Automatically Recovering Geometry and Texture from Large Sets of Calibrated Images

by

J.P. Mellor

B.S., Electrical Engineering, U.S. Naval Academy (1987)
B.S., Nuclear Technology, University of the State of N.Y. (1990)
M.S., Electrical Engineering and Computer Science, M.I.T. (1995)

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 2000

Signature of Author . . . .
Department of Electrical Engineering and Computer Science
15 October 1999

Certified by . . . .
Tomás Lozano-Pérez
Professor of Electrical Engineering and Computer Science
Thesis Supervisor

Certified by . . . .
Seth Teller
Associate Professor of Computer Science and Engineering
Thesis Supervisor

Accepted by . . . .
Arthur C. Smith
Chairman, Departmental Committee on Graduate Students

Automatically Recovering Geometry and Texture from Large Sets of Calibrated Images

by
J.P. Mellor

Submitted to the Department of Electrical Engineering and Computer Science
on 15 October 1999, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Abstract

Three-dimensional models which contain both geometry and texture have numerous applications such as urban planning, physical simulation, and virtual environments. A major focus of computer vision (and recently graphics) research is the automatic recovery of three-dimensional models from two-dimensional images. After many years of research this goal is yet to be achieved. Most practical modeling systems require substantial human input and unlike automatic systems are not scalable.

This thesis presents a novel method for automatically recovering dense *surface patches* using large sets (1000's) of calibrated images taken from arbitrary positions within the scene. Physical instruments, such as Global Positioning System (GPS), inertial sensors, and inclinometers, are used to estimate the position and orientation of each image. Essentially, the problem is to find corresponding points in each of the images. Once a correspondence has been established, calculating its three-dimensional position is simply a matter of geometry. Long baseline images improve the accuracy. Short baseline images and the large number of images greatly simplifies the correspondence problem. The initial stage of the algorithm is completely local and scales linearly with the number of images. Subsequent stages are global in nature, exploit geometric constraints, and scale quadratically with the complexity of the underlying scene.

We describe techniques for: 1) detecting and localizing surface patches; 2) refining camera calibration estimates and rejecting false positive surfels; and 3) grouping surface patches into *surfaces* and growing the surface along a two-dimensional manifold. We also discuss a method for producing high quality, textured three-dimensional models from these surfaces. Some of the most important characteristics of this approach are that it: 1) uses and refines noisy calibration estimates; 2) compensates for large variations in illumination; 3) tolerates significant soft occlusion (e.g. tree branches); and 4) associates, at a fundamental level, an estimated normal (eliminating the frontal-planar assumption) and texture with each surface patch.

Thesis Supervisor: Tomás Lozano-Pérez
Title: Professor of Electrical Engineering and Computer Science

Thesis Supervisor: Seth Teller
Title: Associate Professor of Computer Science and Engineering

About the Author

J.P. Mellor received the degree of Bachelor of Science with distinction in Electrical Engineering from the United States Naval Academy in May of 1987. While at the Naval Academy he was selected as a Trident Scholar, awarded the Captain Boyd Alexander prize for electrical engineering and named EASCON student of the year. He is also a member of Sigma Xi, Tau Beta Pi, and Phi Kappa Phi. Following graduation he entered the naval nuclear power program. While assigned as an instructor at the Naval Nuclear Power Training Unit, Balston Spa, he earned the degree of Bachelor of Science in Nuclear Technology from the University of the State of New York in June of 1990. He also served as the main propulsion assistant and sonar officer on board the submarine USS Drum (SSN-677). In July of 1993 he left the naval service to continue his education at the Massachusetts Institute of Technology. He has been a member of the Artificial Intelligence Laboratory since his arrival at MIT and the Computer Graphics Group since its creation in 1995. In February of 1995 he received the Master of Science degree in Electrical Engineering and Computer Science completing a thesis entitled "Enhanced Reality Visualization in a Surgical Environment". After finishing his masters he worked on the *Intelligent Room* project exploring the role of computer vision in human-computer interaction.

Acknowledgments

This thesis would not have been possible without the love and support of my family, particularly my wife Kathryn who has been incredibly understanding through it all.

I would like to thank my thesis supervisors Tomás Lozano-Pérez and Seth Teller. Throughout my 6+ years at the Artificial Intelligence Laboratory Tomás has been a constant source of wisdom and guidance. His thoughtful advice and confidence in my abilities have helped make working on this thesis enjoyable. Seth's enthusiasm for the *City Project* is inspiring and encouraged me to raise my sights. Thanks to Eric Grimson for serving as a reader and providing many insightful comments.

Special thanks to Doug DeCouto, Mike Bosse, Adam Holt, Neel Master, and Satyan Coorg for collecting (Doug, Mike, and Adam) and mosaicing (Neel and Satyan) the pose image dataset without which this thesis could not have been completed. Thanks are also due to all the other members of the *City Project*.

I also wish to thank all of the individuals who make the Artificial Intelligence Laboratory and the Computer Graphics Group an outstanding research environment. The suggestions and feedback provided by fellow lab members significantly improved this thesis.

This work has been supported in part by the Advanced Research Projects Agency of the Department of the Defense under Office of Naval Research contract N00014-91-J-4038 and Rome Laboratory contract F3060-94-C-0204.

To Kathryn, Phillip and Patrick

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Three-dimensional models which contain both geometry and texture have numerous applications. They are used extensively as virtual environments for entertainment (e.g. games, videos, commercials and movies). Three-dimensional models also have serious applications such as physical simulation and urban planning. Recent advances in hardware and software have improved our ability to store, navigate and display large, complex models. However, model construction is frequently a tedious and time consuming task requiring significant human input. As the size, complexity, and realism of models increase, manual, and even interactive or semi-automatic, model construction will rapidly become impractical.

This thesis presents a novel method for automatically recovering dense *surface patches* using large sets (1000's) of calibrated images taken from arbitrary positions within the scene. Physical instruments, such as Global Positioning System (GPS), inertial sensors, and inclinometers, are used to estimate the position and orientation of each image. The large set of images acquired from a wide range of viewpoints aids in identifying correspondences and enables accurate computation of their three-dimensional position. We describe techniques for:

- Detecting and localizing surface patches.
- Refining camera calibration estimates and rejecting false positive surfels.
- Grouping surface patches into surfaces.
- Growing surfaces along a two-dimensional manifold.

In addition, we also discuss a method for producing high quality, textured three-dimensional models from these surfaces. The initial stage of the algorithm is completely local making it easily parallelizable. Some of our approach's most important characteristics are:

- It is fully automatic.
- It uses and refines noisy calibration estimates.
- It compensates for large variations in illumination.

- It matches image data directly in three-dimensional space.
- It tolerates significant soft occlusion (e.g. tree branches).
- It associates, at a fundamental level, an estimated normal (eliminating the frontal-planar assumption) and texture with each surface patch.

Figure 1-1: Albrecht Dürer, *Artist Drawing a Lute*, 1525.

1.1 Background

The basics of perspective image formation have been known for more than 2000 years and date back to Pythagoras, Euclid, and Ptolemy. In the 15^{th} century, Leono Alberti published the first treatise on perspective and later that century Leonardo da Vinci studied perspective projection and depth perception. By the 16^{th} century these concepts were well known to artists (e.g. Figure 1-1). In the 18^{th} century Johan Heinrich Lambert developed a technique called *space resection* to find the point in space from which a picture was made. Through the end of the 18^{th} century, the study of perspective focused exclusively on image formation. The main motivation was accurately depicting the three-dimensional world on a two-dimensional canvas. Early in the 19^{th} century the photograph was invented and the natural question followed: *Can two-dimensional photographs be used to deduce the three-dimensional world which produced them*?

The problem of recovering three-dimensional information from a set of photographs or images is essentially the correspondence problem: *Given a point in one image, find the corresponding point in each of the other images.* Typically, photogrammetric approaches (Section 1.1.1) require manual identification of correspondences, while computer vision approaches (Section 1.1.2) rely on automatic identification of correspondences. If the images are from nearby positions and similar orientations (short baseline), they often vary only slightly, simplifying the identification of correspondences. Once sufficient correspondences have been identified, solving for the depth is simply a matter of geometry. This applies to both calibrated and uncalibrated images. For calibrated images (known internal calibration, camera position, and orientation) a single correspondence is sufficient to triangulate the three-dimensional position of the scene point which gave rise to the corresponding image points. The uncalibrated case is more complicated requiring additional correspondences to recover the calibration as well as the three-dimensional information.

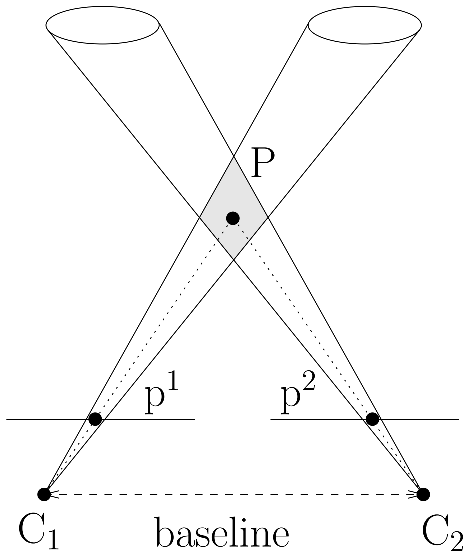


Figure 1-2: Calculating Three-Dimensional Position.

Real images are composed of noisy, discrete samples, therefore the calculated three-dimensional location of a correspondence will contain error. This error is a function of the baseline or distance between the images. Figure 1-2 shows how the location of point P can be calculated given two images taken from known cameras C_1 and C_2 and corresponding points p^1 and p^2 within those images, which are projections of P. The exact location of p^1 in the image is uncertain, as a result P can lie anywhere within the left cone. A similar situation exists for p^2. If p^1 and p^2 are corresponding points, then P could lie anywhere in the shaded region. Clearly, for this situation increasing the baseline between C_1 and C_2 will reduce the uncertainty in the location of P. This leads to a conflict: short baselines simplify the identification of correspondences but produce inaccurate results; long baselines produce accurate results but complicate the identification of correspondences. Real images also frequently contain occlusion and significant variations in illumination both of which complicate the matching process.

1.1.1 Photogrammetry

Figure 1-3: 19^{th} Century Stereoscope.

About 1840, Dominique François Jean Arago, the French geodesist, advocated the use of photography by topographers. A decade later, Aime Laussedat, a Colonel in the French Army Corps of Engineers, set out to prove that photography could be used to prepare topographic maps. Over the next 50 years, his work was so complete that Aime Laussedat is often referred to as the *father of photogrammetry*. Prior to the advent of computing, analog techniques were used to physically reproject single images and stereo pairs. The stereoscopic viewer, shown in Figure 1-3, is one such reprojection device. The availability of computing in the mid-20^{th} century enabled the introduction of analytical techniques. For the first time multiple (more than two) photographs could be analyzed simultaneously [Wolf, 1974, Slama *et al.*, 1980].

In spite of the fact that it originated with ground based imagery, modern photogrammetry uses long-range aerial imagery almost exclusively. This is in contrast with the close-range ground base imagery used as input for this thesis. For accurate results, photogrammetry requires good camera calibration. The internal parameters[1] are measured in a controlled environment and traditionally the external parameters are estimated using points whose three-dimensional position is known or *tie-points*. Classical photogrammetry requires a large amount of human input, typically in the form of identifying corresponding features in multiple photographs. The correspondences are then used to recalculate the external parameters, as well as determine the three-dimensional position of selected features. Two classes of techniques are used

[1] See Appendix A for a discussion of internal and external parameters.

to reestimate the external parameters. Closed-form analytic techniques require fewer correspondences and do not need an initial estimate, but tend to be numerically unstable and sensitive to error in the data. Global optimization techniques, such as *bundle-adjustment*, require both an initial estimate and many correspondences, but generally produce more stable results. The calibration updates described in Section 5.1 fall into the latter category. Recently, automated photogrammetric methods have been explored [Greeve, 1996].

A number of the recent interactive modeling systems are based upon photogrammetry[2]. Research projects such as RADIUS [Collins *et al.*, 1995] and commercial systems such as FotoG [Vexcel, 1997] are commonly used to extract three-dimensional models from images. Good results have been achieved with these systems, however the requirement for human input limits the size and complexity of the recovered model. One approach to reducing the amount of human input is to exploit geometric constraints. The geometric structure typical of urban environments can be used to constrain the modeling process. Becker and Bove [1995] manually identify groups of parallel and perpendicular lines across small sets of images. Shum *et al.*[1998] interactively draw points, lines and planes onto a few panoramic mosaics. Debevec *et al.*'s Facade system [1996] uses a small set of building blocks (cube, cylinder, etc.) to limit the set of possible models from a small set of images (at most a couple dozen). The major strength of these systems is the textured three-dimensional model produced. Debevec *et al.* cite a hundred fold decrease in human input using the block-based approach. Despite this reduction, each image must be processed individually by a human to produce a three-dimensional model, making it difficult to extend these systems to large sets of images.

1.1.2 Computer Vision

The field of computer vision began with the advent of computing in the mid-20^{th} century and in many ways developed in parallel to, but separate from photogrammetry [Horn, 1986, Mayhew and Frisby, 1991, Faugeras, 1993]. A major focus of computer vision is the automatic recovery of three-dimensional information from two-dimensional images. Both calibrated and uncalibrated images can be used and as noted above, recovering three-dimensional information can be reduced to finding correspondences

Uncalibrated Imagery

Several researchers [Longuet-Higgins, 1981, Horn, 1987, Mohr and Arbogast, 1991, Hartley, 1992, Hartley *et al.*, 1992] have shown that the relative camera orientations[3] and the three-dimensional structure of the scene can be recovered from images with known internal calibration and unknown external

[2]Some might consider these to be in the field of computer vision. We have placed them in this section because, like traditional photogrammetric methods, they require human input.

[3]External calibration in an unknown coordinate frame.

calibration. If the internal calibration is also unknown, Faugeras [1992] has shown that the scene structure can only be determined up to an unknown projective transformation. *Structure from motion* and *direct motion estimation* are two popular approaches which use uncalibrated imagery. Structure from motion techniques [Ullman, 1978, Ullman, 1979, Tomasi and Kanade, 1992, Azarbayejani and Pentland, 1995, Seales and Faugeras, 1995] identify and track high-level features (e.g. edges) across several images. Because both structure and motion are recovered, a large set of features must be tracked for a robust solution. Direct motion estimation techniques [Mohr *et al.*, 1993, Szeliski and Kang, 1994, Ayer and Sawhney, 1995, Adelson and Weiss, 1996, Irani *et al.*, 1997] use the *optical flow* of each pixel across images to directly estimate structure and motion. Computing optical flow tends to be sensitive to changes in illumination and occlusion. The need to track features or calculate optical flow across images implies that adjacent images must be close. The cost of acquiring such an image set rapidly becomes prohibitive for large-scale models.

Calibrated Imagery

A popular set of approaches for automatically finding correspondences from calibrated images is relaxation techniques [Marr and Poggio, 1979]. These methods are generally used on a pair of images; start with an *educated guess* for the correspondences; then update them by propagating constraints. These techniques often exploit global constraints such as smoothness [Pollard *et al.*, 1985] and ordering [Ohta and Kanade, 1985]. They don't always converge and don't always recover the correct correspondences. Utilizing only one pair of images at a time has several disadvantages.

- The trade off between easily identifying correspondences and accurate results (discussed above) means that these methods generally produce less accurate results.

- The results from one pair are restricted to the regions visible in both images. To reconstruct a large volume many pairs are required, leading to the difficult problem of integrating the results from multiple pairs.

Another approach is to use multiple images. Several researchers, such as Yachida [1986], have proposed trinocular stereo algorithms and some have proposed using a third image to *check* correspondences [Faugeras and Robert, 1994]. Others have also used special camera configurations to aid in the correspondence problem, [Tsai, 1983, Bolles *et al.*, 1987, Okutomi and Kanade, 1993]. Bolles *et al.* [1987] proposed constructing an epipolar-plane image from a large number of images. In some cases, analyzing the epipolar-plane image is much simpler than analyzing the original set of images. The epipolar-plane image, however, is only defined for a limited set of camera positions. Baker and Bolles [1989] have also attempted to extend the limited set of usable camera positions. Tsai [1983] and Okutomi and Kanade [Okutomi and Kanade, 1993, Kanade and Okutomi, 1994] defined a cost function which was applied directly to a set of images. The extremum of this cost function was then taken as the correct correspondence. Occlusion is assumed to be negligible. In fact, Okutomi and Kanade state that they "invariably obtained better results by using relatively short baselines." This is likely the result of using an image space matching metric (a correlation window) and ignoring perspective distortion and occlusion. Both methods use small sets of images, typically about ten. They also limit camera positions to special configurations. Tsai uses a localized planar configuration with parallel optical axes; and Okutomi and Kanade use short linear configurations.

Some recent techniques use multiple images which are not restricted to rigid camera configurations. Kang *et al.* [1995] use structured light to aid in identifying correspondences. Kang and Szeliski [1996] track a large number of features across a small number of closely spaced ($\sim$ 3 inches) panoramic images (360° field of view). Neither of these approaches is well suited to extended urban environments. Structured lighting is difficult to use outdoors and tracking a dense set of features across multiple large datasets is difficult at best.

Collins [1996] proposed a *space-sweep* method which performs matching in three-dimensional space instead of image space. There are several advantages to matching in three-dimensional space. It naturally operates on multiple images and can properly handle perspective distortion and occlusion, improving the matching process. Collins' formulation assumes that there is a single plane which separates all of the cameras from the scene. As the plane is swept away from the cameras, features from all of the images are projected onto it. Sweep plane locations which have more than a minimum number of features are retained. The resulting output is a sparse set of three-dimensional points. Collins demonstrates his method on a set of seven aerial photographs. Seitz and Dyer [1997] proposed a voxel coloring method which also performs matching in three-dimensional space. Voxels are reconstructed in a special order such that if point P occludes point Q for any image then P is reconstructed before Q. To meet this ordering requirement, no scene points are allowed within the convex hull of the cameras. Collins' sweep-plane is a special case of this ordering requirement. Neither of these approaches is suitable for reconstructions using images acquired from within the scene.

Kutulakos and Seitz [1998b, 1998a] have proposed an extension of voxel coloring called space carving. The separability requirement is removed. An initial estimate of the reconstructed volume is required and must be a relatively tight superset of the actual volume. Voxels are tested in order similar to Seitz and Dyer. Background pixels are removed and matching is performed on raw pixel values. If the projections of a voxel into the images are not consistent, the voxel is carved away. The reconstruction is complete when no more voxels can be removed. The carving operation is brittle. If a single image disagrees the voxel is removed, making this approach particularly sensitive to camera calibration errors, illumination changes, complex reflectance functions, image noise, and temporal variations. In short, it is not well suited to outdoor urban environments.

While computer vision techniques eliminate the need for human input in general they have several characteristics which limit their applicability in creating large, complex, and realistic three-dimensional models.

- They are frequently fragile with respect to occlusion and variations in illumination.
- They frequently operate on only a few images and do not scale to large sets of images.
- The output (e.g. depth maps and isolated edges) are generally not directly useful as a model.

Recently, the last item has been partially addressed by Fua [Fua, 1995, Fua and Leclerc, 1995, Fua and Leclerc, 1996] who starts with a set of three-dimensional points and fits small surface elements to the data. Fitting is accomplished by optimizing an image-based objective function. This is in contrast with the approach presented in this thesis which recovers surface elements directly from the image data. In addition, as presented Fua's approach does not have the ability to fill in *holes* in the three-dimensional points used as input.

Finally, several researchers have used probability theory to aid in recovering three-dimensional information. Cox *et al.* [Cox, 1994, Cox *et al.*, 1996] proposed a maximum-likelihood framework for stereo pairs, which they have extended to multiple images. This work attempts to explicitly model occlusions, although in a somewhat ad hoc manner. Belhumeur [Belhumeur and Mumford, 1992, Belhumeur, 1993, Belhumeur, 1996] develops a detailed Bayesian model of image formation and the structure of the world. These two approaches serve as the inspiration for the probabilistic elements of this thesis.

1.1.3 Discussion

Photogrammetric methods produce high quality models, but require human input. Computer vision techniques function automatically, but generally do not produce usable models, operate on small sets of images and frequently are fragile with respect to occlusion and changes in illumination. The work presented in this thesis draws from both photogrammetry and computer vision. Like photogrammetric methods we produce high quality textured models and like computer vision techniques our method is fully automatic. The major inspiration derives from Bolles *et al.* [Bolles *et al.*, 1987, Baker and Bolles, 1989]. We define a construct called an *epipolar image* and use it to analyze evidence about three-dimensional position and orientation. Like Tsai [1983] and Okutomi and Kanade [Okutomi and Kanade, 1993, Kanade and Okutomi, 1994] we define a cost function that is applied across multiple images, however, we do not evaluate the cost function in image space. Instead, like Collins

[1996], Seitz and Dyer [1997], and Kutulakos and Seitz [Kutulakos and Seitz, 1998b, Kutulakos and Seitz, 1998a] we perform matching in three-dimensional space. We also model occlusion and the imaging process similar to Cox *et al.* [1996] and Belhumeur [Belhumeur and Mumford, 1992, Belhumeur, 1993, Belhumeur, 1996].

There are also several important differences from previous work. The epipolar image we define is valid for arbitrary camera positions within the scene and is capable of analyzing very large sets of images. Our focus is recovering built geometry (architectural facades) in an urban environment. However, the algorithms presented are generally applicable to objects that can be modeled by small planar patches. Surface patches (geometry and texture) or *surfels* are recovered directly from the image data. In most cases, three-dimensional position and orientation can be recovered using purely local information, avoiding the computational costs of global constraints. Some of the significant characteristics of this approach are:

- Large sets of images contain both long and short baseline images and exhibit the benefits of both (accuracy and ease of matching). It also makes our method robust to sensor noise and occlusion, and provides the information content required to construct complex models.

- Each image is calibrated - its position and orientation in a single global coordinate system is estimated. The use of a global coordinate system allows data to be easily merged and facilitates geometric constraints.

- The fundamental unit is a textured surface patch and matching is done in three-dimensional space. This eliminates the need for the frontal-planar assumption made by many computer vision techniques and provides robustness to soft occlusion (e.g. tree branches). Also, the surface patches are immediately useful as a coarse model and readily lend themselves to aggregation to form more refined models.

- The algorithm tolerates significant noise in the calibration estimates and produces updates to those estimates.

- The algorithm corrects for changes in illumination. This allows raw image properties (e.g. pixel values) to be used avoiding the errors and ambiguities associated with higher level constructs such as edges or corners.

- The algorithm scales well. The initial stage is completely local and scales linearly with the number of images. Subsequent stages are global in nature, exploit geometric constraints, and scale quadratically with the complexity of the underlying scene. [4]

[4]This is the worst case complexity. With spatial hashing the expected complexity is linear in the number of reconstructed surfels.

As noted above, not all of these characteristics are unique, but their combination produces a novel method of automatically recovering three-dimensional geometry and texture from large sets of images.

1.2 City Scanning Project

This thesis is part of the City Scanning project whose primary focus is the Automatic Population of Geospatial Databases (APGD). As its name implies, the main target of the City Scanning project is urban environments such as shown in Figure 1-4. In addition to the one presented in this thesis, other approaches to three-dimensional reconstruction have been explored as part of the City Scanning project. Coorg [1998] hypothesizes large (building size) vertical faces from sparse edge information and Chou [1998] attempts to deduce faces by aggregating sparse three-dimensional features. In contrast with these approaches, this thesis performs dense reconstruction directly from the image data. Another major focus of the project is the interactive navigation and rendering of city sized databases. The following sections describe the acquisition and preprocessing phases which produce the calibrated images used as input for the algorithms presented in this thesis. For a more complete description of the project see [Teller, 1998a, Teller, 1998b].

1.2.1 Node Acquisition

A self contained platform called *Argus* (Figure 1-5) continues to be developed to acquire calibrated images[De Couto, 1998]. At the center of Argus is a precision computer controlled pan-tilt head upon which a digital camera is mounted. The camera is initially calibrated using Tsai's method [Tsai, 1983, Tsai, 1987] and is configured to rotate about its center of projection. At each node images are acquired in a hemispherical tiling similar to that shown in Figure 1-6. Argus is also equipped with a number of navigational sensors including GPS, inertial sensors, odometry and inclinometers which enable it to estimate the position and orientation of each image. Currently node positions and orientations are also surveyed by hand which serves as validation. The raw image data, exposure parameters, and absolute position and orientation estimates are recorded by an on-board computer.

1.2.2 Mosaicing and Registration

The position and orientation estimates obtained during the acquisition phase are good, but contain a significant amount of error. Refining these estimates has two parts: mosaicing and registration [Coorg *et al.*, 1998, Coorg, 1998]. Mosaicing exploits the constraint that all of the images in a given node share the same center of projection and form (somewhat more than) a hemisphere.

Figure 1-4: Goal of the city project. Rendering from an idealized model built with human input.

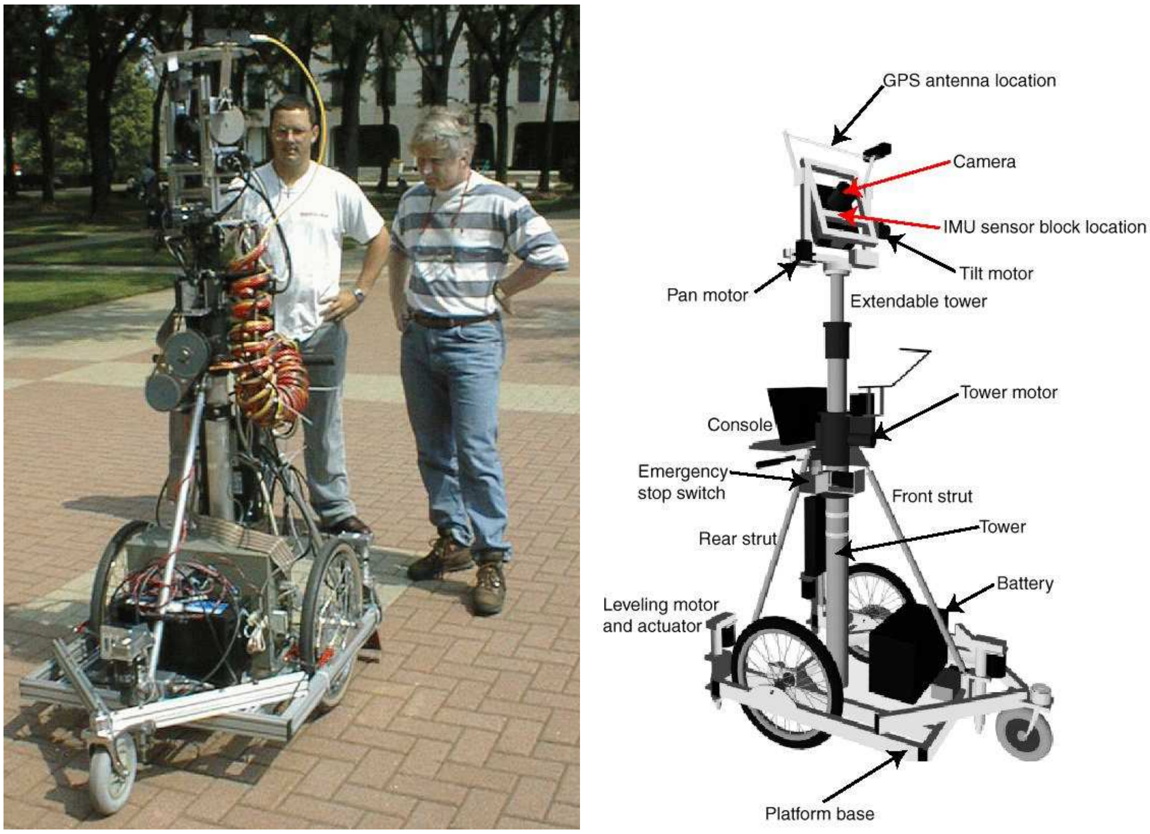


Figure 1-5: Argus.

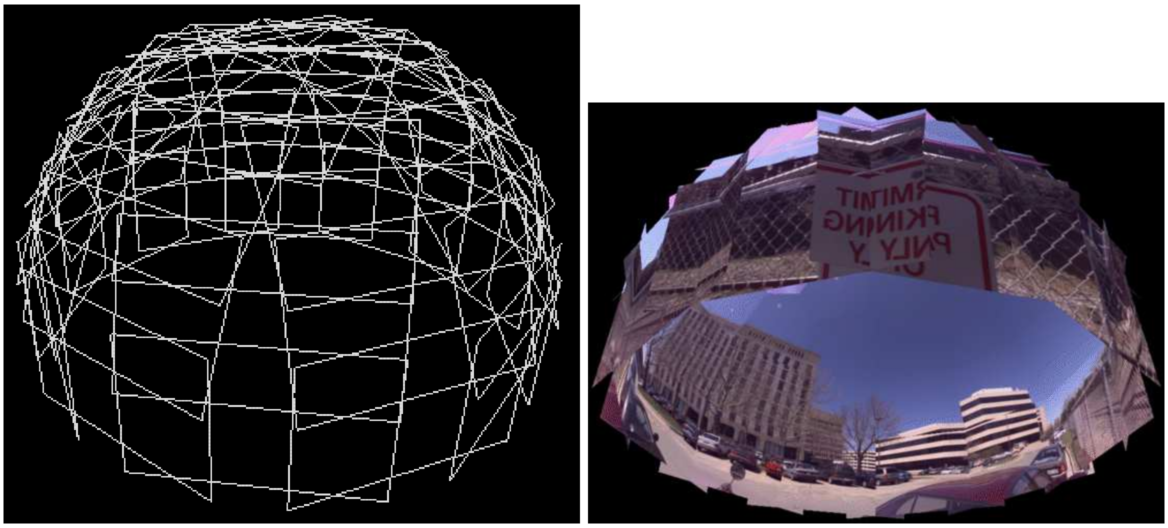


Figure 1-6: Hemispherical tiling for a node.

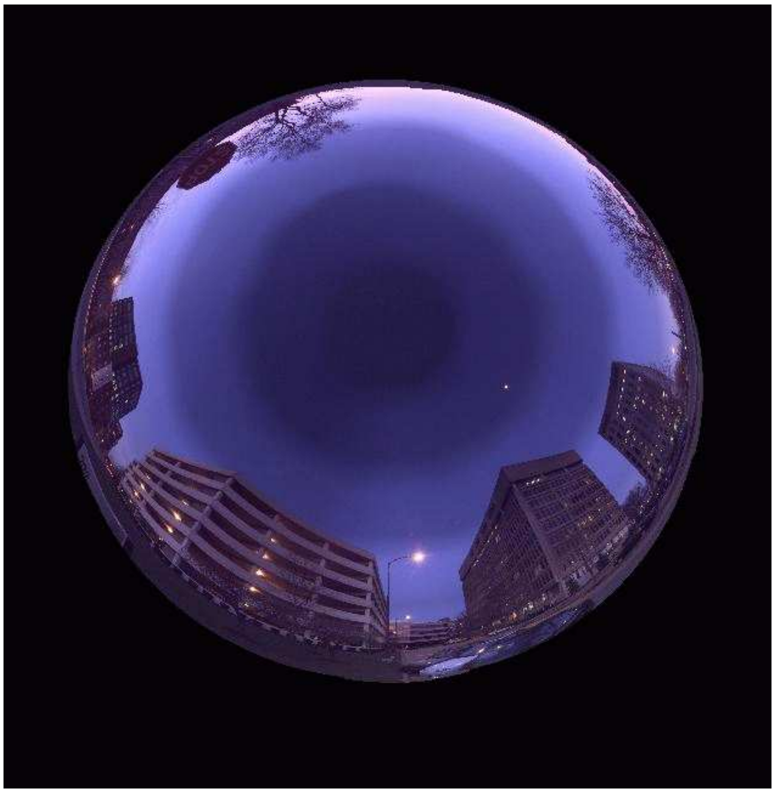


Figure 1-7: Spherical Mosaic for a node.

The relative orientations of a node are optimized by correlating the overlapping regions of adjacent images. A sample of the spherical mosaic produced is shown in Figure 1-7. Once mosaicing is completed a node can be treated as a single image with a single position and orientation. Registration identifies a few prominent features per node and performs a global optimization which refines the location and orientation of each node.

1.3 Thesis Overview

Figure 1-8 shows an overview of the reconstruction pipeline described in this thesis. The calibrated imagery described in the last section serves as input to the pipeline. The left hand column shows the major steps of our approach; the right hand side shows example output at various stages. The output of the pipeline is a textured three-dimensional model. Our approach can be generally characterized as *hypothesize and test*. We hypothesize a surfel and then test whether it is consistent with the data. Chapter 2 presents our basic approach. To simplify the presentation in this chapter we assume perfect data (i.e. perfect calibration, constant illumination, diffuse surfaces, and no occlusion or image noise) and use a synthetic dataset to demonstrate the algorithm. Chapter 3 extends the theory to cover camera calibration error, variations in illumina-

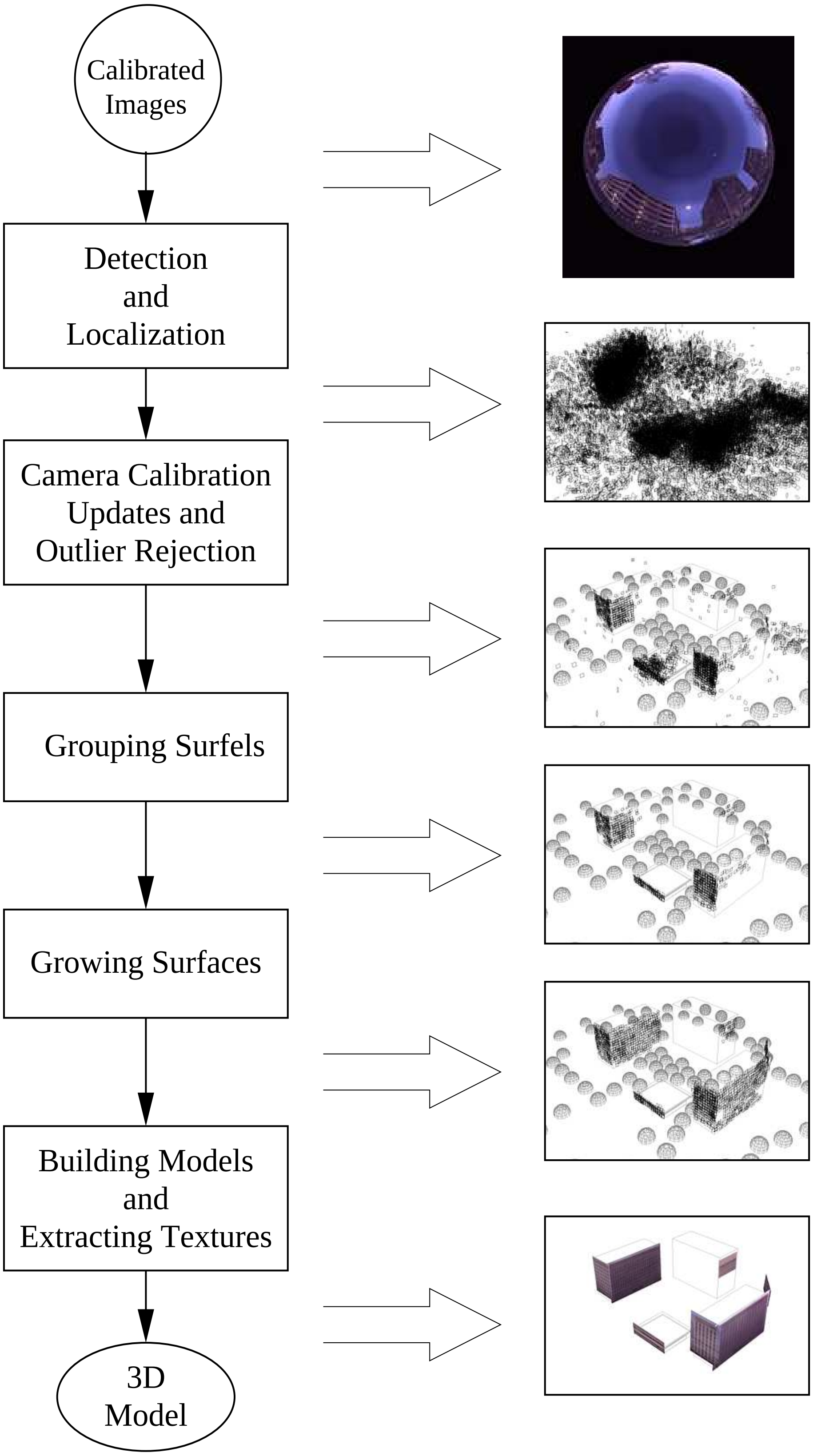


Figure 1-8: Overview of Thesis.

tion, occlusion, and image noise. Chapter 4 explores detecting and localizing surfels. We demonstrate the ability of our algorithm to compensate for camera calibration error, variations in illumination, occlusion, and image noise. We examine its ability to detect and localize three-dimensional surfaces from a significant distance (both in position and orientation) using a purely local algorithm. Noisy data causes some true positives to be missed and compensating for noisy data increases the number of false positives recovered. Chapter 5 introduces several techniques to remove false positives and fill in the missing parts. Camera updates and grouping surfels into surface impose global constraints which remove nearly all of the false positives. Growing surfaces fills in most of the reconstruction. Finally we discuss fitting simple models and extracting textures. We present the algorithms as well as the results of applying them to a large dataset.

1.3.1 Definitions

Some of the more general or fundamental terms used throughout this thesis are defined below. Others will be defined as needed.

- **Internal Parameters**: The camera parameters which govern the image formation process. The exact parameters depend on the camera model used, but usually include: focal length, aspect ratio, and principle point. See Appendix A for a more complete description.

- **External Parameters**: The camera parameters which govern what, in an absolute coordinate system, is imaged - the location of the center of projection and the orientation of the optical axis. See Appendix A for a more complete description.

- **Calibrated Image**: An image for which both the internal and external calibration are known. Also referred to as a pose image.

- **Node**: A set of images acquired from the same location. In other words, all of the images in a node share the same center of projection. The images of a node typically have orientations which tile a hemisphere or more[Coorg *et al.*, 1998].

- **Surfel**: A small planar surface patch or *surface element*. This definition differs from Szeliski's [Szeliski and Tonnesen, 1992] in that it refers to a finite sized patch which includes both geometry and texture.

- **Units**: All of the datasets used in this thesis share a common global coordinate system in which 1 unit equals 1/10 foot.

The pin-hole camera model is used throughout this thesis. As noted in Appendix A it is a linear model and is not capable of modeling nonlinear distortion. Images which have large amounts of distortion can be *unwarped* as a preprocessing step to remove the nonlinear portions, however this was not necessary for the datasets used in this thesis.

Chapter 2

The Basic Approach

In this chapter we describe our basic approach. To simplify the presentation we assume perfect data (i.e. perfect calibration, constant illumination, diffuse surfaces, and no occlusion or image noise). These assumptions are relaxed in the following chapter. The approach presented here was initially inspired by the work of Bolles *et al.* [1987]. The notation used in this chapter is defined in Table 2.1 and Section 2.1 reviews epipolar geometry and epipolar-plane images. We then define a construct called an *epipolar image* (Section 2.2) and show how it can be used to reconstruct three-dimensional information from large sets of images (Section 2.3). Like Tsai [1983] and Okutomi and Kanade [1993] we define a cost function that is applied across multiple images, however, we do not evaluate the cost function in image space. Instead, like Collins [1996], Seitz and Dyer [1997], and Kutulakos and Seitz [1998b, 1998a] we perform matching in three-dimensional space. We also model occlusion and the imaging process similar to Cox [1996] and Belhumeur [Belhumeur and Mumford, 1992, Belhumeur, 1993, Belhumeur, 1996]. There are several important differences, however. An epipolar image is valid for arbitrary camera positions and overcomes some forms of occlusion. Where three-dimensional information cannot be recovered using purely local information, the evidence from the epipolar image provides a principled distribution for use in a maximum-likelihood approach (Section 2.4) [Duda and Hart, 1973]. Finally, we present the results of using epipolar images to analyze a set of synthetic images (Section 2.5).

2.1 Epipolar Geometry

Epipolar geometry provides a powerful constraint for identifying correspondences. Given two cameras with known centers C_1 and C_2 and a point P in the world, the epipolar plane Π_e is defined as shown in Figure 2-1. P projects to p^1 and p^2 on image planes Π_i^1 and Π_i^2 respectively. The projection of Π_e onto Π_i^1 and Π_i^2 produce epipolar lines ℓ_e^1 and ℓ_e^2. This is the essence of the epipolar constraint. Given any point p on epipolar line ℓ_e^1 in image Π_i^1, if the corresponding point is visible in image Π_i^2, then it must lie on the epipolar line ℓ_e^2.

Bolles *et al.* [Bolles *et al.*, 1987] used the epipolar constraint to construct a special image which they called an epipolar-plane image. As noted earlier, an epipolar line ℓ_e^i contains all of the information about the epipolar plane Π_e

P_j	Absolute coordinates of the j^{th} surfel.
n_j	Orientation of the j^{th} surfel.
C_i	Center of projection for the i^{th} camera.
Π_{i}^i	Image plane for the i^{th} camera.
I^i	Calibrated image. $\mathrm{I}^i = \langle \Pi_{\mathrm{i}}^i, \mathrm{C}_i \rangle$.
$\mathbf{I}$	Set of calibrated images. $\{\mathrm{I}^i\}$.
p_j^i	Image point. Projection of P_j onto Π_{i}^i.
Π_{e}^k	The k^{th} epipolar plane.
$\ell_{\mathrm{e}}^{k,i}$	Epipolar line. Projection of Π_{e}^k onto Π_{i}^i.
$\mathrm{p}^\star$	Base image point. Any point in any image.
$\mathbf{p}_j$	Set of projections of P_j, $\left\{\mathrm{p}_j^i\right\}$.
$\mathbf{p}_j^{\mathrm{n}}$	Set of projections of P_j for which the projection is in the front half space, $\left\{\mathrm{p}_j^i \,\middle\vert\, \overrightarrow{\mathrm{C}_i\mathrm{P}_j} \cdot \mathrm{n} < 0\right\}$.
$\mathrm{C}_\star$	Base camera center. Camera center associated with $\mathrm{p}^\star$.
$\mathbf{C}$	Set of camera centers, $\{\mathrm{C}_i\}$. May or may not include $\mathrm{C}_\star$
$\Pi_{\mathrm{i}}^\star$	Base image. Contains $\mathrm{p}^\star$.
$\mathbf{\Pi}_{\mathrm{i}}$	Set of image planes, $\{\Pi_{\mathrm{i}}^i\}$. May or may not include $\Pi_{\mathrm{i}}^\star$
$\ell^\star$	Base line. 3D line passing through $\mathrm{p}^\star$ and $\mathrm{C}_\star$.
$\ell_{\mathrm{e}}^{\star,i}$	Epipolar line. Projection of $\ell^\star$ onto Π_{i}^i.
$\boldsymbol{\ell}_{\mathrm{e}}^\star$	Set of epipolar lines, $\{\ell_{\mathrm{e}}^{\star,i}\}$.
$\mathcal{EP}_k$	Epipolar plane image. Constructed using Π_{e}^k.
$\mathcal{E}_x$	Epipolar image constructed using $\mathrm{p}^\star$. x indexes all $\mathrm{p}^\star$'s.
$\mathcal{T}(\mathrm{P}, \mathrm{I})$	Function which projects P onto I, e.g. $\mathcal{T}(\mathrm{P}_j, \mathrm{I}^i) = \mathrm{p}_j^i$.
$\mathcal{V}(\mathrm{P}, \mathrm{n}, \mathrm{I})$	Ideal image values (no noise or occlusion) at $\mathcal{T}(\mathrm{P}, \mathrm{I})$.
$\mathcal{F}(x)$	Function of the image at point x (e.g. pixel values, features, etc).
$\mathcal{X}(x_1, x_2)$	Matching function. Quality of match between x_1 and x_2.
$\nu(j, \alpha)$	Match quality. Used to analyze $\mathcal{E}$. Also $\nu(\mathrm{P}_j, \mathrm{n}_j)$ and $\nu(\mathrm{S}_j)$.
$\mathrm{d}(\mathrm{p}^i)$	Depth at image point p^i. Distance from C_i to the closest actual surface in the direction $\overrightarrow{\mathrm{C}_i\mathrm{p}^i}$. If low confidence or unknown, then ∞.
$\mathrm{o}(\mathrm{p}_j^i)$	Orientation at image point p_j^i. Orientation of to the closest actual surface in the direction $\overrightarrow{\mathrm{C}_i\mathrm{p}^i}$.
$\mathrm{G}(\mu, \sigma^2, x)$	Gaussian distribution with mean μ and variance σ^2 evaluated at x.
$\{E \mid C\}$	Set of all E's such that C is true.
$p(P \mid C)$	Probability of P conditioned on C.
$\overrightarrow{\mathrm{P}_1\mathrm{P}_2}$	Unit vector in the direction from P_1 to P_2.
M_l	Modeled object. Previously reconstructed.

Table 2.1: Notation used for the basic approach.

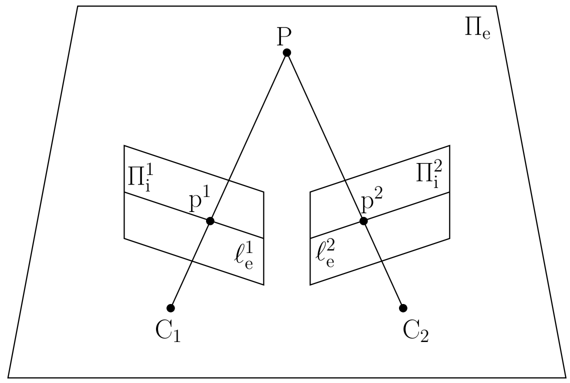


Figure 2-1: Epipolar geometry.

that is present in the i^{th} image Π_{i}^{i}. An epipolar-plane image is built using all of the epipolar lines $\boldsymbol{\ell}_{\mathrm{e}}^{k}$ (the boldface symbol denotes a set, i.e. $\left\{\ell_{\mathrm{e}}^{k,i}\right\}$) from a set of images $\boldsymbol{\Pi}_{\mathrm{i}}$ which correspond to a particular epipolar plane Π_{e}^{k} (Figure 2-2). Since all of the lines $\boldsymbol{\ell}_{\mathrm{e}}^{k}$ in an epipolar-plane image $\mathcal{EP}_k$ are projections of the same epipolar plane Π_{e}^{k}, for any given point p in $\mathcal{EP}_k$, if the corresponding point in any other image Π_{i}^{i} is visible, then it will also be included in $\mathcal{EP}_k$. Bolles, Baker and Marimont exploited this property to solve the correspondence problem for several special cases of camera motion. For example, with images taken at equally spaced points along a linear path perpendicular to the optical axes, corresponding points form lines in the epipolar-plane image; therefore finding correspondences reduces to finding lines in the epipolar-plane image.

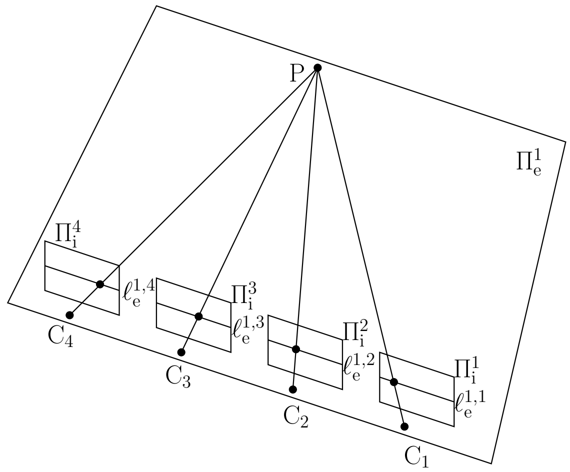


Figure 2-2: Epipolar-plane image geometry.

For a given epipolar plane Π_{e}^{k}, only those images whose camera centers lie on Π_{e}^{k} $\left(\left\{\mathrm{C}_i \;\middle|\; \mathrm{C}_i \cdot \Pi_{\mathrm{e}}^{k} = 0\right\}\right)$ can be included in epipolar-plane image $\mathcal{EP}_k$. For example, using a set of images whose camera centers lie on a plane, an epipolar-plane image can only be constructed for the epipolar plane containing the camera centers. In other words, only a single epipolar line from each image can be analyzed using an epipolar-plane image. In order to analyze all of the points in a set of images using epipolar-plane images, all of the camera centers must be collinear. This can be a serious limitation.

2.2 Epipolar Images

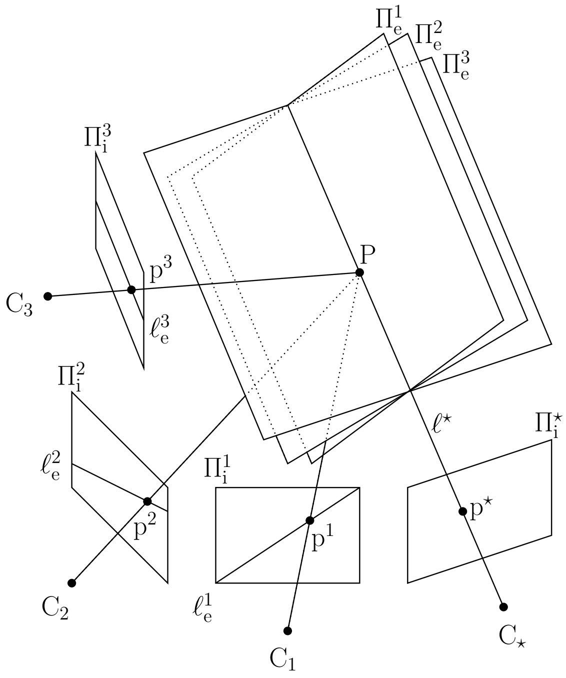


Figure 2-3: Epipolar image geometry.

For our analysis we define an epipolar image $\mathcal{E}$ which is a function of a set of images and a point in one of those images. An epipolar image is similar to an epipolar-plane image, but has one critical difference that ensures it can be constructed for *every* pixel in an arbitrary set of images. Rather than use projections of a single epipolar plane, we construct the epipolar image from the pencil of epipolar planes defined by the line $\ell^\star$ through one of the camera centers $C_\star$ and one of the pixels $p^\star$ in that image $\Pi_i^\star$ (Figure 2-3). Π_e^i is the epipolar plane formed by $\ell^\star$ and the i^{th} camera center C_i. Epipolar line ℓ_e^i contains all of the information about $\ell^\star$ present in Π_i^i. An epipolar-plane image is composed of projections of a plane; an epipolar image is composed of projections of a line. The cost of guaranteeing an epipolar image can be constructed for every pixel is that correspondence information is accumulated for only one point[1] $p^\star$, instead of for an entire epipolar line.

[1]To simplify the presentation in this chapter the discussion will focus on points, however (oriented) points and surfels are interchangeable.

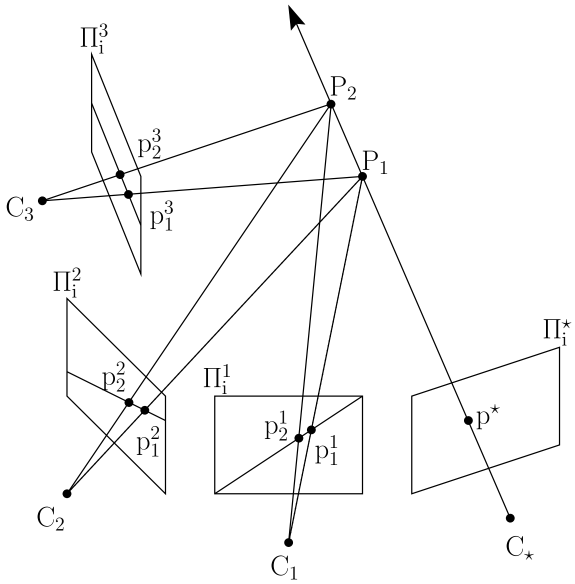


Figure 2-4: Set of points which form a possible correspondence.

To simplify the analysis of an epipolar image we can group points from the epipolar lines according to possible correspondences (Figure 2-4). P_1 projects to p_1^i in Π_i^i; therefore $\mathbf{p}_1$ $(\{p_1^i\})$ has all of the information contained in $\mathbf{\Pi}_i$ about P_1. There is also a distinct set of points $\mathbf{p}_2$ for P_2; therefore $\mathbf{p}_j$ contains all of the possible correspondences for P_j. If P_j is a point on the surface of a physical object and it is visible in both Π_i^i and $\Pi_i^\star$, then measurements taken at p_j^i should (under the simple assumptions of this chapter) match those taken at $p^\star$ (Figure 2-5). Conversely, if P_j is not a point on the surface of a physical object then the measurements taken at p_j^i are unlikely to match those taken at $p^\star$ (Figures 2-6 and 2-7). Epipolar images can be viewed as tools for accumulating evidence about the possible correspondences of $p^\star$. A simple function of j is used to build an ordered set so that $\{P_j\}$ is a set of points along $\ell^\star$ at increasing depths from the image plane.

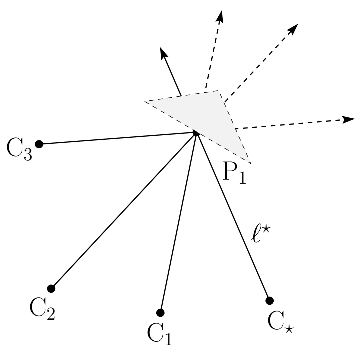


Figure 2-5: P_j is a point on a physical object.

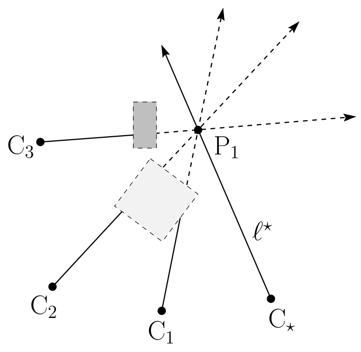


Figure 2-6: Occlusion between C_i and P_j.

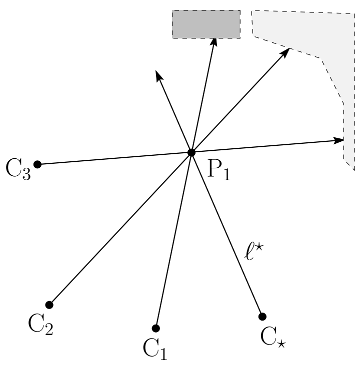


Figure 2-7: Inconsistent background

2.3 Basic Algorithm

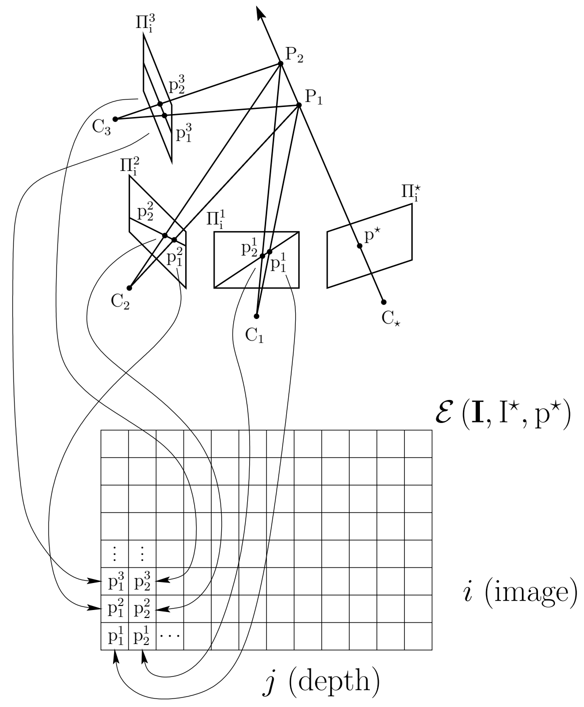


Figure 2-8: Constructing an epipolar image.

An epipolar image $\mathcal{E}$ is constructed by organizing

$$\left\{ \mathcal{F}(p_j^i) \mid \mathcal{F}(\cdot) \text{ is a function of the image} \right\}$$

into a two-dimensional array with i and j as the vertical and horizontal axes respectively. Rows in $\mathcal{E}$ are epipolar lines from different images; columns form sets of possible correspondences ordered by depth[2] (Figure 2-8). The quality $\nu(j)$ of the match[3] between column j and $p^\star$ can be thought of as evidence that $p^\star$ is the projection of P_j and j is its depth. Specifically:

$$\nu(j) = \sum_i \mathcal{X}(\mathcal{F}(p_j^i), \mathcal{F}(p^\star)), \tag{2.1}$$

where $\mathcal{F}(\cdot)$ is a function of the image and $\mathcal{X}(\cdot)$ is a function which measures how well $\mathcal{F}(p_j^i)$ matches $\mathcal{F}(p^\star)$. A simple case is

$$\mathcal{F}(x) = \text{ intensity values at } x$$

[2]The depth of P_j or distance from $C_\star$ to P_j can be trivially calculated from j, therefore we consider j and depth to be interchangeable. We further consider depth and three-dimensional position to be equivalent since we use calibrated images and the three-dimensional position can be trivially calculated from $p^\star$ and the depth.

[3]Interestingly, $\nu(j)$ is related to the tomographic algorithm [Ramachandran and Lakshminarayanan, 1971, Gering and Wells, 1999].

and

$$\mathcal{X}(x_1, x_2) = -\|x_1 - x_2\|^2.$$

Real cameras are finite, and p_j^i may not be contained in the image Π_i^i $\left(\mathrm{p}_j^i \notin \Pi_\mathrm{i}^i\right)$. Only terms for which $\mathrm{p}_j^i \in \Pi_\mathrm{i}^i$ should be included in (2.1). To correct for this, $\nu(j)$ is normalized, giving:

$$\nu(j) = \frac{\displaystyle\sum_{i \,|\, \mathrm{p}_j^i \in \Pi_\mathrm{i}^i} \mathcal{X}(\mathcal{F}(\mathrm{p}_j^i), \mathcal{F}(\mathrm{p}^\star))}{\displaystyle\sum_{i \,|\, \mathrm{p}_j^i \in \Pi_\mathrm{i}^i} 1}. \tag{2.2}$$

Ideally, $\nu(j)$ will have a sharp, distinct peak at the correct depth, so that

$$\operatorname*{argmax}_j(\nu(j)) = \text{ the correct depth at } \mathrm{p}^\star.$$

As the number of elements in $\mathbf{p}_j$ increases, the likelihood increases that $\nu(j)$ will be large when P_j lies on a physical surface and small when it does not. Occlusions do not produce peaks at incorrect depths or false positives[4]. They can however, cause false negatives or the absence of a peak at the correct depth (Figure 2-9). A false negative is essentially a lack of evidence about the correct depth. Occlusions can reduce the height of a peak, but a dearth of concurring images is required to eliminate the peak. Globally this produces holes in the data. While less then ideal, this is not a major issue and can be addressed in two ways: removing the contribution of occluded views, and adding unoccluded views by acquiring more images.

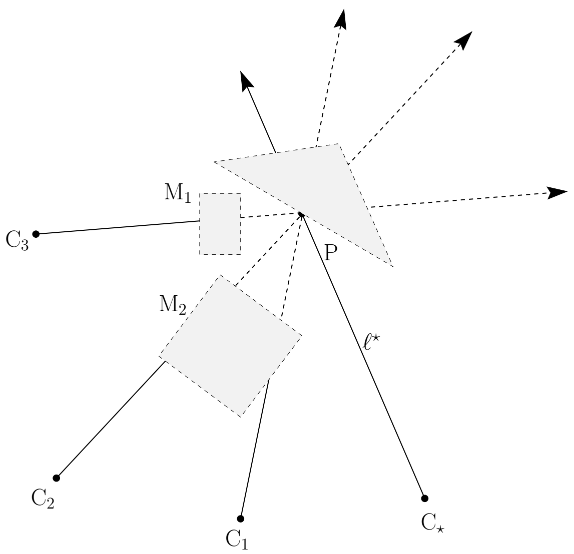


Figure 2-9: False negative caused by occlusion.

[4]Except possibly in adversarial settings.

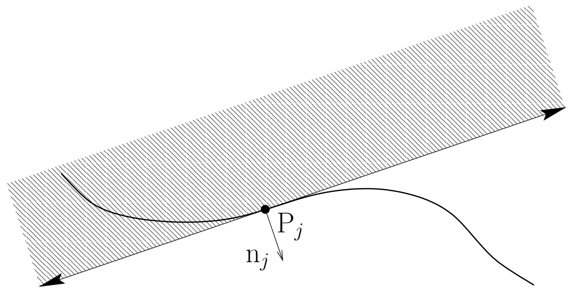


Figure 2-10: Exclusion region (grey) for surfel located at P_j with normal n_j.

A large class of occluded views can be eliminated quite simply. Figure 2-10 shows a surfel located at point P_j with normal n_j. Images with camera centers in the hashed half space cannot possibly have viewed P_j. n_j is not known a priori, but the fact that P_j is visible in $\Pi_i^\star$ limits its possible values. This range of values can then be sampled and used to eliminate occluded views from $\nu(j)$. Let α be an estimate of the normal[5] n_j and $\overrightarrow{C_iP_j}$ be the unit vector along the ray from C_i to P_j, then P_j can only be visible in Π_i^i if $\overrightarrow{C_iP_j} \cdot \alpha < 0$. We denote the set of image points with meet this visibility constraint as $\mathbf{p}_j^\alpha$. At a fundamental level an estimated normal is associated with each reconstructed three-dimensional point (or surfel). For single pixels, the estimated normal is very coarse, but its existence is very useful for grouping individual points (or surfels) into surfaces. Neighborhood information, introduced in Chapter 3 as small planar patches, greatly improves the accuracy of the estimated normals.

If the volume imaged by $\mathbf{\Pi}_i$ is modeled (perhaps incompletely) by previous reconstructions, then this information can be used to improve the current reconstruction. Views for which the depth[6] at p_j^i, or $d(p_j^i)$, is less than the distance from C_i to P_j can also be eliminated. For example, if M_1 and M_2 have already been reconstructed, then the contributions of Π_i^i where $i \in \{1,2,3\}$ can be eliminated from $\nu(j)$ (Figure 2-9). In addition, we weight the contributions based on their forshortening. The updated function becomes:

$$\nu(j,\alpha) = \frac{\sum\limits_{i\in\mathcal{Q}} \left(\overrightarrow{C_iP_j} \cdot \alpha\right) \mathcal{X}(\mathcal{F}(p_j^i), \mathcal{F}(p^\star))}{\sum\limits_{i\in\mathcal{Q}} \overrightarrow{C_iP_j} \cdot \alpha} \tag{2.3}$$

where

$$\mathcal{Q} = \left\{ i \;\middle|\; \begin{array}{l} p_j^i \in \Pi_i^i \\ p_j^i \in \mathbf{p}_j^\alpha \\ d(p_j^i) \geq \|C_i - P_j\|^2 \end{array} \right\}.$$

[5]Actually, α is the azimuth and the elevation is assumed to be 0. This simplifies the presentation without loss of generality.

[6]Distance from C_i to the closest previously reconstructed object or point in the direction $\overrightarrow{C_ip_j^i}$.

Then, if sufficient evidence exists,

$$\underset{j,\alpha}{\operatorname{argmax}}(\nu(j,\alpha)) \;\; \Rightarrow \;\; \begin{cases} j = \text{ depth at } \mathrm{p}^\star \\ \alpha = \text{ estimate of } \mathrm{n}_j \end{cases}.$$

2.4 Probabilistic Formulation

Probabilities can be used to formulate the notion that $\nu(j)$ and $\nu(j,\alpha)$ should be large when P_j lies on a physical surface and small otherwise. Given image data $\mathbf{I}$, q the probability that P_j lies on a physical surface, has depth j and orientation α, and is visible in $\Pi_\mathrm{i}^\star$ can be written formally as

$$q = p(\mathrm{d}(\mathrm{p}^\star) = j, \mathrm{o}(\mathrm{p}^\star) = \alpha \,|\, \mathbf{I}). \tag{2.4}$$

Using Bayes' rule gives

$$q = \frac{p(\mathbf{I} \,|\, \mathrm{d}(\mathrm{p}^\star) = j, \mathrm{o}(\mathrm{p}^\star) = \alpha) p(\mathrm{d}(\mathrm{p}^\star) = j, \mathrm{o}(\mathrm{p}^\star) = \alpha)}{p(\mathbf{I})}.$$

Of all the image points in $\mathbf{I}$ only $\mathbf{p}_j^\alpha$ depend upon α and j. The rest can be factored out yielding

$$\begin{aligned} q &= \frac{p(\mathbf{p}_j^\alpha \,|\, \mathrm{d}(\mathrm{p}^\star) = j, \mathrm{o}(\mathrm{p}^\star) = \alpha) p(\mathrm{d}(\mathrm{p}^\star) = j, \mathrm{o}(\mathrm{p}^\star) = \alpha)}{p(\mathbf{p}_j^\alpha)} && (2.5) \\ \log(q) &= \log(p(\mathbf{p}_j^\alpha \,|\, \mathrm{d}(\mathrm{p}^\star) = j, \mathrm{o}(\mathrm{p}^\star) = \alpha)) + && (2.6) \\ &\quad \log(p(\mathrm{d}(\mathrm{p}^\star) = j, \mathrm{o}(\mathrm{p}^\star) = \alpha)) - \log(p(\mathbf{p}_j^\alpha)) \end{aligned}$$

If we assume that the measured pixel values $\mathcal{F}(\mathrm{p}_j^i)$ contain Gaussian noise with a mean value of $\mathcal{F}(\mathrm{p}^\star)$ and a variance of σ^2 and that individual pixel measurements are independent then the first term becomes

$$-\frac{1}{2} \sum_i \left(\left(\mathcal{F}(\mathrm{p}_j^i) - \mathcal{F}(\mathrm{p}^\star) \right)^2 / \sigma^2 + \log\left(2\pi\sigma^2 \right) \right) \tag{2.7}$$

which is very similar to $\nu(j,\alpha)$ (Equation 2.3). The next term of Equation 2.6 is a prior on the distribution of depths and orientations. If a partial model already exists, it can be used to estimate these distributions. Otherwise we make the standard assumption that all j's and α's are equi-probable.

The last term is more challenging; how do we estimate $p(\mathbf{p}_j^\alpha)$? Applying the same independence assumption as above yields

$$p(\mathbf{p}_j^\alpha) = \sum_i \log(p(\mathrm{p}_j^i)).$$

We could assume that all values of $\mathcal{F}(\cdot)$ are equi-probable in all images, making the denominator irrelevant, and end up with a matching function equivalent to Equation 2.3. Another approach is to use Π_i^i to estimate $p(\mathrm{p}_j^i)$. $p(\mathrm{p}_j^i)$ can be thought of as a significance term. The significance of a match between $\mathcal{F}(\mathrm{p}_j^i)$ and $\mathcal{F}(\mathrm{p}^\star)$ is inversely proportional to $p(\mathrm{p}_j^i)$. We use a nearby (in both position and orientation) image Π_i^k instead of Π_i^i to estimate $p(\mathrm{p}_j^i)$. p_j^i, C_i, and C_k define an epipolar plane Π_e^{ki}. To estimate the distribution we consider epipolar line $\ell_\mathrm{e}^{ki,k}$ (projection of Π_e^{ki} onto image plane Π_i^k). $\ell_\mathrm{e}^{ki,k}$ contains all of the possible matches for p_j^i in Π_i^k and we use this set of possible matches and a mixture of Gaussians model to estimate $p(\mathrm{p}_j^i)$ giving:

$$p(\mathrm{p}_j^i) = \frac{\sum\limits_{\mathrm{p}\in\ell_\mathrm{e}^{ki,k}} \mathrm{G}(\mathcal{F}(\mathrm{p}^\star), \sigma^2, \mathcal{F}(\mathrm{p}))}{\sum\limits_{\mathrm{p}\in\ell_\mathrm{e}^{ki,k}} 1}. \tag{2.8}$$

Since $p(\mathrm{p}_j^i)$ depends only upon p_j^i and Π_i^k, it can be computed once as a pre-processing step. Figure 2-11 shows two probability images produced in this manner. Black represents low probability and white high. $p(\mathrm{p}_j^i)$ can be thought of as a uniqueness term - p_j^i matching $\mathrm{p}^\star$ is only significant if $p(\mathrm{p}_j^i)$ is low.

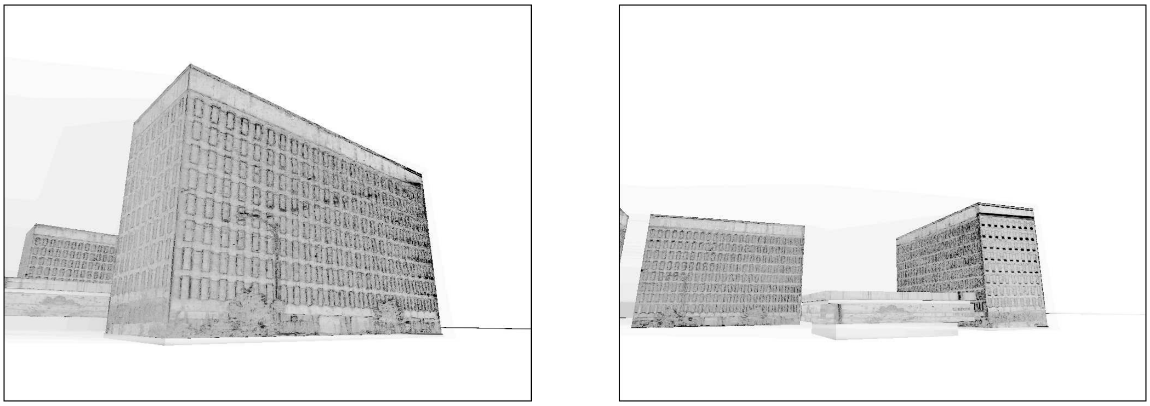


Figure 2-11: Probability images for synthetic data.

Our approach to estimating $p(\mathrm{d}(\mathrm{p}^\star) = j, \mathrm{o}(\mathrm{p}^\star) = \alpha \mid \mathbf{I})$ is conservative. It does not consider subsets of $\mathbf{p}_j^\alpha$ or allow individual elements to be tagged as occluded or outliers. Nevertheless valuable insight was gained from examining $p(\mathrm{p}_j^i)$. In the next section results both with and without considering $p(\mathrm{p}_j^i)$ are presented.

2.5 Results

Synthetic imagery was used to explore the characteristics of $\nu(j)$ and $\nu(j, \alpha)$. A CAD model of Technology Square, the four-building complex housing our laboratory, was built by hand. The locations and geometries of the buildings were determined using traditional survey techniques. Photographs of the buildings were used to extract texture maps which were matched with the survey data. This three-dimensional model was then rendered from 100 positions along a

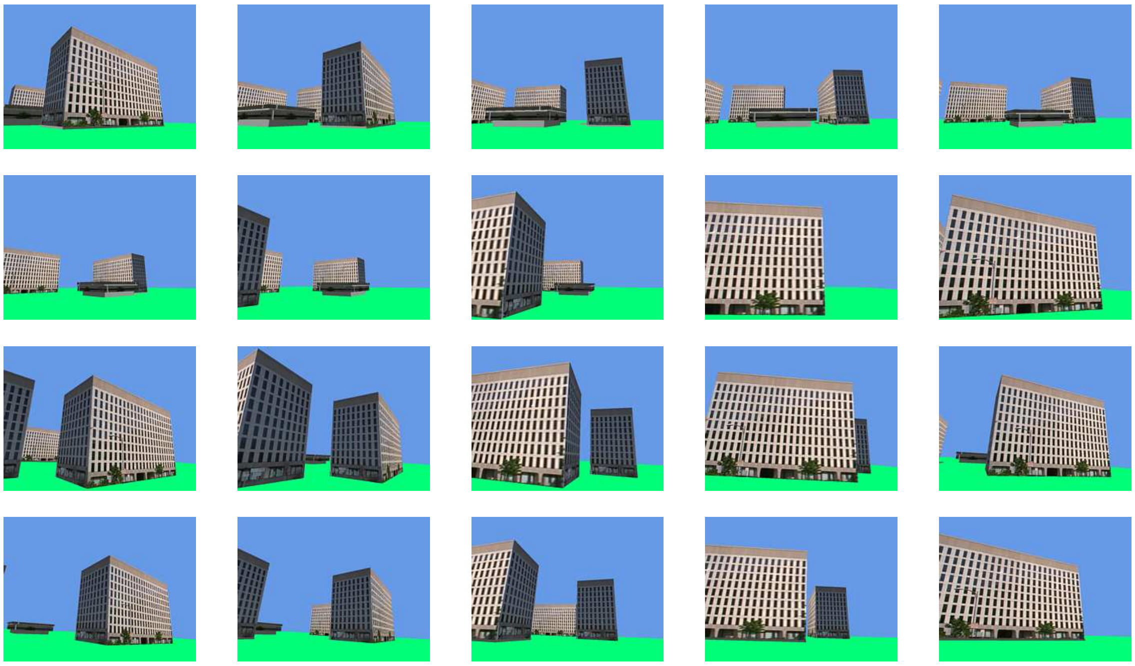


Figure 2-12: Example renderings of the model.

"walk around the block" (Figure 2-12). From this set of images, a $\Pi_i^\star$ and $p^\star$ were chosen and an epipolar image $\mathcal{E}$ constructed. $\mathcal{E}$ was then analyzed using two function, Equations 2.2 and 2.3where

$$\mathcal{F}(x) = \mathrm{rgb}(x) = [\mathrm{r}(x), \mathrm{g}(x), \mathrm{b}(x)] \tag{2.9}$$

and

$$\begin{aligned} &\mathcal{X}([r_1, g_1, b_1], [r_2, g_2, b_2]) = \\ &\quad -\left(\frac{(r_1 - r_2)^2}{\sigma_\mathrm{r}^2} + \frac{(g_1 - g_2)^2}{\sigma_\mathrm{g}^2} + \frac{(b_1 - b_2)^2}{\sigma_\mathrm{b}^2}\right). \end{aligned} \tag{2.10}$$

$\mathrm{r}(x)$, $\mathrm{g}(x)$, and $\mathrm{b}(x)$ are the red, green, and blue pixel values at point x. σ_r^2, σ_g^2, and σ_b^2 are the variances in the red, green, and blue channels respectively. For the synthetic imagery used in this chapter we set $\sigma_\mathrm{r}^2 = \sigma_\mathrm{g}^2 = \sigma_\mathrm{b}^2 = 1$. Elsewhere, the variances are estimated during the internal camera calibration and are assumed to remain constant.

Figures 2-13 and 2-14 show a base image $\Pi_i^\star$ with $p^\star$ marked by a cross. Under $\Pi_i^\star$ is the epipolar image $\mathcal{E}$ generated using the remaining 99 images. Below $\mathcal{E}$ is the matching function $\nu(j)$ (Equation 2.2) and $\nu(j, \alpha)$ (Equation 2.3). The horizontal scale, j or depth, is the same for $\mathcal{E}$, $\nu(j)$ and $\nu(j, \alpha)$. The vertical axis of $\mathcal{E}$ is the image index, and of $\nu(j, \alpha)$ is a coarse estimate of the orientation α at P_j. The vertical axis of $\nu(j)$ has no significance; it is a single row that has been replicated to increase visibility. To the right, $\nu(j)$ and $\nu(j, \alpha)$ are also

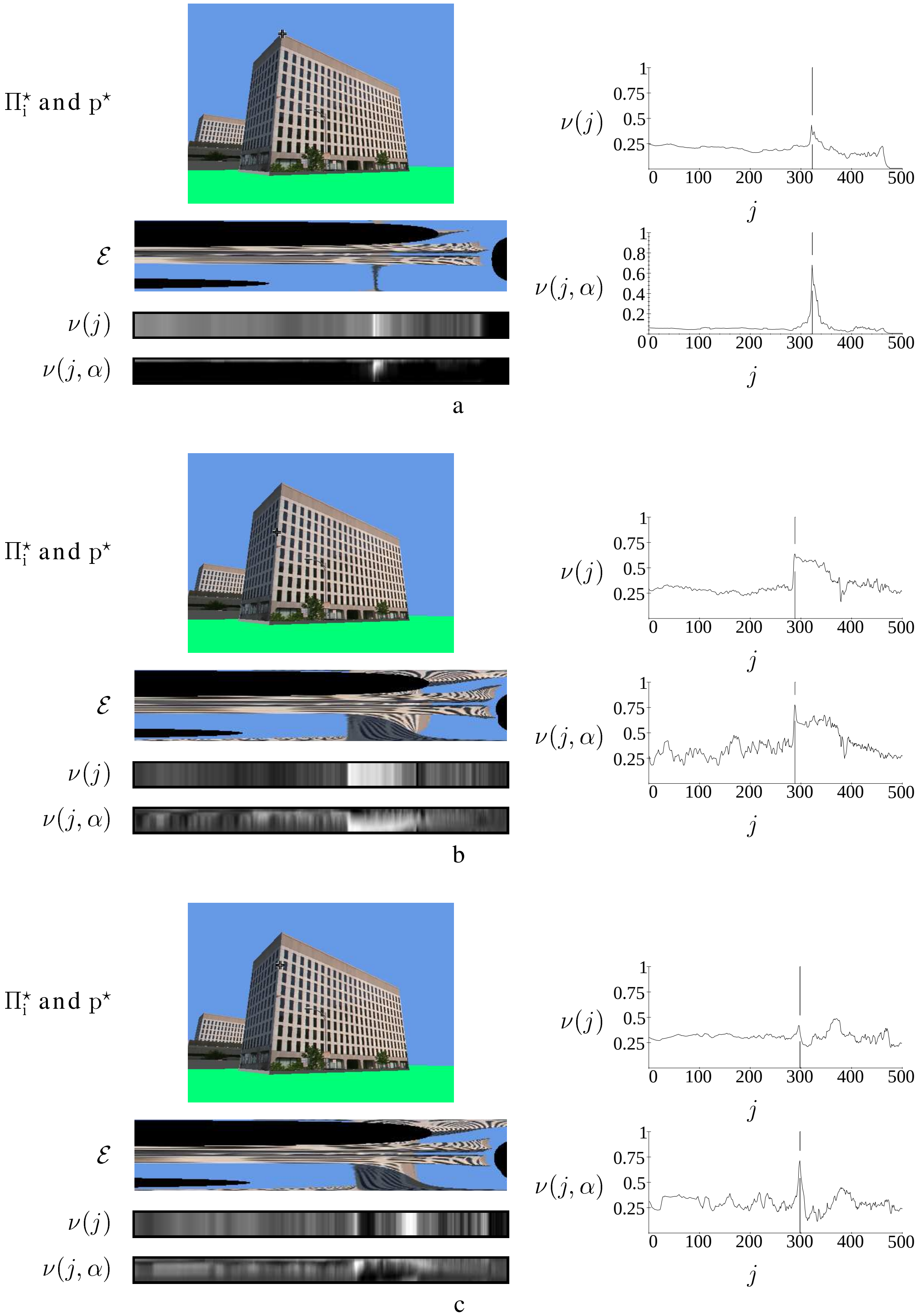


Figure 2-13: $\Pi_i^\star$, $p^\star$, $\mathcal{E}$, $\nu(j)$ and $\nu(j,\alpha)$ (Part I).

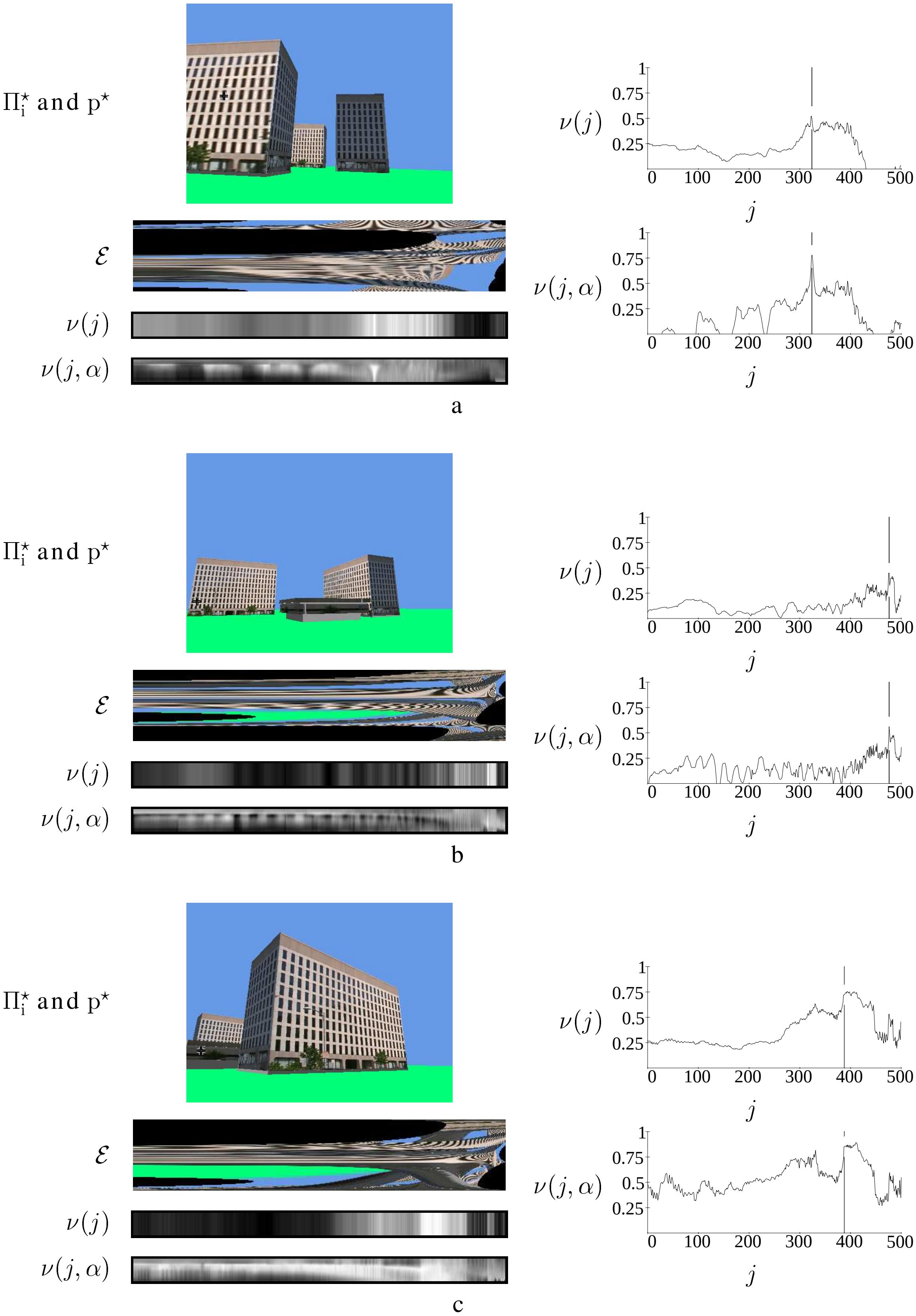


Figure 2-14: $\Pi_i^\star$, $p^\star$, $\mathcal{E}$, $\nu(j)$ and $\nu(j,\alpha)$ (Part II).

shown as two-dimensional plots[7] with the correct depth[8] marked by a line.

Figure 2-13a shows the epipolar image that results when the upper left-hand corner of the foreground building is chosen as $\mathrm{p}^\star$. Near the bottom of $\mathcal{E}$, ℓ_{e}^i is close to horizontal, and p_j^i is the projection of blue sky everywhere except at the building corner. The corner points show up in $\mathcal{E}$ near the right side as a vertical streak. This is as expected since the construction of $\mathcal{E}$ places the projections of P_j in the same column. Near the middle of $\mathcal{E}$, the long side to side streaks result because P_j is occluded, and near the top the large black region is produced because $\mathrm{p}_j^i \notin \Pi_{\mathrm{i}}^i$. Both $\nu(j)$ and $\nu(j,\alpha)$ have a sharp peak[9] that corresponds to the vertical stack of corner points. This peak occurs at a depth of 2375 units[10] ($j = 321$) for $\nu(j)$ and a depth of 2385 ($j = 322$) for $\nu(j,\alpha)$. The actual distance to the corner is 2387.4 units. The reconstructed world coordinates of $\mathrm{p}^\star$ are $[-1441, -3084, 1830]$ and $[-1438, -3077, 1837]$ respectively. The actual coordinates[11] are $[-1446, -3078, 1846]$.

Figure 2-13b shows the epipolar image that results when a point just on the dark side of the front left edge of the building is chosen as $\mathrm{p}^\star$. Again both $\nu(j)$ and $\nu(j,\alpha)$ have a single peak that agrees well with the actual depth. This time, however, the peaks are asymmetric and have much broader tails. This is caused by the high contrast between the bright and dark faces of the building and the lack of contrast within the dark face. The peak in $\nu(j,\alpha)$ is slightly better than the one in $\nu(j)$.

Figure 2-13c shows the epipolar image that results when a point just on the bright side of the front left edge of the building is chosen as $\mathrm{p}^\star$. This time $\nu(j)$ and $\nu(j,\alpha)$ are substantially different. $\nu(j)$ no longer has a single peak. The largest peak occurs at $j = 370$ and the next largest at $j = 297$. The peak at $j = 297$ agrees with the actual depth. The peak at $j = 370$ corresponds to the point where $\ell^\star$ exits the back side of the building. Remember that $\nu(j)$ does not impose the visibility constraint shown in Figure 2-10. $\nu(j,\alpha)$, on the other hand, still has a single peak, clearly indicating the usefulness of estimating α.

In Figure 2-14a, $\mathrm{p}^\star$ is a point from the interior of a building face. There is a clear peak in $\nu(j,\alpha)$ that agrees well with the actual depth and is better than that in $\nu(j)$. In Figure 2-14b, $\mathrm{p}^\star$ is a point on a building face that is occluded (Figure 2-9) in a number of views. Both $\nu(j)$ and $\nu(j,\alpha)$ produce fairly good peaks that agree with the actual depth. In Figure 2-14c, $\mathrm{p}^\star$ is a point on a building face with very low contrast. In this case, neither $\nu(j)$ nor $\nu(j,\alpha)$ provide clear evidence about the correct depth. The actual depth occurs at $j = 387.5$. Both $\nu(j)$ and $\nu(j,\alpha)$ lack sharp peaks in large regions with little or no contrast or excessive occlusion. Choosing $\mathrm{p}^\star$ as a sky or ground pixel will produce a nearly constant $\nu(j)$ or $\nu(j,\alpha)$.

[7] Actually, $\sum_\alpha \nu(j,\alpha) / \sum_\alpha 1$ is plotted for $\nu(j,\alpha)$.

[8] Determined by intersecting $\ell^\star$ with the model.

[9] White indicates the best match, black the worst.

[10] 1 unit is 1/10 of a foot.

[11] Some of the difference may be due to the fact that $\mathrm{p}^\star$ was chosen by hand and might not be the exact projection of the corner.

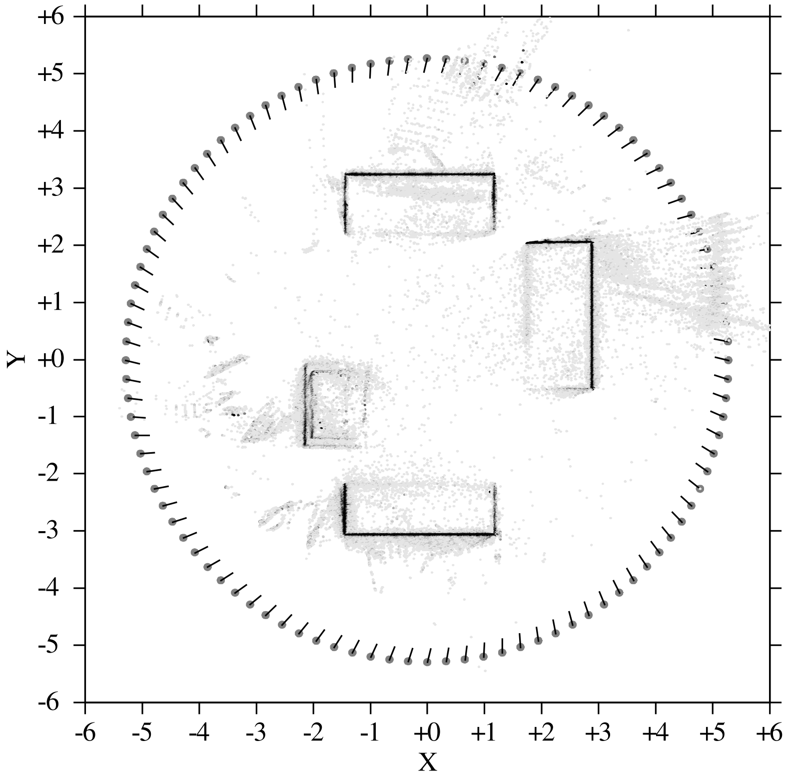


Figure 2-15: Density of reconstructed points.

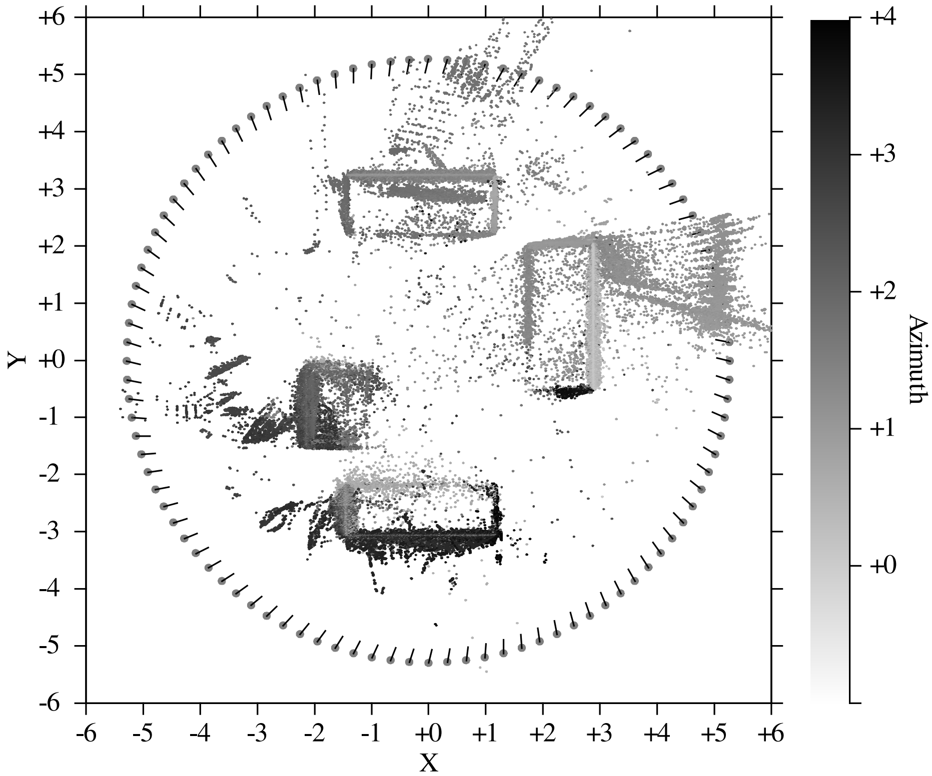


Figure 2-16: Orientation of reconstructed points.

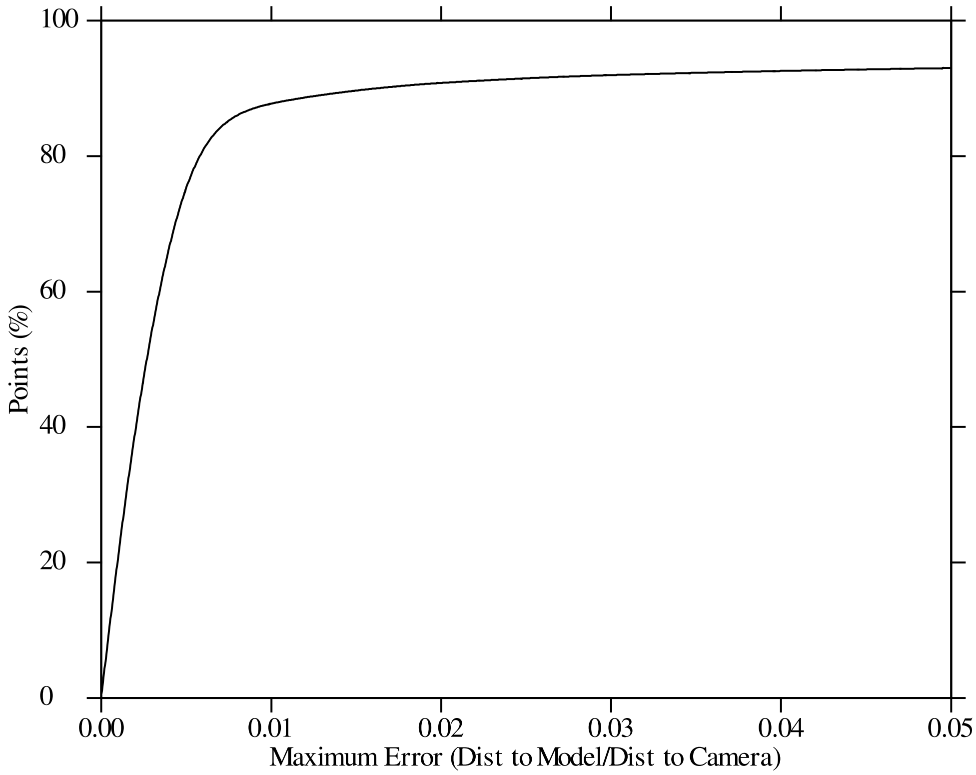


Figure 2-17: Distribution of errors for reconstruction.

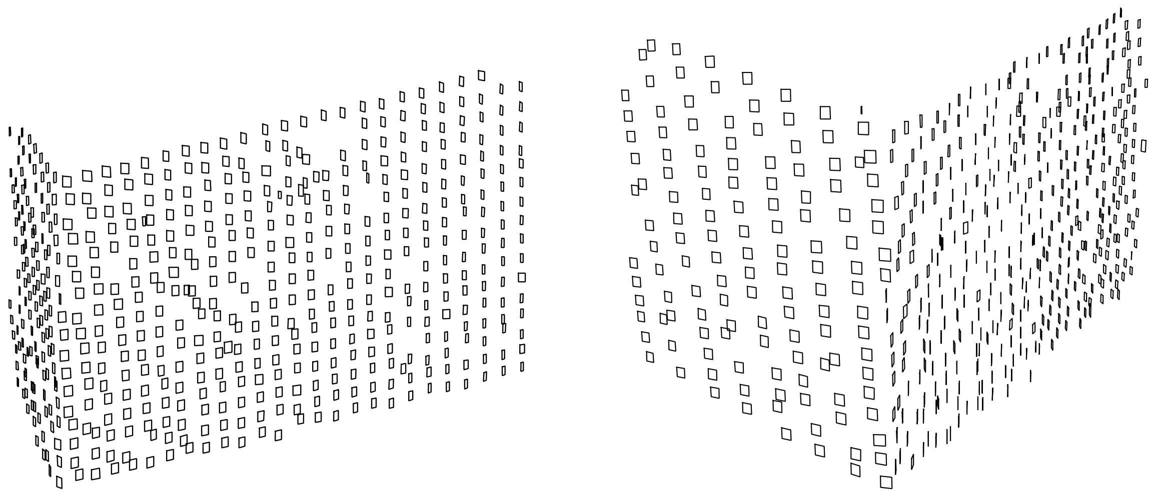


Figure 2-18: Two views of the reconstruction.

Figures 2-15 and 2-16 show the result of running the algorithm on synthetic data. A grey circle and a black line are used mark the location and orientation of each image in the dataset. The x and y coordinates have been divided by 1000 to simplify the plots. For each pixel in the set of images shown in Figure 2-12 an epipolar image was constructed and the point with maximum $\nu(j, \alpha)$ recorded. Points for which $\nu(j, \alpha)$ is at least -2 and $|\mathbf{p}_j|$ is at least 5 are plotted. Global constraints such as ordering or smoothness were not imposed, and no attempt was made to remove low confidence depths or otherwise post-process the maximum of $\nu(j, \alpha)$. In Figure 2-15 the grey levels correspond to the density of reconstructed points, with black the most dense. In Figure 2-16 the grey levels correspond to orientation. The grey scale axis to the right is in units of $\pi/2$. The outlines of the four buildings are clearly visible and the orientations are roughly correct.

What is difficult to appreciate from these plots is the relative numbers of points contributing to the black lines compared to the grey outliers scattered about. Figure 2-17 shows the error distribution for the reconstruction. Plotted values indicate the percentage of reconstructed points which have no more than the indicated error. The error measure is

$$\|\mathrm{P}_j\,(\text{reconstructed}) - \mathrm{P}_j\,(\text{actual})\| \;/\; \|\mathrm{C}_\star - \mathrm{P}_j\,(\text{actual})\|\,.$$

Note that $> 85\%$ of the reconstructed points are within 1% of the correct value. Figure 2-18 shows the lower left corner of the bottom building as viewed from $[-1600, -4000, 1100]$ and $[-2000, -3500, 1200]$. The results are rendered as oriented rectangular surfaces using the reconstructed position and estimated orientation. The size is set so that the projection in $\Pi_i^\star$ is 1 pixel. For clarity, the reconstructed points have been down-sampled by 3 in each direction. The recovered orientations may not be good enough to directly estimate the underlying surface, but they are good enough to distinguish between surfaces. This idea will play an important role in Chapter 5.

Next we explore several modifications to the basic algorithm described in Section 2.3. The synthetic images in Figure 2-12 were rendered with directional lighting in addition to global lighting. As a result, image brightness varies with viewing direction. To compensate we add a brightness correction term γ to the matching function[12]:

$$\begin{aligned}
\mathcal{X}([r_1, g_1, b_1], [r_2, g_2, b_2]) &= -\left(\frac{(\gamma r_1 - r_2)^2}{\sigma_{\mathrm{r}}^2} + \frac{(\gamma g_1 - g_2)^2}{\sigma_{\mathrm{g}}^2} + \frac{(\gamma b_1 - b_2)^2}{\sigma_{\mathrm{b}}^2}\right) \\
\gamma &= \frac{\left(\frac{r_1 r_2}{\sigma_{\mathrm{r}}^2} + \frac{g_1 g_2}{\sigma_{\mathrm{g}}^2} + \frac{b_1 b_2}{\sigma_{\mathrm{b}}^2}\right)}{\left(\frac{r_1^2}{\sigma_{\mathrm{r}}^2} + \frac{g_1^2}{\sigma_{\mathrm{g}}^2} + \frac{b_1^2}{\sigma_{\mathrm{b}}^2}\right)}.
\end{aligned}$$

We also consider two variations of the probabilistic formulation developed in Section 2.4. In one we limit $\log(p(\mathbf{p}_j^\alpha)) \geq -15$ and the other we do not. The

[12]This is similar to normalized correlation, however we impose nonlinear constraints on γ.

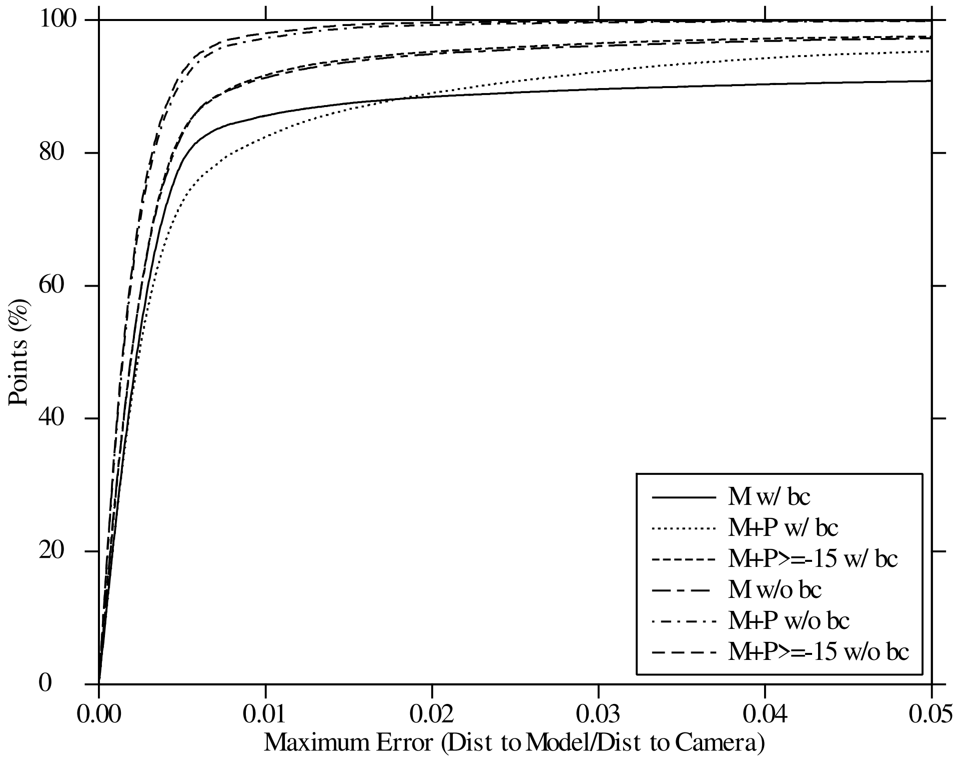


Figure 2-19: Error distribution for variants.

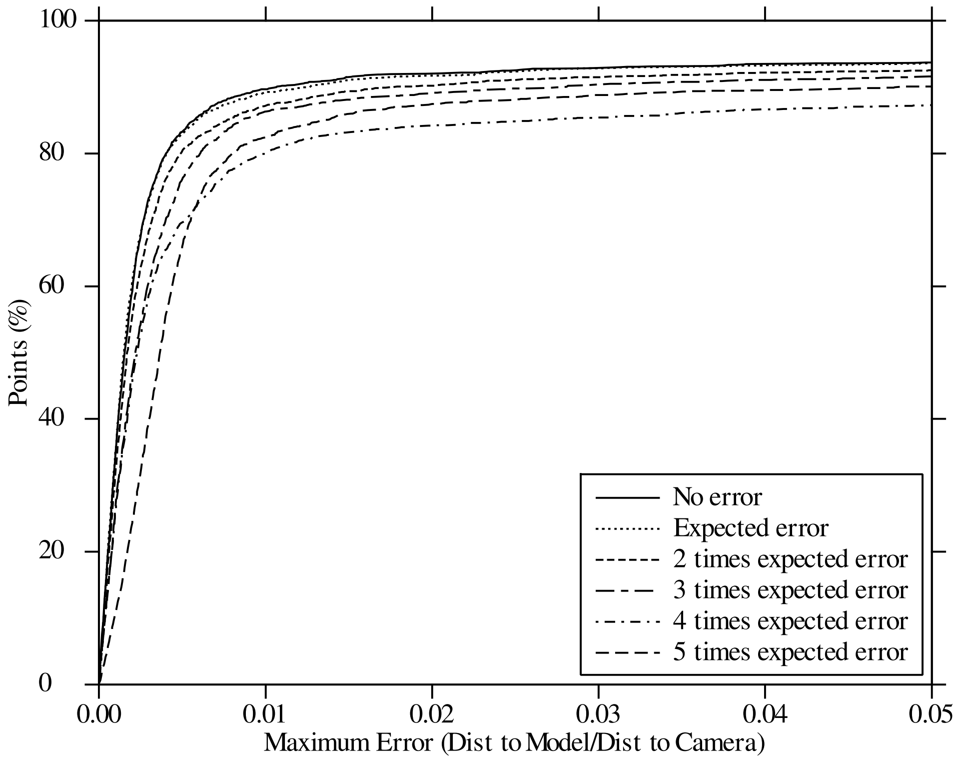


Figure 2-20: Error distribution after adding noise.

upper left image shown in Figure 2-12 was selected as $\Pi_i^\star$. 39% or 67,383 pixels are reconstructible (i.e. non-sky and non-ground). Each pixel in $\Pi_i^\star$ was reconstructed using the six variants shown in Table 2.2. Variants which include brightness compensation are annotated with "w/ bc" and those without "w/o bc". Variants which use the probabilistic formulation are annotated "+P" and those that limit the log probability have ">=-15" added. Figure 2-19 shows error distributions for the variants. Interestingly, variant 6 has the best error distribution, but misses many reconstructible points. Conversely, variant 1 reconstructs nearly all of the points, but sacrifices some accuracy. Variant 3 is the overall best.

Variant	Label	Points Reconstructed	
		Number	Percent
1	M w/ bc	65,582	97
2	M+P w/ bc	63,579	94
3	M+P>=-15 w/ bc	62,564	93
4	M w/o bc	34,710	52
5	M+P w/o bc	24,281	36
6	M+P>=-15 w/o bc	23,485	34

Table 2.2: Performance of algorithm variants.

Finally, we consider the effects of camera calibration noise. Ultimately it is hoped that Argus will achieve positional accuracy on the order of a few centimeters and pose accuracy on the order of a few milliradians. Argus has not yet achieved this design goal, but we use it as a reference. Uniformly distributed noise was added to each coordinate of the position and to the orientation in increments of 2" and $0.1°$ respectively. The results are shown in Figure 2-20.

2.6 Discussion

This chapter defines a construct called an epipolar image and then uses it to analyze large sets of synthetic data. This analysis produces an evidence versus position and surface normal distribution that in many cases contains a clear and distinct global maximum. The location of this peak determines the position. The algorithm presented uses only local calculations and lends itself nicely to parallelization.

Chapter 3

The Challenge of Noisy Data

This chapter extends the approach presented in the previous one to work with noisy data. Specifically data which contains 1) camera calibration error; 2) significant variations in illumination; and, 3) complex occlusion (e.g. building viewed through tree branches) and image noise. With these modifications, our approach shifts from looking for sets of matching points to searching for sets

S_j	j^{th} surfel.
${}^y_x\mathrm{S}_j$	Point in S_j. x and y are indices into S_j. ${}^0_0\mathrm{S}_j \equiv \mathrm{P}_j$.
s^i_j	Projection of S_j onto $\widetilde{\Pi}^i_{\mathrm{i}}$.
$\mathbf{s}_j$	Set of projections of S_j, $\left\{\mathrm{s}^i_j\right\}$.
${}^y_x\mathrm{s}^i_j$	Point in s^i_j. x and y are indices into s^i_j. ${}^0_0\mathrm{s}^i_j \equiv \tilde{\mathrm{p}}^i_j$.
$\mathrm{s}^\star_j$	Base surfel. $\left\{{}^y_x\mathrm{s}^\star_j\right\}$. May contain estimated values.
$\dot{\mathbf{s}}_j$	Matching set of surfels. $\left\{\mathrm{s}^i_j \,\middle|\, \mathrm{s}^i_j \text{ matches } \mathrm{s}^\star_j\right\}$.
$\widetilde{\mathrm{C}}_i$	Estimated center of projection for the i^{th} camera.
$\widetilde{\Pi}^i_{\mathrm{i}}$	Estimated image plane for the i^{th} camera.
$\tilde{\mathrm{I}}^i$	Calibrated image (estimated calibration). $\tilde{\mathrm{I}}^i = \langle \widetilde{\Pi}^i_{\mathrm{i}}, \widetilde{\mathrm{C}}_i \rangle$.
$\tilde{\mathrm{p}}^i_j$	Image point. Projection of P_j onto $\widetilde{\Pi}^i_{\mathrm{i}}$.
$\tilde{\mathbf{p}}_j$	Set of projections of P_j, $\left\{\tilde{\mathrm{p}}^i_j\right\}$.
$\tilde{\tilde{\mathrm{p}}}^i_j$	Noisy Projection of P_j onto $\widetilde{\Pi}^i_{\mathrm{i}}$. Area that P_j projects to given bounded camera calibration error.
${}^v_u\tilde{\tilde{\mathrm{p}}}^i_j$	Point in $\tilde{\tilde{\mathrm{p}}}^i_j$. u and v are indices into $\tilde{\tilde{\mathrm{p}}}^i_j$.
$\dot{\mathrm{p}}^i_j$	${}^{\dot{v}_i}_{\dot{u}_i}\tilde{\tilde{\mathrm{p}}}^i_j$ where $\dot{u}_i$ and $\dot{v}_i$ produce the best match.
$\dot{\mathbf{p}}_j$	Set of matching image points. $\left\{\dot{\mathrm{p}}^i_j \,\middle|\, \dot{\mathrm{p}}^i_j \text{ matches } \mathrm{p}^\star_j\right\}$.
$\dot{\mathrm{P}}_j$	Reconstructed world point. Estimated using $\dot{\mathbf{p}}_j$.
$\tilde{\tilde{\mathbf{p}}}_j$	Set of noisy projections of P_j, $\left\{\tilde{\tilde{\mathrm{p}}}^i_j\right\}$.
$\widetilde{\Pi}^k_{\mathrm{e}}$	The k^{th} estimated epipolar plane.
$\tilde{\ell}^{k,i}_{\mathrm{e}}$	Epipolar stripe. Noisy projection of Π^k_{e} onto $\widetilde{\Pi}^i_{\mathrm{i}}$.
δ^i	Bounded camera calibration error for I^i
$\widetilde{\mathcal{T}}(\mathrm{P}, \mathrm{I}, \delta)$	Noisy projection function, e.g. $\widetilde{\mathcal{T}}(\mathrm{P}_j, \tilde{\mathrm{I}}^i, \delta^i) = \tilde{\tilde{\mathrm{p}}}^i_j$. Note: $\mathcal{T}(\mathrm{P}_j, \tilde{\mathrm{I}}^i, \delta^i) = \tilde{\mathrm{p}}^i_j$.

Table 3.1: Notation used for handling noisy data.

of highly correlated regions. The fact that at a fundemental level we match surfels (small planar regions with position, orientation, and texture) is made explicit in this chapter. The addition of neighborhood information produces several benefits, such as accurate normal estimation, but also makes the epipolar image construct $\mathcal{E}$, described in Chapter 2, less useful. $\mathcal{E}$ can no longer be used to directly visualize possible correspondences. Hypothesizing points along $\ell^{\star}$ also becomes less attractive. This point wil be addressed furter in Chapter 4. Some additional notation used in this chapter is defined in Table 3.1.

3.1 Camera Calibration Errors

The epipolar constraint exploited in Chapter 2 relies on accurate camera calibration. If the calibration is not accurate then the constraint that corresponding points must lie on a pair of epipolar lines (e.g. ℓ_{e}^1 and ℓ_{e}^2 in Figure 2-1) no longer holds. If the calibration error can be bounded then the epipolar lines become epipolar stripes. Consider perturbing C_1 and C_2 in Figure 3-1 about their centers of projection; the intersection of Π_{e} with Π_{i}^1 and Π_{i}^2 sweeps out $\tilde{\ell}_{\mathrm{e}}^1$ and $\tilde{\ell}_{\mathrm{e}}^2$. The area of the epipolar stripe is proportional to the calibration error. What was a one-dimensional search with perfect calibration is now a two-dimensional search. As shown in Figure 3-2, the actual projection of P is not $\tilde{\mathrm{p}}$. If the bounded camera calibration error δ is known then $\tilde{\tilde{\mathrm{p}}}$ can be calculated and $\mathrm{p} \in \tilde{\tilde{\mathrm{p}}}$. The effect can be modeled as a displacement $[\dot{u}, \dot{v}]$ of the actual projection.

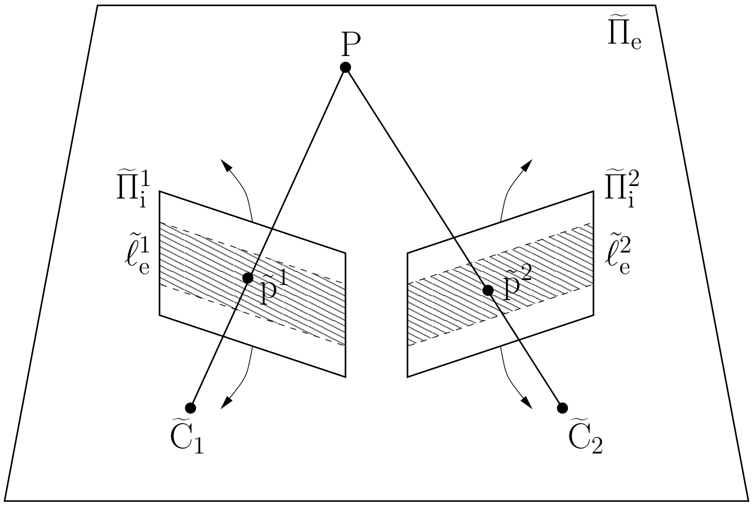


Figure 3-1: Epipolar stripes.

Calibration error complicates the epipolar image $\mathcal{E}$ and its analysis described in Sections 2.2 and 2.3. $\mathbf{p}_j$ does not contain all of the information in $\widetilde{\Pi}_{\mathrm{i}}^i$ about P_j; we must consider $\tilde{\tilde{\mathbf{p}}}_j$. Instead of an array (indexed by j and α) of individual pixels, $\mathcal{E}$ is now composed of two-dimensional regions. Equations 2.2

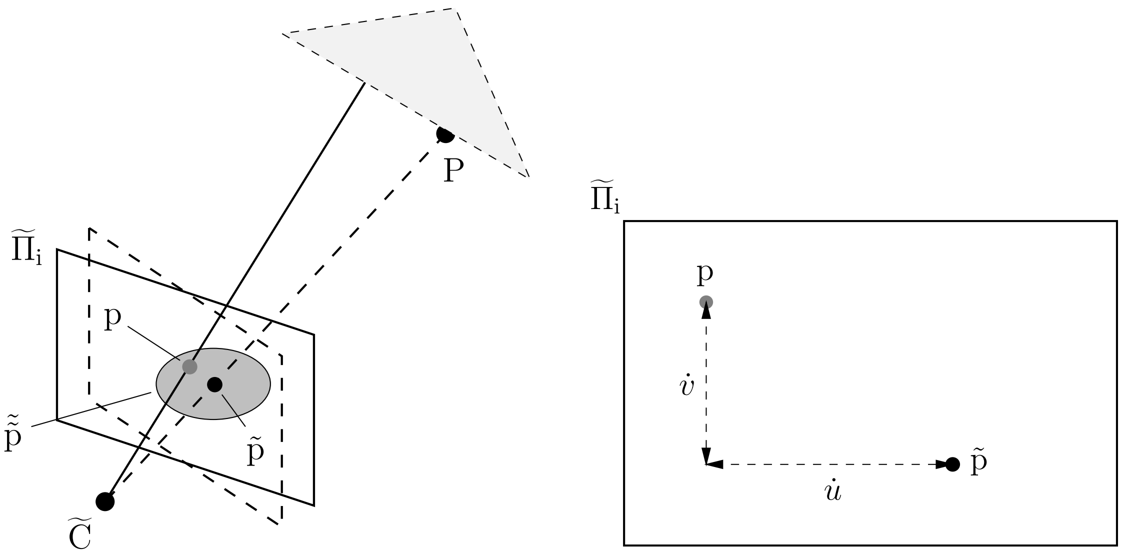


Figure 3-2: Effect of camera calibration error.

and 2.3 become

$$\nu(j) = \frac{\displaystyle\sum_{i \,\left|\tilde{\mathrm{p}}_j^i \in \widetilde{\Pi}_{\mathrm{i}}^i\right.} \max_{u,v} \mathcal{X}(\mathcal{F}({}_u^v\tilde{\mathrm{p}}_j^i), \mathcal{F}(\tilde{\mathrm{p}}^\star))}{\displaystyle\sum_{i \,\left|\tilde{\mathrm{p}}_j^i \in \widetilde{\Pi}_{\mathrm{i}}^i\right.} 1} \tag{3.1}$$

and

$$\nu(j,\alpha) = \frac{\displaystyle\sum_{i\in\mathcal{Q}} \left(\overrightarrow{\widetilde{\mathrm{C}}_i\mathrm{P}_j} \cdot \alpha\right) \max_{u,v} \mathcal{X}(\mathcal{F}({}_u^v\tilde{\mathrm{p}}_j^i), \mathcal{F}(\tilde{\mathrm{p}}^\star))}{\displaystyle\sum_{i\in\mathcal{Q}} \overrightarrow{\widetilde{\mathrm{C}}_i\mathrm{P}_j} \cdot \alpha} \tag{3.2}$$

where

$$\mathcal{Q} = \left\{ i \;\middle|\; \begin{array}{l} \tilde{\mathrm{p}}_j^i \in \widetilde{\Pi}_{\mathrm{i}}^i \\ \mathrm{p}_j^i \in \mathbf{p}_j^\alpha \\ \mathrm{d}(\tilde{\mathrm{p}}_j^i) \geq \left\|\widetilde{\mathrm{C}}_i - \mathrm{P}_j\right\|^2 \end{array} \right\}.$$

Equation 2.8 must also be modified. Epipolar stripe $\tilde{\ell}_{\mathrm{e}}^{ki,k}$ is used to estimate the distribution instead of epipolar line $\ell_{\mathrm{e}}^{ki,k}$ producing

$$p(\mathrm{p}_j^i) = \frac{\displaystyle\sum_{\mathrm{p}\in\tilde{\ell}_{\mathrm{e}}^{ki,k}} \mathrm{G}(\mathcal{F}(\mathrm{p}^\star), \sigma^2, \mathcal{F}(\mathrm{p}))}{\displaystyle\sum_{\mathrm{p}\in\tilde{\ell}_{\mathrm{e}}^{ki,k}} 1}. \tag{3.3}$$

In the last chapter, if a match was found the reconstructed point was automatically P_j. This is no longer the case. $\dot{\mathbf{p}}_j$ represents our best estimate for a set of corresponding points and $\dot{u}_i$ and $\dot{v}_i$ are *shifts* which correct for the calibration error. What world point $\dot{\mathrm{P}}_j$ gave rise to $\dot{\mathbf{p}}_j$? It almost certainly wasn't

P_j. Assuming that the calibration error can be modeled as zero mean additive Gaussian noise then the best estimate of $\dot{\mathrm{P}}_j$ is the one which minimizes the sum of squared calibration errors. Two possible measures of the calibration error are the distance (in three-dimensional space) between $\dot{\mathrm{P}}_j$ and the line of sight through $\dot{\mathrm{p}}_j^i$,

$$\underset{\dot{\mathrm{P}}_j}{\operatorname{argmin}} \sum_{\dot{\mathrm{p}}_j^i \in \dot{\mathbf{p}}_j} \overrightarrow{\tilde{\mathrm{C}}_i \dot{\mathrm{p}}_j^i} \times ((\dot{\mathrm{P}}_j - \tilde{\mathrm{C}}_i) \times \overrightarrow{\tilde{\mathrm{C}}_i \dot{\mathrm{p}}_j^i}) \cdot (\dot{\mathrm{P}}_j - \tilde{\mathrm{C}}_i) \tag{3.4}$$

and the distance (in two-dimensional space) between $\dot{\mathrm{p}}_j^i$ and the projection of $\dot{\mathrm{P}}_j$, $\mathcal{T}(\dot{\mathrm{P}}_j, \tilde{\mathrm{I}}^i)$

$$\underset{\dot{\mathrm{P}}_j}{\operatorname{argmin}} \sum_{\dot{\mathrm{p}}_j^i \in \dot{\mathbf{p}}_j} |\dot{\mathrm{p}}_j^i - \mathcal{T}(\dot{\mathrm{P}}_j, \tilde{\mathrm{I}}^i)|^2. \tag{3.5}$$

The shifts $\{[\dot{u}_i, \dot{v}_i]\}$ introduce additional degrees of freedom. One benefit of this is that $\dot{\mathrm{P}}_j$ can be reconstructed from more than one P_j, allowing $\dot{\mathrm{P}}_j$ to be recovered even if P_j is not very close. This property will be exploited in Chapter 4. The additional degrees of freedom also admit additional false positives. Figure 3-3 shows one example which occurs because similar *looking* points (the upper left corner of a black square) lie within each $\tilde{\tilde{\mathrm{p}}}_j$ (the grey circles). One way

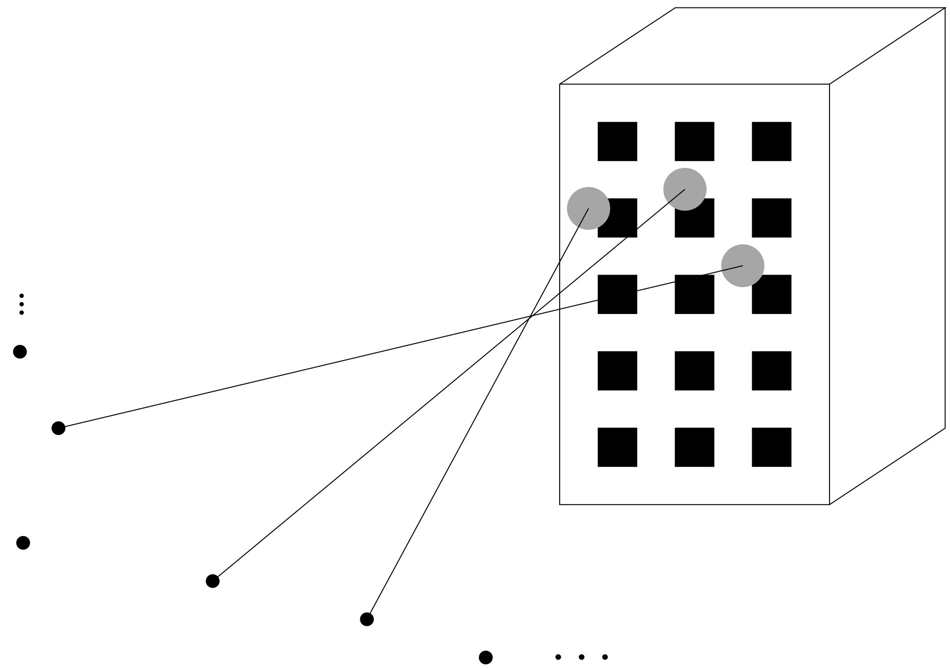


Figure 3-3: False positive induced by compensating for camera calibration error with shifts.

to limit the number of false positives is to constrain the shifts. All of the shifts applied to points in a given image should be consistent with a single camera correction. This will be explored further in Chapter 5.

3.2 Variations in Illumination and Reflectance

The matching function proposed in Equation 2.10 assumes controlled lighting and diffuse reflectance. Only the overall brightness is allowed to change. Natural lighting can vary significantly (e.g. time of day, season, weather, etc.) not only in brightness, but also chromaticity. Real scenes are also not restricted to objects which reflect diffusely. Both of these make it difficult to compare images of the same region taken under differing illumination conditions and from different viewpoints.

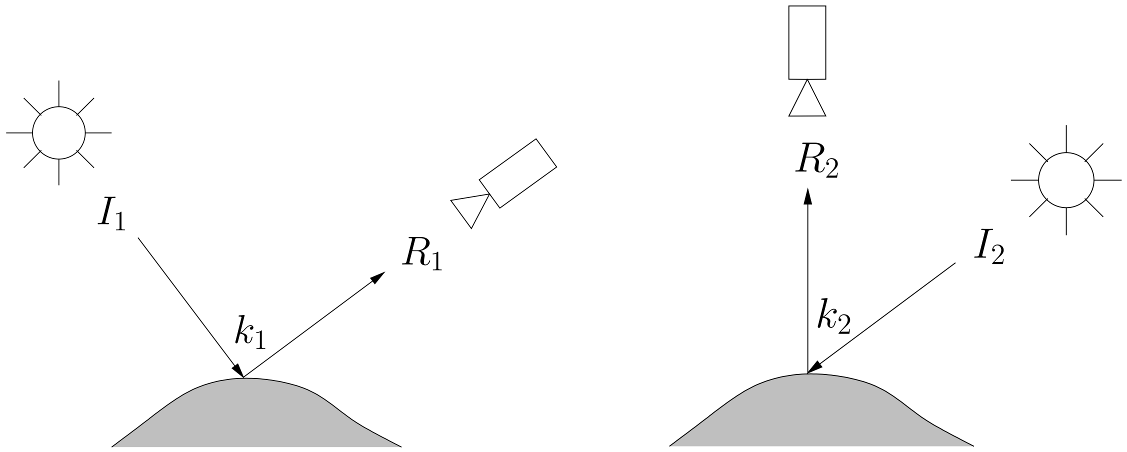


Figure 3-4: Region imaged under different conditions.

A simple linear model can be used to correct images taken under different viewing conditions (illumination and/or reflectance). The radiance R arriving at an image sensor from a given region is simply the irradiance I at the region multiplied by its reflectance k:

$$R = kI. \tag{3.6}$$

and the value measured by the image sensor is:

$$v = \mathrm{f}(R) + d. \tag{3.7}$$

$\mathrm{f}(\cdot)$, an invertible function, and b, an offset, are functions of the image sensor[1]. Figure 3-4 shows a region imaged under different conditions. Clearly, if $k_2 I_2 / k_1 I_1$ is constant for some region then R_2/R_1 will also be constant over the same region. From Equations 3.6 and 3.7 it follows that if for some region

$$I_2 k_2 = I_1 k_1 c \tag{3.8}$$

and

$$\mathrm{f}(xy) = \mathrm{f}(x)\mathrm{f}(y) \tag{3.9}$$

[1]Equation 3.7 is commonly called the image sensor's intensity response function [Vora *et al.*, 1997a, Vora *et al.*, 1997b].

then

$$v_2 = \mathrm{f}(c)v_1 - \mathrm{f}(c)d_1 + d_2$$

or

$$v_2 = mv_1 + d. \tag{3.10}$$

Equation 3.9 is true for typical CCD cameras. Equation 3.10 applies to both diffuse and non-diffuse surfaces and can be calculated independently for each color channel. We often refer to m and d as an illumination correction. It allows us to remap image values obtained under one condition to another and then match them. Substituting Equation 3.10 into Equation 2.10 gives:

$$\begin{aligned} &\mathcal{X}([r_1, g_1, b_1], [r_2, g_2, b_2]) = \qquad (3.11)\\ &- \left(\frac{((m_\mathrm{r} r_1 + d_\mathrm{r}) - r_2)^2}{\sigma_\mathrm{r}^2} + \frac{((m_\mathrm{g} g_1 + d_\mathrm{g}) - g_2)^2}{\sigma_\mathrm{g}^2} + \frac{((m_\mathrm{b} b_1 + d_\mathrm{g}) - b_2)^2}{\sigma_\mathrm{b}^2} \right). \end{aligned}$$

The condition given by Equation 3.8 requires that the product of irradiance and reflectance vary similarly for changes in viewing conditions at each point x in the region $\mathcal{R}$:

$$\forall x \in \mathcal{R} : I_2 r_2 \,/ I_1 r_1 = \mathrm{c}. \tag{3.12}$$

In practical terms, this means that $\mathcal{R}$ must be planar (or well approximated by a plane), uniformly lit, and contain materials with similar reflectance functions. If we are willing to assume that only a limited number of materials appear in each region, then using clustering techniques to calculate the correction removes the requirement that the reflectance functions be similar. For example, in the next chapter, a region containing a window with specular reflection and concrete is successfully corrected with this technique. The spectral content of the irradiance may be changed arbitrarily from one condition to another and the region may contain different colored materials. If Equation 3.9 is not valid and the intensity response function is known, then image values can be converted to radiance and the remapping done there.

A single pixel was sufficient to calculate $\nu(\cdot)$ as presented in Sections 2.3 and 3.1. This is no longer the case once we consider illumination. In order to calculate m and b at least 2 pixels are needed and for good results more should be used. Consider matching images of a surfel (small planar region) S_j with orientation n_j located at P_j. Since orientation is intrinsic to a surfel, Equation 3.1 is no longer useful. We can rewrite Equation 3.2 as

$$\nu(\mathrm{S}_j) = \frac{\displaystyle\sum_{i \in \mathcal{Q}} \left(\overrightarrow{\widetilde{\mathrm{C}}_i \mathrm{P}_j} \cdot \mathrm{n}_j \right) \max_{u,v} \left(\sum_{{}^y_x\mathrm{s}^i_j \in \mathrm{s}^i_j} \mathcal{X}(\mathcal{F}({}^{y+v}_{x+u}\mathrm{s}^i_j), \mathcal{F}({}^y_x\mathrm{s}^\star_j)) \Big/ \sum_{{}^y_x\mathrm{s}^i_j \in \mathrm{s}^i_j} 1 \right)}{\displaystyle\sum_{i \in \mathcal{Q}} \overrightarrow{\widetilde{\mathrm{C}}_i \mathrm{P}_j} \cdot \mathrm{n}_j}. \tag{3.13}$$

In order to obtain a match, S_j must be close (both in position and orientation) to a physical surface and Equation 3.12 must be satisfied. Correcting for viewing conditions allows us to match images taken under different illumination conditions and from different viewpoints, but also admits false positives. Figure 3-5 shows one example. One way to limit the number of false positives is to constrain the correction $(m_\mathrm{r}, d_\mathrm{r}, m_\mathrm{g}, d_\mathrm{g}, m_\mathrm{b}, d_\mathrm{b})$. This is explored in Section 4.2.

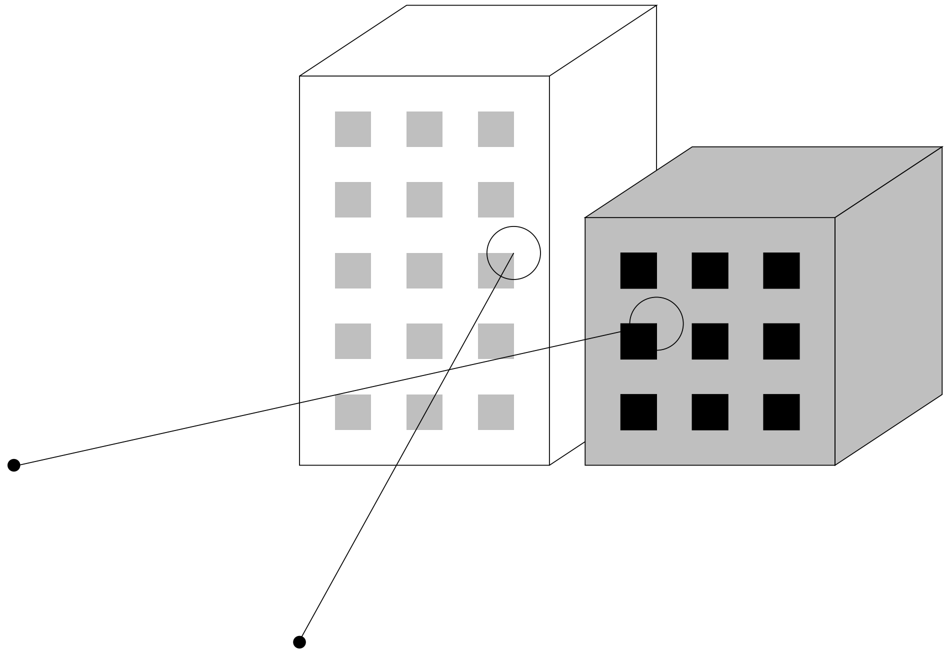


Figure 3-5: False positive induced by correcting for viewing conditions.

There are several drawbacks to matching surfels instead of individual pixels: the cost of computing the match is higher; the shifts introduced in Section 3.1 vary with image position and the distance to the imaged point and therefore are not constant throughout the surfel; and, calculating $p(\mathrm{p}_j^i)$ is problematic. For small calibration errors and small regions it is reasonable to assume that the shifts are piecewise constant and because of the additional information content of a surfel, $p(\mathrm{p}_j^i)$ is less important. On the positive side, the image data of a surfel can be used to efficiently calculate $\dot{u}_i$ and $\dot{v}_i$ and accurately estimate its normal (Sections 4.2 and 4.3).

3.3 Occlusion and Image Noise

Occlusion poses another difficult problem. Figure 3-6 shows two examples typical of urban environments. On the left, the foreground building completely occludes two columns of windows. This view provides no information about the occluded region and if enough other views are completely occluded we will not be able to reconstruct that portion of the background building. The example on the right is potentially more difficult. The tree only partially occludes

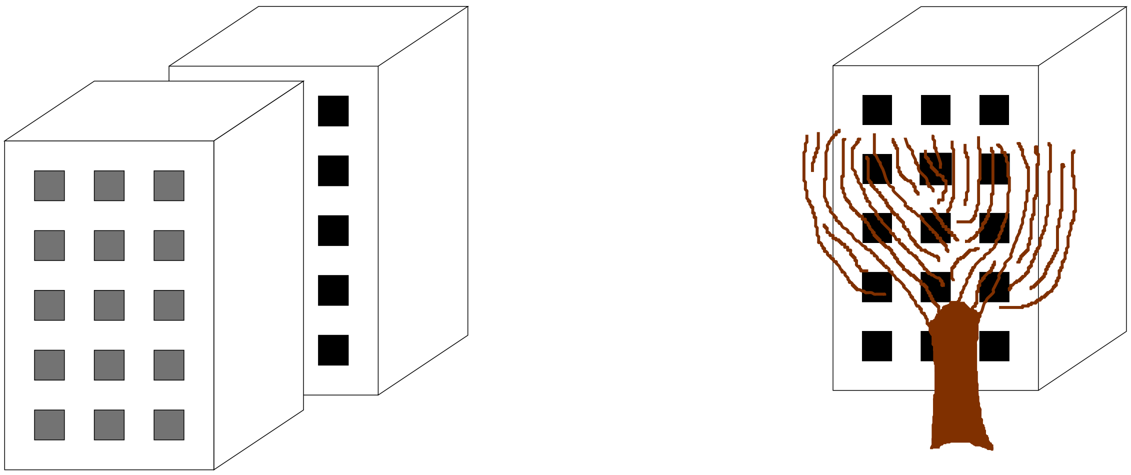


Figure 3-6: Hard (left) and soft (right) occlusion.

the building. The occluded region contains a significant amount of information which we would like to use in the reconstruction process. If we are matching individual pixels this is simple: either a pixel is occluded and does not match well or it is not and matches. On the other hand, multi-pixel surfels can contain both occluded and unoccluded pixels.

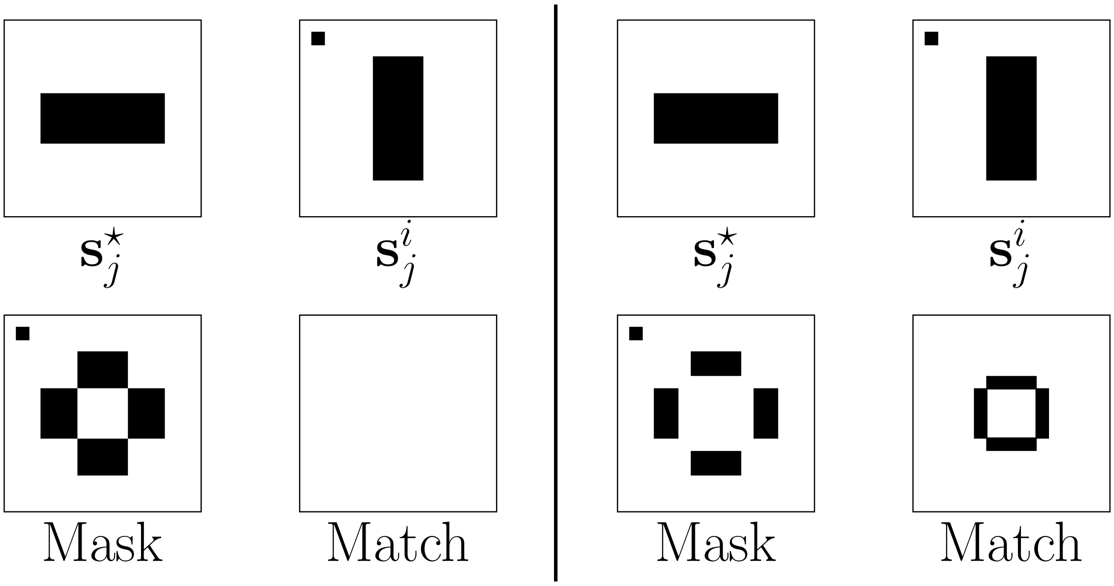


Figure 3-7: Masks.

More generally, either a pixel contains data which is representative of a surface that we intend to model, or it is an outlier. The tree pixels in Figure 3-6 are outliers. Other examples are transient objects (cars, people, etc), imaging aberrations (sensor noise, blooming), and specular reflections. To account for this we use a mask and allow individual pixels to be tagged as outliers and not count towards the match score. Deciding which are outliers and which are not is the hard part. One possible approach, shown on the left side of Figure 3-7, is to tag every pixel which matches poorly as an outlier. Black indicates pixels tagged as outliers and poor matches. Clearly the pair ($\mathrm{s}_j^\star$, s_j^i) should not be considered a match. Less than 20% of the pixels are tagged yet a perfect match is obtained. This approach admits too many false positives, particularly in conjunction with shifts and illumination corrections. A more conservative approach is shown on the right side of Figure 3-7. A pixel is tagged as an outlier only if the match is equally poor throughout a small region (the eight nearest neighbors) around the corresponding pixel. This assumes that the two regions have already been aligned. Notice that the 1 pixel that which is truly an outlier is tagged as such and fewer pixels are inappropriately tagged as outliers. In some cases, the number of outlier pixels is so high that the entire region should be tagged as an outlier. Equation 3.13 can be rewritten to account for both individual pixel and entire region outliers as follows:

$$\nu(\mathrm{S}_j) = \tag{3.14}$$
$$\frac{\displaystyle\sum_{i\in\mathcal{Q}}\left(\overrightarrow{\widetilde{\mathrm{C}}_i\mathrm{P}_j}\cdot\mathrm{n}_j\right)\max_{u,v}\left(\sum_{{}^y_x\mathrm{s}_j^i\in\mathrm{s}_j^i}\mathcal{M}^i({}^y_x\mathrm{S}_j\in\mathrm{S}_j)\mathcal{X}(\mathcal{F}({}^{y+v}_{x+u}\mathrm{s}_j^i),\mathcal{F}({}^y_x\mathrm{s}_j^\star))\Bigg/\sum_{{}^y_x\mathrm{s}_j^i\in\mathrm{s}_j^i}\mathcal{M}^i({}^y_x\mathrm{S}_j\in\mathrm{S}_j)\right)}{\displaystyle\sum_{i\in\mathcal{Q}}\overrightarrow{\widetilde{\mathrm{C}}_i\mathrm{P}_j}\cdot\mathrm{n}_j}$$

where

$$\mathcal{Q}=\left\{ i \;\middle|\; \begin{array}{l} \tilde{\mathrm{p}}_j^i\in\widetilde{\Pi}_{\mathrm{i}}^i \\ \mathrm{p}_j^i\in\mathbf{p}_j^{\mathrm{n}_j} \\ \mathrm{d}(\tilde{\mathrm{p}}_j^i)\geq\left\|\widetilde{\mathrm{C}}_i-\mathrm{P}_j\right\|^2 \\ \mathrm{s}_j^i\neq\text{outlier} \\ \displaystyle\sum_{{}^y_x\mathrm{s}_j^i\in\mathrm{s}_j^i}\mathcal{M}^i({}^y_x\mathrm{S}_j\in\mathrm{S}_j)\Bigg/\sum_{{}^y_x\mathrm{s}_j^i\in\mathrm{s}_j^i}1\geq 0.5 \end{array}\right\}$$

and $\mathcal{M}^i({}^y_x\mathrm{S}_j\in\mathrm{S}_j)$ returns 0 if the pixel has been tagged as an outlier and 1 otherwise.

Equations 3.1, 3.2, 3.13, and 3.15 all match against a distinguished element $\tilde{\mathrm{p}}^\star$ or more generally $\mathrm{s}_j^\star$. This assumes that $\mathrm{s}_j^\star$ is representative of the actual appearance of the underlying world surface. In the presence of occlusion and image noise this is not always the case. Outliers in $\mathrm{s}_j^\star$ are not likely to match the corresponding data in any of the other images resulting in the mismatch penalty being applied $|\mathbf{\Pi}_{\mathrm{i}}|$ times. One way to mitigate this effect is to first estimate $\mathrm{s}_j^\star$ from the image data and then perform the matching. With camera calibration error and variations in the viewing condition, as well as outliers in the image data, this is a difficult task. Instead we successively set $\mathrm{s}_j^\star=\mathrm{s}_j^i$. We can exhaustively test the s_j^i's or if we are willing to occasionally miss a match we can simply try a few of the best.

3.4 Discussion

This chapter introduces several powerful techniques for dealing with data that contains 1) camera calibration error; 2) significant variations in illumination; and, 3) difficult occlusion and image noise. As pointed out above, over-fitting is a concern. The next chapter applies these techniques directly to a large dataset. And the following chapter imposes several geometric constraints to prune false positives.

Chapter 4

From Images to Surfels

Chapter 3 discussed several methods to compensate for noisy data. This chapter will explore these methods in practice. We will focus on two characteristics: the ability to *detect* nearby (in both position and orientation) surfaces and once detected, *localize* their actual position and orientation.

4.1 The Dataset

A Kodak DCS 420 digital camera mounted on an instrumented platform was used to acquire a set of calibrated images in and around Technology Square (the same office complex depicted in the synthetic imagery and Figure 1-4) [De Couto, 1998]. Nearly 4000 images were collected from 81 node points. Other than avoiding inclement weather and darkness, no restrictions were placed on the day and time of, or weather conditions during, acquisition. The location of each node is shown in figure 4-1. At each node, the camera was rotated about the focal point collecting images in a hemi-spherical mosaic (Figure 1-6). Most nodes are tiled with 47 images. The raw images are 1524×1012 pixels and cover a field of view of $41° \times 28°$. Each node contains approximately 70 million pixels. After acquisition, the images are reduced to quarter resolution[1] and mosaiced [Coorg *et al.*, 1998, Coorg, 1998]. Equal area projections of the spherical mosaic from two nodes are shown in Figure 4-2. The top node was acquired on an overcast day and has a distinct reddish tint. The bottom image was acquired on a bright clear day. Significant shadows are present in the bottom image whereas the top has fairly uniform lighting. Following mosaicing, the estimated camera calibrations are refined.

After refinement, the calibration data is good, but not perfect. The pose estimates are within about $1°$ and 1 meter of the actual values[2]. As described in Section 3.1, these errors produce an offset between corresponding points in different images. A $1°$ pose error will displace a feature by over 8 pixels. Our calibration estimates are in an absolute coordinate frame which allows us to integrate images regardless of when or from what source they were collected.

[1] 381×253 pixels.

[2] An interactive tool was used to manually identify a number of correspondences which were then used to bound the camera calibration error.

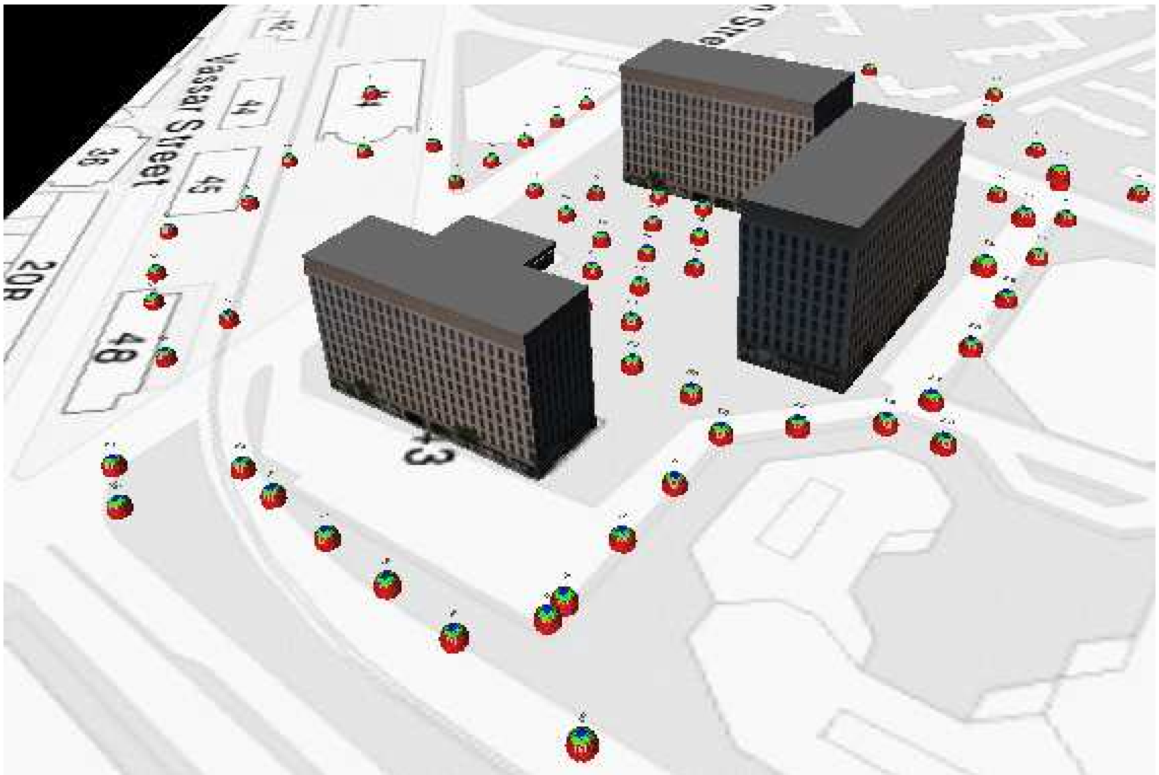


Figure 4-1: Node locations.

This greatly increases the quantity and quality of available data, but because of variations in illumination condition (Section 3.2) also complicates the analysis.

Figure 4-3 and 4-4 show several images from our dataset reprojected[3] onto a surfel which is coincident with an actual surface. The location, orientation, and size of the surfels used are shown in Table 4.1. Surfel 1 was used to generate Figure 4-3 and surfel 2 Figure 4-4. If the camera calibration estimates were perfect and the illumination was constant, the regions in each figure should be identical[4]. The misalignment present in both sets is the result of error in the calibration estimates. Figure 4-3 is representative of the best in the dataset. A large number of the source images have high contrast and none of the regions are occluded. The third row has a distinct reddish tint. The four images in the center of the last row were collected under direct sunlight. And, the last two images were taken near dusk. Figure 4-4 is more typical of the dataset. It is lower in contrast and some of the regions are partially occluded by trees. Figures 4-5 and 4-6 show source images with the reprojected area marked by a circle for several of the regions shown in Figure 4-3 and 4-4. In the worst cases all views of a surfel are similar to the upper left image in Figure 4-6.

---

[3]Using the estimated camera calibration.

[4]Ignoring errors introduced during image formation (discretizing into pixels, etc) and re-sampling.

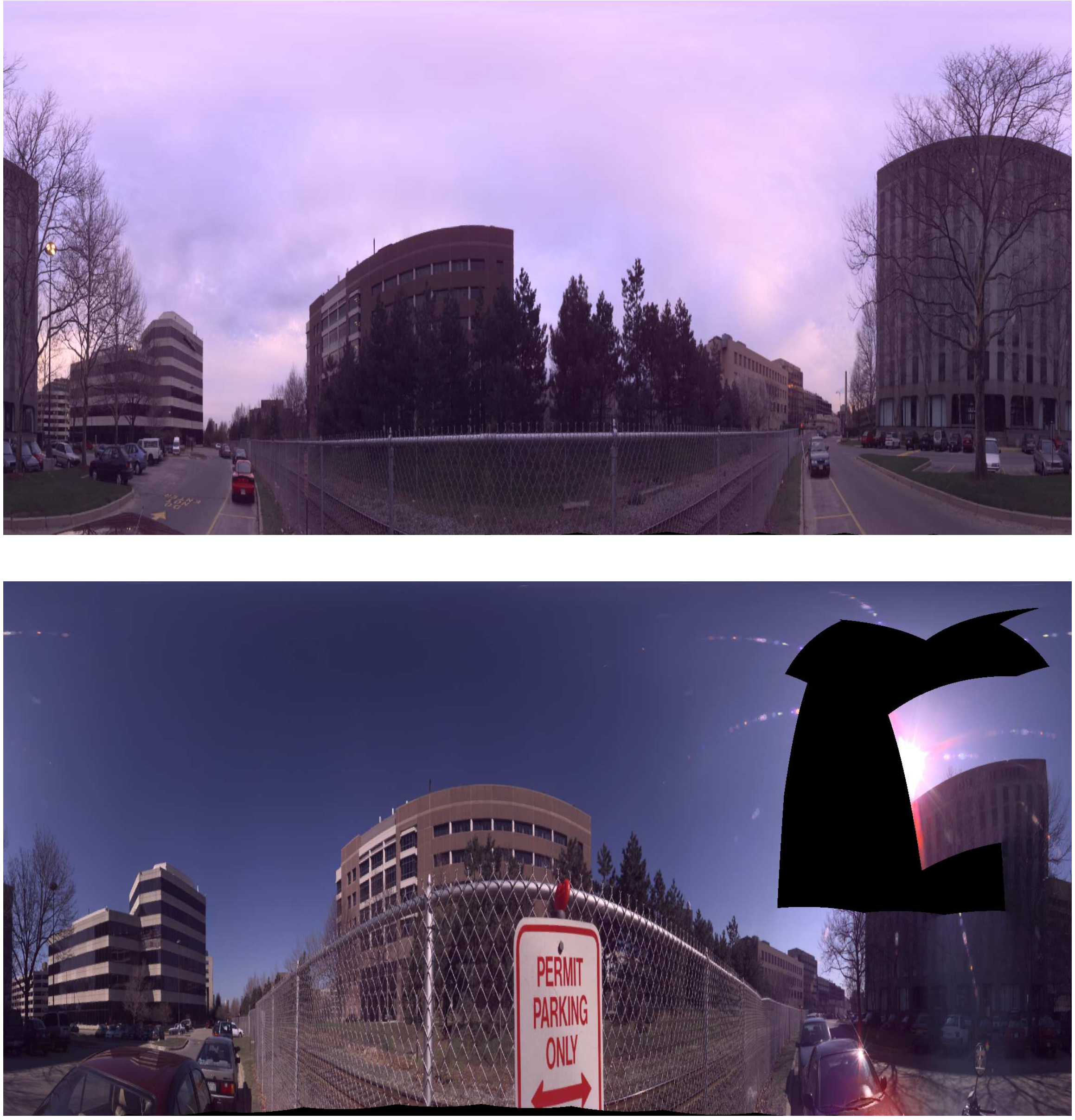


Figure 4-2: Example nodes.

Surfel	Position			Normal			Size (units)		Size (pixels)	
	x	y	z	x	y	z	x	y	x	y
1	-1445	-2600	1200	-1	0	0	110	110	11	11
2	-280	-3078	1500	0	-1	0	110	110	11	11

Table 4.1: Surfel parameters.

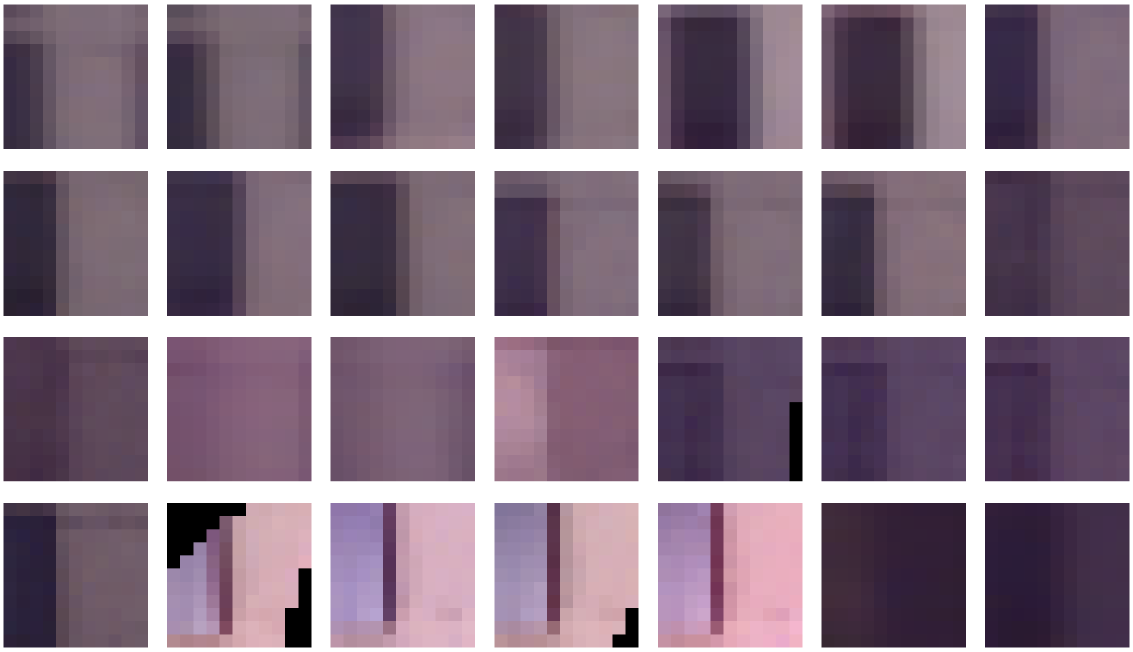


Figure 4-3: Reprojection onto surfel 1 coincident with actual surface.

Figure 4-4: Reprojection onto surfel 2 coincident with actual surface.

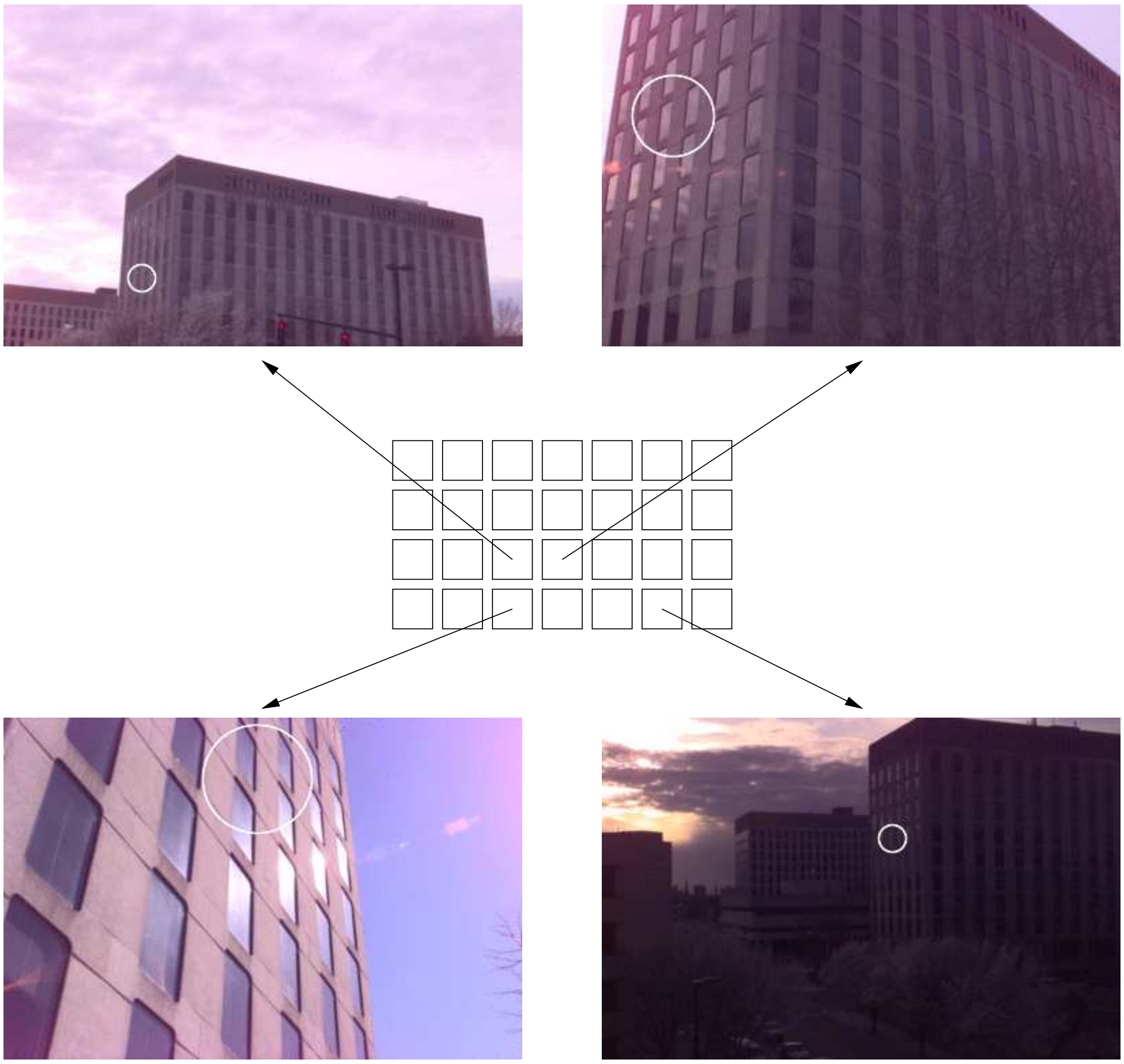


Figure 4-5: Source images for selected regions of surfel 1.

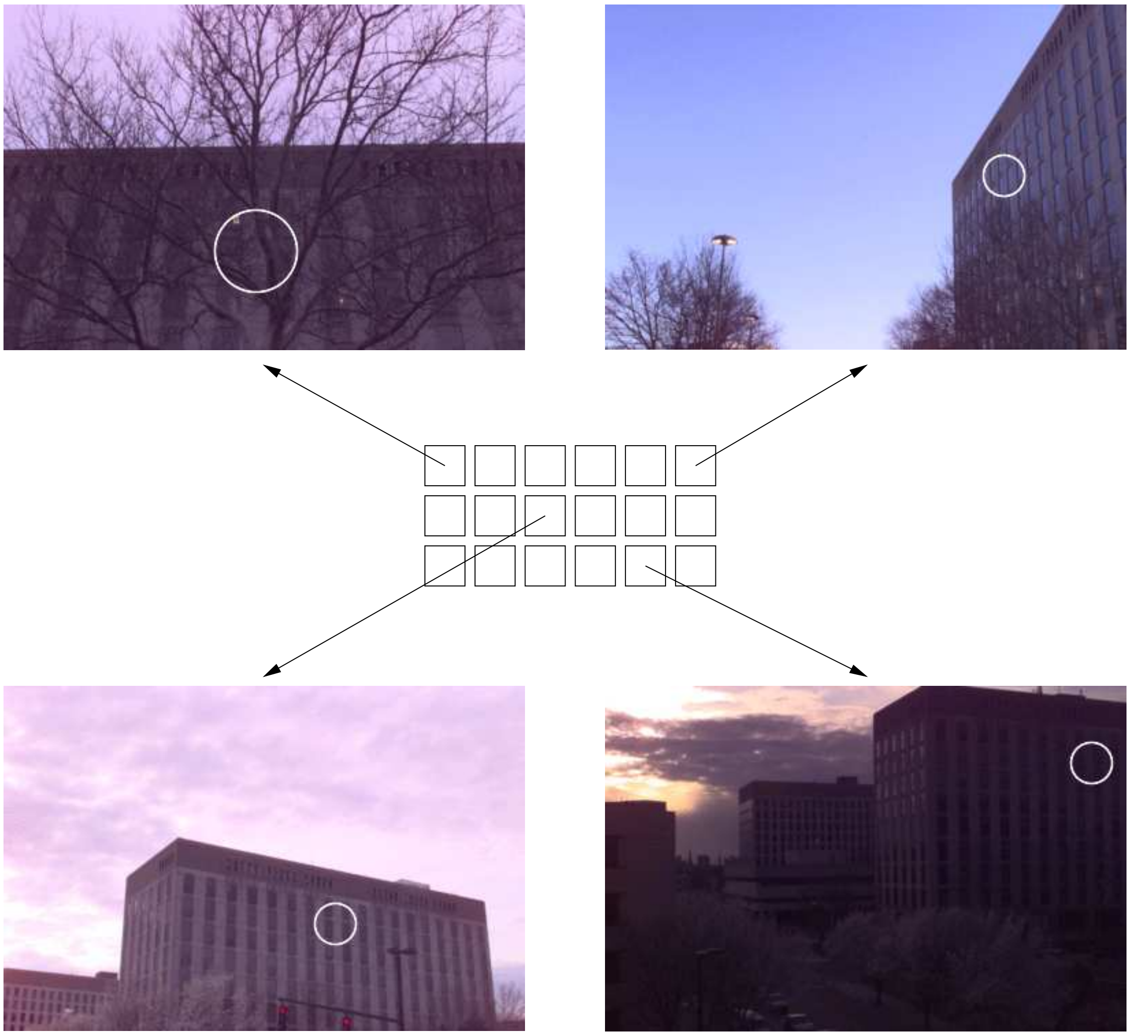


Figure 4-6: Source images for selected regions of surfel 2.

4.2 Detecting Surfels

This section focuses on using $\nu(\mathrm{S}_j)$ to detect nearby (both in position and orientation) surfaces. The following algorithm lists the major steps in our implementation.

1. Hypothesize surfel S_j in world coordinates.
2. Select images $\tilde{\mathbf{I}}$.
3. Reproject regions $\mathbf{s}_j$.
4. Select next $\mathrm{s}_j^\star$.
5. For each region s_j^i:
   (a) Determine best shift $(\dot{u}_i, \dot{v}_i)$.
   (b) Estimate color correction $(m_\mathrm{r}, d_\mathrm{r}, m_\mathrm{g}, d_\mathrm{g}, m_\mathrm{b}, d_\mathrm{b})$.
   (c) Calculate best match. $\sum_{{}^y_x\mathrm{S}_j \in \mathrm{S}_j} \mathcal{X}(\mathcal{F}({}^{y+\dot{v}_i}_{x+\dot{u}_i}\mathrm{s}_j^i), \mathcal{F}({}^y_x\mathrm{s}_j^\star)) \Big/ \sum_{{}^y_x\mathrm{S}_j \in \mathrm{S}_j} 1$
6. Evaluate match set $\nu(\mathrm{S}_j)$:
   - If good enough $\Rightarrow$ done.
   - If not $\Rightarrow$ goto 4.

In Chapter 2 P_j's were determined by sampling along $\ell^\star$. In essence, the image data directly determined the test points. 500 j's and 25 α's were evaluated for each $\mathrm{p}^\star$. Given the dataset described above, this would require evaluating $\sim 5 \times 10^{12}$ points. Since our calibration estimates are oriented in an absolute coordinate system, a more efficient approach would be to choose test points in world coordinates. Sampling the volume of interest (from ground to 2000 units and containing all nodes) every 100 units in position and at 8 different orientations would test less than 2×10^6 points - more than six orders of magnitude fewer.

Once a surfel (such as shown in Table 4.1) is selected, we construct $\tilde{\mathbf{I}}$. Only cameras which image the front side of the surfel and are not too close or too far are considered. We use 120 units as the minimum distance and 6000 as the maximum. In addition, for most of the results presented in this thesis we limit $|\tilde{\mathbf{I}}|$ to 20 images. If necessary, $|\tilde{\mathbf{I}}|$ is reduced by choosing the subset which maximizes the variation in viewpoint. We use the projection of $\overrightarrow{\mathrm{S}_j\tilde{\mathrm{C}}_i}$ onto the plane containing the surfel as a measure of the viewpoint. The subset which maximizes the minimum distance between the projections also maximizes the viewpoint variation.

Each image in $\tilde{\mathbf{I}}$ is then reprojected on the the surfel, producing sets of regions $\mathbf{s}_j$ similar to those shown in Figures 4-3 and 4-4. To facilitate choosing $s_j^\star$, we define a region's *interest* as

$$\text{interest} = \left(\overrightarrow{\tilde{C}_i P_j} \cdot \alpha\right) \left(\sum_{u,v} \frac{\sum_{{}^y_x s_j \in s_j} \mathcal{X}(\mathcal{F}({}^{y+v}_{x+u} s_j^i), \mathcal{F}({}^y_x s_j^i))}{\sqrt{u^2+v^2} \sum_{{}^y_x s_j \in s_j} 1} \right) \Bigg/ \sum_{u,v} 1 \tag{4.1}$$

where

$$(u,v) \in \{(-1,-1),(0,-1),(1,-1),(-1,0),(1,0),(-1,1),(0,1),(1,1)\}$$

and then choose the largest one first. This gives priority to regions with interesting textures (i.e. ones which would produce a significant match), better contrast, and unforeshortened views. Only regions with a minimum interest (we typically use 50 as the threshold) need even be considered. In fact, at most we try the top five $s_j^\star$'s. Region 5 (top row, fifth from the left) is the most interesting in Figure 4-3 with a score of 575.5. Region 2 is the most interesting in Figure 4-4 with a score of 34.5[5].

Next we consider finding the best match between $s_j^\star$ and s_j^i. We do this by evaluating:

$$\max_{u,v}\ \epsilon(u,v) \tag{4.2}$$

where

$$\epsilon(u,v) = \sum_{{}^y_x s_j \in s_j} \mathcal{X}(\mathcal{F}({}^{y+v}_{x+u} s_j^i), \mathcal{F}({}^y_x s_j^\star)) \Bigg/ \sum_{{}^y_x s_j \in s_j} 1\ . \tag{4.3}$$

Steps 5a, 5b, and 5c are actually accomplished simultaneously when finding the best match but for our discussion we will consider the steps separately.

Figures 4-7 and 4-8 show $-\epsilon$ versus u, x shift, and v, y shift, for three regions each from Figures 4-3 and 4-4. The left column shows the error near the correct shift plotted as a three-dimensional surface. The center and right columns show the projection onto the x-error plane and y-error plane respectively. Region five from Figure 4-3 is used as $s_j^\star$ for all of the plots in Figure 4-7. The plots are periodic because of the repetitive texture (windows on the on the building). The different periodicity in the x and y directions reflects the horizontal and vertical spacing of the windows. The first, ninth, and last regions of Figure 4-3 were used as s_j^i for the first, second, and third rows of Figure 4-7 respectively. Region two from Figure 4-4 is used as $s_j^\star$ for all of the plots in Figure 4-8. The first, fourth, and eleventh regions of Figure 4-4 were used as s_j^i for the first, second, and third rows of Figure 4-8 respectively.

The s_j^i's used in the first and third rows of Figure 4-7 and the second row of Figure 4-8 originate from cameras which imaged the surfel from a relatively

[5]For this example we have lowered the threshold to 25 otherwise this region would not be considered.

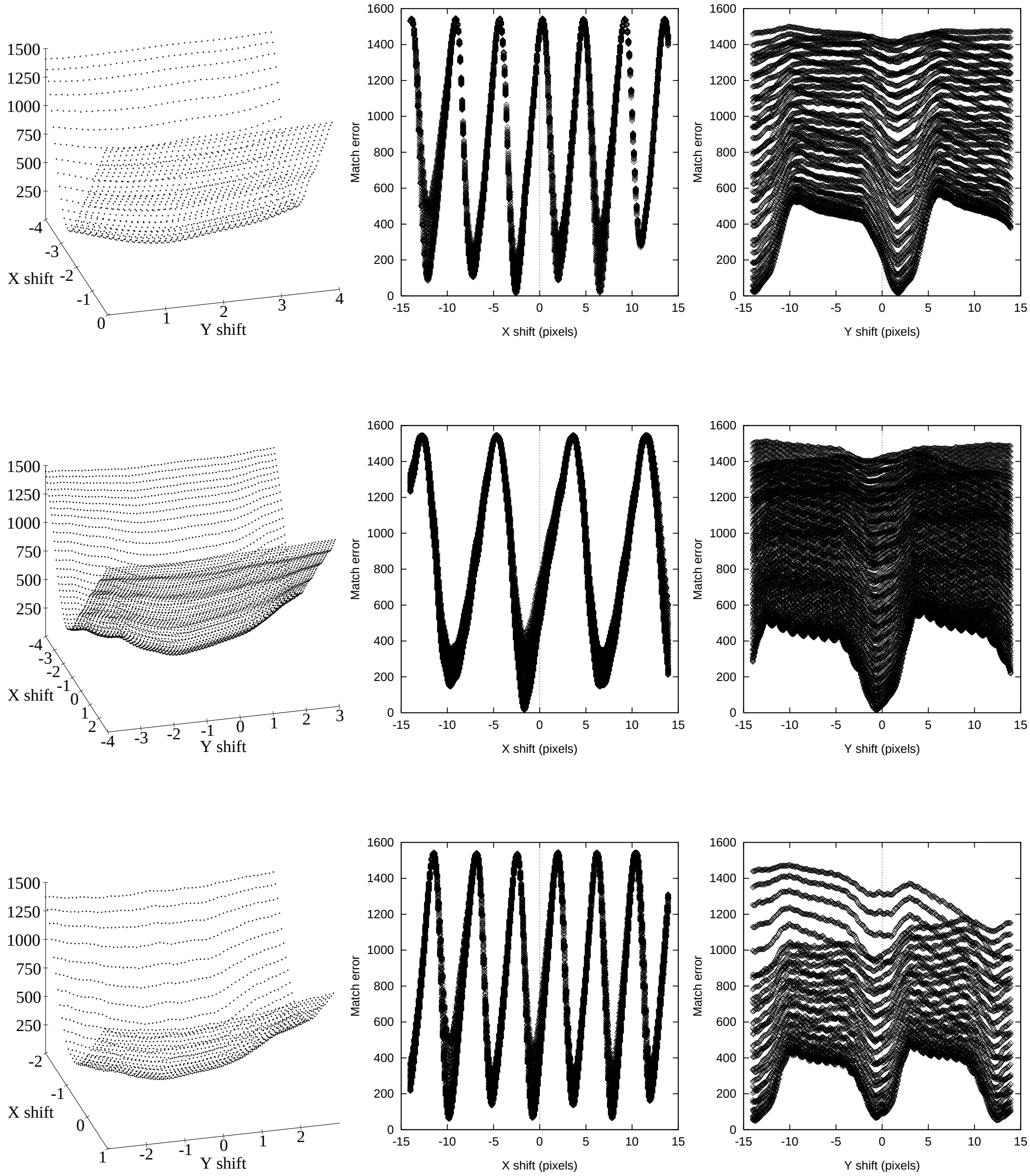


Figure 4-7: Shift error space near correct shift (left), projection onto the x-error plane (center), and y-error plane (right). The periodicity is caused by repetitive texture (windows) on the buildings. The change of periodicity in the x and y directions is produced by the window spacing in the horizontal and vertical directions. The change in periodicity from row to row is the result of foreshortening.

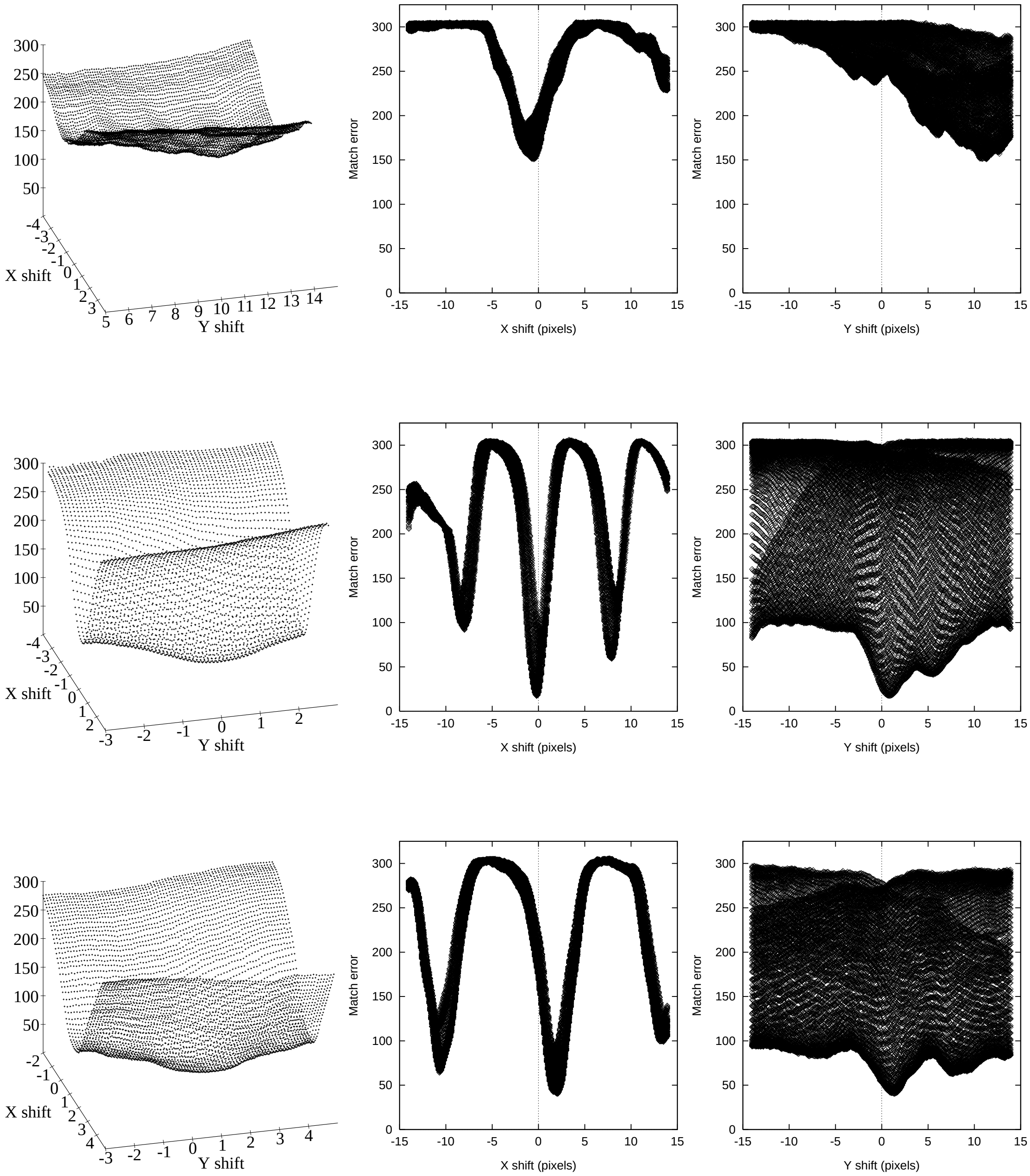


Figure 4-8: Shift error space near correct shift (left), projection onto the x-error plane (center), and y-error plane (right). The periodicity is caused by repetitive texture (windows) on the buildings. The change of periodicity in the x and y directions is produced by the window spacing in the horizontal and vertical directions. The change in periodicity from row to row is the result of foreshortening.

oblique viewpoint. The s_j^i for the third line of Figure 4-7 is a very low contrast image acquired at dusk and the s_j^i for the first line of Figure 4-8 contains significant soft occlusion from tree branches. Note that a shift is required in each case to obtain the best match. All of the plots are smooth with well defined minima making Equation 4.2 a good candidate for optimization techniques [Press *et al.*, 1992]. We have tried a number techniques including direction set methods (e.g. Powell's method) and conjugate gradient methods[6] and all worked well. Multiple minima, however, are a concern. We consider only the nearest minimum and use δ^i to limit the range of valid shifts. For the results presented in this thesis we limited u and v to ± 5. In many cases, only a single minimum exists within this limited range. The multiple minima, particularly those in Figure 4-7, can lead to false positives like the one illustrated in Figure 3-3.

The color correction is determined using linear regression [Press *et al.*, 1992] and limiting the correction. s_j^i is used as the x data and $s_j^\star$ as the y. Each color channel (r, g, and b) is computed separately. Changes in both illumination and reflectance are factored into equation 3.10, however we assume that changes in illumination dominate the correction. Given this assumption, m_r, m_g, and m_b are constrained to be positive. Images collected under a clear sky tend to have a blue tint and those collected under cloudy skies have a reddish tint. While changes in the spectral composition of natural lighting are significant they are limited. Consider the vector $[m_\mathrm{r}, m_\mathrm{g}, m_\mathrm{b}]$; If brightness is the only change between s_j^i and $s_j^\star$, the vector will be in the direction $[1, 1, 1]$. We use the angle between $[m_\mathrm{r}, m_\mathrm{g}, m_\mathrm{b}]$ and $[1, 1, 1]$ as a measure of the variation in spectral composition and limit it to 10°. Further, we limit the overall change in brightness for any channel (m) to a maximum of 10 and a minimum of 1/10. If the best correction exceeds any of these limits, the match is not allowed.

Figures 4-9 and 4-10 show the regions from Figures 4-3 and 4-4 with the shifts and color corrections applied[7]. Once $(\dot{u}_i, \dot{v}_i)$ and $(m_\mathrm{r}, d_\mathrm{r}, m_\mathrm{g}, d_\mathrm{g}, m_\mathrm{b}, d_\mathrm{b})$ have been determined calculating the match score is a straight forward evaluation of $\sum_{{}_x^y\mathrm{s}_j \in \mathrm{S}_j} \mathcal{X}(\mathcal{F}({}_{x+\dot{u}_i}^{y+\dot{v}_i}\mathrm{s}_j^i), \mathcal{F}({}_x^y\mathrm{s}_j^\star)) \Big/ \sum_{{}_x^y\mathrm{s}_j \in \mathrm{S}_j} 1$. To help determine the significance of the match between s_j^i and $s_j^\star$ we use a measure called *uniqueness* which is related to the sharpness of the correlation function peak. A match's uniqueness is the ratio of the best match to the average match of the eight nearest neighbors and

[6]Deriving $\nabla_{u,v}\epsilon(u, v)$ is straight forward and depends only upon the image data and the gradient of the image data. See Appendix B for details.

[7]The correction for the 4^{th} region in row 3 of Figures 4-3 and 4-9 which corrects a specular reflection in the window exceeds the limits described above.

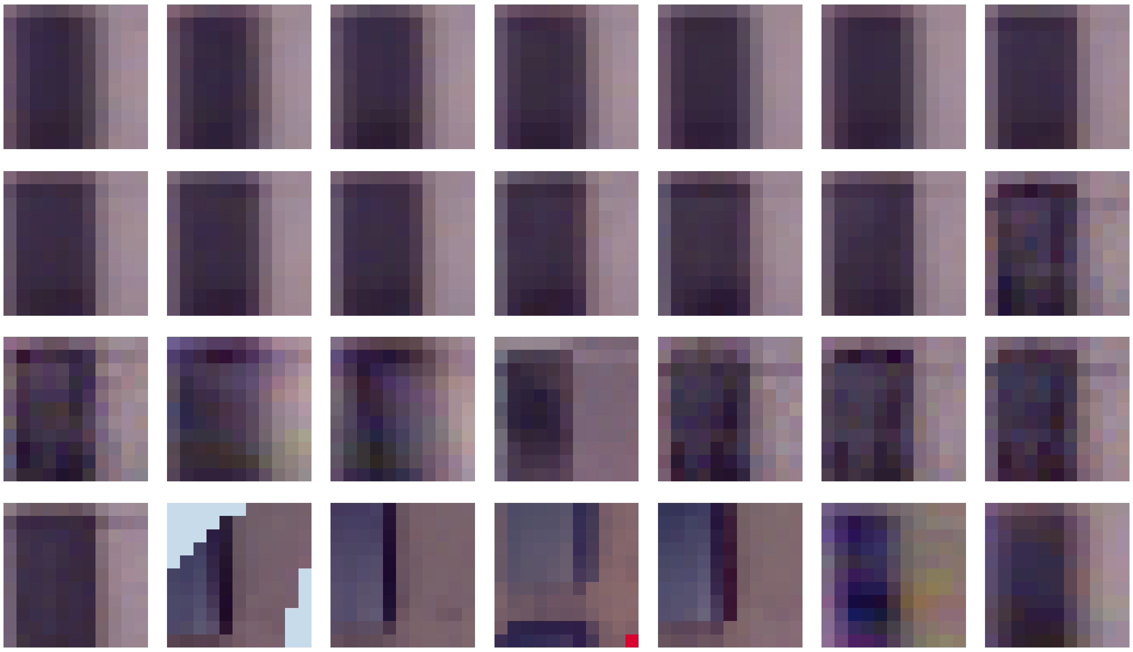


Figure 4-9: Aligned and corrected regions for surfel 1.

Figure 4-10: Aligned and corrected regions for surfel 2.

				Shift		Slope			Intercept		
Region	Match	Score	Unique	x	y	r	g	b	r	g	b
1	X	-14.8	29.2	-2.5	1.7	1.5	1.5	1.6	-29.1	-28.5	-47.2
2	X	-15.0	28.6	-2.3	1.6	1.5	1.5	1.6	-25.8	-28.3	-38.8
3	X	-24.4	18.3	-2.0	-1.9	1.4	1.5	1.5	-35.2	-33.7	-48.4
4	X	-20.3	22.1	-2.0	-1.1	1.5	1.5	1.5	-39.5	-33.9	-46.2
5	X	-0.0	∞	0.0	0.0	1.0	1.0	1.0	0.0	0.0	0.0
6	X	-6.8	64.0	-0.0	0.0	1.0	1.0	0.9	-5.3	-0.5	4.3
7	X	-31.8	14.9	-3.3	-0.7	1.3	1.4	1.5	-15.4	-12.5	-37.0
8	X	-23.7	19.8	-3.2	-0.5	1.3	1.4	1.4	-4.5	-11.9	-14.1
9	X	-15.1	30.0	-1.6	-0.6	1.3	1.4	1.5	-17.4	-19.4	-38.2
10	X	-25.4	18.4	-1.6	0.1	1.3	1.4	1.4	-14.6	-18.8	-23.6
11	X	-34.5	13.9	-3.9	1.4	1.4	1.4	1.6	-31.2	-24.7	-50.2
12	X	-40.6	12.0	-4.3	0.8	1.5	1.5	1.5	-33.1	-32.9	-36.7
13	X	-50.7	9.7	-5.0	1.4	1.2	1.3	1.4	-9.1	-16.3	-21.4
14		-152.4	3.8	-4.6	-0.8	3.7	3.9	3.9	-194.0	-153.0	-222.8
15		-132.8	4.1	-5.2	0.2	4.2	4.0	4.1	-245.1	-156.9	-226.1
16		-329.2	1.7	-1.5	-0.0	5.4	5.3	5.1	-569.0	-384.2	-491.7
17		-1409.6	0.9	2.3	0.2	-1.2	-1.3	-1.4	232.4	195.2	256.5
18		-781.6	1.1	7.0	-3.4	2.0	1.7	1.6	-176.6	-86.9	-102.0
19		-162.4	3.3	-7.0	2.5	3.8	3.9	3.7	-190.0	-140.3	-224.3
20		-190.0	3.1	-7.0	1.1	3.5	3.6	3.4	-164.3	-119.6	-198.7
21		-132.0	4.0	-7.0	0.9	4.0	3.9	3.7	-216.6	-137.1	-218.0
22		-128.3	4.2	-6.6	-1.1	1.4	1.5	1.7	1.7	-5.9	-31.8
23		-904.4	1.1	-5.2	-2.5	1.1	0.9	0.7	-104.0	-55.6	-19.9
24		-960.6	1.1	-2.9	-1.0	0.9	0.9	0.5	-74.4	-54.5	-4.2
25		-1115.9	1.0	-4.0	4.2	0.7	0.5	0.3	-38.2	-0.8	48.2
26		-924.5	1.1	-1.0	-0.4	0.8	0.9	0.7	-72.9	-57.7	-32.2
27		-1052.3	1.0	-3.8	-3.0	5.4	6.2	0.0	-181.7	-128.3	96.1
28	X	-66.4	6.8	-0.7	-0.6	4.8	5.4	4.6	-153.1	-107.5	-187.8

Table 4.2: Match data for surfel 1.

				Shift		Slope			Intercept		
Region	Match	Score	Unique	x	y	r	g	b	r	g	b
1		-177.2	0.9	-1.4	6.0	1.0	1.0	0.9	55.5	49.2	73.2
2	X	-0.0	∞	0.0	0.0	1.0	1.0	1.0	0.0	0.0	0.0
3	X	-21.3	2.7	-0.1	1.1	1.0	1.2	1.2	15.0	-13.5	-24.6
4	X	-17.4	3.2	-0.1	0.7	1.2	1.2	1.2	-3.5	-10.9	-14.6
5		-116.3	1.0	0.0	-0.1	2.1	2.1	1.1	-37.2	-47.1	23.5
6		-249.6	0.8	-7.1	1.9	0.8	0.8	0.4	53.1	31.2	90.1
7		-232.5	0.9	-7.0	-2.1	2.2	0.9	0.1	-86.0	21.4	125.8
8		-233.2	0.9	0.0	0.0	-1.0	-0.9	-1.0	213.9	162.7	233.4
9	X	-31.7	1.9	1.5	1.5	1.5	1.7	1.5	-24.7	-32.9	-19.3
10		-116.2	1.0	-0.2	-0.0	1.7	1.7	1.5	-34.8	-16.0	-3.2
11		-75.3	1.3	-0.3	-1.9	1.9	1.9	1.8	-37.2	-26.2	-18.4
12	X	-41.2	1.8	2.1	1.0	1.7	1.7	1.5	-40.0	-19.4	-16.0
13	X	-44.9	1.8	1.2	2.4	1.6	1.7	1.6	-14.3	-17.4	-13.1
14	X	-59.9	1.5	2.4	2.3	1.7	1.7	1.5	-34.5	-19.5	-10.8
15	X	-46.5	1.8	1.9	0.2	1.3	1.5	1.3	-5.6	-15.6	-5.2
16	X	-44.1	1.8	0.5	0.2	1.4	1.6	1.5	-7.1	-21.1	-17.6
17		-79.8	1.2	-2.2	0.0	4.5	4.3	5.4	-56.7	-0.8	-100.3
18		-69.1	1.2	-1.8	0.8	4.5	5.3	4.4	-50.3	-21.7	-65.0

Table 4.3: Match data for surfel 2.

is defined to be

$$\text{uniqueness} = \frac{\left(\sum_{u,v} \frac{\sum_{{}^y_x\mathrm{S}_j \in \mathrm{S}_j} \mathcal{X}\left(\mathcal{F}\left({}^{y+\dot{v}_i}_{x+\dot{u}_i}\mathrm{s}^i_j\right), \mathcal{F}\left({}^{y+v}_{x+u}\mathrm{s}^\star_j\right)\right)}{\sqrt{u^2+v^2} \sum_{{}^y_x\mathrm{S}_j \in \mathrm{S}_j} 1} \right) \Big/ \sum_{u,v} 1}{\sum_{{}^y_x\mathrm{S}_j \in \mathrm{S}_j} \mathcal{X}\left(\mathcal{F}\left({}^{y+\dot{v}_i}_{x+\dot{u}_i}\mathrm{s}^i_j\right), \mathcal{F}\left({}^y_x\mathrm{s}^\star_j\right)\right) \Big/ \sum_{{}^y_x\mathrm{S}_j \in \mathrm{S}_j} 1} \tag{4.4}$$

where

$$(u, v) \in \{(-1,-1), (0,-1), (1,-1), (-1,0), (1,0), (-1,1), (0,1), (1,1)\}.$$

A region is considered a match if the shift and color correction are within limits, at least half of the points in s^i_j contribute to the match and the match score is good enough and unique enough. For the results presented in this thesis we require the match score to be ≥ -100 and the uniqueness[8] to be ≥ 2. Tables 4.2 and 4.3 show the match data for the regions in Figures 4-9 and 4-10.

Finally, we evaluate the match set. We retain a surfel if the match set contains a minimum number of regions, the regions come from a minimum number of nodes, $\nu(\mathrm{S}_j)$ is greater than or equal to a minimum value and $\mathrm{s}^\star_j$ has an interest which is greater than or equal to a minimum value. For the results presented in this thesis we require the number of regions to be ≥ 6, the number of nodes to be ≥ 5, $\nu(\mathrm{S}_j) \geq -100$, and interest ≥ 50. The sets of regions shown in Figures 4-3 and 4-4 both produce valid matches.

4.2.1 Results

To test the detection characteristics of our algorithm, we evaluated $\nu(\mathrm{S}_j)$ for many surfels near an actual surface. A $100 \times 100 \times 200$ unit volume, the shaded area in Figure 4-11, was selected so that an actual surface divides the surface into two cubes. 396 positions chosen at 20 units intervals were used as test points. Each test point was evaluated every $5°$ for azimuths $\pm 45°$ and every $5°$ for elevations $\pm 45°$. A total of 142,956 surfels were evaluated. Figure 4-12 shows the fraction of the nearly 13,000 surfels tested at each displacement that produced a valid match set. The left side of Figure 4-13 shows the fraction of detections versus displacement and the angle between the actual and tested orientation. The right side shows the fraction of detections versus azimuth and elevation[9]. We also investigated the effect of surfel size on detection rate. As a control, we measured the detections in the empty volume shown as a shaded region in Figure 4-14. An actual surface was selected and tessellated with surfels. The fraction of false negatives ($1-$ the detection rate) for

[8]While growing surfaces (discussed in the next chapter) we allow uniqueness as low as 1.5.

[9]All surfels tested at a particular azimuth and elevation regardless of displacement are included in the fraction.

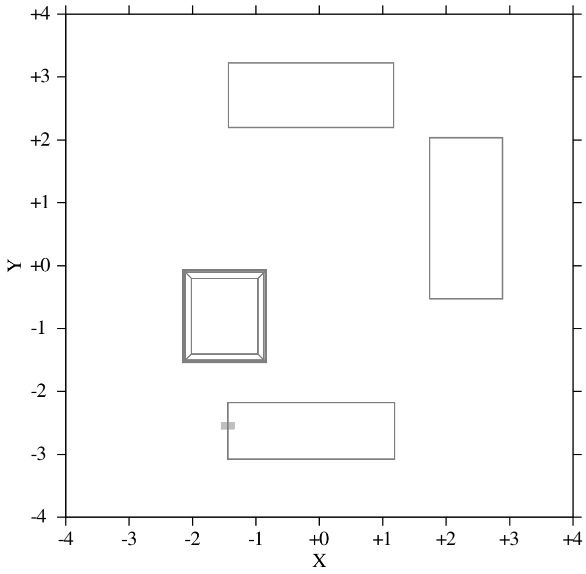


Figure 4-11: Test volume (small shaded rectangle).

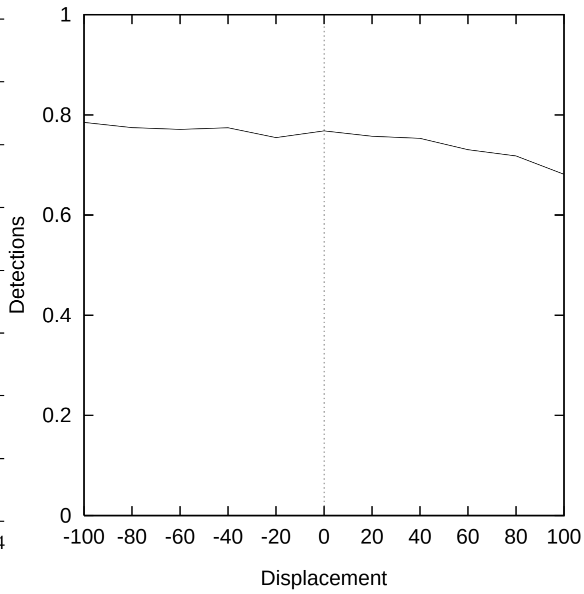
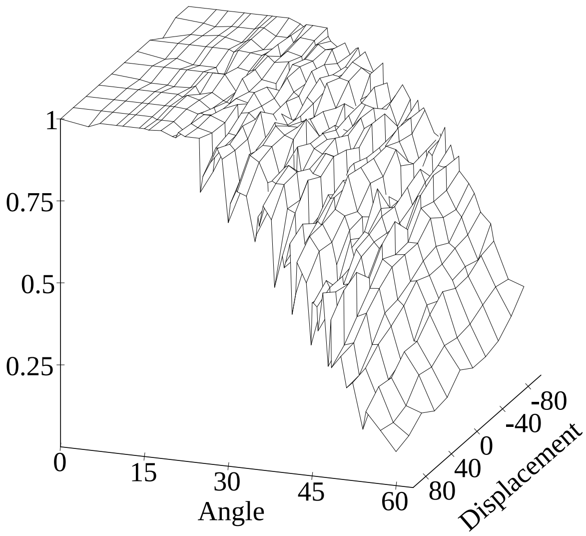


Figure 4-12: Average detection rate.



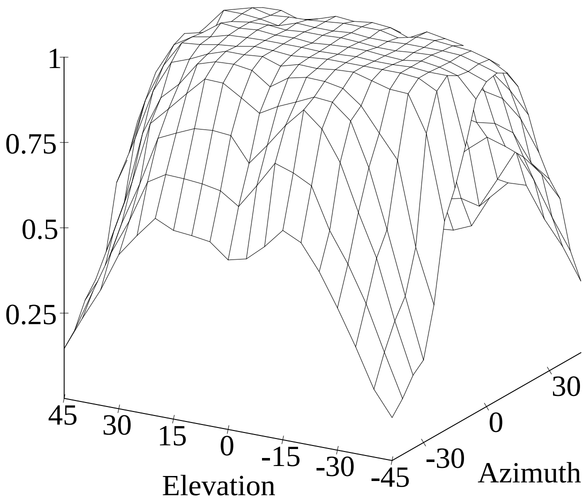


Figure 4-13: Detection rates.

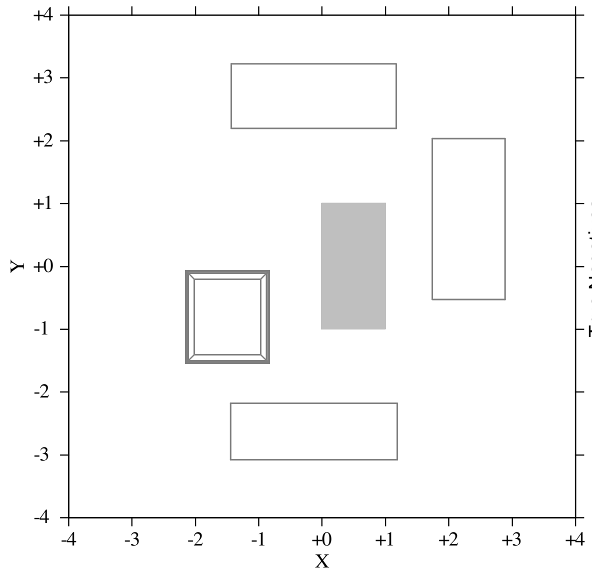


Figure 4-14: Empty test volume (large shaded rectangle).

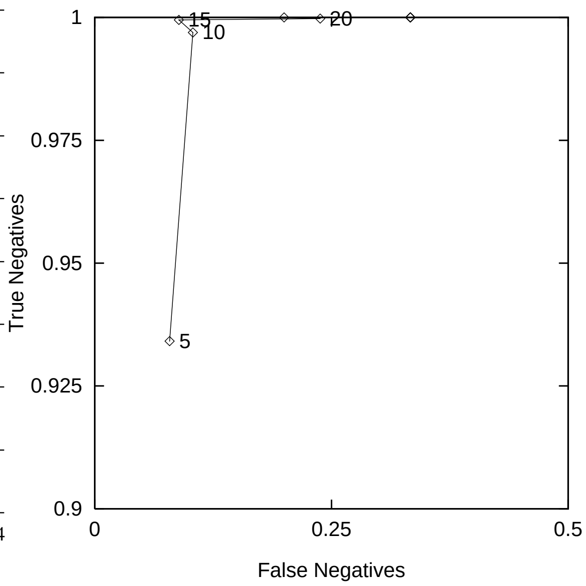
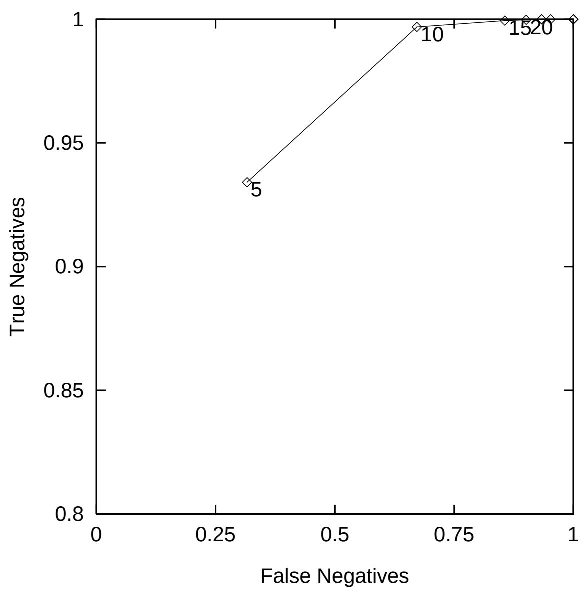


Figure 4-15: Scale effects for an unoccluded surface.



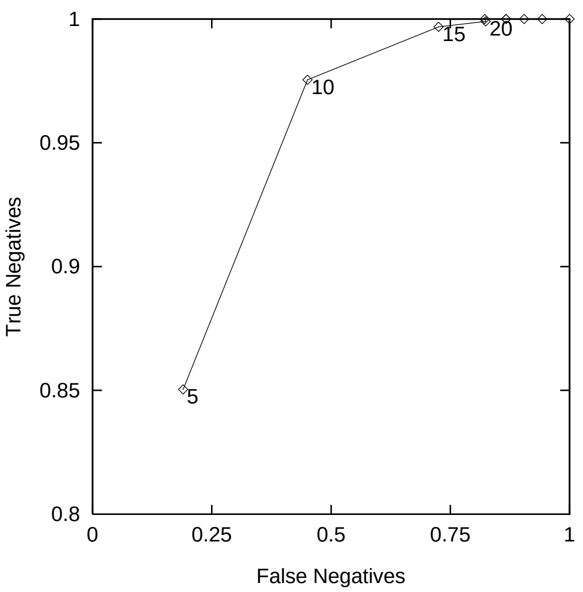


Figure 4-16: Scale effects for an occluded surface without (left) and with (right) masks.

an unoccluded surface versus the fraction of true negatives ($1-$ the detection rate) for the empty volume is shown in Figure 4-15. Surfel sizes from 5 units (lower left corner) to 50 (upper right corner) in increments of 5 units are plotted as *diamonds*. The "bump" in Figure 4-15 is the result of noise and the discrete nature of surfels. Similar curves for an occluded (significant tree coverage) surface are shown in Figure 4-16. The left and right plots in Figure 4-16 are for the same surface. The data shown in the right plot utilizes the mask function described in Section 3.3. A surfel size near 10 units gives the best all around performance. The best surfel size is a function of the data and is related to the size in world coordinates of the significant image features - in our case the windows. Potentially scale-space techniques such as [Li and Chen, 1995] can be used to determine the best surfel size.

4.3 Localizing Position and Orientation

As shown in the previous section, our method is capable of detecting surfaces a significant distance (both in position and orientation) from S_j. The algorithm described so far produces a set of corresponding textures, $\dot{\mathbf{s}}_j$. The underlying surface which gave rise to the textures may have a position and orientation different from that of S_j. The information contained in $\dot{\mathbf{s}}_j$ can be used to estimate the position and orientation of the underlying surface.

Once a surface has been detected, we localize its position and orientation using the following algorithm:

1. Until convergence:
   - (a) Update surfel position P_j.
   - (b) Update surfel orientation n_j.
   - (c) Reevaluate $\nu(\mathrm{S}_j)$.

Steps 1a, 1b, and 1c are actually interdependent and ideally should be calculated simultaneously. For ease of implementation and to speed convergence[10] we have separated them. As will be seen in this section, we obtain good results performing the steps individually.

We use a slightly modified version of Equation 3.4, which considers only matching regions, to update P_j:

$$\underset{\dot{\mathrm{P}}_j}{\operatorname{argmin}} \sum_{\substack{\dot{\mathrm{p}}_j^i \in \dot{\mathbf{p}}_j \\ \mathrm{s}_j^i \in \dot{\mathbf{s}}_j}} \overrightarrow{\widetilde{\mathrm{C}}_i \dot{\mathrm{p}}_j^i} \times ((\dot{\mathrm{P}}_j - \widetilde{\mathrm{C}}_i) \times \overrightarrow{\widetilde{\mathrm{C}}_i \dot{\mathrm{p}}_j^i}) \cdot (\dot{\mathrm{P}}_j - \widetilde{\mathrm{C}}_i). \tag{4.5}$$

[10]Theoretically, convergence is $O(n^3)$ where n is the number of parameters optimized, thus two half-sized problems converge four times faster than the full-sized one. We have observed this speed up in practice. In addition, the symbolic gradient expression needed by conjugate gradient methods is much simpler for the two half-sized problems.

When P_j is updated, $\dot{u}_i$ and $\dot{v}_i$ must also be updated. s_j^i changes with the new P_j but $\dot{\mathrm{p}}_j^i$ does not, thus $(\dot{u}_i, \dot{v}_i)$ must be corrected for the difference between the old and new s_j^i.

A match's dependence upon S_j and n_j can be made more explicit by rewriting Equation 3.15 as follows:

$$\nu(\mathrm{S}_j) = \frac{\sum_{i \in \mathcal{Q}} \left(\overrightarrow{\widetilde{\mathrm{C}}_i \mathrm{P}_j} \cdot \mathrm{n}_j \right) \max_{u,v} \epsilon(\mathrm{n}_j)}{\sum_{i \in \mathcal{Q}} \overrightarrow{\widetilde{\mathrm{C}}_i \mathrm{P}_j} \cdot \mathrm{n}_j} \tag{4.6}$$

where

$$\epsilon(\mathrm{n}_j) = \frac{\sum_{{}^y_x\mathrm{S}_j \in \mathrm{S}_j} \mathcal{X}(\mathcal{F}(\mathcal{T}({}^y_x\mathrm{S}_j, \mathrm{I}^i) + [u, v]), \mathcal{F}(\mathcal{T}({}^y_x\mathrm{S}_j, \mathrm{I}^\star)))}{\sum_{{}^y_x\mathrm{S}_j \in \mathrm{S}_j} 1}. \tag{4.7}$$

To update n_j we evaluate:

$$\operatorname*{argmax}_{\mathrm{n}_j} \sum_{i \in \mathcal{Q}} \epsilon(\mathrm{n}_j). \tag{4.8}$$

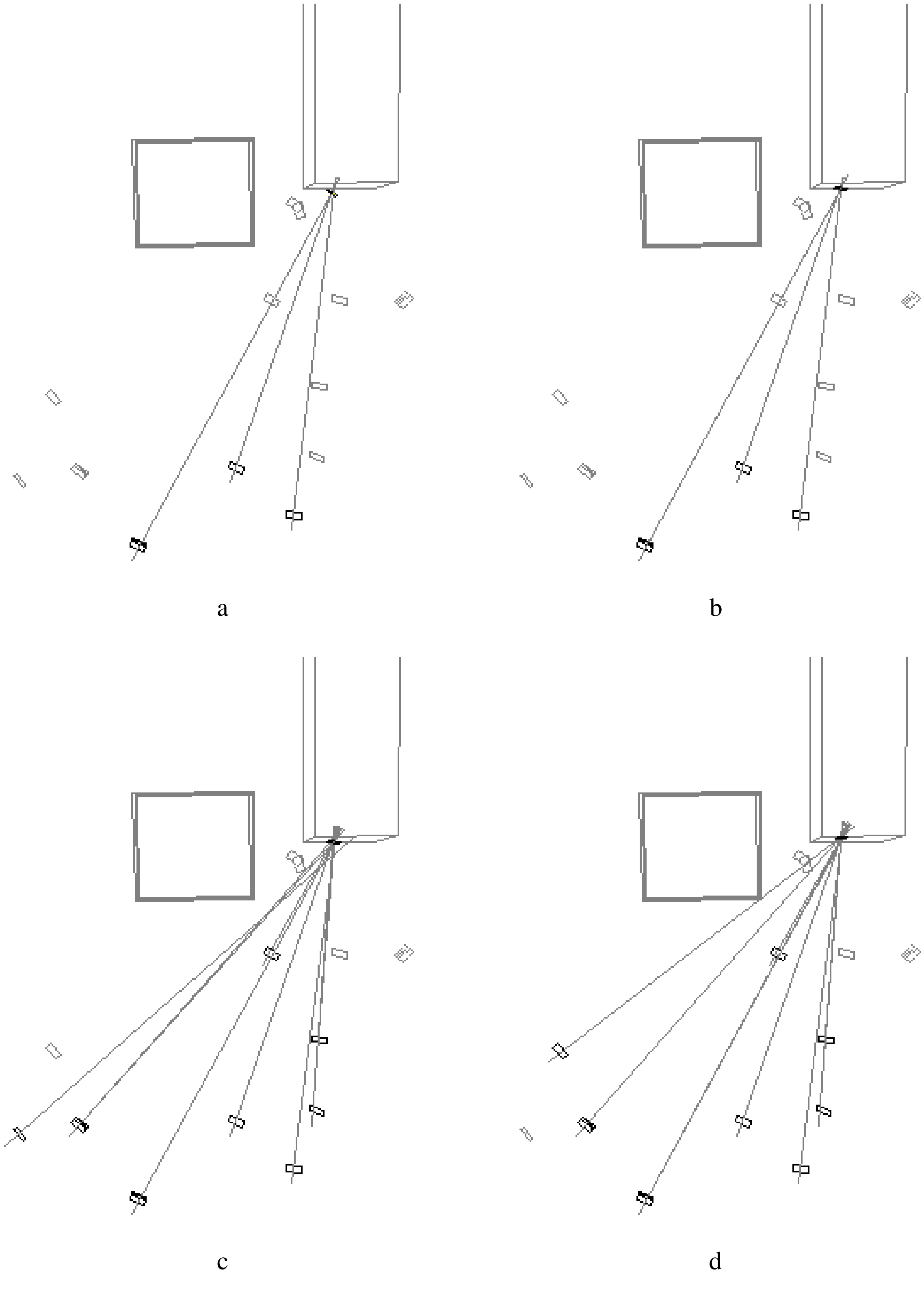
Both direction set methods and conjugate gradient methods[11] work well to solve Equation 4.8.

Once the position and orientation of the surfel have been updated, we reperform Steps 2, 3, 5, and 6 of the detection algorithm. Step 2 is only partially performed. Images with currently matching regions are retained. Images which no longer view the front side of the surfel are eliminated and additional ones that do are added. Steps 1a, 1b, and 1c of the localization algorithm are repeated while the match score is improving until the position update is < 1 unit and the orientation update is $< 1°$ or for 3 iterations if the match score is not improving.

4.3.1 Results

Figure 4-17 depicts the localization process. Images which view the front side of the surfel are shown in grey. Those that contribute to the match set are shown in black. Lines of sight through the center of the matched regions and building outlines are also shown in grey. Figure 4-17a shows the initial detection for a surfel displaced from the actual surface by 100 units in positions and 30° in orientation. Figure 4-17b shows the surfel after updating its position and orientation (Steps 1a and 1b of the localization algorithm). Figure 4-17c shows a new match set using the updated position and orientation (Step 1c). Figure 4-17d shows the final match set at convergence. The same set of test points used

[11] While not as straight forward as in the last section, deriving $\nabla_{\mathrm{n}_j} \epsilon(\mathrm{n}_j)$ leads to an easily evaluated expression that depends only upon known quantities such as the image data, the gradient of the image data, and the position and orientation of surfel. See Appendix B for details.



a b c d

Figure 4-17: Surfel localization.

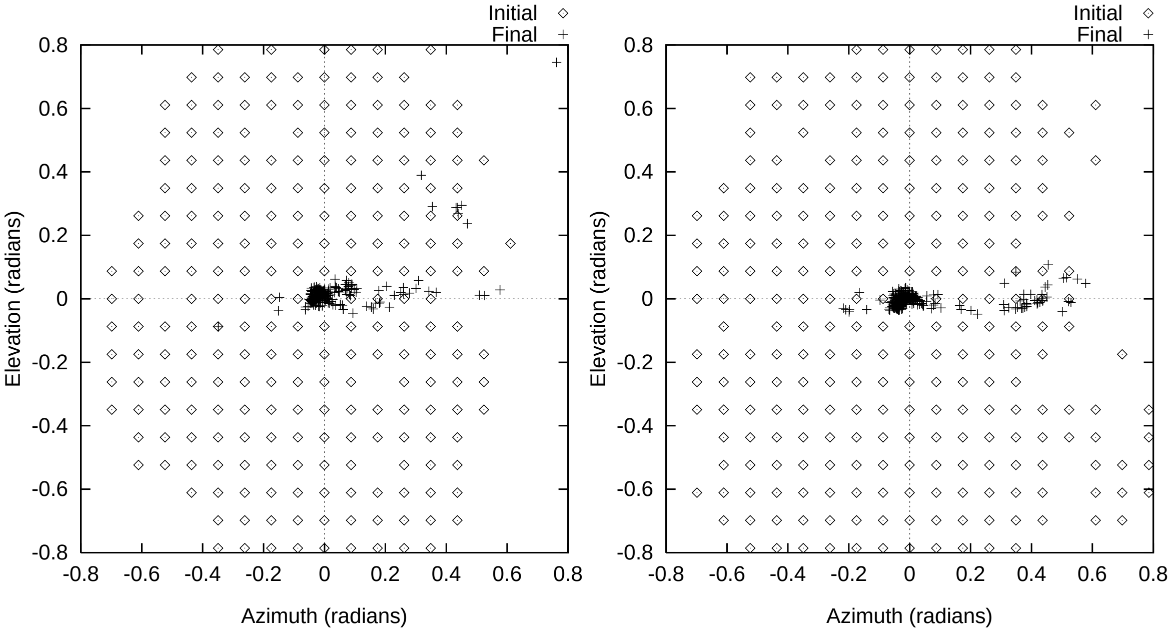


Figure 4-18: Localization examples.

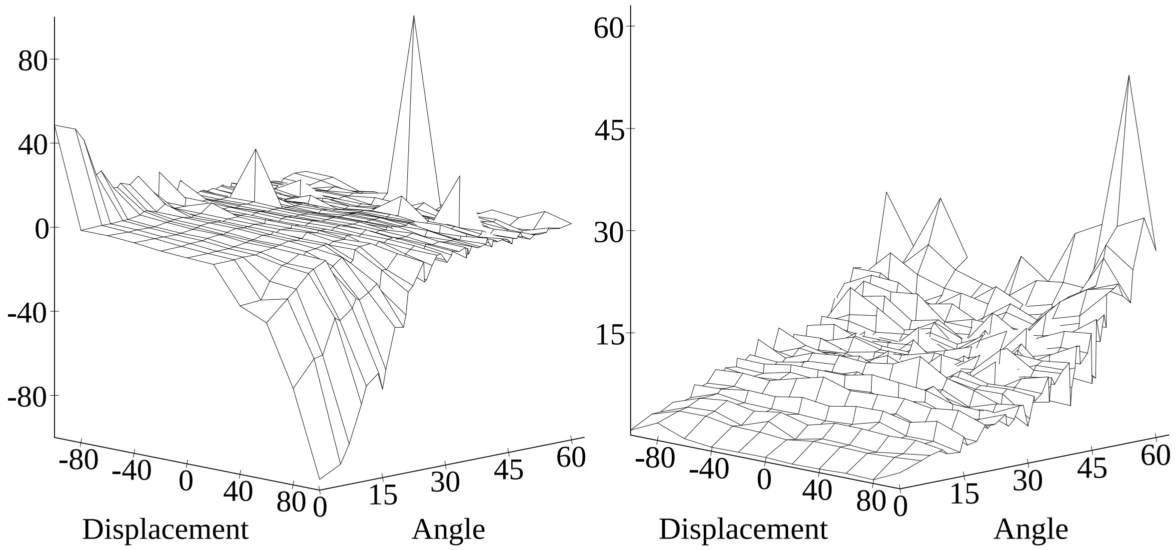


Figure 4-19: Localization summary.

to test detection (Figures 4-12 and 4-13) was also used to evaluate localization. Figure 4-18 shows two test points evaluated at 361 different orientations. The test point for the plot on the left is 80 units in front of the actual surface and the one for the right is 100 units behind. The asymmetry in both of these plots is caused by the asymmetric distribution of cameras shown in Figure 4-24. A *diamond* is plotted at the initial orientation for each detection and a *plus* marks the final estimated orientation. Figure 4-19 shows the aggregate results for the complete set of test points. The plot on the left shows final displacement versus the initial displacement and the angle between the actual and tested orientation. The plot on the right shows the angle between the actual and final estimated orientation versus the initial displacement and the angle between the actual and tested orientation. For displacements upto 100 units in position and 30° in orientation, nearly all of the test points converged to the correct values.

4.4 Discussion

So far we have focused on detecting and localizing nearby surfels and have not addressed bogus matches. As pointed in Chapter 3, compensating for noisy data admits false positives. Figure 4-20 shows the raw surfels[12] detected and localized near one of the buildings imaged in the dataset. There are a significant number of false positives. Notice the parallel layers of surfels near the upper right corner of the building. These are false positives similar to the one shown in Figure 3-3. Many of the surfels are near the surfaces of the building. Figure 4-21 shows the distribution of distances to the nearest model surface. Figures 4-22 and 4-23 show the raw results for the full reconstruction and Figure 4-24 shows the volume used for the full reconstruction. The next chapter addresses ways to eliminate false positives.

The reconstructions shown in this thesis were performed on quarter resolution images. To test the effects of image resolution, we reran the raw detection and localization algorithms on half and eighth resolution images. Several parameters (maximum shifts, interest, and uniqueness) are resolution dependent and these were adjusted appropriately. The results are shown in Figure 4-25. The half resolution reconstruction is very sparse. We suspect the primary cause is image noise. It turns out that half the output resolution actually corresponds to the full resolution of the physical sensor [De Couto, 1998]. Increased image noise degrades the match score, whose tolerance was not adjusted, and interferes with the gradient estimates used to optimize the shifts. Assuming the noise is Gaussian in nature, reducing the resolution also reduces the noise. However, there is a limit to how much the resolution can be reduced; sufficient information must be retained to perform the reconstruction. This implies that there is a range of image resolutions (with favorable noise characteristics and

[12] Only front facing surfels are rendered.

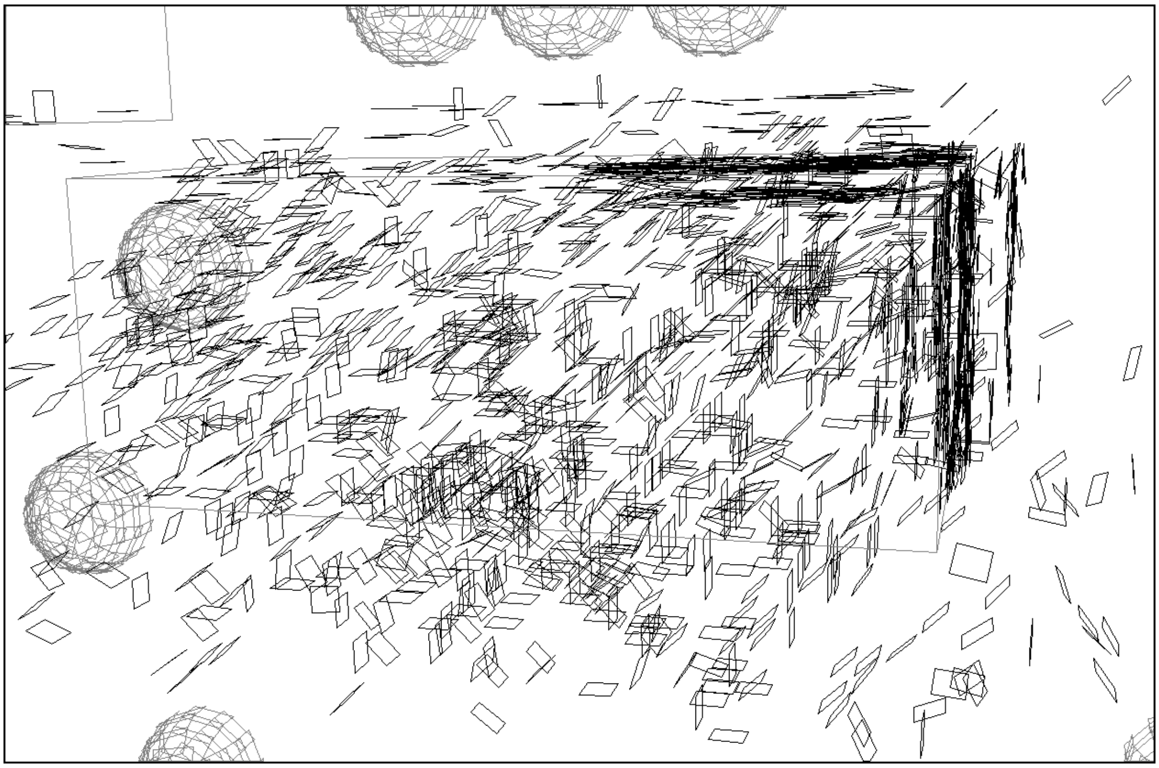


Figure 4-20: Raw surfels (partial reconstruction).

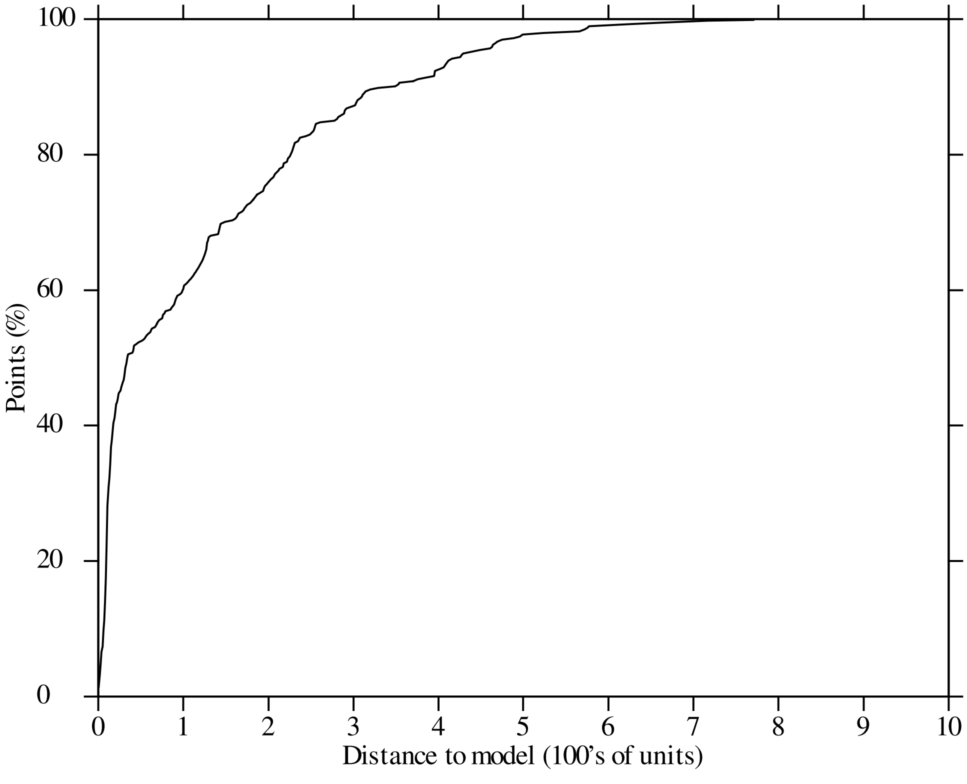


Figure 4-21: Distance to nearest model surface (partial reconstruction).

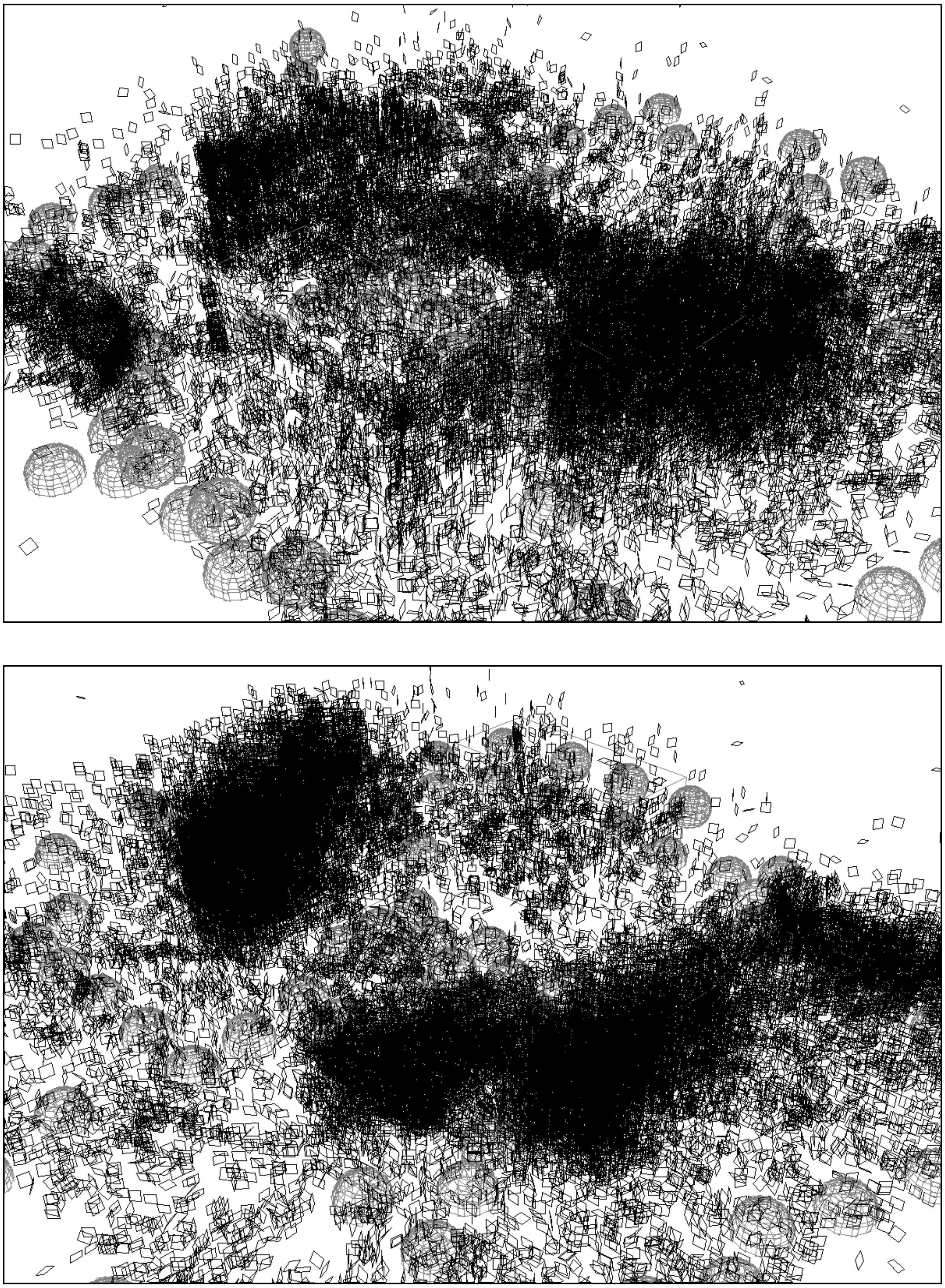


Figure 4-22: Raw surfels (full reconstruction).

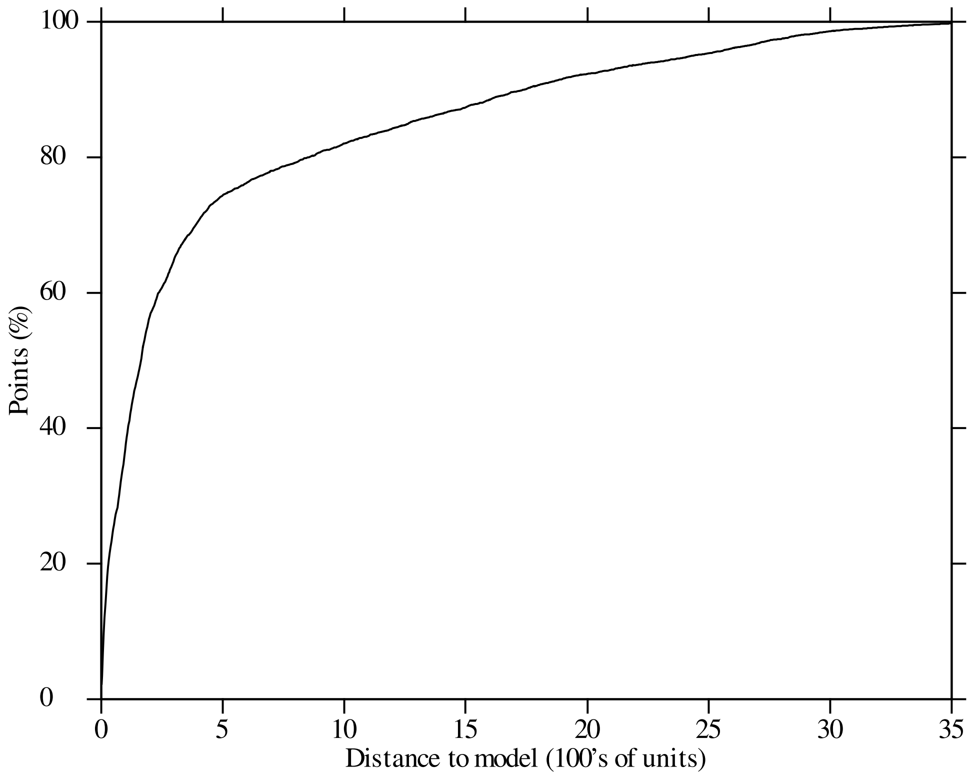


Figure 4-23: Distance to nearest model surface (full reconstruction).

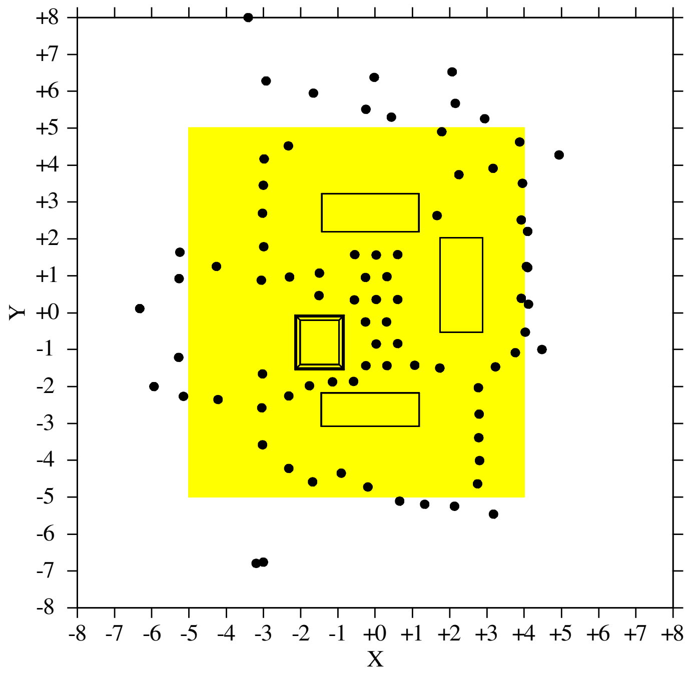
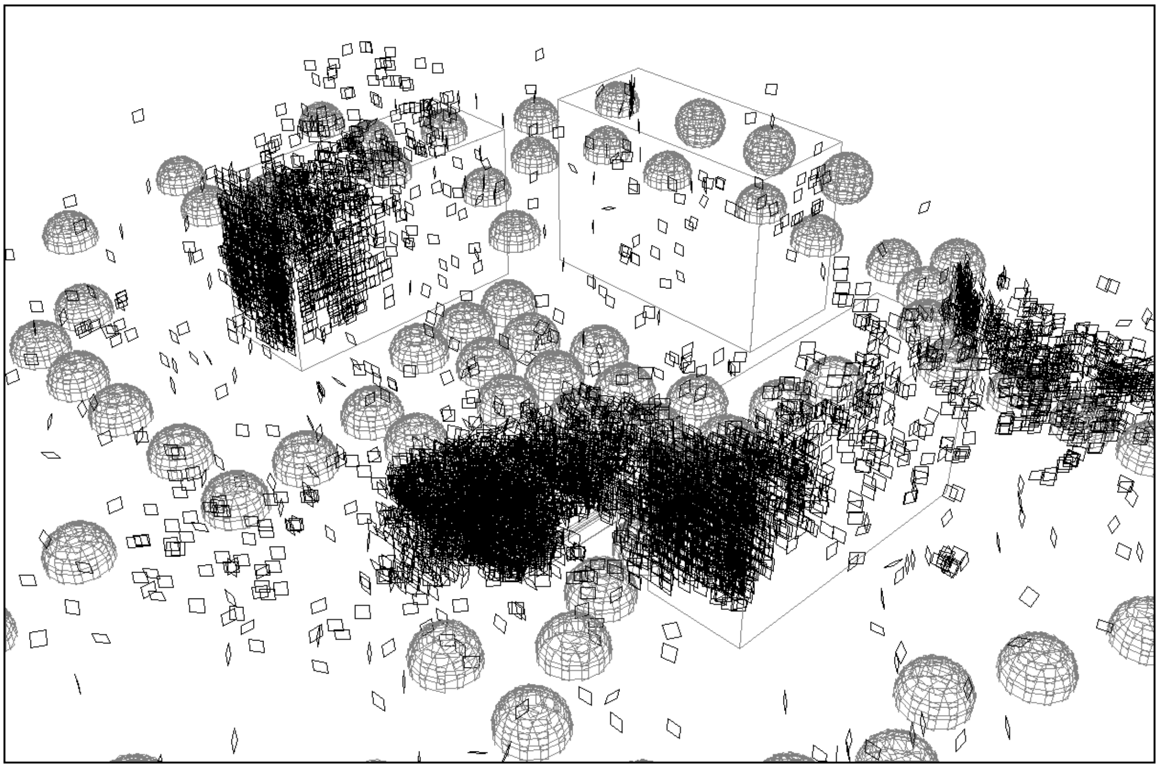


Figure 4-24: Reconstruction volume (shaded area) used for full reconstruction.



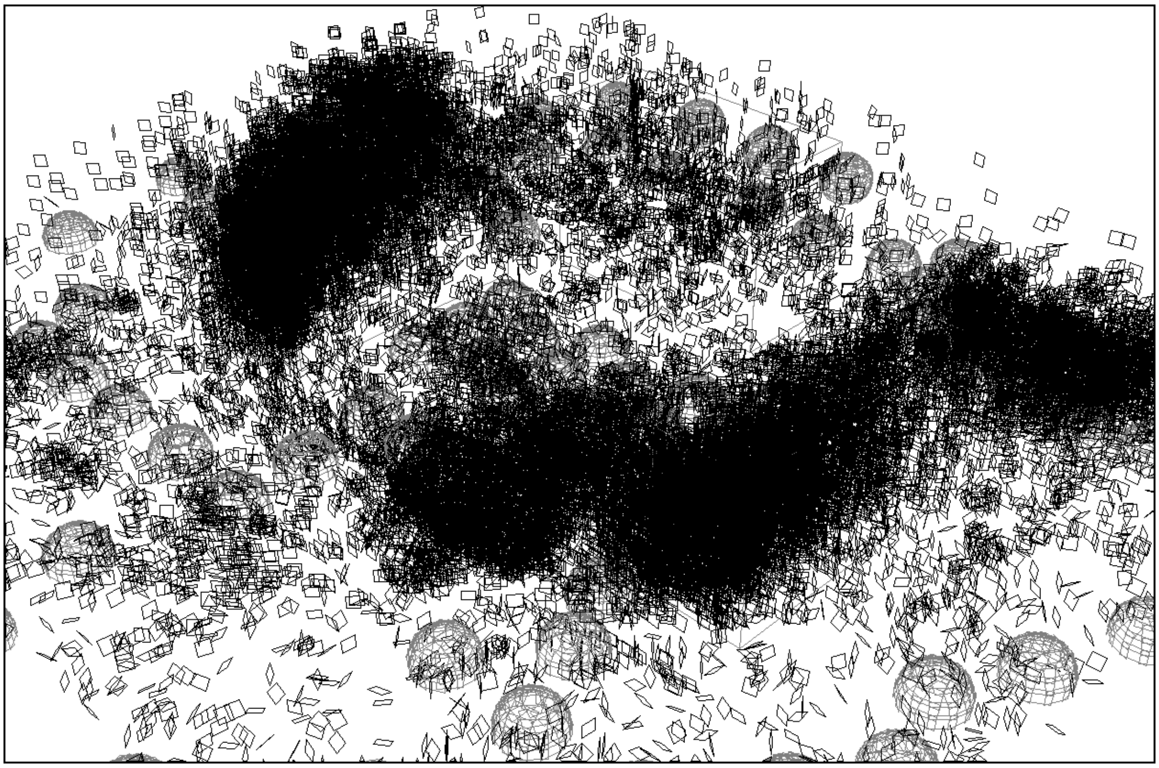


Figure 4-25: Raw surfels for full reconstruction using half (top) and eighth (bottom) resolution images.

sufficient information content) which are essentially equivalent. The quarter and eighth resolution reconstructions seem to support this view.

Chapter 5

From Surfels to Surfaces

The shifts, illumination corrections, and masks introduced in Chapter 3 to compensate for noisy data also increase the occurrence of false positives. The results presented in Chapter 4 are purely local and make no attempt to reject these false positives. This chapter explores several geometric constraints which together eliminate nearly all false positives.

5.1 Camera Updates

As pointed out in Section 3.1 unconstrained shifts introduce false positives such as the one shown in Figure 3-3. Figures 5-1 and 5-2 shows shifts $\{(\dot{u}, \dot{v})\}$ plotted as a vector field in image-space along with the corresponding image. The position of each vector corresponds to the location of the center of a matching region. Note that the shifts appear to be random. The actual image displacements caused by camera calibration error should be consistent with a translation of the camera center and a rotation about it or formally

$$(u, v)_{\tilde{\mathrm{p}}_j^i} = \mathcal{T}(\mathrm{P}_j, \mathrm{I}^i) - \mathcal{T}(\mathrm{P}_j, \tilde{\mathrm{I}}^i) \tag{5.1}$$

where

$$\tilde{\mathrm{p}}_j^i = \mathcal{T}(\mathrm{P}_j, \tilde{\mathrm{I}}^i)$$

and P_j is the location of S_j. To enforce this constraint we use the following algorithm:

1. For each camera $\tilde{\mathrm{I}}^i$.
   (a) Build $\left\{(\dot{u}_i, \dot{v}_i)_{\tilde{\mathrm{p}}_j^i}\right\}$.
   (b) Calculate a first-order camera calibration update, $\tilde{\mathrm{I}}^{i\prime}$.
   (c) Calculate the final camera calibration update, $\tilde{\mathrm{I}}^{i\prime\prime}$.
   (d) Remove matching regions with shifts that are not consistent with the final update, $\tilde{\mathrm{I}}^{i\prime\prime}$.
2. Prune surfels which no longer meet the match criteria.

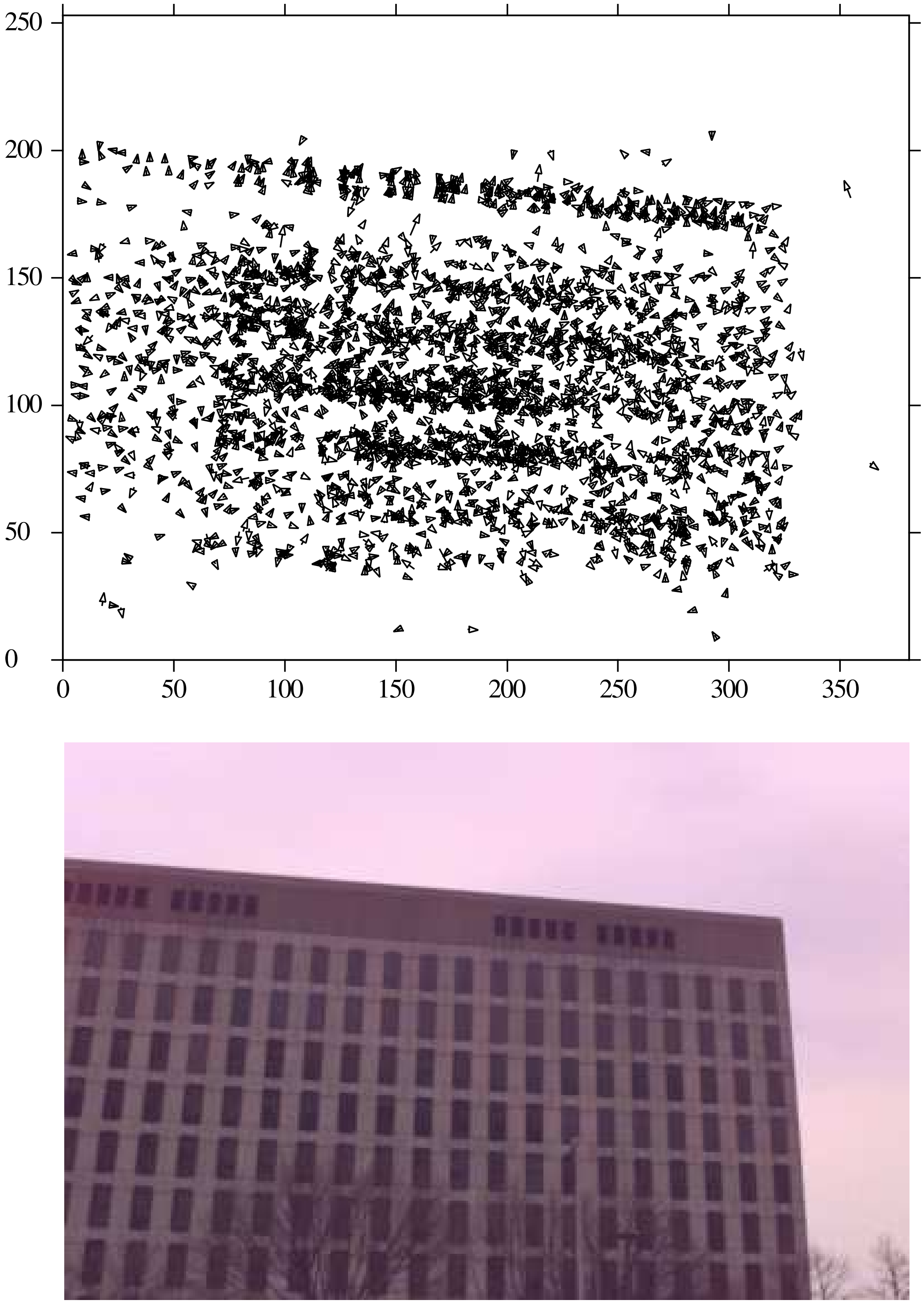


Figure 5-1: Shifts $\{(\dot{u}_i, \dot{v}_i)\}$ plotted as a vector field and image data for node 27 image 13.

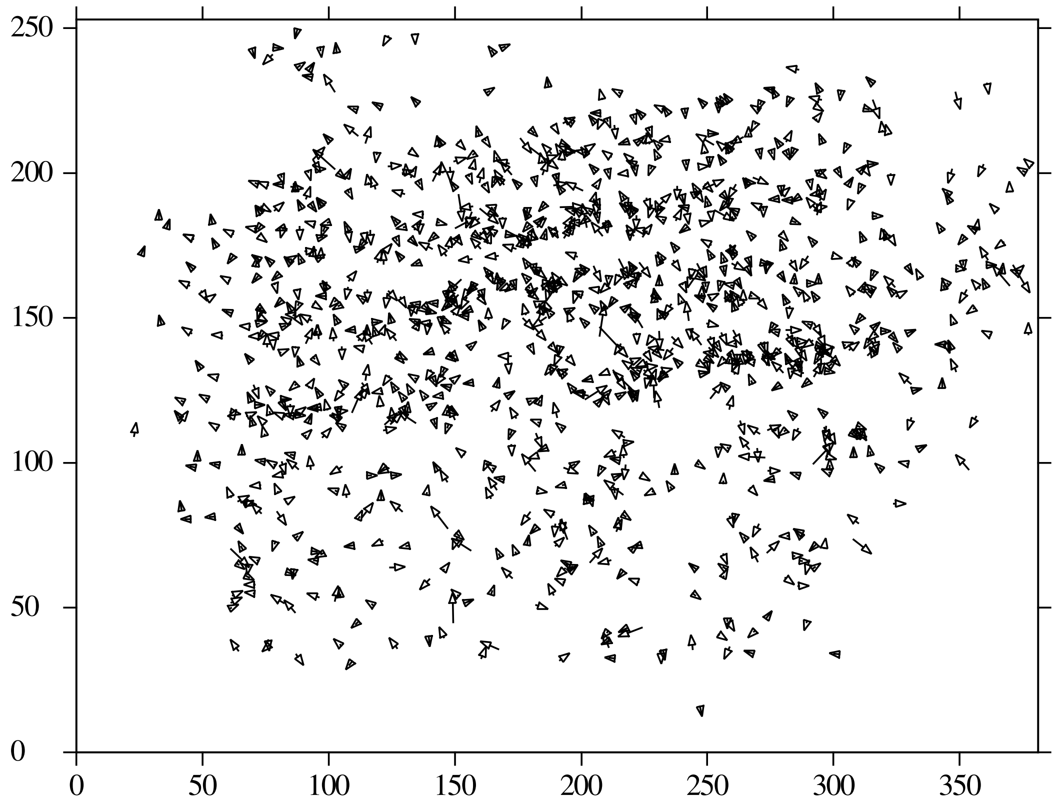


Figure 5-2: Shifts $\{(\dot{u}_i, \dot{v}_i)\}$ plotted as a vector field and image data for for node 28 image 17.

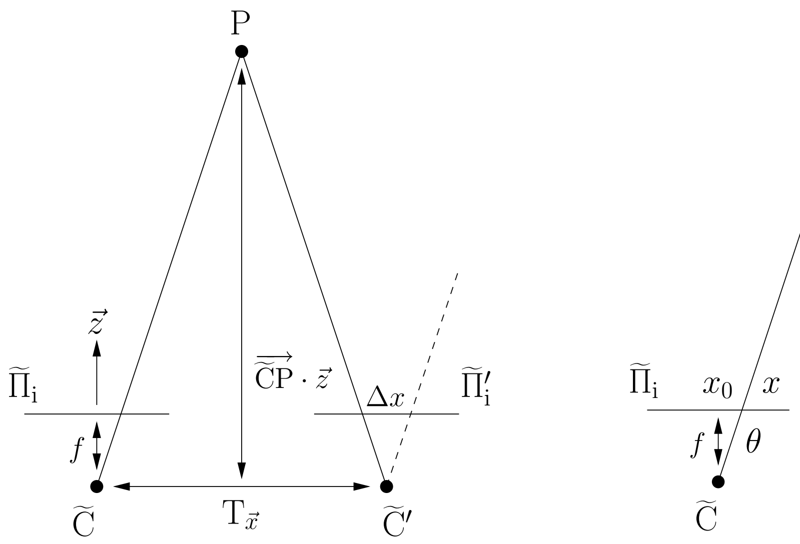


Figure 5-3: Effects of camera calibration updates.

The left side of Figure 5-3 shows the effect of translating a camera in the $\vec{x}$ direction. Δx and $T_{\vec{x}}$ are related as

$$\Delta x = -\frac{f}{\overrightarrow{\widetilde{C}P} \cdot \vec{z}} T_{\vec{x}} \tag{5.2}$$

and similarly for Δy and $T_{\vec{y}}$

$$\Delta y = -\frac{f}{\overrightarrow{\widetilde{C}P} \cdot \vec{z}} T_{\vec{y}}. \tag{5.3}$$

The right side of Figure 5-3 shows rotation about the y axis. Differentiating

$$\theta = \arctan\left(\frac{x - x_0}{f}\right)$$

yields the first order approximation

$$\Delta x = -\frac{f^2 + (x - x_0)^2}{f} \delta\theta. \tag{5.4}$$

Similarly for ϕ, rotation about the x axis,

$$\Delta y = \frac{f^2 + (y - y_0)^2}{f} \delta\phi. \tag{5.5}$$

Finally, for γ, rotation about the optical axis,

$$\begin{aligned} \Delta x &= -(y - y_0)\delta\gamma && (5.6) \\ \Delta y &= (x - x_0)\delta\gamma. && (5.7) \end{aligned}$$

Equations 5.2-5.7 combine to form the following pair of coupled linear equations:

$$\dot{u} = \frac{f^2 + (x - x_0)^2}{f}\delta\theta + \frac{f}{\overrightarrow{\tilde{\mathrm{C}}\mathrm{P}} \cdot \vec{z}}\mathrm{T}_{\vec{x}} + (y - y_0)\delta\gamma \quad (5.8)$$

$$\dot{v} = \frac{f^2 + (y - y_0)^2}{f}\delta\phi + \frac{f}{\overrightarrow{\tilde{\mathrm{C}}\mathrm{P}} \cdot \vec{z}}\mathrm{T}_{\vec{y}} - (x - x_0)\delta\gamma. \quad (5.9)$$

Equations 5.8 and 5.9 are solved in a least squares fashion [Watkins, 1991]. $\tilde{\mathrm{I}}^i$ is updated using $\delta\theta$, $\delta\phi$, $\delta\gamma$, $\mathrm{T}_{\vec{x}}$, and $\mathrm{T}_{\vec{y}}$[1] to produce the initial solution in Step 1b, $\tilde{\mathrm{I}}^{i\prime}$. The final update is

$$\operatorname*{argmin}_{\tilde{\mathrm{I}}^{i\prime\prime}} \sum_{(\dot{u}_i,\dot{v}_i)\in\mathcal{Q}} \|(\dot{u},\dot{v}) - \mathcal{T}(\mathrm{P}_j,\tilde{\mathrm{I}}^{i\prime\prime}) + \mathcal{T}(\mathrm{P}_j,\tilde{\mathrm{I}}^{i})\|^2, \quad (5.10)$$

where

$$\mathcal{Q} = \left\{(\dot{u}_i,\dot{v}_i) \;\middle|\; \epsilon \geq \|(\dot{u},\dot{v}) - \mathcal{T}(\mathrm{P}_j,\tilde{\mathrm{I}}^{i\prime}) + \mathcal{T}(\mathrm{P}_j,\tilde{\mathrm{I}}^{i})\|^2\right\}.$$

Typically ϵ is 1 pixel. Solutions which have enough data (at least 4 points) and fit well are retained. $\tilde{\mathrm{I}}^{i\prime\prime}$ is used to prune inconsistent matches from $\dot{\mathbf{s}}_j$,

$$\mathrm{s}^i_j \setminus \dot{\mathbf{s}}_j \text{ if } \epsilon < \|(\dot{u},\dot{v}) - \mathcal{T}(\mathrm{P}_j,\tilde{\mathrm{I}}^{i\prime\prime}) + \mathcal{T}(\mathrm{P}_j,\tilde{\mathrm{I}}^{i})\|^2.$$

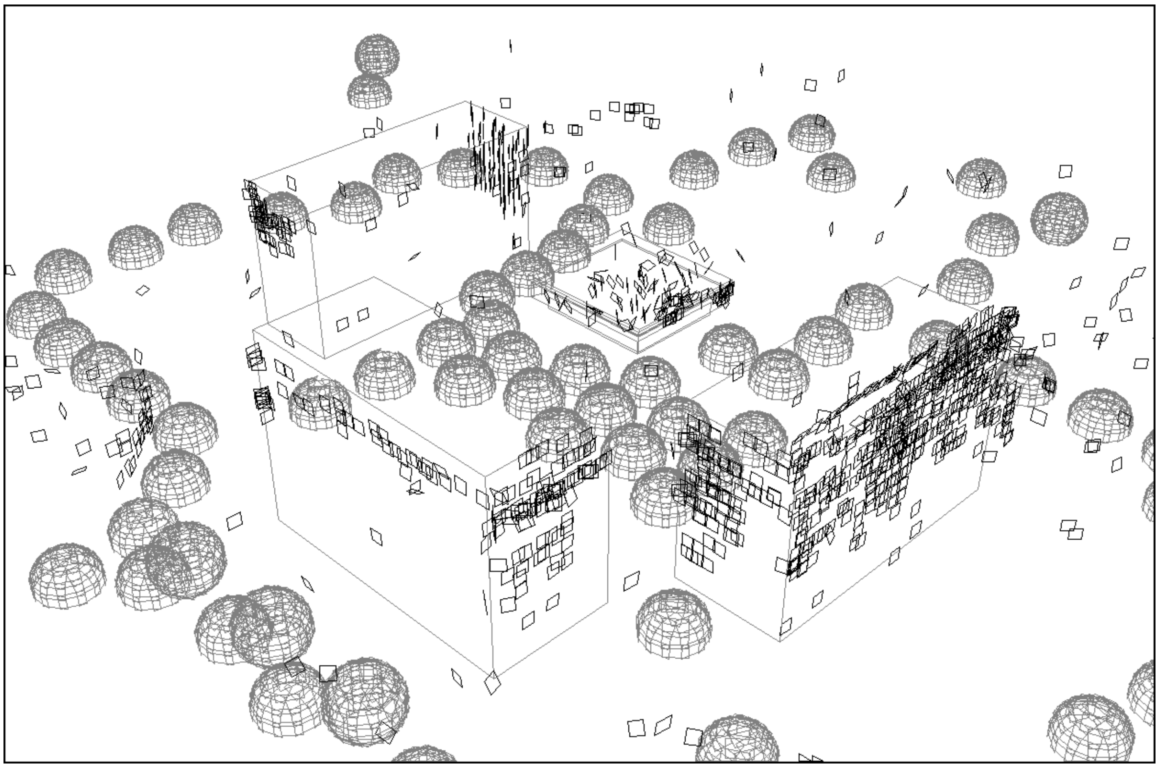
After removing inconsistent matches, if $\dot{\mathbf{s}}_j$ no longer meets the criteria described in Section 4.2, it is discarded. Figure 5-6 shows the average rms shift before and after performing the calibration update. The number of consistent data points (shifts) are plotted along the x axis and the square root of the mean squared shift is plot along the y axis. Note that the necessary shifts are significantly reduces. Ideally, after updating the camera calibration estimates, no shifts would be required. The images used as input contain a small amount of nonlinear distortion which cannot be modeled by the pin-hole camera model. We estimate the rms value of this distortion to be about 1 pixel.

Figure 5-4 shows the consistent surfels remaining after applying this algorithm to the raw reconstruction shown in Figure 4-22 and Figure 5-5 shows the distribution of distances to the nearest model surface. A number of the consistent surfels come from objects which are not in the reference model. The cluster of surfels between the building outlines near the top center of Figure 5-4 is one example. These surfels come from a nearby building. As a test, a small portion of the full reconstruction was rerun with the updated camera calibration. The results are shown in Figure 5-7. For comparison, the consistent surfels from the same area are shown in Figure 5-8.

5.2 One Pixel One Surfel

Figure 5-9 shows a region from one image which contributes to the match set of multiple surfels. Each pixel in each image should contribute to at most one

[1] The internal parameters are held fixed and for stability reasons $\mathrm{T}_{\vec{z}}$ is not updated.



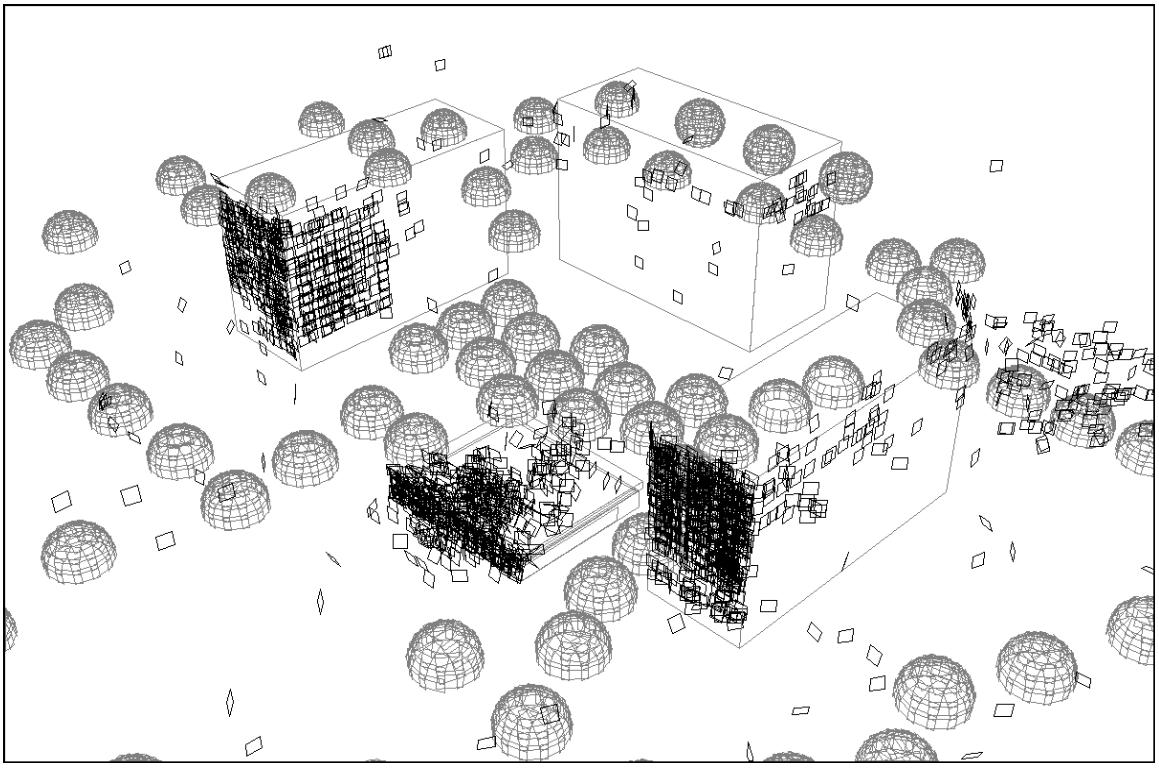


Figure 5-4: Consistent surfels.

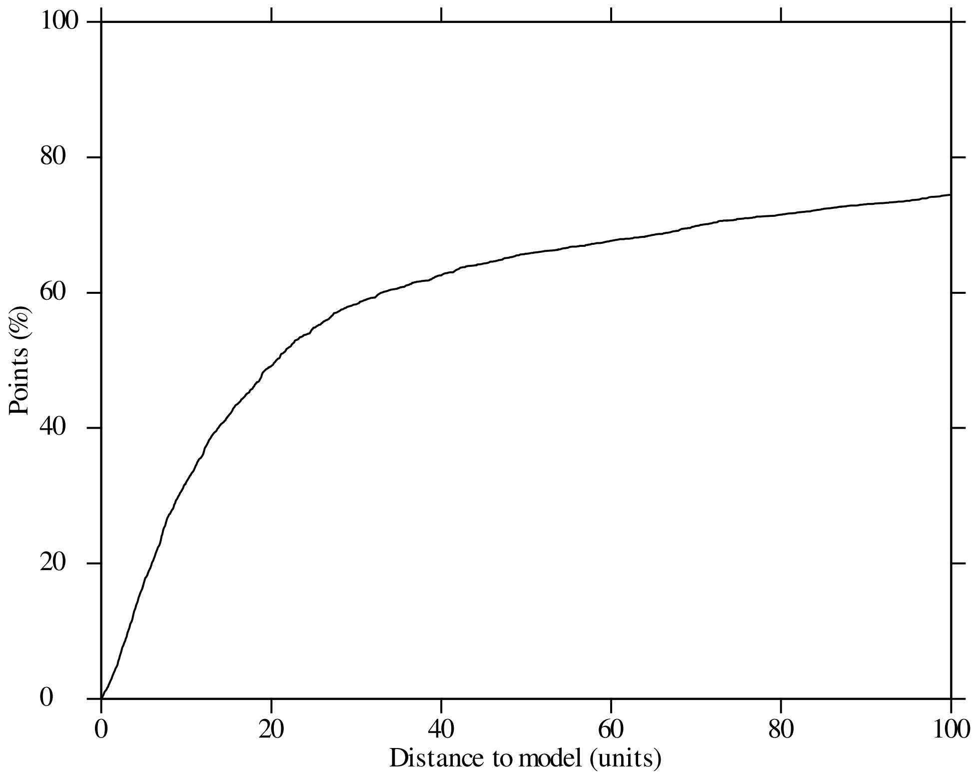


Figure 5-5: Distribution of error for consistent surfels.

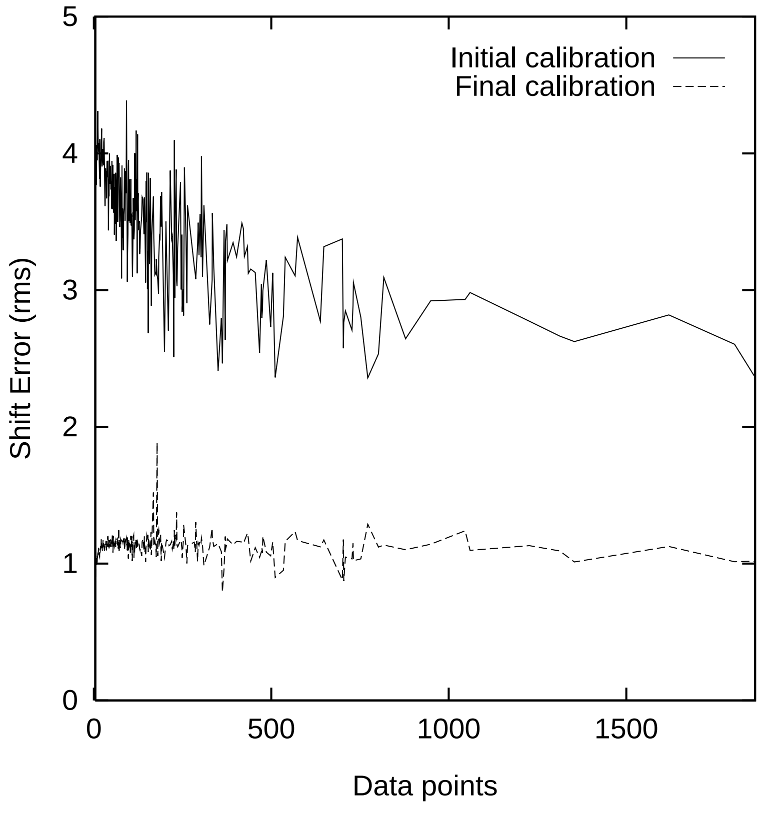


Figure 5-6: Comparison of initial and final calibration estimates.

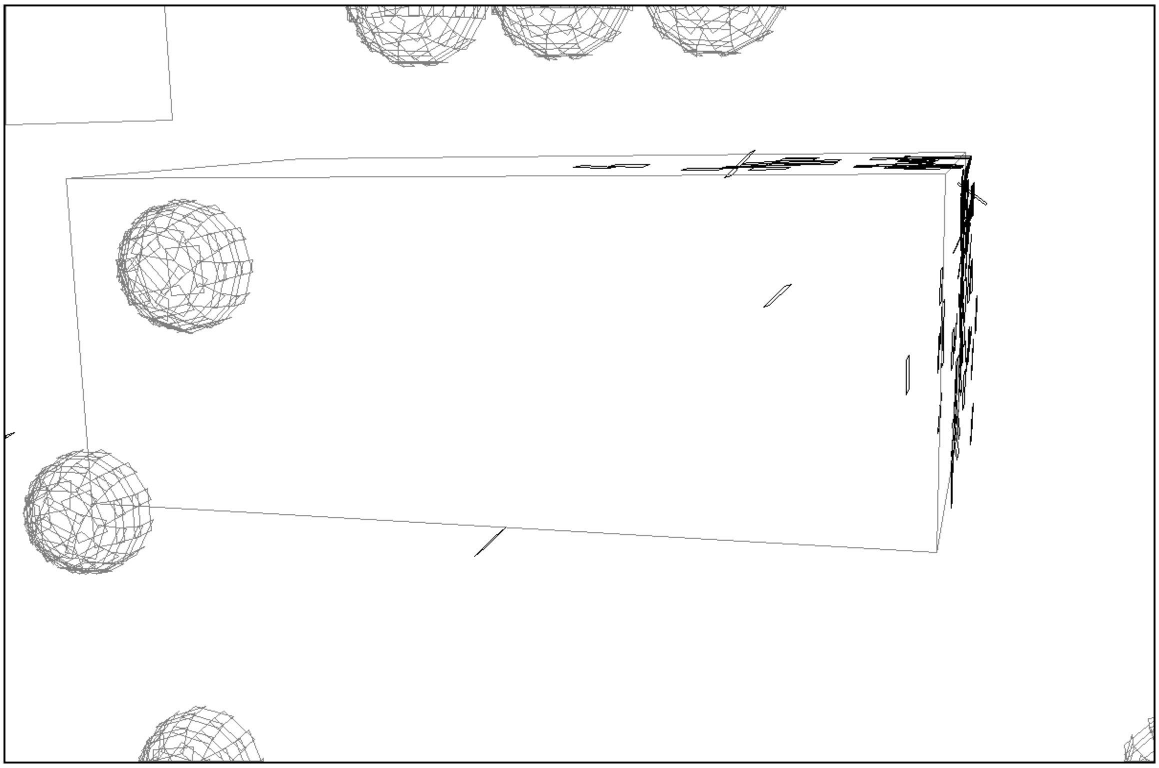


Figure 5-7: Close-up of reconstruction using updated camera calibrations.

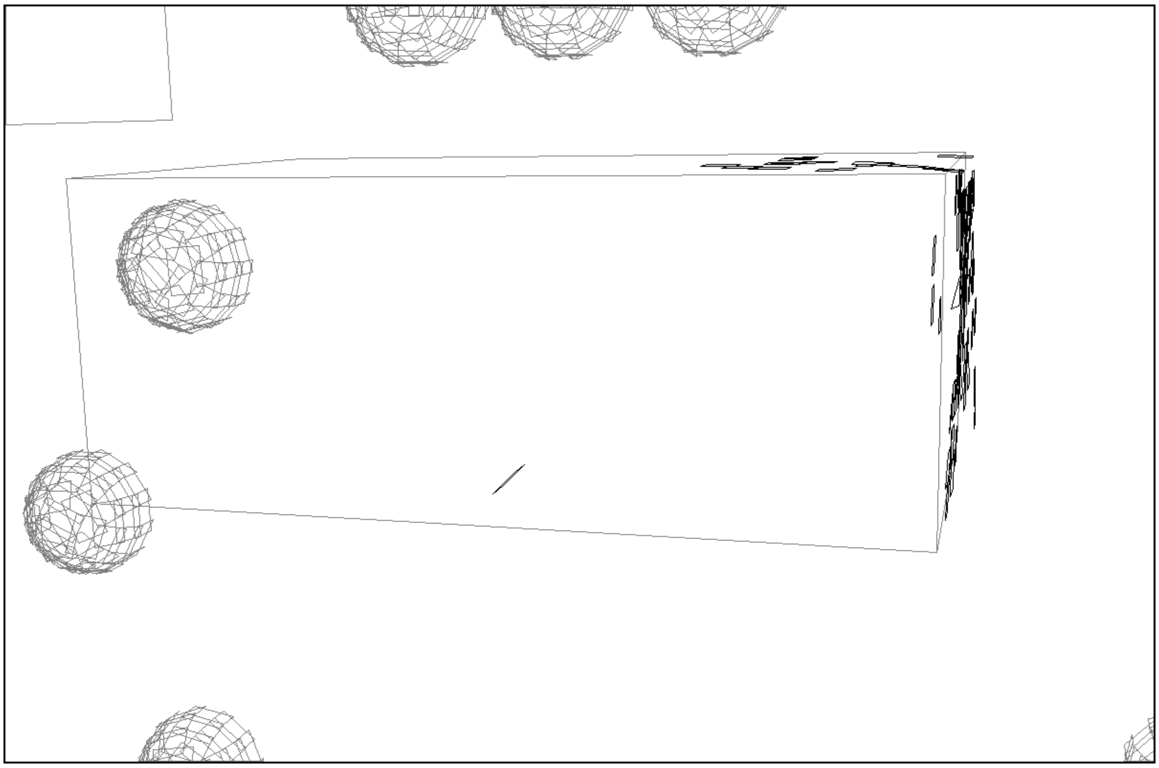


Figure 5-8: Close-up of same area showing only consistent surfels from original reconstruction (estimated camera calibrations).

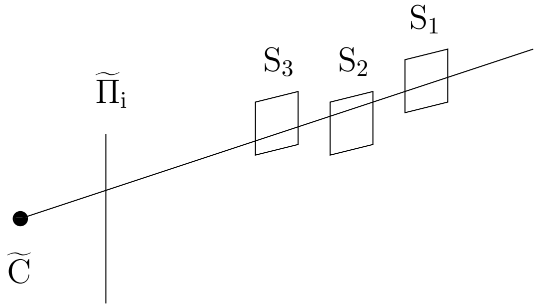


Figure 5-9: A region from one image which contributes to multiple surfels.

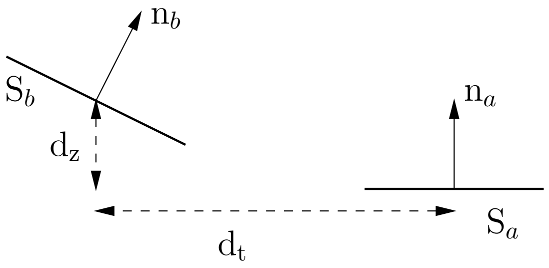


Figure 5-10: Determining if S_b is a neighbor of S_a.

surfel. Deciding which surfel is the hard part. Detection and localization as described in Chapter 4 do not enforce this constraint and as a result even after enforcing a consistent calibration update there are many image regions which contribute to multiple surfels. We eliminate them in a soft manner using the following algorithm:

1. Score each surfel.
2. For each surfel S_a.
   (a) For each region $s_a^i \in \dot{\mathbf{s}}_a$.
      i. For each surfel S_b with a score higher than S_a, if $s_b^i \in \dot{\mathbf{s}}_b$.
         A. De-weight s_a^i.
   (b) If the match score is no longer sufficient, prune S_a.

To score a surfel we use a combination of the number of cameras which contribute to a surfel and the number of neighbors that it has. A surfel S_a is considered a neighbor of S_b if 1) the distance from P_b to the plane containing S_a (the normal distance) is no more than d_z; 2) the distance from the projection of S_b onto the plane containing S_a and P_a (the tangential distance) is no more than d_t; and, 3) the angle between n_a and n_b is no more than β. This notion of neighbors is essentially a smoothness constraint and will be used in the next section to group surfels. Typically, we use $d_z = 15$, $d_t = 300$, and $\beta = \arccos(0.9)$. The de-weighting is done in a continuous manner. S_a is divided into sample points (we use 10×10) each with an initial weight of 1.0. Each sample point is tested against the cone formed by S_b and $\widetilde{C}_i$. Those inside the cone have their weights updated. The new weight is based on the normal distance and the angle between n_a and n_b. If the angle is greater then β, the new weight is 0.0. Otherwise, the new weight is based on the normal distance; 1.0 if it is less than d_z, 0.0 if it is greater than $3d_z$, and varies linearly in between. The lesser of the currently assigned and new weight is retained. After all S_b's have been tested against S_a, the average sample point weight is used as the new

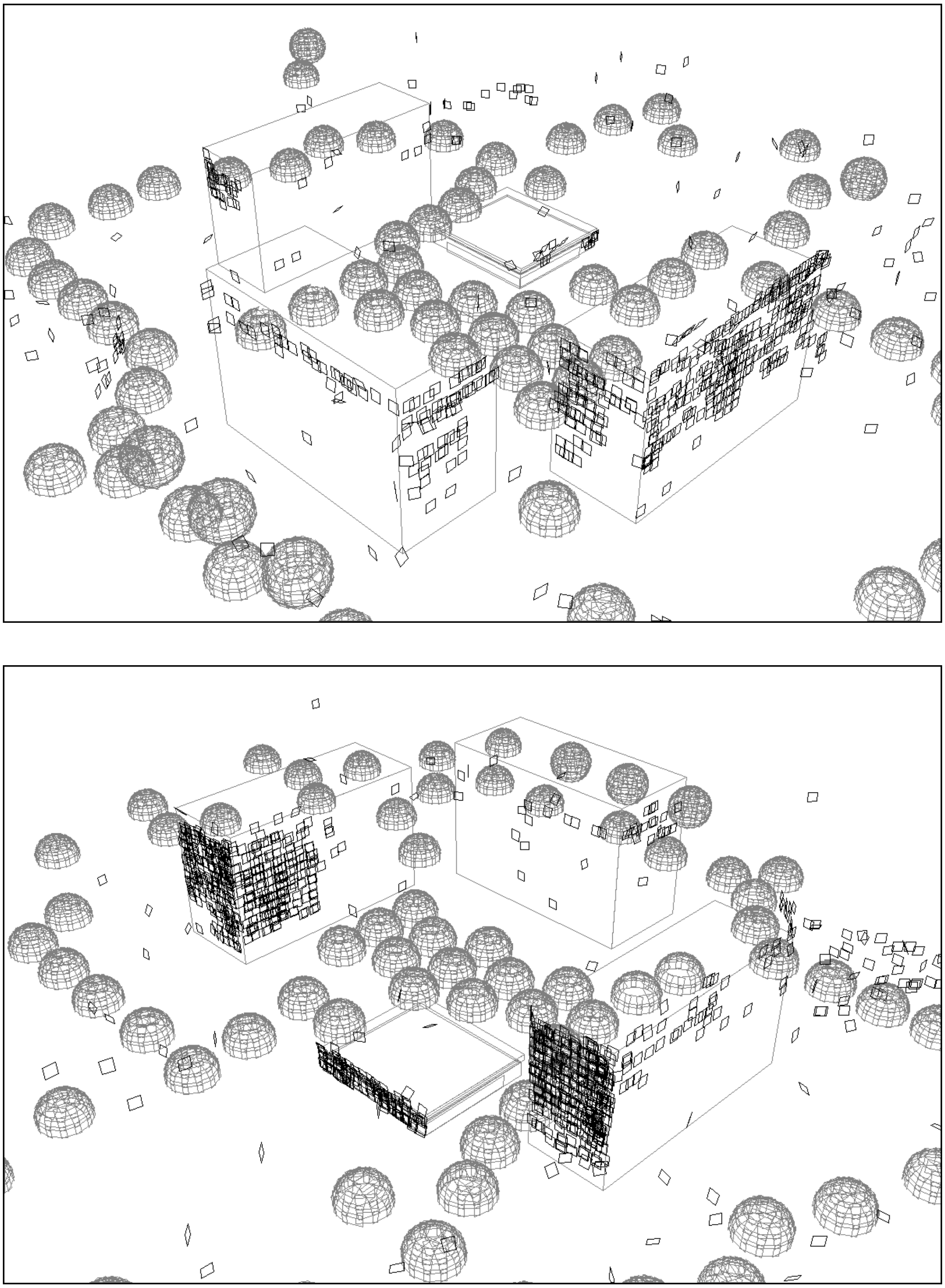


Figure 5-11: Surfels after pruning multiple contributions.

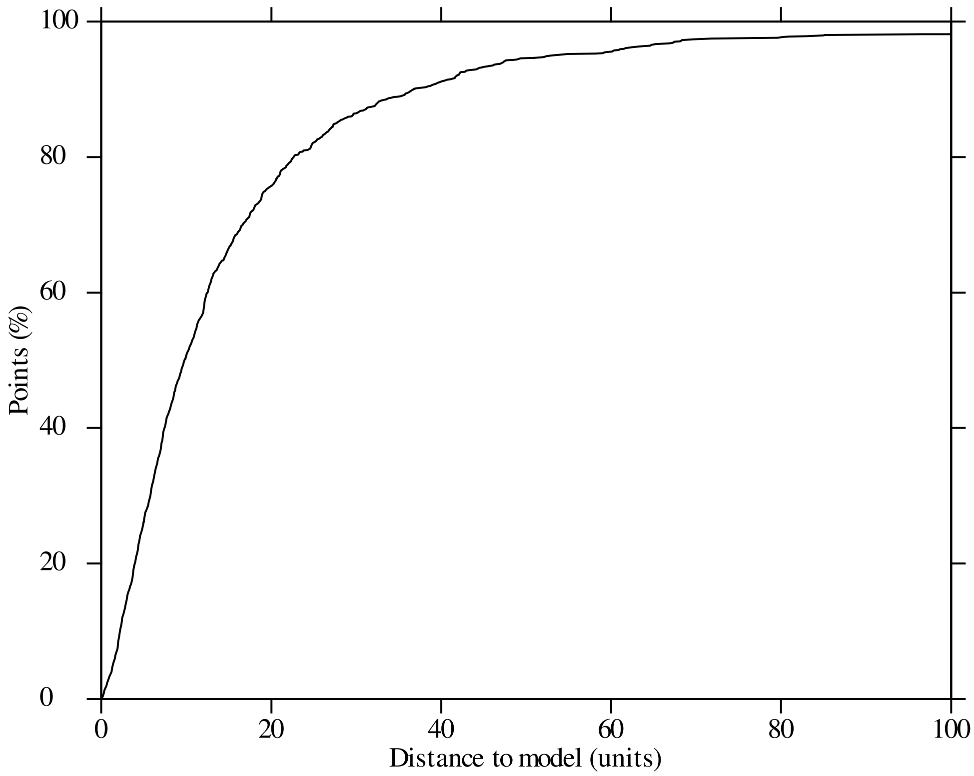


Figure 5-12: Distribution of error for pruned surfels.

weight for s_a^i. When each s_a^i has been tested the sum of the weights is used to determine if S_a is retained. Figure 5-11 shows the reconstruction after pruning multiple contributions and Figure 5-12 shows the distribution of distances to the nearest model surface.

5.3 Grouping Surfels

The buildings we are trying to model are much larger than an individual surfel. Therefore, a large number of surfels should be reconstructed for each actual surface. Using the notion of neighbors described in the last section, we group the reconstructed surfels as follows:

1. For each surfel S_a.
   (a) For each surfel S_b already assigned a group.
      i. If S_a and S_b are neighbors.
         A. If S_a has not already been assigned to a group, then assign S_a to the group containing S_b.
         B. Otherwise merge the groups containing S_a and S_b.
   (b) If S_a has not been assigned to a group, then create a new group and assign S_a to it.

In practice we retain only groups which have at least a minimum number of (typically four) surfels. All of the surfels in a group should come from the same surface. This notion of grouping places no restriction on the underlying surface other than smoothness (e.g. it may contain compound curves). Figure 5-13 shows the reconstruction after grouping and removing groups with fewer than four surfels. Nearly all of the surfaces in the reference model have at least one corresponding group. Figure 5-14 shows the distribution of distances to the nearest model surface.

5.4 Growing Surfaces

Many of the groups shown in figure 5-13 do not completely cover the underlying surface. There are several reasons why surfels corresponding to actual surfaces might not produce a valid match set. The main one is soft occlusion described in Section 3.3. Another is local maxima encountered while finding the best shifts and updating the surfel's normal. In the reconstruction process so far, the mask technique described in Section 3.3 has not been utilized. In this section we make use of it. We also use estimated the shifts and illumination corrections to help place us closer to the correct maxima. We use the following algorithm to grow surfaces:

1. For each group.
   (a) Create an empty list of hypothesized surfels.
   (b) For each hypothesized surfel.
      i. Test using the detection and localization algorithms.
      ii. If a match.
         A. Add to the group.
         B. Test against each surfel in each of the other groups.
            If a neighbor, merge the two groups.
   (c) Use the next surfel in the group S_a to generate new hypotheses and goto Step 1b.

The hypotheses in Step 1c are generated from S_a by considering the eight nearest neighbors in the plane containing S_a as shown in Figure 5-15. S_a is shown in grey and the hypotheses are shown in white. Each of the other reconstructed surfels in the group are orthographically projected onto the hypothesized surfels and hypotheses which are more than half covered are discarded. The shifts and illumination corrections associated with S_a are used as initial values for each hypothesis in Step 1(b)i. In addition we lower the minimum interest to 25 and the minimum uniqueness to 1.2. Figure 5-16 shows the reconstruction after growing. After growing, the coverage of each surface is nearly complete. Figure 5-17 shows the distribution of distances to the nearest

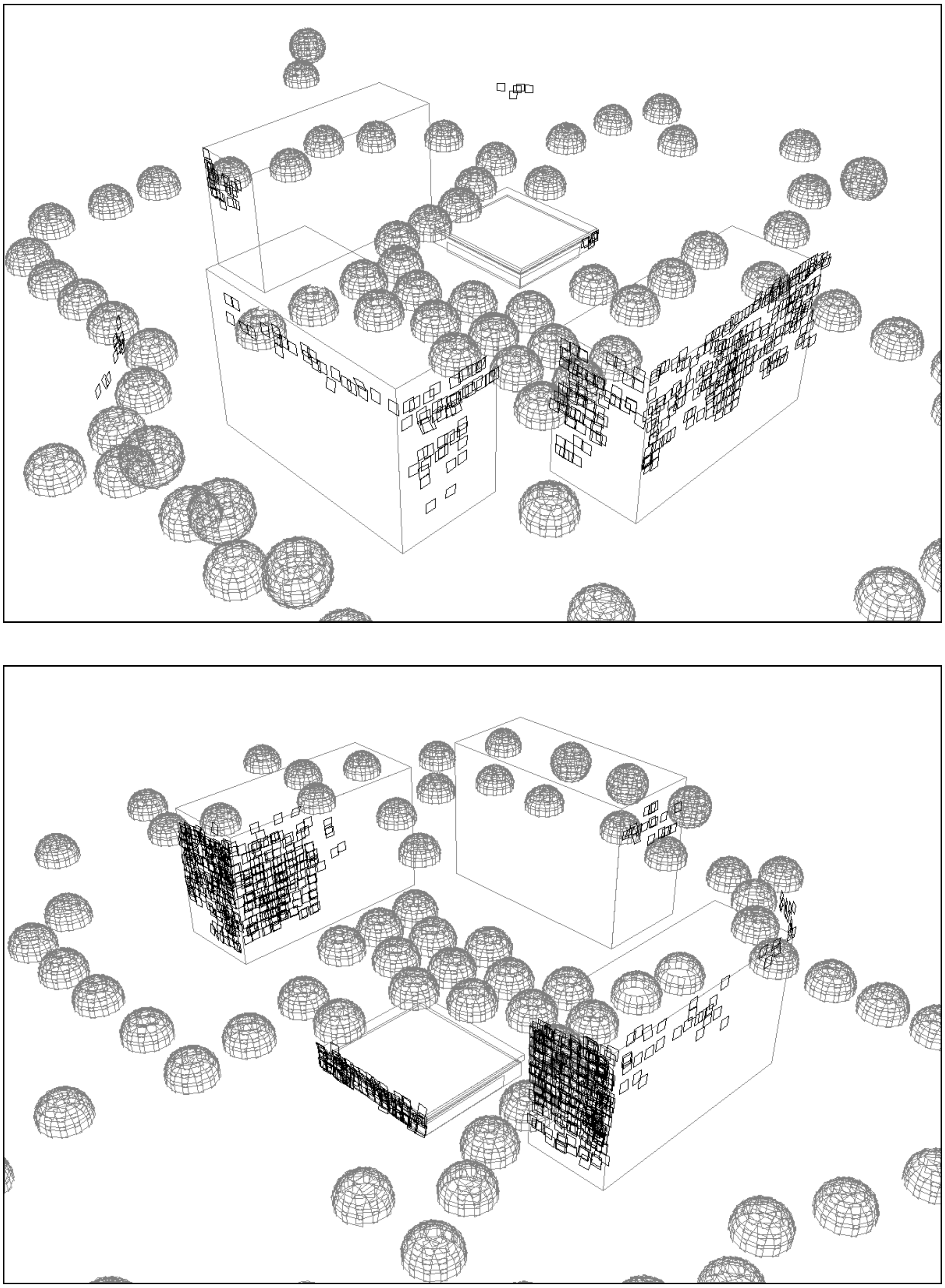


Figure 5-13: Surfels after grouping.

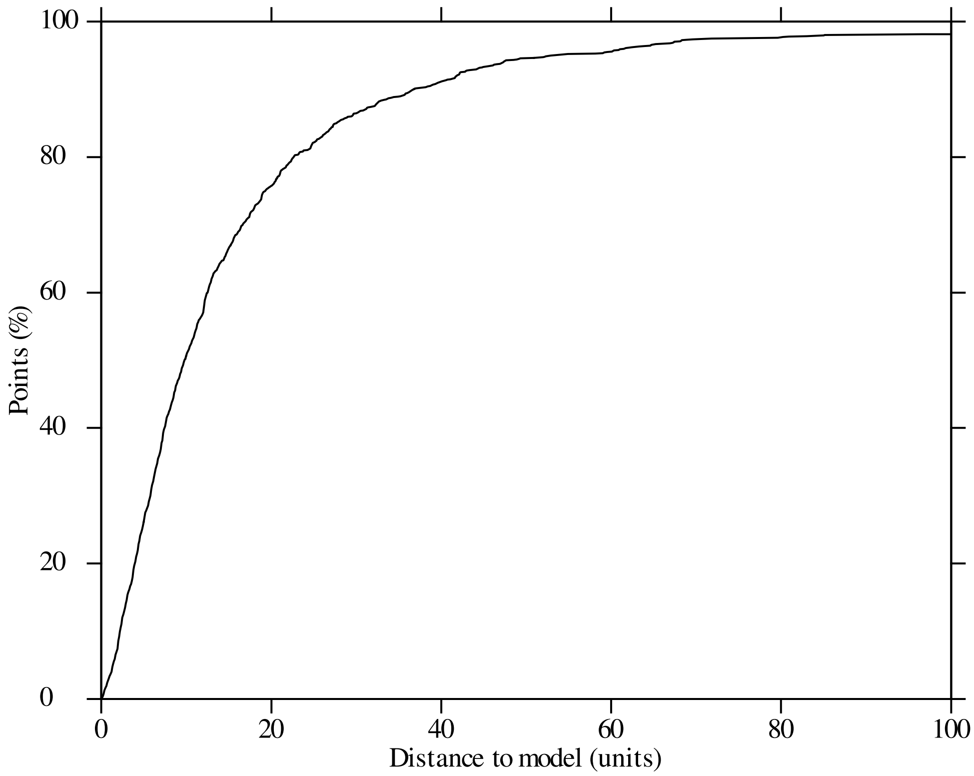


Figure 5-14: Distribution of error for grouped surfels.

model surface. Figure 5-18 shows grown surfels after removing multiple contributions and grouping as described in the previous two sections and Figure 5-19 shows the distribution of distances to the nearest model surface.

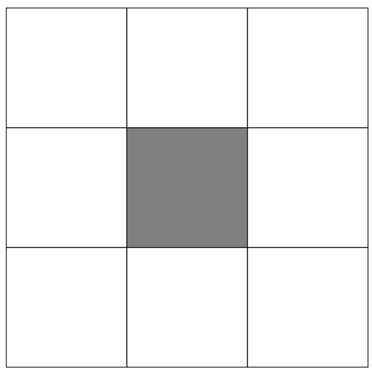


Figure 5-15: Reconstructed surfel (grey) and hypotheses (white).

5.5 Extracting Models and Textures

So far, the only assumption we have made about the structure of the world is that locally it can be approximated by a plane. All of the buildings imaged in

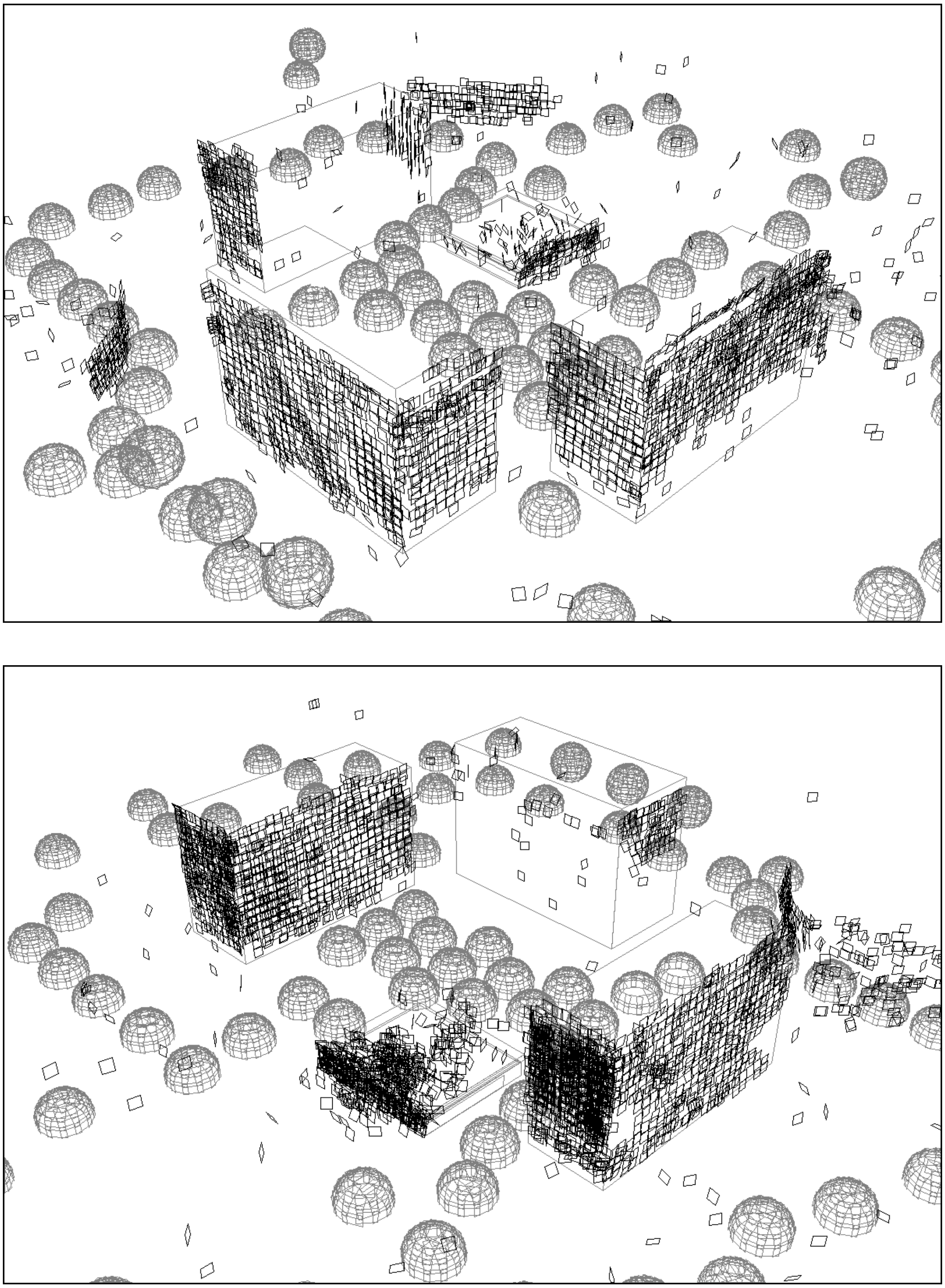


Figure 5-16: Surfels after growing.

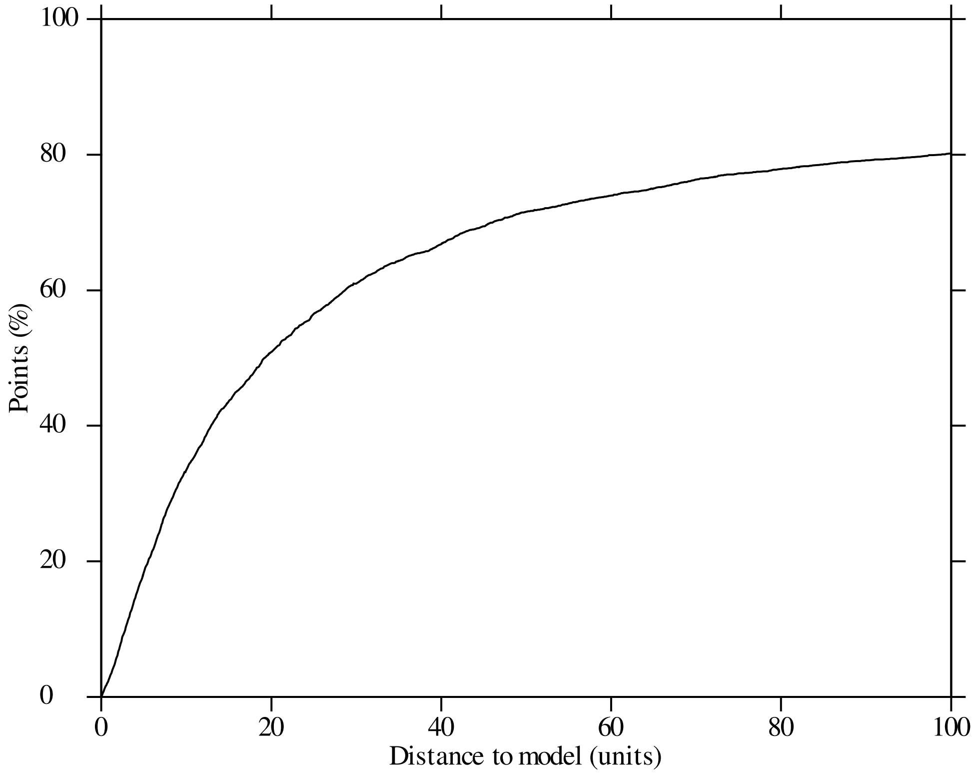


Figure 5-17: Distribution of error for grown surfels.

our dataset are composed of planar faces, therefore we simply fit planes to the groups identified in the previous section. In this case, a face is equivalent to a large surfel. The position and orientation of S_g are determined by

$$\mathrm{P}_g = \sum_{\mathrm{S}_j \in \mathcal{S}_g} w\mathrm{P}_j \Bigg/ \sum_{\mathrm{S}_j \in \mathcal{S}_g} w \tag{5.11}$$

and

$$\mathrm{n}_g = \sum_{\mathrm{S}_j \in \mathcal{S}_g} w\mathrm{n}_j \Bigg/ \sum_{\mathrm{S}_j \in \mathcal{S}_g} w \ , \tag{5.12}$$

where w is the score calculated in Step 1 of the algorithm to remove multiple contributions and $\mathcal{S}_g$ is the set of all surfels in group g. The extent of S_g is determined by orthographically projecting $\mathcal{S}_g$ onto S_g and finding the bounding box. Figure 5-20 shows the reconstructed S_g's. A total of 15 surfaces were recovered. Figure 5-21 shows the distribution of distances to the nearest model surface. As noted previously, many surfels come from structures not in the reference model. Three of the reconstructed surfaces fall into this category, hence Figure 5-21 has a maximum of 80.

Image data from contributors to $\dot{\mathbf{s}}_a$ can easily be related using the illumination corrections calculated during detection and localization. The relationship between s_b^i and $\dot{\mathbf{s}}_a$ where $\dot{\mathbf{s}}_a$ and $\dot{\mathbf{s}}_b$ are members of the same group is more difficult when i is not a contributer of $\dot{\mathbf{s}}_a$. To resolve these relationships we

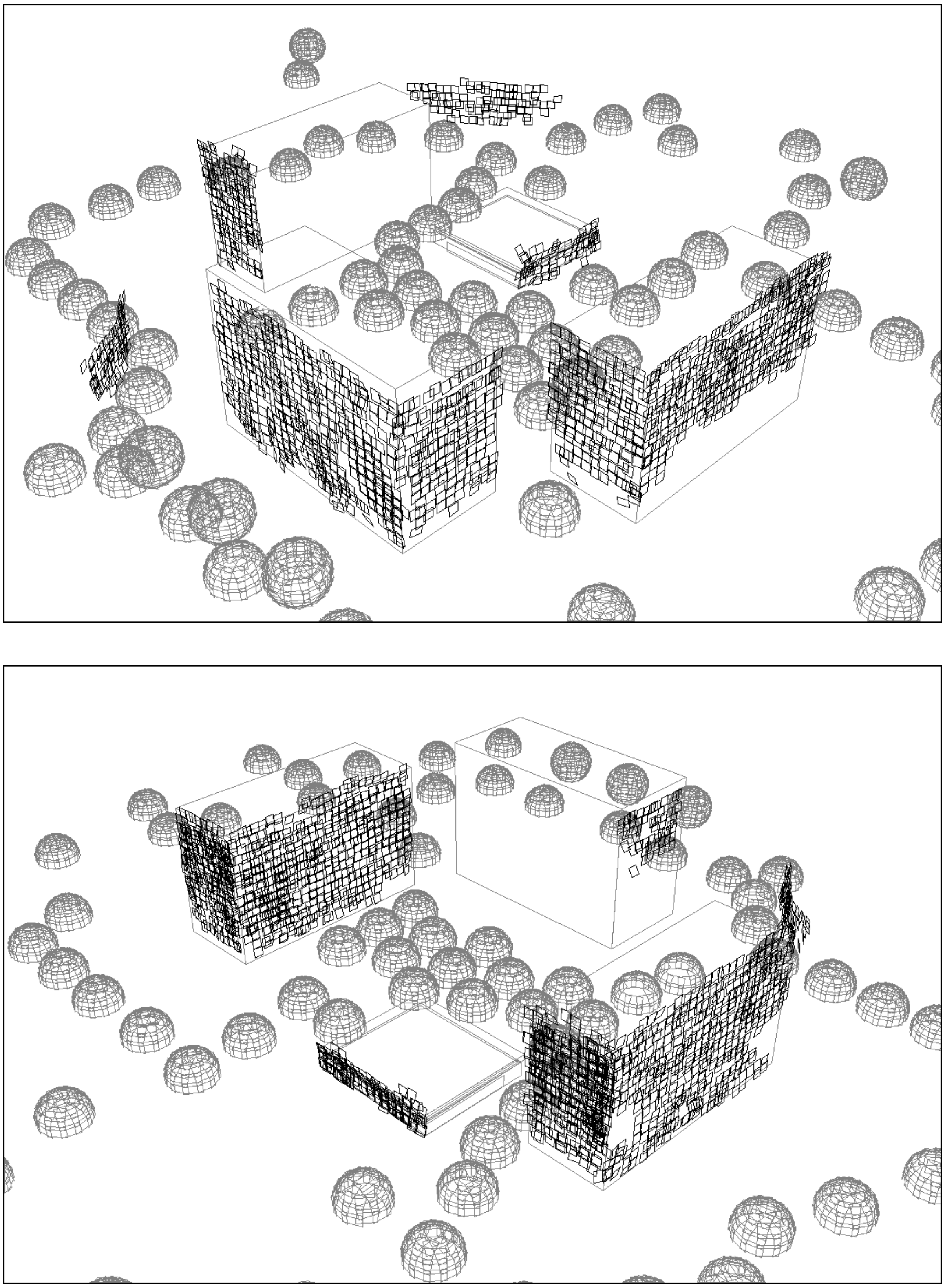


Figure 5-18: Surfels after regrouping.

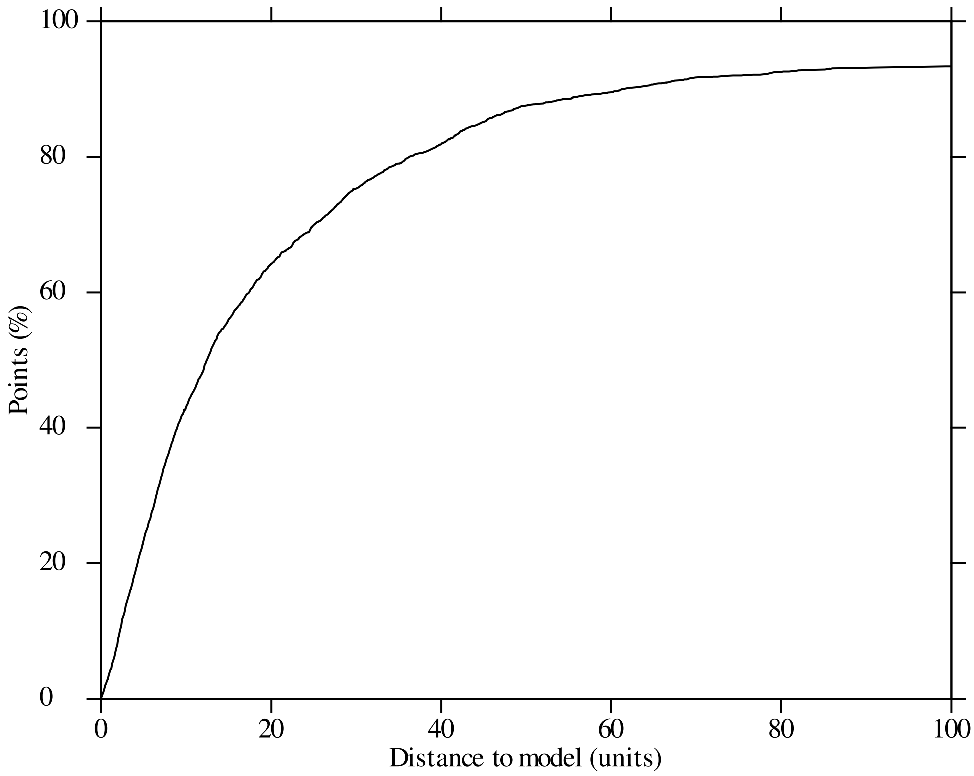


Figure 5-19: Distribution of error for regrouped surfels.

construct a table of the illumination corrections for $\mathcal{I}$, the set of images contributing to at least one surfel in $\mathcal{S}$, using the following algorithm:

1. For each group.
   (a) Find the surfel with the highest score, $\mathrm{S_m}$.
      i. Make $\mathrm{s}_\mathrm{m}^\star$ the root of the tree.
      ii. Add the illumination correction for s_m^i to the table where $i \neq \star$.
   (b) Until no new entries to the table are added.
      i. For each S_j.
         A. If $\exists\, a, b \mid \mathrm{s}_j^a$ is in the table and s_j^b is not,
            Then calculate the illumination correction between s_j^b and $\mathrm{s}_\mathrm{m}^\star$ and enter it in the table.

We assume that each surfel in a group has at least one contributing image in common with at least one other surfel in the group. When building the table we consider only the first correction encountered for each contributing image. Once the table is built we correct each image in $\mathcal{I}$ and reproject it onto S_g. The texture associated with S_g is simply the average. Figure 5-22 shows two views of the reconstructed textures. Notice that the rows of window in adjacent faces are properly aligned. This occurs even though no constraints between faces are imposed.

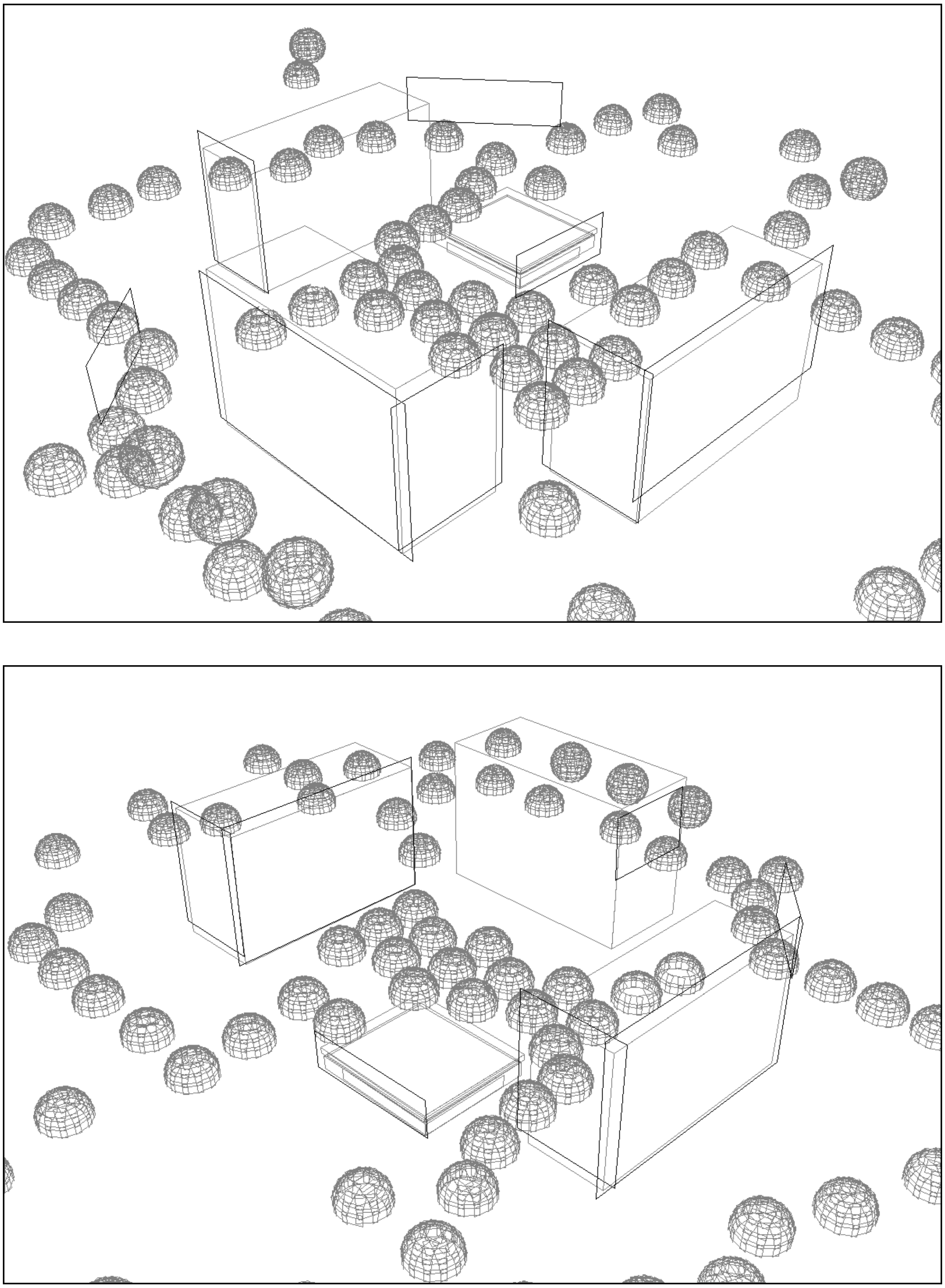


Figure 5-20: Raw model surfaces.

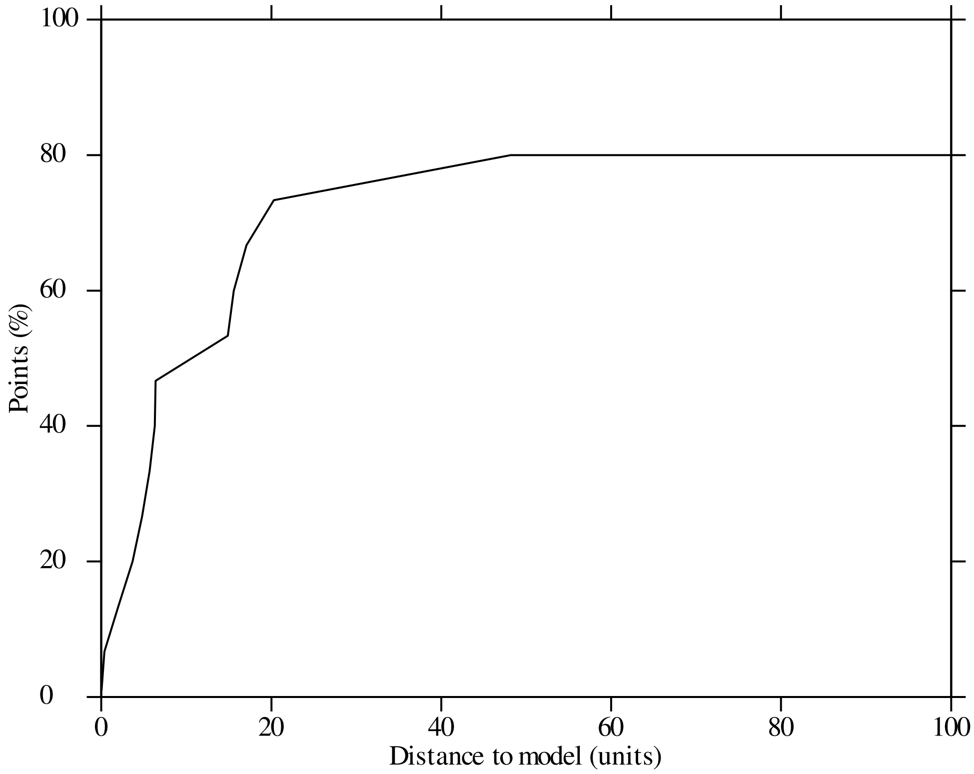
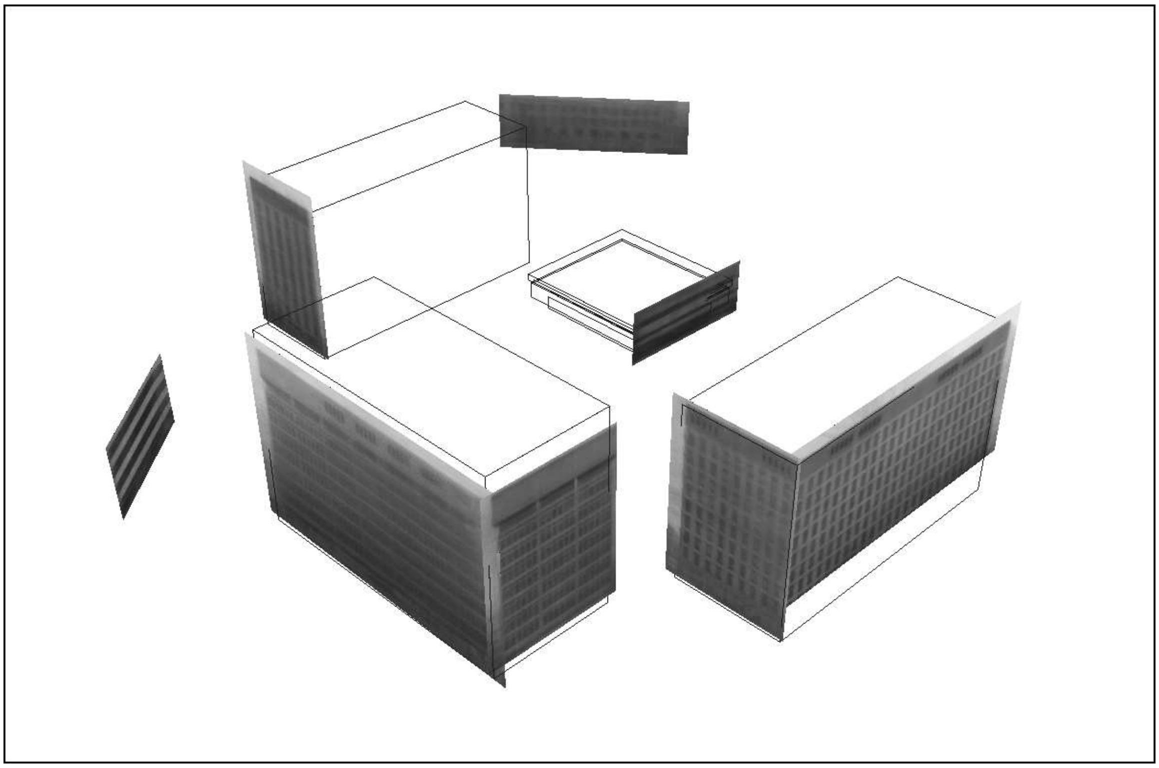


Figure 5-21: Distribution of error for model surfaces.

5.6 Discussion

This chapter uses several simple geometric constraints to remove virtually all false positives from the purely local reconstruction described in Chapter 4. After imposing consistent calibration updates, removing multiple contributions and grouping, the remaining surfels are excellent seeds for growing surfaces. Of the 16 surfaces in the reference model, 12 were recovered. All of the remaining surfaces are severely occluded by trees. Nearly all of the images are similar to the upper left-hand image of Figures 4-4 and 4-6. In spite of this several surfels were recovered on two of the surfaces, however they did not survive the grouping process. Figure 5-23 shows the results of growing these surfels. The top shows the raw surfels after growing and the bottom shows reconstructed textures. In addition to being severly occluded by trees, the other two surfaces have very little texture and one of them suffers from a lack of data. Three surfaces from adjacent buildings not contained in the model were also recovered. The face near the top center of the upper image in Figure 5-22 is from the Parson's lab. The surfaces on the left of the upper and the right of the lower image is from Draper lab.

A summary of the computation required is presented in Table 5.1. The major steps of our algorithm and the section which describes them are listed along with the elapsed and cpu runtimes, number of surfels output, and complexity. A total of 64.5 hours of cpu time was needed to produce the textured model



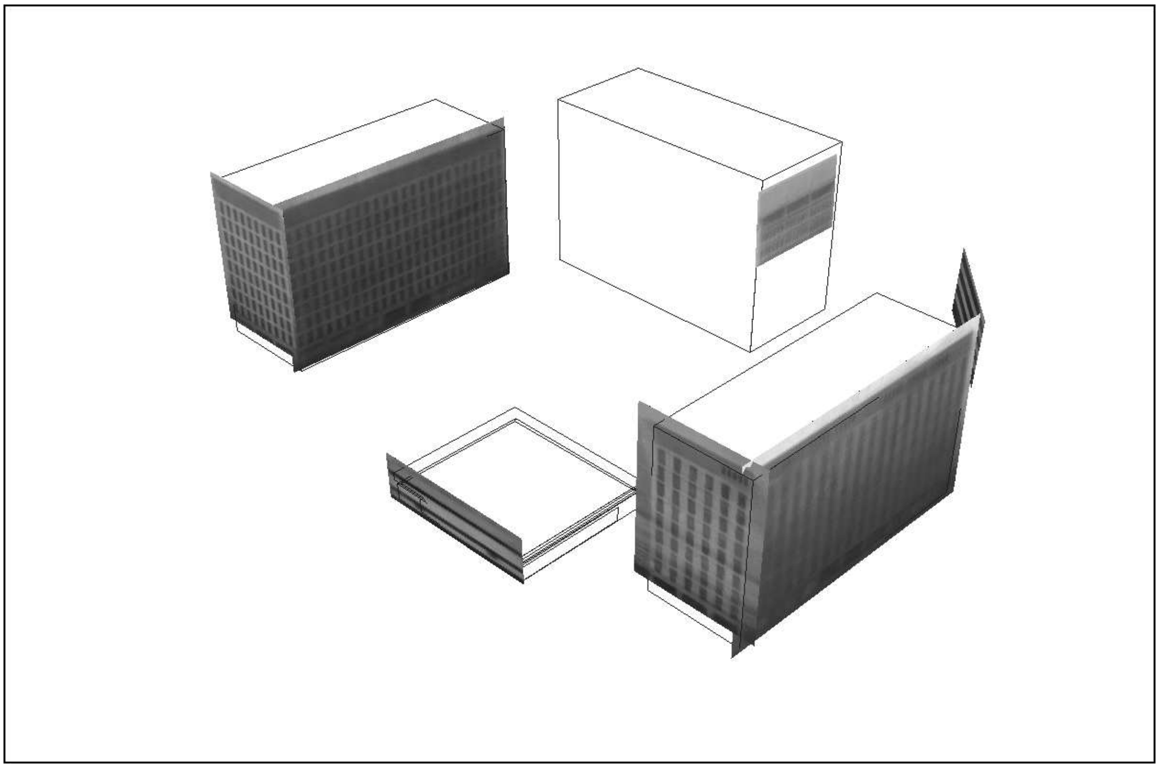
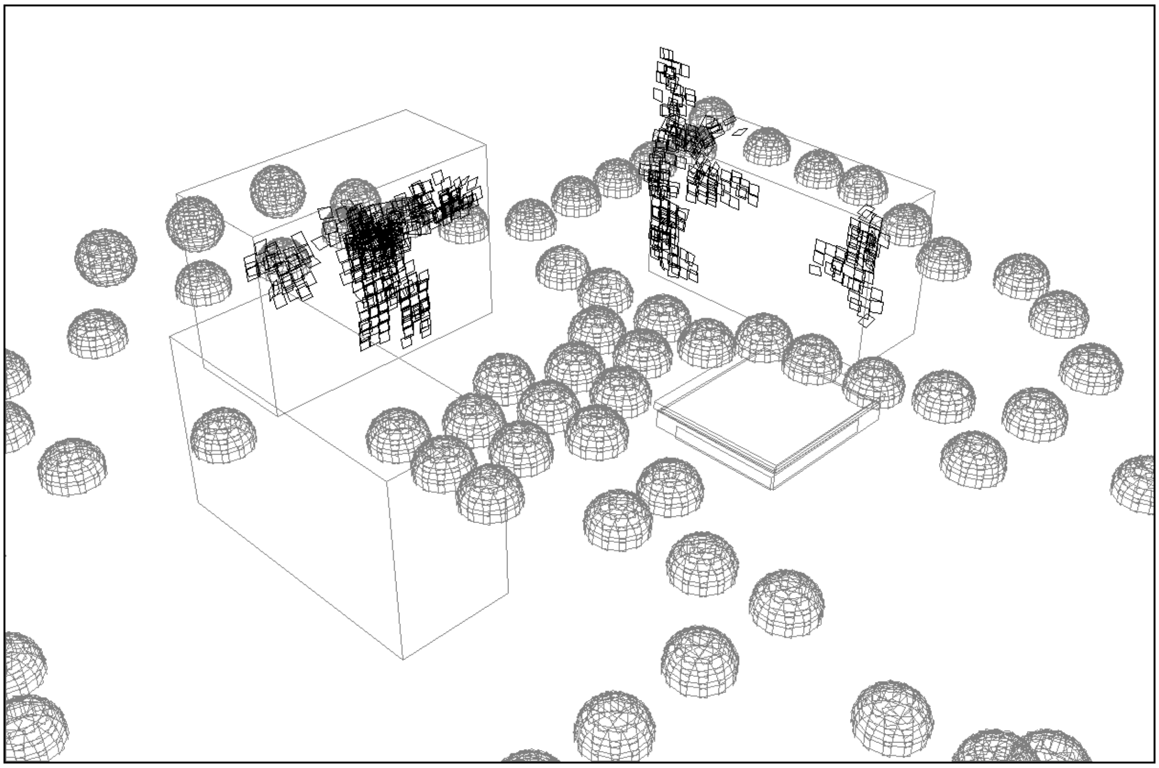


Figure 5-22: Textured model surfaces.



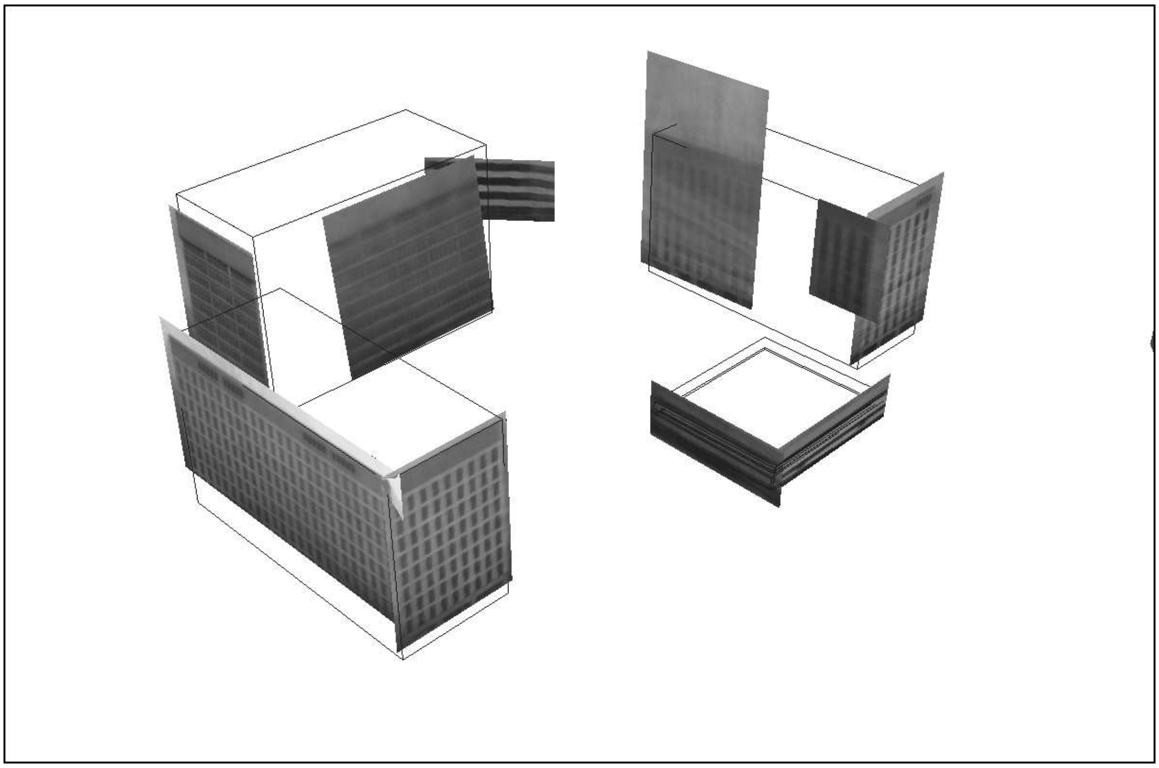


Figure 5-23: Textured model surfaces with two additional surfaces.

Step	Section	Time		Surfels	Complexity
		Elapsed	cpu		
Detection & Localization	4.2 4.3	42hrs†	60hrs	54,212	$O(V \times N \times S)$
Camera Updates	5.1	18.5min	17min	2977	$O(N \times \#^2)$
1 Pixel, 1 Surfel	5.2	3.5min	2min	1636	$O(N \times \#^2)$
Group	5.3	4min	2.5min	1272	$O(\#^2)$
Grow	5.4	6hrs†	4hrs	3007	$O(A \times N)$
Model	5.5	11.25min	9.5min	15	$O(\#)$
Texture	5.5	24.5min	15.5min	15	$O(A \times N)$

Table 5.1: Run times and orders of growth.

shown in Figure 5-22. This is less than 1 minute of cpu time per image and less than 0.2 seconds of cpu time per test point. The "Detection & Localization" and "Grow" steps were performed in parallel on a cluster of 32 400Mhz dual processor Pentium II machines running linux. The remaining steps were performed on a single 200Mhz uniprocessor Pentium Pro machine also running linux. The two times marked with daggers reflect the speed up of parallel processing. All others (including cpu time for "Detection & Localization" and "Grow") are total times. The complexities shown are upper bounds. V is the reconstruction volume, N is the total number of images, S is the surfel size (area in world coordinates), $\#$ is the total number of recovered surfels, and A is the total surface area (in world coordinates) of the scene to be reconstructed.

Detection & Localization

As noted above, the reported elapsed time reflects parallel execution. The total elapsed time is 1,260 hrs (42 hrs $\times$ 30 nodes). The cpu time required is $1/20$ of the total elapsed time. The major cause of this is I/O. The reconstruction is divided into 6000 chunks. Each of these chunks is reconstructed independently and requires loading a large image set from a file server across a network. The work required to test a single surfel is linearly proportional to surfel area in world coordinates (S) and the number of images which view the surfel. The total number of surfels tested is linearly proportional to the reconstruction volume. N is an upper bound on the number of images which view a surfel, but it is not a tight bound. For example, only a fraction of the dataset can image a particular surfel. Therefore, if the dataset is acquired with roughly a fixed node density and images only contribute to surfels which are not too far away (Section 4.2), we would expect the number of images which view a surfel to eventually become independent of N. In essence, once the dataset has reached a certain size, new images extend the reconstruction volume but do not affect the local reconstructions at the core of the volume. For large data sets the expected order of growth is $O(V \times S)$

Camera Updates

The work required to update a single camera is proportional to the number of surfels it contributes to squared and a total of N cameras must be updated. $\#$ is an upper bound on the number of surfels which a camera can view, but it is not a tight one. Similar to the discussion in "Detection & Localization", once the reconstruction reaches a certain size, the number of surfels imaged by a given camera should remain roughly constant. Spatial hashing can be used to help exploit this property. Thus, for large reconstructions the expected order of growth is $O(N)$

1 Pixel, 1 Surfel

The work required to impose the "1 Pixel, 1 Surfel" constraint for a single image contributing to a given surfel is linearly proportional to the number of surfels in a cone formed by the camera's center of projection and the surfel and extending out to the maximum allowed distance from the camera. This constraint must be imposed for each image contributing to each of the $\#$ surfels. If images can contribute only to surfels which are not too close (Section 4.2), then once the reconstruction reaches a certain size, the number of surfels in the cone should remain roughly constant. Therefore, for large reconstructions with large data sets the expected order of growth is $O(\#)$.

Group

The work required to group a surfel is linearly proportional to the number of surfels in the vicinity and all $\#$ surfels must be grouped. The number of surfels in a given area is bounded and by using spatial hashing the expected order of growth becomes $O(\#)$.

Grow

The comments in "Detection & Localization" about elapsed time apply here, except that only 15 surfaces were grown, therefore only 15 nodes were used. Assuming that the growing algorithm effectively stops at surface boundaries, there are a total A/S locations on all faces to be tested. For the faces shown in Figure 5-22 this is a good assumption. For large data sets, each location tested requires $O(S)$ work, giving an expected order of growth of $O(A)$.

Model

The work required to fit a (plane) model is linearly proportional to the number of surfels in the group. Since every surfel belongs to a group, the given order of growth $O(\#)$ is a tight bound.

Texture

The work required to extract the texture associated with a face is linearly proportional to the area of the face and the number of images which view it. N is an upper bound on the number of images which can view a face and since a face can be of arbitrary size it is not clear that we can do better. Therefore, the order of growth is $O(A \times N)$.

Chapter 6

Conclusions

This thesis presents a novel method for automatically recovering dense surfels using large sets (1000's) of calibrated images taken from arbitrary positions within the scene. Physical instruments, such as Global Positioning System (GPS), inertial sensors, and inclinometers, are used to estimate the position and orientation of each image. Long baseline images improve the accuracy; short baselines and the large number of images simplify the correspondence problem. The initial stage of the algorithm is completely local enabling parallelization and scales linearly with the number of images. Subsequent stages are global in nature, exploit geometric constraints, and scale quadratically with the complexity of the underlying scene.

We describe techniques for:

- Detecting and localizing surfels.
- Refining camera calibration estimates and rejecting false positive surfels.
- Grouping surfels into surfaces.
- Growing surfaces along a two-dimensional manifold.
- Producing high quality, textured three-dimensional models from surfaces.

Some of our approach's most important characteristics are:

- It is fully automatic.
- It uses and refines noisy calibration estimates.
- It compensates for large variations in illumination.
- It matches image data directly in three-dimensional space.
- It tolerates significant soft occlusion (e.g. tree branches).
- It associates, at a fundamental level, an estimated normal (eliminating the frontal-planar assumption) and texture with each surfel.

Our algorithms also exploit several geometric constraints inherent in three-dimensional environments and scale well to large sets of images. We believe that these characteristics will be important for systems which automatically recover large-scale high-quality three-dimensional models. A set of about 4000 calibrated images was used to test our algorithms. The results presented in this thesis demonstrate that they can be used for three-dimensional reconstruction. To our knowledge, the City Scanning project (e.g. [Coorg, 1998] and the work presented in this thesis) is the first to produce high-quality textured models from such large image sets. The image sets used in this thesis are nearly two orders of magnitude larger than the largest sets used by other approaches. The approach presented in this thesis, recovering dense surfels by matching raw image data directly in three-dimensional space, is unique amoung the City Scanning approaches.

6.1 Future Work

The major limitation of the work presented in this thesis is the difficulty of performing reconstruction "through the trees". A large part of the difficulty is caused by matching against $s_j^\star$. If $s_j^\star$ is not representative of the underlying texture, then the match set $\dot{\mathbf{s}}_j$ will most likely be insufficient resulting in a false negative. If we knew the underlying texture and used it as $s_j^\star$, the match set should improve significantly. Similarly, estimating the underlying texture, rather than simplying selecting one of the views, and using it as $s_j^\star$ should also improve the match. Potentially this would allow us recover the missing faces in Figure 5-22.

Several other areas could also benefit from further exploration:

- Develop alternate techniques for exploring the volume of interest.
- Explore improvements/optimizations to the basic algorithms.
- Develop a more sophisticated model of illumination and reflectance to improve the correction.
- Explore fitting a broader class of models to surfels.
- Develop techniques to enhance the recovered texture.

6.1.1 Exploring the Volume of Interest

As noted in Section 4.2, we exhaustively test all of the nearly 2×10^6 points in the volume of interest. About 2% of the tested points produce surfels and about 6% of those are consistent with a single set of camera calibration updates. One way of reducing the number of points that need to be tested is to use techniques such as binocular stereo to generate test points that are likely

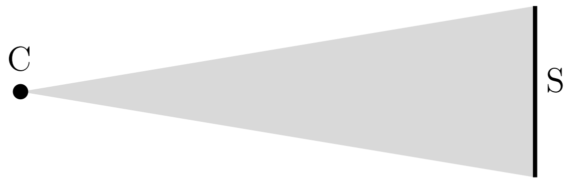


Figure 6-1: Free space.

to produce surfels. Because of error in the estimated three-dimensional position, a small volume around the point should be tested. If a surfel is detected and localized, additional test points can be generate nearby in a manner similar to that used in Section 5.4 to grow surfaces. Another possible approach is to keep track of empty space. For example, if surfel S in Figure 6-1 corresponds to an actual surface, then the shaded area must be empty space and need not be tested. Conservatively estimating regions of empty space during the reconstruction process could significantly reduce the number of points actually tested. Argus, our image acquisition platform is pushed for node to node while obtaining a dataset and the path it follows also identifies empty space that limit test points. In a similar fashion, it may prove useful to keep track of the general location of soft occluders. For example, views which do not have soft occlusion could be selected preferentially or weighted more heavily.

6.1.2 Improving the Basic Algorithm

Several components of the basic algorithm could be improved. Several parameters such as surfel size and thresholds for interest and uniqueness are selected and utilized on a global basis. These values are dependent upon the data and their values should reflect this dependence. For example, the overall image brightness and contrast have a large impact on interest and uniqueness scores. For various reasons (geometry, sun position, etc.), some building faces are frequently brightly illuminated and others are nearly always in shadow. Using the same threshold for both tends to produce a large number of false positives in the former case and a large number of false negatives in the latter. Thresholds based upon the imaging conditions should produce better results. The quality of matches could also be improved by performing comparisons at the appropriate resolution. The effective resolution of reprojected regions vary with viewing condition (e.g. distance and foreshortening). Source images could be loaded into a pyramid of resolutions and prior to matching, a level selected such that all the reprojected textures are of similar resolution. This technique should significantly improve the algorithms ability to match distant or highly forshorted views. As noted in Section 4.3, updating a surfels position and orientation should ideally be done simultaneously. Formulations which combine these optimizations may localize surfels more accurately. And finally, as noted in Section 3.3, our masking algorithm is conservative. More sophisticated methods of identifying outlier pixels would enable the masks to be more widely used and should result in fewer false negatives.

6.1.3 Illumination and Reflectance Models

The correction proposed in Section 3.2 does a good job of compensating for changes in viewing condition (illumination, view point, reflectance, etc.), but the extra degrees of freedom also admit false positives such as shown in Figure 3-5. One way to reduce the number of false positives is to constrain the corrections by measuring the viewing conditions. For example, the total irradiance arriving at a node and an estimate of cloud cover might be sufficient for a rough estimate of the correction. The corrections used to extract the textures shown in Figure 5-22 are consistent within a face, but not necessarily between faces. A direct estimate might make it possible to locate all corrections in a global correction space.

As demonstrated in Figures 4-3 and 4-9 in some cases our method is capable of compensating for specular reflection. These corrections are generally rejected in an attempt to limit false positives. One way to improve matching for surfaces with substantial specular characteristics is to decompose the reflectance into specular and diffuse components and use just the diffuse component for matching. Nayar *et al.* [Nayar *et al.*, 1997] show that the specular component of an objects reflectance tends to be linearly polarized while the diffuse component is not. Using this property and a polarization image[1], Nayar *et al.* present an algorithm for separating the components. Wolff [Wolff and Andreou, 1995, Wolff, 1997] demonstrates a practical device for acquiring polarization images.

Finally, given a number of images of the same world surface taken under different viewing conditions, it should be possible to estimate its bidirectional reflectance distribution function (BRDF) [Horn, 1986]. Tagare and DeFigueredo [Tagare and deFigueiredo, 1993], Oren and Nayar [Oren and Nayar, 1995], and Dana *et al.* [Dana *et al.*, 1996] present a techniques for fitting BRDF's to image data.

6.1.4 More Descriptive Models

The models extracted in Section 5.5 are composed of simple rectangles. The buildings in our dataset are well modeled by planar faces, however many buildings have components which are not. Fitting general models to data is an open problem, however expanding the set of primitives to include cones, cylinders, and spheres as well as planes would greatly increase the descriptive power of our models. Further, the faces recovered should be part of a closed solid. This constraint could easily be enforced by hypothesizing *missing* faces. Imposing

[1] A set of color images of the same scene acquired from the same location each imaged with a polarization filter at a different orientation

this constraint would result in recovering two of the four unrecovered faces. As part of the City Scanning project, Cutler [1999] has investigated aggregating reconstructions from multiple sources and imposing solid constraints.

6.1.5 Better Textures

As described in Section 5.5 the textures displayed in Figure 5-22 are generated by averaging the illumination corrected image data. The results are good but can be improved. Iteratively estimating the *average* texture should improve the results. For example, the current average texture is used to reestimate the illumination corrections for each image and weights for each pixel of each image. The new illumination corrections and weights are then used to reestimate the average texture until it converges. No attempt has been made to *align* the data from individual images. Warping techniques similar to those described by Szeliski [Szeliski, 1996, Szeliski and Shum, 1997] should improve the quality of the extracted texture.

Appendix A

Camera Model

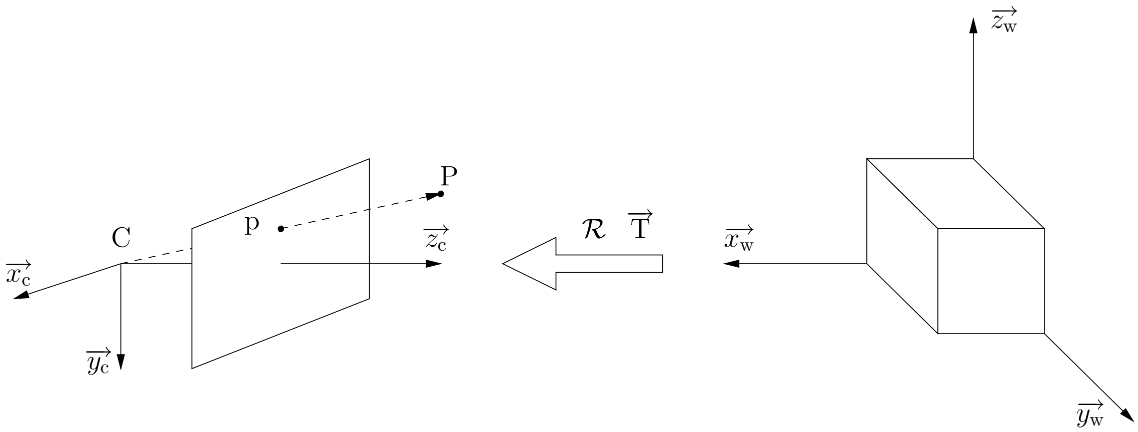


Figure A-1: Camera model.

The pin-hole camera model utilized throughout this thesis uses the perspective projection model of image formation. A 4×3 matrix maps points expressed in homogeneous coordinates from $\mathcal{P}^3$ to $\mathcal{P}^2$. The left side of Figure A-1 shows a camera centered Cartesian coordinate system. The camera's optical axis is coincident with the z axis and its center of projection (C) is at the origin. The image plane is parallel to the xy plane and located a distance f from the origin. Even though the image plane is not required to be parallel to the xy plane, most camera models do not explicitly consider this possibility. Instead the effects of image plane pitch θ_x and tilt θ_y are lumped in with lens distortion under the heading of thin prism distortion. We assume that $\theta_\mathrm{x} = \theta_\mathrm{y} = 0$ for consistency with traditional pin-hole camera models. The point where the image plane and the optical axis intersect is known as the principal point. Under perspective projection a point $\mathrm{P_c} = [X_\mathrm{c}, Y_\mathrm{c}, Z_\mathrm{c}, 1]$ projects to point $\mathrm{p} = [x, y, 1]$ on the image plane by the following equations:

$$
\begin{aligned}
x &= f\frac{X_\mathrm{c}}{Z_\mathrm{c}} \\
y &= f\frac{Y_\mathrm{c}}{Z_\mathrm{c}}
\end{aligned}
$$

In practice, we are not able to directly access the image plane coordinates.

Instead we have access to an array of pixels in computer memory. In order to understand the relationship between image plane coordinates and the array of pixels, we must examine the imaging process. The image plane of a CCD camera is a rectangular array of discrete light sensitive elements. The output from each of these elements is an analog signal proportional to the amount of light incident upon it. The values for each of these elements are read out one element at a time row after row until the entire sensor array has been read. The analog signal is converted to digital values by internal circuitry. Digital cameras use this signal directly. Analog cameras require the use of an additional external A to D converter commonly known as a frame grabber. In addition, color cameras require the signals from adjacent red, green, and blue sensor sites to be combined. The result is an array of digital values which can be read into the memory of a computer. These processes introduce noise and ambiguity in the the image data (e.g. pixel jitter which extreme cases can cause the x and y axes to appear non-orthogonal or skewed [Lenz and Tsai, 1988]). Most camera models omit skew angle θ_{xy} (the angle between the x and y axes minus $90°$). We include $\theta_{\mathrm{x}y}$ but assume it is 0. A single element of the array in memory is commonly called a pixel. We will refer to the row and column number of a given pixel as y' and x' respectively. Several parameters are defined to quantify the relationship between the array in memory and the coordinate system of the image plane. x_0 and y_0 are the pixel coordinates of the principal point. s_{x} and s_{y} are the number of pixels in memory per unit distance in the x and y direction of the image plane. These parameters along with f and θ_{xy} are intrinsic or internal camera calibration parameters. The projection of point $\mathrm{P_c}$ to point $\mathrm{p}' = [x', y', 1]$ in memory is described by the following equations

$$\begin{aligned} x' &= f s_{\mathrm{x}} \frac{x_{\mathrm{c}}}{z_{\mathrm{c}}} + x_0 \\ y' &= f s_{\mathrm{y}} \frac{y_{\mathrm{c}}}{z_{\mathrm{c}}} + y_0, \end{aligned}$$

or more compactly

$$\begin{aligned} \mathrm{p} &= \mathrm{P}\mathcal{C} \qquad (A.1) \\ &= \mathrm{P} \begin{bmatrix} s_{\mathrm{x}} & 0 & 0 \\ \tan\theta_{\mathrm{xy}} & s_{\mathrm{y}} & 0 \\ x_0 & y_0 & 1/f \end{bmatrix} \end{aligned}$$

$\mathcal{C}$ is a 3×3 lower triangular matrix which contains the internal camera parameters.

The right half of Figure A-1 shows an arbitrary Cartesian coordinate system which we will refer to as the world or absolute coordinate system. A point P_{w} in the world coordinate system is transformed into the camera centered coordinate system by the following equation:

$$P_{\mathrm{c}} = P_{\mathrm{w}}\mathcal{R} + \overrightarrow{\mathrm{T}}$$

or

$$P_{\mathrm{c}} = P_{\mathrm{w}} \begin{bmatrix} \mathcal{R} & \mathbf{0} \\ \overrightarrow{\mathrm{T}} & 1 \end{bmatrix} \tag{A.2}$$

Where $\mathcal{R}$ is a 3×3 orthonormal rotation matrix and $\overrightarrow{\mathrm{T}}$ is a 1×3 translation vector. This matrix is commonly referred to as the extrinsic or external camera parameters.

The pin-hole camera is a linear model and is not able to model nonlinear effects such as lens distortion. The major categories of lens distortion are:

1. Radial distortion - the path of a light ray traveling from the object to the image plane through the lens is not always a straight line.

2. Decentering distortion - the optical axis of individual lens components are not always collinear.

3. Thin prism distortion - the optical axis of the lens assembly is not always perpendicular to the image plane.

Appendix B

Gradient Derivations

This appendix presents the symbolic gradient expressions used along with conjugate gradient methods to find the *optimum* shift (Section B.1) and surfel normal (Section B.2).

B.1 Gradient Expression for Updating Shifts

For simplicity, we derive the results for a single color channel (red). To obtain expressions for the remaining color channels simply substitute the appropriate color channel values. Equation 4.3 can be rewritten using Equations 2.9 and 3.11 as

$$\epsilon(u,v) = \epsilon_{\mathrm{r}}(u,v) + \epsilon_{\mathrm{g}}(u,v) + \epsilon_{\mathrm{b}}(u,v) \tag{B.1}$$

where

$$\epsilon_{\mathrm{r}}(u,v) = \sum_{{}^{y}_{x}\mathrm{S}_j \in \mathrm{S}_j} \frac{\left((m_{\mathrm{r}} r_1 + d_{\mathrm{r}}) - r_2\right)^2}{\sigma_{\mathrm{r}}^2} \Bigg/ \sum_{{}^{y}_{x}\mathrm{S}_j \in \mathrm{S}_j} 1 \,, \tag{B.2}$$

$$r_1 = \mathrm{r}\left({}^{y+v}_{x+u}\mathrm{s}^{i}_{j}\right), \tag{B.3}$$

and

$$r_2 = \mathrm{r}\left({}^{y}_{x}\mathrm{s}^{\star}_{j}\right). \tag{B.4}$$

Using linear regression to calculate m_{r} and d_{r} yields[1]

$$m_{\mathrm{r}} = \frac{\sum\limits_{x,y} 1 \sum\limits_{x,y} r_1 r_2 - \sum\limits_{x,y} r_1 \sum\limits_{x,y} r_2}{\sum\limits_{x,y} 1 \sum\limits_{x,y} r_1^2 - \sum\limits_{x,y} r_1 \sum\limits_{x,y} r_1} \tag{B.5}$$

and

$$d_{\mathrm{r}} = \frac{\sum\limits_{x,y} r_1^2 \sum\limits_{x,y} r_2 - \sum\limits_{x,y} r_1 \sum\limits_{x,y} r_1 r_2}{\sum\limits_{x,y} 1 \sum\limits_{x,y} r_1^2 - \sum\limits_{x,y} r_1 \sum\limits_{x,y} r_1}. \tag{B.6}$$

[1] To simplify the presentation x,y is used in place of ${}^{y}_{x}\mathrm{S}_j \in \mathrm{S}_j$.

The gradient of Equation B.1 is

$$\begin{aligned}\nabla_{u,v}\epsilon(u,v) &= \nabla_{u,v}\epsilon_{\mathrm{r}}(u,v)+\nabla_{u,v}\epsilon_{\mathrm{g}}(u,v)+\nabla_{u,v}\epsilon_{\mathrm{b}}(u,v) \\ &= \left[\frac{\partial\epsilon_{\mathrm{r}}}{\partial u},\frac{\partial\epsilon_{\mathrm{r}}}{\partial v}\right]+\left[\frac{\partial\epsilon_{\mathrm{g}}}{\partial u},\frac{\partial\epsilon_{\mathrm{g}}}{\partial v}\right]+\left[\frac{\partial\epsilon_{\mathrm{b}}}{\partial u},\frac{\partial\epsilon_{\mathrm{b}}}{\partial v}\right]\end{aligned} \tag{B.7}$$

Again for simplicity, we derive only $\frac{\partial\epsilon_{\mathrm{r}}}{\partial u}$. To obtain an expression for $\frac{\partial\epsilon_{\mathrm{r}}}{\partial v}$ simply substitute v for u. Differentiating Equations B.2, B.5, and B.6 with respect to u gives

$$\frac{\partial\epsilon_{\mathrm{r}}}{\partial u} = \sum_{x,y}\frac{((m_{\mathrm{r}}r_1+d_{\mathrm{r}})-r_2)\left(m_{\mathrm{r}}\frac{\partial r_1}{\partial u}+\frac{\partial m_{\mathrm{r}}}{\partial u}r_1+\frac{\partial d_{\mathrm{r}}}{\partial u}-\frac{\partial r_2}{\partial u}\right)}{2\sigma_{\mathrm{r}}^2}\Bigg/\sum_{x,y}1\,, \tag{B.8}$$

$$\begin{aligned}\frac{\partial m}{\partial u} = {}& \Bigg(\Bigg(\sum_{x,y}1\sum_{x,y}r_1^2-\sum_{x,y}r_1\sum_{x,y}r_1\Bigg)\Bigg(\sum_{x,y}1\sum_{x,y}r_1\frac{\partial r_2}{\partial u}+\sum_{x,y}1\sum_{x,y}\frac{\partial r_1}{\partial u}r_2- \\ & \sum_{x,y}r_1\sum_{x,y}\frac{\partial r_2}{\partial u}-\sum_{x,y}\frac{\partial r_1}{\partial u}\sum_{x,y}r_2\Bigg)-\Bigg(\sum_{x,y}1\sum_{x,y}r_1r_2-\sum_{x,y}r_1\sum_{x,y}r_2\Bigg) \\ & \Bigg(2\sum_{x,y}1\sum_{x,y}r_1\frac{\partial r_1}{\partial u}-2\sum_{x,y}r_1\sum_{x,y}\frac{\partial r_1}{\partial u}\Bigg)\Bigg)\Bigg/\Bigg(\sum_{x,y}1\sum_{x,y}r_1^2-\sum_{x,y}r_1\sum_{x,y}r_1\Bigg)^2,\end{aligned} \tag{B.9}$$

and

$$\begin{aligned}\frac{\partial b}{\partial u} = {}& \Bigg(\Bigg(\sum_{x,y}1\sum_{x,y}r_1^2-\sum_{x,y}r_1\sum_{x,y}r_1\Bigg)\Bigg(2\sum_{x,y}r_1\frac{\partial r_1}{\partial u}\sum_{x,y}r_2+\sum_{x,y}r_1^2\sum_{x,y}\frac{\partial r_2}{\partial u}- \\ & \sum_{x,y}\frac{\partial r_1}{\partial u}\sum_{x,y}r_1r_2-\sum_{x,y}r_1\sum_{x,y}\frac{\partial r_1}{\partial u}r_2\sum_{x,y}r_1\sum_{x,y}r_1\frac{\partial r_2}{\partial u}\Bigg)-\Bigg(\sum_{x,y}r_1^2\sum_{x,y}r_2-\sum_{x,y}r_1\sum_{x,y}r_1r_2\Bigg) \\ & \Bigg(2\sum_{x,y}1\sum_{x,y}r_1\frac{\partial r_1}{\partial u}-2\sum_{x,y}r_1\sum_{x,y}\frac{\partial r_1}{\partial u}\Bigg)\Bigg)\Bigg/\Bigg(\sum_{x,y}1\sum_{x,y}r_1^2-\sum_{x,y}r_1\sum_{x,y}r_1\Bigg)^2.\end{aligned} \tag{B.10}$$

u and v are shifts in the $\overrightarrow{x_{\mathrm{c}}}$ and $\overrightarrow{y_{\mathrm{c}}}$ directions respectively in image i, therefore differentiating Equations B.3 and B.4 with respect to u produces

$$\begin{aligned}\frac{\partial r_1}{\partial u} &= \frac{\partial r_1}{\partial x}\frac{\partial x}{\partial u} \\ &= \frac{\partial r_1}{\partial x,}\end{aligned} \tag{B.11}$$

and

$$\begin{aligned}\frac{\partial r_2}{\partial u} &= \frac{\partial r_2}{\partial x}\frac{\partial x}{\partial u} \\ &= 0.\end{aligned} \tag{B.12}$$

Substituting Equations B.9, B.10, B.11, and B.12 into Equation B.8 results in an expression which depends only upon the red channel of the raw image data and the gradient of the red channel (r_1, r_2, and $\frac{\partial r_1}{\partial x}$). Similar results apply for $\frac{\partial\epsilon_{\mathrm{r}}}{\partial v}$ as well as the green and blue channels.

B.2 Gradient Expression for Updating Normals

Again for simplicity, we derive the results for a single color channel (red). To obtain expressions for the remaining color channels simply substitute the appropriate color channel values. Similar to Equation B.1, Equation 4.7 can be rewritten as

$$\epsilon(\theta,\phi) = \epsilon_{\mathrm{r}}(\theta,\phi) + \epsilon_{\mathrm{g}}(\theta,\phi) + \epsilon_{\mathrm{b}}(\theta,\phi) \tag{B.13}$$

where

$$\epsilon_{\mathrm{r}}(\theta,\phi) = \sum_i \left(\sum_{x,y} \frac{((m_{\mathrm{r}} r_1 + d_{\mathrm{r}}) - r_2)^2}{\sigma_{\mathrm{r}}^2} \Big/ \sum_{x,y} 1 \right), \tag{B.14}$$

$$r_1 = \mathrm{r}\left(\mathcal{T}({}_x^y\mathrm{S}_j, \mathrm{I}^i) + [u,v]\right), \tag{B.15}$$

and

$$r_2 = \mathrm{r}\left(\mathcal{T}({}_x^y\mathrm{S}_j, \mathrm{I}^\star)\right). \tag{B.16}$$

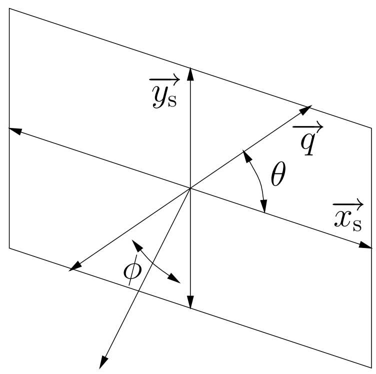


Figure B-1: Surfel with attached coordinate system.

Figure B-1 shows surfel S_j with an attached coordinate system. Points on S_j can be specified as an offset along $\overrightarrow{x_{\mathrm{s}}}$ and $\overrightarrow{y_{\mathrm{s}}}$ relative to P_j,

$${}_x^y\mathrm{S}_j = \mathrm{P}_j + xw\overrightarrow{x_{\mathrm{s}}} + yw\overrightarrow{y_{\mathrm{s}}}. \tag{B.17}$$

w is a scale parameter which determines the spacing between points on the surfel. The normal is parameterized as a small rotation relative to the n_j. θ specifies the axis of rotation

$$\overrightarrow{q} = \overrightarrow{x_{\mathrm{s}}}\cos\theta + \overrightarrow{y_{\mathrm{s}}}\sin\theta \tag{B.18}$$

and ϕ specify the angle of rotation about $\overrightarrow{q}$. Accounting for the updated normal, Equation B.17 becomes

$${}_x^y\mathrm{S}_j = \mathrm{P}_j + xw\mathcal{R}(\theta,\phi)\overrightarrow{x_{\mathrm{s}}} + yw\mathcal{R}(\theta,\phi)\overrightarrow{y_{\mathrm{s}}} \tag{B.19}$$

where

$$\mathcal{R}(\theta,\phi) = \tag{B.20}$$
$$\begin{bmatrix} q_{\mathrm{x}}^2 + (1-q_{\mathrm{x}}^2)\cos\phi & q_{\mathrm{x}}q_{\mathrm{y}}(1-\cos\phi) + q_{\mathrm{z}}\sin\phi & q_{\mathrm{x}}q_{\mathrm{z}}(1-\cos\phi) - q_{\mathrm{y}}\sin\phi \\ q_{\mathrm{x}}q_{\mathrm{y}}(1-\cos\phi) - q_{\mathrm{z}}\sin\phi & q_{\mathrm{y}}^2 + (1-q_{\mathrm{y}}^2)\cos\phi & q_{\mathrm{y}}q_{\mathrm{z}}(1-\cos\phi) + q_{\mathrm{x}}\sin\phi \\ q_{\mathrm{x}}q_{\mathrm{z}}(1-\cos\phi) + q_{\mathrm{y}}\sin\phi & q_{\mathrm{y}}q_{\mathrm{z}}(1-\cos\phi) - q_{\mathrm{x}}\sin\phi & q_{\mathrm{z}}^2 + (1-q_{\mathrm{z}}^2)\cos\phi \end{bmatrix}.$$

The image point to which ${}^{y}_{x}\mathrm{S}_j$ projects is

$$\mathcal{T}({}^{y}_{x}\mathrm{S}_j, \mathrm{I}) = [x', y']$$

where

$$x' = f\frac{\overrightarrow{x_{\mathrm{c}_i}} \cdot ({}^{y}_{x}\mathrm{S}_j - \mathrm{C}_i)}{\overrightarrow{z_{\mathrm{c}_i}} \cdot ({}^{y}_{x}\mathrm{S}_j - \mathrm{C}_i)} + x_0 + u, \tag{B.21}$$

and

$$y' = f\frac{\overrightarrow{y_{\mathrm{c}_i}} \cdot ({}^{y}_{x}\mathrm{S}_j - \mathrm{C}_i)}{\overrightarrow{z_{\mathrm{c}_i}} \cdot ({}^{y}_{x}\mathrm{S}_j - \mathrm{C}_i)} + y_0 + v. \tag{B.22}$$

The gradient of Equation B.13 is

$$\begin{aligned} \nabla_{\theta,\phi}\epsilon(\theta,\phi) &= \nabla_{\theta,\phi}\epsilon_{\mathrm{r}}(\theta,\phi) + \nabla_{\theta,\phi}\epsilon_{\mathrm{g}}(\theta,\phi) + \nabla_{\theta,\phi}\epsilon_{\mathrm{b}}(\theta,\phi) \\ &= \left[\frac{\partial\epsilon_{\mathrm{r}}}{\partial\theta}, \frac{\partial\epsilon_{\mathrm{r}}}{\partial\phi}\right] + \left[\frac{\partial\epsilon_{\mathrm{g}}}{\partial\theta}, \frac{\partial\epsilon_{\mathrm{g}}}{\partial\phi}\right] + \left[\frac{\partial\epsilon_{\mathrm{b}}}{\partial\theta}, \frac{\partial\epsilon_{\mathrm{b}}}{\partial\phi}\right] \end{aligned} \tag{B.23}$$

First we will examine $\frac{\partial}{\partial\theta}$. Differentiating Equation B.14 gives

$$\frac{\partial\epsilon_{\mathrm{r}}}{\partial\theta} = \sum_{x,y} \frac{((m_{\mathrm{r}}r_1 + d_{\mathrm{r}}) - r_2)\left(m_{\mathrm{r}}\frac{\partial r_1}{\partial\theta} + \frac{\partial m_{\mathrm{r}}}{\partial\theta}r_1 + \frac{\partial d_{\mathrm{r}}}{\partial\theta} - \frac{\partial r_2}{\partial\theta}\right)}{2\sigma_{\mathrm{r}}^2} \Big/ \sum_{x,y} 1\,. \tag{B.24}$$

Differentiating Equations B.9 and B.10 and using the chain rule produces

$$\frac{\partial m_{\mathrm{r}}}{\partial\theta} = \frac{\partial m_{\mathrm{r}}}{\partial x}\frac{\partial x}{\partial\theta} + \frac{\partial m_{\mathrm{r}}}{\partial y}\frac{\partial y}{\partial\theta} \tag{B.25}$$

and

$$\frac{\partial d_{\mathrm{r}}}{\partial\theta} = \frac{\partial d_{\mathrm{r}}}{\partial x}\frac{\partial x}{\partial\theta} + \frac{\partial d_{\mathrm{r}}}{\partial y}\frac{\partial y}{\partial\theta}. \tag{B.26}$$

The expressions for $\frac{\partial m_{\mathrm{r}}}{\partial x}$ and $\frac{\partial m_{\mathrm{r}}}{\partial y}$ are given in the last section. Differentiating Equations B.15 and B.16 and using the chain rule yields

$$\frac{\partial r_1}{\partial\theta} = \frac{\partial r_1}{\partial x}\frac{\partial x}{\partial\theta} + \frac{\partial r_1}{\partial y}\frac{\partial y}{\partial\theta} \tag{B.27}$$

and

$$\frac{\partial r_2}{\partial\theta} = \frac{\partial r_2}{\partial x}\frac{\partial x}{\partial\theta} + \frac{\partial r_2}{\partial y}\frac{\partial y}{\partial\theta}. \tag{B.28}$$

Equations B.21 and B.22 produce

$$\frac{\partial x}{\partial \theta} = f \frac{\overrightarrow{z_{\mathrm{c}_i}} \cdot ({}^{y}_{x}\mathrm{S}_j - \mathrm{C}_i)\, \overrightarrow{x_{\mathrm{c}_i}} \cdot \frac{\partial {}^{y}_{x}\mathrm{S}_j}{\partial \theta} - \overrightarrow{x_{\mathrm{c}_i}} \cdot ({}^{y}_{x}\mathrm{S}_j - \mathrm{C}_i)\, \overrightarrow{z_{\mathrm{c}_i}} \cdot \frac{\partial {}^{y}_{x}\mathrm{S}_j}{\partial \theta}}{\left(\overrightarrow{z_{\mathrm{c}_i}} \cdot ({}^{y}_{x}\mathrm{S}_j - \mathrm{C}_i)\right)^2} \tag{B.29}$$

and

$$\frac{\partial y}{\partial \theta} = f \frac{\overrightarrow{z_{\mathrm{c}_i}} \cdot ({}^{y}_{x}\mathrm{S}_j - \mathrm{C}_i)\, \overrightarrow{y_{\mathrm{c}_i}} \cdot \frac{\partial {}^{y}_{x}\mathrm{S}_j}{\partial \theta} - \overrightarrow{y_{\mathrm{c}_i}} \cdot ({}^{y}_{x}\mathrm{S}_j - \mathrm{C}_i)\, \overrightarrow{z_{\mathrm{c}_i}} \cdot \frac{\partial {}^{y}_{x}\mathrm{S}_j}{\partial \theta}}{\left(\overrightarrow{z_{\mathrm{c}_i}} \cdot ({}^{y}_{x}\mathrm{S}_j - \mathrm{C}_i)\right)^2}. \tag{B.30}$$

Differentiating Equation B.19 yields

$$\frac{\partial {}^{y}_{x}\mathrm{S}_j}{\partial \theta} = xw \frac{\partial \mathcal{R}(\theta, \phi)}{\partial \theta} \overrightarrow{x_{\mathrm{s}}} + yw \frac{\partial \mathcal{R}(\theta, \phi)}{\partial \theta} \overrightarrow{y_{\mathrm{s}}} \tag{B.31}$$

where

$$\frac{\partial \mathcal{R}(\theta, \phi)}{\partial \theta} = \tag{B.32}$$
$$\begin{bmatrix} 2q_{\mathrm{x}} \frac{\partial q_{\mathrm{x}}}{\partial \theta}(1 - \cos\phi) & (q_{\mathrm{x}} \frac{\partial q_{\mathrm{y}}}{\partial \theta} + \frac{\partial q_{\mathrm{x}}}{\partial \theta} q_{\mathrm{y}})(1 - \cos\phi) + \frac{\partial q_{\mathrm{z}}}{\partial \theta} \sin\phi & (q_{\mathrm{x}} \frac{\partial q_{\mathrm{z}}}{\partial \theta} + \frac{\partial q_{\mathrm{x}}}{\partial \theta} q_{\mathrm{z}})(1 - \cos\phi) - \frac{\partial q_{\mathrm{y}}}{\partial \theta} \sin\phi \\ (q_{\mathrm{x}} \frac{\partial q_{\mathrm{y}}}{\partial \theta} + \frac{\partial q_{\mathrm{x}}}{\partial \theta} q_{\mathrm{y}})(1 - \cos\phi) - \frac{\partial q_{\mathrm{z}}}{\partial \theta} \sin\phi & 2q_{\mathrm{y}} \frac{\partial q_{\mathrm{y}}}{\partial \theta}(1 - \cos\phi) & (q_{\mathrm{y}} \frac{\partial q_{\mathrm{z}}}{\partial \theta} + \frac{\partial q_{\mathrm{y}}}{\partial \theta} q_{\mathrm{z}})(1 - \cos\phi) + \frac{\partial q_{\mathrm{x}}}{\partial \theta} \sin\phi \\ (q_{\mathrm{x}} \frac{\partial q_{\mathrm{z}}}{\partial \theta} + \frac{\partial q_{\mathrm{x}}}{\partial \theta} q_{\mathrm{z}})(1 - \cos\phi) + \frac{\partial q_{\mathrm{y}}}{\partial \theta} \sin\phi & (q_{\mathrm{y}} \frac{\partial q_{\mathrm{z}}}{\partial \theta} + \frac{\partial q_{\mathrm{y}}}{\partial \theta} q_{\mathrm{z}})(1 - \cos\phi) - \frac{\partial q_{\mathrm{x}}}{\partial \theta} \sin\phi & 2q_{\mathrm{z}} \frac{\partial q_{\mathrm{z}}}{\partial \theta}(1 - \cos\phi) \end{bmatrix}$$

and

$$\frac{\partial \overrightarrow{q}}{\partial \theta} = \overrightarrow{y_{\mathrm{s}}} \cos\theta - \overrightarrow{x_{\mathrm{s}}} \sin\theta. \tag{B.33}$$

Substituting Equations B.25, B.26, B.27, B.28, B.29, B.30, B.31, B.32, and B.33 into Equation B.14 results in an expression for $\frac{\partial \epsilon_{\mathrm{r}}}{\partial \theta}$ that is easily evaluated and depends only upon known quantities. Specifically, the red channel of the raw image data (r_1 and r_2), the gradient of the red channel ($\frac{\partial r_1}{\partial x}$ and $\frac{\partial r_1}{\partial y}$), a scale parameter ($w$), the initial surfel ($\mathrm{P}_j$, $\overrightarrow{x_{\mathrm{s}}}$, and $\overrightarrow{y_{\mathrm{s}}}$), the best shifts ($\dot{u}$ and $\dot{v}$), and the camera parameters (f, x_0, y_0, $\overrightarrow{x_{\mathrm{c}_i}}$, $\overrightarrow{y_{\mathrm{c}_i}}$, $\overrightarrow{z_{\mathrm{c}_i}}$, and C_i).

The derivation for $\frac{\partial}{\partial \phi}$ is nearly identical. $\frac{\partial \mathcal{R}(\theta,\phi)}{\partial \phi}$ is the only significant difference,

$$\frac{\partial \mathcal{R}(\theta, \phi)}{\partial \phi} = \tag{B.34}$$
$$\begin{bmatrix} (1 - q_{\mathrm{x}}^2)\sin\phi & q_{\mathrm{x}} q_{\mathrm{y}} \sin\phi + q_{\mathrm{z}} \cos\phi & q_{\mathrm{x}} q_{\mathrm{z}} \sin\phi - q_{\mathrm{y}} \cos\phi \\ q_{\mathrm{x}} q_{\mathrm{y}} \sin\phi - q_{\mathrm{z}} \cos\phi & (1 - q_{\mathrm{y}}^2)\sin\phi & q_{\mathrm{y}} q_{\mathrm{z}}(1 - \sin\phi) + q_{\mathrm{x}} \cos\phi \\ q_{\mathrm{x}} q_{\mathrm{z}}(1 - \sin\phi) + q_{\mathrm{y}} \cos\phi & q_{\mathrm{y}} q_{\mathrm{z}}(1 - \sin\phi) - q_{\mathrm{x}} \cos\phi & (1 - q_{\mathrm{z}}^2)\sin\phi \end{bmatrix}.$$

Similar results apply for the green and blue channels.

Bibliography

[Adelson and Weiss, 1996] E. H. Adelson and Y. Weiss. A unified mixture framework for motion segmentation: Incorporating spatial coherence and estimating the number of models. In *Computer Vision and Pattern Recognition (CVPR '96 Proceedings)*, pages 321–326, June 1996. San Francisco, CA.

[Ayer and Sawhney, 1995] S. Ayer and H. Sawhney. Layered representation of motion video using robust maximum-likelihood estimation of mixture models and mdl encoding. In *International Conference on Computer Vision (ICCV '95 Proceedings)*, pages 777–784, June 1995. Cambridge, MA.

[Azarbayejani and Pentland, 1995] A. Azarbayejani and A. Pentland. Recursive estimation of motion, structure, and focal length. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(6):562–575, June 1995.

[Baker and Bolles, 1989] H. Harlyn Baker and Robert C. Bolles. Generalizing epipolar-plane image analysis on the spatiotemporal surface. *International Journal of Computer Vision*, 3(1):33–49, May 1989.

[Becker and Bove, 1995] Shawn Becker and V. Michael Bove, Jr. Semiautomatic 3-D model extraction from uncalibrated 2-D camera views. In *Visual Data Exploration and Analysis*, volume 2410, pages 447–461. SPIE, February 1995. San Jose, CA.

[Belhumeur and Mumford, 1992] Peter N. Belhumeur and David Mumford. A bayesian treatment of the stereo correspondence problem using half-occluded regions. In *Computer Vision and Pattern Recognition (CVPR '92 Proceedings)*, pages 761–764, June 1992. Champaign, IL.

[Belhumeur, 1993] Peter N. Belhumeur. A binocular stereo algorithm for reconstruction sloping, creased, and broken surfaces in the presence of half-occlusion. In *International Conference on Computer Vision (ICCV '93 Proceedings)*, pages 534–539, May 1993. Berlin, Germany.

[Belhumeur, 1996] Peter N. Belhumeur. A bayesian approach to binocular stereopsis. *International Journal of Computer Vision*, 19(3):237–262, 1996.

[Bolles *et al.*, 1987] Robert C. Bolles, H. Harlyn Baker, and David H. Marimont. Epipolar-plane image analysis: An approach to determining structure from motion. *International Journal of Computer Vision*, 1(1):7–55, 1987.

[Chou and Teller, 1998] G. T. Chou and S. Teller. Multi-level 3d reconstruction with visibility constraints. In *Image Understanding Workshop (IUW '98 Proceedings)*, volume 2, pages 543–550, November 1998. Monterey, CA.

[Collins *et al.*, 1995] R. Collins, C. Jaynes, F. Stolle, X. Wang, Y. Cheng, A. Hanson, and E. Riseman. A system for automated site model acquisition. In *Integrating Photogrammetric Techniques with Scene Analysis and Machine Vision II*, volume 7617. SPIE, April 1995. Orlando, FL.

[Collins, 1996] Robert T. Collins. A space-sweep approach to true multi-image matching. In *Computer Vision and Pattern Recognition (CVPR '96 Proceedings)*, pages 358–363, June 1996. San Francisco, CA.

[Coorg *et al.*, 1998] Satyan Coorg, Neel Master, and Seth Teller. Acquisition of a large pose-mosaic dataset. In *Computer Vision and Pattern Recognition (CVPR '98 Proceedings)*, pages 872–878, June 1998. Santa Barbara, CA.

[Coorg, 1998] Satyan Coorg. *Pose Imagery and Automated 3-D Modeling of Urban Environments*. PhD thesis, MIT, 1998.

[Cox *et al.*, 1996] Ingemar J. Cox, Sunita L. Hingorani, Satish B. Rao, and Bruce M. Maggs. A maximum likelihood stereo algorithm. *Computer Vision and Image Understanding*, 63(3):542–567, May 1996.

[Cox, 1994] Ingemar J. Cox. A maximum likelihood n-camera stereo algorithm. In *Computer Vision and Pattern Recognition (CVPR '94 Proceedings)*, pages 733–739, June 1994. Seattle, WA.

[Cutler, 1999] Barbara M. Cutler. Aggregating building fragments generated from geo-referenced imagery into urban models. Master's thesis, MIT, 1999.

[Dana *et al.*, 1996] Kristin J. Dana, Bram van Ginneken, Shree K. Nayar, and Jan J. Koenderink. Reflectance and texture of real-world surfaces. Technical Report CUCS-048-96, Columbia University, December 1996.

[De Couto, 1998] Douglas S. J. De Couto. Instrumentation for rapidly acquiring pose-imagery. Master's thesis, MIT, 1998.

[Debevec *et al.*, 1996] Paul E. Debevec, Camillo J. Taylor, and Jitendra Malik. Modeling and rendering architecture from photographs: A hybrid geometry- and image-based approach. In *Computer Graphics (SIGGRAPH '96 Proceedings)*, pages 11–20, August 1996.

[Duda and Hart, 1973] Richard O. Duda and Peter E. Hart. *Pattern Classification and Scene Analysis*. John Wiley & Sons, New York, NY, 1973.

[Faugeras and Robert, 1994] Olivier Faugeras and Luc Robert. What can two images tell us about a third one? In *European Conference on Computer Vision (ECCV '94 Proceedings)*, pages 485–492, May 1994. Stockholm, Sweden.

[Faugeras, 1992] Olivier Faugeras. What can be seen in three dimensions with an uncalibrated stereo rig? In *European Conference on Computer Vision (ECCV '92 Proceedings)*, pages 563–578, May 1992. Santa Margherita Ligure, Italy.

[Faugeras, 1993] Olivier Faugeras. *Three-Dimensional Computer Vision*. MIT Press, Cambridge, MA, 1993.

[Fua and Leclerc, 1995] P. Fua and Y. G. Leclerc. Object-centered surface reconstruction: Combining multi-image stereo and shading. *International Journal of Computer Vision*, 16(1):35–56, September 1995.

[Fua and Leclerc, 1996] P. Fua and Y.G. Leclerc. Taking advantage of image-based and geometry-based constraints to recover 3-D surfaces. *Computer Vision and Image Understanding*, 64(1):111–127, July 1996.

[Fua, 1995] P. Fua. Reconstructing complex surfaces from multiple stereo views. In *International Conference on Computer Vision (ICCV '95 Proceedings)*, pages 1078–1085, June 1995. Cambridge, MA.

[Gering and Wells, 1999] David T. Gering and William M. Wells, III. Object modeling using tomography and photography. In *IEEE Workshop on Multi-view Modeling and Analysis of Visual Scenes*, pages 11–18, June 1999. Fort Collins, CO.

[Greeve, 1996] C. W. Greeve, editor. *Digital Photogrammetry: an Addendum to the Manual of Photogrammetry*. American Society of Photogrammetry and Remote Sensing, Falls Church, VA, 1996.

[Hartley *et al.*, 1992] Richard Hartley, Rajiv Gupta, and Tom Chang. Stereo from uncalibrated cameras. In *Computer Vision and Pattern Recognition (CVPR '92 Proceedings)*, pages 761–764, June 1992. Champaign, IL.

[Hartley, 1992] Richard Hartley. Estimation of relative camera positions for uncalibrated cameras. In *European Conference on Computer Vision (ECCV '92 Proceedings)*, pages 579–587, May 1992. Santa Margherita Ligure, Italy.

[Horn, 1986] Berthold Klaus Paul Horn. *Robot Vision*. MIT Press, Cambridge, MA, 1986.

[Horn, 1987] Berthold Klaus Paul Horn. Closed-form solution of absolute orientation using unit quaternions. *Journal of the Optical Society of America*, 4(4):629–642, April 1987.

[Irani *et al.*, 1997] M. Irani, B. Rousso, and S. Peleg. Recovery of ego-motion using region alignment. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(3):268–272, March 1997.

[Kanade and Okutomi, 1994] Takeo Kanade and Masatoshi Okutomi. A stereo matching algorithm with an adaptive window: Theory and experiment. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(9):920–932, September 1994.

[Kang and Szeliski, 1996] Sing Bing Kang and Richard Szeliski. 3-D scene recovery using omnidirectional multibaseline stereo. In *Computer Vision and Pattern Recognition (CVPR '96 Proceedings)*, pages 364–370, June 1996. San Francisco, CA.

[Kang *et al.*, 1995] Sing Bing Kang, Jon A. Webb, C. Lawrence Zitnick, and Takeo Kanade. An active multibaseline stereo system with active illumination and realtime image acquisition. In *International Conference on Computer Vision (ICCV '95 Proceedings)*, pages 88–93, June 1995. Cambridge, MA.

[Kutulakos and Seitz, 1998a] Kiriakos N. Kutulakos and Steven M. Seitz. A theory of shape by space carving. CS Technical Report 692, University of Rochester, May 1998.

[Kutulakos and Seitz, 1998b] Kiriakos N. Kutulakos and Steven M. Seitz. What do n photographs tell us about 3D shape? CS Technical Report 680, University of Rochester, January 1998.

[Lenz and Tsai, 1988] Reimar K. Lenz and Roger Y. Tsai. Techniques for calibration of the scale factor and image center for high accuracy 3-D machine vision metrology. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(5):713–720, September 1988.

[Li and Chen, 1995] Xiaoping Li and Tongwen Chen. Optimal $\mathcal{L}_1$ approximation of the gaussian kernel with application to scale-space construction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(10):1015–1019, October 1995.

[Longuet-Higgins, 1981] H. C. Longuet-Higgins. A computer algorithm for reconstructing a scene from two projections. *Nature*, 293:133–135, September 1981.

[Marr and Poggio, 1979] D. Marr and T. Poggio. A computational theory of human stereo vision. *Proceedings of the Royal Society of London*, B(204):301–328, 1979.

[Mayhew and Frisby, 1991] John E. W. Mayhew and John P. Frisby, editors. *3D Model Recognition from Stereoscopic Cues*. MIT Press, Cambridge, MA, 1991.

[Mohr and Arbogast, 1991] R. Mohr and E. Arbogast. It can be done without camera calibration. *Pattern Recognition Letters*, 12(1):39–43, January 1991.

[Mohr *et al.*, 1993] R. Mohr, F. Veillon, and L. Quan. Relative 3D reconstruction using multiple uncalibrated images. In *Computer Vision and Pattern Recognition (CVPR '93 Proceedings)*, pages 543–548, June 1993. New York, NY.

[Nayar *et al.*, 1997] Shree K. Nayar, Xi-Sheng Fang, and Terrance Boult. Separation of reflection components using color and polarization. *International Journal of Computer Vision*, 21(3):163–186, February 1997.

[Ohta and Kanade, 1985] Y. Ohta and T. Kanade. Stereo by two-level dynamc programming. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 7(2):139–154, April 1985.

[Okutomi and Kanade, 1993] Masatoshi Okutomi and Takeo Kanade. A multiple-baseline stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(4):353–363, April 1993.

[Oren and Nayar, 1995] Michael Oren and Shree K. Nayar. Generalization of the lambertian model and implications for machine vision. *International Journal of Computer Vision*, 14(3):227–251, April 1995.

[Pollard *et al.*, 1985] S. B. Pollard, J. E. W. Mayhew, and J. P. Frisby. Pmf: A stereo correspondence algorithm using a disparity gradient constraint. *Perception*, 14:449–470, 1985.

[Press *et al.*, 1992] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, 1992.

[Ramachandran and Lakshminarayanan, 1971] G. N. Ramachandran and A. V. Lakshminarayanan. Three dimensional reconstruction from radiographs and electron micrographs: Applications of convolutions instead of fourier transforms. *Proceedings of the National Academy of Science*, 68:2236–2240, 1971.

[Seales and Faugeras, 1995] W. Brent Seales and Olivier D. Faugeras. Building three-dimensional object models from image sequences. *Computer Vision and Image Understanding*, 61(3):308–324, May 1995.

[Seitz and Dyer, 1997] Steven M. Seitz and Charles R. Dyer. Photorealistic scene reconstruction by voxel coloring. In *Computer Vision and Pattern Recognition (CVPR '97 Proceedings)*, pages 1067–1073, 1997. Puerto Rico.

[Shum *et al.*, 1998] Heung-Yeung Shum, Mei Han, and Richard Szeliski. Interactive construction of 3D models from panoramic mosaics. In *Computer Vision and Pattern Recognition (CVPR '98 Proceedings)*, pages 427–433, June 1998. Santa Barbara, CA.

[Slama *et al.*, 1980] Chester C. Slama, Charles Theurer, and Soren W. Henriksen, editors. *Manual of Photogrammetry*. American Society of Photogrammetry and Remote Sensing, Falls Church, VA, 4th edition, 1980.

[Szeliski and Kang, 1994] Richard Szeliski and Sing Bing Kang. Recovering 3D shape and motion from image streams using non-linear least squares. *Journal of Visual Communications and Image Representation*, 5(1):10–28, 1994.

[Szeliski and Shum, 1997] Richard Szeliski and Harry Shum. Creating full-view panoramic mosaics and texture-mapped 3D models. In *Computer Graphics (SIGGRAPH '97 Proceedings)*, pages 251–258, August 1997.

[Szeliski and Tonnesen, 1992] Richard Szeliski and David Tonnesen. Surface modeling with oriented particle systems. In *Computer Graphics (SIGGRAPH '92 Proceedings)*, volume 26, pages 185–194, July 1992.

[Szeliski, 1996] Richard Szeliski. Video mosaics for virtual environments. *IEEE Computer Graphics and Applications*, 16(2):22–30, March 1996.

[Tagare and deFigueiredo, 1993] Hemant D. Tagare and Rui J. P. deFigueiredo. A framework for the construction of reflectance maps for machine vision. *Computer Vision, Graphics And Image Processing: Image Understanding*, 57(3):265–282, May 1993.

[Teller, 1998a] Seth Teller. Automated urban model acquisition: Project rationale and status. In *Image Understanding Workshop (IUW '98 Proceedings)*, volume 2, pages 455–462, November 1998. Monterey, CA.

[Teller, 1998b] Seth Teller. Toward urban model acquisition from geo-located images. In *Pacific Graphics '98*, 1998. Singapore.

[Tomasi and Kanade, 1992] Carlo Tomasi and Takeo Kanade. Shape and motion from image streams under orthography: A factorization method. *International Journal of Computer Vision*, 9(2):137–154, November 1992.

[Tsai, 1983] Roger Y. Tsai. Multiframe image point matching and 3-D surface reconstruction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 5(2):159–174, March 1983.

[Tsai, 1987] Roger Y. Tsai. A versatile camera calibration technique for high-accuracy three dimensional machine vision metrology using off-the-shelf TV cameras and lenses. *IEEE Journal of Robotics and Automation*, RA-3(4):323–344, August 1987.

[Ullman, 1978] Shimon Ullman. The interpretation of structure from motion. A.I. Memo 476, MIT, October 1978.

[Ullman, 1979] Shimon Ullman. The interpretation of structure from motion. *Proceedings of the Royal Society of London*, B(203):405–426, 1979.

[Vexcel, 1997] Vexcel. The fotog family of products and services, 1997. Available at http://www.vexcel.com/fotog/product.html.

[Vora *et al.*, 1997a] Poorvi L. Vora, Joyce E. Farrell, Jerome D. Tietz, and David H. Brainard. Digital color cameras - response models. Technical Report HPL-97-53, Hewlett-Packard Laboratories, March 1997.

[Vora *et al.*, 1997b] Poorvi L. Vora, Joyce E. Farrell, Jerome D. Tietz, and David H. Brainard. Digital color cameras - spectral response. Technical Report HPL-97-54, Hewlett-Packard Laboratories, March 1997.

[Watkins, 1991] D. Watkins. *Fundamentals of Matrix Computations*. John Wiley and Sons, Inc., 1991.

[Wolf, 1974] Paul R. Wolf. *Elements of Photogrammetry*. McGraw-Hill, New York, NY, 1974.

[Wolff and Andreou, 1995] Lawrence B. Wolff and Andreas G. Andreou. Polarization camera sensors. *Image and Vision Computing*, 13(6):497–510, August 1995.

[Wolff, 1997] Lawrence B. Wolff. Polarization vision: A new sensory approach to image understanding. *Image and Vision Computing*, 15(2):81–93, February 1997.

[Yachida, 1986] M. Yachida. 3D data acquisition by multiple views. In O. D. Faugeras and G. Giralt, editors, *Robotics Research: the Third International Symposium*, pages 11–18. MIT Press, Cambridge, MA, 1986.