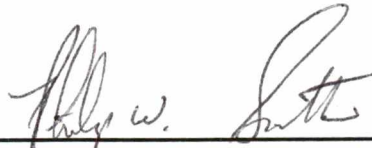


To the Graduate Council:

I am submitting herewith a thesis written by Mark D. Elstrom entitled "A Stereo-Based Technique for the Registration of Color and LADAR Images." I have examined the final copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Electrical Engineering.

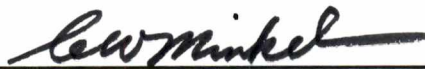


P. W. Smith, Major Professor

We have read this thesis
and recommend its acceptance:



Accepted for the Council:



Associate Vice Chancellor
and Dean of The Graduate School

**A STEREO-BASED TECHNIQUE FOR THE
REGISTRATION OF COLOR AND LADAR IMAGES**

A Thesis

Presented for the

Master of Science

Degree

The University of Tennessee, Knoxville

Mark D. Elstrom

August 1998

ABSTRACT

In many remote robotic tasks involving tele-operation, operator performance is enhanced by integrating information acquired from various sensors, such as color, infrared, and laser range cameras. A major difficulty in designing such a multi-sensor system is the development of methods for combining the various data sources into one world model for examination by the operator. By registering different sensing modalities with LADAR range images, visualization techniques can be employed for sensor integration. Using texture mapping, registered images can be displayed through projection onto the range model to provide for 3D visualization of the data acquired from each sensor.

Presented in this thesis is a novel stereo-based method for registering color and LADAR images. In this approach, corresponding features in perspective-projected LADAR intensity images and images acquired using a color camera are employed to calculate depth using triangulation, given a rough initial guess of the relative sensor positions. The sum-squared-error of the stereo and LADAR range values is then calculated, and a downhill simplex algorithm is iteratively applied to refine the rotation and translation estimates while rejecting outliers.

Finally, the rotation and translation values are used along with the LADAR range data to reproject the color image to the viewpoint of the scanner. This resulting registered color image is used as a texture map to create a photo-realistic, 3D color model of the scene.

ACKNOWLEDGMENTS

I would like to thank my mother, Sugi McIntosh, for her outstanding support and continuous encouragement throughout my entire educational career. My deepest gratitude goes to my fiancée, Jusenía, for her unwavering encouragement and understanding, especially during the writing of this document. Thanks to Dr. M. A. Abidi for selecting me for the Graduate Research Assistant position in the Imaging, Robotics and Intelligent Systems laboratory and to Dr. R. C. Gonzales for choosing me as a Graduate Teaching Assistant, both of which made obtaining my master's degree financially possible. I also wish to thank my advisors, Dr. P. W. Smith and Dr. M. A. Abidi, for their guidance throughout my program. Finally, thanks to the members of my committee, Dr. P. W. Smith, Dr. M. A. Abidi and Dr. R. T. Whitaker for their help and constructive criticism.

The work in this thesis was supported by the DOE's University Research Program in Robotics (Universities of Florida, Michigan, New Mexico, Tennessee, and Texas) under grant DOE-DE-FG02-86NE37968.

TABLE OF CONTENTS

CHAPTER	PAGE
1. Introduction	1
1.1 Related Work	7
2. Registration of LADAR and Projective Imagery	10
2.1 The Perceptron	11
2.2 Transformation from LADAR to Euclidean and Projective Spaces	13
2.2.1 Transformation from LADAR Coordinate System to Cartesian Coordinate System	14
2.2.2 Transformation from LADAR Coordinate System to Perspective Projection	17
2.2.3 Generating Panoramic Range Images	20
2.3 Feature Correspondence	24
2.3.1 Manual Feature Correspondence	25
2.3.2 Automatic Feature Correspondence	27
2.4 Determining the Epipolar Geometry of the LADAR-Color Camera System	33
2.4.1 Linear method (Least-Squares Algorithm)	34
2.4.2 Nonlinear method (Simplex Algorithm)	36
2.5 Reprojection	44
3. Experimental Results and Discussion	46

CHAPTER	PAGE
3.1 Synthetic Data Results	46
3.2 Discussion of Synthetic Data Results	50
3.3 Registration Results using Actual Imagery	51
3.4 Discussion of Registration Results using Actual Imagery	61
4. Conclusions	64
BIBLIOGRAPHY	67
VITA	72

LIST OF FIGURES

FIGURE	PAGE
1.1 Raw LADAR range image of a scene containing various pipes and valves	2
1.2 An untextured three-dimensional world model created from multiple range images taken of the scene from different viewpoints.	3
1.3 Color, infrared, and gamma images of the same scene, but taken from different locations.	4
1.4 Multi-sensor visualization scheme which describes how information from other sensors is combined with range data to create complete 3-D models	5
2.1 Flowchart describing the entire registration process, beginning with an unregistered range/color image pair as input and ending with a registered pair as its result.	10
2.2 The Perceptron laser range camera and mobile scanning system.	11
2.3 The scanning method employed by the azimuth-elevation mirror sys- tem used in the Perceptron.	12
2.4 (a) A Perceptron intensity/range image pair.	13
2.5 Flowchart describing the process of converting raw LADAR images to Cartesian coordinates and projective images.	14
2.6 The coordinate axis and scanning angles used by the Perceptron.	15
2.7 Azimuth and elevation rotations used to calculate Cartesian coordinates.	16
2.8 A raw range image and the $2\frac{1}{2}$ -D model created from it.	17

FIGURE	PAGE
2.9 A raw intensity image and its perspective projection.	19
2.10 Panoramic intensity/range image pair of computer lab after transformation to spherical coordinates.	22
2.11 Panoramic intensity/range image pair of building lobby after transformation to spherical coordinates.	23
2.12 A sample intensity/color image pair	24
2.13 Flowchart describing the feature correspondence process using both manual and automatic methods.	25
2.14 Range data illustrating noisy data at edges.	26
2.15 Manually drawn lines in intensity/ color image pair.	27
2.16 Automatically detected corners in intensity/color image pair.	29
2.17 Automatically determined correspondence set in an intensity/range image pair.	32
2.18 Flowchart describing the process of finding the relative orientation of the LADAR camera and the color camera.	34
2.19 Rectification of intensity and color images.	38
2.20 Rotation of cameras to maximize baselines.	39
2.21 Determination of stereo equations when cameras have a z- translation.	41
2.22 Comparison of registration results using the least-squares and simplex methods.	43
2.23 Registered color image with and without z-buffering.	45
3.1 Perspective projections of synthetic 'apple' model.	46

3.2	Convergence of $(\mathbf{R}, \vec{T})$ in the stereo-simplex algorithm for the synthetic apple model.	48
3.3	LADAR/stereo depth errors for 'apple' model with outliers included. . .	49
3.4	LADAR/stereo depth errors for 'apple' model with outliers removed. . .	49
3.5	Smoothing of spike noise in the synthetic 'apple' model.	51
3.6	Automatically detected corners in 'boxes' image pair.	52
3.7	Registered color image of the 'boxes' scene with z-buffering.	52
3.8	Color texture-mapped $2\frac{1}{2}$ -D model of 'boxes' scene.	53
3.9	Convergence of $(\mathbf{R}, \vec{T})$ in the stereo-simplex algorithm for the 'boxes' scene.	54
3.10	Automatically detected corners in 'desk' image pair.	55
3.11	Color texture-mapped $2\frac{1}{2}$ -D model of 'desk' scene.	55
3.12	Convergence of $(\mathbf{R}, \vec{T})$ in the stereo-simplex algorithm for the 'desk' scene.	56
3.13	Automatically detected corners in 'office' image pair.	57
3.14	Color texture-mapped $2\frac{1}{2}$ -D model of 'office' scene.	57
3.15	Convergence of $(\mathbf{R}, \vec{T})$ in the stereo-simplex algorithm for the 'office' scene.	58
3.16	Automatically detected corners in 'people' image pair.	59
3.17	Color texture-mapped $2\frac{1}{2}$ -D model of 'people' scene.	59
3.18	Convergence of $(\mathbf{R}, \vec{T})$ in the stereo-simplex algorithm for the 'people' scene.	60
3.19	Failure of registration algorithm due to identical camera placement. . .	61

3.20 Failure of registration algorithm due to large axial rotation difference. .	62
--	----

CHAPTER 1

Introduction

In many remote robotic tasks involving tele-operation, the performance of the operator is enhanced by integrating information acquired from various sensors. For example, video provides color visuals of a scene that are easy for an operator to interpret. Infrared and gamma radiation cameras return images containing information about heat and radiation from a scene respectively. Temperature and radiation data are useful for tasks such as locating steam leaks or detecting unexpected radiation sources.

One of the more important sensors available for use in robotic systems is the laser range, or LADAR (**L**ight **A**mplitude **D**etection **A**nd **R**anging), camera. Laser range sensors provide information about the distance from the camera of objects in the workspace. This range data, also known as $2\frac{1}{2}$ -D data, is important for current approaches to mobile navigation [1, 2, 3] and/or object manipulation [4]. The output of a laser range camera, however, is an image in which each pixel value is related to the distance from the camera to a corresponding scene point. Figure 1.1 shows a range image in which close objects are dark and far away objects are light. Interpretation of this raw image by even skilled operators is difficult.

To simplify the task of interpreting range data, a three-dimensional world model can be constructed by combining multiple range images of a scene acquired from



Figure 1.1: *Raw LADAR range image of a scene containing various pipes and valves. Range is shown as a gray value from black to white representing distances that are near and far, respectively.*

various viewpoints. Figure 1.2 shows an example of this type of 3-D world model created solely from range data by using level-set theory [5]. This world model alone is easier for humans to interpret than a raw range image, but it is still incomplete. This model relates only the relative physical geometry of the scene. No other sensory information is included, so the operator cannot use this type of model to directly deduce such properties as color, temperature, or radiation level for objects in the scene. If other sensory data is combined with a 3-D geometric world model, certain tasks become much easier for execution via tele-operation. For example, a color model of a robot's environment is very useful for object recognition, while a 3-D model combined with infrared or gamma imagery aids in the detection and localization of leaks in pressure vessels. Thus, the addition of information from these sensors creates a model of a workspace environment that is more complete, and thus easier

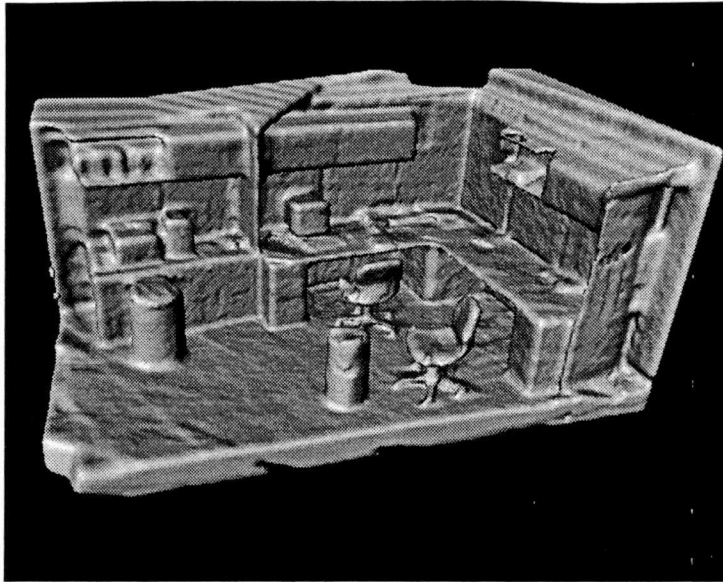


Figure 1.2: *A three-dimensional world model created from multiple range images taken of the scene from different viewpoints. No additional information, such as color, has been included as a texture map.*

to interpret, than a model created from range data alone.

Sensors such as color, infrared, and gamma cameras are often employed without any range information in many human-driven tasks. However, the image from each sensor must be displayed in separate windows. This type of interface forces the operator to interpret multiple images that provide information from different viewpoints as Figure 1.3 shows. Locating and identifying the same object in each image can be difficult. In addition, these images contain little geometric information about unknown scenes, whereas LADAR images contain information about scene geometry, but little else.

The availability of range information allows the images acquired from the various projective sensors to be integrated into one model that is viewable from any desired direction. In our sensor fusion approach, images acquired using the various 2-D

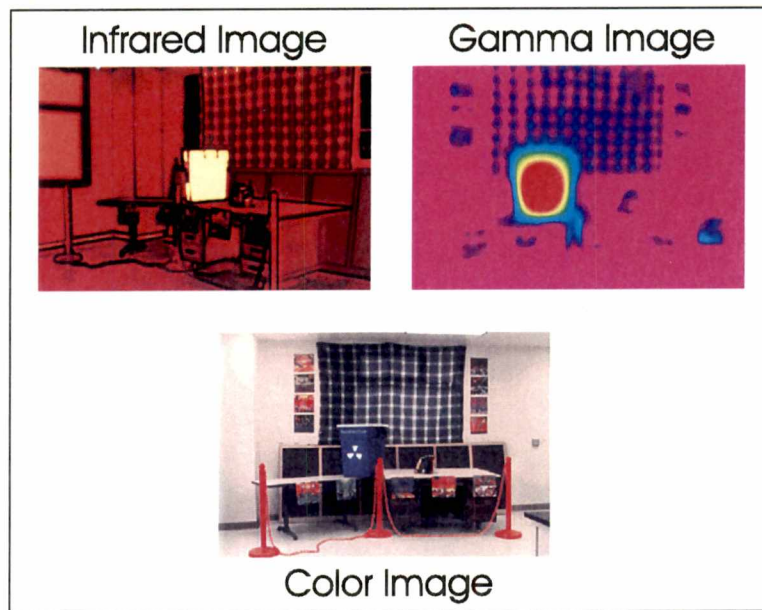


Figure 1.3: *Color, infrared, and gamma images of the same scene, but taken from different locations. Without accompanying range information, no common frame of reference can be established, and each image must be viewed in a separate window.*

sensors are projected onto the 3-D world model as requested by the operator. The model thus serves as a backdrop for displaying multi-sensory imagery. This projection is accomplished through the use of a computer graphics technique called *texture mapping*. Operators can therefore choose the type of sensor data they wish to view and overlay its corresponding texture map on the geometric world model as illustrated in Figure 1.4.

The major difficulty in constructing world models from multiple sensors arises from the physical necessity for each sensor to be in a different location, and thus return information about different portions of the scene. In order for images obtained from other sensors to be used as texture maps, they must be aligned, or registered, with the range image. In other words, a mapping must be computed between the range data and the image planes of the various projective sensors. Once this reg-

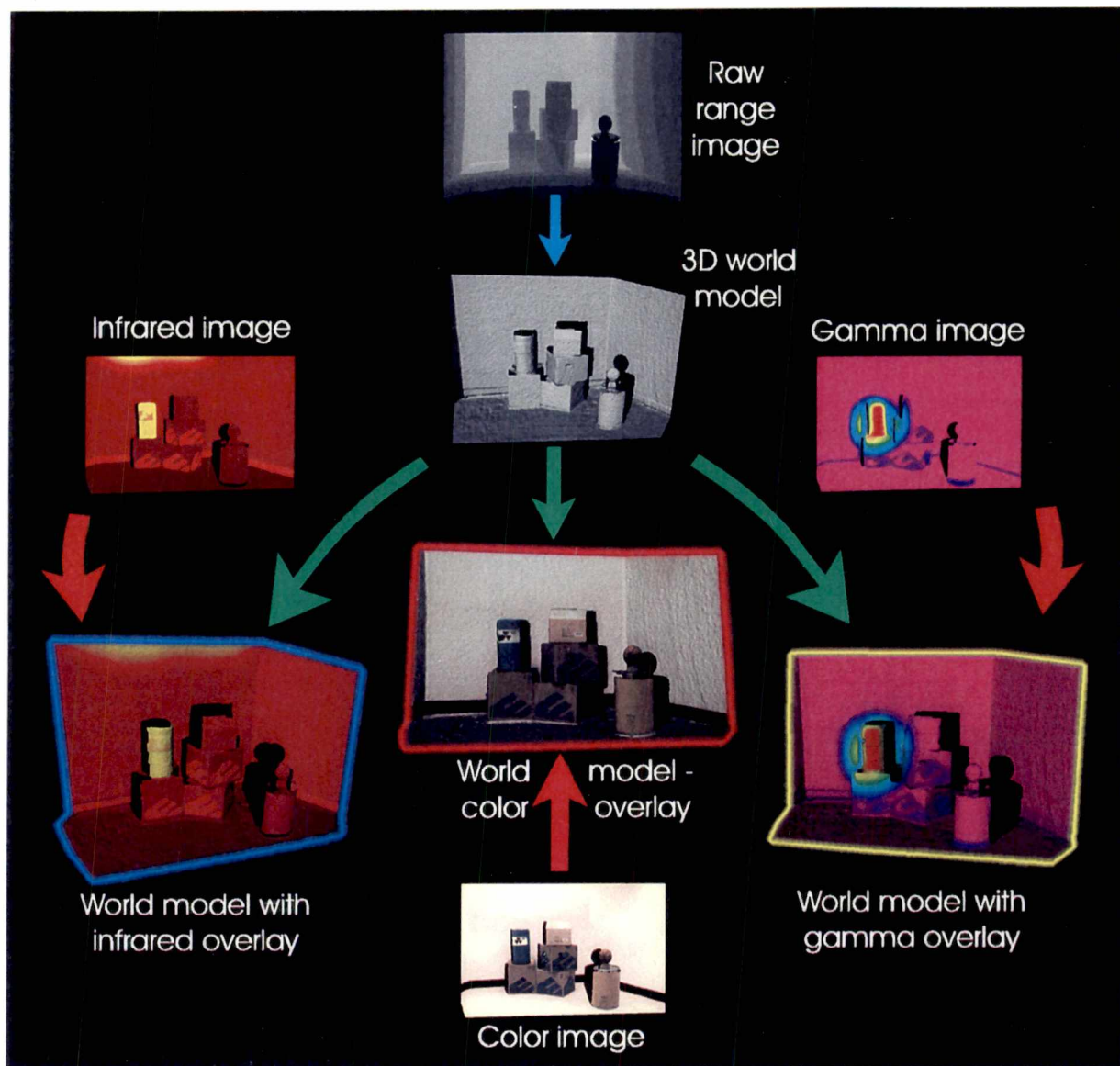


Figure 1.4: *Multi-sensor visualization scheme which describes how information from color, infrared, and radiation sensors is combined with laser range data to create complete 3-D models that are easily interpretable.*

istration is accomplished, either through geometric or other methods, the texture mapping technique we describe can be readily employed for sensor integration.

In this paper, a novel stereo-based technique for registering color and laser range images acquired using externally uncalibrated sensors is presented. Taking advantage of the intensity image returned by many commercial LADAR systems, our algorithm employs a method to identify corresponding feature points in the intensity and color images without epipolar constraints. Using this matched point set, the relative orientation between the LADAR and color cameras is computed by minimizing the difference between range estimates from the laser range camera and those computed using stereoscopic techniques. The recovered epipolar geometry and laser range data are then used to reproject the color image to appear as if it was acquired from the LADAR sensor's point of view, producing a registered color/range pair.

The remainder of this section is devoted to discussion of research performed by others that is relevant to our own research. Section 2 describes the steps of the registration process in detail, including coordinate transforms, feature correspondence, epipolar geometry, and image reprojection. Section 3 shows the results of the registration process applied to both synthetic and real data. Section 4 is a discussion of the results. Section 5 describes additional work to be performed in the future, and our conclusions are discussed in Section 6.

1.1 Related Work

Because of the increased availability of low-cost, high-speed laser range scanning systems, research and applications involving the use of range imagery is becoming increasingly common. An often encountered problem when using range data is the need to register range images of different portions of a scene. For example, the creation of complete 3-D models from $2\frac{1}{2}$ -D images usually requires registered data sets. Several different techniques, including least-squares minimization [6, 7, 8, 9], iterative-closest-point(ICP) techniques [10, 11], other correspondenceless methods [12, 13, 14], and various non-linear methods [15, 16, 17] have been proposed as solutions to this problem.

Least-squares minimization, proposed by Arun, Huang, and Blostein [6], registers a pair of range images by finding point correspondences in the images and calculating the geometric or affine mapping that produces the smallest amount of error between each corresponding point pair in a least-squares sense. The iterative-closest-point method, as described by Besl and McKay [10], registers a range image pair without the need to find point correspondences beforehand. Instead, the correspondence problem is incorporated into the ICP minimization algorithm, which produces the point correspondences and the mapping between the two images as its output. While both the least-squares method and ICP perform adequately, they each require two sets of three-dimensional data. Since our objective is to register a range image with a color image, however, we have depth data for only one of our images. Because of this, our registration technique uses neither least-squares minimization nor ICP.

Instead, our registration approach uses a stereo-based method to determine the correct mapping function between our images.

Other existing method of registering images employ traditional camera calibration techniques [18, 19, 20, 21]. Calibration techniques of this type are quite numerous, and most require only limited explicit range information, usually in the form of a specially fabricated and positioned target. Once the calibration process is performed, the cameras are very rigidly mounted to prevent any changes in the cameras' positions, which would require the calibration process to be repeated. The tedious measurements involved and the inflexible nature of this registration technique make it unsuitable for our purpose. We desire a system that is self-calibrating, since the cameras are likely to be operated in remote and/or hazardous environments. Also, in our case, the range information for one image is readily available. This leads to the desire for a registration technique that is a compromise between traditional camera calibration methods that require no range and techniques such as least-squares minimization and ICP that require two sets of range data.

Since we do not have two range images, it is necessary to extract features in order to use our registration approach. Furthermore, each located feature must exist in both images, forming a correspondence pair. A large amount of research has been dedicated to the area of image matching, resulting in many different correspondence techniques. These include high-level feature extraction techniques such as shape detection [22, 23, 24, 25, 26], edge detection [27, 28, 29, 30, 31], and corner detection [32], low-level area-based techniques such as correlation [33, 34, 35] and cepstral filtering [36], and manual correspondence. Manual point-matching is a common correspon-

dence method, involving an operator who must visually choose corresponding points in each image. Not only is this method time-consuming and tedious for the operator, but it suffers from unavoidable human error. An automated correspondence method, using either high-level or low-level matching, is much faster, less cumbersome to implement, and gives pixel-accurate correspondences. Our registration technique takes this approach, relying on a combination of feature extraction and area-based matching using corner detection and a form of correlation respectively. This combination of techniques is necessary, since neither one alone will produce the required point correspondences. Feature matching requires geometric constraints, which are not available, and higher level features, such as squares or conics, cannot be used since they might not exist in an arbitrary scene. Area-based methods are also undesirable due to differences in the gray levels of the color and LADAR images, for the same objects.

CHAPTER 2

Registration of LADAR and Projective Imagery

To register a color image with a range image using our stereo-based technique, four steps must be performed. First, the range image is converted from its raw form into a perspective projection so that traditional stereo-based techniques can be used to calculate range. Second, points in each image that correspond to the same point in 3-D space must be found. Third, the relative rotation and translation, $(\mathbf{R}, \vec{T})$, between the two cameras is calculated based on the two sets of corresponding points. Finally, the color image is reprojected using the rotation and translation parameters and the range data to appear as it would from the point of view of the laser range scanner. The registration process is illustrated in Figure 2.1. The result, a color image that is registered with the LADAR image, is then used as a texture map for a model generated from the range data.

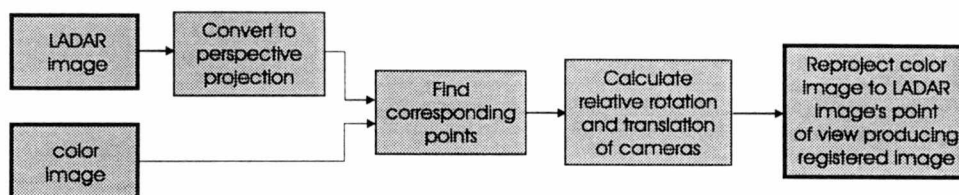


Figure 2.1: *Flowchart describing the entire registration process.*

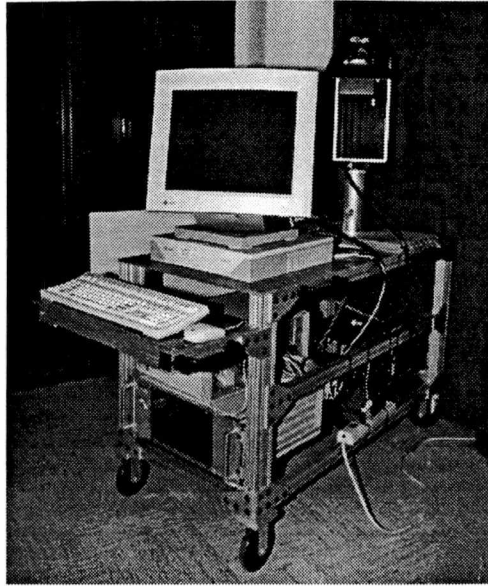


Figure 2.2: *The Perceptron P5000 laser range camera and custom-built mobile scanning system used to acquire range and intensity images.*

2.1 The Perceptron

The laser range scanner used for our work is the Perceptron P5000 by Perceptron, Inc. [37]. It is shown in Figure 2.2 mounted on a mobile scanning cart. This custom platform consists of a pan-tilt motor system and a workstation that controls the scanning direction, receives the range data from the Perceptron, and stores it on disk.

The Perceptron uses two rotating mirrors to deflect an amplitude modulated laser beam when scanning a scene. More specifically, the Perceptron is an azimuth-elevation scanner [38], meaning that the laser beam undergoes a horizontal deflection from the first mirror and a vertical deflection from the second mirror as in Figure 2.3. When scanning, the laser beam moves row by row through the scene, returning a range-related measurement called ‘counts’ that is based on the phase difference

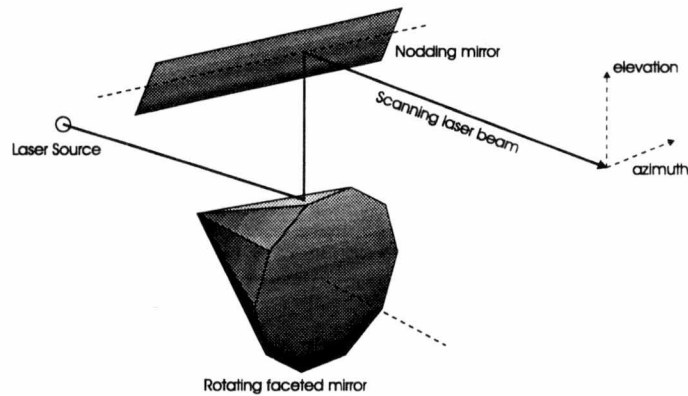
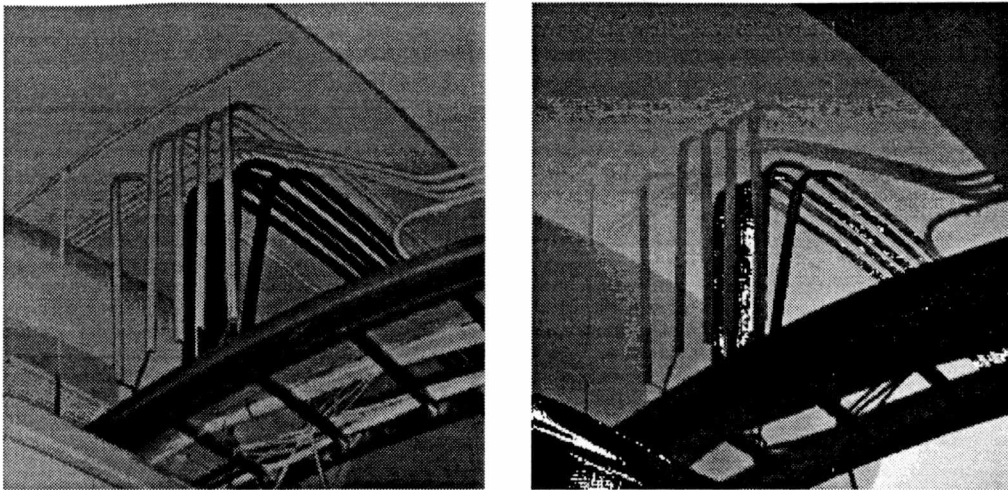


Figure 2.3: *The azimuth-elevation mirror system used in the Perceptron.*

between the outgoing and returned signal. This raw range data must then be transformed into both a camera-centered spherical coordinate system and a Cartesian reference frame.

In addition to range information, the Perceptron also produces an image based on the intensity of the returned laser beam. This image, which looks similar to a black and white photograph, is perfectly registered with the range image as Figure 2.4 illustrates. It is this intensity image that is used for determining point correspondences in our registration technique. For later use by the feature correspondence section of our algorithm, the intensity image must be transformed from the distorted azimuth-elevation coordinates into a perspective projection. The following section describes the conversion of the LADAR images from their raw forms into Cartesian coordinates, spherical coordinates, and finally a perspective projection.



(a)

(b)

Figure 2.4: (a) A Perceptron intensity image of a typical scene containing various pipes. Intensity values are calculated from the amplitudes of the laser beam returning to the Perceptron. (b) The corresponding Perceptron range image, computed using the phase of the returned laser beam.

2.2 Transformation from LADAR to Euclidean and Projective Spaces

In order to use range data acquired from a typical range scanner, it is necessary to derive equations to transform the unprocessed range information from the laser range scanner into various formats. The raw data is simply a long list of numerical values that are related to the actual range and intensity values. The radial direction of each range measurement is not explicitly stated, but is calculated from the position of the measurement in the list. For our registration technique, the raw range data must be converted into Cartesian coordinates, and the intensity data must be converted into a perspective projection. Figure 2.5 gives an overview of this process.

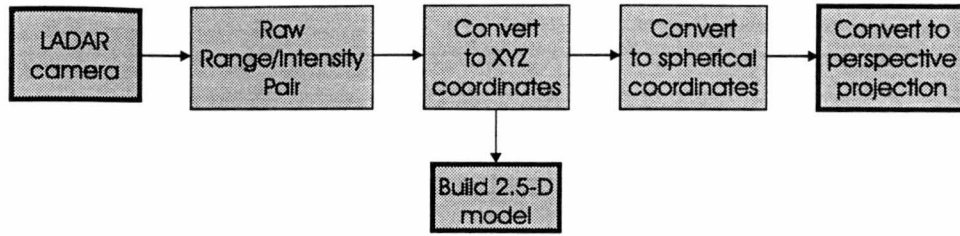


Figure 2.5: *Flowchart describing the process of converting raw LADAR images to both Cartesian coordinates for the purpose of creating $2\frac{1}{2}$ -D scene models and projective images necessary for the registration algorithm.*

2.2.1 Transformation from LADAR Coordinate System to Cartesian Coordinate System

The first step in converting the raw 'counts' data to Cartesian coordinates is to convert the numerical values returned by the Perceptron into actual range values from the scanner. This is done with the following equation:

$$R = r_0 + nr,$$

where R is the range value, r_0 is the standoff distance, n is the number of counts ranging from 0 to 4095, and r is the range increment per count. The azimuth and elevation angles, α and β , of a single range measurement are related to its row, i , and column, j , in the range image by:

The equations relating α and β to i and j are:

$$\alpha = \frac{\left(\frac{w}{2} - j\right)}{r_\alpha}$$

$$\beta = \frac{\left(\frac{h}{2} - i\right)}{r_\beta},$$

where w is the width of the range image in pixels, h is the height of the range image

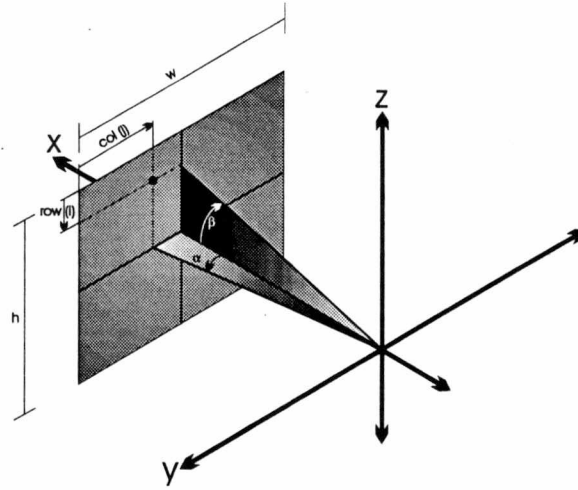


Figure 2.6: *The coordinate axis system and scanning angles (azimuth-elevation) used by the Perceptron in the acquisition of range images.*

in pixels, r_α is the horizontal angular resolution of the range scanner in degrees per pixel, and r_β is the vertical angular resolution of the range scanner in degrees per pixel. Figure 2.6 demonstrates this relationship.

The transformation from (r, α, β) coordinates to (X_L, Y_L, Z_L) Euclidean coordinates is accomplished by performing two rotations. First, each image point P_0 is rotated about the z-axis by its azimuth angle, α , and then about the y-axis by its elevation angle, β , giving the coordinates of the range point in Cartesian coordinates, P_2 . These two rotations are illustrated in Figure 2.7. The Cartesian world coordinates, $[X_L \ Y_L \ Z_L]^T$, resulting from the two rotations are given by:

$$\begin{aligned}
P_2 &= R_\beta R_\alpha P_0 \\
P_2 &= \begin{bmatrix} \cos \beta & 0 & -\sin \beta \\ 0 & 1 & 0 \\ \sin \beta & 0 & \cos \beta \end{bmatrix} \begin{bmatrix} \cos \alpha & -\sin \alpha & 0 \\ \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r \\ 0 \\ 0 \end{bmatrix} \\
P_2 &= \begin{bmatrix} \cos \beta & 0 & -\sin \beta \\ 0 & 1 & 0 \\ \sin \beta & 0 & \cos \beta \end{bmatrix} \begin{bmatrix} r \cos \alpha \\ r \sin \alpha \\ 0 \end{bmatrix} \\
P_2 &= \begin{bmatrix} r \cos \alpha \cos \beta \\ r \sin \alpha \\ r \cos \alpha \sin \beta \end{bmatrix}. \tag{2.1}
\end{aligned}$$

Figure 2.8 contains an example of a range image and its corresponding model computed using Equation 2.1.

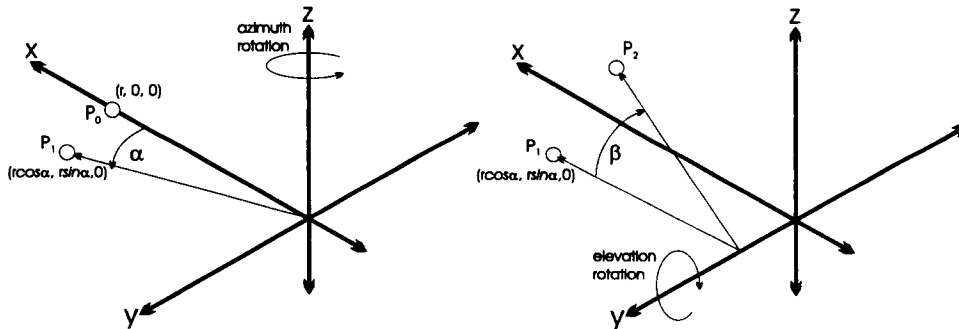
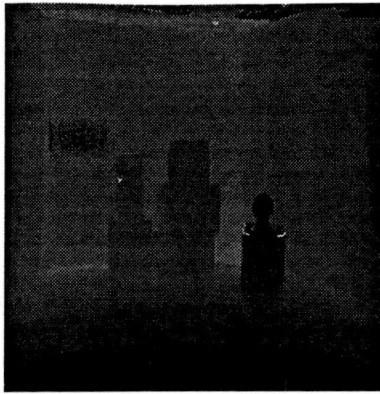
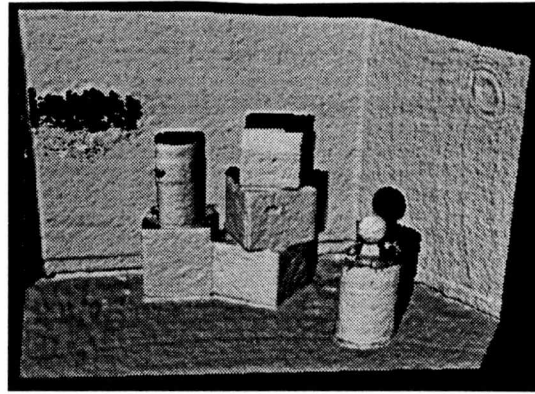


Figure 2.7: The sequence of azimuth and elevation rotations needed to calculate Cartesian coordinates from the raw range information, (r, α, β) .



(a)



(b)

Figure 2.8: (a) A raw Perceptron range image of a scene containing objects of various shapes, such as planes, cylinders, and a sphere. (b) The $2\frac{1}{2}$ -D model created from the range information after conversion to Cartesian coordinates.

2.2.2 Transformation from LADAR Coordinate System to Perspective Projection

To simplify the search for corresponding features, the LADAR intensity image is transformed to create a rectilinear image, similar to that produced by the color camera. The transformation requires (x_l, y_l) , the perspective image plane coordinates of a given intensity point, to be solved in terms of (α, β) , the known azimuth and elevation angles for a given point. However, mapping each pixel from the actual intensity image into a perspective space would leave regions in the transformed image with undefined gray levels. To ensure that each pixel in the perspective image contains an intensity value, back-projection is used to determine each perspective pixel's corresponding position in the intensity image. The intensity value at this point in the actual intensity image is then placed in the perspective pixel location. Since the calculated pixel location in the intensity image is usually non-integer, bilinear interpolation is used to calculate the desired perspective intensity value.

To perform this back-projection, the expressions that relate (α, β) as functions of (x_l, y_l) must be determined. To derive these equations, Cartesian coordinates, (X_L, Y_L, Z_L) , expressed as functions of (α, β) are perspective projected to give their corresponding image plane coordinates, (x_l, y_l) :

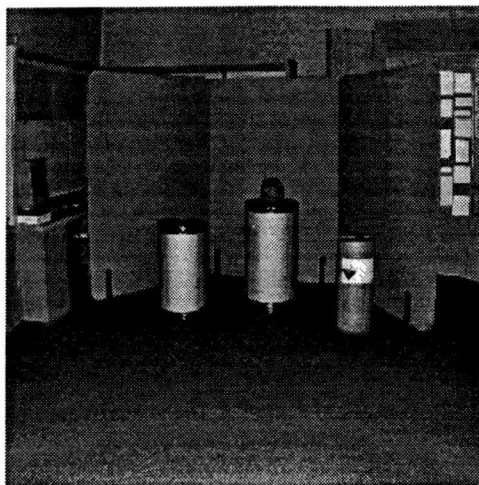
$$\begin{aligned} x_l &= -\lambda \frac{Y_L}{X_L} \\ x_l &= -\lambda \frac{r \sin \alpha}{r \cos \alpha \cos \beta} \\ x_l &= -\lambda \frac{\tan \alpha}{\cos \beta} \end{aligned} \tag{2.2}$$

$$\begin{aligned} y_l &= \lambda \frac{Z_L}{X_L} \\ y_l &= \lambda \frac{r \cos \alpha \sin \beta}{r \cos \alpha \cos \beta} \\ y_l &= \lambda \tan \beta, \end{aligned} \tag{2.3}$$

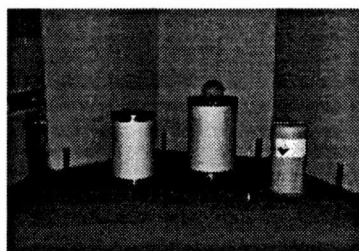
where λ is the arbitrary focal length of the reprojected image, chosen to match the focal length of the color camera. Solving Eqs. 2.2 and 2.3 in reverse yields the desired relation:

$$\alpha = \tan^{-1} \left(\frac{-x_l}{\sqrt{y_l^2 + \lambda^2}} \right) \tag{2.4}$$

$$\beta = \tan^{-1} \left(\frac{y_l}{\lambda} \right). \tag{2.5}$$



(a)



(b)

Figure 2.9: (a) A raw *Perceptron* intensity image of a scene containing objects with planar and cylindrical surfaces. (b) Perspective image projection of the intensity image, matched to the field-of-view of color images acquired from the Kodak DCS 460 digital camera equipped with a 35mm lens.

Equations 2.4 and 2.5 are used to transform the *Perceptron* intensity images into perspective projections. For each point in the perspective image, the range image azimuth and elevation angles are determined, and the corresponding intensity value is used in the perspective image. Figure 2.9 shows an example of a raw intensity image and its perspective projection. Note that only a small portion of the original intensity image was transformed to form the projective image. The size of the reprojected image shown here was chosen to match the field-of-view of our color camera. However, the entire image could be reprojected to match cameras with varying fields-of-view.

2.2.3 Generating Panoramic Range Images

Previously, equations for transforming range images from azimuth-elevation to Cartesian coordinates were determined. From these equations, spherical coordinates can easily be determined to generate panoramic range images. Panoramic images are useful because of their very large field-of-view, which would allow an entire room to be seen, for example.

These panoramic images are created by merging range images acquired by rotating the scanner around a fixed camera coordinate center. The Perceptron-equipped mobile scanning system used for our experiments (See Figure 2.2) can be programmed to pan and tilt, acquiring range images periodically over a longitudinal range of 360° and a latitudinal range of approximately 160° due to a limitation imposed by the camera mounting hardware. This process of compositing, or mosaicing, images has been performed on color perspective images in the past [39]. Szeliski [40] [41] has performed this type of image mosaicing using local registration techniques to find the transformations between the images. Registration of the acquired range images is not required by our system, however, since the scanning direction of each image is returned by motor encoders.

To build a panoramic image using our system, each range point is converted from its local (α, β) coordinates to a global (α, β) coordinate system by adding the azimuth and elevation angles of the pan/tilt system to α and β . The next step is to derive equations relating spherical coordinates, (θ_s, ϕ_s) , to the global azimuth-elevation coordinates, (α, β) :

$$\begin{aligned}
\theta_s &= \tan^{-1} \left(\frac{Y_L}{X_L} \right) \\
&= \tan^{-1} \left(\frac{r \sin \alpha}{r \cos \alpha \cos \beta} \right) \\
&= \tan^{-1} \left(\frac{\tan \alpha}{\cos \beta} \right)
\end{aligned} \tag{2.6}$$

$$\begin{aligned}
r \cos \phi_s &= Z_L \\
r \cos \phi_s &= r \cos \alpha \sin \beta \\
\phi_s &= \cos^{-1}(\cos \alpha \sin \beta).
\end{aligned} \tag{2.7}$$

Solving Equations 2.6 and 2.7 in terms of α and β yields

$$\alpha = \cos^{-1} \sqrt{\frac{\cos^2 \phi_s \tan^2 \theta_s + 1}{\tan^2 \theta_s + 1}} \tag{2.8}$$

$$\beta = \sin^{-1} \left(\frac{\cos \phi_s}{\cos \alpha} \right). \tag{2.9}$$

Equations 2.8 and 2.9 are then used to convert each acquired range image to spherical coordinates. Plotting the range points with θ_s as the horizontal axis and ϕ_s as the vertical axis results in a panoramic range image. Figure 2.10 shows images of a computer lab converted to spherical coordinates. Figure 2.11 shows another panoramic intensity/range pair, this time of a building lobby.

While these panoramic images contain a vast amount of geometric information, they also require vast computer resources for storage and manipulation. Each panoramic image is approximately 7000×4000 pixels. Building a 3-D model at this

resolution would require approximately 15 gigabytes of memory! Due to the extreme computational requirements, we are currently limited to working with single range/color image pairs, but the construction of sensory-integrated panoramic models in the future remains an intriguing possibility.

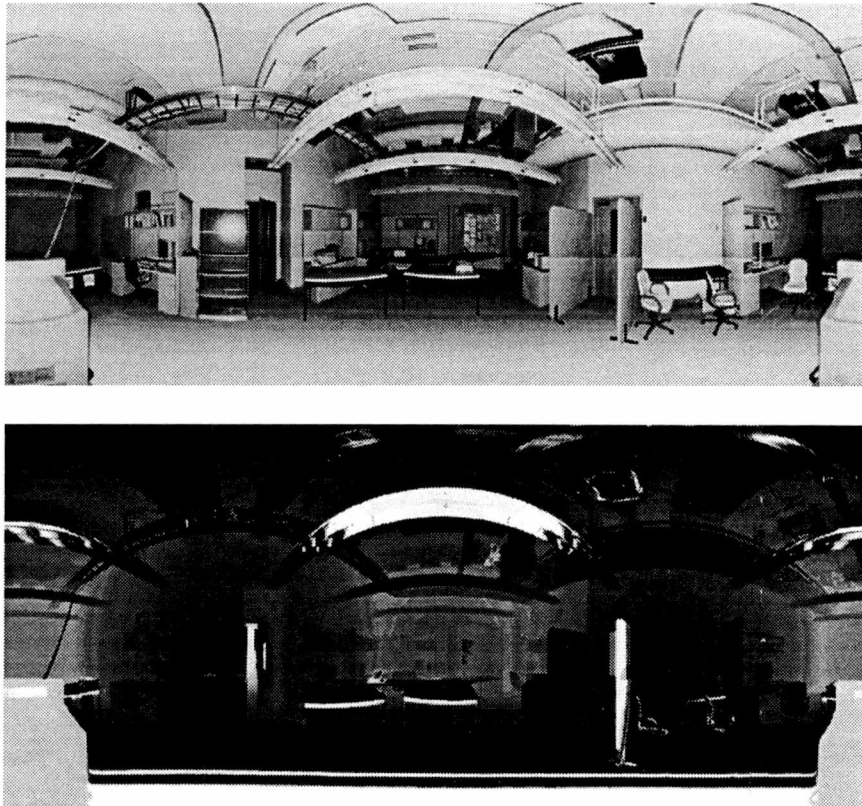


Figure 2.10: *Panoramic intensity/range image pair of computer lab after transformation to spherical coordinates.*

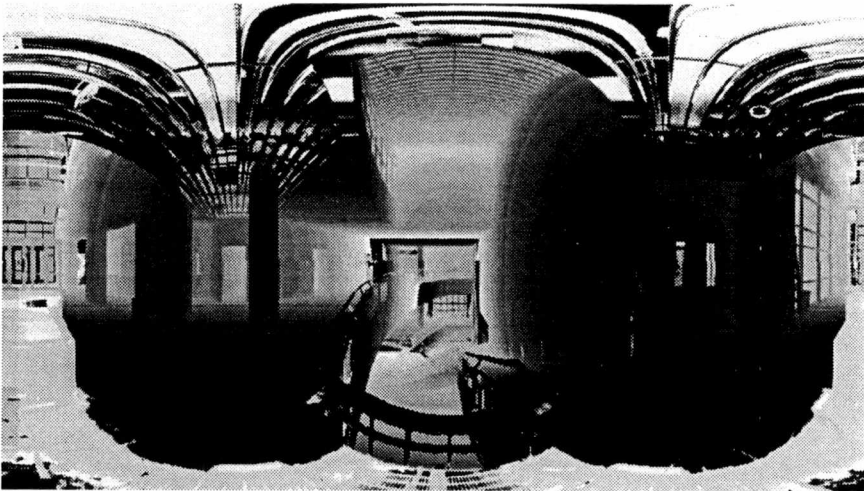
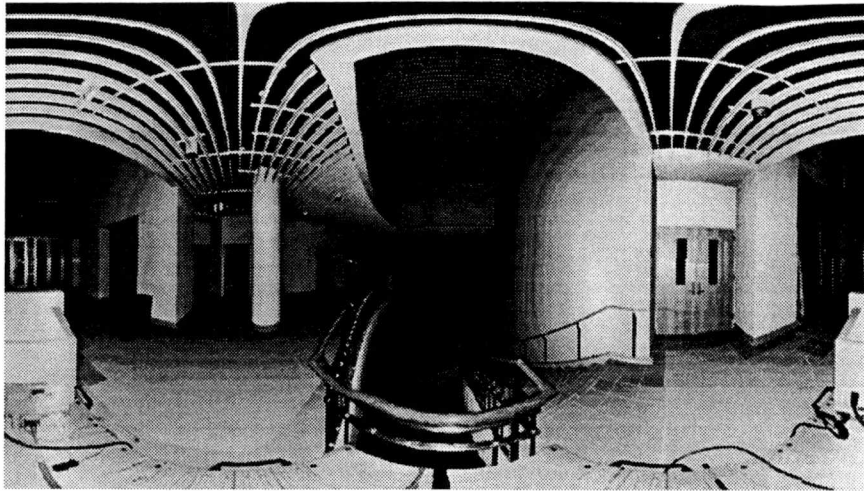
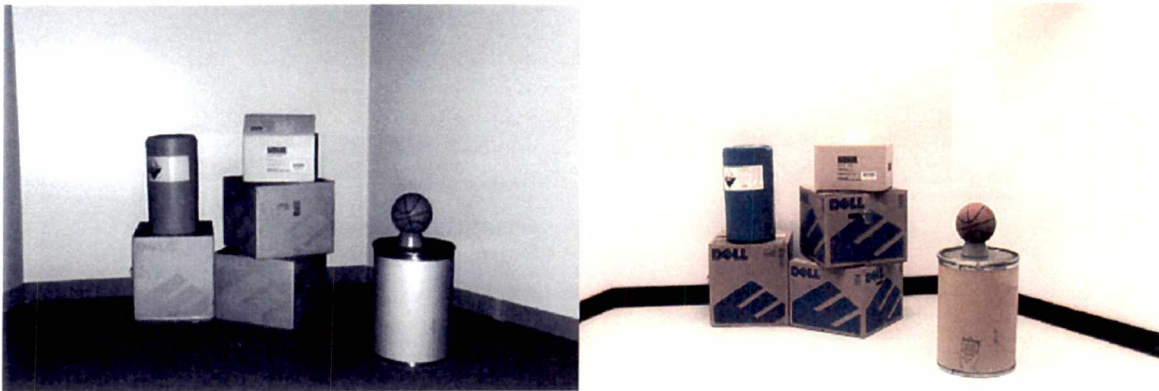


Figure 2.11: *Panoramic intensity/range image pair of building lobby after transformation to spherical coordinates.*

2.3 Feature Correspondence

Once the LADAR intensity image has been transformed into a projective space, the location of corresponding features in the intensity and color images is determined. Figure 2.12 shows an intensity image and a color image of the same scene. Notice that, since both images types are formed using similar mechanisms, the locations of many objects are identifiable in each image. However, there are three main differences in image formation between the two sensors: 1) the illumination in the intensity image comes solely from the laser range scanner instead of the natural illumination of the scene, 2) the wavelength of the illumination is that of the laser, which affects the apparent intensity of different colored objects, and 3) the reflective surfaces and surfaces oriented so that the laser beam strikes obliquely create noise by preventing the laser beam from returning to the LADAR scanner.

While these differences in illumination cause variations in gray level neighbor-



(a)

(b)

Figure 2.12: (a) *Perspective-projected intensity image of the 'boxes' scene.* (b) *Color image of the same scene acquired with the Kodak DCS 460 digital camera equipped with a 35mm lens, placed on tripod near the LADAR camera.*

hoods, many of the edges and corners resulting from depth discontinuities and object color differences are visible in both images, making feature correspondence possible. Unfortunately, the gray level differences are significant enough in the neighborhoods of these features to prevent existing matching techniques from performing adequately without geometric constraints. The automated matching technique we have developed overcomes this problem and is able to locate corresponding features in the images using a hybrid feature/area based approach. Our system also allows the process of finding corresponding features between two images to be performed manually. Figure 2.13 illustrates our feature correspondence procedure.

2.3.1 Manual Feature Correspondence

Manually locating corresponding points in the intensity image and the color image produces sets of matching points that are free from gross errors in correspondence. To

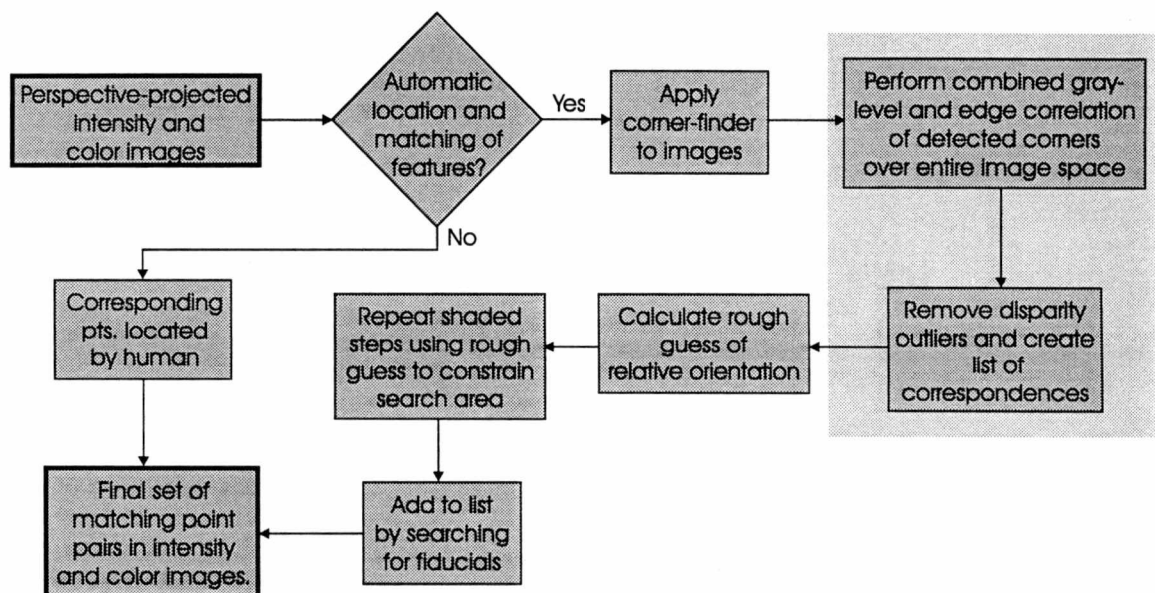


Figure 2.13: Flowchart describing the feature correspondence process using both manual and automatic methods.

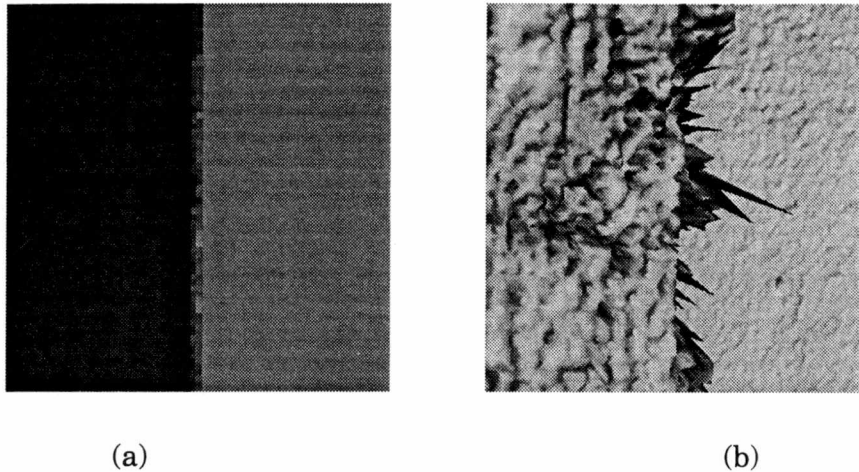
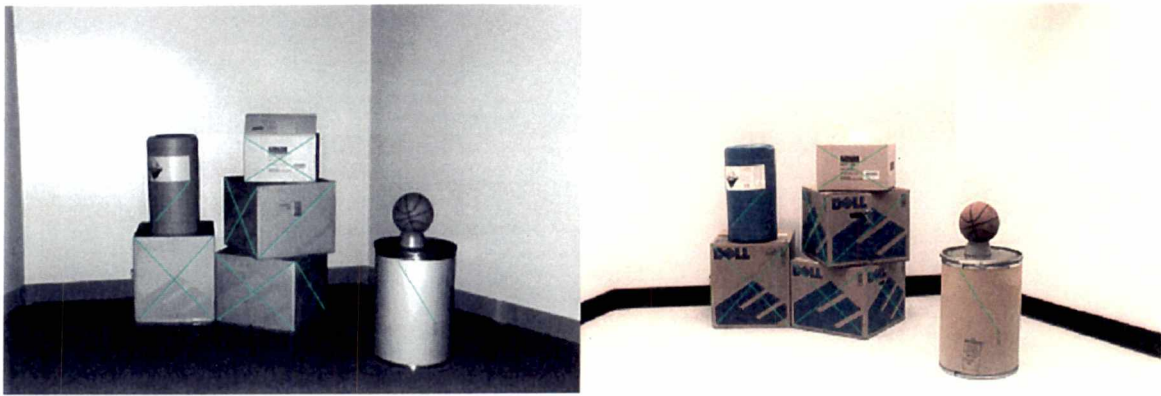


Figure 2.14: (a) A portion of a range image containing a sharp step edge. (b) A 3-D model created from the range image showing noisy data at the edge that is not easily visible in the range image.

obtain enough corresponding points, matching lines are used instead of single points. Lines are drawn between opposite corners of planar (or nearly planar) surfaces where the range data is most accurate, unlike at the edges where the range values are less accurate as shown in Figure 2.14. Several point correspondences are found by taking points along each line at constant intervals. As Figure 2.15 demonstrates, only a few lines are necessary to obtain enough points to compute the system's epipolar geometry.

Two problems arise from finding point correspondences manually. First, the process is very time-consuming and tedious for the person that is performing the point matching. The vast majority of the time spent registering a single pair of images can be spent on this step alone. The second problem caused by manual point correspondence is the error incurred by the use of a human in the process. Even the most careful or well-trained operator is unlikely to be able to locate and mark all corresponding points with pixel accuracy.



(a)

(b)

Figure 2.15: (a) *Perceptron intensity image of 'boxes' scene with manually drawn lines used for feature correspondence.* (b) *The accompanying color image with corresponding manually drawn lines.*

2.3.2 Automatic Feature Correspondence

Matching features between an intensity image and a color image automatically is difficult since the large differences in gray levels for the same objects do not allow the direct application of area-based techniques such as correlation. Thus, identifiable features must first be located in each image in order to find corresponding points in the intensity and color images. One possible feature candidate is intensity edges. Although simple to locate, edges have the disadvantage of being the precise locations in the range image where the data is the least reliable (See Figure 2.14). Using such unreliable range information would be detrimental, since the range values are used to compute the registration. The best places in the range image to obtain accurate data are on smooth surfaces. By finding various corners on planar surfaces, as in the manual correspondence method, many accurate data points are recoverable.

Our corner finder is based on the one described in [32]. Given an image, $f(i, j)$, vertical and horizontal Sobel edge detectors [42] are first applied to the image:

$$f_v(i, j) = f(i, j) * S_v$$

$$f_h(i, j) = f(i, j) * S_h,$$

where $f_v(i, j)$ is the vertical gradient image of $f(i, j)$, $f_h(i, j)$ is the horizontal gradient image of $f(i, j)$, and S_v and S_h are the corresponding Sobel masks.

The second step is the calculation of the following four terms:

$$N_{11}(i, j) = \sum_{i_0=i-2}^{i+2} \sum_{j_0=j-2}^{j+2} (f_v^2(i_0, j_0))$$

$$N_{12}(i, j) = \sum_{i_0=i-2}^{i+2} \sum_{j_0=j-2}^{j+2} (f_v(i_0, j_0) \cdot f_h(i_0, j_0))$$

$$N_{21}(i, j) = \sum_{i_0=i-2}^{i+2} \sum_{j_0=j-2}^{j+2} (f_h(i_0, j_0) \cdot f_v(i_0, j_0))$$

$$N_{22}(i, j) = \sum_{i_0=i-2}^{i+2} \sum_{j_0=j-2}^{j+2} (f_h^2(i_0, j_0)),$$

where $f_v^2(i, j)$ is the element-wise square of $f_v(i, j)$, $f_v(i, j) \cdot f_h(i, j)$ is the element-wise product of $f_v(i, j)$ and $f_h(i, j)$, and $f_h^2(i, j)$ is the element-wise square of $f_h(i, j)$.

After the four terms are computed, values for $w(i, j)$ and $q(i, j)$ are found using:

$$w(i, j) = \frac{2\det(N)}{\text{trace}(N)}$$

$$q(i, j) = \frac{4\det(N)}{\text{trace}^2(N)},$$

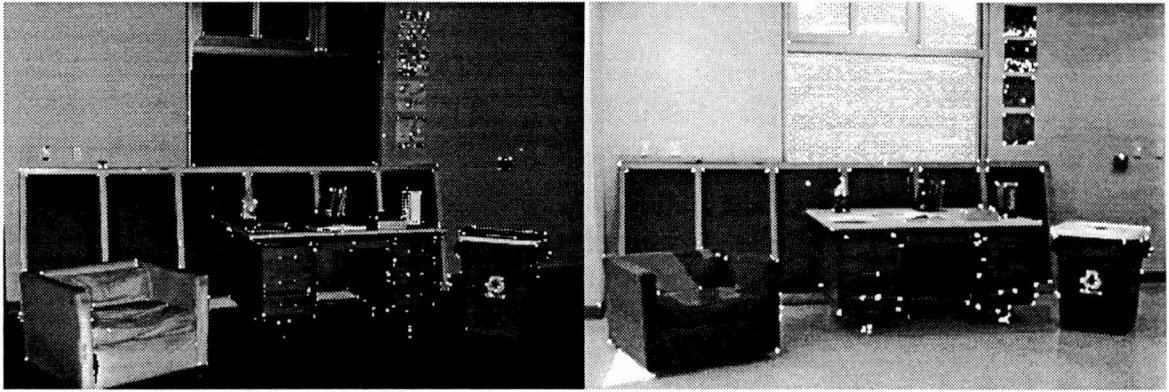
where $N = \begin{bmatrix} N_{11} & N_{12} \\ N_{21} & N_{22} \end{bmatrix}$.

The fourth step is to calculate $w^*(i, j)$:

$$w^*(i, j) = \begin{cases} w(i, j) & \text{if } w(i, j) > w_{min} \text{ and } q(i, j) > q_{min} \\ 0 & \text{otherwise} \end{cases},$$

where w_{min} and q_{min} are threshold values.

The final step in the corner detection process is to set all $w^*(i, j)$ to zero that are not relative maxima. Applying the entire corner-detection process to both the intensity and color images yields a distinct set of corner points for each. These points are labeled $P_l^i, i = 1...M$, for the laser intensity image and $P_c^j, j = 1...N$, for the color image. Applying the corner corner finder to our image pair, typically provides several hundred candidate points from each image as Figure 2.16 shows.



(a)

(b)

Figure 2.16: (a) Corners detected by the corner-finder algorithm in the 'desk' scene intensity image. (b) Corners detected in the accompanying color image of the scene.

After applying the corner detector, the normalized gray-level and edge correlation coefficients of each candidate corner point P_l^i with each point P_c^j are calculated and averaged to obtain m_{ij} :

$$m_{ij} = \frac{(G_l^i \circ G_c^j) + (E_l^i \circ E_c^j)}{2},$$

where G_l^i is a 15×15 gray-level neighborhood of the intensity image point P_l^i , G_c^j is a 15×15 gray-level neighborhood of the color image point P_c^j , and E_l^i and E_c^j are the edge-detector responses of G_l^i and G_c^j respectively.

For each point P_l^i , $C_1 = \max_j \{m_{ij}\}$ and $C_2 = \max_j \{m_{ij} \cap \{\bar{C}_1\}\}$ are calculated. The matching point, P_c^j , for a given P_l^i is taken to be the one for which C_1 is the greatest. Points with multiple possible matches — $\max_j \{m_{ij}\}$ is equivalent for two or more values of j — in the color image are eliminated by keeping only points with matches such that $(C_1 - C_2) > T_1$, where T_1 is set by the user.

While this process produces a list of possible corresponding features, a portion of these answers are grossly incorrect since no geometric matching constraints have been utilized. Because this list will be used by a least-squares algorithm to provide an initial estimate of the epipolar geometry, the list must be further reduced by a process we defined as disparity culling. This is accomplished by first calculating the disparity for each point pair as follows:

$$d_k = \sqrt{(x_l^k - x_c^k)^2 + (y_l^k - y_c^k)^2},$$

where d_k is the disparity of the k th correspondence pair, x_l^k is the x-coordinate of the k th intensity image point, x_c^k is the x-coordinate of the k th color image point, y_l^k is

the y-coordinate of the k th intensity image point, and y_c^k is the y-coordinate of the k th color image point.

Next, the following value is calculated for each point pair:

$$\Delta d_k = d_k - \text{median}(\{d_1, d_2, \dots, d_N\}),$$

where N is the number of point correspondences in the initial list. Finally, all point correspondences for which $\Delta d_k > T_2$, where T_2 is user-specified, are eliminated from the registration process. At this point, only a relatively small number of matches remain, but they are nearly all correct matches.

Applying the least-squares method described in the following section to this set of points yields a rough guess of the rotation and translation parameters. This rough geometric information is then used to constrain the search area. This is done by using the initial guess of $(\mathbf{R}, \vec{T})$, in combination with the range data, to approximate the location of the corresponding point in the color image. The previously described correspondence process is then repeated, only this time using a small window centered around the expected locations to constrain the search for possible matches. The constrained search provides many more correct matches than the first search does, since there are far fewer points to correlate with.

The disparity culling procedure is listed as follows:

Step 1 Perform unconstrained search

Step 2 Calculate disparity of each corresponding point pair

Step 3 Calculate median of these disparities

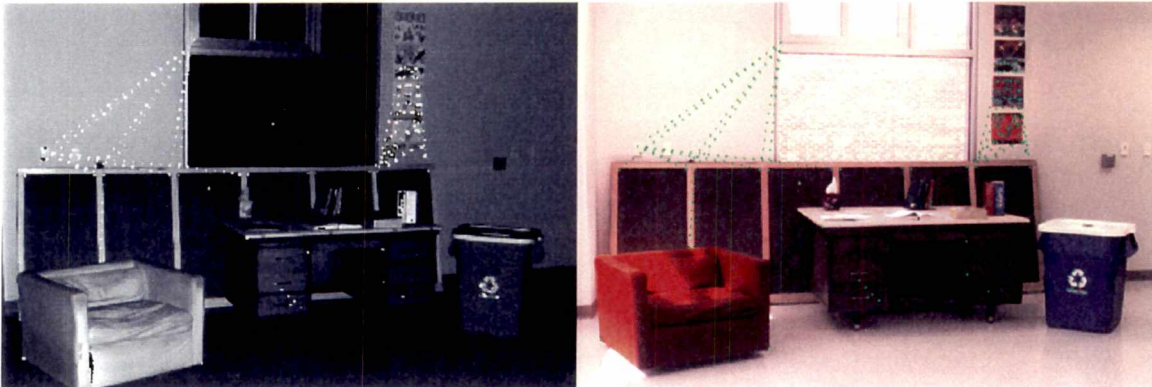
Step 4 Subtract the median from each disparity value

Step 5 Remove point pairs for which the result of Step 4 is greater than the threshold, T_2

Step 6 Calculate $(\mathbf{R}, \vec{T})$ with remaining point pairs using least-squares method

Step 7 Repeat Steps 1-5, only with the initial search space constrained by $(\mathbf{R}, \vec{T})$

To increase the number of point correspondences, lines are drawn between the different matched points. The lines are determined to lie on planar surfaces if the variance between the depths calculated from the equations of these lines and the actual range values is below a user-defined threshold value. By taking points at constant intervals along these lines, the final correspondence set is obtained as shown in Figure 2.17.



(a)

(b)

Figure 2.17: (a) Final corresponding point set resulting from applying the automatic point correspondence method to the 'desk' scene intensity image. (b) Matching points found in the 'desk' scene color image.

2.4 Determining the Epipolar Geometry of the LADAR-Color Camera System

Once the set of corresponding points between the two images is obtained, the rotation and translation, $(\mathbf{R}, \vec{T})$, between the color camera and the laser range scanner is computed. Two methods are described here: one linear and one nonlinear. The first technique is a linear least-squares solution to the problem and is used to calculate the initial guess of the solution that is required by the second non-linear algorithm. This non-linear algorithm takes the initial guess from the least-squares method and produces an accurate solution of $(\mathbf{R}, \vec{T})$. Only the relative extrinsic parameters of the system are estimated using this procedure. It is assumed that all intrinsic properties, such as focal length and scaling, are already known for both the LADAR scanner and the color camera. In our case, these parameters are reported by the digital camera. Focal length was not included in the calibration routine, because the value of the focal length is used to create the perspective-projected image from the LADAR image. An incorrect initial guess of the focal length would introduce scale variance into the feature matching process.

Figure 2.18 shows the steps involved in determining the relative orientation of the LADAR camera and the color camera. The linear least-squares technique is applied to the correspondence list obtained from the feature correspondence process discussed previously, and the stereo-depth of each point is calculated. The sum-squared-error of these depth values from the LADAR measured values is used to drive a non-linear estimation algorithm based on minimizing this error value. When

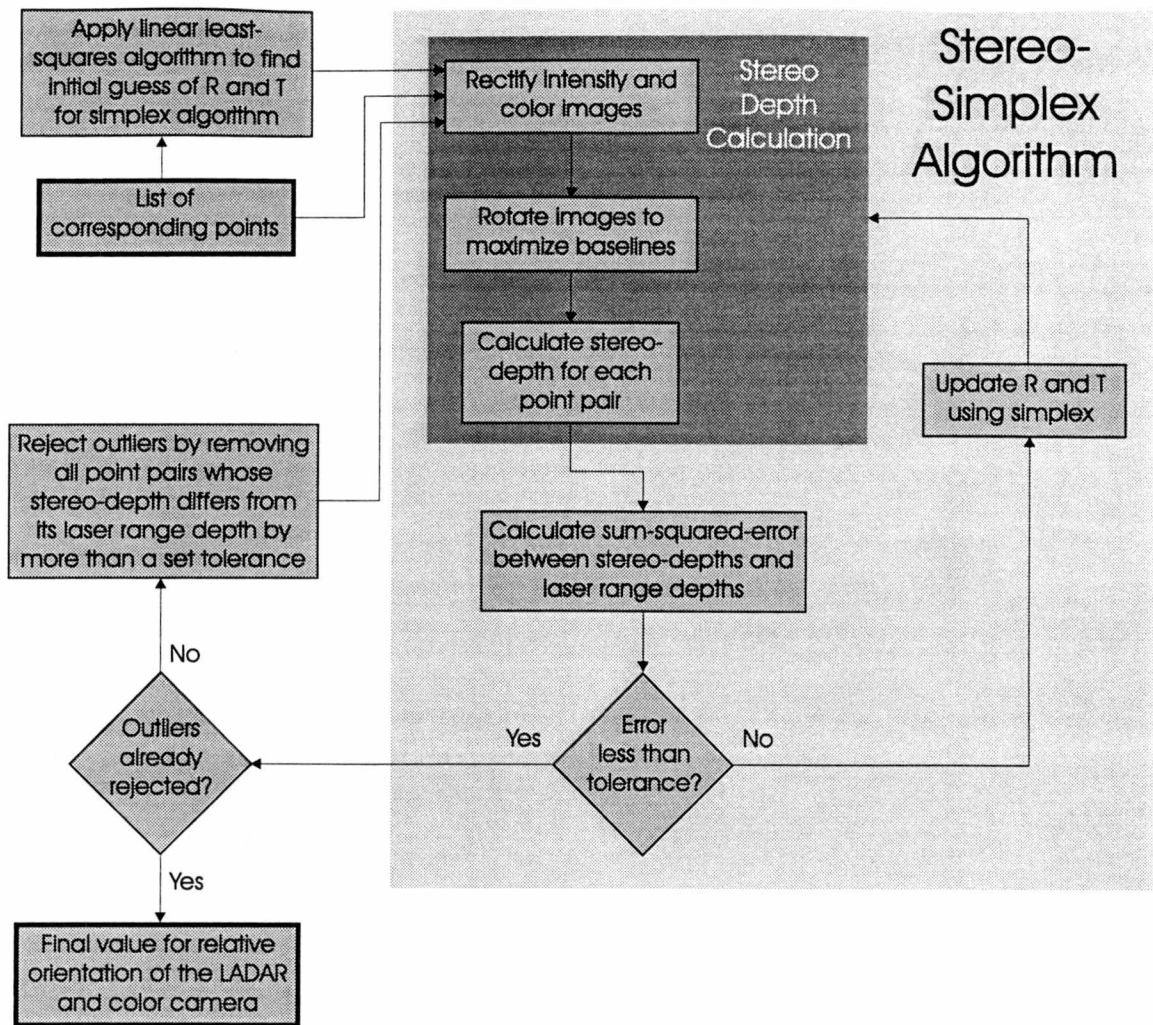


Figure 2.18: Flowchart describing the process of finding the relative orientation of the LADAR camera and the color camera.

the sum-squared-error falls below a certain value, the process ceases, and the final estimates of the relative extrinsic camera parameters, $(\mathbf{R}, \vec{T})$, are obtained.

2.4.1 Linear method (Least-Squares Algorithm)

As mentioned previously, a linear least-squares method is used in our registration technique to provide an initial guess of the rotation and translation parameters. This method was chosen partly because it is fast and simple, but mainly because it does

not require an initial guess.

Based on a modified version of Tsai's algorithm [9, 43], it begins with the following equation:

$$\begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \begin{bmatrix} (X_L)_k \\ (Y_L)_k \\ (Z_L)_k \end{bmatrix} + \begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix} = \begin{bmatrix} (X_C)_k \\ (Y_C)_k \\ (Z_C)_k \end{bmatrix}, \quad (2.10)$$

where r_{ij} is the ij th element of the rotation matrix, T_x , T_y , and T_z are the relative translation parameters, X_k^L , Y_k^L , and Z_k^L are the 3-D coordinates of the k th point in the laser range scanner's frame of reference, and X_k^C , Y_k^C , and Z_k^C are the 3-D coordinates of the k th point in the color camera's frame of reference.

Applying the perspective transform and simplifying yields:

$$(x_c)_k(r_{21}X_k^L + r_{22}Y_k^L + r_{23}Z_k^L + T_y) = (y_c)_k(r_{11}X_k^L + r_{12}Y_k^L + r_{13}Z_k^L + T_x),$$

where $(x_c)_k$ and $(y_c)_k$ are the image coordinates in the color camera of the k th point.

The above equation has eight unknowns, T_x , T_y , and r_{ij} , $i = 1, 2$, $j = 1 \dots 3$. These unknowns are recovered using singular value decomposition given eight pairs of noncoplanar corresponding points. The third row of the rotation matrix is then found by taking the cross-product of the first two rows, since $\mathbf{R}$ must be orthonormal. Even given these constraints, the rotation matrix calculated in this way is not generally orthonormal, so it is orthogonalized by using singular value decomposition to decompose the matrix into $\mathbf{U}$, $\mathbf{D}$, and $\mathbf{V}$. Replacing $\mathbf{D}$ with an identity matrix and recomposing yields an orthonormal estimate of the rotation matrix, $\mathbf{R}$. The final system parameter, T_z , is found by back-substitution into Equation 2.10 as follows:

$$T_z = \frac{\lambda(r_{11}X^L + r_{12}Y^L + r_{13}Z^L + T_x)}{x_c} - (r_{31}X^L + r_{32}Y^L + r_{33}Z^L).$$

The linear method of determining the rotation and translation parameters is very fast, simple to execute, and does not require an initial guess as the nonlinear method does. It is, however, sensitive to errors caused by noisy data or mismatches, and, thus, is only used to calculate an initial guess.

2.4.2 Nonlinear method (Simplex Algorithm)

To calculate more accurate estimates of the rotation and translation parameters, a non-linear minimization approach is employed. Starting with an initial guess of the rotation and translation parameters obtained from the least-squares method, stereo depths for all corresponding point pairs are determined and the error from the LADAR depths calculated using:

$$E_z = \sum_{n=1}^N (Z_l - Z_s)^2. \quad (2.11)$$

These stereo depths, Z_s , could be calculated using equations derived for general camera configurations that calculate the intersection of the direction cosines from each camera. However, these lines may not exactly intersect, so calculating stereo-depth in this manner requires a minimization step. To avoid two minimization steps, the intensity and color images are converted to form a parallel baseline system, for which simplified closed-form expressions for stereo depth exist. The sum-squared-error of these depths from the laser range depths is then calculated:

$$E_z = \sum_{n=1}^N \left[(Z_l - Z_s^x)^2 + (Z_l - Z_s^y)^2 \right], \quad (2.12)$$

where E_z is the sum-squared-error of the stereo-depths from the laser range depths, N is the number of point correspondences, z_l is the laser range z value, z_s^x is the stereo depth obtained using x-disparity, and z_s^y is the stereo depth obtained using y-disparity. Since Z_s^x and Z_s^y can both be expressed as functions of $(\mathbf{R}, \vec{T})$, the solution to the relative orientation problem is found by minimizing E_z over $(\mathbf{R}, \vec{T})$. A downhill simplex algorithm was chosen to perform the minimization because of its ability to find the minimum of a function while avoiding the need to calculate derivatives.

To derive expressions for Z_s^x and Z_s^y as functions of $(\mathbf{R}, \vec{T})$ for use by the above minimization process, the intensity-color image pair is first rectified to allow the use of general parallel-axis equations for estimating range. Next, the images are rotated by the same angle to maximize the x- and y- baselines. Finally, the stereo-depths of the corresponding point pairs are related to $(\mathbf{R}, \vec{T})$ using both the x- and y- disparities. It should be noted that this method does not require the generation of rectified and rotated images. Instead, equations are derived in each step and combined at the end to form a relation to $(\mathbf{R}, \vec{T})$. This final transformation equation is applied only to the corresponding point set, not the entire image, for the purpose of estimating the epipolar geometry.

The first step in calculating the stereo depths, given $(\mathbf{R}, \vec{T})$, is to rotationally align the color image and the LADAR intensity image by reprojecting the color image using the initial guess of the rotation angles, θ , ϕ , and ψ . The rotational alignment, or rectification step, is illustrated in Figure 2.19.

The stereo-based range estimation process begins by rectifying the two images. To accomplish this, each color image point is back-projected to an arbitrary distance,

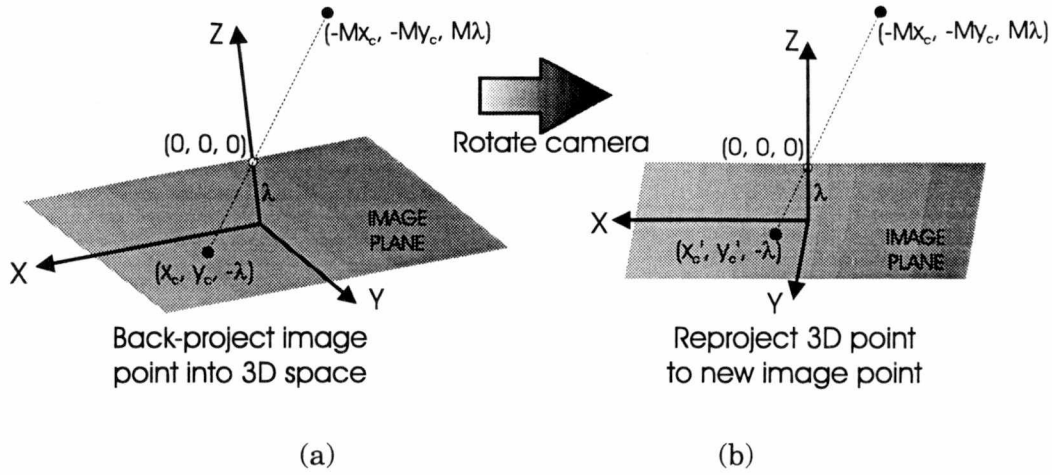


Figure 2.19: (a) Back-projection of a given color image point into 3-D space. (b) Reprojection of the 3-D point onto the color image plane that has been rotated so that it is oriented in the same direction as the LADAR camera.

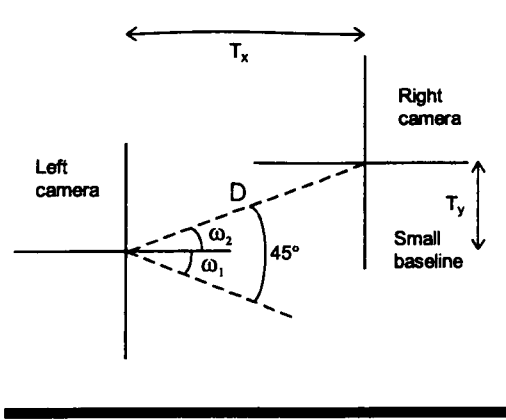
M , along its line of sight. The resulting point in space is rotated and reprojected as follows:

$$P(R_\psi R_\phi R_\theta)^{-1}w = PR_\theta^{-1}R_\phi^{-1}R_\psi^{-1}w \quad (2.13)$$

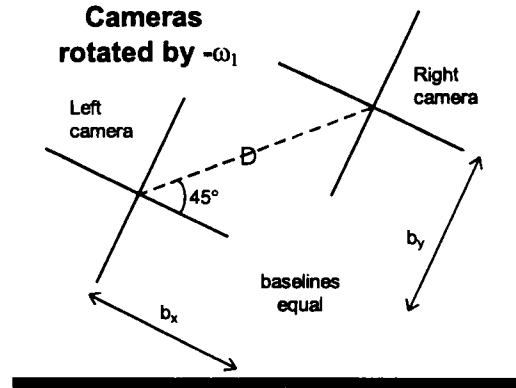
where P is the perspective transformation matrix, R_θ , R_ϕ , and R_ψ are rotation matrices relative to the color camera's focal point, and $w = [-Mx_c \ -My_c \ M\lambda]^T$. Evaluating Eq. 2.13 yields the following expressions for x'_c and y'_c :

$$\begin{aligned} x'_c &= \lambda \left(\frac{x_c(\sin \theta \sin \phi \sin \psi - \cos \theta \cos \psi) + y_c(\sin \theta \sin \phi \cos \psi + \cos \theta \sin \psi) - \lambda \sin \theta \cos \phi}{x_c(\cos \theta \sin \phi \sin \psi + \sin \theta \cos \psi) + y_c(\cos \theta \sin \phi \cos \psi - \sin \theta \sin \psi) - \lambda \cos \theta \cos \phi} \right) \\ y'_c &= \lambda \left(\frac{x_c \cos \phi \sin \psi + y_c \cos \phi \cos \psi + \lambda \sin \phi}{x_c(\cos \theta \sin \phi \sin \psi + \sin \theta \cos \psi) + y_c(\cos \theta \sin \phi \cos \psi - \sin \theta \sin \psi) - \lambda \cos \theta \cos \phi} \right) \end{aligned}$$

The second step in the stereo range calculation process is to rotate both images about their optical axes so that the x- and y- stereo baselines are equal. This eliminates the possibility of very small or zero length baselines as illustrated in Figure 2.20.



(a)



(b)

Figure 2.20: (a) Original camera orientations after the rectification step. (b) Camera orientations after they have been rotated about the optical axis to maximize the x - and y - baselines.

From Figure 2.20(a), the camera rotation angle that maximizes the baselines is determined as follows:

$$\omega_2 = \tan^{-1} \left(\frac{T_y}{T_x} \right)$$

$$\omega_1 = 45^\circ - \omega_2$$

Once the cameras are rotated by ω_1 , the new baselines, b_x and b_y , are calculated:

$$\begin{aligned} b_x(T_x, T_y) &= b_y(T_x, T_y) = D \cdot \cos 45^\circ \\ b_x(T_x, T_y) &= b_y(T_x, T_y) = \sqrt{T_x^2 + T_y^2} \cdot \cos 45^\circ \\ b_x(T_x, T_y) &= b_y(T_x, T_y) = \sqrt{\frac{T_x^2 + T_y^2}{2}} \end{aligned} \quad (2.14)$$

where D is the distance between the camera centers shown in Figure 2.20.

Since the cameras are rotated by ω_1 , the image coordinates of both the LADAR image points and the color image points are rotated by $-\omega_1$. The rotation of the perspective LADAR image coordinates, (x_l, y_l) , is performed as follows:

$$\begin{aligned}
 \begin{bmatrix} \hat{x}_l \\ \hat{y}_l \end{bmatrix} &= \begin{bmatrix} \cos \omega_1 & -\sin \omega_1 \\ \sin \omega_1 & \cos \omega_1 \end{bmatrix} \begin{bmatrix} x_l \\ y_l \end{bmatrix} \\
 \begin{bmatrix} \hat{x}_l \\ \hat{y}_l \end{bmatrix} &= \begin{bmatrix} \frac{T_x+T_y}{\sqrt{2(T_x^2+T_y^2)}} & \frac{T_y-T_x}{\sqrt{2(T_x^2+T_y^2)}} \\ \frac{T_x-T_y}{\sqrt{2(T_x^2+T_y^2)}} & \frac{T_x+T_y}{\sqrt{2(T_x^2+T_y^2)}} \end{bmatrix} \begin{bmatrix} x_l \\ y_l \end{bmatrix} \\
 \begin{bmatrix} \hat{x}_l \\ \hat{y}_l \end{bmatrix} &= \begin{bmatrix} \frac{T_x+T_y}{\sqrt{2(T_x^2+T_y^2)}}(x_l) + \frac{T_y-T_x}{\sqrt{2(T_x^2+T_y^2)}}(y_l) \\ \frac{T_x-T_y}{\sqrt{2(T_x^2+T_y^2)}}(x_l) + \frac{T_x+T_y}{\sqrt{2(T_x^2+T_y^2)}}(y_l) \end{bmatrix} \quad (2.15)
 \end{aligned}$$

where $\hat{x}_l$ is the x-coordinate of the rotated LADAR image point and $\hat{y}_l$ is the y-coordinate of the rotated LADAR image point. The rotation of the color image coordinates is performed similarly and gives the following result:

$$\begin{bmatrix} \hat{x}_c \\ \hat{y}_c \end{bmatrix} = \begin{bmatrix} \frac{T_x+T_y}{\sqrt{2(T_x^2+T_y^2)}}(x'_c) + \frac{T_y-T_x}{\sqrt{2(T_x^2+T_y^2)}}(y'_c) \\ \frac{T_x-T_y}{\sqrt{2(T_x^2+T_y^2)}}(x'_c) + \frac{T_x+T_y}{\sqrt{2(T_x^2+T_y^2)}}(y'_c) \end{bmatrix} \quad (2.16)$$

where $\hat{x}_c$ is the x-coordinate of the rotated color image point and $\hat{y}_c$ is the y-coordinate of the rotated color image point.

At this stage of the process, the hypothetical images given by $(\hat{x}_l, \hat{y}_l)$ and $(\hat{x}_c, \hat{y}_c)$ form a parallel optical axis stereo system as shown in Figure 2.21.

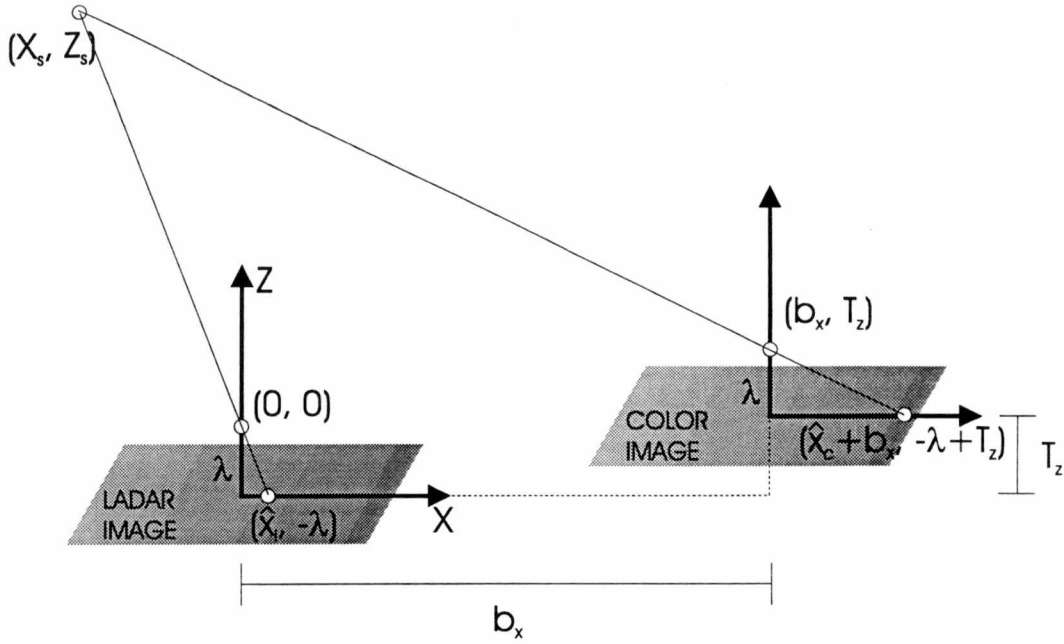


Figure 2.21: Arrangement of camera axes in derivation of stereo equations used when the cameras have not only an x -translation, but also a z -translation.

Thus, the equation of the x -direction cosine emanating from the pixel $(\hat{x}_l, \hat{y}_l)$ is given by:

$$X_s = -\frac{\hat{x}_l}{\lambda}(Z_s + \lambda) + \hat{x}_l \quad (2.17)$$

where X_s is the stereo-determined x -coordinate in 3-D space, Z_s is the stereo-determined z -coordinate in 3-D space, $\hat{x}_l$ is the rotated x -coordinate in the LADAR image, and λ is the focal length of both the projective LADAR image and the color camera.

The equation of the x -direction cosine for the corresponding feature point $(\hat{x}_c, \hat{y}_c)$ in the color image is written as:

$$X_s = -\frac{\hat{x}_c}{\lambda}(Z_s + \lambda - T_z) + \hat{x}_c + b_x \quad (2.18)$$

where $\hat{x}_c$ is the rotated x-coordinate in the rectified color image, T_z is the displacement of the two cameras in the z-direction, and b_x is the length of the x- baseline.

Equating the right sides of Eqs. 2.17 and 2.18 and solving for Z_s results in the desired stereo depth equation using the x- disparities:

$$Z_s^x = \frac{\lambda b_x + T_z \hat{x}_c}{\hat{x}_c - \hat{x}_l} \quad (2.19)$$

The stereo depth equation using the y -disparities is derived in an identical manner:

$$Z_s^y = \frac{\lambda b_y + T_z \hat{y}_c}{\hat{y}_c - \hat{y}_l} \quad (2.20)$$

Substituting equations 2.14, 2.15, 2.16, 2.19, and 2.20 into Equation 2.12 produces the following expression for the sum-squared-error of the LADAR and stereo-based ranges as a function of $(\mathbf{R}, \vec{T})$:

$$E_z(\mathbf{R}, \vec{T}) = \sum_{n=1}^N \left[\left(Z_l - \frac{\lambda(T_x^2 + T_y^2) + T_z[x'_c(T_x + T_y) + y'_c(T_y - T_x)]}{(x'_c - x_l)(T_x + T_y) + (y'_c - y_l)(T_y - T_x)} \right)^2 + \left(Z_l - \frac{\lambda(T_x^2 + T_y^2) + T_z[x'_c(T_x - T_y) + y'_c(T_x + T_y)]}{(x'_c - x_l)(T_x - T_y) + (y'_c - y_l)(T_x + T_y)} \right)^2 \right] \quad (2.21)$$

This expression is then minimized over $(\mathbf{R}, \vec{T})$ until E_z falls below a set threshold. The result of the process is an estimate of the sensor system's epipolar geometry, $(\mathbf{R}, \vec{T})$. To improve this estimate, outliers are removed by eliminating correspondence points whose individual depth error, $(Z_l - Z_s^x)^2 + (Z_l - Z_s^y)^2$, is above a user-defined threshold value. Repeating the minimization algorithm with the truncated data set produces the final estimate of $(\mathbf{R}, \vec{T})$. Figure 2.22 shows a comparison of the registra-

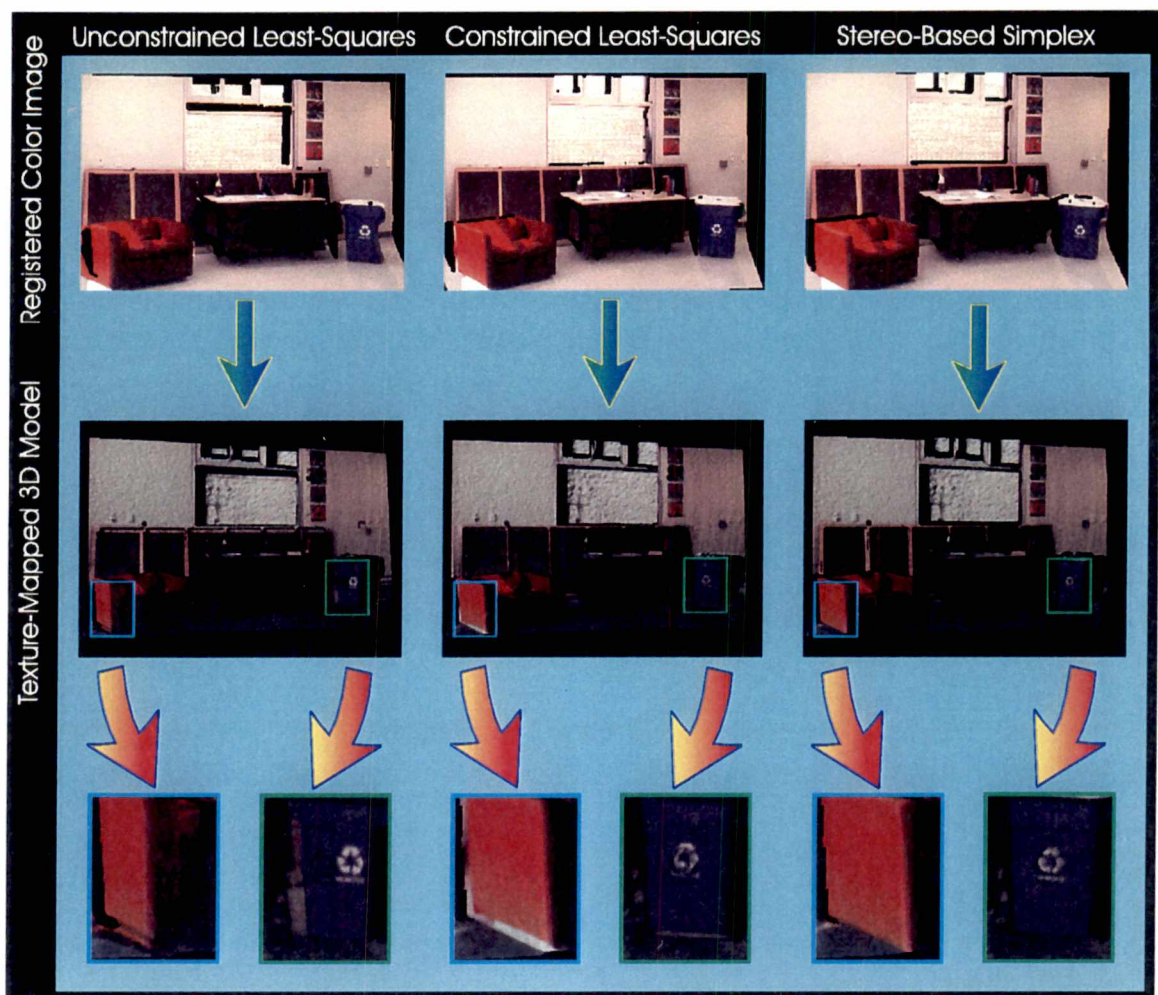
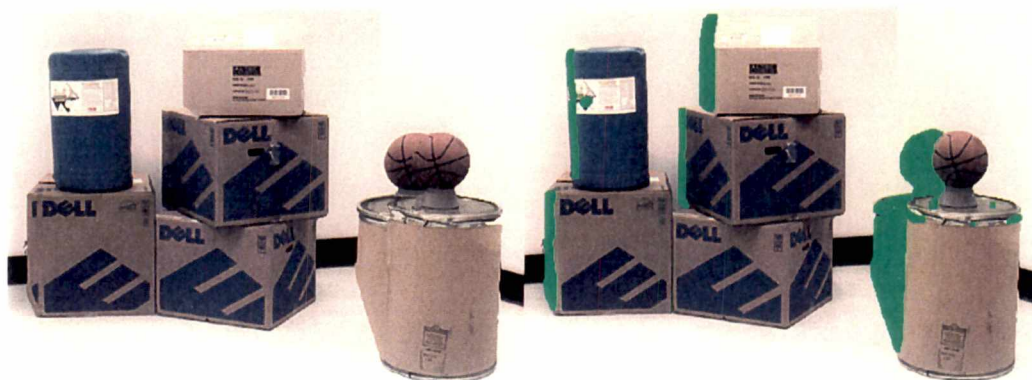


Figure 2.22: Comparison of registration results using the least-squares method, both unconstrained and constrained, and the simplex method. The original color image is shown followed by the registered color image texture-mapped onto the $2\frac{1}{2}$ -D model. Areas where the registration quality is most apparent are enlarged to show detail.

tion results using the initial value of $(\mathbf{R}, \vec{T})$ computed by the least-squares method, given the correspondences from the unconstrained search, the more accurate estimate from the constrained search and least-squares estimation, and the final result generated by the non-linear minimization method. It can be seen from the figure that the non-linear method produces the most accurate registration results. Notice that the alignment of color and range edges improves with each step.

2.5 Reprojection

With the position and orientation of the color camera relative to the laser range scanner known, the color image is reprojected to the range scanner's frame of reference to create a texture map. While this step is not absolutely essential, it simplifies the texture-mapping process and allows occluded regions to be identified. Using the LADAR image, the color of each unoccluded point in the intensity image is determined through an inverse perspective projection into the color camera space. However, the back-projection of occluded points in the intensity image will also intersect the color image plane, and thus incorrect pixel values will be included in the resulting image as demonstrated in Figure 2.23(a). To overcome this occlusion problem, z-buffering is used to associate a depth value with each color pixel as it is used. If another point in the intensity image attempts to use a color pixel that has already been used, the depth value is checked. If the depth value for the new point is less than that of the old one, the old point in the intensity image is labeled as an occluded region, and the new point is used to specify the pixel color. If the depth value for the new point is greater than that of the old one, the new point is labeled as occluded. Figure 2.23(b) shows a registered color image formed with z-buffering. Once every pixel in the intensity image has been replaced, the registration process is complete. The reprojected color image is registered with the intensity image and thus, the range image, and is available for use as a texture map for a Euclidean model created from the range data.



(a)

(b)

Figure 2.23: (a) Registered color image of the 'boxes' scene without using z-buffering. Note the incorrect pixel usage at occluded points. (b) Registered color image with z-buffering. Occluded points have been labeled as such by replacing the incorrect color values with a chosen, uniform color.

CHAPTER 3

Experimental Results and Discussion

In this section, the methods employed to test the registration algorithm using both synthetic and real data are described. Simulated results using synthetic range data are shown, along with actual registration results obtained for real scenes.

3.1 Synthetic Data Results

For initial testing of the non-linear estimation algorithm, a three-dimensional *Inventor* model of an apple was used to generate simulated range and intensity images. Two intensity images were obtained by projecting the model vertices onto two different image planes with user-specified relative position and orientation, as shown in Figure 3.1. Note that occlusions were not considered in this test. The range

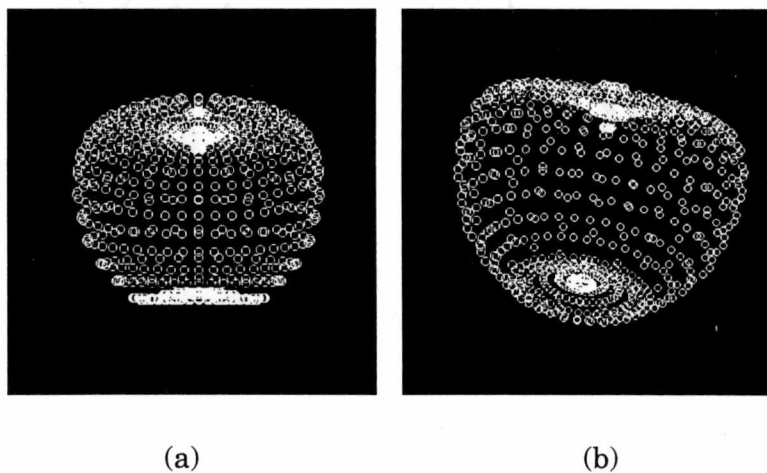


Figure 3.1: (a) *Projection of the vertices of the 'apple' Inventor model in first viewpoint.*
(b) *Projection of 'apple' vertices in the other viewpoint.*

information was computed directly from the 3-D coordinates of the points in the apple model. To make these simulations more realistic, both random and spike noise were added to the range data. Also included in the simulation were pixel quantization errors.

The following results were obtained by applying the stereo-simplex algorithm to the corresponding point pairs from Figure 3.1. Uniformly distributed random noise of ± 0.25 units was added to the data points and spike noise of ± 3 units was added to 10% of the data points. For reference, the radius of the apple model is one unit. Correspondence error of ± 3 pixels was also added. Table 3.1 shows the rotation and translation values calculated by the stereo-simplex algorithm, first with all data points, and then with outliers rejected.

Table 3.1: *Rotation and translation parameters calculated by the simplex algorithm applied to the synthetic apple data.*

	Actual Values	Before Outlier Rejection (208 points)	% Error	After Outlier Rejection (189 points)	% Error
T_x	5.0000	4.8540	-2.9204	4.9986	-0.0279
T_y	2.0000	1.9464	-2.6790	2.0219	1.0961
T_z	2.0000	1.9099	-4.5038	1.9792	-1.0380
θ	25.0000	24.4575	-2.1699	25.1758	0.7034
ϕ	10.0000	9.8894	-1.1058	10.1563	1.5632
ψ	5.0000	5.5486	10.9712	4.9148	-1.7039

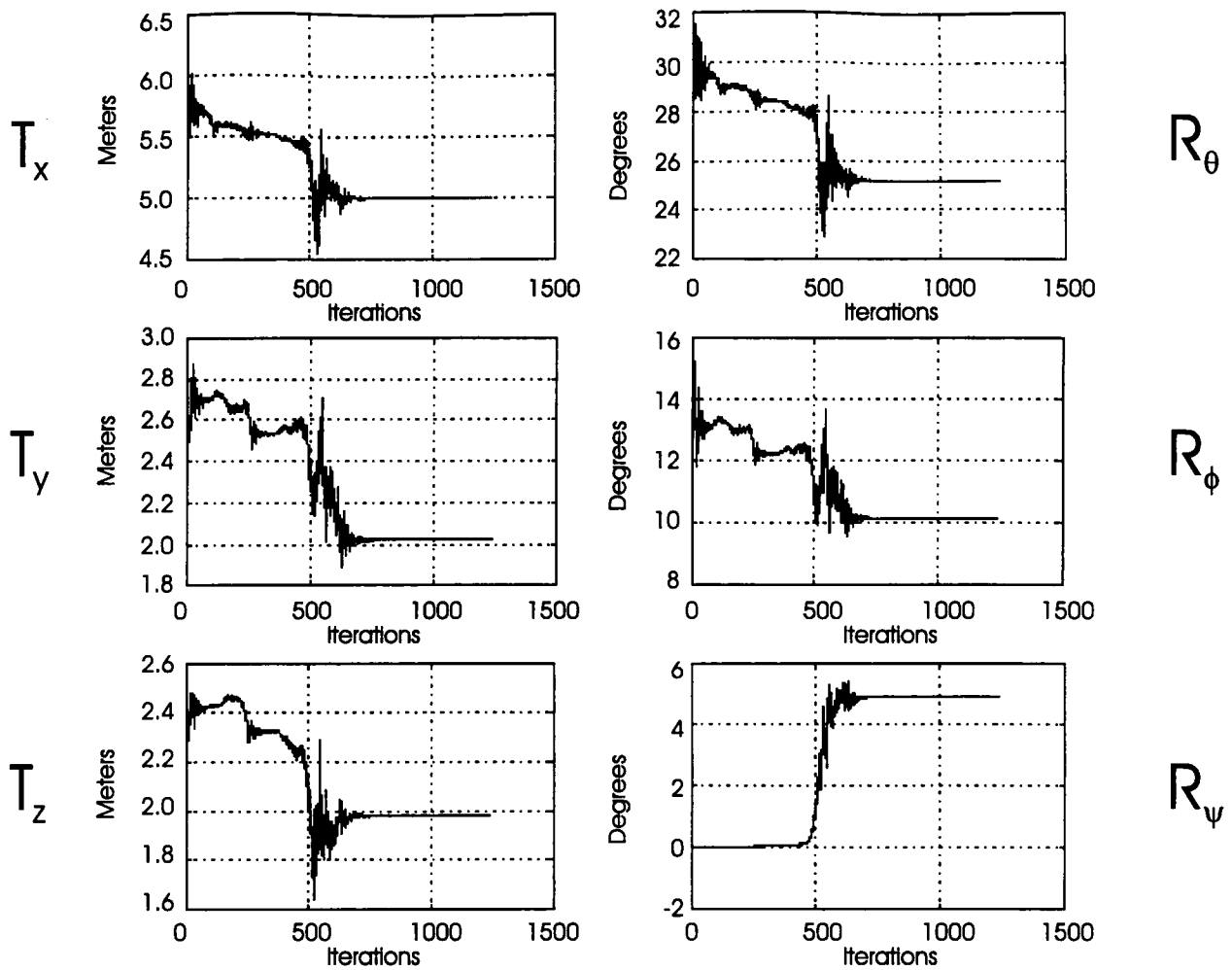


Figure 3.2: *Convergence of the rotation and translation parameters in the stereo-simplex algorithm for the synthetic apple model.*

Figure 3.2 shows the convergence of the simplex algorithm for each orientation parameter. Figure 3.3 shows the error of the LADAR depth from the stereo-calculated depth for each data point. Outliers are rejected by deleting points whose depth error is above a set threshold from the simplex calculation as Figure 3.4 illustrates.

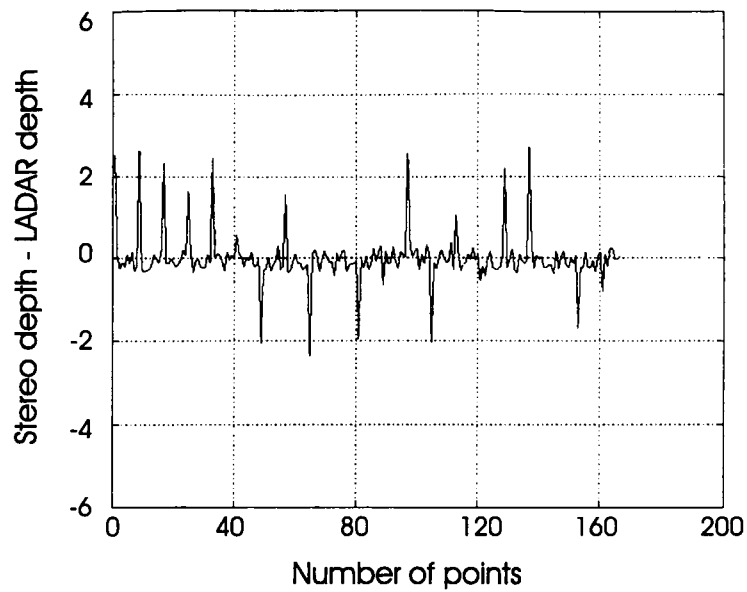


Figure 3.3: *Difference of the LADAR depth values from the stereo-calculated depth values for each 'apple' vertex with outliers **included**.*

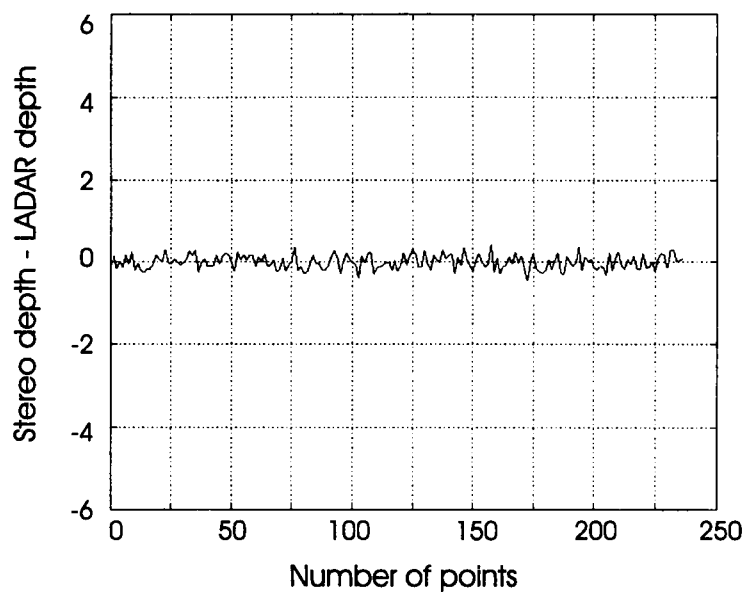
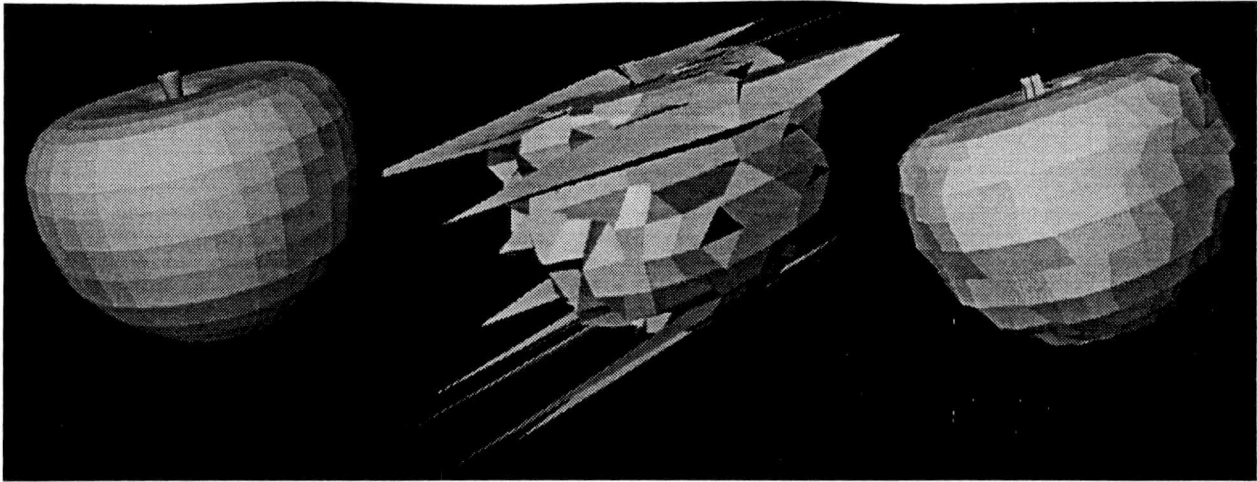


Figure 3.4: *Difference of the LADAR depth values from the stereo-calculated depth values for each 'apple' vertex with outliers **removed**.*

3.2 Discussion of Synthetic Data Results

During our simulated experiments, the simplex algorithm performed well even when applied to extremely noisy data. Absolute range values for the ‘apple’ scene used in our experiments were between two and eight meters. Using random noise uniformly distributed between 0 and 0.25 units, spike noise with amplitudes varying from 0 to 3 units, and correspondence error of ± 3 pixels in a 512x512 image, the non-linear estimation technique still converged to the correct solution within 2%, as shown in Table 3.1. In our testing, we also discovered that the initial guess to the non-linear estimation algorithm did not need to be very accurate. Using initial values such as ‘two meter translation to the right’ allowed the non-linear registration algorithm to converge in the vast majority of our experiments.

During the simulation of the registration process with the synthetic data, it also became apparent that it is possible to smooth the range data for points which have been matched in the stereo process. During the non-linear stage of the registration process, points whose stereo depth is significantly different from the LADAR depth are removed from the error minimization calculation. The points that are removed are usually spike noise points. Replacing these LADAR range values with estimates based on their stereo depth smooths the model without losing structural detail as shown in Figure 3.5.



(a)

(b)

(c)

Figure 3.5: (a) *The original 'apple' Inventor model.* (b) *The 'apple' model with random and spike noise added.* (c) *The 'apple' model with spike noise removed by replacing the noisy points with points based on the stereo-calculated depth values.*

3.3 Registration Results using Actual Imagery

To test both the registration and correspondence algorithms, several scenes were scanned with the Perceptron laser range scanner. For each scan, a color image was taken with a Kodak DCS 460 digital color camera equipped with a 35mm lens placed on a tripod near the Perceptron. This camera produces 3072×2060 digital images in full color. Each image was resampled to a smaller size (768×512) for use in our registration algorithm.

A typical scene and set of corresponding corner points for our experiments is shown in Figure 3.6. Using these automatically located and matched corner points, the algorithm determined the relative rotation and translation of the color camera relative to the LADAR camera and produced a registered version of the color image as shown in Figure 3.7. Figure 3.8 shows the registered color image texture-mapped



(a)



(b)

Figure 3.6: (a) Corners found in 'boxes' intensity image by the automatic feature correspondence technique. (b) Matching corners found in 'boxes' color image.

onto the $2\frac{1}{2}$ -D model of the 'boxes' scene. Plots showing the convergence of the simplex algorithm are shown in Figure 3.9. Table 3.2 shows the rotation and translation parameters at different stages in the registration process.

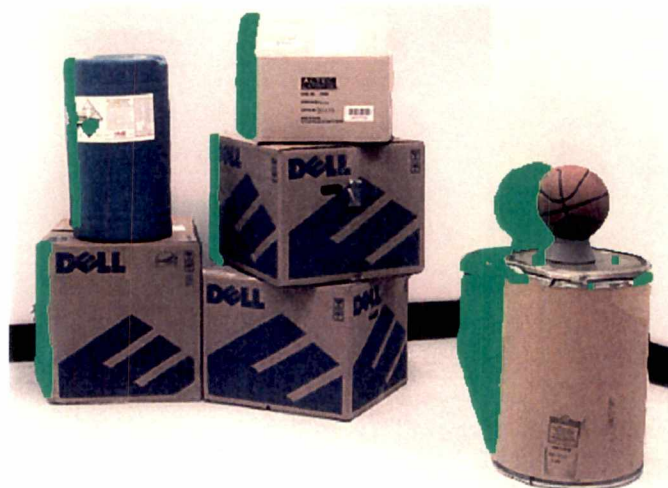


Figure 3.7: Registered color image of the 'boxes' scene with z-buffering.

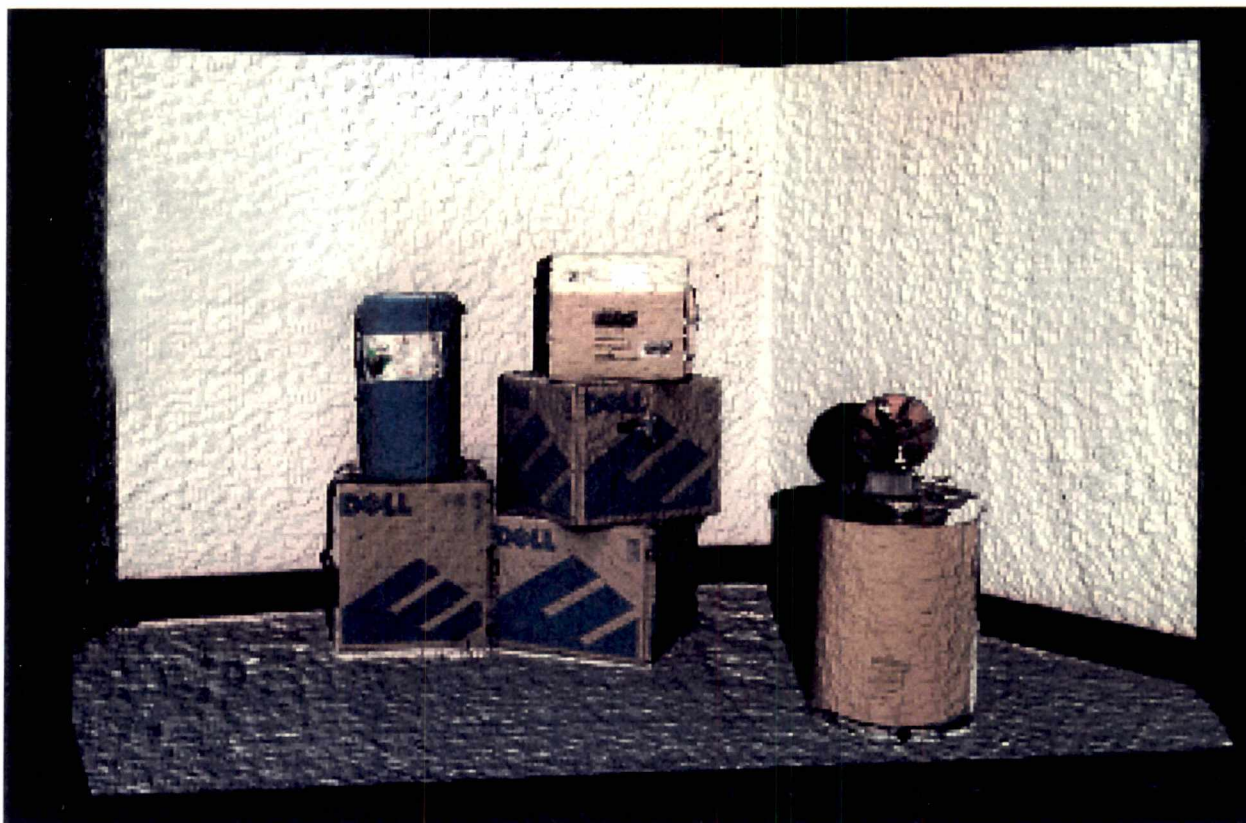


Figure 3.8: *Registered color image of the 'boxes' scene mapped onto the $2\frac{1}{2}$ -D model created from the range data.*

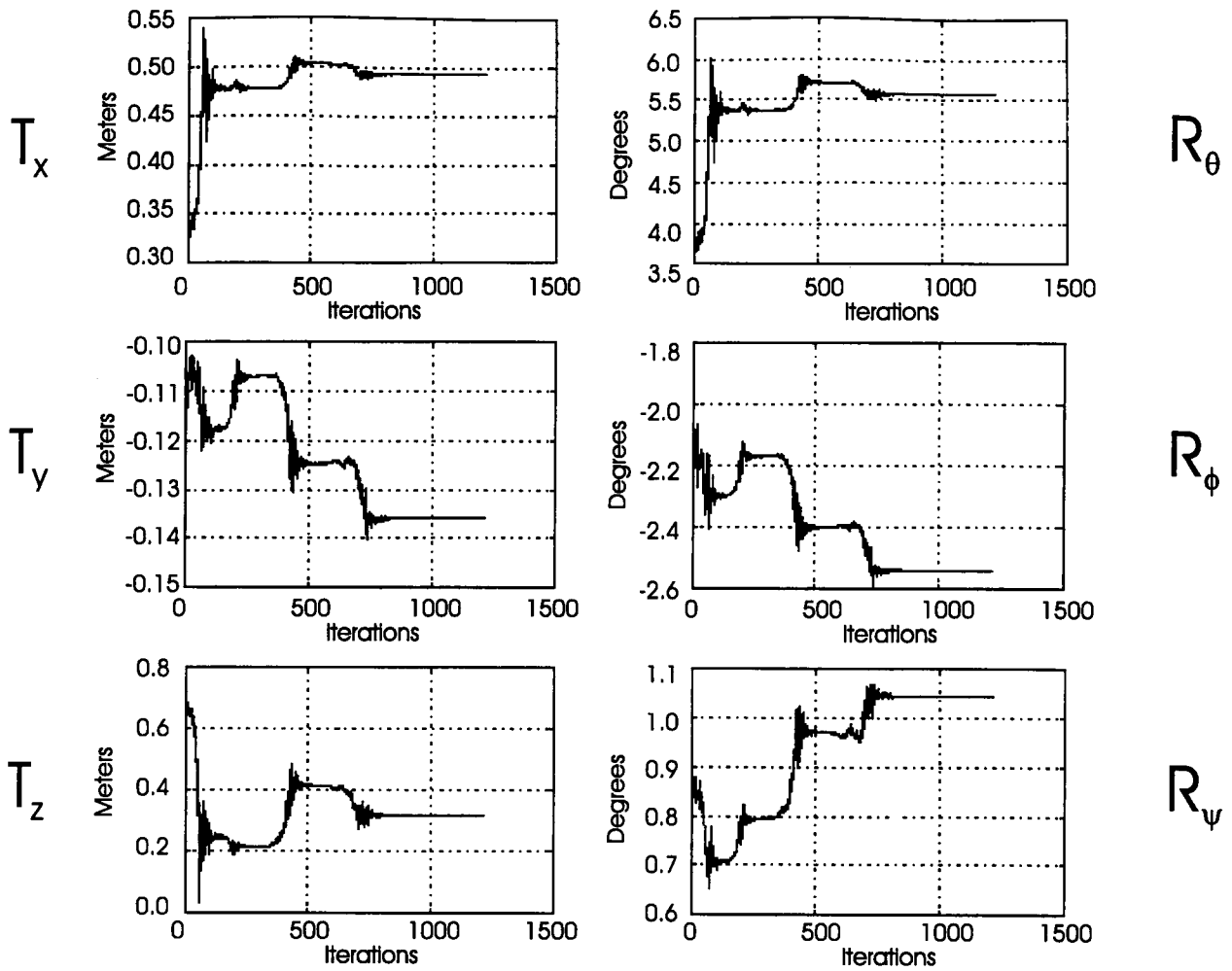


Figure 3.9: Convergence of rotation and translation parameters in the stereo-simplex algorithm for the 'boxes' scene.

Table 3.2: Calculation of rotation and translation parameters by the least-squares and the simplex methods applied to the 'boxes' scene.

	Unconstrained Least Squares (28 points)	Constrained Least Squares (36 points)	Initial Simplex Solution (36 points)	Simplex Sol'n- Outliers Removed (31 points)
T_x	0.0989m	0.3446m	0.4048m	0.4932m
T_y	-0.1015m	-0.1078m	-0.1419m	-0.1359m
T_z	0.0904m	0.0664m	-0.1049m	0.0318m
θ	0.7194°	3.6413°	4.2448°	5.5787°
ϕ	-2.1800°	-2.1612°	-2.5339°	-2.5381°
ψ	0.1998°	0.8377°	0.9051°	1.0446°

The registration algorithm works equally well with more complicated scenes. Figures 3.10-3.18 and Tables 3.3-3.5 show the results of the registration algorithm when applied to three other scenes. The first is a scene containing a desk, a chair, and a table. The next scene is an office scene containing cabinets, desks, and computers. The final scene is a complex scene containing actual people.

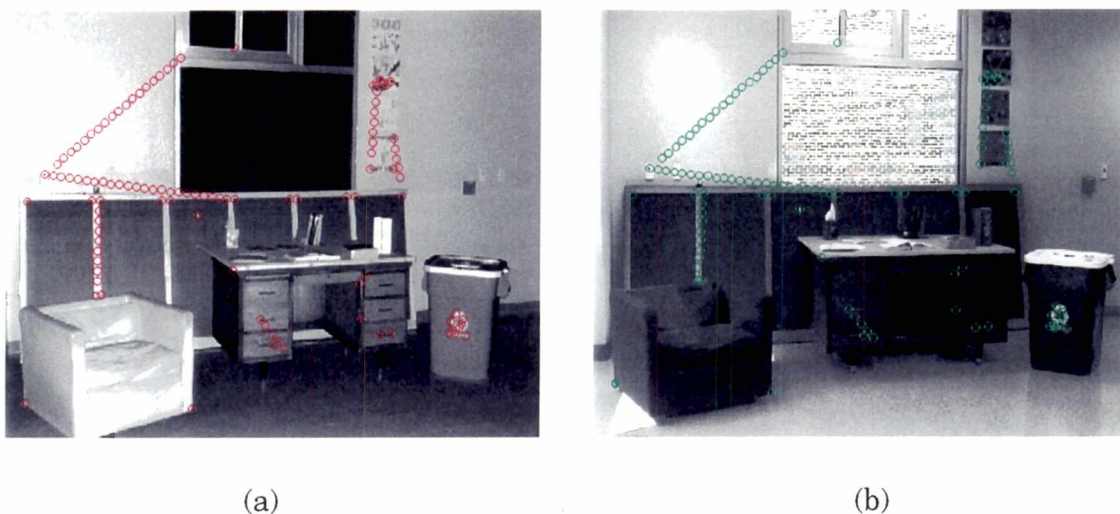


Figure 3.10: (a) Corners found in the 'desk' intensity image by the automatic point correspondence technique. (b) Matching corners found in the 'desk' color image.



Figure 3.11: Registered color image of the 'desk' scene mapped onto the $2\frac{1}{2}$ -D model created from the range data.

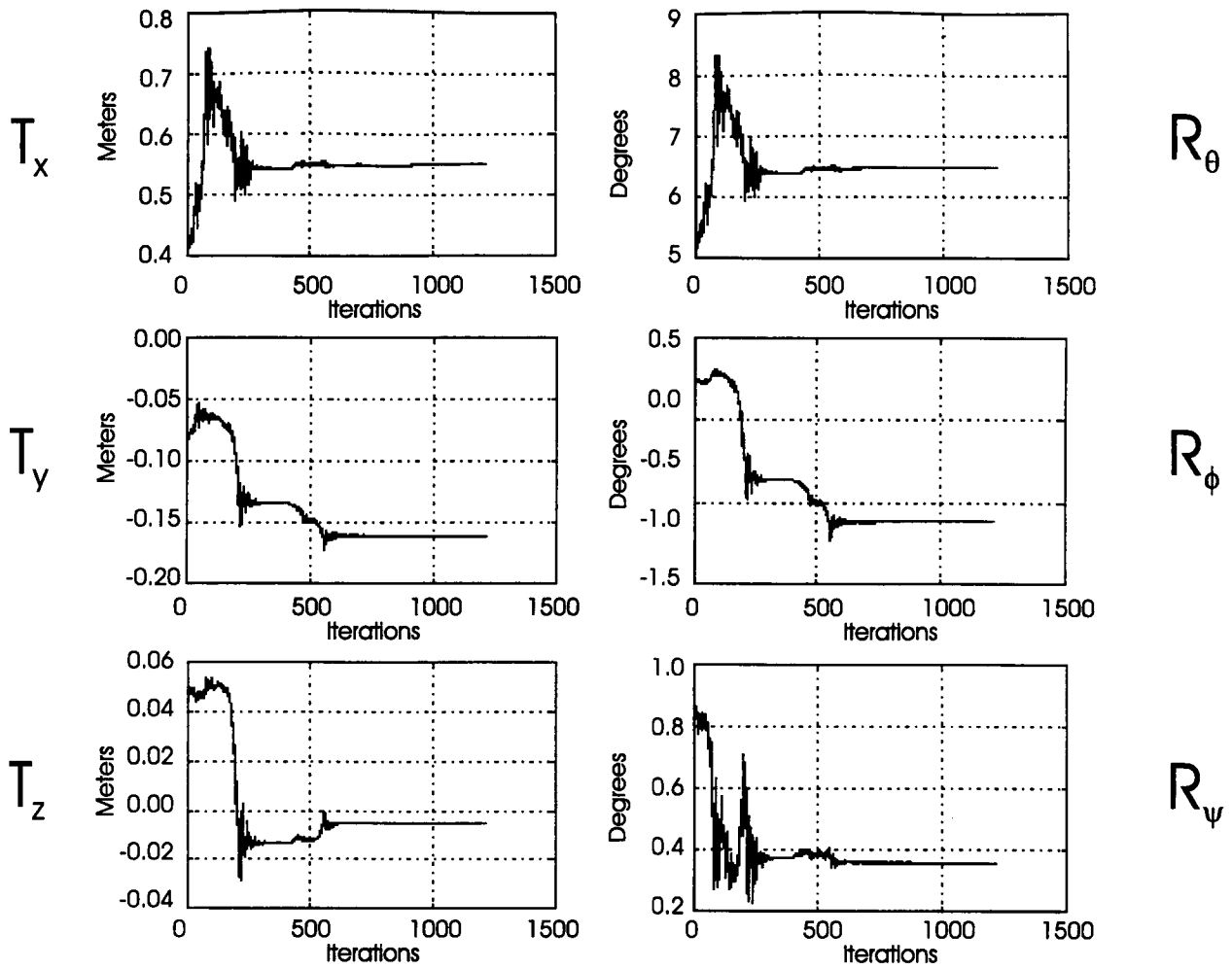


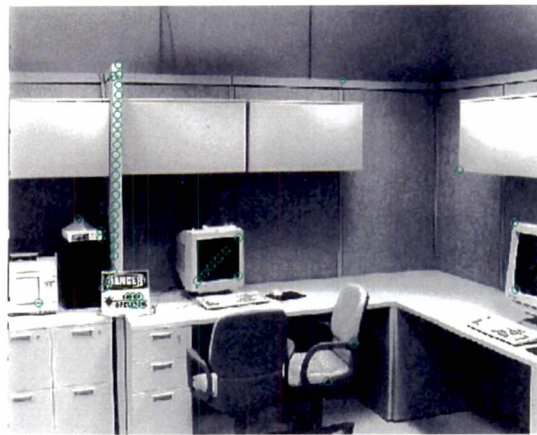
Figure 3.12: Convergence of rotation and translation parameters in the stereo-simplex algorithm for the 'desk' scene.

Table 3.3: Calculation of rotation and translation parameters by the least-squares and the simplex methods applied to the 'desk' scene.

	Unconstrained Least Squares (28 points)	Constrained Least Squares (115 points)	Initial Simplex Solution (115 points)	Simplex Sol'n- Outliers Removed (110 points)
T_x	0.5004m	0.4342m	0.5910m	0.5506m
T_y	0.3140m	-0.0778m	-0.1516m	-0.1603m
T_z	-0.2528m	0.0491m	-0.0156m	-0.0053m
θ	5.4241°	5.1634°	6.8101°	6.4787°
ϕ	3.7478°	-0.2482°	-1.0173°	-1.1156°
ψ	0.6745°	0.8267°	0.3715°	0.3594°



(a)



(b)

Figure 3.13: (a) Corners found in the 'office' intensity image by the automatic point correspondence technique. (b) Matching corners found in the 'office' color image.



Figure 3.14: Registered color image of 'office' scene mapped onto the $2\frac{1}{2}$ -D model created from the range data.

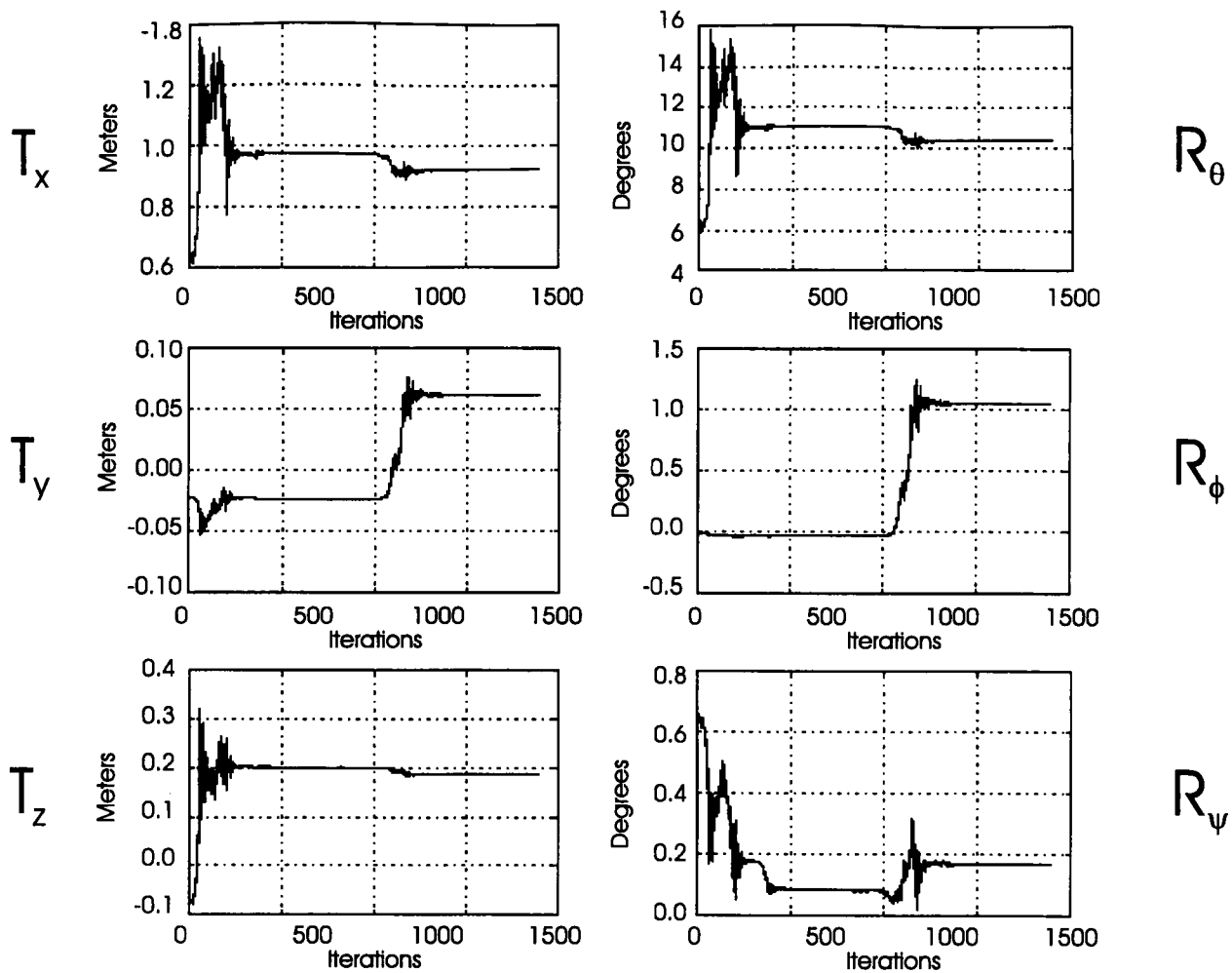
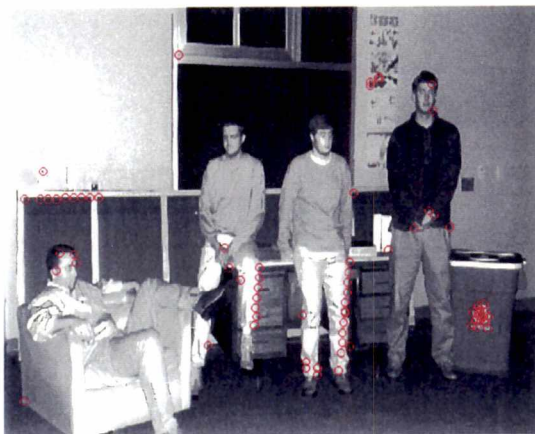


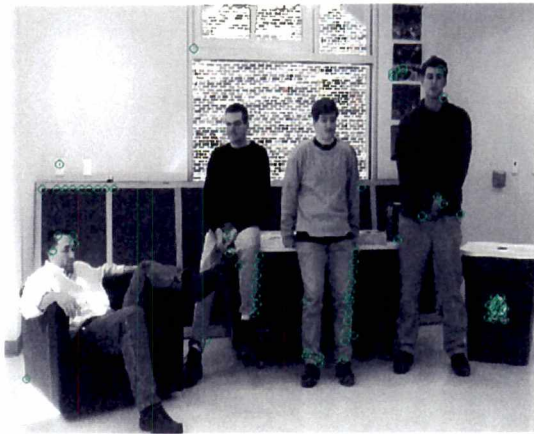
Figure 3.15: Convergence of rotation and translation parameters in the stereo-simplex algorithm for the 'office' scene.

Table 3.4: Calculation of rotation and translation parameters by the least-squares and the simplex methods applied to the 'office' scene.

	Unconstrained Least Squares (27 points)	Constrained Least Squares (51 points)	Initial Simplex Solution (51 points)	Simplex Sol'n- Outliers Removed (48 points)
T_x	0.2284m	0.6155m	0.8908m	0.9232m
T_y	0.2434m	-0.0218m	0.0807m	0.0614m
T_z	-0.3589m	-0.0757m	0.1483m	0.1881m
θ	2.4358°	6.1413°	9.8081°	10.4157°
ϕ	2.7364°	-0.0165°	1.2778°	1.0500°
ψ	1.0095°	0.6307°	0.2151°	0.1669°



(a)



(b)

Figure 3.16: (a) Corners found in the 'people' intensity image by the automatic point correspondence technique. (b) Matching corners found in the 'people' color image.



Figure 3.17: Registered color image of 'people' scene mapped onto the $2\frac{1}{2}$ -D model created from the range data.

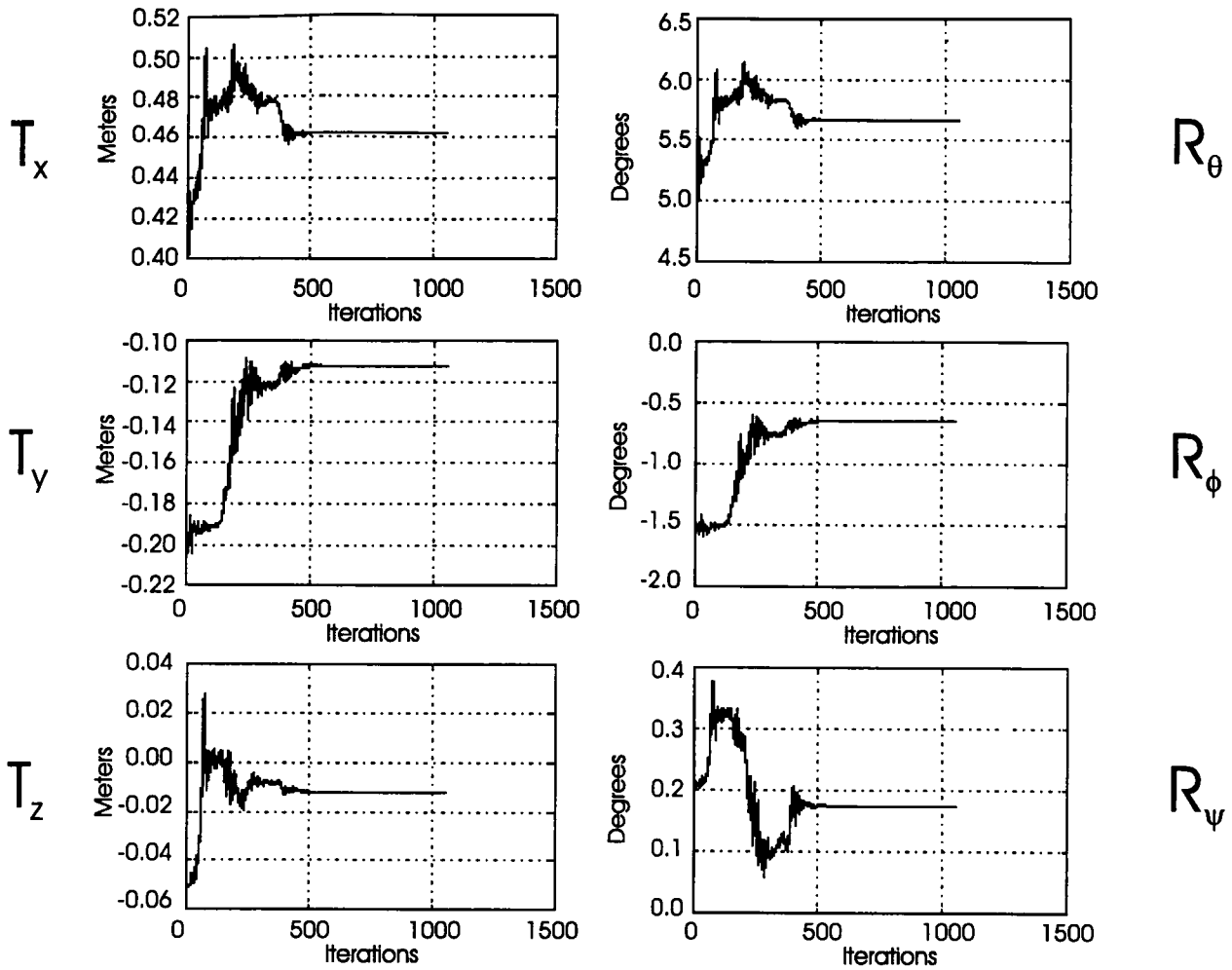


Figure 3.18: Convergence of rotation and translation parameters in the stereo-simplex algorithm for the 'people' scene.

Table 3.5: Calculation of rotation and translation parameters by the least-squares and the simplex methods applied to the 'people' scene.

	Unconstrained Least Squares (35 points)	Constrained Least Squares (74 points)	Initial Simplex Solution (74 points)	Simplex Sol'n- Outliers Removed (72 points)
T_x	0.5976m	0.4228m	0.4614m	0.4618m
T_y	-0.1365m	-0.1950m	-0.1124m	-0.1120m
T_z	-0.0031m	-0.0496m	-0.0151m	-0.0121m
θ	7.0043°	5.2582°	5.6630°	5.6669°
ϕ	-0.8700°	-1.5109°	-0.6447°	-0.6435°
ψ	-0.1393°	0.2048°	0.1454°	0.1759°

3.4 Discussion of Registration Results using Actual Imagery

Our registration technique also performed well when applied to actual LADAR and color imagery. The algorithm was able to successfully register several image pairs ranging from simple arranged scenes of boxes and barrels to complex scenes of offices and people. Figures 3.6, 3.10, 3.13, and 3.16 show the results of applying the corner finder to each image. It is interesting to observe that the more complex scenes produced the largest number of detected corners and thus, were the easiest to register. Scenes with large areas containing little or no detail produce the fewest number of corners, and thus, were the most difficult to register accurately.

Figures 3.8, 3.11, 3.14, and 3.17 show the registered color image of each scene texture-mapped onto the $2\frac{1}{2}$ -D model created from the range information. As can be seen the registration process produced realistic, accurate color models of each scene. The only noticeable flaws are those caused by occlusions and noisy range data.

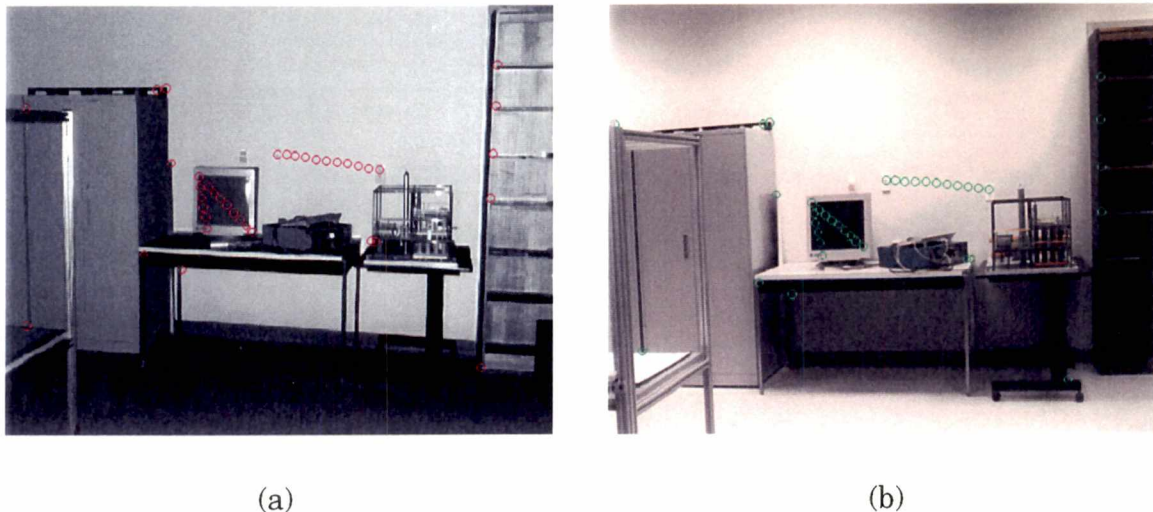
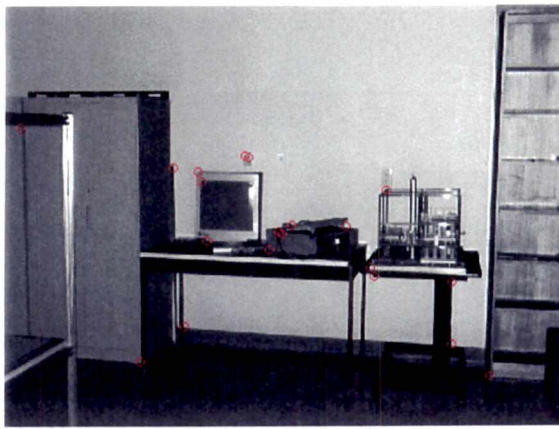


Figure 3.19: (a) *Point correspondences located in the intensity image.* (b) *Point correspondences located in the color image acquired from nearly the same position.*



(a)



(b)

Figure 3.20: (a) *Point correspondences located in the intensity image.* (b) *Point correspondences located in the color image with a large rotation about the optical axis.*

During testing of the registration algorithm with actual imagery, two situations were encountered that would cause the registration process to fail. First, if the LADAR and color images are acquired from nearly identical locations, there will not be enough disparity between corresponding points in the images to calculate an accurate stereo-depth measurement. In this situation, the stereo-simplex algorithm will converge to an incorrect value for $(\mathbf{R}, \vec{T})$. This is expected since the registration approach is stereo-based, and the baselines are small or nonexistent. Figure 3.19 shows such an image pair.

The second situation that could cause failure occurs if the cameras are placed too far apart or have a large difference in the angle by which they are rotated about their optical axes. Figure 3.20 shows an example of this. When this situation occurs, the feature correspondence algorithm fails due to its scale and rotation variance. Windows centered at corresponding points in the image pair do not resemble each other enough to correlate well, so very few, if any, correct matches are found. However,

the registration algorithm will still function properly if corresponding points are found manually.

CHAPTER 4

Conclusions

In this paper, a stereo-based method of registering color and LADAR images acquired from an uncalibrated camera system was successfully developed and implemented. The motivation for this work is the desire to supplement range information with data from additional sensors for use in tele-operated robotic tasks. By registering the images acquired from multiple sensors, data from each sensor can be viewed from a common 3-D frame of reference. In the case of a color camera, registered color images of a scene are used as texture maps to a three-dimensional model created from the range information. Chapter 1 explained the motivation for this research in detail, followed by a discussion of past work relevant to this research.

Chapter 2 described the entire registration process in five steps: 1) range image acquisition hardware, 2) coordinate system transformations, 3) feature correspondence, 4) determination of relative external camera orientation, and 5) color image reprojection. In order to use LADAR images in the registration process, the images were converted from the raw output form of the laser range camera described in Section 2.1 into standard coordinate system representations. Section 2.2 discussed the math necessary to transform the raw range information measured by an azimuth-elevation scanner, such as the Perceptron used for our work, into both a two-dimensional perspective projection and three-dimensional cartesian and spheri-

cal coordinate systems. Section 2.2 ended with a discussion of the process of creating panoramic range images, since it followed naturally from the already derived spherical coordinate transformation equations. Using the results from this section to create a standard projective stereo pair from the LADAR/color images, a method for performing feature detection and correspondence was discussed in Section 2.3. These correspondences were used as input to a novel stereo-based technique for determining the epipolar geometry of a LADAR/color camera system, as presented in Section 2.4. Section 2.5 explained the method used to reproject the color image to the point-of-view of the LADAR image, completing the registration process.

The next chapter discussed the results of applying the registration technique to a synthetic data set and several actual scenes. The synthetic data results prompted a discussion of the possibility of data smoothing as an added effect of the registration process.

Future work in this research would focus mainly on five areas. The first is the development of a more robust feature location and matching algorithm, to allow for the automatic registration of images acquired from widely disparate camera locations with less overlapping image area. One possibility for this is the use a combined edge/corner approach instead of using corners alone. The second possibility for future work is to test the registration algorithm with non-color images, such as thermal and gamma images. This would only require minor adjustments to the registration algorithm. The smoothing of range data with the outlier rejection step of the non-linear minimization step of our algorithm is another area of future work. Since the relative orientation of the LADAR and color camera is known, edges found using an

edge detector can be matched in the two images. The stereo-calculated range can then be used to smooth the noisy range values at the edges. The fourth possibility for future work is the addition of internal camera calibration parameters, such as focal length, to the registration process. Finally, this research could also be used to register two range images instead of a range/color image pair.

BIBLIOGRAPHY

BIBLIOGRAPHY

- [1] C. S. Andersen, C. B. Madsen, J. J. Sorensen, and N. Kirkeby, "Navigation Using Range Images on a Mobile Robot," *Robotics and Automation Systems* **10**(2), pp. 147–160, 1992.
- [2] C. Wu, Y. H. Kim, and J. H. Kim, "Scene Analysis of Range Images for Mobile Robot Navigation," in *Proceedings of the IEEE Southeast Con '93*, April 1993.
- [3] P. Boulanger and F. Blais, "Range Image Segmentation, Free Space Determination, and Position Estimate for a Mobile Vehicle," in *Proceedings of the SPIE*, vol. 1831, pp. 444–455, The International Society for Optical Engineering, 1993.
- [4] M. M. Trivedi, M. A. Abidi, R. C. Gonzales, C. Chen, and C. Bidlack, "Multi-sensor Based Robotic Systems for Inspection and Manipulation Tasks," in *Proceedings of the 38th Conference on Remote Systems Technology*, vol. 1, pp. 31–35, 1990.
- [5] R. T. Whitaker, "A level-set approach to 3d reconstruction from range data," *To Appear: International Journal of Computer Vision*.
- [6] K. S. Arun, T. S. Huang, and S. D. Blostein, "Least-Squares Fitting of Two 3-D Point Sets," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **14**, pp. 239–256, February 1992.
- [7] B. Horn, "Closed-form solution of absolute orientation using unit quaternions," *Journal of the Optical Society of America*, pp. 629–642, 1987.
- [8] B. Horn, H. Hilden, and S. Negahdaripour, "Closed-form solution of absolute orientation using orthonormal matrices," *Journal of the Optical Society of America*, pp. 1127–1135, 1988.
- [9] E. Trucco and A. Verri, *Introductory Techniques for 3-D Computer Vision*, Prentice-Hall, New Jersey, 1998.
- [10] P. K. Besl and N. D. McKay, "A Method for Registration of 3-D Shapes," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **14**, pp. 239–256, February 1992.
- [11] G. Godin, M. Rioux, and R. Baribeau, "Three-dimensional registration using range and intensity information," in *Proceedings of the SPIE*, vol. 2350, pp. 279–290, The International Society for Optical Engineering, 1994.
- [12] Z. Lin, T. S. Huang, and S. D. Blostein, "Motion estimation from 3-D Point Sets with and without Correspondences," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 194–201, June 1986.

- [13] J. Aloimonos and I. Rigoutsos, "Determining the 3-D Motion of a Rigid Planar Patch Without Correspondence, Under Perspective Projection," in *Proceedings of the IEEE Computer Society Workshop on Motion: Representation and Analysis*, pp. 167–174, May 1986.
- [14] Z. Lin, H. Lee, and T. Huang, "Finding 3-D Point Correspondences in Motion Estimation," in *Proceedings of the 8th International Conference on Pattern Recognition*, pp. 303–305, October 1986.
- [15] J. K. Aggarwal and M. J. Magee, "Determining Motion Parameters Using Intensity Guided Range Sensing," *Pattern Recognition* **19**, pp. 169–180, 1986.
- [16] J. Aggarwal and B. Sabata, "Motion from a Sequence of Range Images," in *Proceedings of Workshop on Motion*, pp. IP1–IP10, October 1990.
- [17] T. Masuda and N. Yokoya, "A Robust method for Registration and Segmentation of Multiple Range Images," *Computer Vision and Image Understanding* **61**, pp. 295–307, May 1995.
- [18] B. K. Horn, "Relative Orientation," *International Journal of Computer Vision* **4**, pp. 59–78, 1990.
- [19] B. Caprile and V. Torre, "Using Vanishing Points for Camera Calibration," *International Journal of Computer Vision* **4**, pp. 127–140, 1990.
- [20] R. I. Hartley, "Estimation of Relative Camera Positions for Uncalibrated Cameras," in *Proceedings of the 5th International Conference on Computer Vision*, pp. 1064–1070, 1992.
- [21] O. D. Faugeras, *Three-Dimensional Computer Vision: A Geometric Viewpoint*, MIT Press, Cambridge, Massachusetts, 1993.
- [22] Y. Muller and R. Mohr, "Planes and Quadrics Detection Using Hough Transform," in *IEEE Proceedings, Seventh International Conference on Pattern Recognition*, pp. 1101–1103, 1984.
- [23] X. Lin and W. G. Wee, "Shape Detection Using Range Data," in *IEEE International Conference on Robotics and Automation*, vol. 1, pp. 34–39, March 1985.
- [24] F. S. Cohen and R. D. Rimey, "A Maximum Likelihood Approach to Segmenting Range Data," *IEEE Journal of Robotics and Automation* **4**, pp. 277–286, 1988.
- [25] P. K. Besl and R. C. Jain, "Segmentation Through Variable-Order Surface Fitting," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 10, pp. 167–192, March 1988.
- [26] R. W. Taylor, M. Savini, and A. P. Reeves, "Fast Segmentation of Range Imagery into Planar Regions," *Computer Vision, Graphics, and Image Processing* **45**, pp. 42–60, January 1989.

- [27] D. Marr and E. Hildreth, "Theory of Edge Detection," in *Proceedings of the Royal Society of London*, vol. 207, pp. 187–217, 1980.
- [28] J. Canny, "A Computational Approach to Edge Detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **8**, pp. 679–698, November 1986.
- [29] B. Parvin and G. Medioni, "Adaptive Multiscale Feature Extraction from Range Data," *Computer Vision, Graphics, and Image Processing* **45**, pp. 346–356, March 1989.
- [30] H. M. Cung, P. Cohen, and P. Boulanger, "Multiscale Edge Detection and Classification in Range Images," in *IEEE International Conference on Robotics and Automation*, vol. 3, pp. 2038–2044, May 1990.
- [31] G. D. Godin and M. D. Levine, "Structured Edge Map of Curved Objects in a Range Image," in *Proceedings of the IEEE Comp. Soc. Conf. Comp. Vision, Patt. Rec.*, pp. 276–281, June 1989.
- [32] R. Haralick and L. Shapiro, *Computer and Robot Vision*, Addison-Wesley, Reading, MA, 1993.
- [33] J. Altmann and H. Reitbock, "A Fast Correlation Method for Scale and Translation Invariant pattern Recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **6**, pp. 46–57, 1984.
- [34] A. Grun and E. Baltsavias, "Adaptive Least Squares Correlation with Geometrical Constraints," in *Proceedings of the SPIE*, vol. 595, The International Society for Optical Engineering, 1985.
- [35] T. Kanade and M. Okutomi, "A Stereo Matching Algorithm with an Adaptive Window," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **16**, pp. 920–932, 1994.
- [36] P. W. Smith and N. Nandhakumar, "An Improved Power Cepstrum Based Stereo Correspondence Method for Textured Scenes," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **18**(3), pp. 338–348, 1996.
- [37] Perceptron, Inc., *Product Information*, 23855 Research Drive, Farmington Hills, MI 48335, 1994.
- [38] P. Besl, *Surfaces in Image Understanding*, Springer-Verlag, New York, NY, 1986.
- [39] D. L. Milgram, "Computer Methods for Creating Photomosaics," *IEEE Transactions on Computers, Correspondence*, pp. 1113–1119, 1975.
- [40] R. Szeliski, "Video Mosaics for Virtual Environments," *IEEE Computer Graphics and Applications*, pp. 22–30, March 1996.
- [41] R. Szeliski and H. Y. Shum, "Creating Full View Panoramic Image Mosaics and Environment Maps," in *Computer Graphics (SIGGRAPH '97)*, pp. 251–258, August 1997.

- [42] R. C. Gonzalez and P. Wintz, *Digital Image Processing*, Addison-Wesley, Reading, Massachusetts, 1987.
- [43] R. Y. Tsai, "A Versatile Camera Calibration Technique for High-Accuracy 3D Machine Vision Metrology Using Off-the-Shelf TV Cameras and Lenses," *IEEE Journal of Robotics and Automation* 3(4), pp. 323–344, 1987.

VITA

Mark D. Elstrom was born in Bishopville, South Carolina in October of 1974. Shortly after, he moved to Millington, TN where he lived until he graduated from Millington Central High School in 1992. He then began study toward his BSEE at the University of Tennessee, Knoxville. During his undergraduate career, Mr. Elstrom's course of study centered on two main areas: controls and computer vision. Upon receiving his BS degree in May, 1996, he began coursework toward an MSEE degree while serving as a Graduate Teaching Assistant. The following year, he took a position as a research assistant for Dr. Mongi A. Abidi in the Imaging, Robotics and Intelligent Systems Laboratory. The research performed for his thesis work was done under the direction of both Dr. Philip W. Smith and Dr. Mongi A. Abidi. He is expected to graduate in August 1998 with specializations in image processing and computer vision.