Geometry and Texture from Thousands of Images

J.P. Mellor (j.p.mellor@rose-hulman.edu)
Computer Science Department, Box 120,
Rose-Hulman Institute of Technology,
Terre Haute, IN 47803, USA

5 August 2000

Abstract. This paper 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. Some of the most important characteristics of our 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.

Keywords: APGD, 3D Reconstruction, Multi-Camera Stereo, Large-Scale Environments, Textured Model, Architectural Facades, Urban Scenes, Approximate Camera Calibration, non-Lambertian surfaces.

1. Introduction

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) require manual identification of correspondences, while computer vision approaches (Section 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.

1.1. Photogrammetry

A number of the recent interactive modeling systems are based upon photogrammetry[1]. 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.

1.2. Computer Vision

One approach to automatically solving the correspondence problem is to use multiple images. Collins (Collins, 1996) proposed a *space-sweep* method which matches multiple images in three-dimensional. 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. The 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. The carving operation, however, 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.

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 here 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.

Our approach 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). The also make 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 camera's internal parameters are also known.
- 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 textured surface patch or *surface element* is referred to as a surfel. 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.
- 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[2].

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.

1.4. City Scanning Project

The work presented in this paper is part of the MIT City Scanning project whose primary focus is the Automatic Population of Geospatial Databases (APGD). A self contained image acquisition platform called *Argus* is used to acquire calibrated images (De Couto, 1998). At each location or node images are acquired in a hemispherical tiling. The position and orientation estimates obtained during the acquisition phase are good, but contain a significant amount of error. The estimates are refined using techniques described in (Coorg et al., 1998; Coorg, 1998). For a more complete description of the project see (Teller, 1998a; Teller, 1998b).

1.5. Overview

Figure 1 shows an overview of the reconstruction pipeline described in this paper. The calibrated imagery described above 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. Section 2 presents our approach. Section 3

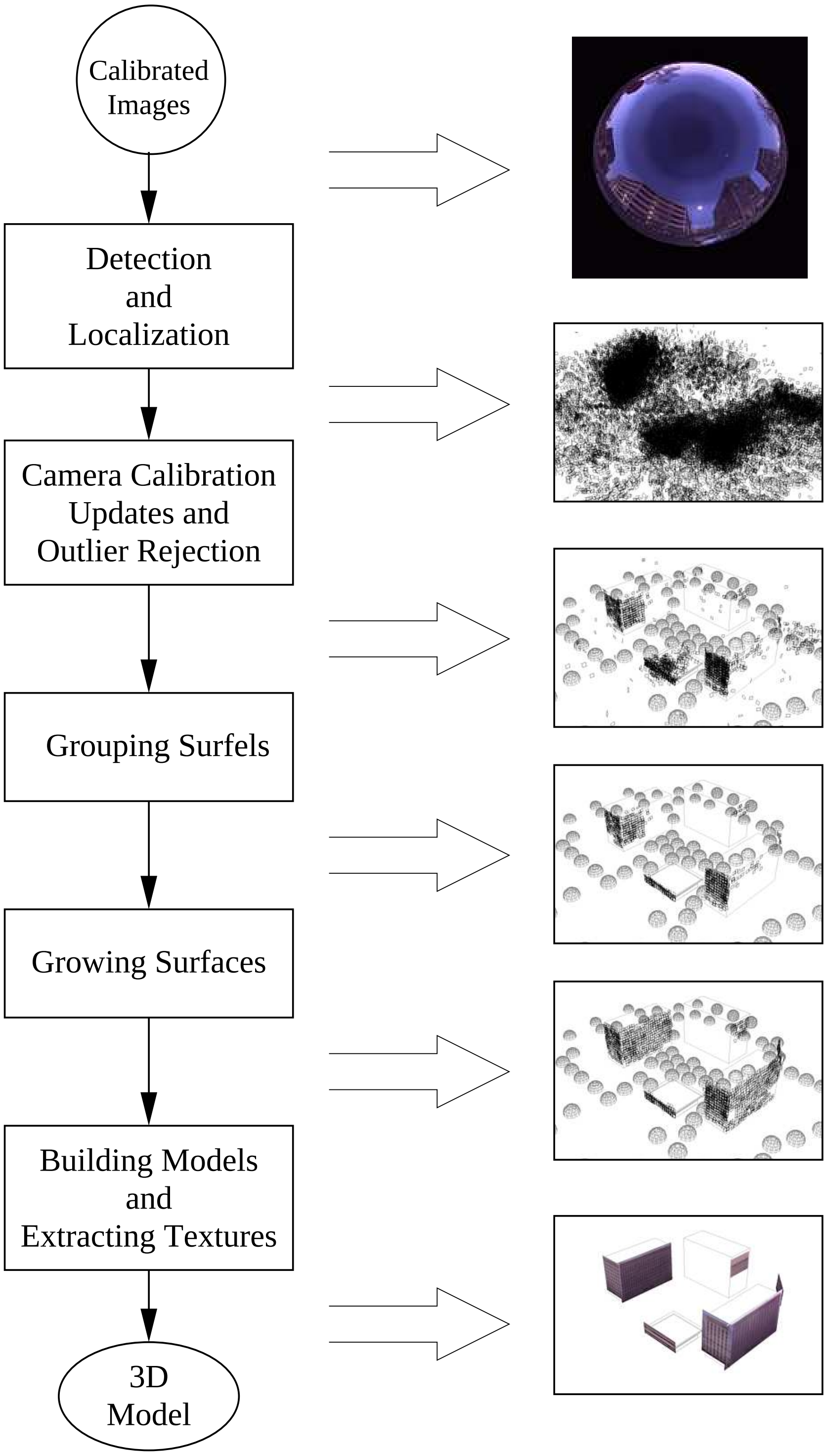


Figure 1. Overview.

describes the dataset used for this paper. Section 4 explores the difficulties associated with camera calibration error, variations in illumination and reflectance, occlusion, and image noise. Section 5 examines detecting and localizing surfels. Section 6 introduces several techniques to remove false positives and fill in the missing parts. Finally we discuss fitting simple models and extracting textures. We present the results of applying our method to a large dataset.

2. Our Approach

Ideally, if a hypothesized surfel is coincident with an actual surface, reprojecting images onto the surfel should produce a set of regions which are highly correlated. On the other hand, if the surfel is not coincident with a surface, the subimages should not be correlated. This is the basic idea behind our approach (Mellor et al., 1997). For perfect calibration, constant illumination, Lambertian (i.e. diffuse) surfaces, and no complex occlusion a simple function[3] such as

$$\nu(\mathrm{S}_j) = \frac{\sum\limits_{i \in \mathcal{Q}} w_i \mathcal{X}(\mathcal{F}(\mathrm{p}_j^i), \mathcal{F}(\mathrm{p}_j^\star))}{\sum\limits_{i \in \mathcal{Q}} w_i} \tag{1}$$

where

$$\mathcal{Q} = \left\{ i \;\middle|\; \begin{array}{l} \mathrm{p}_j^i \in \Pi_\mathrm{i}^i \\ w_i > 0 \end{array} \right\},$$

$$w_i = \overrightarrow{\mathrm{P}_j \mathrm{C}_i} \cdot \mathrm{n}_j,$$

$$\mathcal{F}(x) = \mathrm{rgb}(x) = [\mathrm{r}(x), \mathrm{g}(x), \mathrm{b}(x)]\,, \tag{2}$$

and

$$\mathcal{X}([r_1, g_1, b_1]\,, [r_2, g_2, b_2]) = \tag{3}$$

Table I. Notation.

S_j	The j^{th} surfel.
P_j	Absolute coordinates of S_j.
n_j	Orientation of S_j.
${}_x^y\mathrm{S}_j$	Point in S_j. x and y are indices into S_j. ${}_0^0\mathrm{S}_j \equiv \mathrm{P}_j$.
C_i	Position of the i^{th} camera.
$\widetilde{\mathrm{C}}_i$	Estimated position.
Π_i^i	The i^{th} camera's image plane.
$\widetilde{\Pi}_\mathrm{i}^i$	Estimated image plane.
I^i	Calibrated image. $\mathrm{I}^i = \langle \Pi_\mathrm{i}^i, \mathrm{C}_i \rangle$.
$\tilde{\mathrm{I}}^i$	Calibrated (estimated) image. $\tilde{\mathrm{I}}^i = \langle \widetilde{\Pi}_\mathrm{i}^i, \widetilde{\mathrm{C}}_i \rangle$.
δ^i	Bounded camera calibration error for I^i
p_j^i	Projection of P_j onto Π_i^i.
$\tilde{\mathrm{p}}_j^i$	Projection of P_j onto $\widetilde{\Pi}_\mathrm{i}^i$.
$\tilde{\tilde{\mathrm{p}}}_j^i$	Noisy projection of P_j. Area on $\widetilde{\Pi}_\mathrm{i}^i$ given δ^i.
s_j^i	Projection of S_j onto $\widetilde{\Pi}_\mathrm{i}^i$.
${}_u^v\tilde{\mathrm{p}}_j^i$	Point in $\tilde{\tilde{\mathrm{p}}}_j^i$. u and v are indices into $\tilde{\tilde{\mathrm{p}}}_j^i$.
${}_x^y\mathrm{s}_j^i$	Point in s_j^i. x and y are indices into s_j^i. ${}_0^0\mathrm{s}_j^i \equiv \tilde{\mathrm{p}}_j^i$.
$\dot{\mathrm{p}}_j^i$	${}_{\dot{u}_i}^{\dot{v}_i}\tilde{\mathrm{p}}_j^i$ where $\dot{u}_i$ and $\dot{v}_i$ produce the best match.
$\dot{\mathrm{s}}_j^i$	Surfel that is the best match, i.e. ${}_0^0\mathrm{s}_j^i \equiv \dot{\mathrm{p}}_j^i$.
$\dot{\mathrm{P}}_j$	Reconstructed world point. Estimated using $\left\{\dot{\mathrm{p}}_j^i\right\}$.
$\mathcal{T}\left(\mathrm{P}, \mathrm{I}\right)$	Function projecting P onto I, e.g. $\mathcal{T}\left(\mathrm{P}_j, \mathrm{I}^i\right) = \mathrm{p}_j^i$.
$\widetilde{\mathcal{T}}\left(\mathrm{P}, \mathrm{I}, \delta\right)$	Noisy projection function, e.g. $\widetilde{\mathcal{T}}\left(\mathrm{P}_j, \tilde{\mathrm{I}}^i, \delta^i\right) = \tilde{\tilde{\mathrm{p}}}_j^i$. Note: $\mathcal{T}\left(\mathrm{P}_j, \tilde{\mathrm{I}}^i\right) = \tilde{\mathrm{p}}_j^i$.
$\mathcal{F}\left(x\right)$	Function of image at point x (e.g. pixel values, features).
$\mathcal{X}\left(x_1, x_2\right)$	Matching function. Quality of match between x_1 and x_2.
$\nu\left(\mathrm{S}_j\right)$	Match quality.
$\boldsymbol{X}$	Set of all X's, $\left\{X^i\right\}$.
$\{E \mid C\}$	Set of E's such that C is true.
$\overrightarrow{\mathrm{P}_1\mathrm{P}_2}$	Unit vector from P_1 to P_2.

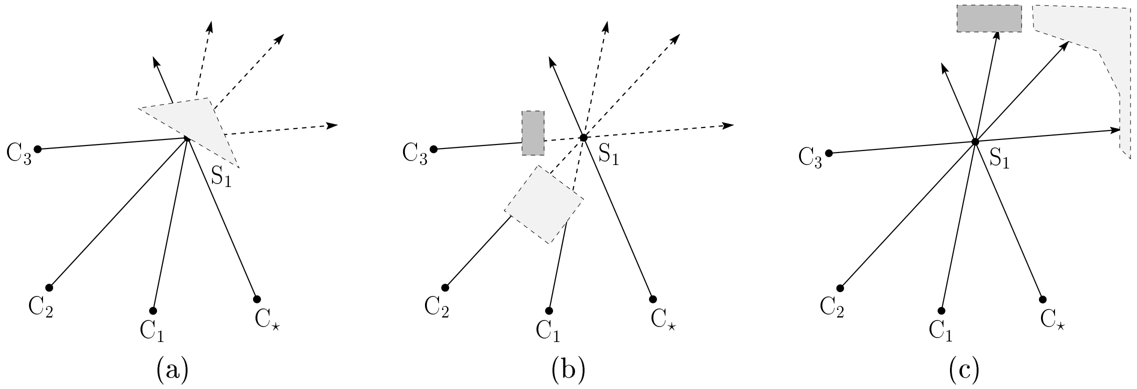


Figure 2. (a) Surfel S_j coincident with a physical surface; (b) Occlusion between C_i and S_j; (c) Inconsistent background.

$$- \left(\frac{(r_1 - r_2)^2}{\sigma_{\mathrm{r}}^2} + \frac{(g_1 - g_2)^2}{\sigma_{\mathrm{g}}^2} + \frac{(b_1 - b_2)^2}{\sigma_{\mathrm{b}}^2} \right)$$

can be used to distinguish situations similar to Figure 2(a) from those like Figures 2(b) and 2(c). $\mathrm{r}(x)$, $\mathrm{g}(x)$, 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[4].

As the number of elements in $\mathcal{Q}$ increases, the likelihood increases that $\nu(\mathrm{S}_j)$ will be large when S_j lies on a physical surface and small when it does not. Occlusions do not create peaks when S_j is not coincident with a surface (false positives)[5]. They can however, cause false negatives (the lack of a peak when S_j is coincident with a surface) as shown in Figure 2(b). A false negative is essentially a lack of evidence about an actual surface. 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.

This simple metric works quite well on synthetic data—93% of the reconstructible points were recovered and 90% of these were within 1% of the correct values. For a more complete discussion see (Mellor et al., 1997; Mellor, 1999).

3. 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 in Cambridge, MA (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 3. At each node, the camera was rotated about the focal point collecting images in a hemispherical mosaic. Most nodes are tiled with 47 images. The raw images are 1524× 1012 pixels and cover a field of view of 41° × 28°. Each node contains approxi-

Figure 3. Node locations.

mately 70 million pixels. After acquisition, the images are reduced to quarter resolution[6] and mosaiced (Coorg et al., 1998; Coorg, 1998). Equal area projections of the spherical mosaic from two nodes are shown in Figure 4. The top node was acquired on an overcast day and has a distinct reddish tint. The bottom one 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[7]. 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. This greatly increases the quantity and quality of available data, but because of variations in illumination condition also complicates the analysis.

Figures 5 and 6 show several images from our dataset reprojected[9] onto a surfel which is coincident with an actual surface. The location, orientation, and size[10] of the surfels used are shown in Table II. Surfel 1 was used to generate Figure 5 and surfel 2, Figure 6. If the camera calibration estimates were perfect, the illumination was constant, and the surfaces Lambertian, then the regions in each figure should be identical[11]. The misalignment present in both sets is the result of error in the calibration estimates. Figure 5 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

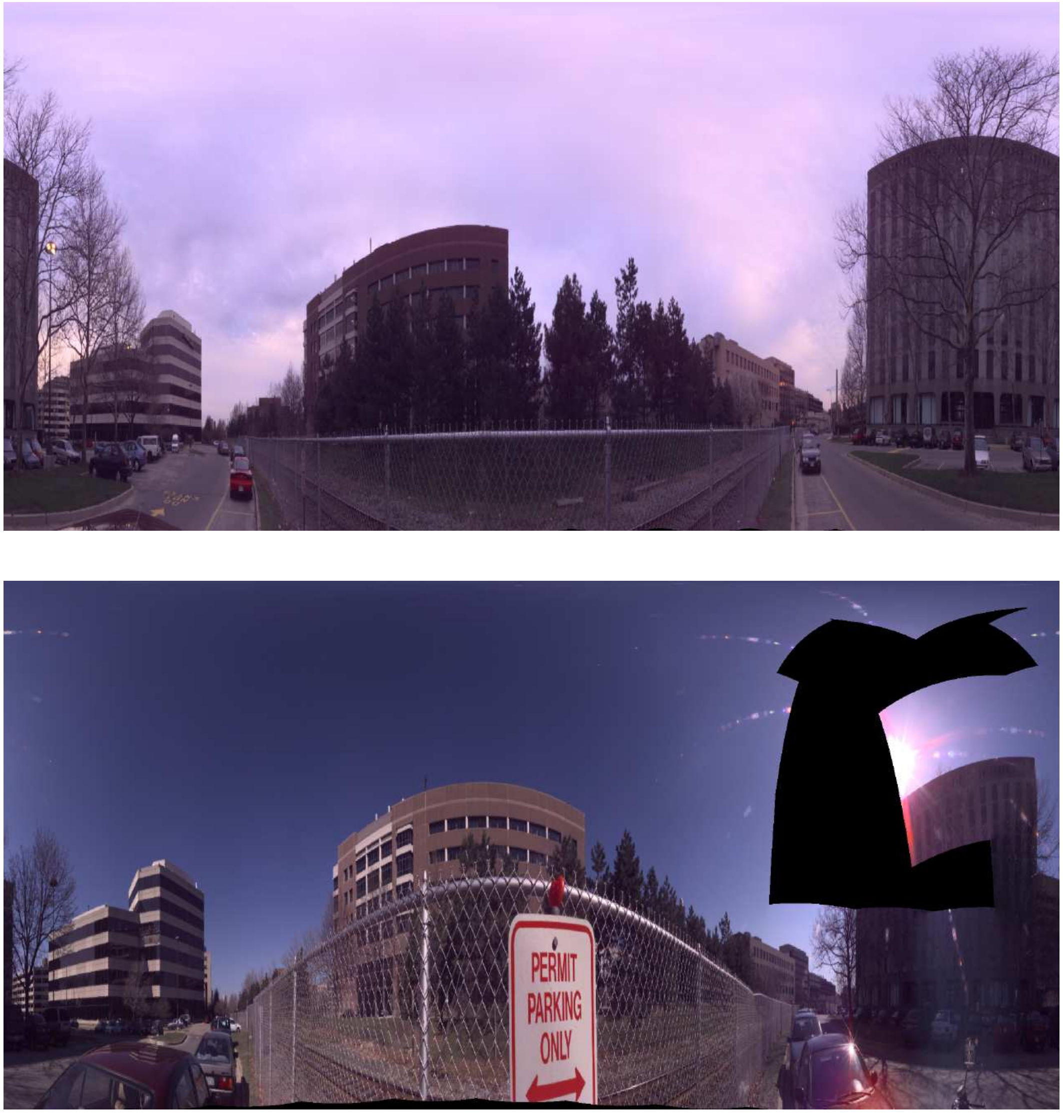


Figure 4. Example nodes.

Table II. Surfel parameters.

	Position			Normal			Size (units[9])		Size (pixels)	
Surfel	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

the last row were collected under direct sunlight. And, the last two images were taken near dusk. Figure 6 is more typical of the dataset. It is lower in contrast and some of the regions are partially occluded by trees. Figures 7 and 8 show source images with the reprojected area marked by a circle for several of the regions shown in Figures 5 and 6. In the worst cases all views of a surfel are similar to the upper left image in Figure 8.

4. The Challenge of Noisy Data

Noisy data presents several interesting challenges. We must extend our algorithm to handle camera calibration error, significant variation in illumination condition and reflectance, and image noise (e.g. soft occlusion from tree branches) (Mellor et al., 1998).

4.1. Camera Calibration Error

As shown in Figure 9, 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. Therefore, to compensate for camera calibration error we allow the source images to be shifted in the image plane prior to reprojection. Equation 1 becomes

$$\nu(\mathrm{S}_j) = \frac{\sum_{i \in \mathcal{Q}} w_i \max_{u,v} \mathcal{X}(\mathcal{F}({}^v_u\tilde{\mathrm{p}}^i_j), \mathcal{F}(\tilde{\mathrm{p}}^\star_j))}{\sum_{i \in \mathcal{Q}} w_i}. \tag{4}$$

In the ideal case, when 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 the *shifts* which correct for 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 sights through $\dot{\mathrm{p}}^i_j$,

$$\operatorname*{argmin}_{\dot{\mathrm{P}}_j} \sum_{\dot{\mathrm{p}}^i_j \in \dot{\mathbf{p}}_j} \overrightarrow{\widetilde{\mathrm{C}}_i \dot{\mathrm{p}}^i_j} \times \left((\dot{\mathrm{P}}_j - \widetilde{\mathrm{C}}_i) \times \overrightarrow{\widetilde{\mathrm{C}}_i \dot{\mathrm{p}}^i_j} \right) \cdot (\dot{\mathrm{P}}_j - \widetilde{\mathrm{C}}_i) \tag{5}$$

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

$$\operatorname*{argmin}_{\dot{\mathrm{P}}_j} \sum_{\dot{\mathrm{p}}^i_j \in \dot{\mathbf{p}}_j} |\dot{\mathrm{p}}^i_j - \mathcal{T}(\dot{\mathrm{P}}_j, \tilde{\mathrm{I}}^i)|^2. \tag{6}$$

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 Section 5.1. The additional degrees of freedom also admit additional false positives. Figure 10 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 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 in Section 6.

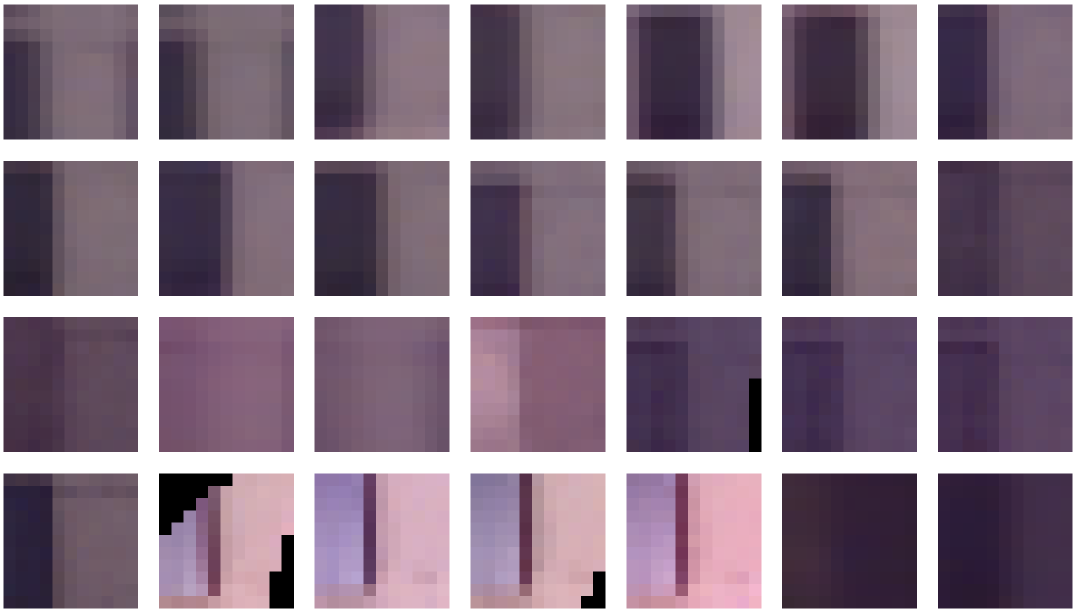


Figure 5. Reprojection onto surfel 1 coincident with actual surface.

Figure 6. Reprojection onto surfel 2 coincident with actual surface.

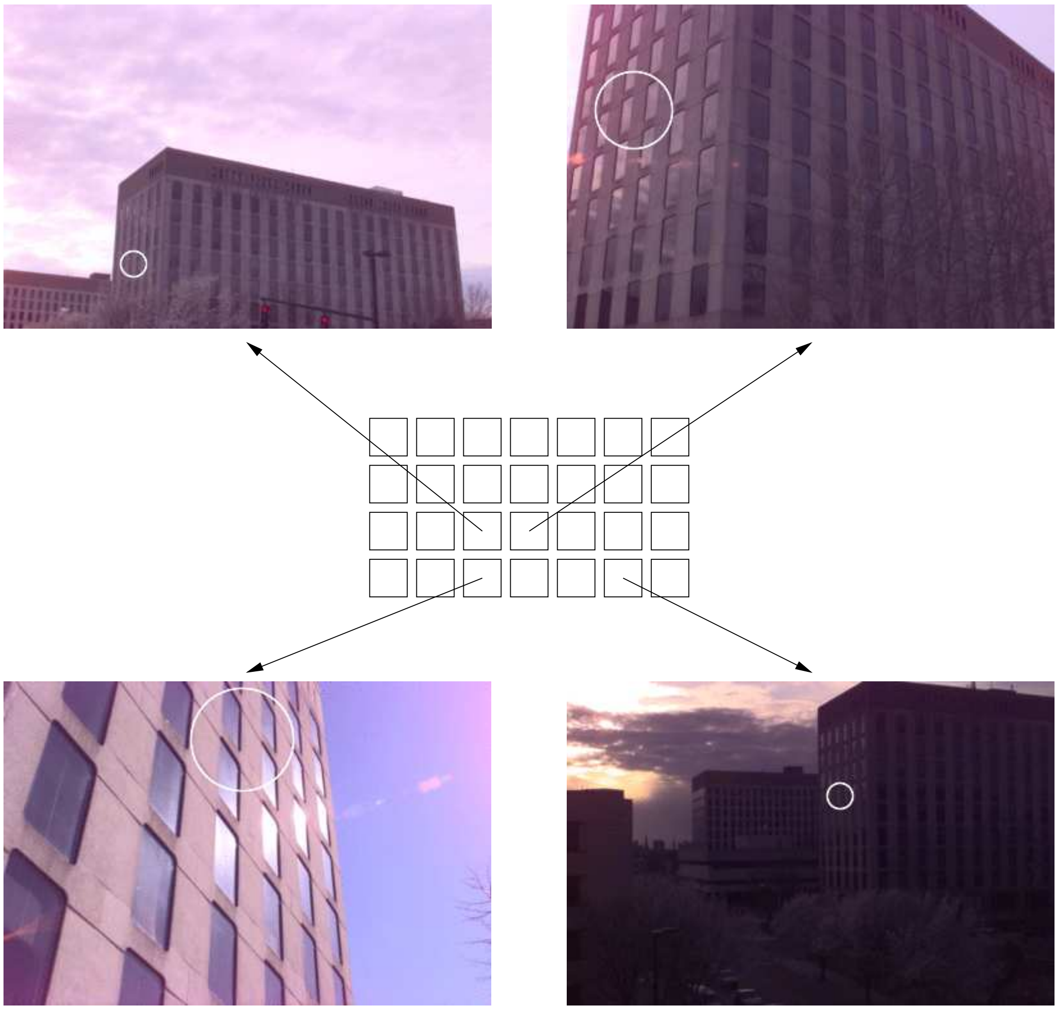


Figure 7. Source images for selected regions of surfel 1.

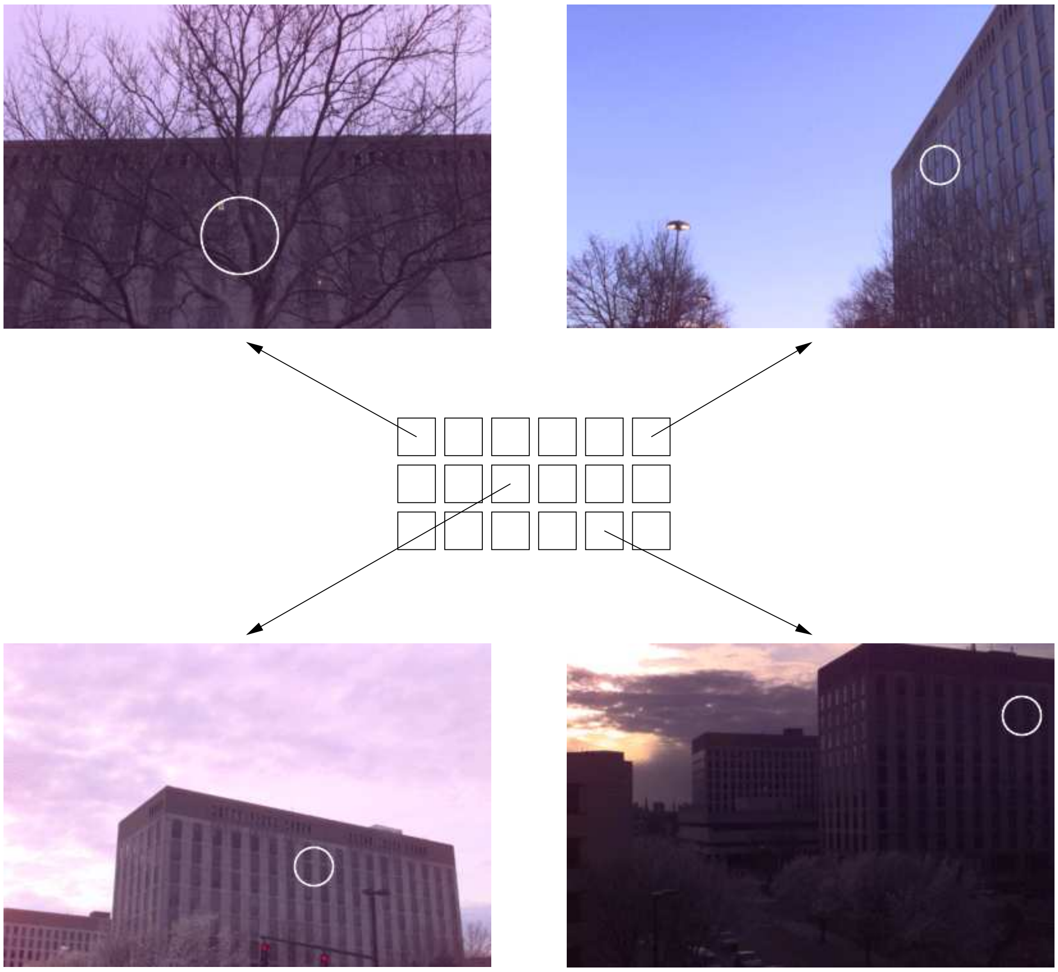


Figure 8. Source images for selected regions of surfel 2.

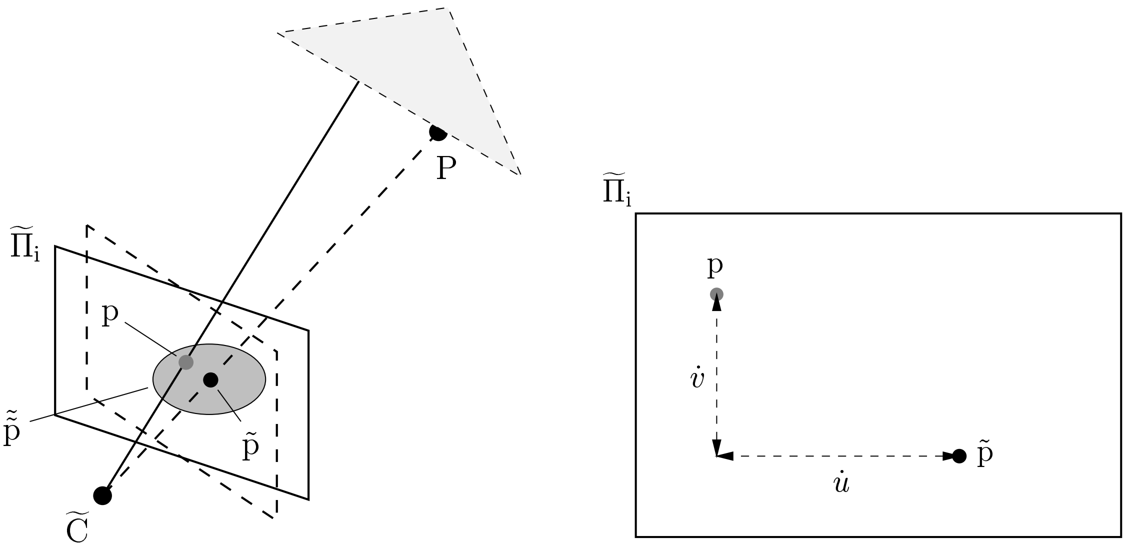


Figure 9. Effect of camera calibration error.

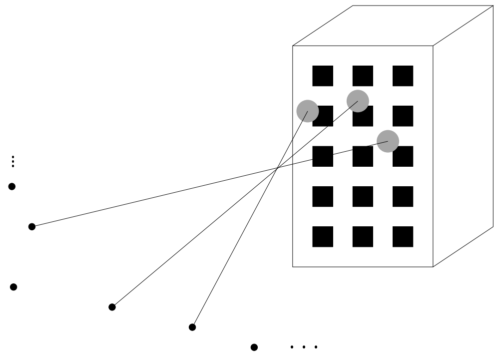


Figure 10. False positive induced by compensating for camera calibration error with shifts.

4.2. Variations in Illumination and Reflectance

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 diffuse objects. Both of these make it difficult to compare images of the same region taken under differing illumination conditions and from different viewpoints.

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:

$$R = kI. \tag{7}$$

and the value measured by the image sensor is:

$$v = \mathrm{f}(R) + d. \tag{8}$$

$\mathrm{f}(\cdot)$, an invertible function, and b, an offset, are functions of the image sensor[12]. Figure 11 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 7 and 8 it follows that if for some region

$$I_2 k_2 = I_1 k_1 c \tag{9}$$

and

$$\mathrm{f}(xy) = \mathrm{f}(x)\mathrm{f}(y) \tag{10}$$

then

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

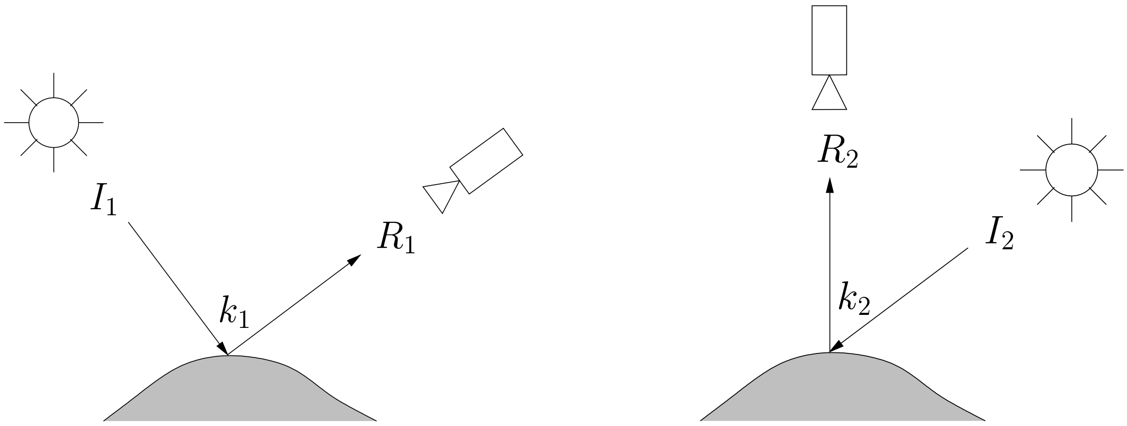


Figure 11. Region imaged under different conditions.

or

$$v_2 = m v_1 + d. \tag{11}$$

Equation 10 is true for typical CCD cameras. Equation 11 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 viewing condition to another and then match them. Substituting Equation 11 into Equation 3 gives[13]:

$$\begin{aligned}\mathcal{X}([r_1, g_1, b_1], [r_2, g_2, b_2]) = & \qquad (12)\\ -\Bigg(& \frac{((m_{\mathrm{r}} r_1 + d_{\mathrm{r}}) - r_2)^2}{\sigma_{\mathrm{r}}^2} + \\ & \frac{((m_{\mathrm{g}} g_1 + d_{\mathrm{g}}) - g_2)^2}{\sigma_{\mathrm{g}}^2} + \\ & \frac{((m_{\mathrm{b}} b_1 + d_{\mathrm{g}}) - b_2)^2}{\sigma_{\mathrm{b}}^2} \Bigg).\end{aligned}$$

The condition given by Equation 9 requires that the product of irradiance and reflectance vary similarly for changes in viewing conditions at each point x in the region $\mathcal{R}$:

$$\forall x \in \mathcal{R} : I_2 r_2 \,/ I_1 r_1 = \mathrm{c}. \tag{13}$$

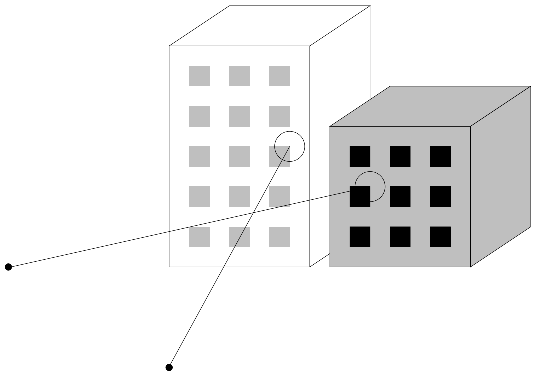


Figure 12. False positive induced by correcting for viewing conditions.

In practical terms, this means that $\mathcal{R}$ must be planar (or well approximated by a plane), uniformly lit, and contain materials with similar[14] 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. 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 10 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 $\nu(\mathrm{S}_j)$ as presented in Sections 2 and 4.1. This is no longer the case once we consider illumination. In order to calculate m and d at least 2 pixels are needed and for good results more should be used. The change can be made explicit by rewriting Equation 4 as

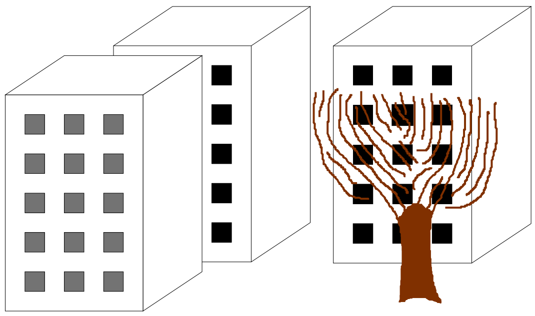
$$\nu(\mathrm{S}_j) = \frac{\displaystyle\sum_{i\in\mathcal{Q}} w_i \max_{u,v} \frac{\displaystyle\sum_{{}^{y}_{x}\mathrm{s}^i_j \in \mathrm{s}^i_j} \mathcal{X}\left(\mathcal{F}\left({}^{y+v}_{x+u}\mathrm{s}^i_j\right), \mathcal{F}\left({}^{y}_{x}\mathrm{s}^{\star}_j\right)\right)}{\displaystyle\sum_{{}^{y}_{x}\mathrm{s}^i_j \in \mathrm{s}^i_j} 1}}{\displaystyle\sum_{i\in\mathcal{Q}} w_i}. \tag{14}$$

In order to obtain a match, S_j must be close (both in position and orientation) to a physical surface and Equation 13 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 12 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 5.1.

There are some drawbacks to matching surfels instead of individual pixels. The cost of computing the match is higher. In addition, the shifts introduced in Section 4.1 vary with image position and the distance to the imaged point and therefore are not constant throughout the surfel. For small calibration errors and small regions it is reasonable to assume that the shifts are piecewise constant. 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 5.1 and 5.2).

4.3. Occlusion and Image Noise

Occlusion poses another difficult problem. Figure 13 shows two examples typical of urban environments. In Figure 13(a), 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. Figure 13(b) is potentially more difficult. The tree only partially occludes 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.



(a) (b)

Figure 13. (a) Hard and (b) soft occlusion.

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 13(b) are outliers. Other examples are transient objects (cars, people, etc), imaging aberrations (sensor noise, blooming), and

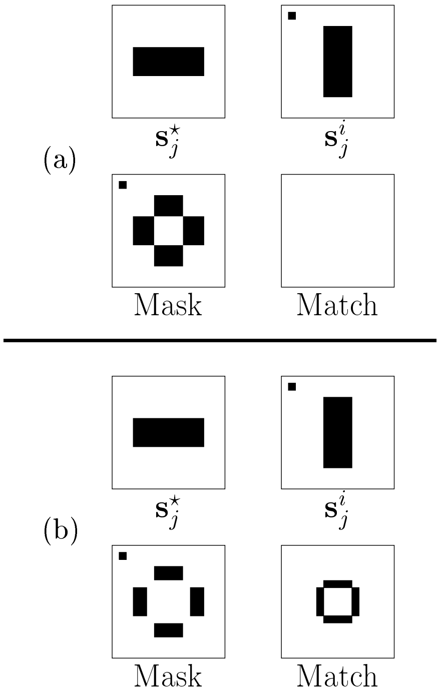


Figure 14. Masks.

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 in Figure 14(a), is to tag every pixel which matches poorly as an outlier. Black indicates pixels tagged as outliers and poor matches. Clearly the pair $(\mathrm{s}_j^\star,\ \mathrm{s}_j^i)$ 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 the shifts and illumination corrections already introduced. A more conservative approach is shown in Figure 14(b). 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 one 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 14 can be rewritten to account for both individual pixel and entire region outliers as follows:

$$\nu(\mathrm{S}_j) = \tag{15}$$
$$\frac{\displaystyle\sum_{i\in\mathcal{Q}} w_i \max_{u,v} \frac{\displaystyle\sum_{{}^y_x\mathrm{s}_j^i \in \mathrm{s}_j^i} \mathcal{M}^i({}^y_x\mathrm{S}_j)\mathcal{X}\left(\mathcal{F}\left({}^{y+v}_{x+u}\mathrm{s}_j^i\right), \mathcal{F}\left({}^y_x\mathrm{s}_j^\star\right)\right)}{\displaystyle\sum_{{}^y_x\mathrm{s}_j^i \in \mathrm{s}_j^i} \mathcal{M}^i({}^y_x\mathrm{S}_j)}}{\displaystyle\sum_{i\in\mathcal{Q}} w_i}$$

where

$$\mathcal{Q} = \left\{ i \;\middle|\; \begin{array}{l} \tilde{\mathrm{p}}_j^i \in \widetilde{\Pi}_{\mathrm{i}}^i \\ w_i > 0 \\ \mathrm{s}_j^i \neq \text{outlier} \\ \dfrac{\displaystyle\sum_{{}^y_x\mathrm{s}_j^i \in \mathrm{s}_j^i} \mathcal{M}^i({}^y_x\mathrm{S}_j \in \mathrm{S}_j)}{\displaystyle\sum_{{}^y_x\mathrm{s}_j^i \in \mathrm{s}_j^i} 1} \geq 0.5 \end{array} \right\}$$

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

Equations 4, 14, and 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 $|\mathcal{Q}|$ 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 $s_j^\star = s_j^i$. 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.

5. Images to Surfels

Now that we have several methods to compensate for noisy data, we 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.

5.1. Detecting Surfels

We use the following algorithm to detect surfels.

1. Hypothesize surfel S_j in world coordinates.
2. Select images $\tilde{\mathbf{I}}$.
3. Reproject regions $\mathbf{s}_j$.
4. Select next $s_j^\star$.
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.
6. Evaluate match set $\nu(S_j)$:
    – If good enough $\Rightarrow$ done.
    – If not $\Rightarrow$ goto 4.

Once a surfel (such as shown in Table II) 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 results 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 $\overrightarrow{\mathrm{P}_j\widetilde{\mathrm{C}}_i}$ 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 onto the surfel, producing a set of regions $\mathbf{s}_j$ similar to those shown in Figures 5 and 6. To facilitate choosing $s_j^\star$, we define a region's *interest* as

$$\text{interest} = \qquad (16)$$
$$w_i \frac{\displaystyle\sum_{u,v} \frac{\displaystyle\sum_{{}^y_x\mathrm{S}_j \in \mathrm{S}_j} \mathcal{X}\left(\mathcal{F}\left({}^{y+v}_{x+u}s_j^i\right), \mathcal{F}\left({}^{y}_{x}s_j^i\right)\right)}{\sqrt{u^2+v^2} \displaystyle\sum_{{}^y_x\mathrm{S}_j \in \mathrm{S}_j} 1}}{\displaystyle\sum_{u,v} 1}$$

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 5 with a score of 575.5. Region 2 is the most interesting in Figure 6 with a score[15] of 34.5.

Steps 5a, 5b, and 5c are actually accomplished simultaneously when finding the best match but for our discussion we will consider the steps separately. We find the best match between $\mathrm{s}_j^\star$ and s_j^i by evaluating:

$$\max_{u,v} \ \epsilon(u,v) \tag{17}$$

where

$$\epsilon(u,v) = \frac{\sum\limits_{{}^y_x\mathrm{S}_j \in \mathrm{S}_j} \mathcal{X}(\mathcal{F}({}^{y+v}_{x+u}\mathrm{s}_j^i), \mathcal{F}({}^y_x\mathrm{s}_j^\star))}{\sum\limits_{{}^y_x\mathrm{S}_j \in \mathrm{S}_j} 1}. \tag{18}$$

Figure 15 shows $-\epsilon$ versus u, x shift, and v, y shift, for two regions each from Figures 5 and 6. 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 5 is used as $\mathrm{s}_j^\star$ for the top two rows and region two from figure 6 is used for the bottom two rows. The plots are periodic because of repetitive texture (e.g. windows on the on the building). The different periodicity in the x and y directions primarily reflects the horizontal and vertical spacing of the windows. The ninth and last regions of Figure 5 were used as s_j^i for the first and second rows and the first and fourth regions of Figure 6 were used as s_j^i for the third and fourth rows.

The s_j^i's used in the second and fourth rows originate from cameras which imaged the surfel from a relatively oblique viewpoint. The s_j^i for the second row is a very low contrast image acquired at dusk (similar to the lower right image of Figure 7) and the s_j^i for the third row 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 17 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[16] 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 to ± 5. In many cases, only a single minimum exists within this limited range. The multiple minima, particularly those such as in the second row, can lead to false positives like the one illustrated in Figure 10.

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 $\mathrm{s}_j^\star$ as the y. Each color channel (r, g, and b) is computed separately. Changes in both illumination and reflectance are factored into equation 11, 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 $\mathrm{s}_j^\star$, 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.

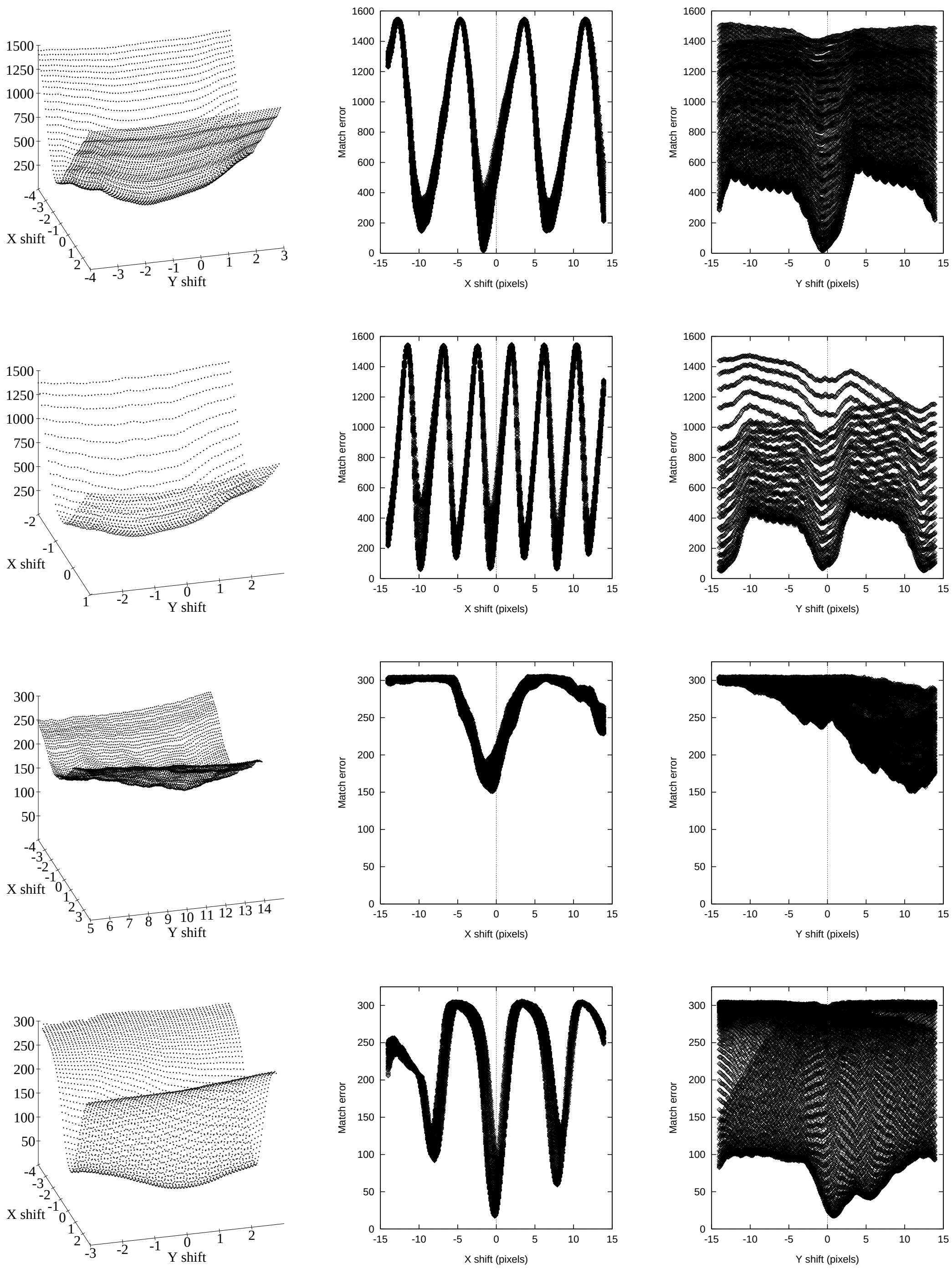


Figure 15. Shift error space near correct shift (left), projection onto the x-error plane (center), and y-error plane (right). The periodicity is caused by repetitive texture (windows) on the buildings. The change of periodicity in the x and y directions is produced by the window spacing in the horizontal and vertical directions. The change in periodicity from row to row is the result of foreshortening.

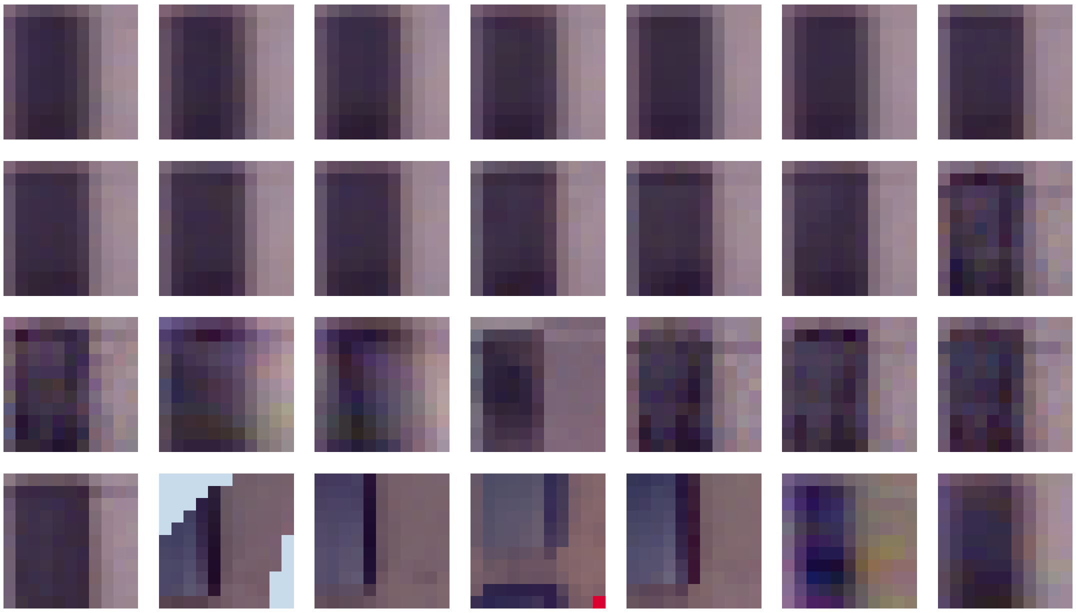


Figure 16. Aligned and corrected regions for surfel 1.

Figure 17. Aligned and corrected regions for surfel 2.

Table III. Match data for surfel 1.

Region	Match	Score	Unique	Shift		Slope			Intercept		
				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 IV. Match data for surfel 2.

Region	Match	Score	Unique	Shift x	Shift y	Slope r	Slope g	Slope b	Intercept r	Intercept g	Intercept 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

Figures 16 and 17 show the regions from Figures 5 and 6 with the shifts and color corrections applied[17]. 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 $\epsilon(\dot{u}, \dot{v})$. To help determine the significance of the match between s_j^i and $\mathrm{s}_j^\star$ 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 is defined to be

$$\text{uniqueness} = \frac{\sum_{u,v} \eta(u,v) \Big/ \sum_{u,v} 1}{\epsilon(\dot{u}, \dot{v})} \tag{19}$$

where

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

and

$$\eta(u,v) = \frac{\sum_{{}^y_x\mathrm{S}_j \in \mathrm{S}_j} \mathcal{X}(\mathcal{F}({}^{y+\dot{v}_i}_{x+\dot{u}_i}\mathrm{s}_j^i), \mathcal{F}({}^{y+v}_{x+u}\mathrm{s}_j^\star))}{\sqrt{u^2+v^2} \sum_{{}^y_x\mathrm{S}_j \in \mathrm{S}_j} 1}.$$

A region is considered a match if the shift and color correction are within limits, at least half of the points in s_j^i contribute to the match and the match score is good enough and unique enough. We require the match score to be ≥ -100 and the uniqueness[18] to be ≥ 2. Tables III and IV show the match data for the regions in Figures 16 and 17.

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 $s_j^\star$ has an interest which is greater than or equal to a minimum value. We require the number of regions to be ≥ 6, the number of nodes to be ≥ 5, $\nu(S_j) \geq -100$, and interest ≥ 50. The sets of regions shown in Figures 5 and 6 both produce valid matches.

To test the detection characteristics of our algorithm, we evaluated $\nu(S_j)$ for many surfels near an actual surface. A $100 \times 100 \times 200$ unit volume, the shaded area in Figure 18(a), was selected so that an actual surface divides the volume into two cubes. 396 positions chosen at 20 units intervals were used as test points. Each test point was evaluated every 5° for azimuths ±45° and every 5° for elevations ±45°. A total of 142,956 surfels were evaluated. Figure 18(b) shows the fraction of the nearly 13,000 surfels tested at each displacement that produced a valid match set. Figure 18(c) shows the fraction of detections versus displacement and the angle between the actual and tested orientation. Figure 18(d) shows the fraction of detections versus azimuth and elevation[19]. 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 19(a). An actual surface was selected and tessellated with surfels. The fraction of false negatives (1− the detection rate) for an unoccluded surface versus the fraction of true negatives (1− the detection rate) for the empty volume is shown in Figure 19(b). 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 19(b) is the result of noise and the discrete nature of surfels. Similar curves for an occluded (significant tree coverage) surface are shown in Figures 19(c) and 19(d). The data shown in Figure 19(c) does not utilize the mask function described in Section 4.3, while that shown in Figure 19(d) does. 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.

5.2. 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, $\mathbf{\dot{s}}_j$. 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 $\mathbf{\dot{s}}_j$ 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 $\nu(S_j)$.

Steps 1a, 1b, and 1c are actually interdependent and ideally should be calculated simultaneously. For ease of implementation and to speed convergence[20] we have separated them. We have obtained good results performing the steps individually, but plan to integrate them in the future.

We use Equation 5, which considers only matching regions, to update P_j.

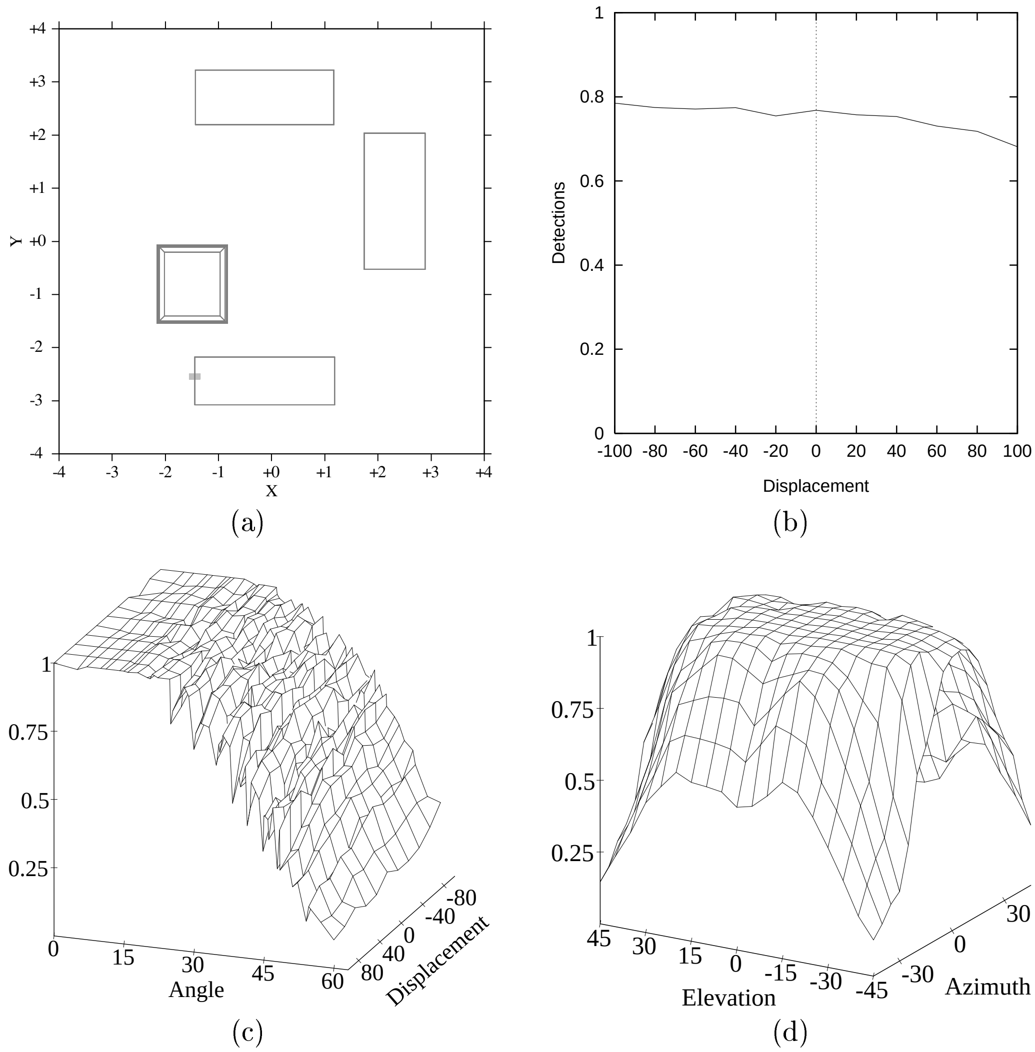


Figure 18. (a) Test volume (small shaded rectangle) for calculating detection rates; (b) Average detection rate; (c) and (d) detailed detection rates.

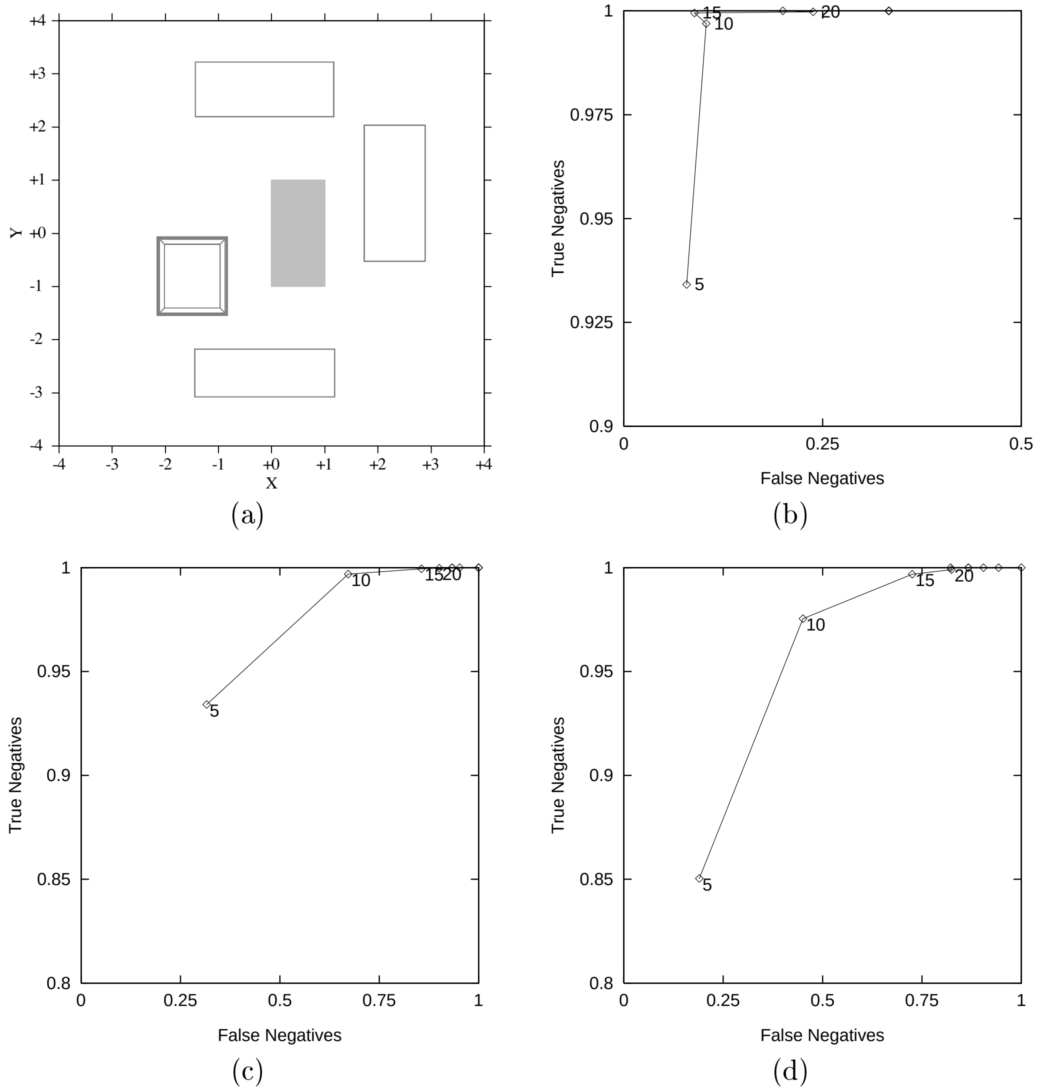


Figure 19. (a) Empty test volume (large shaded rectangle) for testing surfel scale effects; (b) surfel scale effects for an unoccluded surface; (c) an occluded surface without mask; (d) an occluded surface with mask.

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 $\dot{p}_j^i$ does not, thus $(\dot{u}_i, \dot{v}_i)$ must be corrected for the difference between the old and new s_j^i. To update n_j we evaluate:

$$\operatorname*{argmax}_{n_j} \sum_{i \in \mathcal{Q}} \nu(S_j). \tag{20}$$

Both direction set methods and conjugate gradient methods[21] work well to solve Equation 20.

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.

Figure 20 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 20a shows the initial detection for a surfel displaced from the actual surface by 100 units in positions and 30° in orientation. Figure 20b shows the surfel after updating its position and orientation (Steps 1a and 1b of the localization algorithm). Figure 20c shows a new match set using the updated position and orientation (Step 1c). Figure 20d shows the final match set at convergence. The same set of test points used to test detection (Figure 18) was also used to evaluate localization. Figures 21(a) and 21(b) show two test points evaluated at 361 different orientations. The test point for Figure 21(a) is 80 units in front of the actual surface and the one for Figure 21(b) is 100 units behind. The asymmetry in both of these plots is caused by the asymmetric distribution of cameras shown in Figure 25. A *diamond* is plotted at the initial orientation for each detection and a *plus* marks the final estimated orientation. Figures 21(c) and 21(d) show the aggregate results for the complete set of test points. Figure 21(c) shows final displacement versus the initial displacement and the angle between the actual and tested orientation. Figure 21(d) 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.

5.3. Results

As pointed out above, compensating for noisy data admits false positives. Figure 22 shows the raw surfels[22] 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[23]. These are false positives similar to the one shown in Figure 10. Many of the surfels are near the surfaces of the building.

Figure 23 shows the results of applying the detection and localization algorithms to the dataset described in Section 3. Figure 24 shows the distribution of distances to the nearest model surfaces. Notice that while there are a significant number of false positives, most of the detected surfels are near actual surfaces. Figure 25 shows

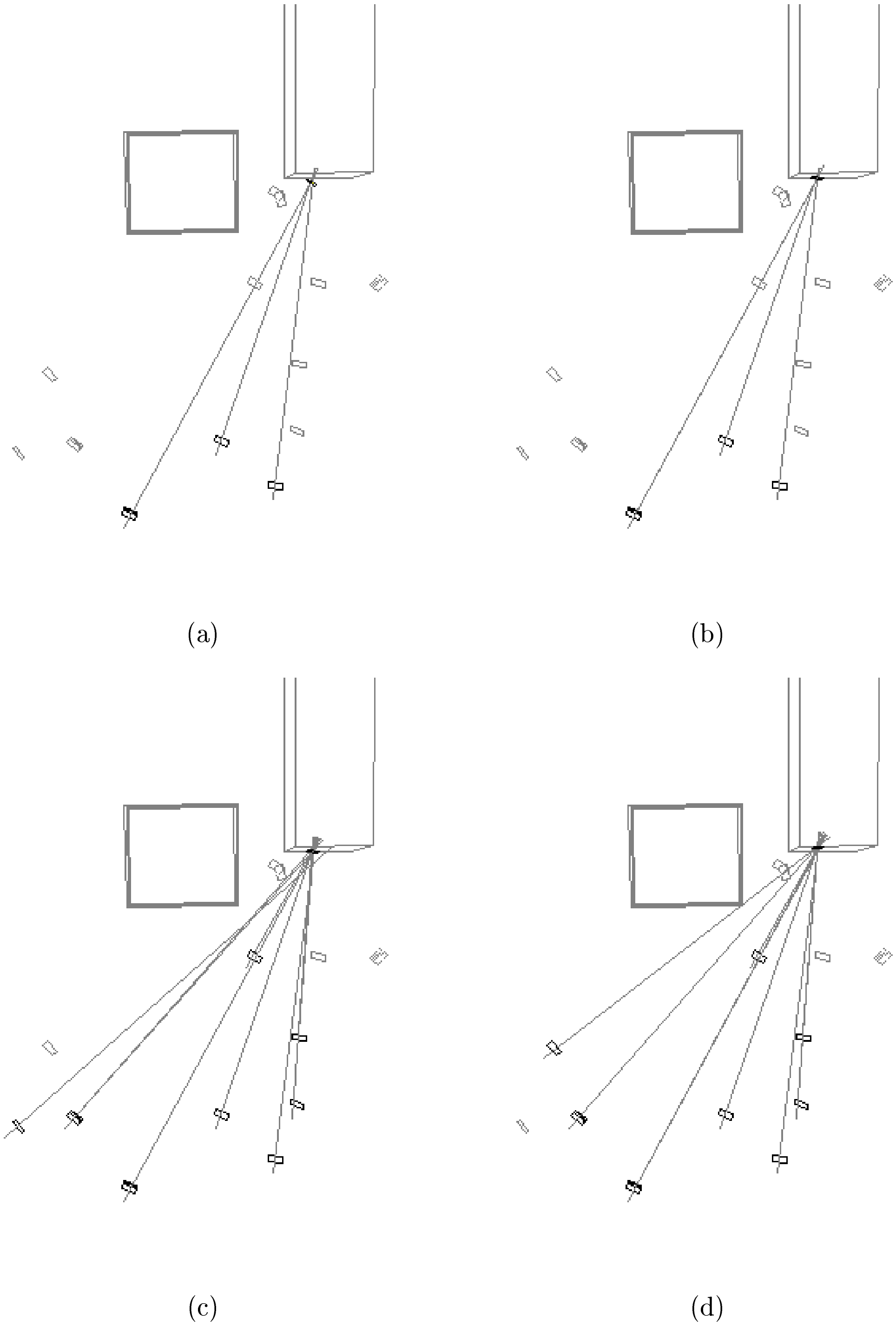


Figure 20. Surfel localization.

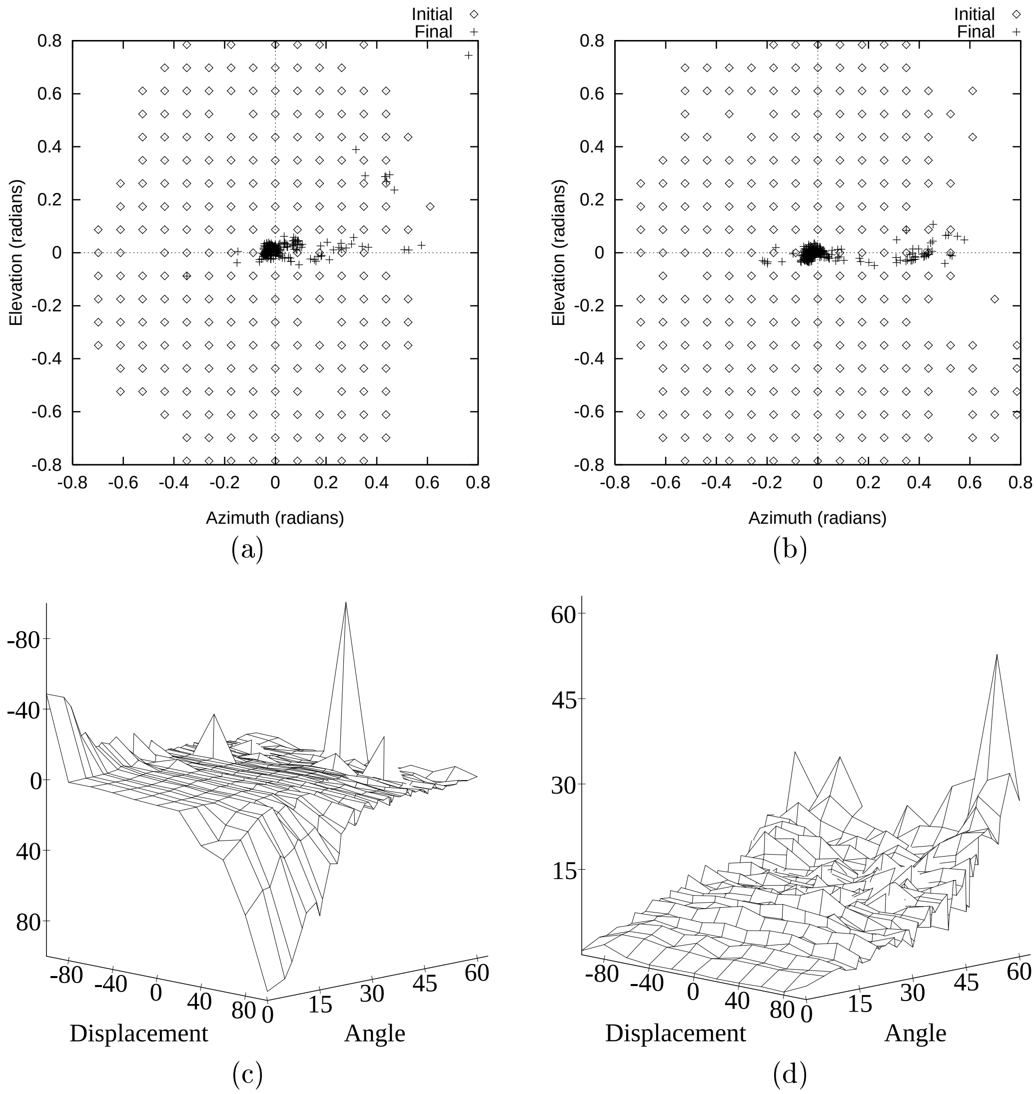


Figure 21. (a) and (b) localization examples; (c) and (d) localization summary.

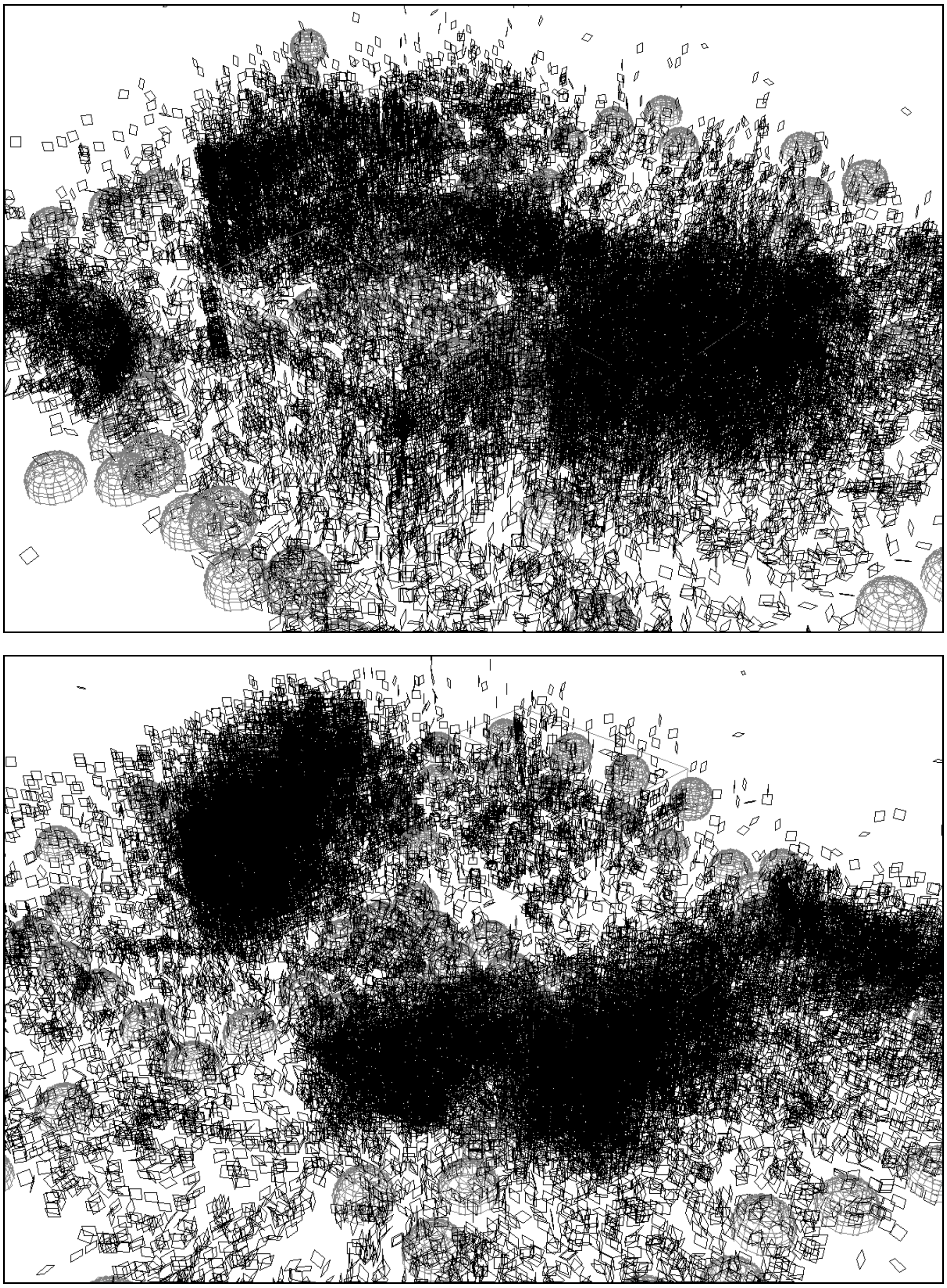


Figure 23. Raw surfels (full reconstruction).

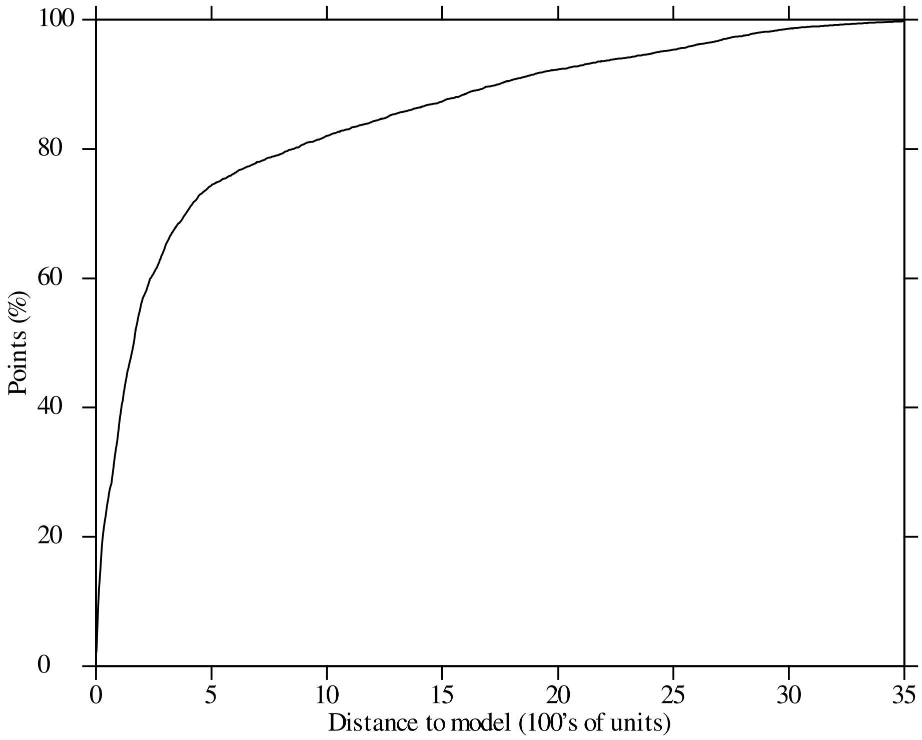


Figure 24. Distance to nearest model surface (full reconstruction).

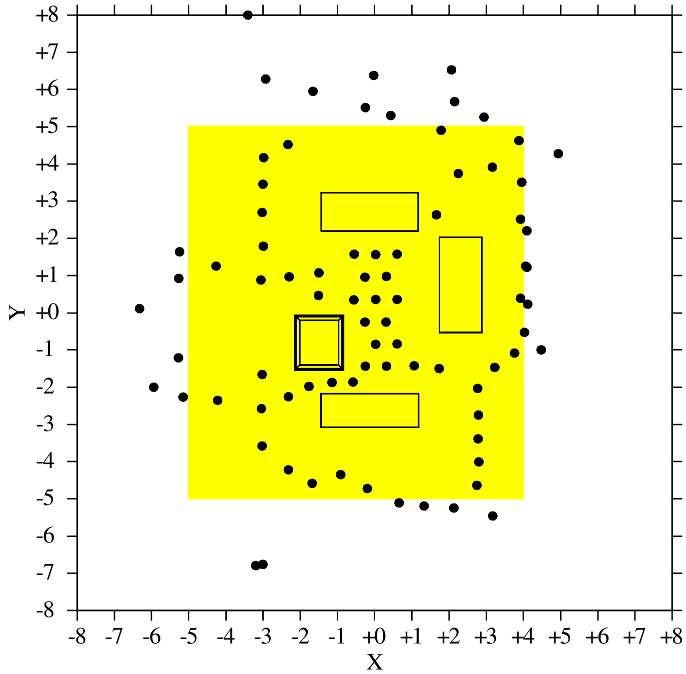


Figure 25. Reconstruction volume (shaded area) used for full reconstruction.

Figure 22. Raw surfels (partial reconstruction).

the reconstruction volume. We hypothesized surfels every 100 units in position and 45° in orientation.

The reconstructions shown 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 26. 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 sufficient information content) which are essentially equivalent. The quarter and eighth resolution reconstructions seem to support this view.

6. Surfels to Surfaces

The results presented in the last section are purely local and make no attempt to reject false positives. This section explores several geometric constraints which together eliminate nearly all false positives.

6.1. Camera Updates

The shifts introduced in Section 4.1 are unconstrained. Figure 27 shows the image shifts 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 (detected surfel). 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

$$(u, v)_{\tilde{\mathrm{p}}_j^i} = \mathcal{T}(\mathrm{P}_j, \mathrm{I}^i) - \mathcal{T}(\mathrm{P}_j, \tilde{\mathrm{I}}^i). \quad (21)$$

To enforce this constraint we use the following algorithm[24]:

1. For each camera:
   a) Collect the shifts used to detect surfels.
   b) Calculate a first-order camera calibration update using these shifts and a linear least-squares solution.
   c) Use the first-order solution to filter the shifts.
   d) Calculate the final camera calibration update using non-linear optimization techniques and Equation 21.

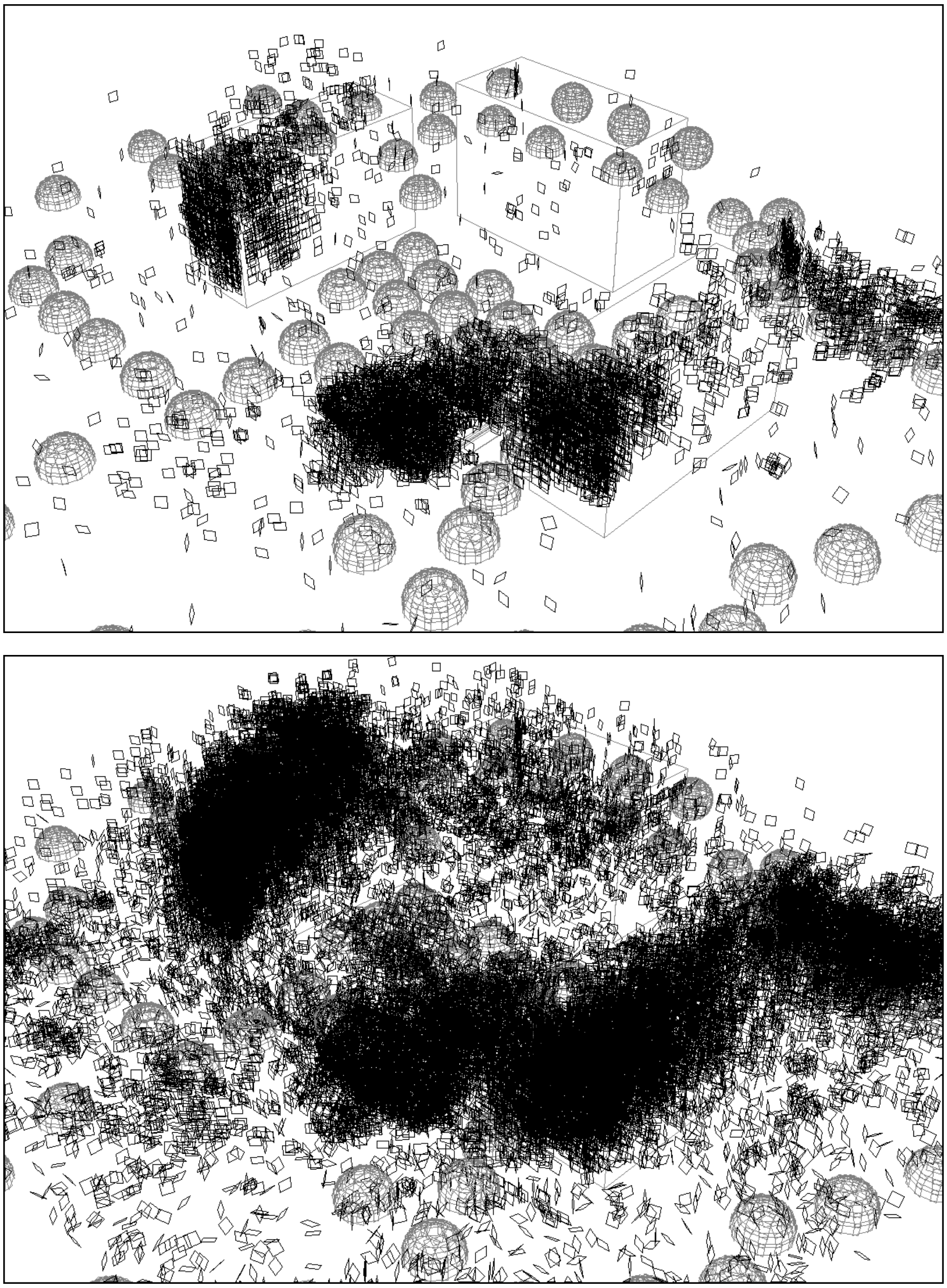


Figure 26. Raw surfels for full reconstruction using half (top) and eighth (bottom) resolution images.

e) Remove matching regions with shifts that are not consistent with the final update (i.e. $\|(\dot{u}, \dot{v}) - \mathcal{T}(\mathrm{P}_j, \tilde{\mathrm{I}}^{i\prime}) + \mathcal{T}(\mathrm{P}_j, \tilde{\mathrm{I}}^{i})\|^2 \geq 1$ pixel).

2. Prune surfels which no longer meet the match criteria (i.e. too many matching regions have been removed because of inconsistent shifts).

Figure 28 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 reduced. 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 29 shows the consistent surfels remaining after applying this algorithm to the raw reconstruction shown in Figure 23 and Figure 30 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 29 is one example. These surfels come from a nearby building.

6.2. One Pixel One Surfel

Each pixel in each image should contribute to at most one surfel. Deciding which surfel is the hard part. Detection and localization as described in the last section do not enforce this constraint and as a result even after enforcing a consistent calibration update there are many

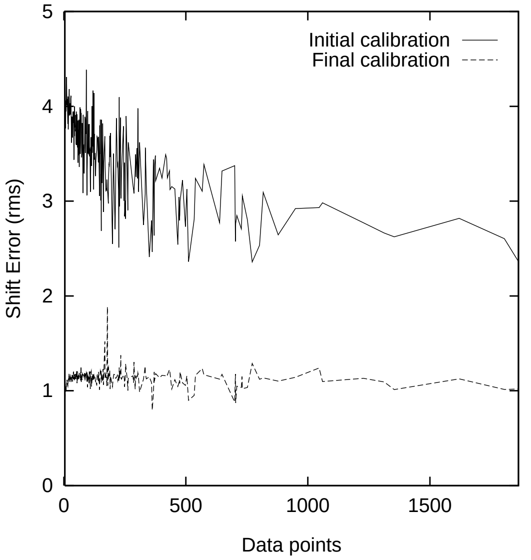


Figure 28. Comparison of initial and final calibration estimates.

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 $\mathrm{s}_a^i \in \dot{\mathbf{s}}_a$.

      i) For each surfel S_b with a score higher than S_a, if $\mathrm{s}_b^i \in \dot{\mathbf{s}}_b$.

         A) De-weight s_a^i's contribution to $\dot{\mathbf{s}}_a$.

   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 the center of S_b to the plane containing S_a (the normal distance) is no more than $\mathrm{d_z}$; 2) the distance from the projection of the center of S_b onto the plane containing S_a and the center of S_a (the tangential distance) is no more than $\mathrm{d_t}$; and, 3) the

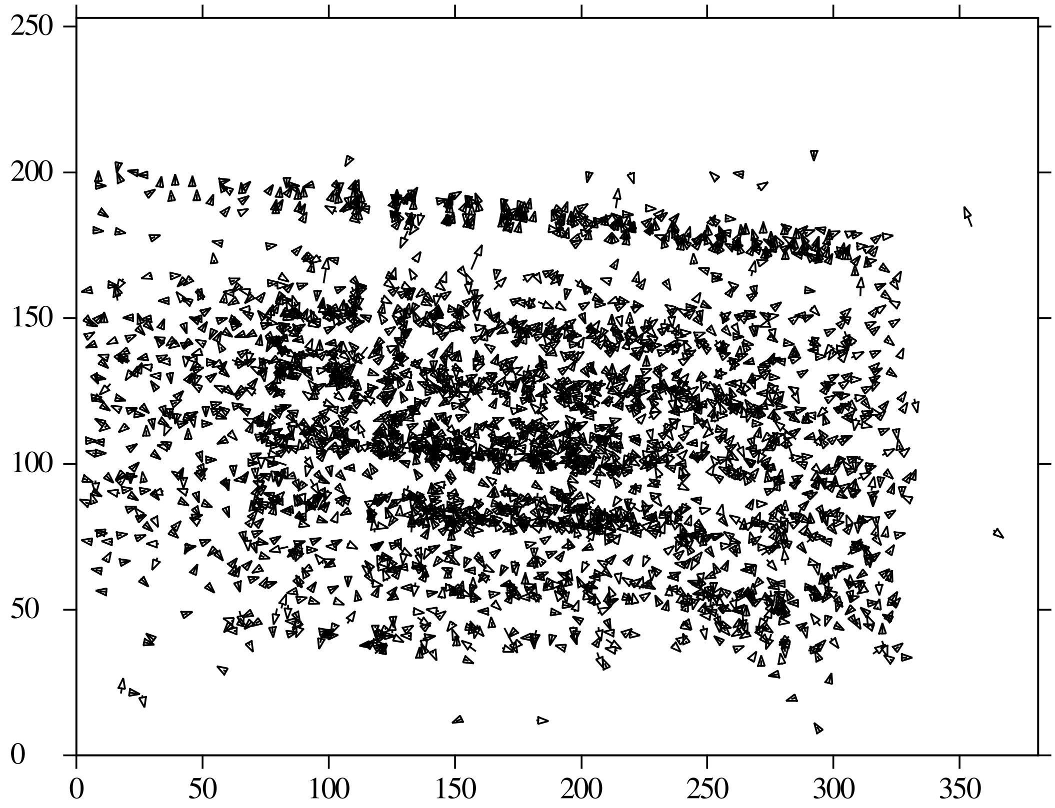


Figure 27. Shifts $\{(\dot{u}_i, \dot{v}_i)\}$ plotted as a vector field and image data for node 27 image 13.

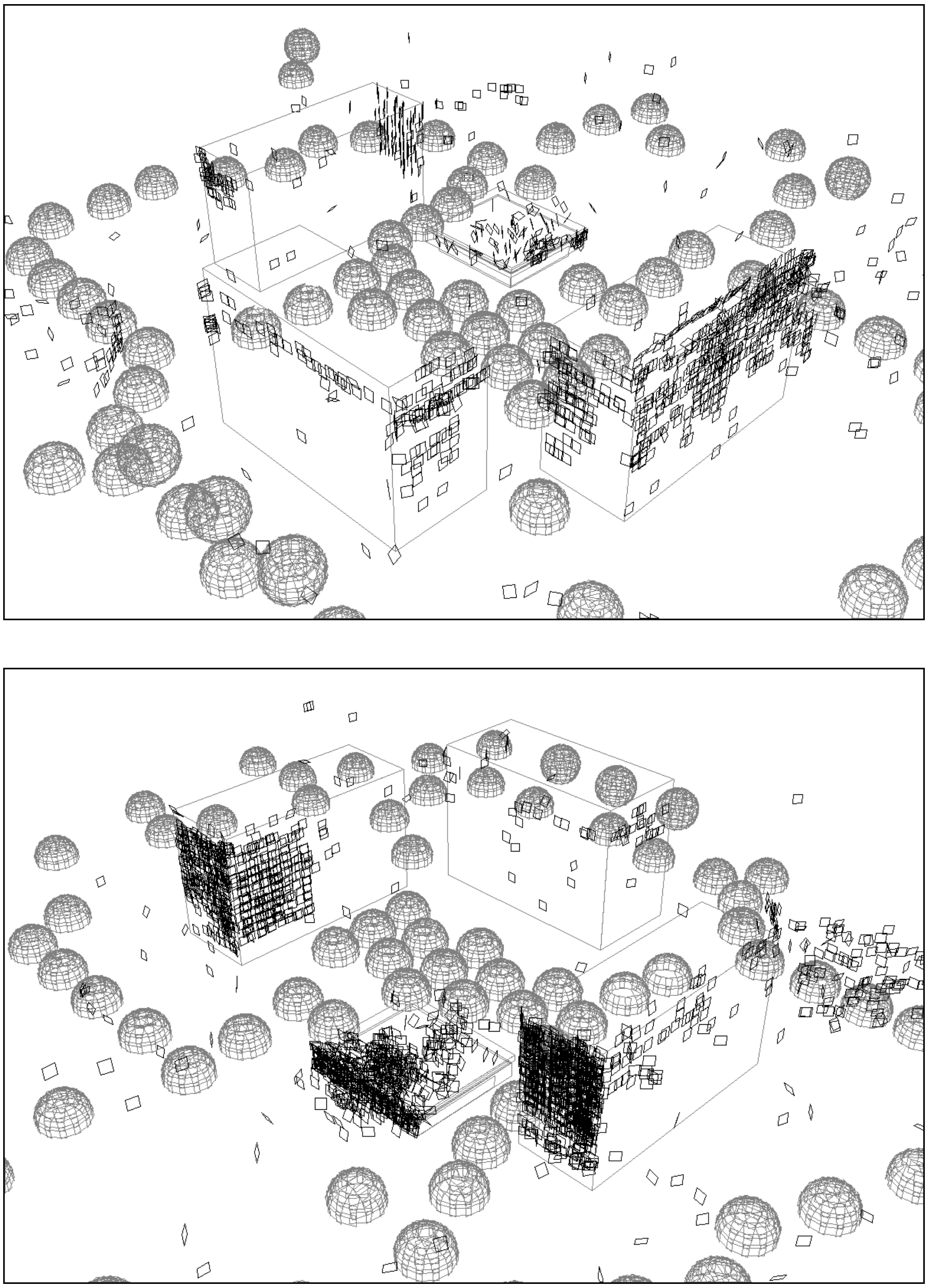


Figure 29. Consistent surfels.

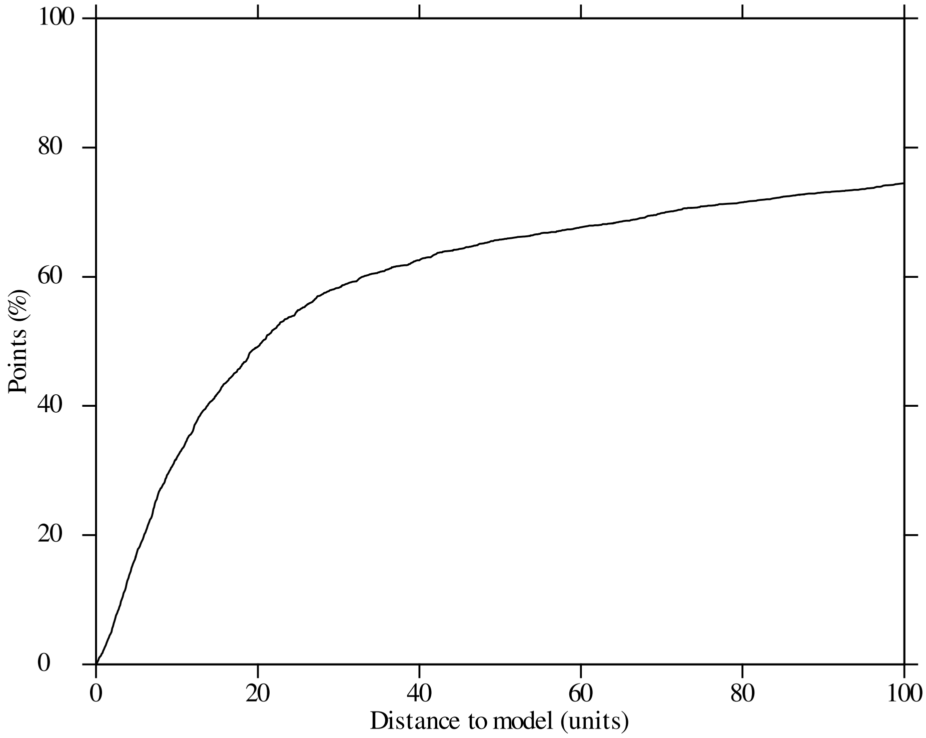


Figure 30. Distribution of error for consistent surfels.

angle between two orientations is no more than β. This notion of neighbors is essentially a smoothness constraint and is also used to group surfels. Typically, we use $\mathrm{d_z} = 15$, $\mathrm{d_t} = 300$, 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 $\widetilde{\mathrm{C}}_i$. 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 $\mathrm{d_z}$, 0.0 if it is greater than $3\mathrm{d_z}$, 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 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 31 shows the reconstruction after pruning multiple contributions and Figure 32 shows the distribution of distances to the nearest model surface.

6.3. 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 above, we group the reconstructed surfels as follows:

1. For each surfel S_a.
   a) For each surfel S_b already assigned a group.

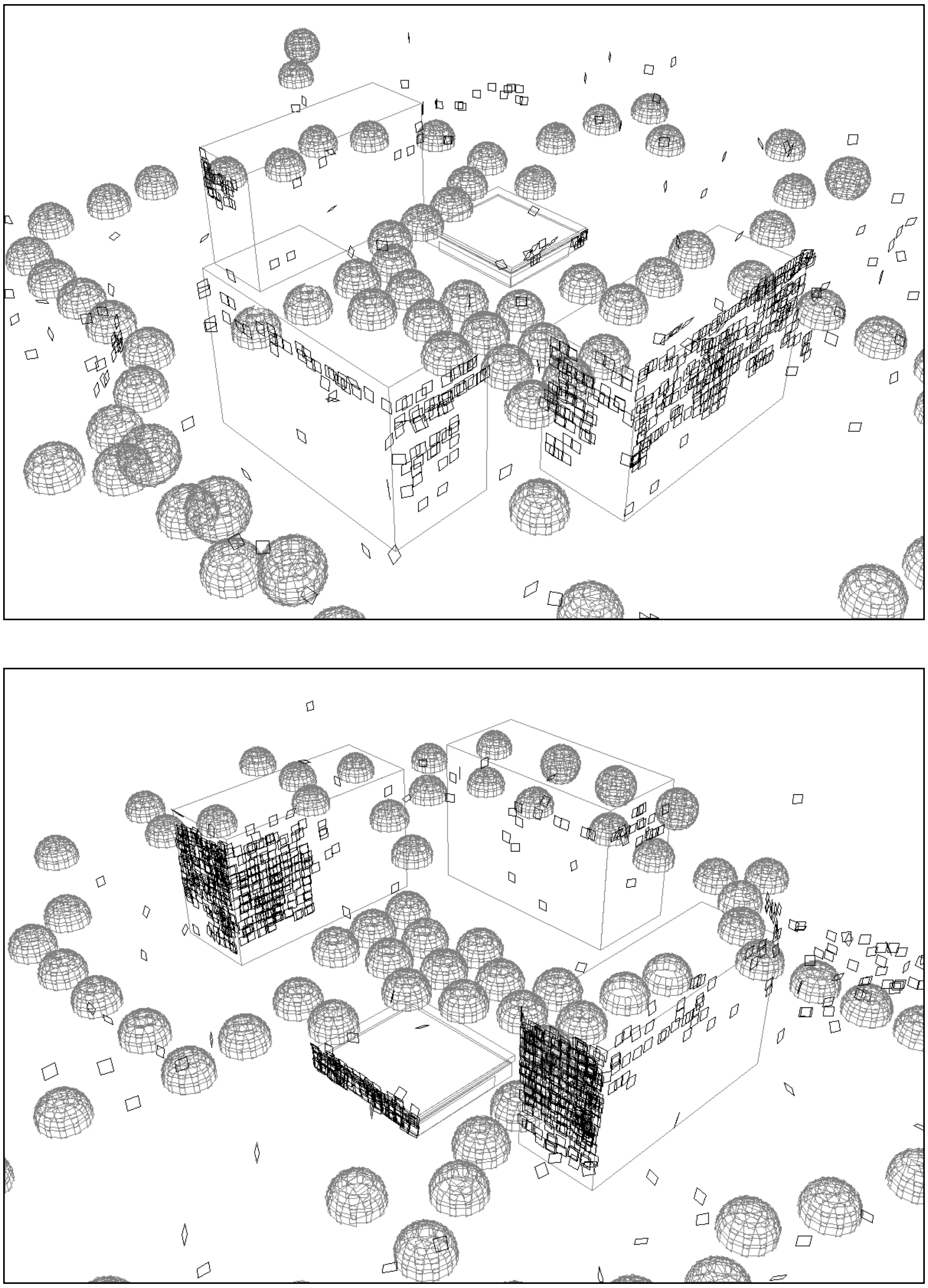


Figure 31. Surfels after pruning multiple contributions.

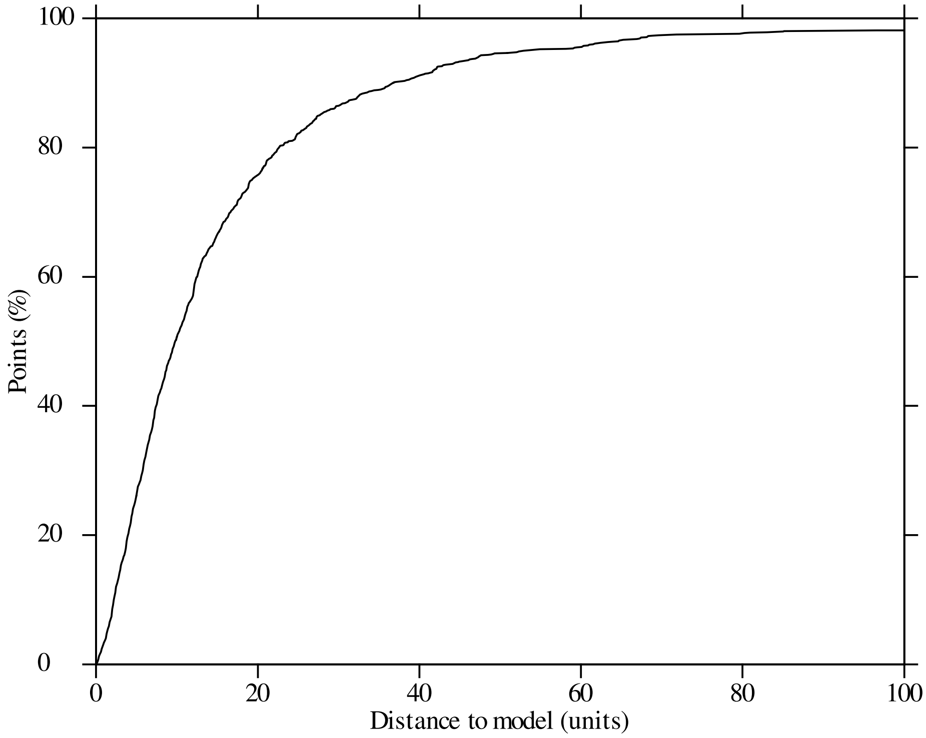


Figure 32. Distribution of error for pruned surfels.

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 33 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 34 shows the distribution of distances to the nearest model surface.

6.4. Growing Surfaces

Many of the groups shown in Figure 33 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 from tree branches. 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 4.3 has not been utilized. In this section we make use of it. We also use estimated the shifts and

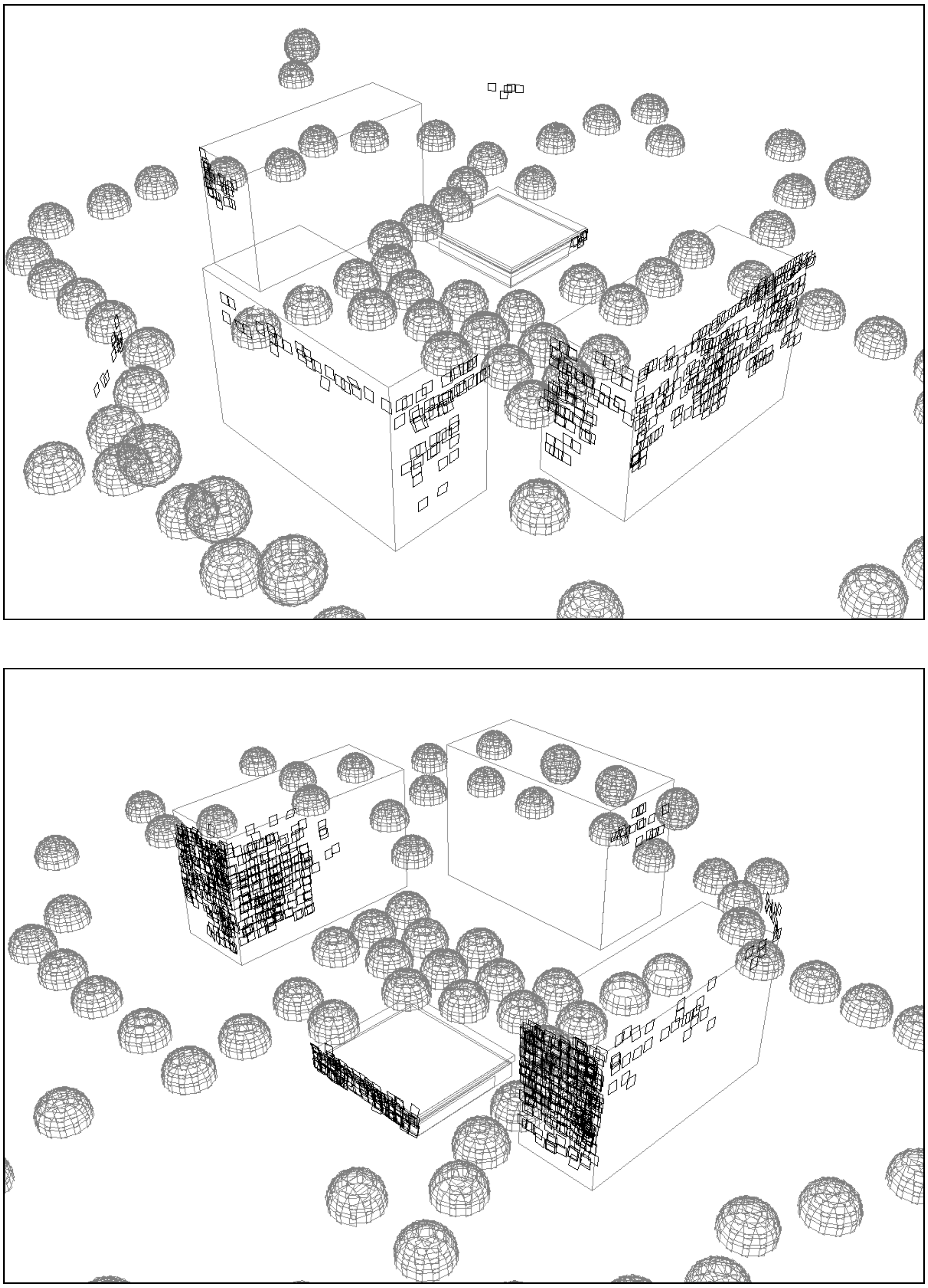


Figure 33. Surfels after grouping.

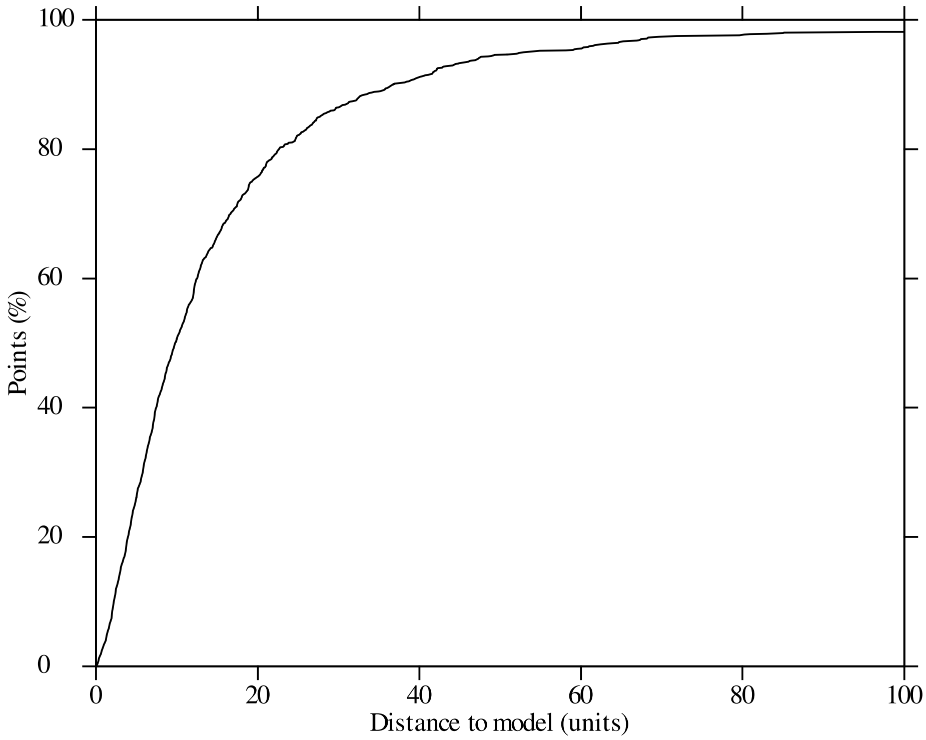


Figure 34. Distribution of error for grouped surfels.

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. 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 35 shows the reconstruction after growing. After growing, the coverage of each surface is nearly complete. Figure 36 shows the distribution of distances to the nearest model surface.

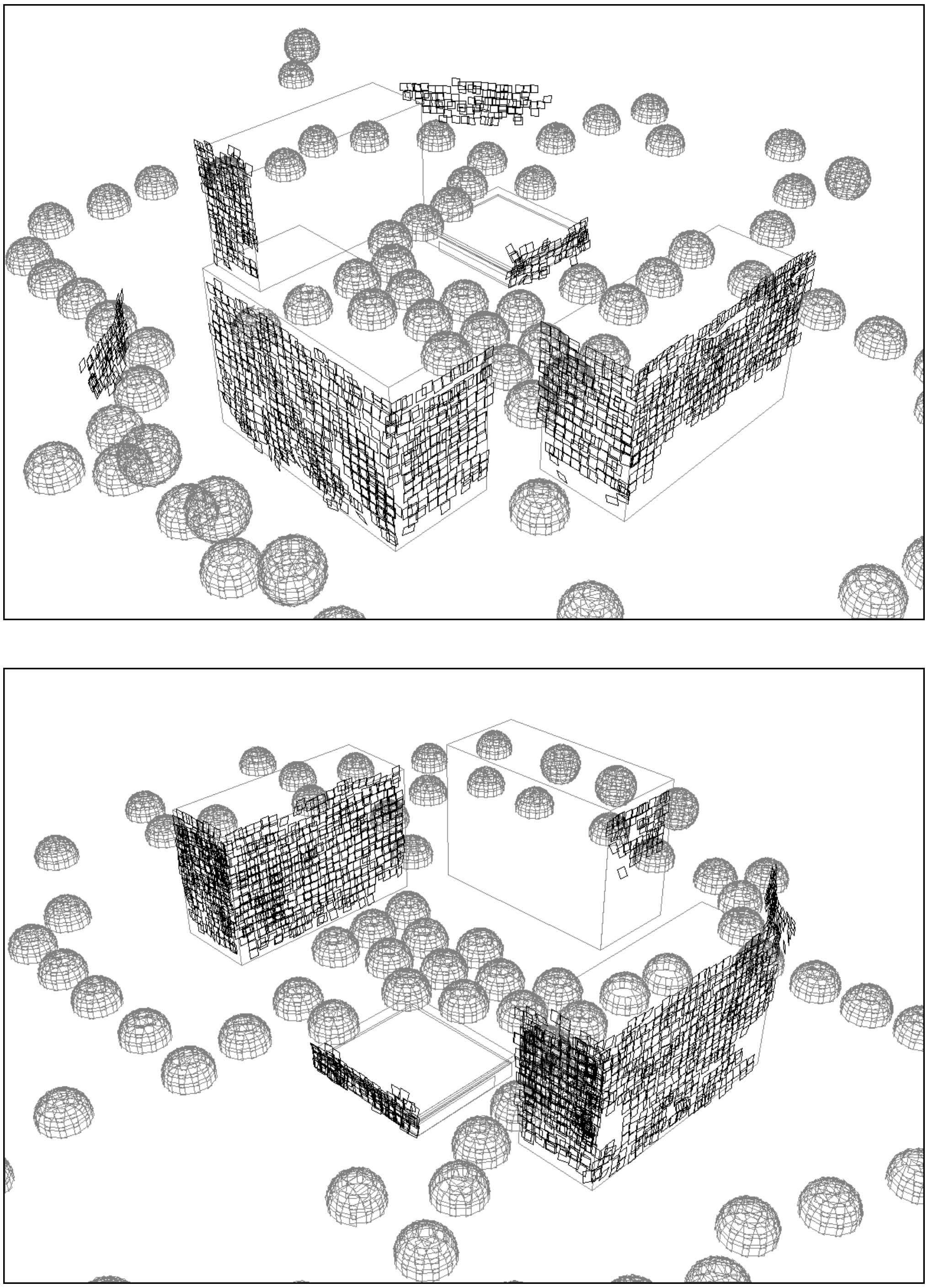


Figure 35. Surfels after regrouping.

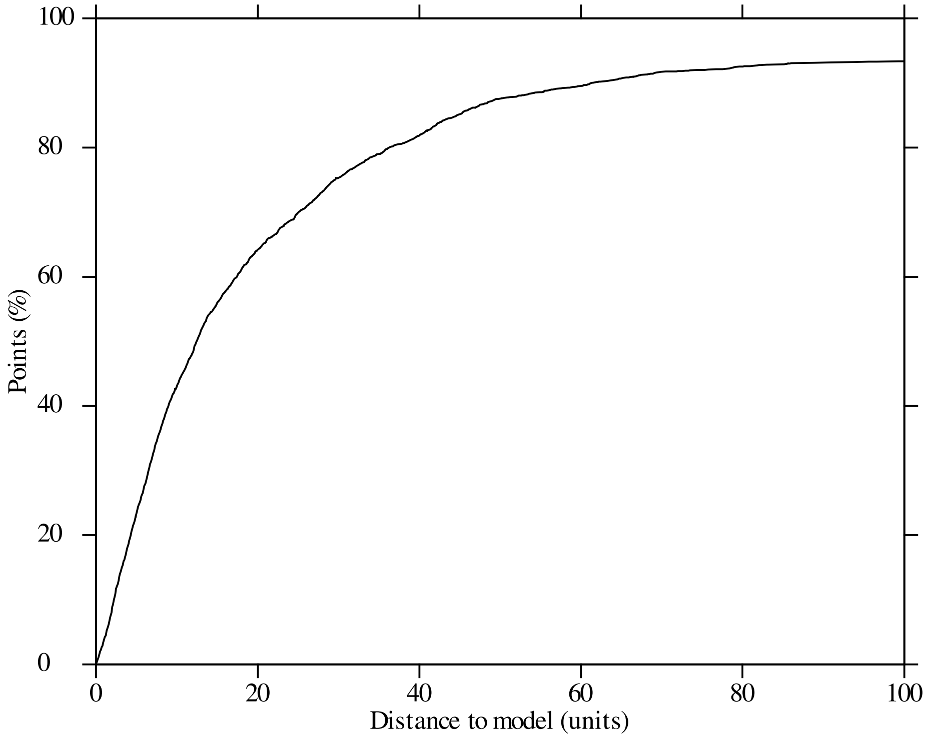


Figure 36. Distribution of error for grown surfels.

6.5. 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 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. Figure 37 shows the reconstructed faces. A total of 15 surfaces were recovered. Figure 38 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 38 has a maximum of 80.

Using the illumination corrections calculated during detection and localization we can transform the images which contribute to a face into a common color space. To obtain the texture associated with each face, we simply reproject the color corrected images and average the results. Figure 39 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. For more details about the model fitting stage see Chapter 5 of (Mellor, 1999).

6.6. Discussion

This section uses several simple geometric constraints to remove virtually all false positives from the purely local reconstruction described in Section 5. After imposing

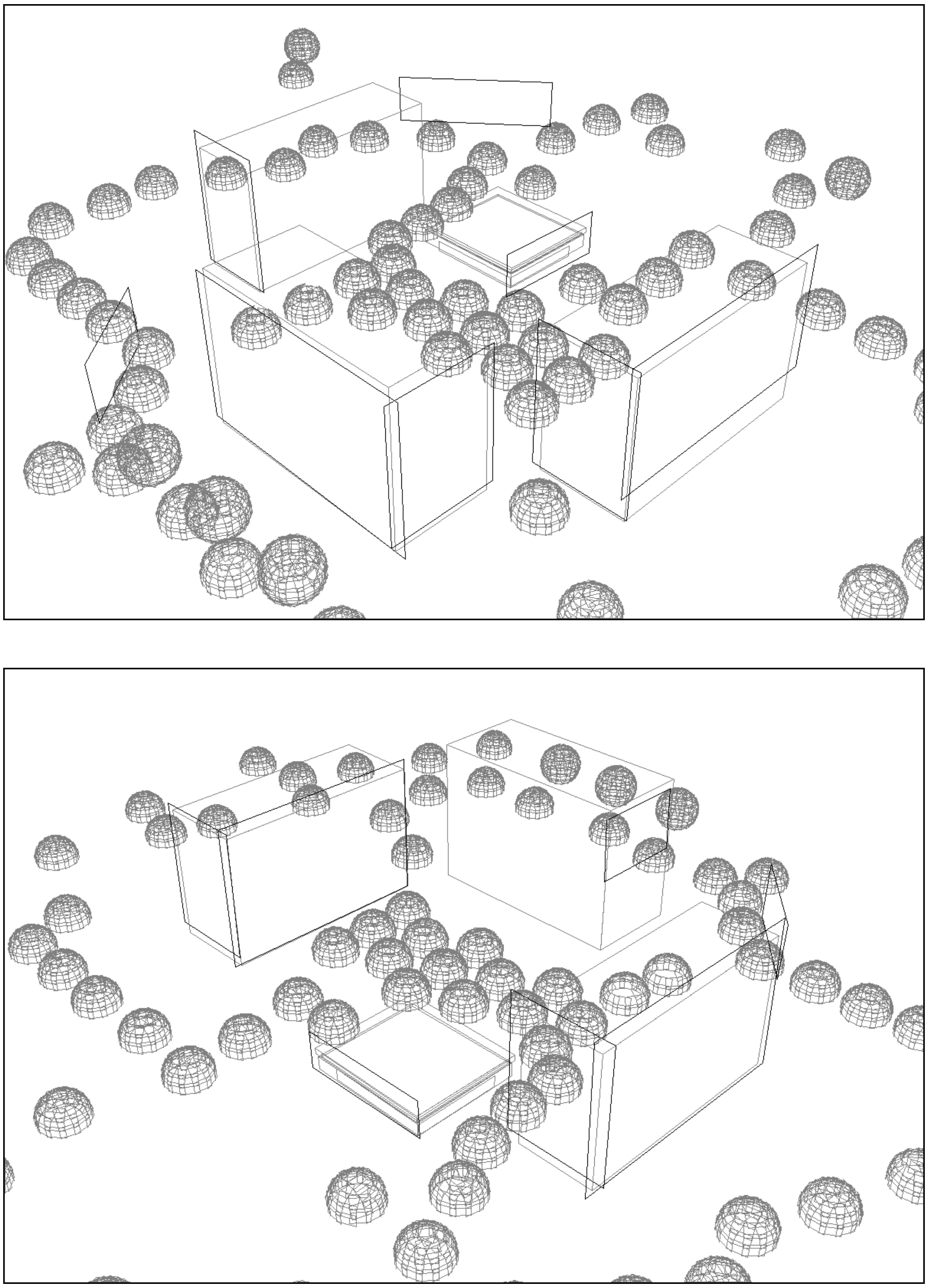


Figure 37. Raw model surfaces.

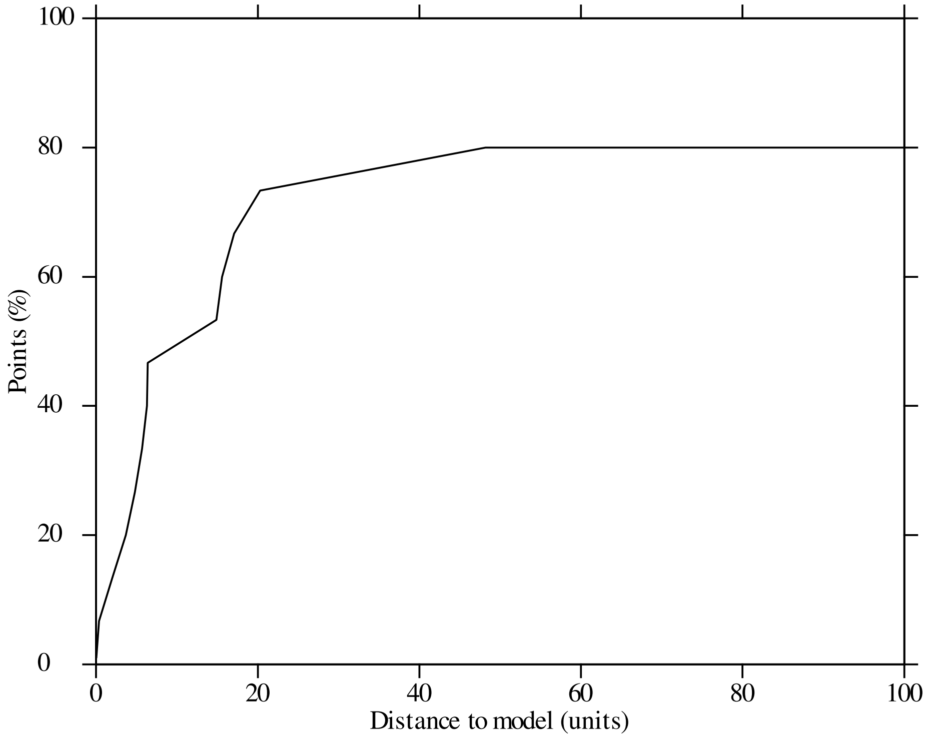


Figure 38. Distribution of error for model surfaces.

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 in Figure 6 and 8. In spite of this several surfels were recovered on two of these surfaces, however they did not survive the grouping process. In addition to being severely 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 39 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 shown in Table V. 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 shown in Figure 39. 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 paral-

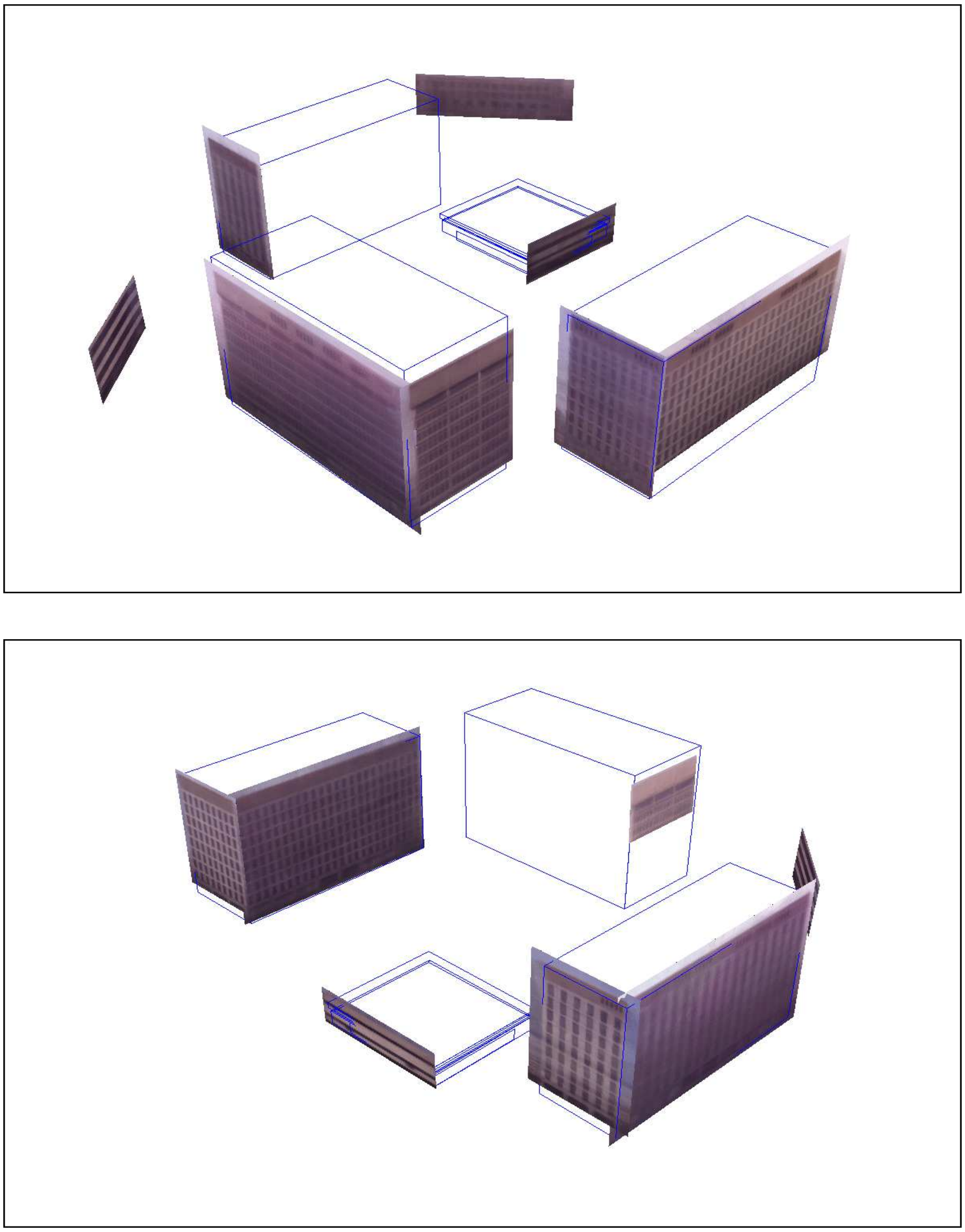


Figure 39. Textured model surfaces.

lel 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, 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, 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 39 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 $O(A \times N)$.

7. Conclusions

This paper 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 (in the worst case) 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 scales well

Table V. Run times and orders of growth.

Step	Section	Time Elapsed	Time cpu	Surfels	Complexity
Detection & Localization	5	42hrs†	60hrs	54,212	$O(V \times N \times S)$
Camera Updates	6.1	18.5min	17min	2977	$O(N \times \#^2)$
1 Pixel, 1 Surfel	6.2	3.5min	2min	1636	$O(N \times \#^2)$
Group	6.3	4min	2.5min	1272	$O(\#^2)$
Grow	6.4	6hrs†	4hrs	3007	$O(A \times N)$
Model	6.5	11.25min	9.5min	15	$O(\#)$
Texture	6.5	24.5min	15.5min	15	$O(A \times N)$

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 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 paper) is the first to produce high-quality textured models from such large image sets. The image sets used are nearly two orders of magnitude larger than the largest sets used by other approaches. The approach presented in this paper, recovering dense surfels by matching raw image data directly in three-dimensional space, is unique among the City Scanning approaches.

7.1. Future Work

The major limitation of the work presented in this paper is the difficulty of performing reconstruction "through the trees". A large part of the difficulty is caused by selecting one region and matching others against it. If the key region is not representative of the underlying texture, then the match set will most likely be insufficient resulting in a false negative. If we knew the underlying texture and used it as the key region, the match set should improve significantly. Similarly, estimating the underlying texture, rather than simply selecting one of the views, and using it as should also improve the match. Potentially this would allow us recover the missing faces in Figure 39.

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.

Acknowledgements

This paper describes research performed at the Massachusetts Institute of Technology Artificial Intelligence Laboratory and was 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 Lab contract F3060-94-C-0204.

Notes

[1] 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.

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

[3] Interestingly, if is $w_i = 1$ then Equation 1 is closely related to the tomographic algorithm (Ramachandran and Lakshminarayanan, 1971; Gering and Wells, 1999).

[4] $\star$ is one of the elements in $\mathcal{Q}$ chosen to match against the other elements. We will discuss how it is chosen in Sections 4.3 and 5.1.

[5] Except possibly in adversarial settings.

[6] 381×253 pixels.

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

[9] Using the estimated camera calibration.

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

[11] Ignoring errors introduced during image formation (discretizing into pixels, etc) and resampling.

[12] Equation 8 is commonly called the image sensor's intensity response function (Vora et al., 1997a; Vora et al., 1997b).

[13] This matching metric is closely related to normalized correlation, however we impose nonlinear constraints on m and d.

[14] The region may contain materials with reflectances which depend upon viewing direction, however the dependence must be similar throughout.

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

[16] 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 of (Mellor, 1999) for the derivation.

[17] The correction for the 4^{th} region in row 3 of Figures 5 and 16 which corrects a specular reflection in the window exceeds the limits described above.

[18] While growing surfaces (discussed in the Section 6) we allow uniqueness as low as 1.5.

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

[20] 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.

[21] While not as straight forward as the previous gradient, 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 of (Mellor, 1999) for the derivation.

[22] Only front facing surfels are rendered.

[23] In fact, the points near angle 0 and displacements of ± 100 in Figure 21 converge to layers such as these.

[24] See Section 5.1 of (Mellor, 1999) for additional details.

References

Becker, S. and V. M. Bove, Jr.: 1995, 'Semi-automatic 3-D Model Extraction from Uncalibrated 2-D Camera Views'. In: *Visual Data Exploration and Analysis*, Vol. 2410. pp. 447–461. San Jose, CA.

Collins, R., C. Jaynes, F. Stolle, X. Wang, Y. Cheng, A. Hanson, and E. Riseman: 1995, 'A System for Automated Site Model Acquisition'. In: *Integrating Photogrammetric Techniques with Scene Analysis and Machine Vision II*, Vol. 7617. Orlando, FL.

Collins, R. T.: 1996, 'A Space-Sweep Approach to True Multi-Image Matching'. In: *Computer Vision and Pattern Recognition (CVPR '96 Proceedings)*. pp. 358–363. San Francisco, CA.

Coorg, S.: 1998, 'Pose Imagery and Automated 3-D Modeling of Urban Environments'. Ph.D. thesis, MIT.

Coorg, S., N. Master, and S. Teller: 1998, 'Acquisition of a Large Pose-Mosaic Dataset'. In: *Computer Vision and Pattern Recognition (CVPR '98 Proceedings)*. pp. 872–878. Santa Barbara, CA.

De Couto, D. S. J.: 1998, 'Instrumentation for Rapidly Acquiring Pose-Imagery'. Master's thesis, MIT.

Debevec, P. E., C. J. Taylor, and J. Malik: 1996, 'Modeling and Rendering Architecture from Photographs: A Hybrid Geometry- and Image-Based Approach'. In: *Computer Graphics (SIGGRAPH '96 Proceedings)*. pp. 11–20.

Gering, D. T. and W. M. Wells, III: 1999, 'Object Modeling Using Tomography and Photography'. In: *IEEE Workshop on Multi-view Modeling and Analysis of Visual Scenes*. pp. 11–18. Fort Collins, CO.

Kutulakos, K. N. and S. M. Seitz: 1998a, 'A Theory of Shape by Space Carving'. CS Technical Report 692, University of Rochester.

Kutulakos, K. N. and S. M. Seitz: 1998b, 'What Do N Photographs Tell Us About 3D Shape?'. CS Technical Report 680, University of Rochester.

Li, X. and T. Chen: 1995, '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.

Mellor, J.: 1999, 'Automatically Recovering Geometry and Texture from Large Sets of Calibrated Images'. Technical Report AITR–1674, MIT.

Mellor, J., S. Teller, and T. Lozano-Pérez: 1997, 'Dense Depth Maps from Epipolar Images'. In: *Image Understanding Workshop (IUW '97 Proceedings)*, Vol. 2. pp. 893–900. New Orleans, LA.

Mellor, J., S. Teller, and T. Lozano-Pérez: 1998, 'Dense Surface Patches from Thousands of Pose Images'. In: *Image Understanding Workshop (IUW '98 Proceedings)*, Vol. 2. pp. 537–542. Monterey, CA.

Press, W. H., S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery: 1992, *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press.

Ramachandran, G. N. and A. V. Lakshminarayanan: 1971, '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.

Seitz, S. M. and C. R. Dyer: 1997, 'Photorealistic Scene Reconstruction by Voxel Coloring'. In: *Computer Vision and Pattern Recognition (CVPR '97 Proceedings)*. pp. 1067–1073. Puerto Rico.

Shum, H.-Y., M. Han, and R. Szeliski: 1998, 'Interactive Construction of 3D Models from Panoramic Mosaics'. In: *Computer Vision and Pattern Recognition (CVPR '98 Proceedings)*. pp. 427–433. Santa Barbara, CA.

Szeliski, R. and D. Tonnesen: 1992, 'Surface Modeling with Oriented Particle Systems'. In: *Computer Graphics (SIGGRAPH '92 Proceedings)*, Vol. 26. pp. 185–194.

Teller, S.: 1998a, 'Automated Urban Model Acquisition: Project Rationale and Status'. In: *Image Understanding Workshop (IUW '98 Proceedings)*, Vol. 2. pp. 455–462. Monterey, CA.

Teller, S.: 1998b, 'Toward Urban Model Acquisition from Geo-Located Images'. In: *Pacific Graphics '98*. Singapore.

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

Vora, P. L., J. E. Farrell, J. D. Tietz, and D. H. Brainard: 1997a, 'Digital Color Cameras - Response Models'. Technical Report HPL-97-53, Hewlett-Packard Laboratories.

Vora, P. L., J. E. Farrell, J. D. Tietz, and D. H. Brainard: 1997b, 'Digital Color Cameras - Spectral Response'. Technical Report HPL-97-54, Hewlett-Packard Laboratories.

Notes

[1] 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.

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

[3] Interestingly, if is $w_i = 1$ then Equation 1 is closely related to the tomographic algorithm (Ramachandran and Lakshminarayanan, 1971; Gering and Wells, 1999).

[4] $\star$ is one of the elements in $\mathcal{Q}$ chosen to match against the other elements. We will discuss how it is chosen in Sections 4.3 and 5.1.

[5] Except possibly in adversarial settings.

[6] 381×253 pixels.

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

[9] Using the estimated camera calibration.

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

[11] Ignoring errors introduced during image formation (discretizing into pixels, etc) and resampling.

[12] Equation 8 is commonly called the image sensor's intensity response function (Vora et al., 1997a; Vora et al., 1997b).

[13] This matching metric is closely related to normalized correlation, however we impose nonlinear constraints on m and d.

[14] The region may contain materials with reflectances which depend upon viewing direction, however the dependence must be similar throughout.

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

[16] 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 of (Mellor, 1999) for the derivation.

[17] The correction for the 4^{th} region in row 3 of Figures 5 and 16 which corrects a specular reflection in the window exceeds the limits described above.

[18] While growing surfaces (discussed in the Section 6) we allow uniqueness as low as 1.5.

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

[20] 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.

[21] While not as straight forward as the previous gradient, 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 of (Mellor, 1999) for the derivation.

[22] Only front facing surfels are rendered.

[23] In fact, the points near angle 0 and displacements of ± 100 in Figure 21 converge to layers such as these.

[24] See Section 5.1 of (Mellor, 1999) for additional details.

List of Figures

List of Tables