\ \frac{\partial v}{\partial q_1} & \frac{\partial v}{\partial q_2} & \frac{\partial v}{\partial q_3} & \frac{\partial v}{\partial q_4} & \frac{\partial v}{\partial t_x} & \frac{\partial v}{\partial t_y} & \frac{\partial v}{\partial t_z} & \frac{\partial v}{\partial X} & \frac{\partial v}{\partial Y} & \frac{\partial v}{\partial Z} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \qquad (3\text{-}7)$$

The same camera parameterization is also used for lines. However in this case, two sets of parameters are needed for 3D lines: 6-vector initial Plucker coordinates $L_0^T = (p_1, p_2, p_3, p_4, p_5, p_6)^T$ which are constant and 4-vector optimization angles $(\theta_1, \theta_2, \theta_3, \alpha)$. Optimization angles are used to update the initial Plucker coordinates and then the updated line is projected into the image space. Having this parameterization, the line projection function is defined as equation 3-8. The Jacobians are also shown in equation 3-9.

$$(l_1, l_2, l_3)^T = f_l(q_1, q_2, q_3, q_4, t_x, t_y, t_z, \theta_1, \theta_2, \theta_3, \alpha, p_1, p_2, p_3, p_4, p_5, p_6) \qquad (3\text{-}8)$$

$$J_l = \frac{\partial \begin{pmatrix} l_1 \\ l_2 \\ l_3 \end{pmatrix}}{\partial s_l} = \begin{bmatrix} \frac{\partial l_1}{\partial q_1} & \frac{\partial l_1}{\partial q_2} & \frac{\partial l_1}{\partial q_3} & \frac{\partial l_1}{\partial q_4} & \frac{\partial l_1}{\partial t_x} & \frac{\partial l_1}{\partial t_y} & \frac{\partial l_1}{\partial t_z} & \frac{\partial l_1}{\partial \theta_1} & \frac{\partial l_1}{\partial \theta_2} & \frac{\partial l_1}{\partial \theta_3} & \frac{\partial l_1}{\partial \alpha} \\ \frac{\partial l_2}{\partial q_1} & \frac{\partial l_2}{\partial q_2} & \frac{\partial l_2}{\partial q_3} & \frac{\partial l_2}{\partial q_4} & \frac{\partial l_2}{\partial t_x} & \frac{\partial l_2}{\partial t_y} & \frac{\partial l_2}{\partial t_z} & \frac{\partial l_2}{\partial \theta_1} & \frac{\partial l_2}{\partial \theta_2} & \frac{\partial l_2}{\partial \theta_3} & \frac{\partial l_2}{\partial \alpha} \\ \frac{\partial l_3}{\partial q_1} & \frac{\partial l_3}{\partial q_2} & \frac{\partial l_3}{\partial q_3} & \frac{\partial l_3}{\partial q_4} & \frac{\partial l_3}{\partial t_x} & \frac{\partial l_3}{\partial t_y} & \frac{\partial l_3}{\partial t_z} & \frac{\partial l_3}{\partial \theta_1} & \frac{\partial l_3}{\partial \theta_2} & \frac{\partial l_3}{\partial \theta_3} & \frac{\partial l_3}{\partial \alpha} \end{bmatrix} \qquad (3\text{-}9)$$

$$L^T \sim \left(\left\|(p_1, p_2, p_3)\right\| \left(\frac{(p_1, p_2, p_3)}{\left\|(p_1, p_2, p_3)\right\|} \right)^T \quad | \quad \left\|(p_4, p_5, p_6)\right\| \left(\frac{(p_4, p_5, p_6)}{\left\|(p_4, p_5, p_6)\right\|} \right)^T \right) \qquad (3\text{-}10)$$

Maple 15.0 software package is used to generate the C code related to the point and line projection and Jacobian functions that are defined above. The output of this step is the optimized values for camera poses and 3D structure of the scene. The same process is repeated when a new pair of stereo video frames is added; all the information extracted from previous pairs and the new pair is optimized in the hybrid bundle adjustment process. Once all the pairs are added, the final output of the process is the 3D structure of the scene which includes 3D coordinates of feature points and lines.

Identify salient planar regions: Roof patches can be modeled as planar, convex polygonal patches with straight line segments connecting the corner points (Scholze, et al., 2002). The planar surfaces are modeled using a homography matrix. A half-plane extraction process is first performed by identifying points that are coplanar with a line segment. For each line segment, a rectangular region is defined and the 3D points that are enclosed in that region are assumed to be coplanar with the line. The length of the region is equal to the length of the line and its width is supposed to be 20% of the length. A half-plane π is formulated using a feature point X and a line L. Additional half-planes are also found from previously defined intersection context of coplanar lines and 3-point RANSAC-based approaches. The correct half-planes in the set of generated options are identified using the defined equations for image intensity similarity of the planes over multiple views. These half-planes are then merged to create larger planar regions. Two half-planes are merged if the angle difference between their normal vectors is less than 5^0, their perpendicular distance is less than 10cm, and there are no other planes in between (Scholze, et al., 2002).

Since the planes are built by merging half-planes that each correspond to a 3D line, a set of 3D lines can be associated to each plane. Having this information, the convex hull of the associated 3D lines is calculated for each plane; this helps to differentiate between border lines and other interior lines. In order to define the boundaries of a plane, a closed delineation is applied on its convex hull. This process is mostly based on heuristic rules. Two rules are used here. First, the end points of the border lines are updated if they intersect to each other or their inner end points are very close; the original end points are replaced with the intersection point and the convex hull is recalculated. Second, a priori knowledge about the geometry of roof structures is used to refine the calculated convex hulls. It is known that the angle between adjacent border lines in a roof plane can be reasonably modeled using 15^0 steps (Scholze, et al., 2002). A convex hull is replaced with the closest polygon that satisfies this assumption.

3.2. Design of Experiments

This section provides details of experiments that are designed to validate the proposed framework. To provide a thorough performance evaluation and address the defined objectives of this research, two groups of experiments have been designed and implemented in different environments. The first group evaluates the intermediate steps of the proposed framework as separate entities. On the other hand, the second group evaluates the performance of the whole framework as a package.

The primary control variable in these experiments is the technical properties of the sensor system which need to be fixed while collecting the necessary data. According to the formulation that is provided in section 2.3, two high-resolution cameras (2448×2048 pixels if the field of view and area covered by one pixel are assumed to be 700cm and 1cm, respectively) and two fixed focal length lenses (*f = 16mm, 25mm*) are used to setup a stereo camera system. The cameras have the capability to provide a video frame at any given time if asked by the software prototype; this enables capturing frames from two cameras at the exact same time. The stereo setup is attached to an extendible pole that can cover heights required for videotaping up to a two-story residential building.

3.2.1. Experiments for testing individual steps

Multi-step stereo camera calibration: In order to study the impact of using conventional stereo camera calibration procedures on the accuracy of 3D coordinates of

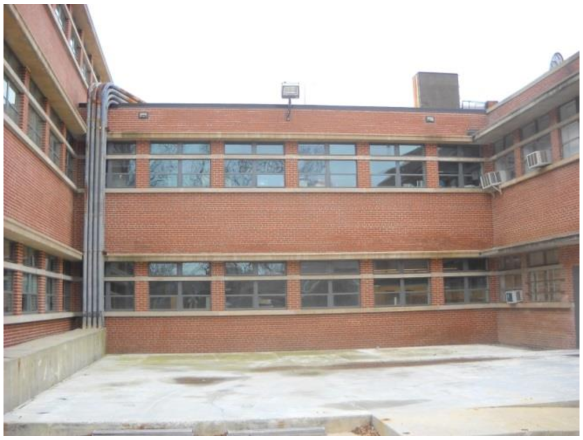


Figure 3.3: Well-textured planar environment for stereo camera calibration experiments

points versus the proposed one, a well-textured planar scene (Figure 3.3) is selected for two reasons: a) it allows controlling the Z coordinate of 3D points in the desired range by simply changing the distance between a stereo camera system and the planar face; and b) well-textured regions provide the opportunity to detect and match enough number of feature points in stereo views such that the results can be statistically significant. A set of two Flea2 cameras are used; these cameras are capable of streaming raw video data and comply with the aforementioned requirements. An appropriate baseline distance is also selected based on the following mathematical analysis and typical range values that are encountered in infrastructure applications (i.e., 10m to 25m).

Given the use of a stereo system, the baseline distance between the left and right cameras (b) can be selected based on a simple formulation presented in (Gallup, 2011) for analyzing the reconstruction accuracy in a stereo setup (Figure 3.4).

$$\varepsilon_z = \frac{bf}{d} - \frac{bf}{d+\varepsilon_d} = \frac{z^2\varepsilon_z}{bf + z\varepsilon_d} \approx \frac{z^2}{bf} \times \varepsilon_d \tag{3-11}$$

where z is the depth in cm, ε_z is the expected measurement error in cm, f is the focal length measured in pixels, and ε_d is the disparity error of a feature correspondence. As

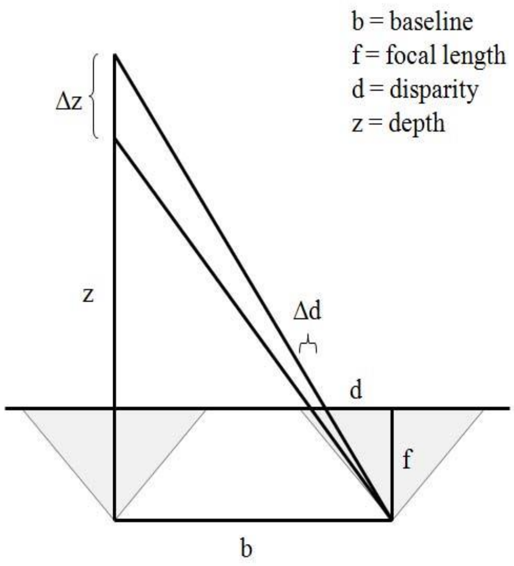


Figure 3.4: Depth error as a function of disparity error (Gallup, 2011)

an example, in case of using two 5MP cameras with 16mm fixed focal length lenses and assuming $z = 20m$, $\varepsilon_z = 2cm$, $f = 6500$, and $\varepsilon_d = 0.1$, the baseline distance can be calculated as 30cm.

A checkerboard with an appropriate number of black and white squares in two perpendicular directions is also required for the calibration process. The number of squares and their dimensions are selected according to the scene (a pattern of 13×14 squares each with a dimension of 60mm).

For camera calibration, six sets of stereo video streams are captured from the board under different conditions. In the first set which will be used for testing conventional procedures, the distance between the camera and the board changes in the range of $5m \le D_1 \le 15m$ while capturing the videos. Captured video frames should cover different views and angles of the board while the camera moves smoothly toward and/or away from the board. The next five sets are needed to test the proposed stereo camera calibration procedure. In these sets, the distance between the camera system and the board is fixed to $D_2 = 5m$, $D_3 = 10m$, $D_4 = 15m$, $D_5 = 20m$, and $D_6 = 25m$, respectively. These limits have been selected according to the typical range values that we encounter in building applications. The sensor system is also used to collect stereo videos from the planar scene while the distance of the camera to the planar scene changes from $5m \leq D \leq 25m$. This data is a control variable and will be used for 3D reconstruction of the scene in two scenarios: a) using conventional calibration procedures (parameters acquired from the 1st set of calibration videos); and b) using the proposed multi-step calibration procedure (multiple set of parameters acquired from the 2nd to 6th set of calibration videos).

The performance of the proposed calibration procedure is assessed based on the following metrics: a) spatial accuracy of the initial estimation for 3D coordinates of points with different range values (only one set of stereo frames is used in this case); and b) spatial accuracy of a dense 3D point cloud. For the first metric, stereo frames corresponding to $D = \{5,10,15,20,25m\}$ are extracted from the planar scene video to detect and match feature points. Calibration parameters from the conventional and proposed procedures are then used to estimate 3D coordinates of feature points from left and right views of stereo frames. Spatial distance between pairs of feature points is then calculated for each case and compared to the ground truth data that is acquired using total station surveying. For the second metric, calibration parameters from the conventional and proposed procedures are used separately in a dense 3D reconstruction package and the spatial accuracy of the results is evaluated. The sample size at all experiments is considered to be 50 which corresponds to 90% confidence level and ±10% confidence interval.

Improved affine invariant line descriptors: A modified version of the MSLD algorithm (Wang, et al., 2009) is proposed and hence need to be evaluated. Its performance is tested on real image pairs extracted from video streams. These pairs are not necessarily the left and right views of the stereo video streams and they could be left or right frames from different time stamps. Two primary criteria are used in this evaluation: ratio of correct matches CM to total number of lines that are visible in both views TL (i.e., recall), and ratio of correct matches CM to total number of matches TM (i.e., precision). CM, TM, and TL are determined manually and via visual inspection. To achieve a realistic comparison, all thresholds and decision making parameters are set to the values that have been recommended in (Wang, et al., 2009). The same matching criteria have been also applied in these experiments. For example, the dimension of the descriptor vectors is set to 72. The NNDR (nearest/next ratio) ratio and the global threshold are also set to 0.8 and 0.55, respectively.

In order to achieve 95% confidence and ±5% confidence interval, 400 image pairs are extracted from video streams or taken with a digital camera. This data have been collected from four different environments (Figure 3.5). These environments are a

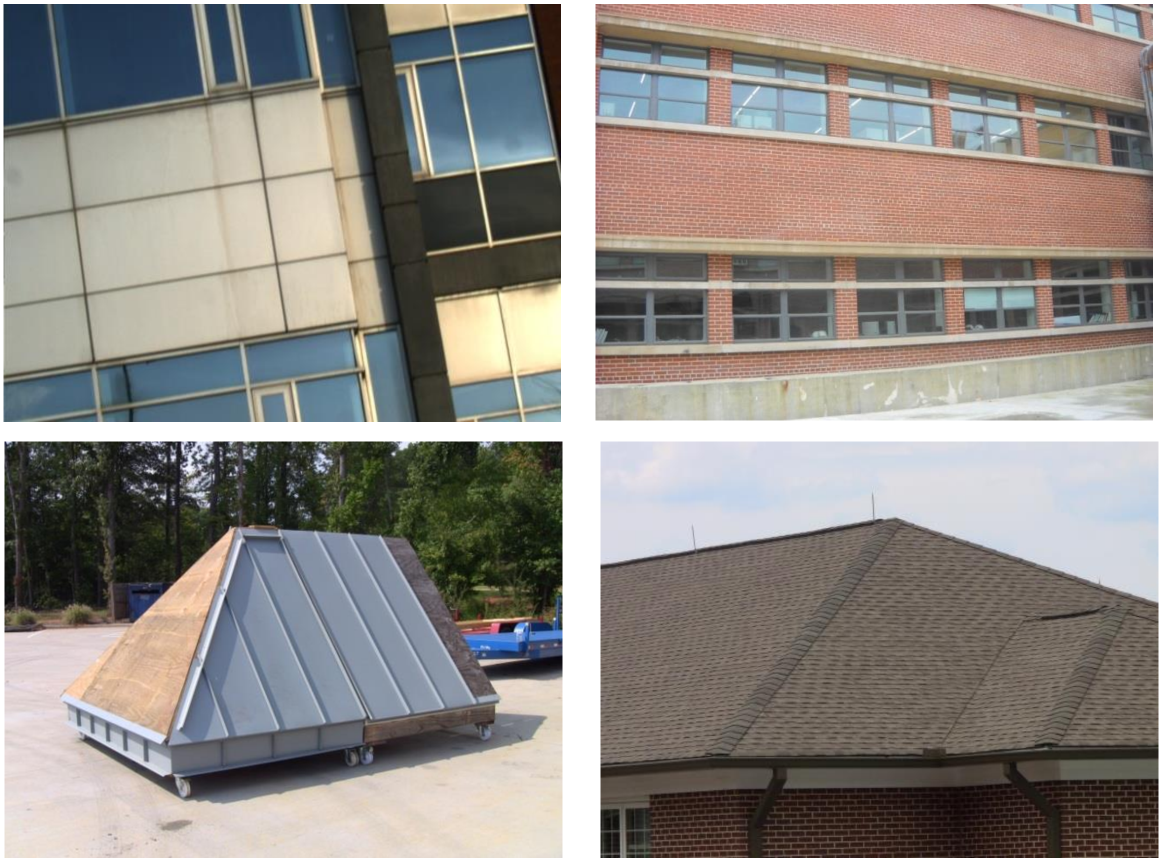


Figure 3.5: Four different environments to test improved affine invariant line descriptor

building façade with poorly-textured aluminum panels, a building façade with well-textured brick pattern, a roof model with surfaces that are a combination of metal panels and plywood, and finally a residential roof structure. The image pairs have different resolutions and are captured with different cameras and lens specifications (e.g., 8, 5, and/or 3 megapixel resolution + 8, 16, and/or 25mm focal length). The data set is categorized into five groups based on the kind of transformation/change that exist between the two views (each group consists of 80 samples): rotation, scale, image blur, illumination, and viewpoint change. It need to be mentioned that other than illumination which has been changed using image editing software programs, all other cases are extracted directly from the collected data with no modification/editing. In all experiments, line segments are detected using the LSD algorithm proposed in (Von Gioi, et al., 2010) which is a parmeterless algorithm and does not any parameter tuning.

Hybrid Structure from Motion: The ultimate goal of the experiments in this section is to evaluate the effect of combining point and line information in the SfM pipeline. This effect has been already tested for visual odometry problems but no such study could be found in the literature for a large-scale 3D reconstruction problem. Two sets of experiments have been designed for this purpose. The first set is performed in a controlled, yet realistic setting that includes a roof model. The model has been constructed with actual materials that are typically used in a sheet metal roofing project. It consists of poorly-textured areas as well as well-textured plywood parts. The repetitive pattern of the sheet metal also provides a challenging environment for feature detection and matching. This could result in noisy feature correspondences which is necessary to test the robustness of the algorithm when wrong matches exist. The second set, on the other hand, is performed in a large-scale environment which is a building façade with brick patterns and three faces. The environment is selected such that an abundant number of point and/or line features could be detected, so there is enough information to run the

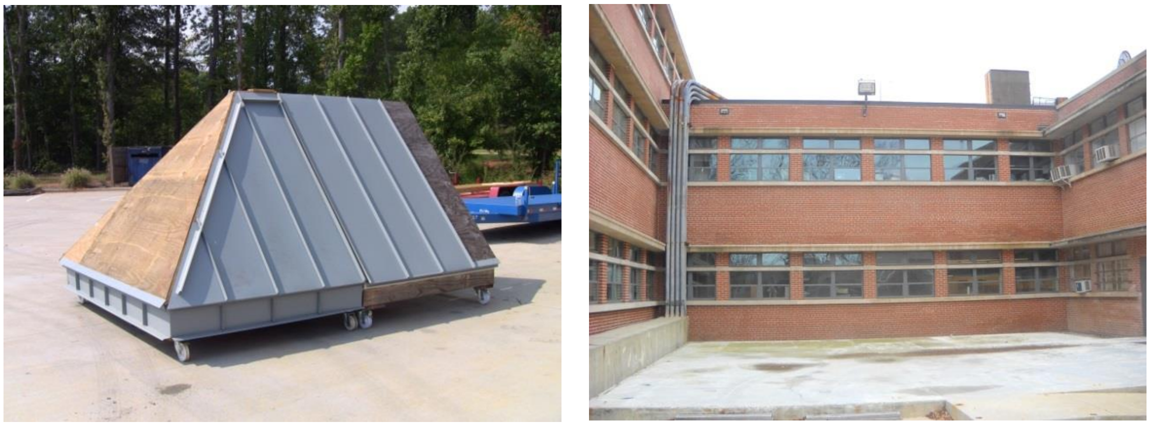


Figure 3.6: Two different environments to test the hybrid SfM algorithm

hybrid SfM with any or a combination of the existing data types. The repetitive patterns on the walls again create a challenging environment for the algorithm to be tested. Snapshots of these two environments are demonstrated in Figure 3.6.

Two Flea2 cameras (2448×2048 pixels) and TAMRON lenses with $f = 25mm$ are used in all of the experiments in this section. The stereo camera system is setup using the cameras and lenses while the baseline distance is selected as 30cm according to the analysis provided in the multi-step stereo camera calibration procedure. The sensor system is calibrated for 5 megapixel resolution using a board with a pattern of 13×14 squares each with a dimension of 60mm. Once the system is calibrated, the scenes are videotaped from such that the distance between the camera and the object of interest changes between 5 to 15 meters. The reason for not selecting a specific distance is to be able to generalize the outcome of the experiments. These are the control variables in our experiments.

Each experiment is repeated for two camera resolutions: 5 and 3 megapixel. It is necessary to mention that although the camera system is calibrated only for 5 megapixel resolution, the same data with some changes is applicable for the experiments with 3 megapixel resolutions. It is known that the distortion coefficients are the same regardless of the camera resolutions used but the focal length should be scaled based on the current resolution. In each experiment, three scenarios are tested: a) only point features are used; b) only line features are used; and c) a combination of point and line features are used. The software prototype is architecture such that the use of point or line features can be controlled using a flag that could be set to true or false. At each one of these scenarios, the following metrics are evaluated: total number of views that have been successfully processed, average reprojection error, and spatial accuracy of the reconstructed scene.

Identify salient planar regions: Two sets of experiments are designed to evaluate the performance of the proposed algorithm for this section. The first set includes identifying salient planar regions in a building façade. The scene is selected to be well-textured and hence the capability to detect an abundant number of point and line features is expected. It also resembles the very common scenario that has been tested in most of the previous studies in the literature (i.e., a scene with three orthogonal vanishing directions). The geometry of the scene allows collecting visual data while there is no occlusion. On the other hand, the second set considers a residential roof structure with more complex geometry and several intersecting planes. The texture of the roof planes are such that reasonable amount of feature points can be detected while there also exist straight edges. In this case, a planar region can be partially occluded depending on the angle of view. These two environments are demonstrated in Figure 3.7.

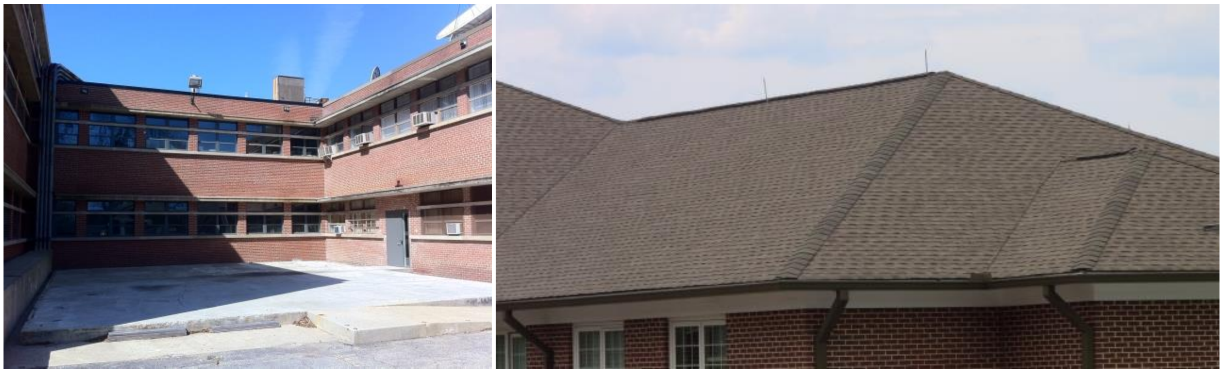


Figure 3.7: A building façade and a residential roof structure

The necessary input data for this section includes a sparse 3D point cloud, 3D line segments, and camera matrices at each view. For the building façade experiment, this data has been generated while doing the experiments related to the hybrid SfM approach. However for the residential roof structure, the data is generated while doing the experiments related to 3D reconstruction of roof structures (these experiments are introduced in the next section). Hence, all the constraints and control variables that are defined in those two sections apply here.

At each experiment, the performance of the planar region detection method is evaluated and compared with (Wang, et al., 2013) which is the most state-of-the-art study in the literature. The performance metrics for the detection accuracy include true positive (TP), false positive (FP), and false negative (FN). TP represents the planar regions that are detected correctly; FP shows the non-planar regions that are detected as a planar region; and FN represents the planar regions that are not detected.

Close-range video-based roof reconstruction: The goal here is to evaluate the overall hypothesis of this research and validate the entire framework. Four separate experiments are designed in this section ranging from very simple to complex scenarios. The first experiment is conducted on the roof model that has been already used in some of the previous experiments. The roof model is an ideal case for proof of concept because it provides a controlled, yet realistic environment that has most of the challenges that one may encounter in a real-life case. The second experiment includes one side of a residential roof structure with a simple rectangular roof plane. The simple geometry in this case is used to show the feasibility of generating a measureable 3D wire-diagram for every roof plane. The roof plane can be videotaped from the ground with 100% visibility (i.e., no occlusion). Moreover, the practical constraints in data collection and processing are those that will be encountered in a real jobsite. The two other experiments include the roof structure of two residential buildings with more complex geometry and several intersecting planes; they represent real-life scenarios for using the proposed framework. Again, the videos are captured from the ground. However in this case, a roof plane can be partially occluded by other planes depending on the angle of view. Sample views of these environments are shown in Figure 3.8.

The same stereo camera system and calibration information that were used in the experiments related to the hybrid bundle adjustment are used. The experiments are repeated for two different camera resolutions (5 and 3 megapixel) in order to evaluate the effect of resolution on the output accuracy. The average distance between the camera set and the roof structures is kept at roughly 20m during the data collection process. The camera motion is also smooth to create the minimum motion blur.

The metrics that are used in this evaluation are the following: completeness of the wire-diagram (whether all boundaries of a roof plane are reconstructed or not), Euclidean

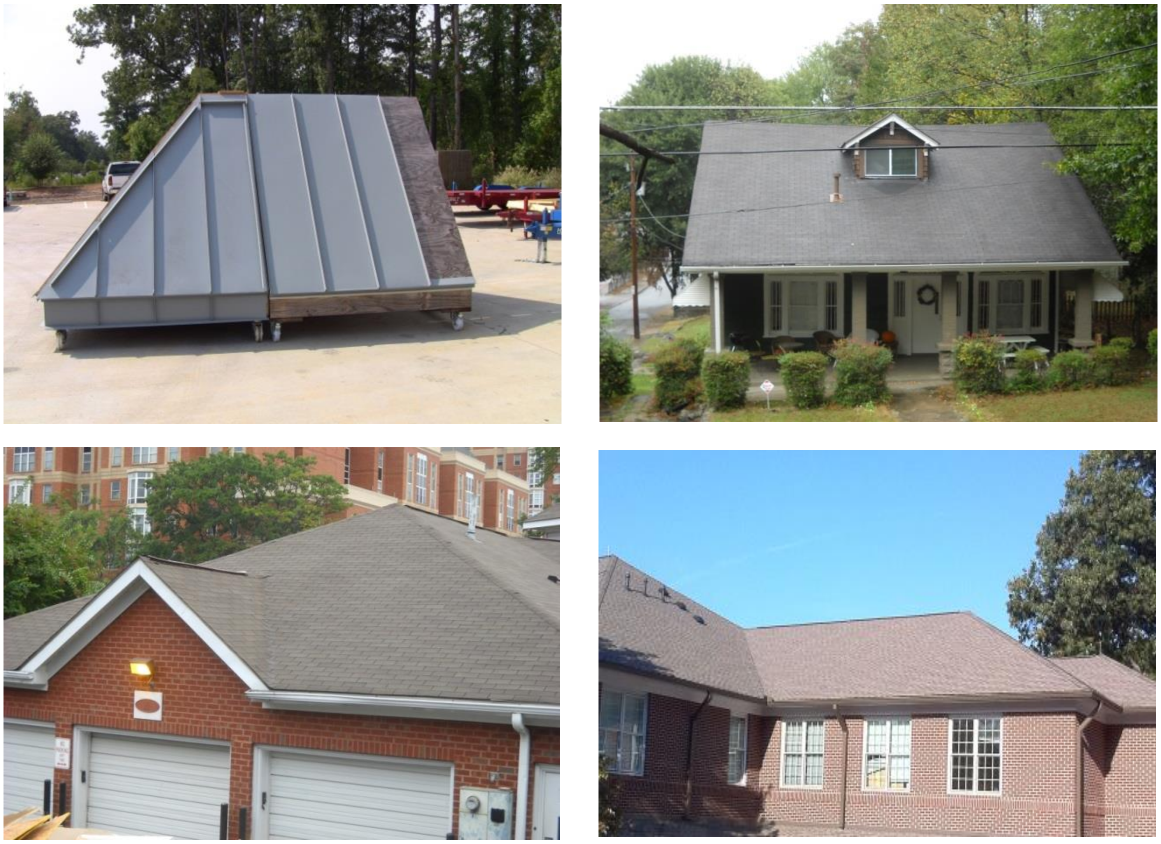


Figure 3.8: Four different environments to test the roof reconstruction framework

accuracy of the end-to-end dimensions of the edges (the absolute value of the difference between corresponding measurements in the 3D wire-diagrams and ground truth data is reported as the error), and the number of manual data that is needed to fix a missing edge. The mean and standard deviation of errors are finally used to calculate the 95 percentile error.

CHAPTER 4

VALIDATION

4.1. Multi-Step Stereo Camera Calibration

Prior to perform the designed experiments for this section, a preliminary study was implemented to assess the amount of uncertainty that may exist in estimating different calibration parameters. The existing stereo camera set (two Flea2 cameras with a resolution of 2448×2048 pixels as well as two fixed focal length lenses with $f = 25mm$ and a baseline distance of $b = 30cm$) was used. In this experiment, four different scenarios for D (i.e., distance between the camera set and calibration patter) were used including $D = 10m$, $D = 20m$, $D = 30m$, and $D = 10 - 30m$. As can be seen, in the first three cases, D was kept constant at predefined values and in the last case, D was varying from 10 to 30 meters. Once the video streams were collected, the camera calibration software was run 5 times for each case. The reason for multiple runs of the software for the same data was to study whether they could all result in the same calibration parameters or not. Table 4.1 illustrates the results of this experiment for the first and last cases (other experiments followed the same pattern and hence were not presented due to the limited space). It can be inferred from Table 4.1 that there is a significant variation for estimated intrinsic camera parameters (i.e., focal length, principal point, and distortion coefficients) even for experiments with similar D values. This may happen because of the complex structure of the lens or slight changes in the zoom/focus while collecting data. However, this variation is almost negligible for estimated extrinsic parameters (i.e., rotation and translation). This may indicate that intrinsic camera parameters are somehow needed to be included in the optimization processes in the SfM pipeline so that the values with maximum likelihood could be achieved.

Table 4.1: Estimated camera calibration parameters in different experiments

		D = 10m					D = 10-30m				
		1	2	3	4	5	1	2	3	4	5
Left Camera Parameters	f_x	5638	5620	5609	5680	5574	5563	5589	5542	5615	5601
	f_y	5614	5604	5583	5707	5546	5592	5556	5539	5563	5641
	C_x	478	495	515	483	451	511	537	518	494	502
	C_y	421	435	471	449	406	383	723	405	394	463
	k_1	-0.11	-0.08	-0.12	-0.07	-0.08	-0.12	-0.1	-0.09	-0.14	-0.08
	k_2	-4.67	-3.72	-5.03	-5.14	-4.75	-7.63	-5.43	-5.89	-7.14	-6.54
	p_1	-0.003	-0.003	-0.002	-0.004	-0.002	-0.004	-0.003	-0.004	-0.004	-0.002
	p_2	-0.007	-0.005	-0.005	-0.006	-0.004	-0.008	-0.008	-0.005	-0.007	-0.008
	k_3	-76	-64	-86	-67	-92	-83	-55	-68	-76	-59
Right Camera Parameters	f_x	5589	5602	5574	5561	5469	5614	5659	5635	5587	5605
	f_y	5588	5611	5569	5573	5504	5607	5640	5608	5572	5598
	C_x	498	479	462	505	481	540	564	503	485	528
	C_y	438	452	401	468	459	424	473	416	449	485
	k_1	-0.07	-0.07	-0.06	-0.09	-0.06	-0.05	-0.07	-0.1	-0.05	-0.08
	k_2	0.94	1.4	0.72	0.83	1.04	-2.7	-1.9	-2.12	-2.96	-2.54
	p_1	0	0.01	0.02	0	0.15	-0.003	-0.001	0	-0.003	-0.002
	p_2	-0.006	-0.008	-0.008	-0.007	-0.005	-0.008	-0.006	-0.005	-0.008	-0.01
	k_3	-95	-104	-81	-92	-89	-107	-102	-98	-121	-115
Rotation	R_1	0.016	0.016	0.015	0.017	0.015	0.015	0.016	0.016	0.015	0.016
	R_2	0	-0.001	-0.001	0.0005	0	0	-0.001	0	0.0002	0
	R_3	0.0101	0.0105	0.0103	0.0101	0.0109	0.0104	0.0106	0.0105	0.0105	0.102
Translation	T_1	-309	-307	-309	-308	-308	-310	-309	-307	-309	-308
	T_2	1.6	2.1	1.7	1.5	1.9	1.9	1.7	1.6	1.6	1.5
	T_3	-40	-43	-41	-41	-38	-44	-42	-45	-40	-43

The designed sets of experiments were then performed according to the specified details. The previously mentioned camera system and calibration board were used to capture the six sets of required data for calibration from the building with brick pattern facade. Using the developed automatic calibration software, 50 stereo frames were extracted in each case (i.e., $D = 5m$, $D = 10m$, $D = 15m$, $D = 20m$, and $D = 25m$) and the calibration parameters were calculated. Then, another set of stereo video streams were captured from the façade while the distance between the camera system and the façade was changing in the range of $5m \le D \le 25m$. Figures 4.1, 4.2, and 4.3 demonstrate some of the intermediate results.

For evaluating the first performance metric (i.e., spatial accuracy of the initial estimation for 3D coordinates of points with different range values), stereo frames corresponding to $D = \{5,10,15,20,25m\}$ were extracted from the façade video and 3D coordinates of feature points were calculated using the sets of estimated calibration parameters. Spatial distance between pairs of 3D feature points was then compared to the ground truth data. Table 4.2 illustrates the average error at each scenario (sample size of 50). The results indicate that a more accurate initial estimation can be done for a point at a range of Z using the calibration parameters that correspond to $D \approx Z$; this supports the hypothesis in this research.

Table 4.2: Average spatial distance error for different calibration scenarios

Calibration Scenario	**Average spatial distance error (cm)**				
	$Z \approx 5m$	$Z \approx 10m$	$Z \approx 15m$	$Z \approx 20m$	$Z \approx 25m$
$D = 5m$	**±2.6**	±4.8	±9.8	±17.7	±23.8
$D = 10m$	±2.9	**±4.3**	±8.5	±15.9	±20.2
$D = 15m$	±3.5	±5	**±6.3**	±14.5	±19.6
$D = 20m$	±4.2	±6.1	±9	**±11.3**	±18.4
$D = 25m$	±5.1	±7.5	±12.7	±15.2	**±15**
$5m \le D \le 25m$	±3.2	±4.9	±10.1	±15.6	±19.2

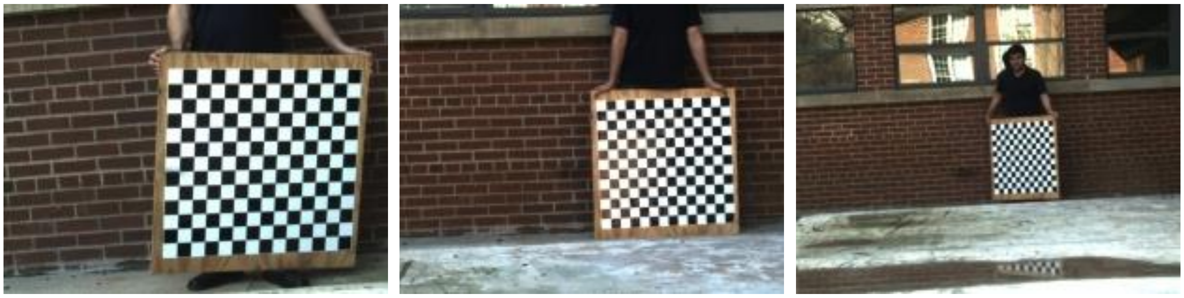


Figure 4.1: Video frames for calibration at *D=10m*, *D=20m*, and *D=30m*

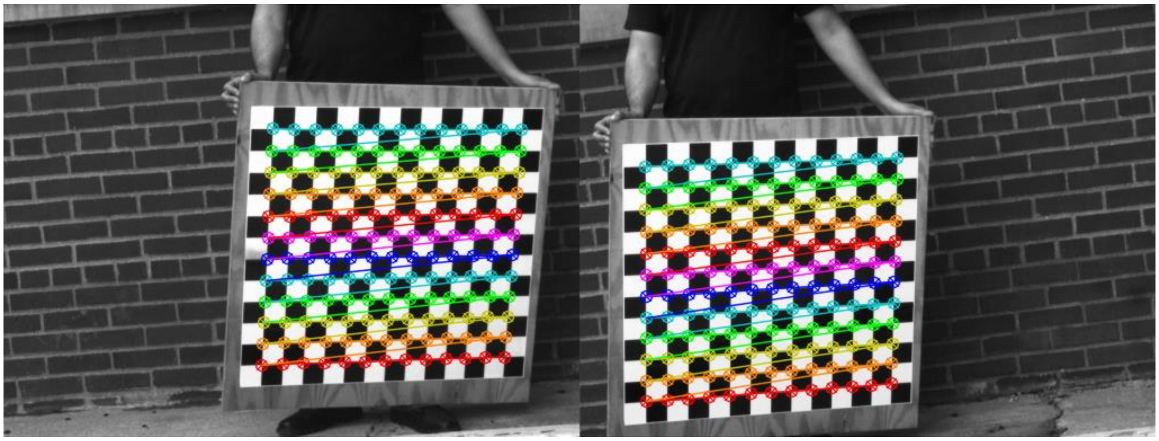


Figure 4.2: Automatically detected and matched calibration board corners

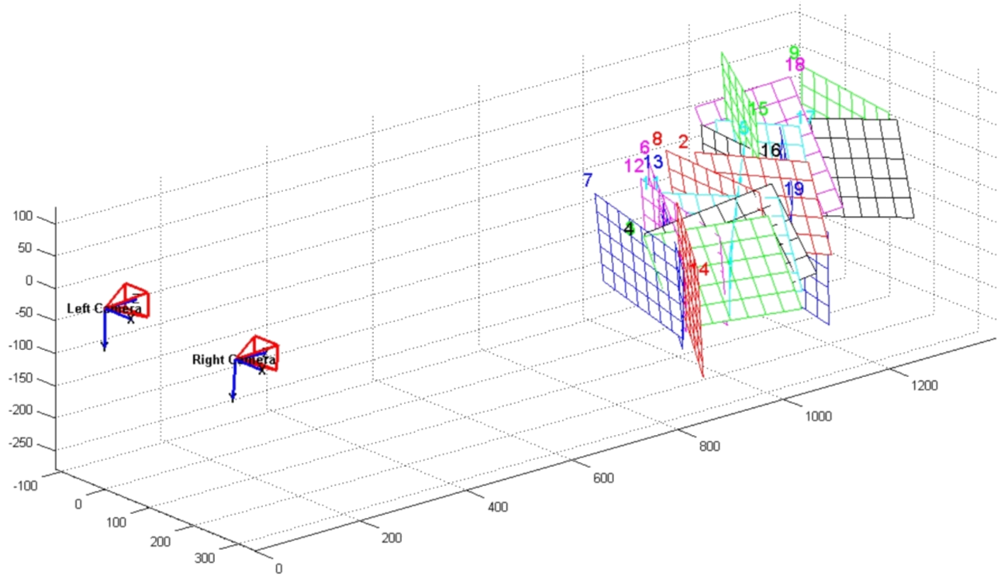


Figure 4.3: Visualization of the extrinsic parameters

To evaluate the second performance metric (i.e., spatial accuracy of a dense 3D point cloud), two dense 3D point clouds of the façade were generated using all the frames in the façade video. The key-frame selection method proposed in (Rashidi, et al., 2013) was use to extract frames that have minimum motion blur and appropriate number of feature points while the camera motion between two consecutive key-frames is larger than a minimum value. The same thresholds that have been proposed in (Rashidi, et al., 2013) were used for this purpose. In addition, the patch-based multi-view stereo software which is based on (Furukawa & Ponce, 2010) and available online, was used to generate the dense 3D reconstructions; the output of the proposed stereo reconstruction algorithm (i.e., camera location and projection matrices for each view and a sparse 3D point cloud) was used as the input data to this software.

The results of the conventional camera calibration procedure were used to generate the first point cloud (Figure 4.4). The second one was generated using the calibration parameters acquired from the proposed procedure (Figure 4.5). 50 pairs of points were selected randomly at each case and the spatial distance was compared to the ground truth data. Total station surveying was used to acquire the ground truth data. The average error in the first point cloud was ±12.6cm while this average error was ±9.5cm in the second point cloud. This shows an average of 3.1cm (25%) improvement in the accuracy of results because of using the proposed multi-step stereo camera calibration procedure. The relative improved accuracy can also be visually seen by comparing the point clouds in Figures 4.4 and 4.5. The second point cloud is sharper in planar areas. Again this supports the presented hypothesis in this research. It is necessary to mention that this accuracy may be further improved by modifying the multi-view geometry process which is out of the scope of this research.

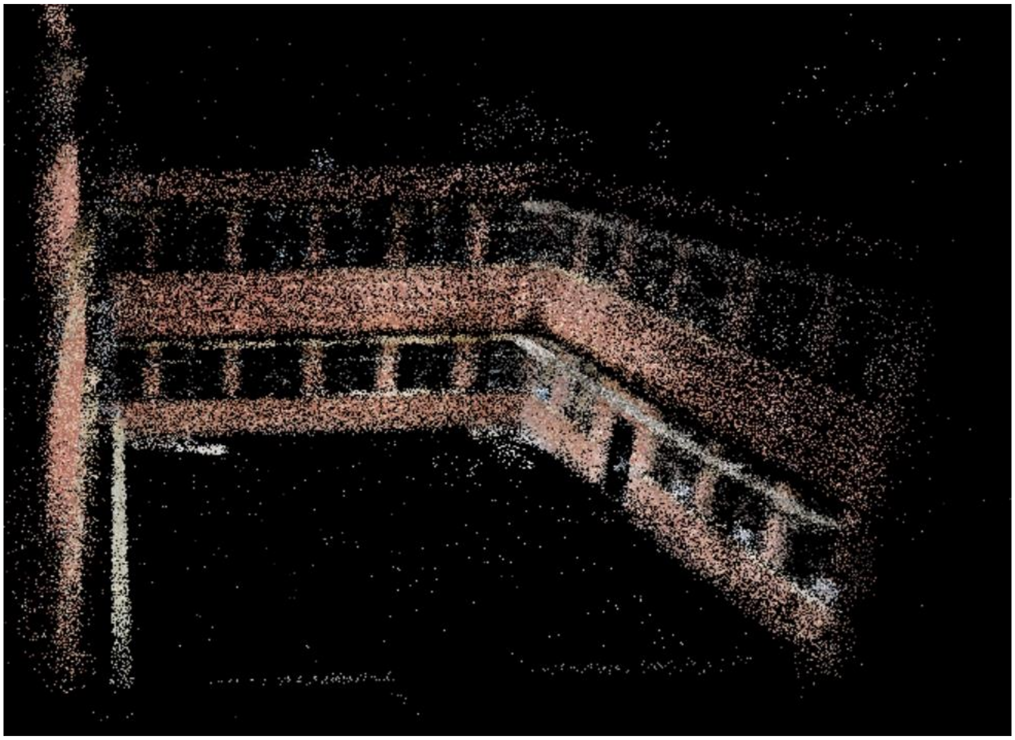


Figure 4.4: Dense 3D point cloud using the conventional stereo camera calibration

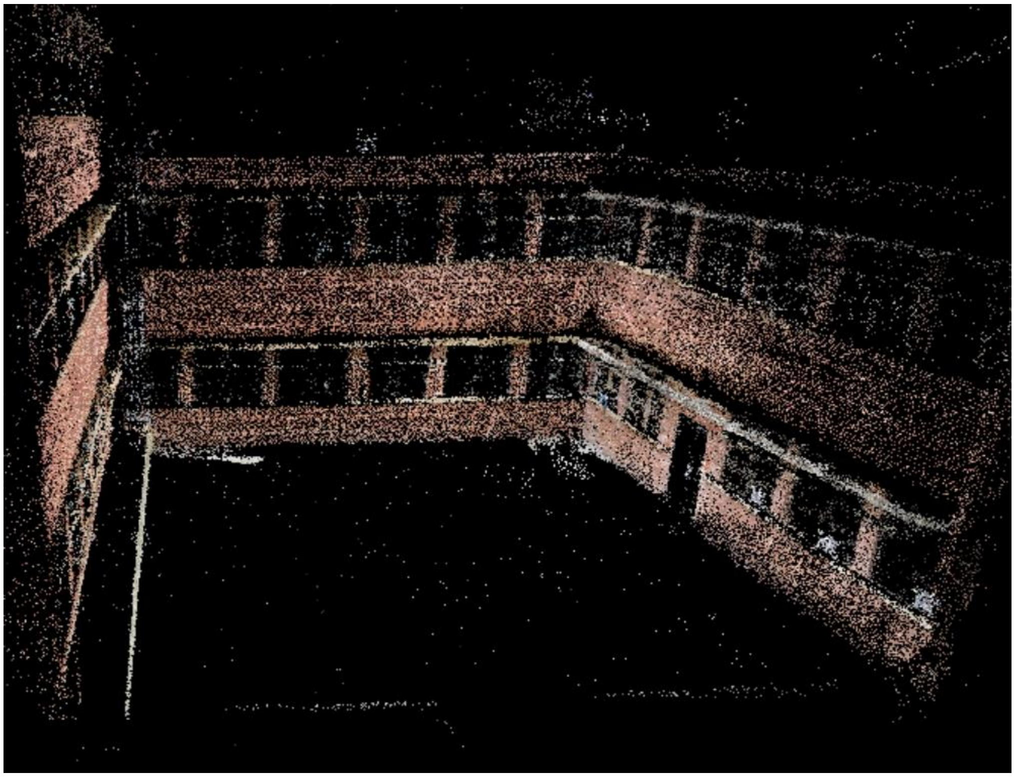


Figure 4.5: Dense 3D point cloud using the proposed multi-step stereo camera calibration

4.2. Improved Affine Invariant Line Descriptors

Stereo video streams and photographs were captured from the four different environments introduced in the design of experiment section with different camera and lens configurations. Image/frame pairs were then extracted from the data such that they cover a wide range of changes such as rotation, scale, image blur, illumination, and viewpoint changes. Some of these samples are illustrated in Figure 4.6. Scale-space representations of the images were first generated and then the LSD method (Von Gioi, et al., 2010) was applied to detect line features at the local extrema. For each detected line segment, two multi-dimensional descriptor vectors were found using the original MSLD algorithm (Wang, et al., 2009) and the proposed improved version. Line segments were then matched by comparing those descriptors.

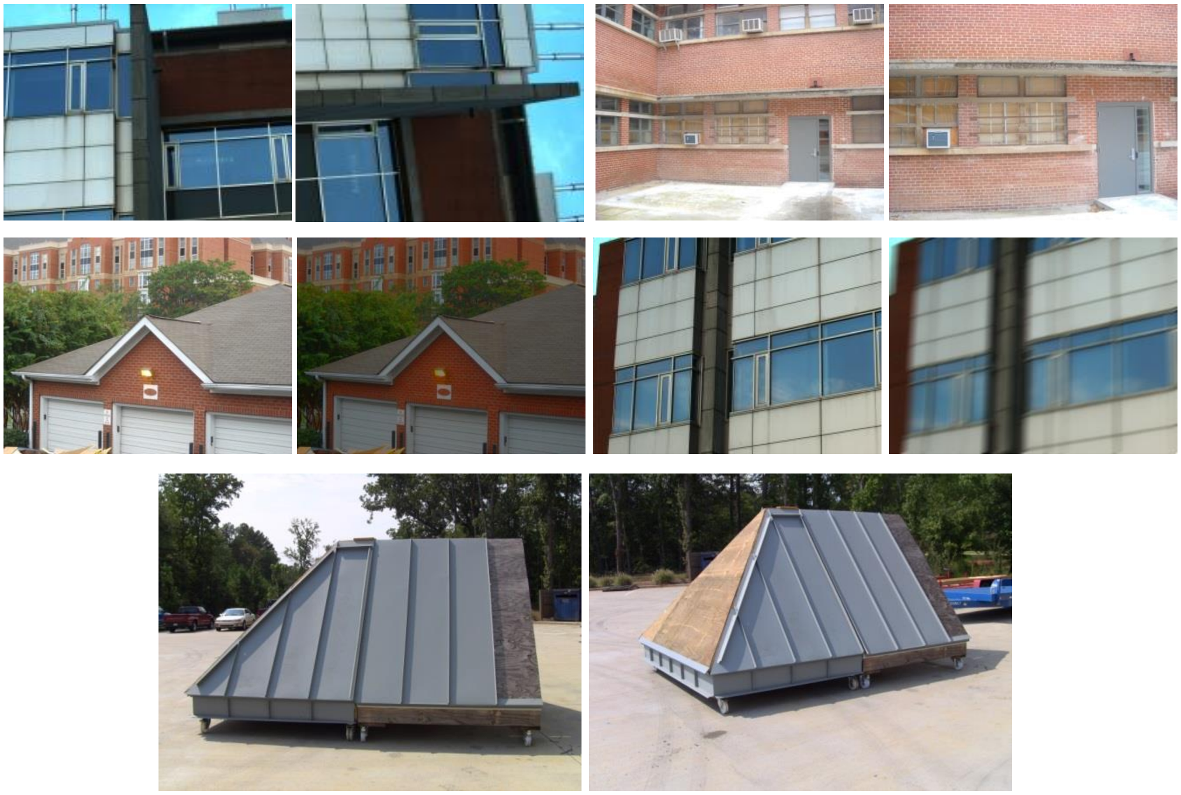


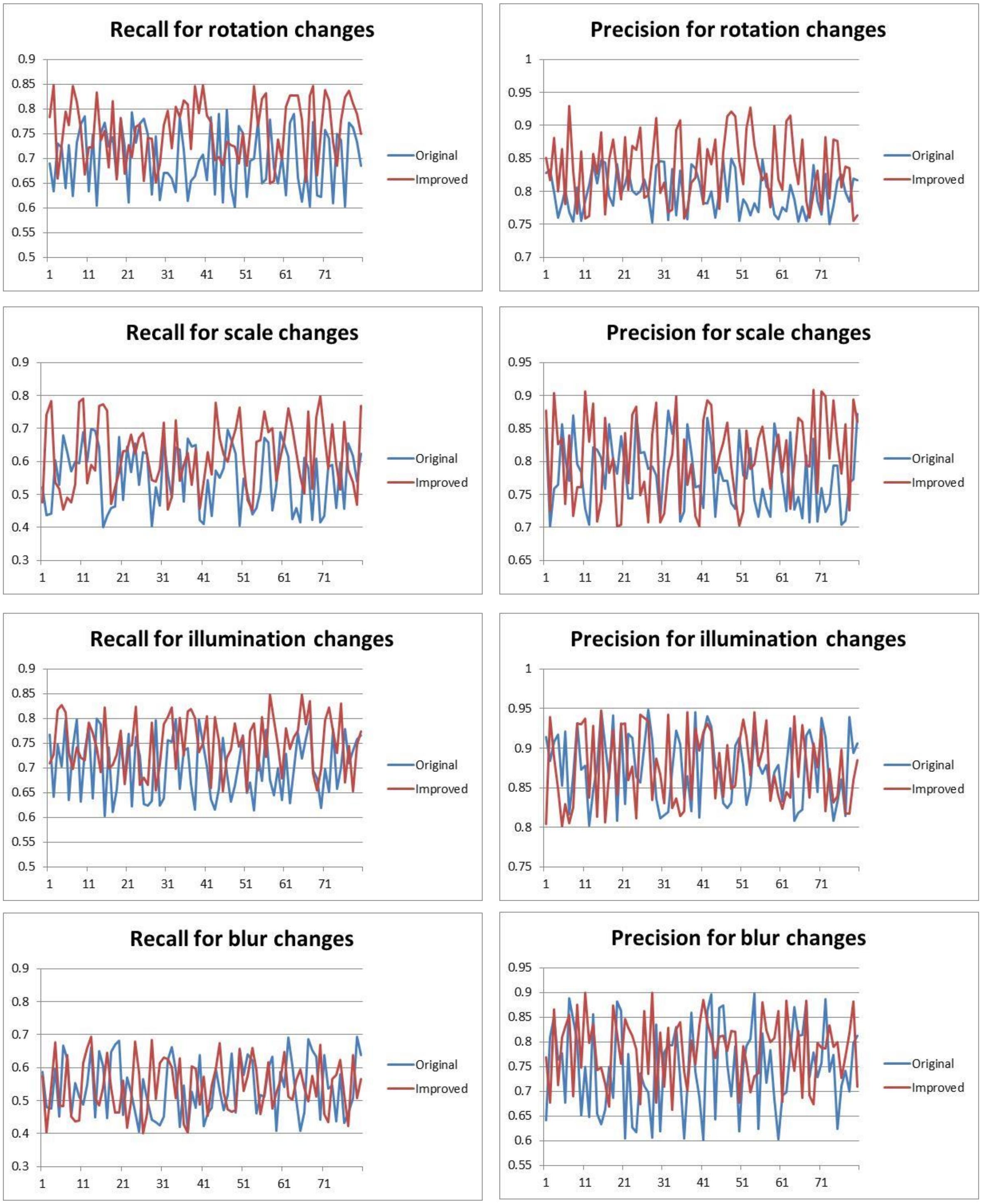
Figure 4.6: Sample image pairs for rotation, scale, illumination, blur, and viewpoint changes

The results of this comparison are presented in Figure 4.7 according to two metrics: recall and precision. Recall is the ratio of correct matches to total number of lines that are visible in both views and precision is the ratio of correct matches to total number of matches.

As can be inferred from Figure 4.7, the improved MSLD performs better in terms of rotation, scale, and viewpoint changes. On the other hand, the performance of the original and improved MSLD algorithms is more or less the same in illumination and blur changes. Table 4.3 numerically demonstrates these comparisons by presenting the average recall and precision at each scenario. In general, the improved MSLD outperforms the original algorithm in most cases (+4% increase in recall and +5% increase in precision) which is due to the use of assigning dynamic pixel support regions and converting the regions to the canonical form.

Table 4.3: Average recall and precision for original and improved MSLD

Change Scenario	**Recall**		**Precision**	
	Original	**Improved**	**Original**	**Improved**
Rotation	0.69	0.74	0.82	0.88
Scale	0.57	0.62	0.79	0.84
Illumination	0.72	0.74	0.89	0.90
Blur	0.56	0.56	0.77	0.80
Viewpoint	0.60	0.65	0.72	0.78



Recall for rotation changes

Precision for rotation changes

Recall for scale changes

Precision for scale changes

Recall for illumination changes

Precision for illumination changes

Recall for blur changes

Precision for blur changes

Figure 4.7: Recall and precision of the original and improved MSLD in different scenarios

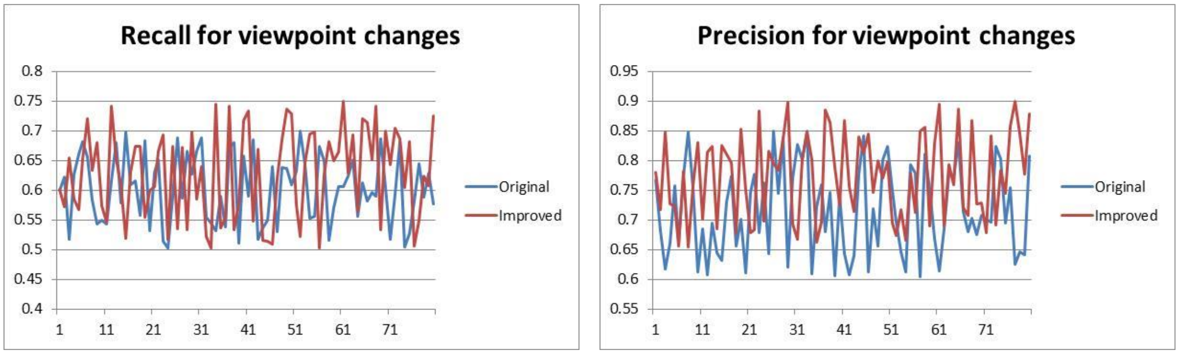


Figure 4.7 cont.: Recall and precision of the original and improved MSLD in different scenarios

4.3. Hybrid Structure from Motion

This section presents the outcome of several experiments aiming to validate the proposed hybrid SfM algorithm. For a comprehensive evaluation, two environments with different characteristics were selected and stereo video streams were captured. The experiments were repeated for two different camera resolutions and the scenes were reconstructed three times using points, lines, and a combination of points and lines. The following paragraphs present the detailed analysis of these experiments.

Roof model: A roof model that is covered with actual roofing materials was the subject in the first experiment. 72 stereo frames were extracted from the captured video data using the key-frame selection algorithm proposed in (Rashidi, et al., 2013). These frames had a resolution of 2448×2048 (5 megapixels). The same stereo frames were down-sampled using an image editing program to a resolution of 1900×1600 (3 megapixels). These two sets of frames allow analyzing the effect of image resolution of the output. Once the input data was ready, the "line flag" in our hybrid SfM software prototype was turned off and the data was only processed using the point features. Figure 4.8 illustrates a snapshot of the results for the 5 and 3 megapixel resolutions. Table 4.4 also compares the results of the scenarios according to different performance metrics. As can be seen, most of the reconstructed points in both scenarios belong to the background trees and the face of the model that is covered with plywood.

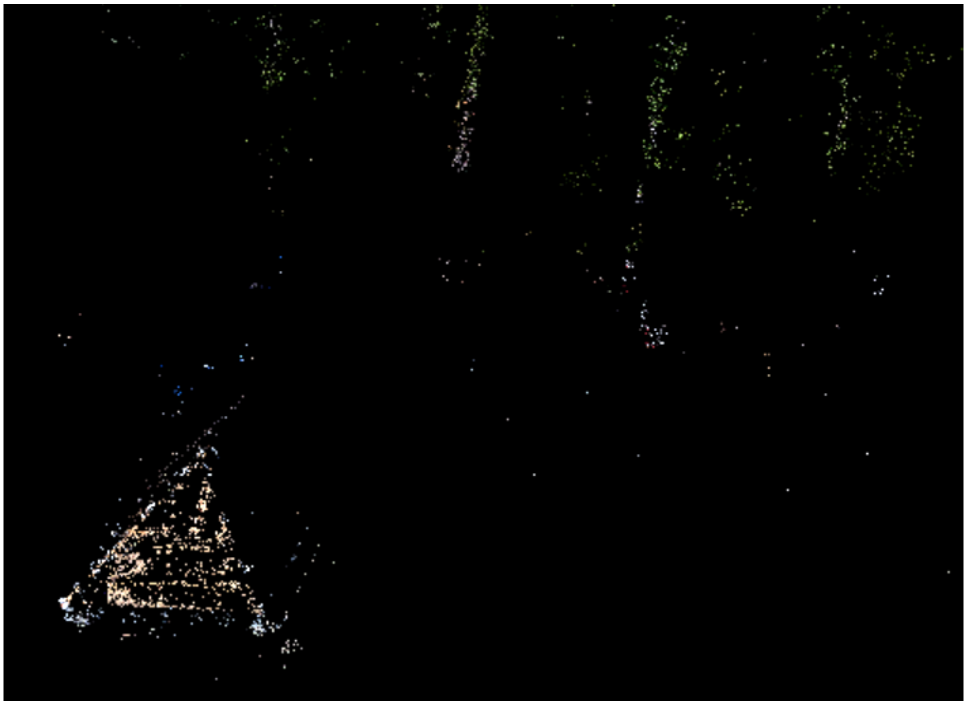


Figure 4.8 (a): Point-based 3D reconstruction of the roof model (resolution = 3MP)

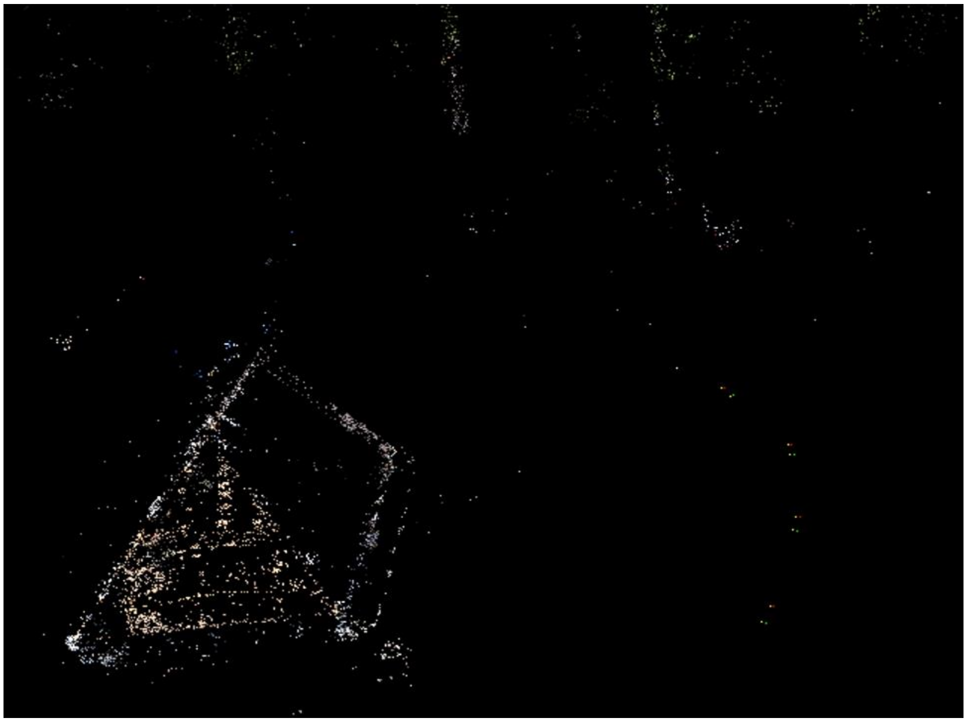
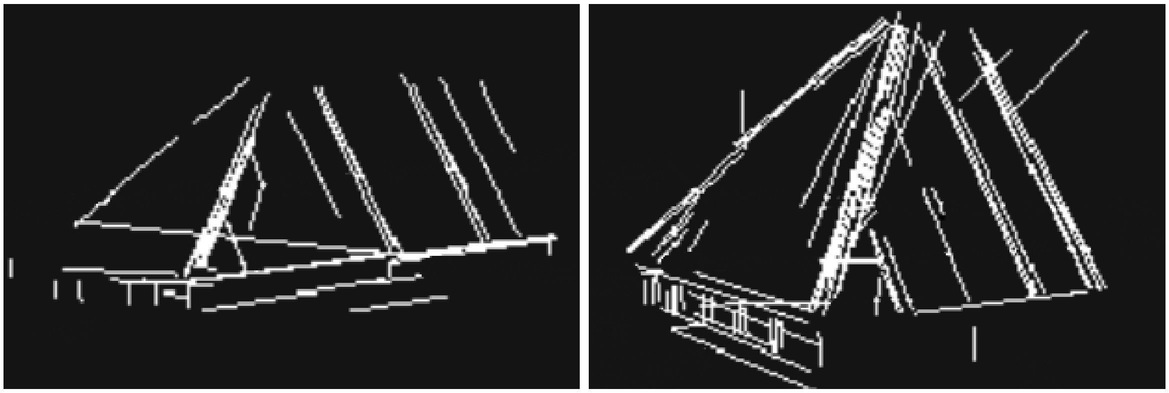


Figure 4.8 (b): Point-based 3D reconstruction of the roof model (resolution = 5MP)

Table 4.4: Performance evaluation for point-based 3D reconstruction of the roof model

Resolution	Focal Length (mm)	No. of reconstructed views	Total No. of points	Avg. reprojection error (pixel)	Avg. spatial distance error (cm)
2448×2048	25	69	6974	0.012	4.23
1900×1600	19.5	57	5235	0.027	5.74

In the next step, the "point flag" was turned off and the "line flag" was turned on. The same input data (i.e., 72 stereo video frames) was then used to achieve a line-based 3D reconstruction. The results are illustrated in Figure 4.9 and a numerical comparison is presented in Table 4.5. None of the two reconstructions were completely successful in this case. The main reasons could be the following: a) at least 13 line triplets are needed to calculate a trifocal tensor while the same can be done with 7 point triplets; b) degeneracy for lines is far more severe than the degeneracy for points; c) lines provide less mathematical constraints on camera pose and hence the probability of failure is higher in this case; and d) two separate lines that are along each other can only provide the same mathematical constraints because the equation of infinite lines are used in the reconstruction process.



3 MP **5 MP**

Figure 4.9: Line-based 3D reconstruction of the roof model

Table 4.5: Performance evaluation for line-based 3D reconstruction of the roof model

Resolution	Focal Length (mm)	No. of reconstructed views	Total No. of lines	Avg. reprojection error (pixel)	Avg. spatial distance error (cm)
2448×2048	25	33	94	4.18	13.7
1900×1600	19.5	31	76	7.25	19.5

Finally, the combination of point and line data was used to generate a sparse 3D point cloud and a 3D line set from the roof model. Figure 4.10 demonstrates the reconstruction outcome for the two resolutions and the numerical comparison is made in Table 4.6. As can be seen, the robustness of the algorithm has increased due to the simultaneous use of points and lines. Higher number views could be processed during the reconstruction phase which means more robustness in estimating the camera motion in the environment. The accuracy of the reconstruction is more or less the same as the accuracy of the point-based case. In comparison, the accuracy level for reconstructed lines is less than the accuracy of points. Another significant advantage of this reconstruction is the clear visual perception of the object that is due to the use of line in addition to points.

Table 4.6: Performance evaluation for hybrid 3D reconstruction of the roof model

Resolution	Focal Length (mm)	No. of reconstructed views	Total No. of points	Total No. of lines	Avg. reprojection error (pixel)		Avg. spatial distance error (cm)	
					points	lines	points	lines
2448×2048	25	70	6851	126	0.013	0.59	4.26	7.4
1900×1600	19.5	64	5347	98	0.029	0.84	5.81	9.1

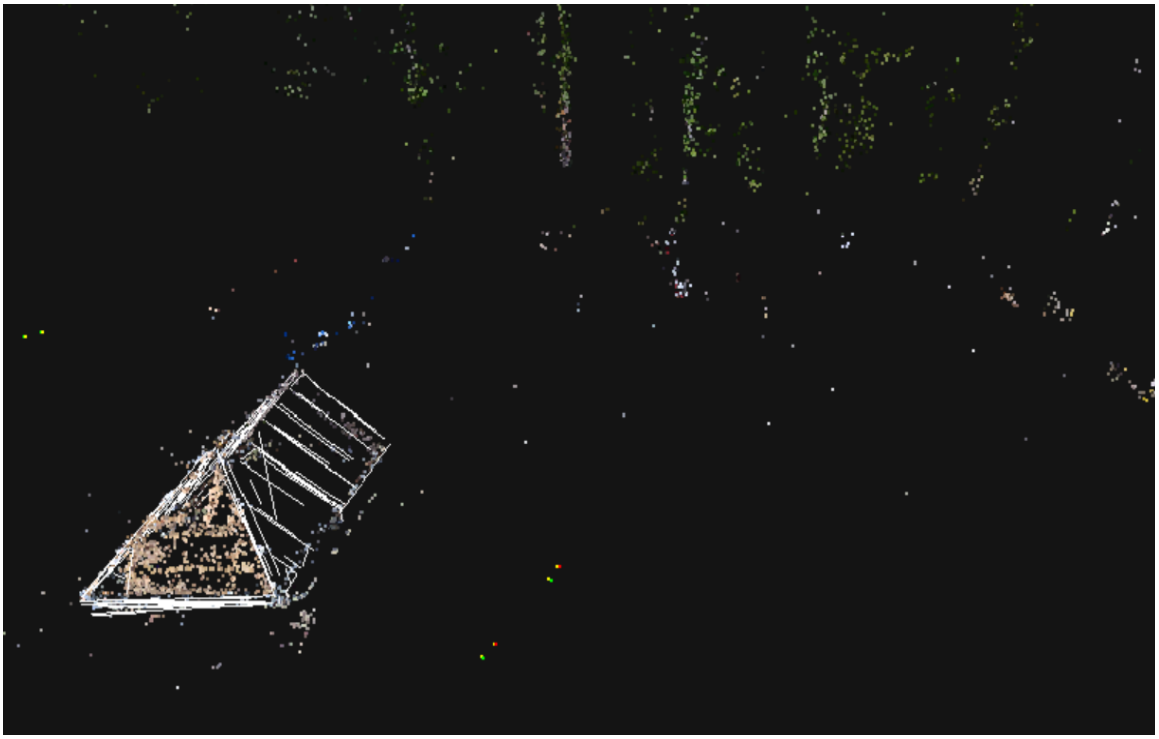


Figure 4.10 (a): Hybrid 3D reconstruction of the roof model (resolution = 3MP)

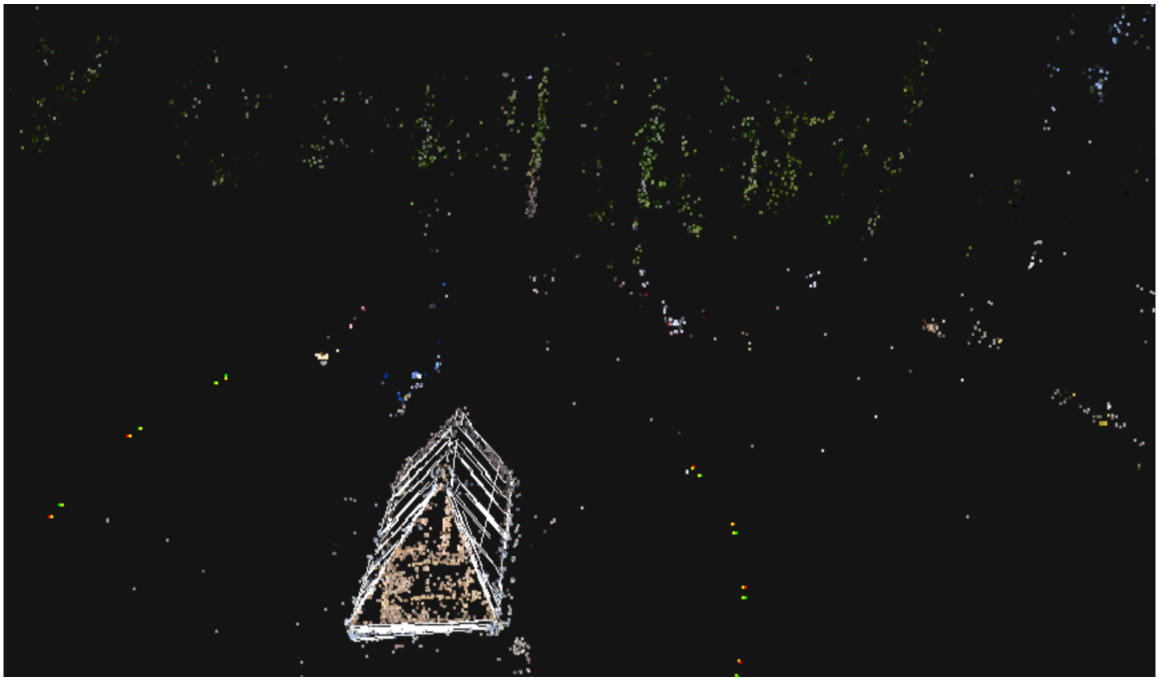


Figure 4.10 (b): Hybrid 3D reconstruction of the roof model (resolution = 5MP)

Building façade: The subject of the next experiment is a building façade with brick pattern. The goal here is to test the algorithm on a larger scale object and analyze the outcome. Same as the previous case, all experiments were repeated for two different image resolutions. 427 stereo frames were extracted and processed to produce sparse 3D point clouds and line sets. The resolution of the original video streams was 5 megapixel and the same dataset was also down-sampled to 3 megapixel in order to be able to study the resolution effect.

Initially, the point-based 3D reconstruction experiment was performed. Due to the visual characteristics of the façade surfaces, many distinctive point features could be detected, matched, and reconstructed. This is different than the case presented for the roof model, as no feature point could be detected on sheet metal area. The very high number of feature points in the façade case allowed a more accurate reconstruction which can be verified both visually and numerically. It needs to be reminded that most of the reconstructed points in the roof model case were from the background trees and the parts with plywood texture. The output of the point-based reconstruction is illustrated in Figure 4.11 and the numerical comparisons are presented in Table 4.7.

Table 4.7: Performance evaluation for point-based 3D reconstruction of the façade

Resolution	Focal Length (mm)	No. of reconstructed views	Total No. of points	Avg. reprojection error (pixel)	Avg. spatial distance error (cm)
2448×2048	25	418	13002	0.009	3.65
1900×1600	19.5	407	11809	0.009	4.27

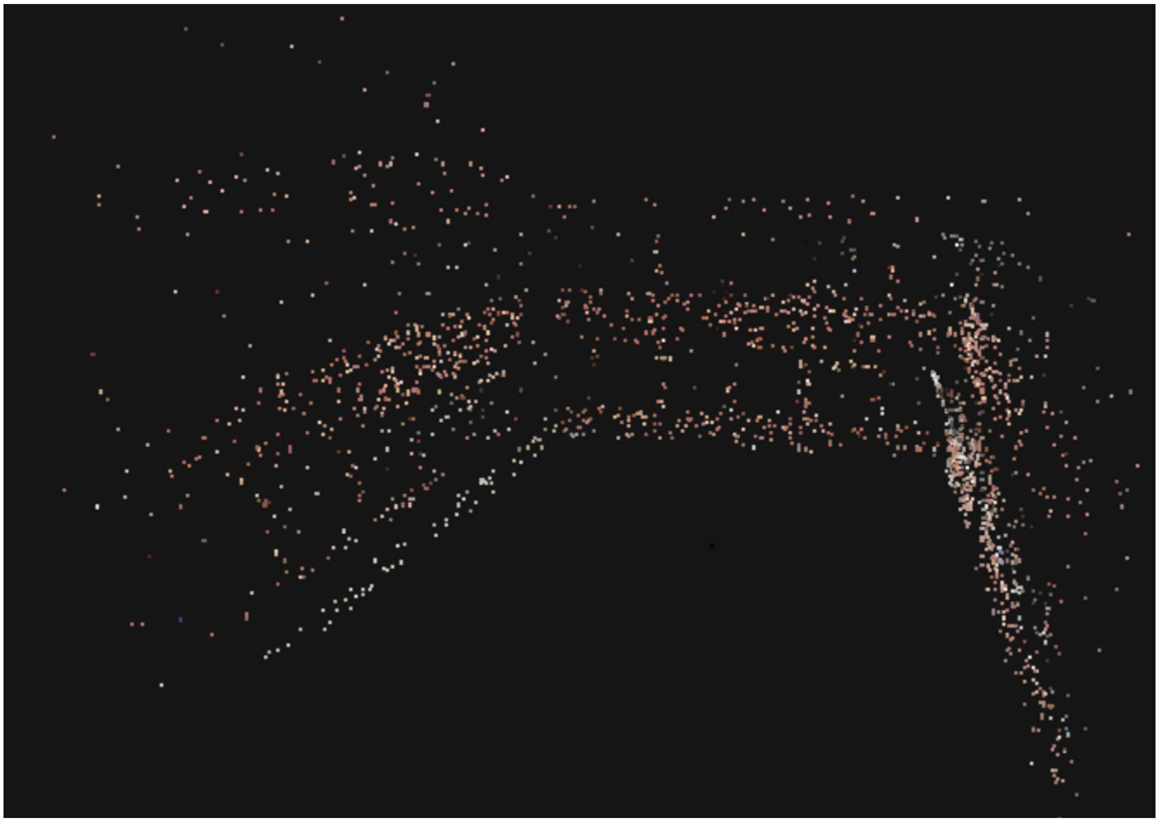


Figure 4.11 (a): Point-based 3D reconstruction of the façade (resolution = 3MP)

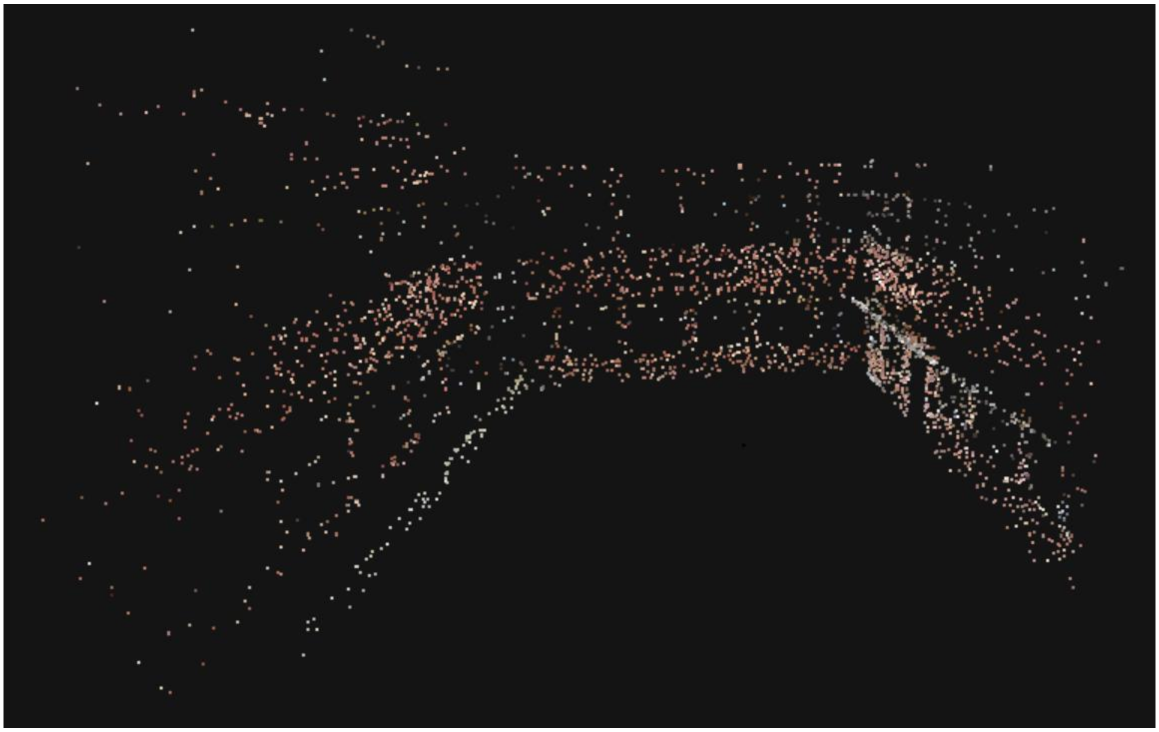
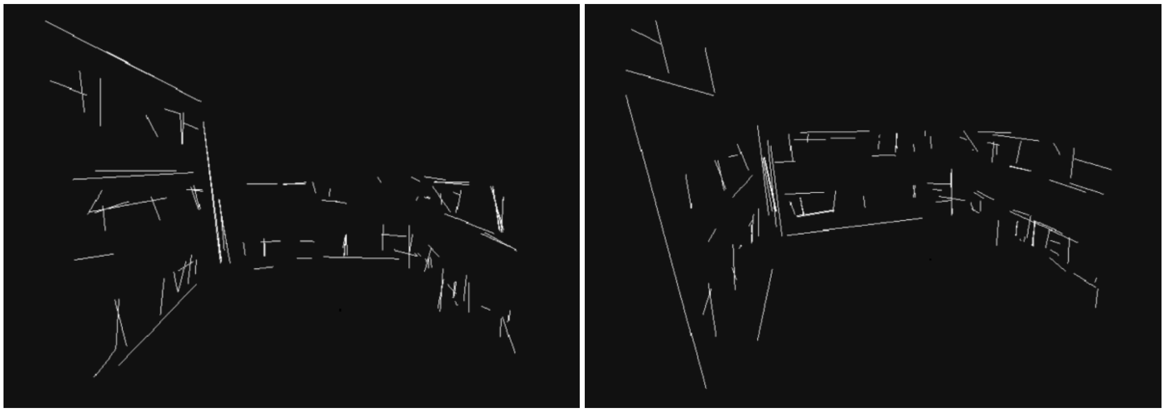


Figure 4.11 (b): Point-based 3D reconstruction of the façade (resolution = 5MP)

A line-based 3D reconstruction was performed on the dataset from the façade in the next step. Compared to the roof model case, the algorithm produced a better reconstruction mainly due to the prevalence of point features which helped in the line matching process. Figure 4.12 shows the results for 5 and 3 megapixel resolutions. Table 4.8 provides the numerical analysis of the results.



3 MP **5 MP**

Figure 4.12: Line-based 3D reconstruction of the façade

Table 4.8: Performance evaluation for line-based 3D reconstruction of the façade

Resolution	Focal Length (mm)	No. of reconstructed views	Total No. of lines	Avg. reprojection error (pixel)	Avg. spatial distance error (cm)
2448×2048	25	338	164	3.85	9.2
1900×1600	19.5	329	153	4.11	11.3

The last step was to perform a hybrid 3D reconstruction of the façade using the complete package. Figure 4.13 and Table 4.9 demonstrate the reconstruction results and analysis. The analyses from the two experiments (i.e., roof model and façade) indicate that the hybrid reconstruction is a more robust approach that is capable of producing more accurate representation of the underlying geometry. Line degeneracy was also an important factor in the reconstruction. Lines in 3D space lying on the epipolar plane could not be reconstructed using the two views because they intersect the camera baseline. Therefore, in case of estimating the measurement error, lines that are close to intersecting the baseline can be poorly localized in the reconstruction. In general, the degeneracy for lines is far more severe than points. There is three-parameter family of lines which cannot be recovered: one parameter for the position of the baseline, and the other two for the start of lines through each point on the baseline (Hartley & Zisserman, 2004).

Table 4.9: Performance evaluation for hybrid 3D reconstruction of the façade

Resolution	Focal Length (mm)	No. of reconstructed views	Total No. of points	Total No. of lines	Avg. reprojection error (pixel)		Avg. spatial distance error (cm)	
					points	lines	points	lines
2448×2048	25	427	13509	361	0.012	1.07	4.74	6.7
1900×1600	19.5	421	12082	322	0.014	1.35	5.03	7.5

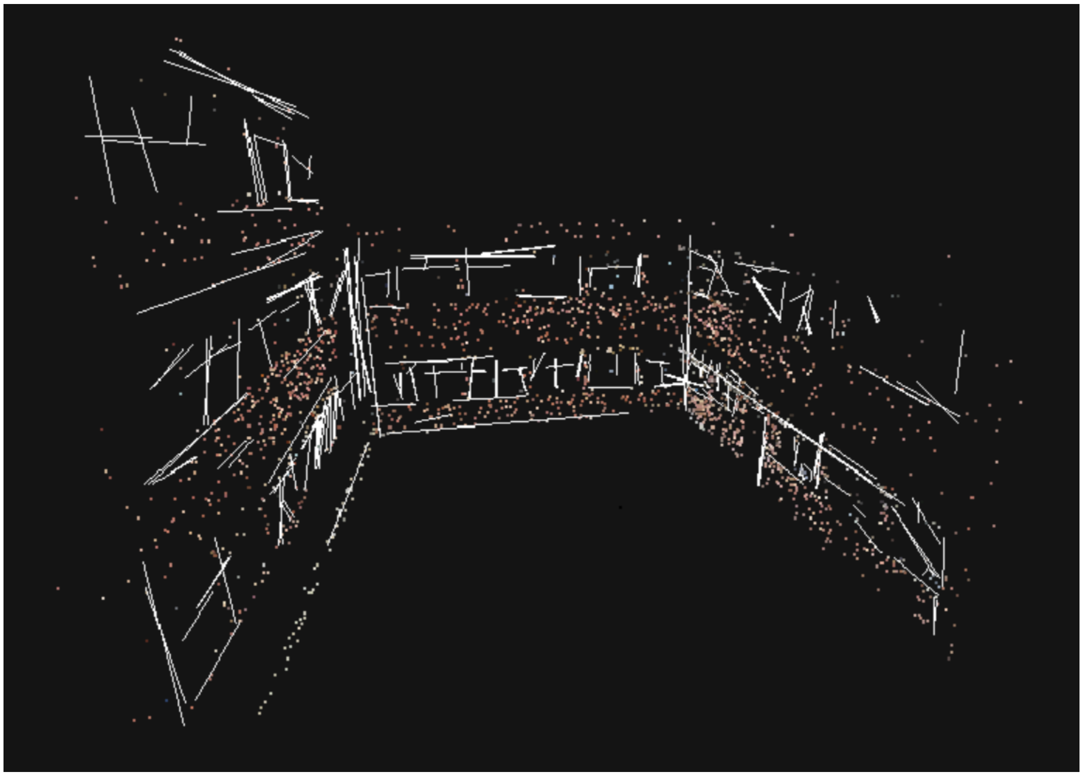


Figure 4.13 (a): Hybrid 3D reconstruction of the façade (resolution = 3MP)

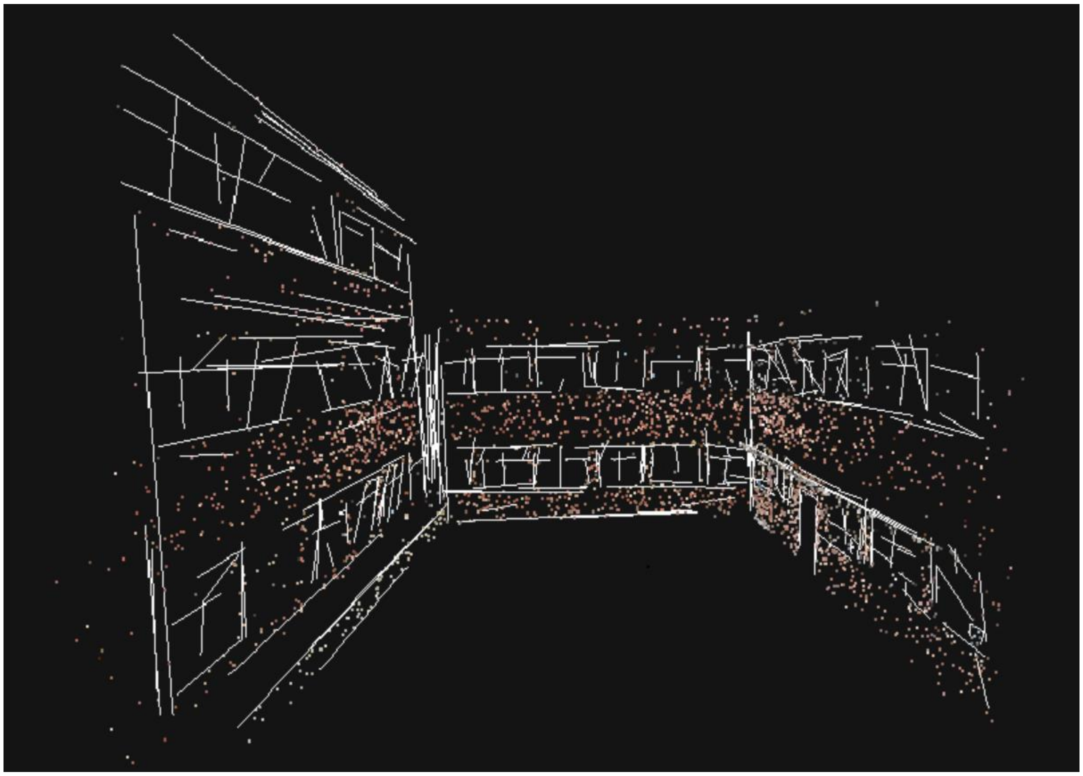


Figure 4.13 (b): Hybrid 3D reconstruction of the façade (resolution = 5MP)

An issue that can be noticed in the results of the hybrid 3D reconstruction of the façade is the existence of line segments that have significant amount of error in 3D location and/or end-points. Two reasons could be listed for such errors. First, the reprojection error for a line segment in the hybrid bundle adjustment process is calculated by using the homogeneous coordinates of projected lines. In such a scenario, the value of the third coordinate typically has significant scale difference with the first and second coordinates. Such a scale difference can introduce errors in the existence of noise. Second, the hybrid bundle adjustment works with Plucker coordinates of infinite lines. Once the final estimations for these coordinates are acquired, two 3D points that represent the end-points of the line have to be found. A system of linear equations is constructed for that purpose. The solution for this system could be erroneous because of the noise in the input data.

4.4. Identify Salient Planar Regions

This section presents the results of the experiments on two environments that include planar surfaces: the building façade with brick pattern and a residential roof structure. At each experiment, the performance of the proposed method was evaluated in comparison with the algorithm presented in (Wang, et al., 2013) as the benchmark. Table 4.10 demonstrates the results of this comparison. In this table, TP represents the planar regions that are detected correctly; FP shows the non-planar regions that are detected as a planar region; and FN represents the planar regions that are not detected.

Table 4.10: Performance evaluation of the proposed method to identify planar regions

Method	**Façade**			**Roof Structure**		
	TP	**FP**	**FN**	**TP**	**FP**	**FN**
Proposed Method	6	0	1	14	1	1
Wang et al. (2013)	6	1	1	12	0	3

As indicated in Table 4.10, the first experiments led to the same results for both methods in terms of TP and FN. The main reason is that sufficient number of feature points could be detected in the local neighborhood of each line segment which is the primary assumption in the benchmark method; hence the perfoemance of both methods is at the highest. Figure 4.14 indicates two line segments and the neighborhood area around them that is calculated based on a normal distribution function; it is assumed that the feature points in those areas are coplanar with their corresponding line segments.

Results from the second experiment further highlight the performance improvement because of using the proposed method. The roof structure scene displays characteristics that do not always satisfy the requirements of the benchmark method; accordingly, the method presented in (Wang, et al., 2013) fails to detect two planar regions compared to the proposed method.

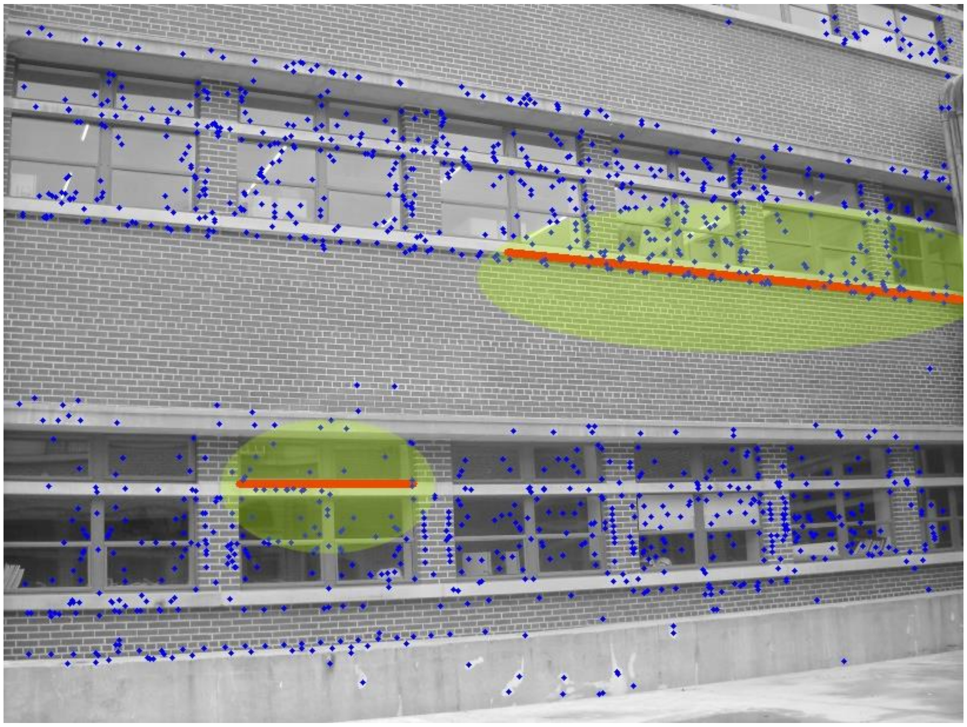


Figure 4.14: Examples of line segments and their neighborhood calculated from a normal distribution function

4.5. Close-Range Video-Based Roof Reconstruction

This section presents the results of four experiments aiming to evaluate the performance of the proposed videogrammetric roof reconstruction framework. The framework is a collection of all the previously evaluated steps (i.e., section 4.1 to 4.4) in addition to some other steps that have been extensively evaluated in the literature. It is designed specifically to take advantage of the knowledge about characteristics of a roof structure. The framework holds the promise to produce a measureable 3D wire-diagram for each plane in a roofing structure such that no/minimum manual input is required. The experiments begin with very simple scenarios and then extended to complex roof structures with several intersecting planes. The previously mentioned calibrated stereo camera setup (i.e., two video cameras with a resolution of 2448×2048 pixels + two fixed focal length lenses with $f = 25mm$ + an extendible pole) was used to collect the video streams in all the experiments. The videos were also pre-processed using the algorithm presented in (Rashidi, et al., 2013) in order to extract key-frames with minimum motion blur and sufficient visual data. These experiments are presented below.

As suggested in (Scholze, et al., 2002) and in all four experiments, roof planes were modeled as planar, convex polygonal patches. It is supposed that straight line segments connect the corner points at each plane. Moreover, only the simplest polygons (triangular and quadrangular) were considered and the composition of these two primitives was used to describe more complex planes. According to the same study, the angles between adjacent roof patch borders were modeled using 15^0 angle steps. Therefore, the finite set of possible angles is

$$\Phi = \left\{15^0, 30^0, 45^0, 60^0, 75^0, 90^0, 105^0, 120^0, 135^0, 150^0, 165^0\right\}$$

Roof model: The roof model provides a controlled, yet realistic environment in order to prove the concept of the videogrammetric roof reconstruction. No occlusion involved in this experiment and most of the practical constraints that are available in a construction jobsite could be avoided. On the other hand, the geometry is very simple and only three intersecting planes need to be measured. The experiment was performed on the same video data that was used in section 4.3. In this experiment, the pairs of line segments that are nearly coplanar were located by a distance threshold of 5cm and an angle threshold of 5^0. The three major planes on the object were successfully detected and the boundaries of the planes were determined by intersecting the reconstructed lines on each plane and finding the convex hull from the data. The 95 percentile error in measuring the end-to-end dimensions of the roof plane boundaries was also calculated compared to the ground truth data. Figure 4.15 demonstrates the results and the numerical analysis is presented in Table 4.11.

Table 4.11: Performance evaluation for 3D reconstruction of the roof model

Resolution	Focal Length (mm)	No. of missing edges	No. of generated wire-diagrams out of 3	95 percentile measurement error (cm)	Manual input
2448×2048	25	0	3	2.83	NONE
1900×1600	19.5	0	3	3.17	NONE

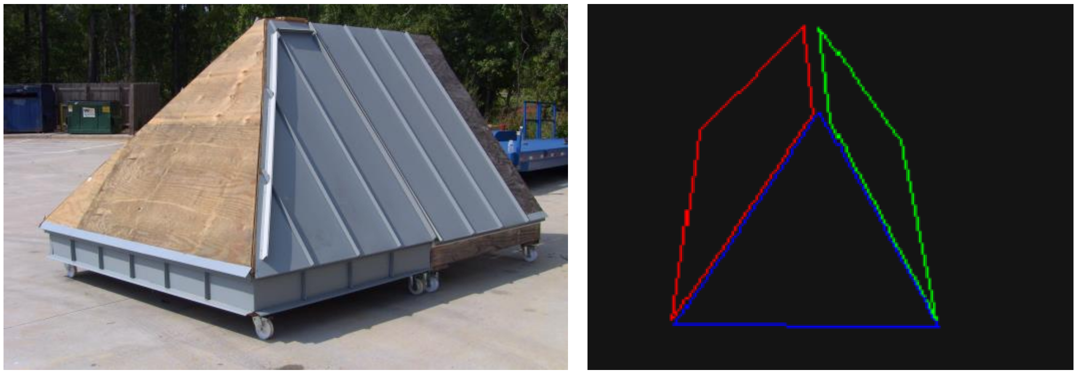


Figure 4.15: Measureable 3D wire-diagrams for planes in the roof model

Simple residential roof: After evaluating the framework in a controlled setting, this experiment tests the package in a more realistic environment. The goal in this experiment is to keep the geometry as simple as possible but add practical constraints that one may encounter when using the framework such as cluttered environment in the background, lack of accessibility, limited visibility, and fragmented straight line segments. The experiment also tests the applicability of the framework for real-size roof structures. The target roof belongs to a one-story residential building and has a very simple geometry. The underlying geometry includes a rectangular plane and two slopped planes that are inside the main plane. 35 stereo video frames were extracted from the recorded data. In this experiment, a distance threshold of 5cm and an angle threshold of 5^0 were used to find line pairs that are nearly coplanar. The major planes in the scene were successfully detected and the 95 percentile measurement error was calculated. Figure 4.16 and Table 4.12 summarize the findings in this experiment.

Table 4.12: Performance evaluation for 3D reconstruction of the simple residential roof

Resolution	Focal Length (mm)	No. of missing edges	No. of generated wire-diagrams out of 3	95 percentile measurement error (cm)	Manual input
2448×2048	25	0	3	3.27	NONE
1900×1600	19.5	0	3	4.52	NONE

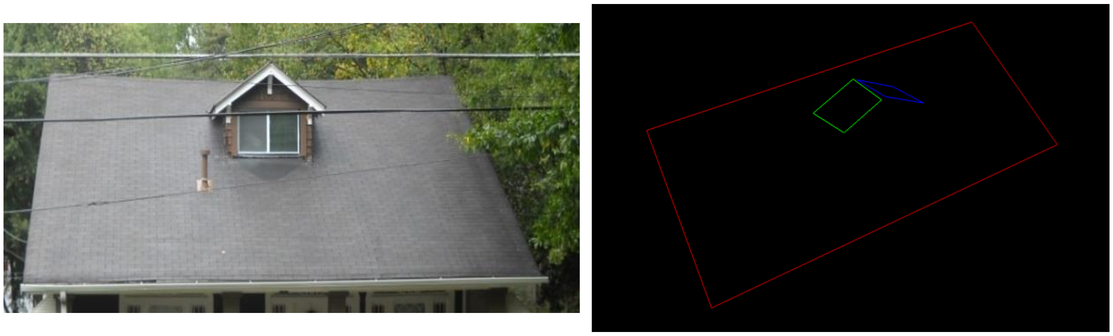


Figure 4.16: Measureable 3D wire-diagrams for planes in the simple residential roof

Complex roof structures: This section presents two more experiments on roof structures with a more complex geometry and several intersecting planes. They represent real-life scenarios for using the proposed framework and impose all possible practical constraints; these include large-scale environment, partial occlusion of roof planes due to the angle of view, perspective distortion of the views because of wide baselines, difficulties in videotaping the structure from the ground, and potential moving objects in the background.

The first case is a roof structure that is located on top of a two-story residential building that has a façade with brick pattern. Although the texture of the roof covering material is such that enough number of feature points could be detected to run the package, the façade texture was also very rich in terms of the existence of feature points. This allowed robust estimation of the camera motion in the environment. The proposed framework could successfully generate a sparse 3D point cloud and a 3D line set. In detecting the roof planes, the algorithm failed to detect one of the planes and also a surface was incorrectly labeled as a roof plane; however, the rest of the planes were identified correctly. Therefore, two manual inputs were needed to correct the mistakes of the algorithm. Once the wire-diagrams were generated, the end-to-end dimensions of the planes were compared with the ground truth data that was collected using total station surveying. In general, this experiment supported the research hypothesis and showed that the videogrammetric roof reconstruction framework is capable of producing measureable 3D wire-diagrams for roof planes with no/minimum manual input. Several intermediate results are demonstrated in Figures 4.17-4.23 and Table 4.13 shows the numerical analysis.

Table 4.13: Performance evaluation for 3D reconstruction of the first complex roof

Resolution	Focal Length (mm)	No. of missing edges	No. of generated wire-diagrams out of 14	95 percentile measurement error (cm)	Manual input
2448×2048	25	0	14	4.79	TWO
1900×1600	19.5	1	13	6.38	FIVE

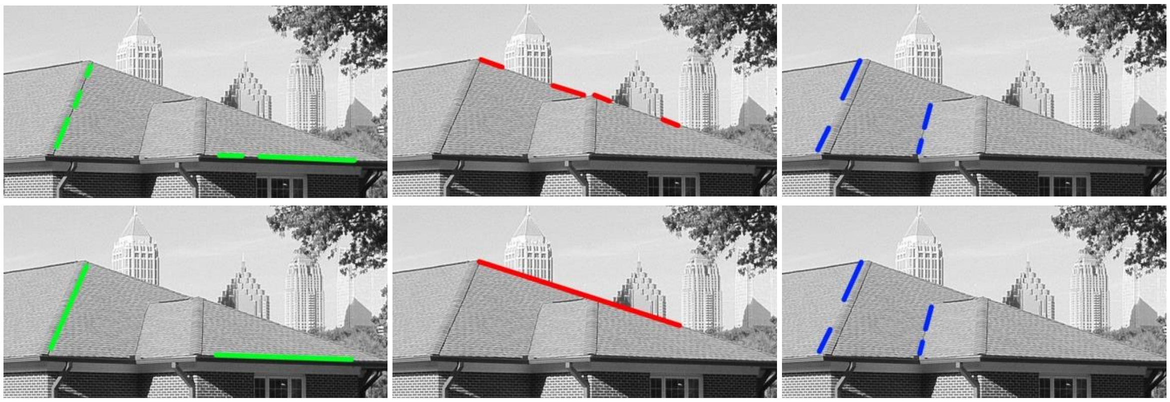


Figure 4.17: Sample results for collinear line merging in reconstructing the first complex roof (first column: true positive (TP); second column: false positive (FP); and third column: false negative (FN))

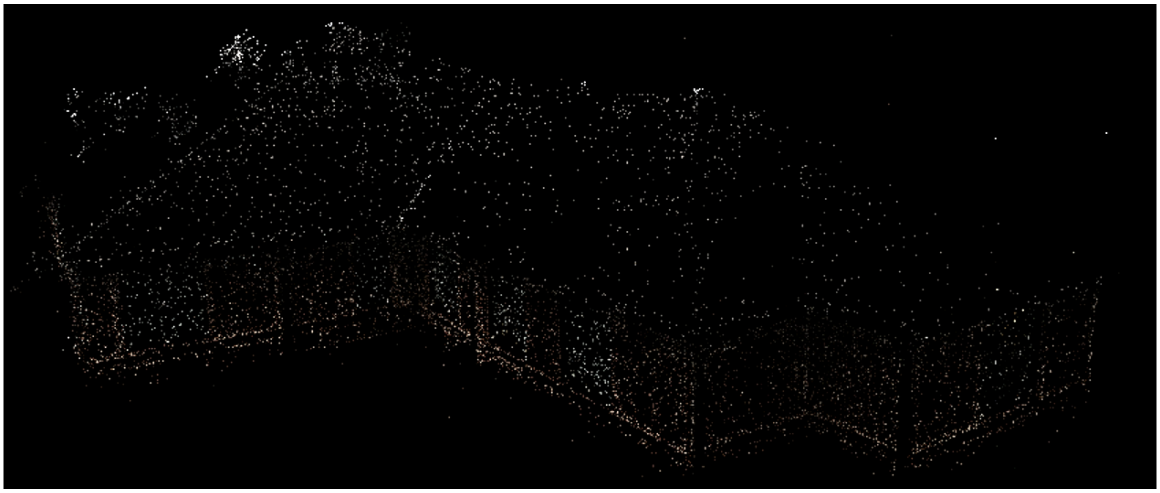


Figure 4.18: Sparse 3D point cloud generated for the first complex roof (some redundant parts of the point cloud have been deleted manually for a better visualization)

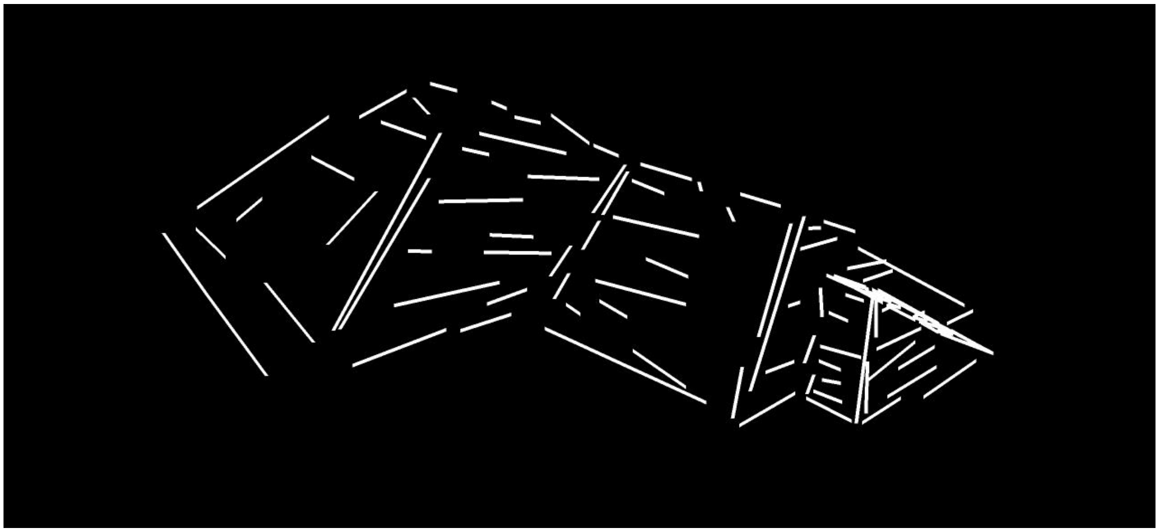


Figure 4.19: 3D line set generated for the first complex roof (redundant parts have been deleted manually for a better visualization)

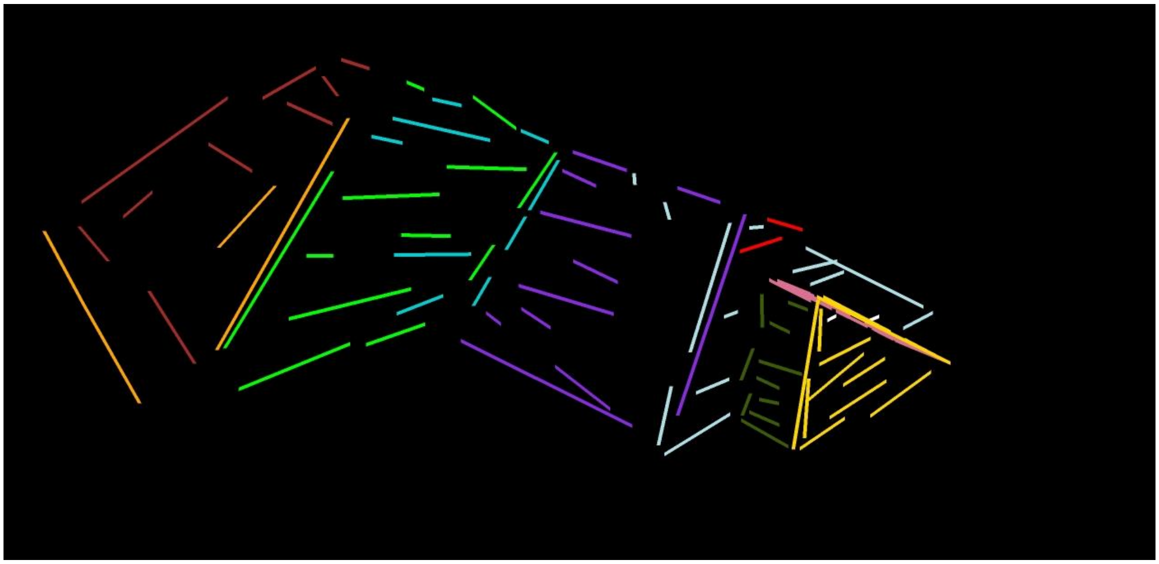


Figure 4.20: Hypotheses for coplanar line segments. Line with the same color present a plane hypothesis.

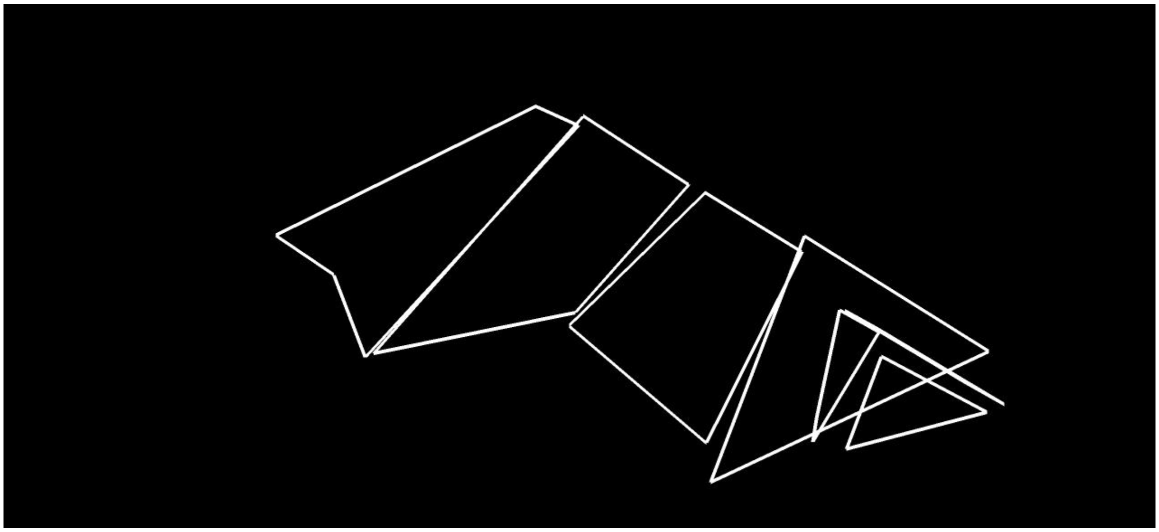


Figure 4.21: Extracted roof planes BEFORE imposing the knowledge about the geometry of a roof structure and manual inputs

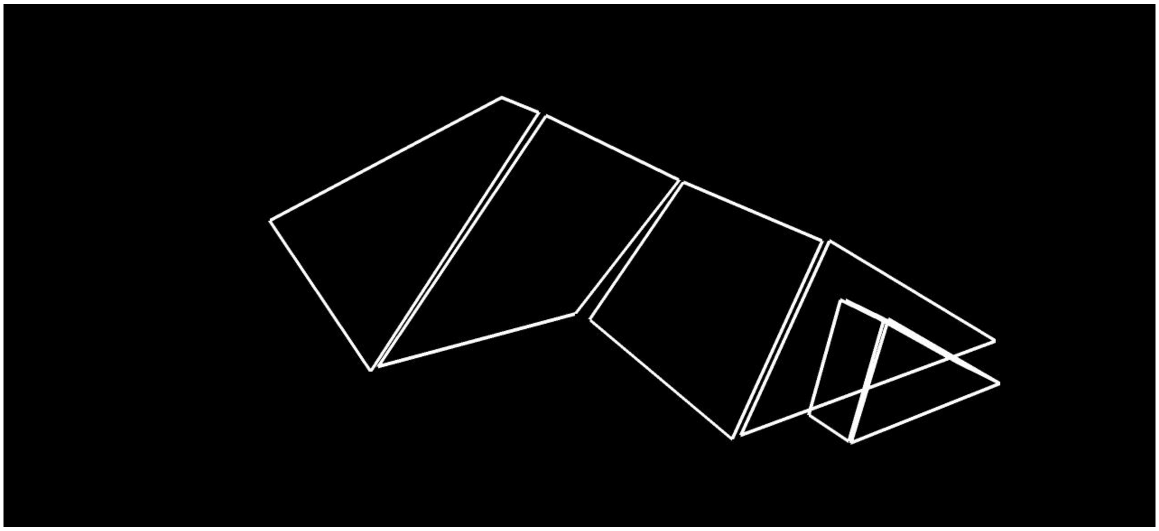


Figure 4.22: Extracted roof planes AFTER imposing the knowledge about the geometry of a roof structure and manual inputs

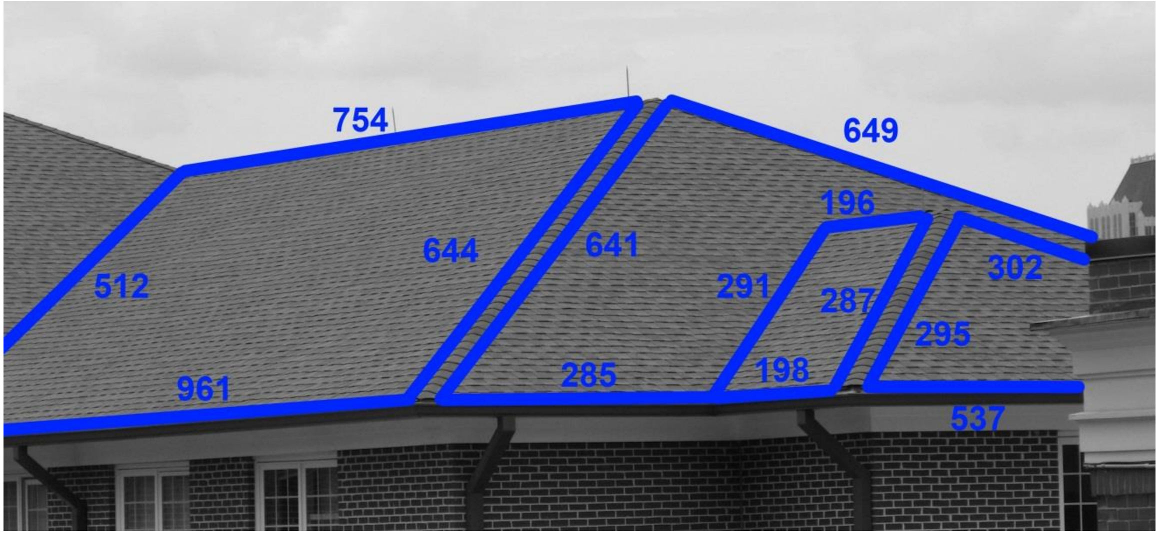


Figure 4.23: Extracted measurements for the first complex roof. The numbers and lines have been printed manually on an image from the roof structure.

The second case is another residential roof structure with a complex geometry that includes a total of 12 planes. The algorithm could successfully identify all the existing roof planes but it also incorrectly marked two areas in the background trees as planar faces. Two manual inputs were therefore needed to remove the false positives. On the other hand, there was a missing edge on one of the trapezoidal faces which needed to be manually added. Using the modified data, a 3D wire-diagram was generated for each of the roof planes and the end-to-end measurements were extracted. As Table 4.14 indicates, the package demonstrated a nearly similar behavior as the previous experiment. The generated 3D wire-diagrams and measurements are shown in Figures 4.24 and 4.25.

Table 4.14: Performance evaluation for 3D reconstruction of the second complex roof

Resolution	Focal Length (mm)	No. of missing edges	No. of generated wire-diagrams out of 12	95 percentile measurement error (cm)	Manual input
2448×2048	25	1	11	4.93	THREE
1900×1600	19.5	3	11	7.12	SEVEN

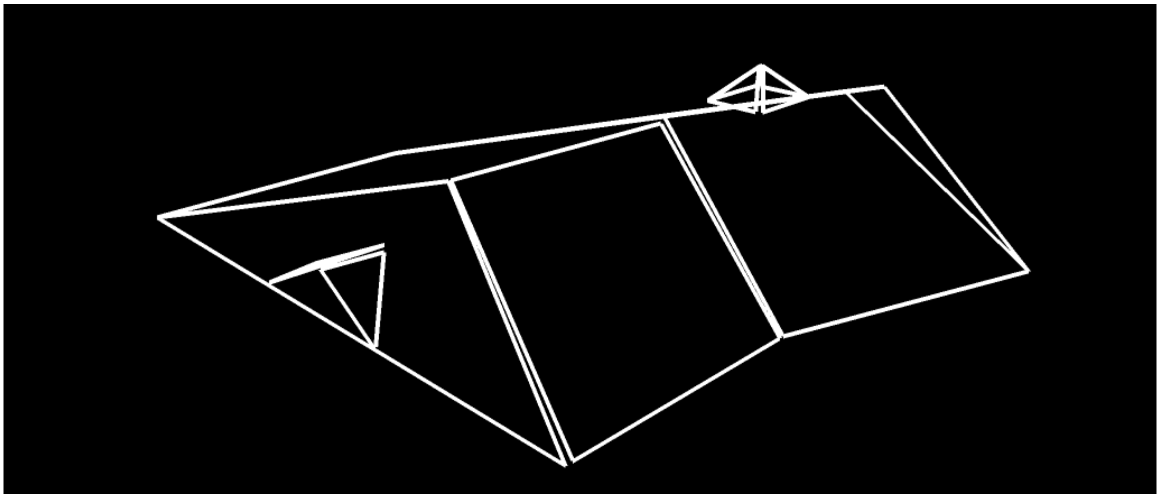


Figure 4.24: Extracted roof planes for the second complex roof. Redundant data has been deleted manually for a better visualization.

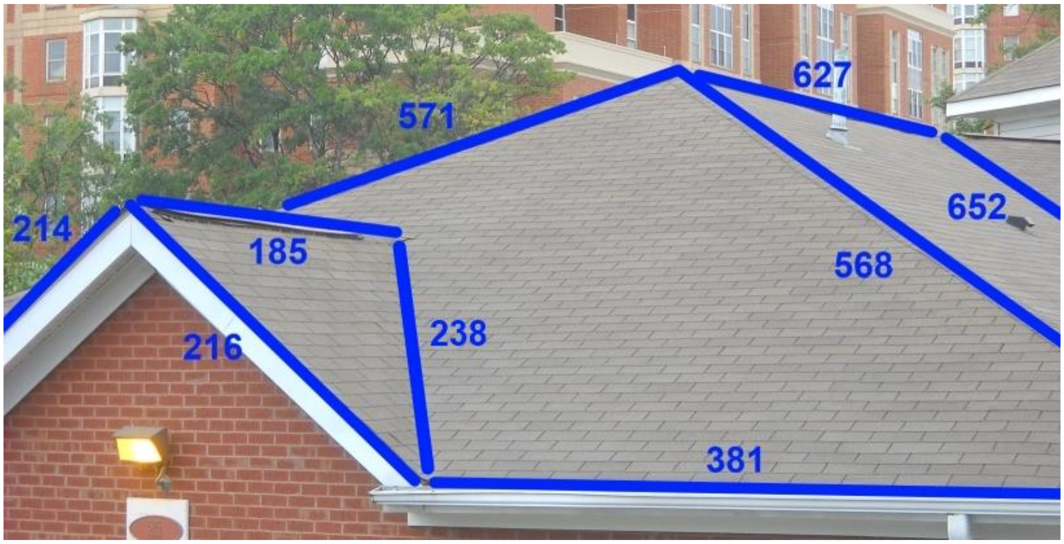


Figure 4.25: Extracted measurements for the second complex roof. The numbers and lines have been printed manually on an image from the roof structure.

CHAPTER 5

CONCLUSIONAND FUTURE WORK

5.1. Conclusion

A roofing contractor typically needs to acquire dimensions of a roof structure several times over the course of its build because a structure is never built to the exact drawing dimensions. Current surveying practices in the roofing industry are labor intensive, time consuming, and/or unsafe. Tape measuring is still the standard practice in the industry despite its apparent limitations. A videogrammetric framework was presented in this research as an alternative method for roof surveying. Compared to the existing methods, it is less expensive, more automated, safer, and simpler to use. When using this method, a roofing contractor collects stereo video streams of a target roof. A 3D wire-diagram is then generated for every roof plane and necessary measurements are extracted.

Four different experiments were used to validate the entire framework. They all supported the research hypothesis presented in this study. They showed the capability of the framework to produce a sparse 3D point cloud and a 3D line set for a typical roof structure that consists of several intersecting planes, provided that the structure can be properly videotaped from the ground using an extendible pole. The reconstruction of the scene can then be used to identify salient planar surfaces on the roof and locate the boundary lines. Although there may exist a few number of missing edges/planes or falsely identified surfaces in the results, they could be corrected via the minimum amount of manual input (i.e., selecting a surface and deleting it or connecting the corner points of an identified plane). The amount of manual input, if any, is tremendously less than the inputs that are required for the existing methods. Another advantage of the method is the level of measurement accuracy (i.e., errors less than ±5cm) which is significantly higher than the accuracy that can be achieved by exiting roof surveying methods that use aerial images (i.e., errors in the order of ±15cm and higher). Therefore, the proposed method is a viable replacement for aerial measurements that is extensively being used in the industry for roof area estimation, damage assessment, appraisal, and insurance claims. However, the current version of the framework cannot satisfy the level of accuracy that is needed for special tasks such as digital fabrication of sheet metal roof panels which requires errors less than 0.5in. or 0.75in; for such a purpose, total station surveying is still the best choice.

These experiments also revealed a very important point about the proper way that a roof structure should be videotaped if a certain straight line needs to be included in the reconstruction. One of the very common cases that results in line degeneracy and hence the failure in reconstruction of the line is the following: if the camera motion in the environment is more or less parallel to the target physical line, the configuration will be degenerate. It is therefore recommended to collect the video data such that the camera motion covers at least two vertical directions; this results in a higher probability for a successful reconstruction. Another solution would be tilting the stereo camera setup for 90 degrees and collecting another round of data by following approximately the same camera path that is used in the first round.

In addition to evaluating the entire framework, a number of intermediate steps were also validated as separate entities. These include multi-step stereo camera calibration procedure, improved descriptor vectors for line segments, hybrid SfM, and efficient recognition of salient planar regions in a scene.

A multi-step stereo camera calibration procedure was proposed aiming to enhance the Euclidean accuracy of 3D reconstruction in far-range scenarios. It recommended using a set of discrete values for representing the distance between the sensor system and the calibration board (D). For each D , a set of stereo video streams were collected while the distance between the camera and the board was fixed to D . Conventional stereo camera calibration algorithms were then used to calculate calibration parameters for the given D . Repeating this process for all the values resulted in a multiple set of parameters each corresponding to a specific D . The sets of calibration parameters were then used in the SfM process with the following assumption: for each 3D point, the set of calibration parameters that have the closest D value to the point's Z coordinate should be used. The experimental results demonstrated that this procedure is capable of reducing the spatial measurement errors by 25%.

Descriptor vectors that are constructed for each line segment based on the MSLD algorithm were also improved by incorporating a dynamic support region and canonical form representation. The support region was determined based on the zero-crossings of the Laplacian function; the region is therefore scale-invariant and insensitive to a wide range of viewpoint transformations. The canonical representation was also used to compensate for the image distortion. The improved algorithm outperformed the original one in terms of rotation, scale, and viewpoint changes. However, its performance remained more or less the same for illumination and blur changes. In average, the improved algorithm resulted in 4% increase in recall and 5% increase in precision.

A mathematical formulation was defined for a hybrid SfM approach that allows camera motion estimation and 3D structure inference from a combination of point and line features. The extensive set of experiments indicated that the average reprojection error and average spatial distance error for point features in a hybrid reconstruction is more or less the same as the case of a point-based 3D reconstruction. However, these metrics are significantly higher for line features; this means that a hybrid approach is more robust than a line-based 3D reconstruction. On the other hand, the hybrid approach generates 3D information with more visual clues about the underlying geometry (points + edges).

Motivated by the fact that man-made environments are often composed of piecewise planar or nearly planar primitives, a multi-step method was presented for identifying salient planar regions in built environments. The method is based on hypothesizing candidate planes from a cloud of reconstructed 3D points and line segments. The first set of candidates were found using a combination of points and lines. The information that had not been used in the first step was further processed to find candidate planes from a pair of line segments or a set of three points. An image intensity similarity function was finally used to verify each plane hypothesis; those that do not satisfy the minimum requirements were discarded and the rest was searched for nearly identical plane equations to be merged. The performance of the method was evaluated in comparison with the most state-of-the-art algorithm in the literature as the benchmark. The results indicated that the proposed method outperforms the benchmark method both in terms of the plane detection accuracy and computational efficiency.

5.2. Future Work

While performing the research, several additional questions were raised that could be the subject of future research efforts. Moreover, a number of open problems exist that need to be solved. As a result, the following directions and ideas are presented.

The proposed multi-step stereo camera calibration procedure was based upon the observation that a constant distance between the camera system and the calibration board can decrease the uncertainty range for estimations. This point was not proven mathematically and only its correctness was supported through several experiments. A comprehensive mathematical analysis to study the relationship between the distance and uncertainty range could be very helpful for understanding the nature of problem. It can also enable quantifying the impact of each parameter on the final outcome.

In the field of aerial photogrammetry, a very sophisticated calibration process is performed for airborne mapping cameras. Such a process may be applicable in close-range 3D reconstruction of large-scale environments, but it has not been studied yet. A study that investigates the performance improvement/loss and its trade-off with the extra computational requirements could be invaluable.

In the recent years, drones have been used in numerous applications to collect visual data. Compared to planes, drones are more flexible and less expensive. Moreover, they can fly in much lower heights and are capable of collecting data from close distances. Since the presented methods in this research are generic and hence applicable to the data collected via drones or Unmanned Aerial Vehicles (UAVs), one may study the proposed framework if such data is used as input. However, it needs to be considered that the use cases for drones and UAVs are restricted by several regulations.

As explained in previous sections, measurements with errors less than ±5cm could be achieved in this research. The followings are some recommendations that could be considered in future research efforts to scale down the amount of error. First, algorithms that use a series of measurements observed over time and produce estimates of unknown variables (e.g., Kalman Filter or Extended Kalman Filter) can be used to build a model for the state of the system and maximize the a posteriori probability of those previous measurements. These algorithms may be used for camera motion estimation or 3D coordinates of visual features. Second, additional sensor types such as GPS or INS could be fused into the sensor system in order to provide extra data for the current location, relative movement, etc. Those extra data could prevent the optimization from local optima and increase the robustness of the framework. Moreover, they can be used as filters to remove potential mismatches in corresponding features. Third, probabilistic approaches that estimate the geometry with the most likelihood can be used. The integration of these algorithms with the existing BIM models (as a source for estimating the initial geometry) could potentially increase the overall performance. Fourth, the use of a more advanced hardware is the last solution that can increase the measurement accuracy in the cost of higher cost or computational requirements. This could mean: cameras with higher resolution, less motion blur, higher signal to noise ratio, and/or integrated sensors such as GPS; rectilinear lenses capable of keeping lines that appear straight in the real world straight on the image sensor.

REFERENCES

Akinlar, C. & Topal, C., 2011. EDLines: A real-time line segment detector with a false detection control. Pattern Recognition Letters, 32(13), pp. 1633-1642.

Baillard, C. & Zisserman, A., 2000. A plane-sweep strategy for the 3D reconstruction of buildings from multiple images. In proceedings of ISPRS Congress and Exhibition, Amsterdam.

Bartoli, A., 2007. A random sampling strategy for piecewise planar scene segmentation. Computer Vision and Image Understanding, 105(1), pp. 42-59.

Bartoli, A. & Sturm, P., 2005. Structure-from-motion using lines: representation, triangulation, and bundle adjustment. Computer Vision and Image Understanding, 100, pp. 416-441.

Bay, H., Ess, A., Tuytelaars, T. & Gool, V. L., 2008. Speeded-Up Robust Features (SURF). Computer Vision and Image Understanding, 110(3), pp. 346-359.

Bay, H., Ferrari, V. & von Gool, L., 2005. Wide-baseline stereo matching with line segments. In proceedings of IEEE International Conference on Computer Vision and Pattern Recognition.

Bazargani, H., Omidi, E. & Talebi, A., 2012. Adaptive Extended Kalman Filter for asynchronous shuttering error of stereo vision localization. In proceedings of IEEE International Conference on Robotics and Biomimetics.

Brilakis, I., Fathi, H. & Rashidi, A., 2011. Progressive 3D reconstruction of infrastructure with videogrammetry. Automation in Construction, 20(7), pp. 884-895.

Chandraker, M., Lim, J. & Kriegman, D., 2009. Moving in stereo: efficient structure and motion using lines. In proceedings of IEEE International Conference on Computer Vision, Kyoto.

Cheng, L., Gong, J., Li, M. & Liu, Y., 2011. 3D building model reconstruction from multi-view aerial imagery and Lidar data. Photogrammetric Engineering and Remote Sensing, 77(2), pp. 125-139.

Christy, A. & Horaud, R., 1999. Iterative pose computation from line correspondences. Computer Vision and Image Understanding, 73(1), pp. 137-144.

Coaker, L. H., 2009. Reflector-less total station measurements and their accuracy, precision and reliability, Dissertation, University of Southern Queensland.

Coffelt, D. & Hendrickson, C., 2010. Life-cycle costs of commercial roof systems. Journal of Architectural Engineering, 16(1), pp. 29-36.

Cory, J., 2009. Roofing contractors make use of a new estimating tool they say is safer and more accurate than hand-measuring: satellites. Available at: http://www.replacementcontractoronline.com/industry-news.asp?sectionID=316 [Accessed 2012].

Cui, Y., Zhao, X. & Jing, C., 2012. An approach of aerial photogrammetry measurement based on 3D model. Key Engineering Materials, 500, pp. 736-742.

Dai, F., Rashidi, A., Brilakis, I. & Vela, P., 2013. Comparison of image- and time-of-flight-based technologies for 3D reconstruction of infrastructure. Journal of Construction Engineering and Management, 139(1), pp. 69-79.

Dang, T., Hoffmann, C. & Stiller, C., 2009. Continuous stereo self-calibration by camera parameter tracking. IEEE Transactions on Image Processing, 18(7), pp. 1536-1550.

Desolneux, A., Ladjal, S., Moisan, L. & Morel, J., 2002. Dequantizing image orientation. IEEE Transactions on Image Processing, 11(10), pp. 1129-1140.

Du, S., van Wyk, B., Tu, C. & Zhang, X., 2010. An improved Hough transform neighborhood map for straight line segments. IEEE Transactions on Image Processing, 19(3), pp. 573-585.

EagleView Technologies, 2011. Satellite or aerial photography?

EagleView Technologies, 2012. Case study: Aerial measurements shown to increase C-SAT.

Fan, B., Wu, F. & Hu, Z., 2012. Robust line matching through line–point invariants. Pattern Recognition, 45, pp. 794-805.

Fernandes, L. & Oliveira, M., 2008. Real-time line detection through an improved Hough transform voting scheme. Pattern Recognition, 41(1), pp. 299-314.

Frahm, J. et al., 2010. Fast robust reconstruction of large scale environments. In proceedings of 44th Annual Conference on Information Sciences and Systems.

Fredericks, T. et al., 2005. Occupational injuries and fatalities in the roofing contracting industry. Journal of Construction Engineering and Management, 131(11), pp. 1233-1240.

Fujiyoshi, H. et al., 2003. Fast 3D position measurement with two unsynchronized cameras. In proceedings of IEEE International Symposium on Computational Intelligence in Robotics and Automation.

Furukawa, Y., Curless, B., Seitz, S. & Szeliski, R., 2009. Manhattan-world stereo. In proceedings of IEEE Conference on Computer Vision and Pattern Recognition.

Furukawa, Y. & Ponce, J., 2006. High-fidelity image-based modeling, UIUC.

Furukawa, Y. & Ponce, J., 2009. Accurate camera calibration from multi-view stereo and bundle adjustment. International Journal of Computer Vision, 84(3), pp. 257-268.

Furukawa, Y. & Ponce, J., 2010. Accurate, dense, and robust multi-view stereopsis. IEEE Transactions on Pattern Analysis and Machine Intelligence, 32(8), p. 1362–1376.

Gallup, D., 2011. Efficient 3D reconstruction of large-scale urban environments from street-level video, Dissertation, University of North Carolina.

Geiger, A., Zeigler, J. & Stiller, C., 2011. StereoScan: dense 3D reconstruction in real-time. In proceedings of IEEE Intelligent Vehicles Symposium.

Goesele, M. et al., 2007. Multi-view stereo for community photo collections. In proceedings of IEEE International Conference on Computer Vision, Seattle.

Golparvar-Fard, M., Pena-Mora, F. & Savarese, S., 2009. D4AR - a 4-dimensional augmented reality model for automating construction progress monitoring data collection, processing and communication. Journal of Information Technology in Construction, Volume 14, pp. 129-153.

Golparvar‐Fard, M., Peña‐Mora, F. & Savarese, S., 2013. Automated progress monitoring using unordered daily construction photographs and IFC‐based building information models. Journal of Computing in Civil Engineering, Volume in press, doi:10.1061/(ASCE)CP.1943-5487.0000205.

Hartley, R., 1997. Lines and points in three views and the trifocal tensor. International Journal of Computer Vision, 22(2), pp. 125-140.

Hartley, R. & Zisserman, A., 2004. Multiple view geometry in computer vision. 2nd Edition, Cambridge University Press.

Hashiba, H., Kameda, K., Tanaka, S. & Sugimura, T., 2003. Digital roof model (DRM) using high resolution satellite image and its application for 3D mapping of city region. In proceedings of IEEE International Geoscience and Remote Sensing Symposium.

Hernandez, C. & Schmitt, F., 2004. Silhouette and stereo fusion of 3D object modeling. Computer Vision and Image Understanding, 96(3), pp. 367-392.

House, B. & Nickels, K., 2006. Increased automation in stereo camera calibration techniques. Systemics, Cybernetics and Informatics, 4(4), pp. 48-51.

Irschara, A., Zach, C., Klopschitz, M. & Bischof, H., 2012. Large-scale, dense city reconstruction from user-contributed photos. Computer Vision and Image Understanding, 116(1), pp. 2-15.

Izquierdo, S., Rodrigues, M. & Fueyo, N., 2008. A method for estimating the geographical distribution of the available roof surface area for large-scale photovoltaic energy-potential evaluations. Solar Energy, 82, pp. 929-939.

Jaw, J. & Cheng, C., 2008. Building roof reconstruction by fusing laser range data and aerial images. In proceedings of ISPRS Congress.

Jog, G., Fathi, H. & Brilakis, I., 2011. Automated computation of the fundamental matrix for vision based construction site applications. Advanced Engineering Informatics, 25(4), pp. 725-735.

Khaleghi, B., Baklouti, M. & Karray, F., 2009. SILT: Scale-invariant line transform. In proceedings of IEEE International Symposium on Computational Intelligence in Robotics and Automation.

Kim, H. & Lee, S., 2012. Simultaneous line matching and epipolar geometry estimation based on the intersection context of coplanar line pairs. Pattern Recognition Letters, 33, pp. 1349-1363.

Leica 3D Disto - Tutorial area and volume: roof measurement, Leica Geosystems.

Lindeberg, T., 1998. Feature detection with automatic scale selection. International Journal of Computer Vision, 30(2), pp. 77-116.

Lourakis, M. & Argyros, A., 2009. SBA: a software package for generic sparse bundle adjustment. ACM Transaction on Mathematical Software, 36(1), Article 2.

Lowe, D., 2004. Distinctive image features from scale-invariant key points. International Journal of Computer Vision, 2(60), pp. 91-110.

Moons, T., Fr’ere, D., Vanderkerckhove, J. & Van Gool, L., 1998. Automatic modeling and 3D reconstruction of urban house roofs from high resolution aerial imagery. In proceedings of European Conference on Computer Vision, Freiburg, Germany.

NRCA Roofing Manual, 2012. National Roofing Contractors Association.

OSHA, 2009. Commonly Used Statistics. Available at: http://www.osha.gov/oshstats/commonstats.html [Accessed October 2012].

OSHA, 2009. Fall Protection. Available at: http://www.osha.gov/SLTC/fallprotection/index.html [Accessed October 2012].

Peng, J., 2011. Comparison of three dimensional measurement accuracy using stereo vision, Dissertation, University of Regina.

Persson, T., 2009. Building of a stereo camera system, Dissertation, Blekinge Institute of Technology.

Podbreznik, P. & Potocnik, B., 2010. Estimating correspondence between arbitrarily selected points in two widely-separated views. Advanced Engineering Informatics, 24(3), pp. 367-376.

Pollefeys, M. et al., 2008. Detailed real-time urban 3D reconstruction from video. International Journal of Computer Vision, 78(2-3), pp. 143-167.

Pradeep, V. & Lim, J., 2012. Egomotion estimation using assorted features. International Journal of Computer Vision, 98(2), pp. 1-15.

Ramalingam, S., Bouaziz, S. & Sturm, P., 2011. Pose estimation using both points and lines for geo-localization. In proceedings of IEEE International Conference on Robotics and Automation.

Rashidi, A., Dai, F., Brilakis, I. & Vela, P., 2013. Optimized selection of key frames for monocular videogrammetric surveying of civil infrastructure. Advanced Engineering Informatics, 27(2), pp. 270-282.

Rau, J., 2012. A line-based 3D roof model reconstruction algorithm: TIN-merging and reshaping (TMR). Melbourne, Australia.

Ricolfe-Viala, C. & Sanchez-Salmeron, A., 2010. Lens distortion models evaluation. Applied Optics, 49(30), pp. 5914-5928.

Sampath, A. & Shan, J., 2010. Segmentation and reconstruction of polyhedral building roofs from aerial lidar point clouds. IEEE Transactions on Geoscience and Remote Sensing, 48(3), pp. 1554-1567.

Schindler, G., Krishnamurthy, P. & Dellaert, F., 2006. Line-based structure from motion for urban environments. In proceedings of International Symposium on 3D Data Processing.

Schmid, C. & Zisserman, A., 1997. Automatic line matching across views. In proceedings of IEEE International Conference on Computer Vision and Pattern Recognition.

Schmid, C. & Zisserman, A., 2000. The geometry and matching of lines and curves over multiple views. International Journal of Computer Vision, 40(3), pp. 199-233.

Scholze, S., Moons, T. & Van Gool, L., 2002. A probabilistic approach to building roof reconstruction using semantic labeling. In proceedings of DAGM Symposium, Zurich.

Scholze, S., Moons, T. & van Gool, L., 2002. A probabilistic approach to roof extraction and reconstruction. International Archives of Photogrammetry Remote Sensing and Spatial Information Sciences, 34(3/B), pp. 231-236.

Seitz, S. et al., 2006. A comparison and evaluation of multi-view stereo reconstruction algorithms. In proceedings of IEEE Conference on Computer Vision and Pattern Recognition.

Sinha, S., Steedly, D. & Szeliski, R., 2009. Piecewise planar stereo for image-based rendering. In proceedings of International Conference on Computer Vision.

Sinha, S., Steedly, D. & Szeliski, R., 2010. A multi-stage linear approach to structure from motion. In proceedings of ECCV Workshop on Reconstruction and Modeling of Large-Scale 3D Virtual Environments.

Smith, J. & Chang, S., 1995. Single color extraction and image query. In proceedings of International Conference on Image Processing.

Strecha, C. et al., 2008. On benchmarking camera calibration and multi-view stereo for high resolution imagery. In proceedings of IEEE Conference on Computer Vision and Pattern Recognition.

Suveg, I. & Vosselman, G., 2004. Reconstruction of 3D building models from aerial images and maps. ISPRS Journal of Photogrammetry and Remote Sensing, 58, pp. 202-224.

Svedman, M., 2005. 3D structure from stereo vision using unsynchronized cameras, Royal Institute of Technology.

Svedman, M. et al., 2005. Structure from stereo vision using unsynchronized cameras for simultaneous localization and mapping. In proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems.

Tang, P., Akinci, B. & Huber, D., 2009. Quantification of edge loss of laser scanned data at spatial discontinuities. Automation in Construction, 18(8), pp. 1070-1083.

Tomono, M., 2009. Robust 3D SLAM with a stereo camera based on an edge-point ICP algorithm. In proceedings of IEEE International Conference on Robotics and Automation, Kobe, Japan.

Tuytelaars, T. & van Gool, L., 2000. Wide baseline stereo matching based on local, affinely invariant regions. In proceedings of British Machine Vision Conference.

Von Gioi, R., Jakubowicz, J., Morel, J. & Randall, G., 2010. LSD: a fast line segment detector with a false detection control. IEEE Transaction on Pattern Analysis and Machine Intelligence, 32(4), pp. 722-732.

Wang, J., Shi, F., Zhang, J. & Liu, Y., 2008. A new calibration model of camera lens distortion. Pattern Recognition, 41(2), pp. 607-615.

Wang, L., Neumann, U. & You, S., 2009. Wide-baseline image matching using line signatures. In proceedings of IEEE 12th International Conference on Computer Vision.

Wang, Y., Lin, K. & Chung, P., 2013. 3D reconstruction of piecewise planar models from multiple views utilizing coplanar and region constraints. Journal of Information Science and Engineering, 29(2), pp. 361-378.

Wang, Z., Wu, F. & Hu, Z., 2009. MSLD: a robust descriptor for line matching. Pattern Recognition, Volume 42, pp. 941-953.

Werner, T. & Zisserman, A., 2002. New techniques for automated architectural reconstruction from photographs. In proceedings of European Conference on Computer Vision.

Wolf, L., 2006. Wide baseline matching between unsynchronized video sequences. International Journal of Computer Vision, 68(1), pp. 43-52.

Wood, D., 2012. Evaluating and estimating roofing damage. Estimating Today (American Society of Professional Estimators), October, pp. 11-17.

Xu, G., Chen, L. & Gao, F., 2012. Study on binocular stereo camera calibration method. In proceedings of International Conference on IASP.

Zhang, D. et al., 2010. Exploitation of photogrammetry measurement system. Optical Engineering, 49(3).

Zhang, Z., 2000. A flexible new technique for camera calibration. IEEE Transactions on Pattern Analysis and Machine Intelligence, 22(11), pp. 1330-1334.