

**Laser Rangefinder Calibration
for a Walking Robot**

Eric Krotkov

CMU-RI-TR-90-30

The Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213

December 1990

©1990 Carnegie Mellon University

This research was sponsored by NASA under Grant NAGW-1175. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the funding agency.

Contents

1	Introduction	1
2	Problem Definition	4
3	Target Location in Body Coordinates	5
3.1	Single-Leg Prototype	5
3.2	Six-Legged Ambler	6
4	Target Location in Sensor Coordinates	6
4.1	Erim and Perceptron Sensors	6
4.2	Target Location from Erim Imagery	9
4.3	Target Location from Perceptron Imagery	12
5	Identification of the Transformation	16
6	Results	18
6.1	Experimental Setup and Procedure	18
6.2	Accuracy	20
6.3	Precision	23
6.4	Execution Time	24
7	Discussion	25
A	Grassfire Transform	25
	Acknowledgements	26
	References	26

List of Figures

1	Experimental setup with one-legged robot	2
2	Experimental setup with six-legged robot	3
3	Reference frames for one-legged robot	5
4	Image analysis constants	7
5	Reference frame \mathcal{S}	8
6	Erim images of prototype leg	9
7	Thresholded Erim range image of leg (β_{thresh})	10
8	Grassfire transforms of thresholded Erim range image	11
9	Perceptron images of Ambler leg	13
10	Thresholded Perceptron range image of leg (β_{leg})	14
11	Perceptron range image of top of leg (β_{top})	15
12	Processed Perceptron reflectance image of top of leg (α_{thresh})	16
13	Typical errors, for Erim	21
14	Statistics of error distributions	22
15	Variation of the computed parameters between data sets	23

Abstract

In rugged terrain, walking robots that select footholds can be more mobile and more energy efficient than machines that roll on wheels or crawl on tracks. To achieve these footholds requires calibration of the terrain sensors with respect to the walking mechanism.

We present an implemented technique to calibrate scanning laser rangefinders to legged robots. The procedure accommodates two scanners, one manufactured by Erim and the other by Perceptron, and two walkers, a one-legged robot and a six-legged robot. The technique acquires two sets of corresponding three-dimensional points and identifies the rigid transformation that maps one onto the other with least squared error, i.e., it solves the absolute orientation problem.

We report experimental results with the two different scanners and vehicles. For the Erim and the one-legged robot, the technique achieves an accuracy of 6–12 cm with a precision no lower than 2–5 cm. For the Perceptron and the Ambler, the accuracy is 2–7 cm with a precision no lower than 2–5 cm. These results have proven to be satisfactory for constructing terrain maps and using them to select footholds during our rough terrain walking experiments.

1 Introduction

In order to act autonomously and intelligently, mobile robots must be able to sense their environment, and to relate the sensor readings to their actions. For example, threading a needle requires coordination of the eye, or whatever senses the relative positions of needle and thread, and the hand, or whatever acts on them. Similarly, walking requires coordination of the eye, which senses where on the terrain to place the foot, and the leg, or whatever supports and propels the robot. This paper presents a technique to establish such “leg-eye” coordination for a one-legged robot (Figure 1) and for a six-legged robot (Figure 2), the Ambler prototype planetary rover [2].

Recently, researchers have questioned the need for calibration, and techniques to avoid it have gained favor. In the case of the Ambler, it is likely that we too could survive without calibration; because the machine is big, rugged, and heavy, many objects that are obstacles *before* the Ambler steps on them are *planar* afterwards. But for missions to distant, rugged regions like planetary surfaces, Antarctica, and the ocean floor, survival is not enough. *Energy-efficient* locomotion is essential. With a calibrated sensor, the Ambler can select where to step, and thus can prevent spending significant fractions of the total power budget on stumbling rather than productive advance. This ability to select footholds is central to the fundamental advantages of high mobility and energy efficiency that walkers enjoy over rolling and crawling machines. Achieving those footholds requires calibration.

Other walking robots face the same requirements. However, walkers that rely on a human operator to designate footholds, such as the Adaptive Suspension Vehicle [12], do not require an automated solution. Other researchers [3, 6, 14] have concentrated on range imaging for autonomous outdoor navigation. Although this work has advanced the state of the art in range image understanding, it does not directly address the calibration issues raised here. The photogrammetry literature does address those issues, and we will make use of it in Section 5.

In this paper we present a single calibration procedure that works for multiple scenarios; the same code calibrates the Erim scanner with respect to a one-legged robot, and calibrates the Perceptron scanner with respect to the six-legged Ambler.

The paper is organized as follows. In Section 2 we define the problem. Next, we describe in detail how to acquire two three-dimensional point sets, one in a vehicle-centered reference frame (Section 3), the other in a sensor-centered reference frame (Section 4). Then we show how to identify the rigid transformation that best relates the two point sets, i.e., we present a solution to the absolute orientation problem. In Section 6 we report experimental results on accuracy, precision, and execution time. We conclude by discussing possible improvements and extensions.



Figure 1: Experimental setup with one-legged robot

The figure shows the prototype leg, the calibration target on the upper leg, and the Erim laser scanner mounted above the leg.

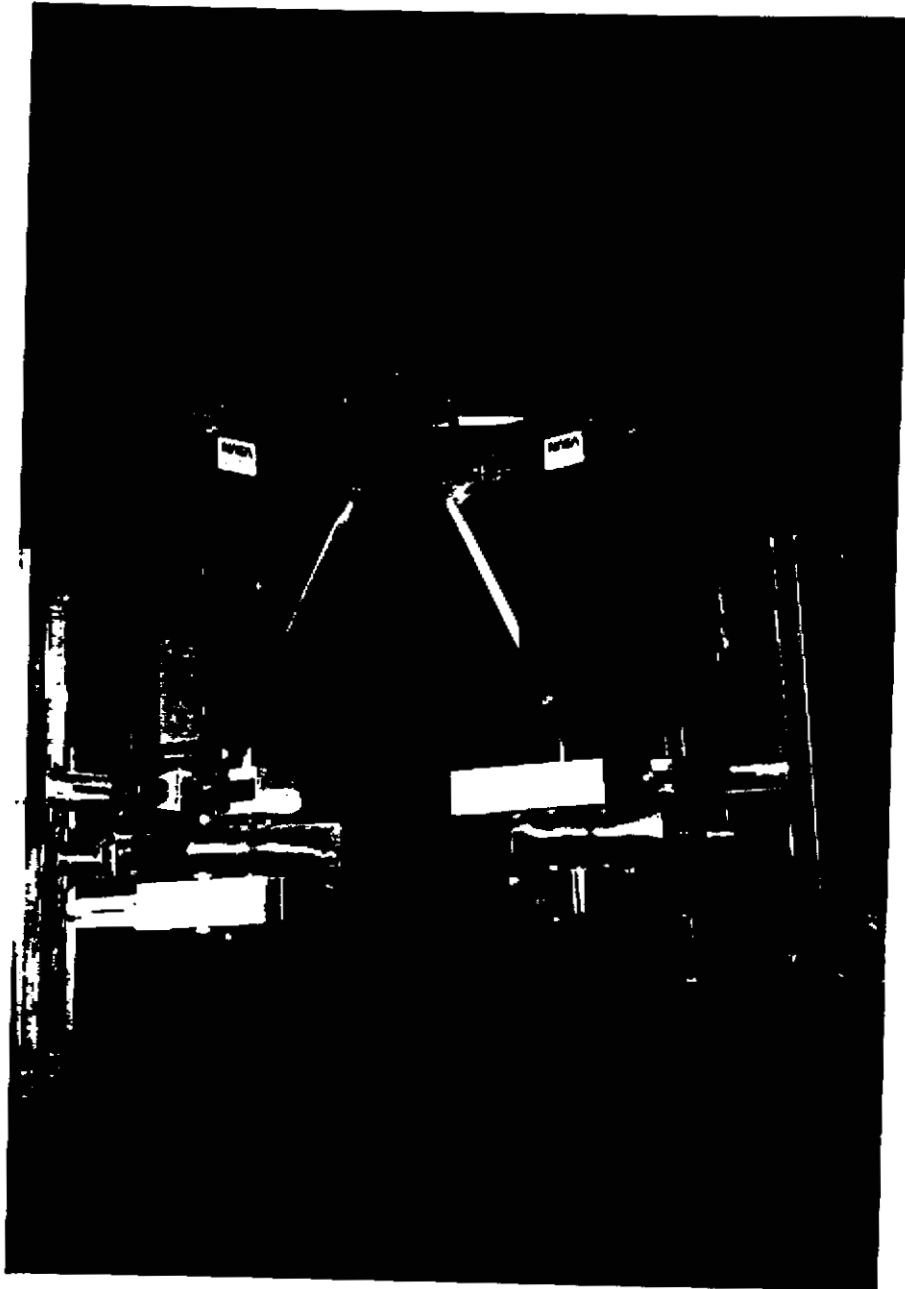


Figure 2: Experimental setup with six-legged robot

The figure shows the Ambler, and the Perceptron laser scanner mounted on the bridge between the two leg stacks. The calibration targets, mounted on the top face of the vertical leg links, are not visible.

2 Problem Definition

The overall problem is to identify the rigid transformation relating a vehicle-centered reference frame to a sensor-centered reference frame. The origin of the *scanner frame* \mathcal{S} is attached to the scanner and lies somewhere nearby it. The coordinates of a point referred to this frame will be written in lower-case, e.g., $\vec{r}_S = [x, y, z]^T$. The origin of the *body frame* \mathcal{B} is attached to the walking robot. The coordinates of a point referred to this frame will be written in upper-case, e.g., $\vec{r}_B = [X, Y, Z]^T$, to distinguish it from points in the scanner frame.

We attach a number T of targets to the legs. Then, we move the legs to a number L of different stations. At each, we identify the position \vec{r}_B of each target in the body frame (by reading joint positions and using known kinematics, see Section 3), and we identify the position \vec{r}_S of each target in the scanner frame (by image analysis, see Section 4). After acquiring a sufficient number of pairs of measurements, we seek the rotation \mathbf{R} and translation \vec{t} that refer a vector in \mathcal{S} to \mathcal{B} :

$$\vec{r}_{B_i} = \mathbf{R}\vec{r}_{S_i} + \vec{t}, \quad 1 \leq i \leq L \times T, \quad (1)$$

where

- $\vec{t} = [t_x, t_y, t_z]^T$ is the translation vector relating the two origins, and
- \mathbf{R} is a 3×3 rotation matrix ($\det \mathbf{R} = +1$, $\mathbf{R}^T \mathbf{R} = \mathbf{I}$).

In practice, it is unlikely that \mathbf{R} and \vec{t} exist that satisfy Equation 1, because measurements are not exact and may be contaminated by noise. Instead, we seek \mathbf{R} and \vec{t} that best satisfy Equation 1 in the least-squares sense.

Problem Statement Find \mathbf{R} and \vec{t} minimizing the sum of squares of errors

$$E = \sum_{i=1}^{L \times T} \|\vec{e}_i\|^2, \quad (2)$$

where

- $\|\vec{x}\|^2 = \vec{x} \cdot \vec{x}$ is the square of the length of the vector \vec{x} , and
- the error of the i^{th} pair of measurements is $\vec{e}_i = \vec{r}_{B_i} - \mathbf{R}\vec{r}_{S_i} - \vec{t}$.

This problem is closely related to a number of other problems that arise in photogrammetry and computer vision. Given the pairs of measurements, the problem is equivalent to the *absolute orientation* problem in photogrammetry, and to the *exterior orientation* part of the *camera calibration* problem in computer vision (see Chapter 13 of [7], and references therein).

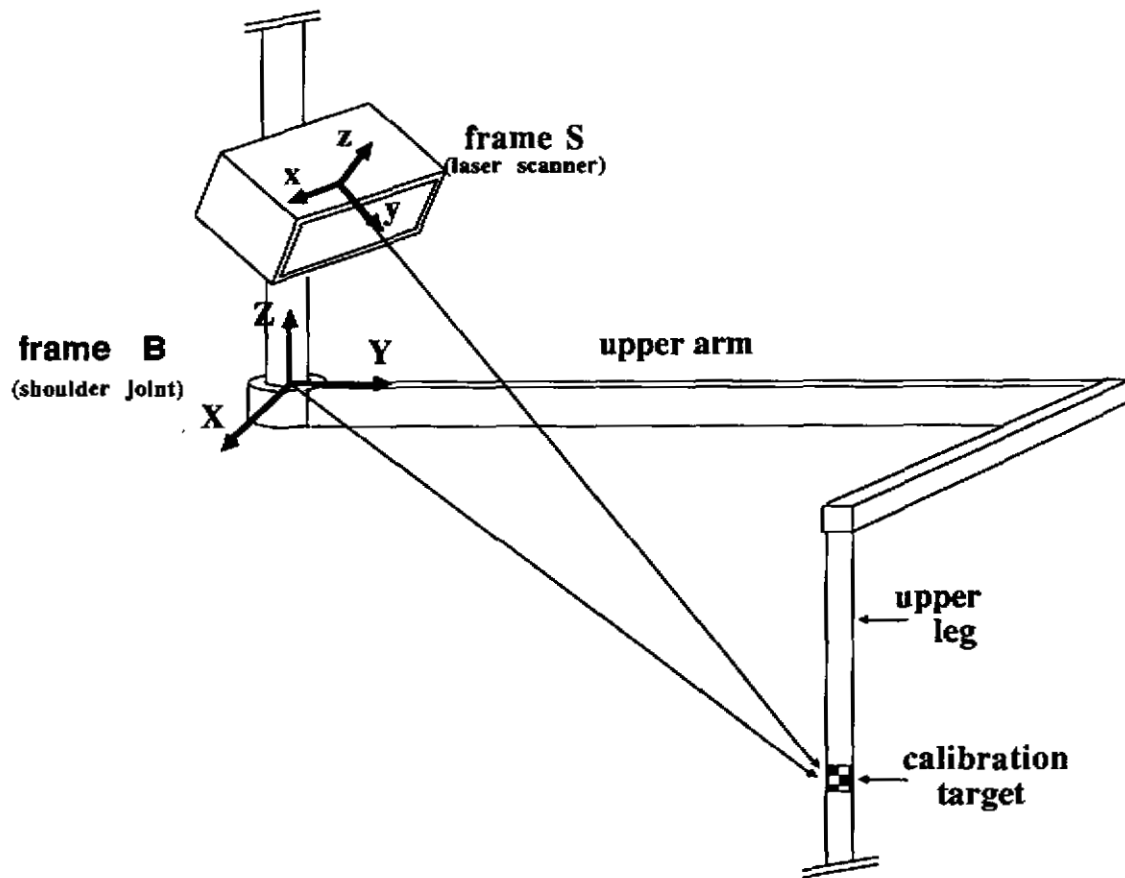


Figure 3: Reference frames for one-legged robot

3 Target Location in Body Coordinates

This section describes the procedure for measuring the target locations \vec{r}_B assumed in the problem statement in Section 2. With the leg at one station, the procedure produces a set of measurements $\{\vec{r}_B\}$, where $\vec{r}_{B_i} = [X_i, Y_i, Z_i]^T, 1 \leq i \leq T$. Repeating the procedure for a number L of stations yields a set of measurements of cardinality $L \times T$.

3.1 Single-Leg Prototype

For the one-legged prototype, the origin of B coincides with the shoulder joint, and lies at the center of the supporting shaft, on the plane of the upper arm (Figure 3). Calibration targets (pieces of reflective tape) are attached to the upper leg.

We now describe how the target coordinates are computed, beginning with the Z -component. Since the upper leg can not move relative to the hip (where the forearm meets the upper leg), the targets lie at fixed and measurable distances from the hip. We manually measure the constant vertical distance from each target to the hip; let us call it Z_{target} . We also manually measure the constant vertical distance from the hip to the origin of \mathcal{B} ; let us call this Z_{hip} . Then the vertical distance from a target to the shoulder is a constant: $Z = Z_{hip} + Z_{target}$. Since all points on the upper leg are below the shoulder (the origin of \mathcal{B}), and the Z -axis points upward, it follows that $Z < 0$.

It remains to determine the X and Y coordinates of a given target. This reduces to the identification of the planar position of the hip from two joint angles and two link lengths, which is an easy kinematics problem whose solution is not discussed here.

3.2 Six-Legged Ambler

For the six-legged Ambler, the origin of \mathcal{B} lies at the center of the downward-facing surface of the structural bridge that connects the two leg stacks and supports the scanner (Figure 2). Calibration targets (pieces of brown paper) are attached to the top of the vertical links.

The joint angles and link lengths are known, so elementary kinematics suffice to determine the target coordinates with respect to \mathcal{B} . Unlike the previous section, the solution is not easy to derive immediately, because the geometry of the Ambler legs is more complex.

4 Target Location in Sensor Coordinates

This section describes the procedure for measuring the target locations \vec{r}_S assumed in the problem statement in Section 2. It begins by describing the Erim and Perceptron sensors and their common reference frame. It then presents a procedure that starts with the sensor images, analyzes them to identify the target locations in image space, maps the image locations to a spherical-polar space, and finally converts these to a Cartesian space. Figure 4 records the constants used for the image analysis.

With the leg at one station, the procedure produces a set of measurements $\{\vec{r}_S\}$, where $\vec{r}_{Si} = [x_i, y_i, z_i]^T, 1 \leq i \leq T$. Repeating the procedure for a number L of stations yields a set of measurements of cardinality $L \times T$.

4.1 Erim and Perceptron Sensors

We consider two scanning laser rangefinders, one manufactured by Erim, and the other by Perceptron. They both are optical-wavelength radar systems that use a laser diode source operating in the near-infrared region (Erim 820 nm, Perceptron 810 nm) that is amplitude modulated and scanned across the field of view using a nodding mirror and a rotating polygon

Constant	Erim	Perceptron	Remark
K_{near}	3.0	-	Minimum target range, in meters.
K_{far}	4.2	4.0	Maximum target range, in meters.
I_{near}	40	-	Range image intensity at K_{near} .
I_{far}	55	400	Range image intensity at K_{far} .
K_{grass}	3	3	Minimum target size, in pixels.
K_{ref}	250	4081	Minimum target intensity, in reflectance image.
K_{top}	-	130	Maximum number of rows top of leg may occupy.

Figure 4: Image analysis constants

mirror. The nodding mirror changes (tilts) the elevation and the polygon mirror changes (pans) the azimuth of the emitted signal. The infrared light is reflected off the desired target, gathered by the receiver optics, and focused onto the detector. The envelope of the output of the detector is at the same frequency as that of the laser source, but is shifted in phase by an amount proportional to the distance traveled by the beam (range). Using a digital phase detector the range is assigned for each pixel along the scan line. The devices digitize two images: a *range image*, with pixel values proportional to phase difference; and a *reflectance image*, with pixel values proportional to reflected energy.

The Erim acquires data in 64×256 pixel images at a rate of 2 Hz [15]. The scanner digitizes to 8 bits over approximately 20 meters, which provides a nominal range resolution of 7.62 cm. The measurements cover 80 degrees in the horizontal direction (azimuth) and 30 degrees in the vertical direction (elevation).

The Perceptron acquires data in 256×256 pixel images at a rate of 2 Hz. The scanner digitizes to 12 bits over approximately 40 meters, which provides a nominal range resolution of 0.98 cm. The measurements cover 60 degrees in azimuth and 60 degrees in elevation.

Figure 5 illustrates the sensor reference frame \mathcal{S} we use for both the Erim and the Perceptron measurements. As shown, the y -axis coincides with the direction of travel of the laser beam projected through the central point of the scanner. The angle θ (azimuth) corresponds to a rotation about the z -axis. The angle ϕ (elevation) corresponds to a rotation about the x -axis.

This reference frame is not a standard spherical-polar system. In a standard spherical-polar reference frame, θ is measured from the positive x -axis in the xy plane, and $\theta > 0$ in the counter-clockwise direction. In the scanner frame, θ is measured from the positive y -axis in the POQ plane, and $\theta > 0$ in the clockwise direction. In a standard spherical-polar reference frame, ϕ is measured from the positive z -axis in the zR plane. In the scanner frame, ϕ is measured from the positive y -axis in the yz plane. In the figure, both θ and ϕ are positive.

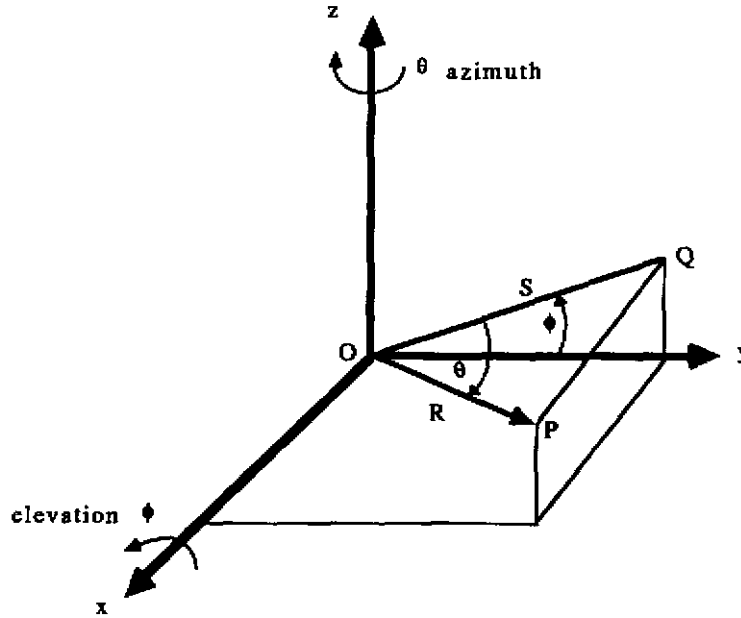


Figure 5: Reference frame S

Both Erim and Perceptron measurements use this coordinate system.

Given the sensor measurement (r, c, d) (i.e., row, column, range), the transformation to spherical-polar coordinates is

$$\phi = r\Delta_\phi + \phi_0, \quad \theta = c\Delta_\theta + \theta_0, \quad \rho = kd, \quad (3)$$

where

- Δ_ϕ is the angular increment, in degrees/row, of the nodding mirror,
- Δ_θ is the angular increment, in degrees/column, of the panning mirror,
- ϕ_0 is the initial orientation, in degrees, of the nodding mirror,
- θ_0 is the initial orientation, in degrees, of the panning mirror, and
- k is the scanner range resolution in meters/bit.

Given the spherical polar coordinates ϕ, θ, ρ , the transformation to Cartesian coordinates is given by

$$x = \rho \sin \theta, \quad y = \rho \cos \theta \cos \phi, \quad z = \rho \cos \theta \sin \phi. \quad (4)$$

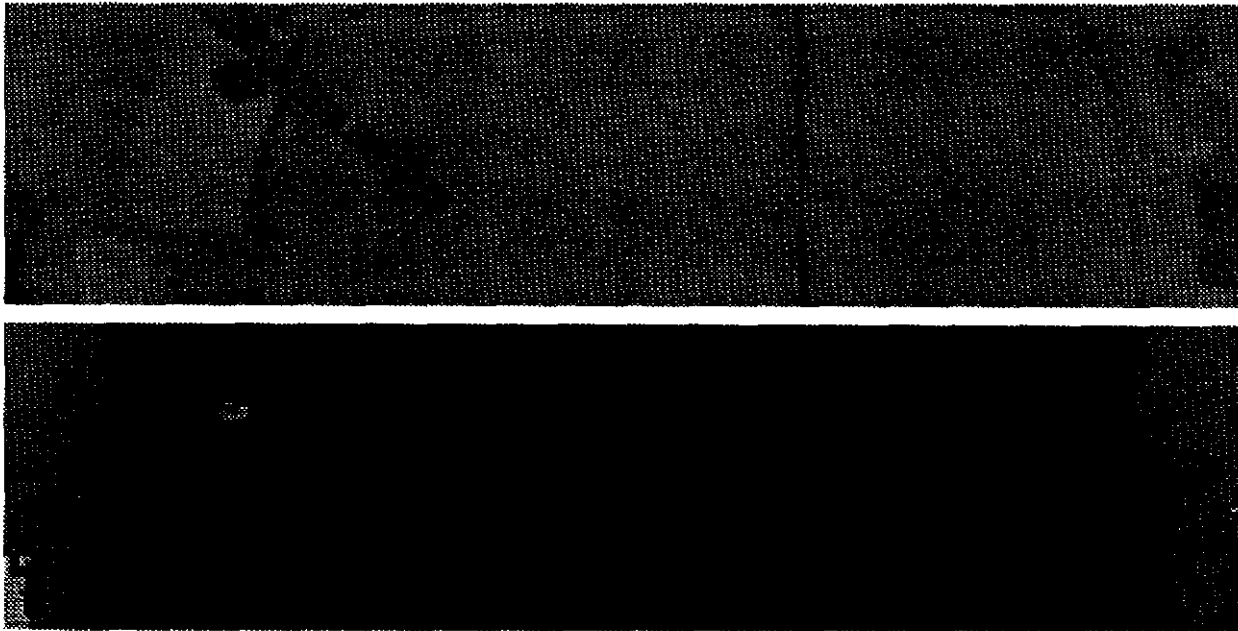


Figure 6: Erim images of prototype leg

The leg appears in the left-hand side of the reflectance (top) and range images as a tapered cylinder. The bump on its lower left side is a cable reel; the cable appears faintly below the reel. In the reflectance image, the calibration target appears as a bright band below the reel. In the center of the images is sand.

4.2 Target Location from Erim Imagery

Given a pair of reflectance and range images $\alpha(u, v)$ and $\beta(u, v)$, as in Figure 6, the task is to compute the image coordinates of the target. The four following steps perform this task.

1. Register the range and reflectance images.

Although the image pair is nominally registered, we find that in practice they are not in perfect registration. In particular, we observe that the range image lags the reflectance image by two horizontal pixels. In this case, we trivially register the images by shifting one of them by two columns.

2. Locate the leg in image space.

Initially, we attempted to locate the upper leg by examining the difference of two images (either reflectance or range) taken with the leg in two different poses. If the only difference between the images is the pose of the leg, then subtracting one image from the other will yield only the pixels that changed from frame to frame, viz., leg pixels. This approach does not work, because pixel values change significantly from

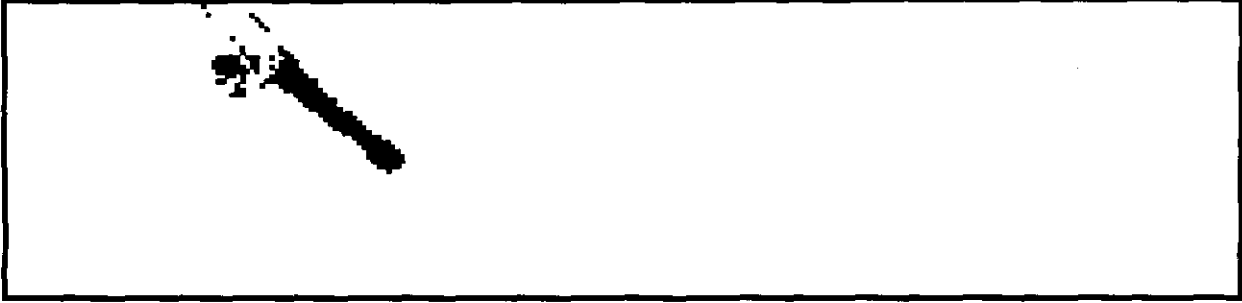


Figure 7: Thresholded Erim range image of leg (β_{thresh})

frame to frame not only because of leg motion, but also because of random noise, and the *mixed pixel* effect [6]. So instead we take the following approach.

- (a) Bound the range to the target. Because the calibration procedure commands the leg, the pose of the leg is known approximately. This establishes an interval $[K_{near}, K_{far}]$ that bounds the range to the target on the leg. This distance interval determines a range image intensity interval $[I_{near}, I_{far}]$ that bounds the pixel values of the target on the leg.
- (b) Threshold the range image, removing pixels with ranges that are either too close or too far to lie on the upper leg in the vicinity of the target(s):

$$\beta_{thresh}(u, v) = \begin{cases} 1 & \text{if } I_{near} \leq \beta(u, v) \leq I_{far}, \\ 0 & \text{otherwise,} \end{cases} \quad (5)$$

where

- $\beta(u, v)$ is the input range image value at pixel (u, v) ,
- I_{near} and I_{far} are derived from the constants K_{near} and K_{far} .

Figure 7 illustrates the results of the thresholding operation. It shows that the output binary image contains regions that do not belong to the leg, and is noisy.

- (c) Filter the thresholded range image by first shrinking and then growing, in order to eliminate regions that are too small to be the target. To eliminate small regions, threshold (*shrink*) the output of the Grassfire transform as follows:

$$\beta_{shrink}(u, v) = \begin{cases} G(\beta_{thresh}(u, v)) & \text{if } G(\beta_{thresh}(u, v)) \geq K_{grass}, \\ 0 & \text{otherwise,} \end{cases} \quad (6)$$

where

- G is the forward Grassfire transform (defined in Appendix A),

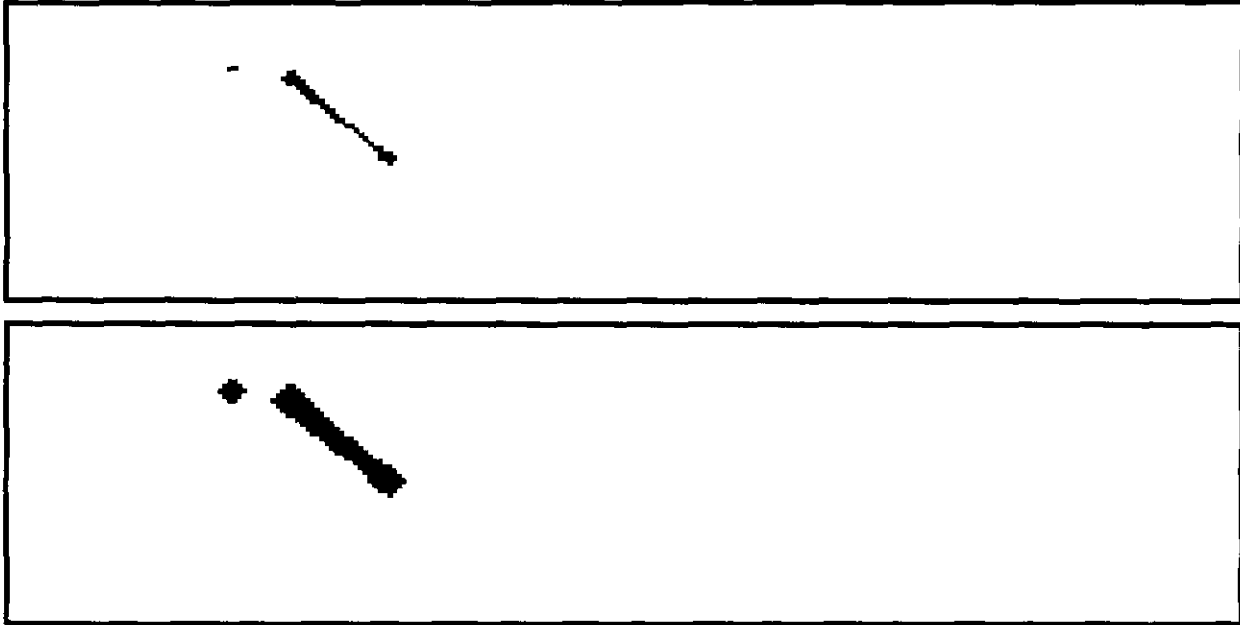


Figure 8: Grassfire transforms of thresholded Erim range image
 The results of the forward transform appear in the top panel, and the results of the reverse transform appear in the bottom.

- $\beta_{thresh}(u, v)$ is given by Equation 5,
- K_{grass} is the minimum size of the target, in pixels.

To restore (*grow*) the target region to its original size, apply the reverse Grassfire transform as follows:

$$\beta_{leg}(u, v) = G^{-1}(\beta_{shrink}(u, v)) , \quad (7)$$

where

- G^{-1} is the reverse Grassfire transform (defined in Appendix A), and
- $\beta_{shrink}(u, v)$ is given by Equation 6.

Figure 8 illustrates the results of applying the forward and reverse Grassfire transforms to the thresholded image shown in Figure 7.

3. Identify target pixels on the leg.

The target is distinguishable from the leg only in the reflectance image. The range image is of no utility in identifying those leg pixels that belong to the target. The procedure to identify the image coordinates of the target is as follows.

Since the target is chosen to be made of a material that provides sharp contrast to the leg, it is possible to identify target pixels by thresholding on reflectance intensity. We take the target pixels to be those pixels that both belong to the leg, and exceed an intensity threshold, as follows:

$$\alpha_{target}(u, v) = \begin{cases} 1 & \text{if } \beta_{leg}(u, v) \neq 0 \text{ and } \alpha(u, v) > K_{ref}, \\ 0 & \text{otherwise,} \end{cases} \quad (8)$$

where

- $\alpha(u, v)$ is the input reflectance image value at pixel (u, v) ,
- $\beta_{leg}(u, v)$ is given by Equation 7, and
- K_{ref} is a constant intensity threshold.

4. Compute the centroid (\bar{u}, \bar{v}) of the target pixels in α_{target} .

The completed image analysis computes the following parameters of the target: row \bar{u} ; column \bar{v} ; and sensed range $\beta(\bar{u}, \bar{v})$.

4.3 Target Location from Perceptron Imagery

The method described in this section is similar to the procedure detailed in the previous section. However, the order and type of operations differ enough to warrant a complete description, at the expense of some redundancy.

The six following steps compute the image coordinates of the target.

1. Acquire a temporal sequence of pairs of reflectance and range images $\alpha_i(u, v)$ and $\beta_i(u, v)$, as in Figure 9. Apply a median filter to the range images, computing $\beta(u, v)$. Arbitrarily select one of the $\alpha_i(u, v)$ to be $\alpha(u, v)$.

We apply the median filter because of its robustness in the presence of outliers.

No filtering is necessary for the procedure described in the previous section. However, we find it to be required here. This might be due to the finer resolution of the Perceptron, the larger distances involved with the Ambler testbed, or other factors.

2. Locate the leg in image space.
 - (a) Threshold the range image, using Equation 5 to compute $\beta_{thresh}(u, v)$.
 - (b) Find the largest connected component \mathcal{C} in $\beta_{thresh}(u, v)$, and create a binary image of it as follows (Figure 10):

$$\beta_{leg}(u, v) = \begin{cases} 1 & \text{if } (u, v) \in \mathcal{C}, \\ 0 & \text{otherwise.} \end{cases} \quad (9)$$

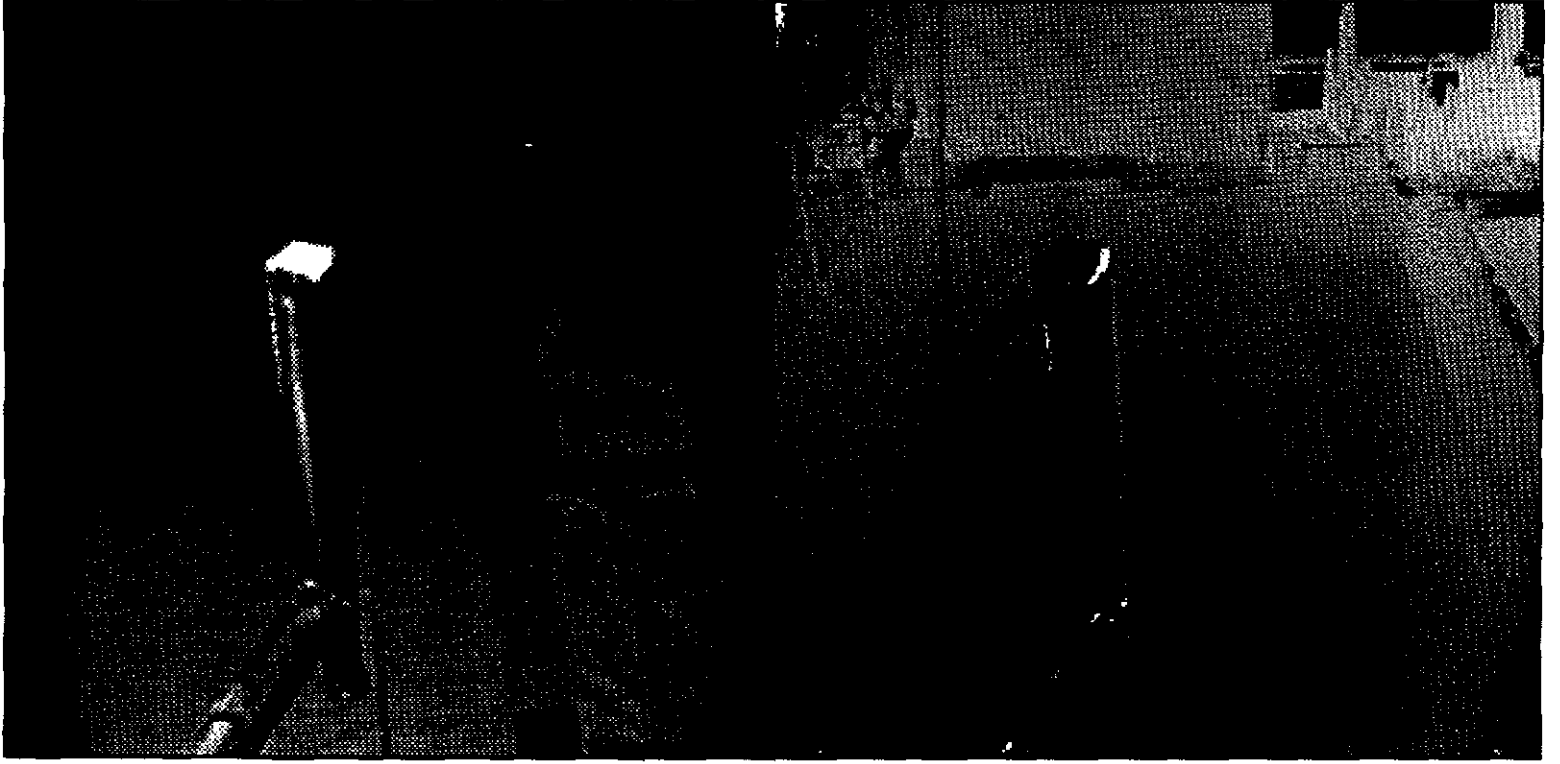


Figure 9: Perceptron images of Ambler leg

The contrast of the reflectance (left) and range images has been enhanced. The extensional link of the leg appears in the bottom left, and the vertical link appears in the center. In the center of the reflectance image, the calibration target is visible as a white patch at the top of the vertical link.

3. Find the top of the leg, as follows (Figure 11):

$$\beta_{top}(u, v) = \begin{cases} 1 & \text{if } \beta_{leg}(u, v) \neq 0 \text{ and } u - u_0 \leq K_{top}, \\ 0 & \text{otherwise,} \end{cases} \quad (10)$$

where

- $\beta_{leg}(u, v)$ is given by Equation 9,
- u_0 is the smallest u for which $\beta_{leg}(u, v) \neq 0$ (i.e., the highest point on the leg), and
- K_{top} is the maximum number of rows the leg top may extend in the image.

4. Identify target pixels at the top of the leg.

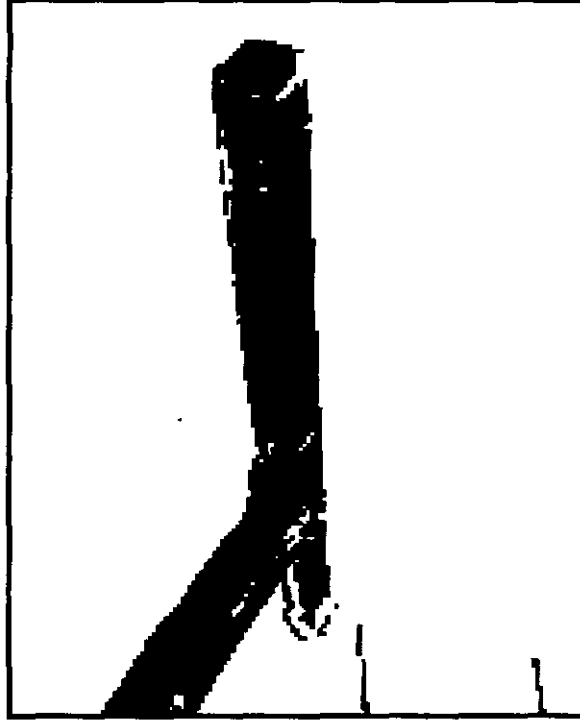


Figure 10: Thresholded Perceptron range image of leg (β_{leg})

This typical result includes pixels that do not belong to the leg, and does not include some pixels that do belong to the leg.

- (a) Identify potential target pixels $\alpha_{thresh}(u, v)$ as follows (Figure 12):

$$\alpha_{thresh}(u, v) = \begin{cases} 1 & \text{if } \beta_{top}(u, v) \neq 0 \text{ and } \alpha(u, v) > K_{ref}, \\ 0 & \text{otherwise,} \end{cases} \quad (11)$$

where

- $\alpha(u, v)$ is the input reflectance image value at pixel (u, v) ,
 - $\beta_{top}(u, v)$ is given by Equation 10, and
 - K_{ref} is a constant intensity threshold.
- (b) Filter this processed reflectance image by first shrinking and then growing, in order to eliminate pixels that belong to regions that are too small or too elongated to belong to the target. To eliminate small regions, threshold the output of the Grassfire transform as follows:

$$\alpha_{shrink}(u, v) = \begin{cases} G(\alpha_{thresh}(u, v)) & \text{if } G(\alpha_{thresh}(u, v)) \geq K_{grass}, \\ 0 & \text{otherwise,} \end{cases} \quad (12)$$

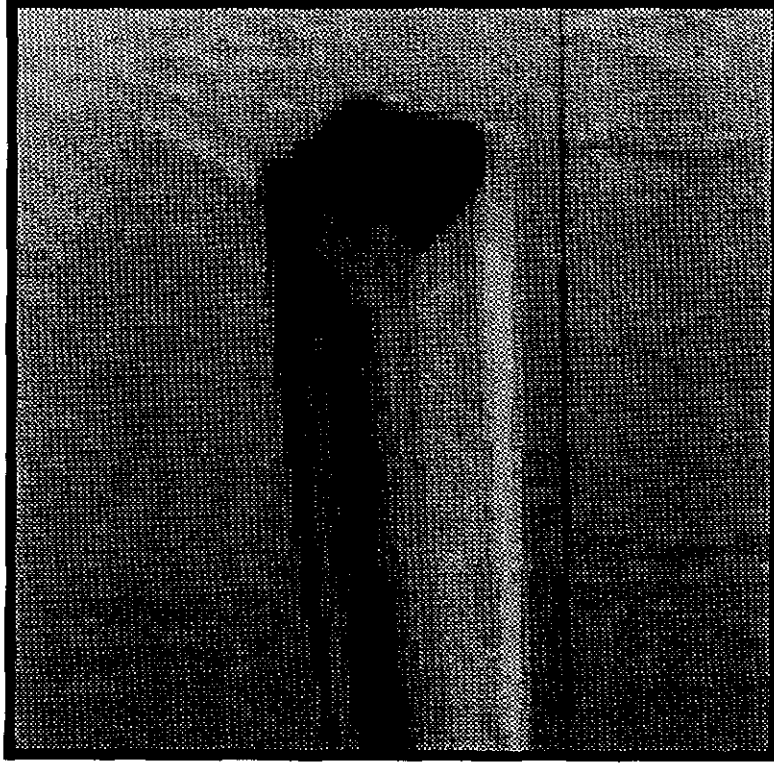


Figure 11: Perceptron range image of top of leg (β_{top})

The contrast of this image has been enhanced and inverted. Note the vertical bands on the left and right sides of the leg. The left bands are due to variations in material where the rack (i.e., part of the rack-and-pinion drive) meets the aluminum leg structure. The right bands are “ghosts” that occur immediately to the right of the image of a depth discontinuity.

where

- G is the forward Grassfire transform (defined in Appendix A),
- $\alpha_{thresh}(u, v)$ is given by Equation 11,
- K_{grass} is the minimum size of the target, in pixels.

To restore the target region to its original size, apply the reverse Grassfire transform as follows:

$$\alpha_{target}(u, v) = G^{-1}(\alpha_{shrink}(u, v)) , \quad (13)$$

where

- G^{-1} is the reverse Grassfire transform (defined in Appendix A), and
- $\alpha_{shrink}(u, v)$ is given by Equation 12.

The results of applying these transforms are represented well by Figure 8, and so



Figure 12: Processed Perceptron reflectance image of top of leg (α_{thresh})

This figure illustrates the results of applying Equation 11 to the image shown in Figure 11. The bands that appear on the right side of the leg in Figure 11 have been removed, but the bands on the left remain.

are not shown. It suffices to note that the transforms do remove the bands shown in Figure 12.

5. Compute the centroid (\bar{u}, \bar{v}) of the target pixels in α_{target} .
6. Compute the median range value γ in a small, typically 3×3 , spatial neighborhood around (\bar{u}, \bar{v}) . This provides further protection against outliers.

The completed image analysis computes almost the same parameters of the target as in the previous section: row \bar{u} ; column \bar{v} ; and median range γ .

5 Identification of the Transformation

This section presents our solution to the least-squares absolute orientation problem. As discussed in Section 2, the problem is to identify the rigid transformation minimizing the

squared error in mapping the point set $\{\vec{r}_S\}$ into $\{\vec{r}_B\}$.

The problem is non-trivial because there are more unknowns than equations. There are $3LT$ scalar equations (Equation 1). The number of unknowns is at least six (three translation vector coefficients plus at least three independent variables to parameterize the rotation) and at most twelve (three translation coefficients plus nine — not independent — rotation matrix coefficients).

There are a number of approaches to solving this under-determined system. We will briefly review them before presenting our solution method.

First, the problem can be solved using Equation 1 and at least four corresponding points. With four points there are twelve equations, enough to solve for the maximum number of twelve unknowns. The equations are linear, so unless the points are coplanar, the existence and uniqueness of a solution are guaranteed. The disadvantage of this approach is that it does not guarantee that \mathbf{R} is orthonormal.

Second, the above difficulty can be circumvented by using Equation 1, and in addition enforcing the orthonormality constraints on the rotation matrix. Only three points are necessary (in fact, more than sufficient), but the equations are non-linear, leading to a constrained minimization problem that must be solved iteratively, for instance, using Lagrange multipliers.

Third, the problem can be solved when three points are measured by selectively neglecting the additional constraints available from the three measurements [11, 13]. This approach applies only to the case of three points, no more and no less, and thus lacks generality and robustness.

Fourth, exact closed-form (non-iterative) solutions are possible using an orthonormal matrix to represent the rotation. One solution uses the singular-value decomposition of an arbitrary matrix [1], while another uses the eigenvalue-eigenvector decomposition of a symmetric matrix [9]. The disadvantage of these approaches is the complexity involved with the six nonlinear constraints that ensure that the matrix is orthonormal.

Fifth, exact closed-form solutions are possible using unit quaternions to represent the rotation. The solution for the desired quaternion is the eigenvector associated with an eigenvalue (either the smallest [5] or largest [8]) of a symmetric matrix, whose elements are combinations of sums of products of corresponding coordinates. The quaternion representation affords two advantages: simplicity — it is simpler to enforce a unit norm for a quaternion than it is to ensure that a matrix is orthonormal; and closed-form solution — no iteration is required.

We prefer the latter approach because of its elegance and effectiveness, and implement the technique in [5]. For convenience in the physical interpretation of the solution rotation, we do not use the solution quaternion \hat{q} itself. Instead, we first express \hat{q}_{min} as an orthonormal matrix \mathbf{R} using the equations in [8], p. 641, and then parameterize \mathbf{R} under the roll-pitch-yaw convention. Let ϕ correspond to roll, θ correspond to pitch, and ψ correspond to yaw.

We specify the order of rotation as

$$RPY(\phi, \theta, \psi) = Rot(Z, \phi) Rot(Y, \theta) Rot(X, \psi) ,$$

that is, a rotation of ψ about X , followed by a rotation θ about Y , and finally, a rotation ϕ about Z . The orthonormal matrix corresponding to $RPY(\phi, \theta, \psi)$ is given in [10], p. 47.

Note that the identified transformation maps a vector referred to frame \mathcal{S} into a vector referred to frame \mathcal{B} . This is the *covariant* transformation. The *contravariant* transformation maps frame \mathcal{S} into frame \mathcal{B} ; it is the inverse of the covariant transformation, and transforms frames, not vectors. Because the contravariant corresponds more closely to our physical interpretation of the rotation, our discussion of the rotation parameters in the sequel will always be in terms of the contravariant transformation. Specifically, we give the roll, pitch, and yaw parameters of the contravariant rotation and the negative translation parameters of the covariant. This corresponds to rotating frame \mathcal{S} about the axes of the initial \mathcal{S} , and translating the rotated frame along the axes of \mathcal{B} .

We have implemented an optional, second stage to cope with unexpected behavior by the sensor. Typically, we observe such behavior when operating the sensor under conditions that exceed its operating range, e.g., high temperature. This second stage searches for two parameters, gain k_1 and offset k_2 , that minimize the squared error E . We replace the expression $\rho = kd$ in Equation 3 by

$$\rho = k_1 d + k_2 . \tag{14}$$

At each iteration, the second stage invokes the exact solution procedure described above, using the revised sensor model given by Equation 14.

6 Results

In this section we present experimental results from the calibration procedure. First, we describe the experimental setup and state the experimental procedure. Next, we quantify the accuracy of the procedure, and discuss its utility for building terrain maps and choosing footholds. Then, we analyze the precision of the procedure for various trials.

6.1 Experimental Setup and Procedure

Figures 1 and 2 illustrate the experimental setup. Observing the physical pose of the scanner in the two figures, we see that it sits above and in front of the supporting structures, and that it looks down toward the ground. Thus, to map the scanner frame into the body frame, we would first rotate it a substantial amount about the scanner x -axis (tilt it up from looking at the ground to look at the horizon), and then translate it along the negative body Y -axis

(back) and along the negative body Z -axis (down). According to the convention established in the previous section, this mapping is equivalent to the following rigid motion parameter values:

$$t_x \approx 0, t_y < 0, t_z < 0, \phi \approx 0, \theta \approx 0, \psi \gg 0.$$

To formally state the experimental procedure, let i and j be counters, B and S be sets of target position vectors, L be the number of leg poses, and T be the number of targets on each leg.

1. $i \leftarrow 1, B \leftarrow \emptyset, S \leftarrow \emptyset$
2. while $i \leq L$ do
 - (a) Move leg to pose i
 - (b) Compute \vec{r}_{Bij} , $1 \leq j \leq T$, as in Section 3
 - (c) $B \leftarrow B \cup \{\vec{r}_{Bij}\}$, $1 \leq j \leq T$
 - (d) Acquire reflectance and range images
 - (e) Compute \vec{r}_{Sij} , $1 \leq j \leq T$, as in Section 4
 - (f) $S \leftarrow S \cup \{\vec{r}_{Sij}\}$, $1 \leq j \leq T$
 - (g) $i \leftarrow i + 1$
3. Compute \mathbf{R} and \vec{i} from B and S , as in Section 5

We have automated this procedure entirely, and have executed it hundreds of times, perhaps a thousand times. Over the course of these trials, we have tested the procedure under a wide variety of conditions, including 50°F differences in temperature, ambient illuminations ranging from bright sunlight to night-time fluorescent lighting, and background materials ranging from sand to people to heavy machinery.

Typical values from one Erim and one Perceptron data set, respectively, are the following, in units of meters and degrees:

$$t_x = 0.2, t_y = -0.6, t_z = -0.8, \phi = -0.5, \theta = 1.8, \psi = 75.1,$$

$$t_x = 0.0, t_y = -0.1, t_z = -1.0, \phi = 0.4, \theta = -0.3, \psi = 41.7.$$

These rigid motion parameters agree with our expectations, without exception, and with each other, except for the rotation about the x -axis.

6.2 Accuracy

Accuracy of Rigid Motion Parameters

It would be revealing to compare the computed rigid motion parameters obtained by our calibration procedure with those measured by direct techniques. However, the origin of \mathcal{S} does not physically lie at a distinguished position that we can measure directly, i.e., with devices such as calipers or rulers. Thus, such a comparison is not possible.

Accuracy of Points Transformed into \mathcal{B}

An alternative basis for evaluating the accuracy of the computed parameter values is the distance between corresponding points after applying the transformation. We consider two distances.

The first distance we consider is related to what we earlier called “error.” Recall from Section 2 that the error of the i^{th} pair of measurements is

$$\vec{e}_i = \vec{r}_{B_i} - \mathbf{R}\vec{r}_{S_i} - \vec{t} .$$

To illustrate the distribution of errors, Figure 13 plots three projections of the \vec{e}_i for a typical Erim data set. The errors for a typical Perceptron data set are similar, and so are not shown. The figure suggests that the errors tend to cancel. For example, in the front view, the magnitude of the errors in the positive Z direction approximately equals the magnitude of the errors in the negative Z direction. The figure also indicates the absence of outliers.

The second distance we consider is the square root of the sum of component-wise differences. Here, the distance d_i between the i^{th} pair of measurements is given by

$$d_i^2 \equiv \vec{e}_i \cdot \vec{e}_i , \tag{15}$$

and \bar{d} denotes the mean of the distribution of the d_i . Graphically, \bar{d} is the average length of the line segments in Figure 13.

Figure 14 shows \bar{d} for ten Erim trials. We observe that the mean varies from 5.5 to 11.2 cm, and that it varies significantly between data sets of different sizes, and between data sets of the same size. This range of values and variations is typical of other trials. We conclude that the accuracy of the calibration procedure for the Erim is 6–12 cm.

Figure 14 also shows \bar{d} for ten Perceptron trials. The results show that the mean values range from 1.8 to 6.7 cm, and vary significantly between data sets. This range of values and variations is typical of other trials. We conclude that the accuracy of the calibration procedure for the Perceptron is 2–7 cm, or two times greater than for the Erim. This difference in accuracy may be due to better leg position sensing with the Ambler, or superior accuracy on the part of the Perceptron scanner, or both.

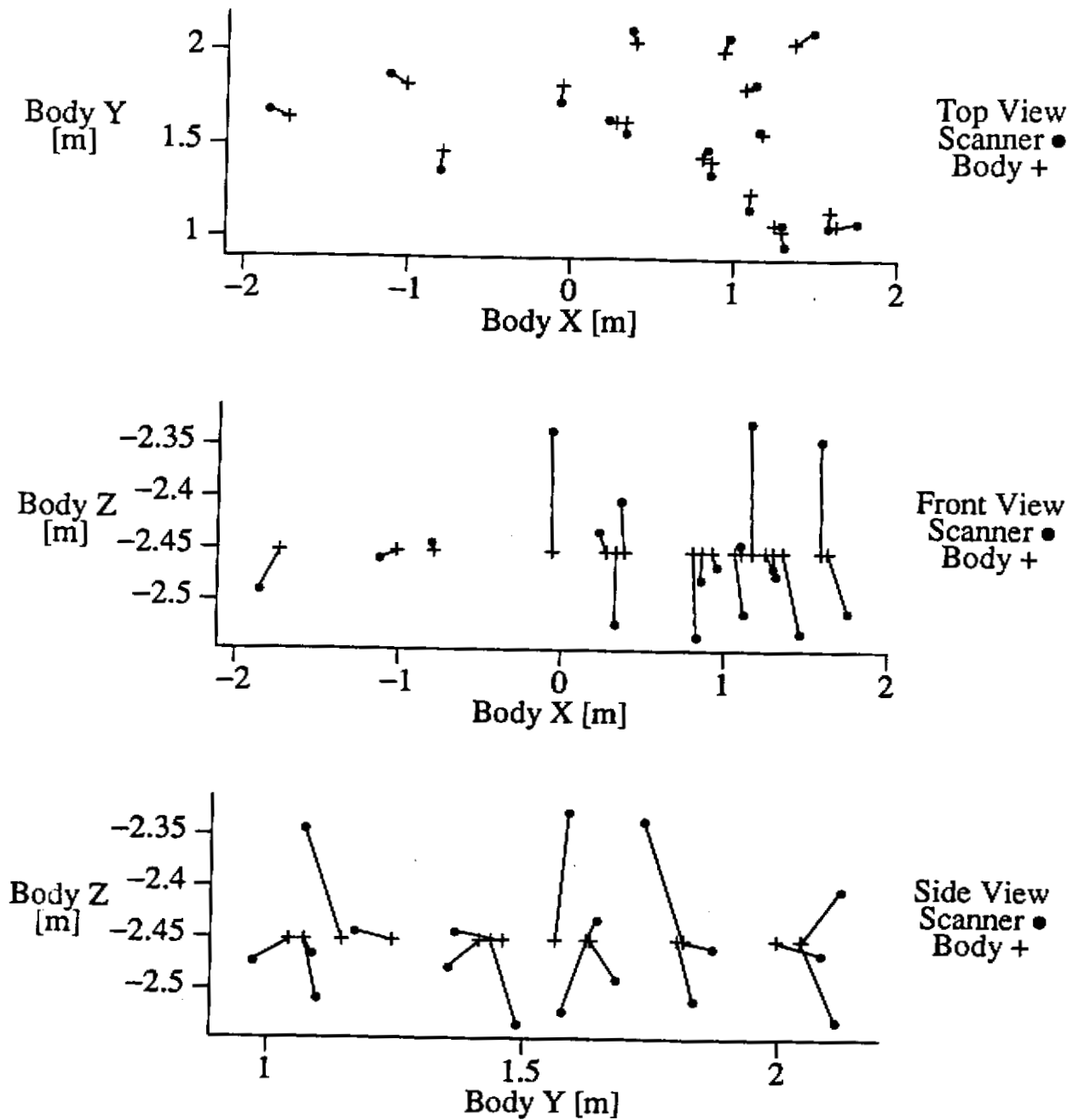


Figure 13: Typical errors, for Erim

The three two-dimensional plots are top, front, and side views. The cross symbol represents the position measured in \mathcal{B} ; the bullet symbol represents the position measured in \mathcal{S} and then referred to \mathcal{B} ; the lines connect corresponding measurements.

Trial	N	\bar{d} (cm)	s_D (cm)	Trial	N	\bar{d} (cm)	s_D (cm)
1	8	5.5	2.4	11	12	2.5	1.0
2	8	5.7	2.4	12	12	2.2	1.3
3	8	6.4	1.5	13	12	6.7	4.3
4	8	5.8	1.9	14	12	2.9	1.6
5	18	9.6	4.3	15	12	6.7	3.8
6	18	10.1	3.1	16	12	6.0	3.2
7	18	8.6	3.8	17	12	4.0	3.0
8	50	11.2	3.6	18	12	2.7	1.5
9	50	8.9	2.9	19	12	1.8	1.9
10	50	10.7	4.3	20	12	2.9	1.5

Figure 14: Statistics of error distributions
Trials 1-10 use the Erim, and trials 11-20 use the Perceptron.

Accuracy of Maps

Another measure of the computed rigid motion parameters is their utility for building terrain maps from which footholds can be selected.

To achieve high map accuracy we have added a third stage to the calibration procedure described in Section 5. This stage searches for the vertical translation t_z that produces maps with elevations closest to a given elevation (often, it is convenient to use the elevation of the ground). This compensates for changes in the scanner performance caused by factors such as range drift and temperature variation, without requiring acquisition of new data.

We have not yet determined the accuracy of the maps to our complete satisfaction (that will be the subject of a future report). Rather than reserve comment, in the following we report results that are approximate and representative, but not definitive. By *relative* accuracy we mean accuracy in determining relative positions, for instance, to compute the dimensions of an object. By *absolute* accuracy we mean accuracy in determining absolute positions, for instance, to compute the position of an object in some external reference frame.

For both the Erim and Perceptron, using the third stage, we reliably achieve a relative accuracy of 5-10 cm, and an absolute accuracy of 10-20 cm. We believe that some improvement is still possible. Nevertheless, this degree of fidelity has proved to be adequate for all of the rugged terrain walking experiments conducted to date.

Trial	N	Δt_x (cm)	Δt_y (cm)	Δt_z (cm)	$\Delta \phi$ (deg)	$\Delta \theta$ (deg)	$\Delta \psi$ (deg)
A,B	18	16.1	7.3	-6.5	0.4	2.0	-0.8
B,C	18	-3.4	10.6	-0.2	0.1	-0.5	1.4
A,C	18	12.7	-3.2	-6.7	0.6	1.5	0.6
D,E	12	7.4	3.5	2.1	1.6	0.8	-0.7
E,F	12	-3.8	-3.6	-5.7	-0.7	-0.5	1.6
D,F	12	3.6	-0.1	-3.6	0.8	0.3	0.7

Figure 15: Variation of the computed parameters between data sets Trials A–C use the Erim, and Trials D–F use the Perceptron.

6.3 Precision

Precision of Rigid Motion Parameters

To evaluate the precision of the computed rigid motion parameters, we execute the calibration procedure several consecutive times, and observe the difference in the computed parameters. Between data sets, we do not move the scanner or alter any settings. Thus, in the ideal case we expect the procedure to compute the same rigid motion parameters for each trial.

Figure 15 shows by how much the estimated parameters change between three data sets for each sensor. Some parameters change by significant amounts; the largest observed differences are 16.1 cm and 2.0° for the Erim and 7.4 cm and 1.6° for the Perceptron.

The variations in the rigid motion parameters may be due to numerical instability of our computations. To investigate this, we compute the condition number—the ratio of the largest eigenvalue to the smallest—of the symmetric matrix described in Section 5. In the experiments, we observe condition numbers between 1 and 120. These values do not approach the reciprocal of the computing machine’s floating point double precision. This demonstrates that the matrix is not ill-conditioned. Further, this suggests that the commanded target positions do not lie in a configuration that could cause the solution to degenerate.

The variations in the rigid motion parameters may be due to poor leg position sensing, or poor image analysis, or both. If poor leg position sensing reported target locations imprecisely, then the rigid motion parameters would vary, even with perfect image analysis. Similarly, the parameters would vary if poor image analysis reported target locations imprecisely, even with infinitely repeatable leg position sensing.

For the Erim trials, both factors are conflated, making it difficult to identify their relative magnitudes. We suspect that the dominant cause of poor calibration repeatability is leg position sensing that does not account adequately for the significant structural compliance of the leg members and rails. However, we do not suggest that image analysis errors are

negligible.

For the Perceptron trials, sensing the position of the stiff Ambler legs is highly repeatable, so the dominant cause of poor calibration precision is image acquisition and analysis. Examining the target positions measured in the range images, we find that the position of the target changes less than one pixel between trials (low geometric variation), but that the sensed ranges in the range image can vary significantly (high radiometric variation), even after applying temporal median and spatial averaging filters.

It is unlikely that these range variations are due to mixed pixels, because we exclude all measurements near the edge of the target. It is also unlikely that they are due to changes in material across the target, because the target is homogeneous.

We observe that the range variations correlate strongly with ambient temperature; the higher the temperature, the greater the variations. We have not been able to eliminate the variations, but can achieve acceptable precision by conducting calibration during cool conditions, where inferior to 75° is our heuristic measure of cool.

Precision of Points Transformed into \mathcal{B}

To assess the precision of the points transformed into \mathcal{B} , we evaluate the standard deviation s_D of the distribution of the d_i . This statistic quantifies the scatter of the points mapped into \mathcal{B} , illustrated graphically for one particular data set as the variation of line segment lengths in Figure 13.

Figure 14 shows s_D for ten Erim trials. We observe that the values of s_D range from 1.5 to 4.3 cm, and that they vary significantly between data sets of different sizes, and between data sets of the same size. This range of values is somewhat better than for other trials, where it is not uncommon to observe standard deviations of 10 cm. We conclude that the precision of the calibration procedure for the Erim is no better than 2–5 cm.

Figure 14 shows s_D for ten Perceptron trials. The results show that the precision ranges from 1.0 to 4.3, which is commensurate to that of the Erim. This is several times worse than the resolution of the Perceptron. As in the case of the Erim, the reported range of values is somewhat better than for other trials.

6.4 Execution Time

The time required to execute the calibration procedure with 10–20 points totals approximately 15 minutes for either scanner. The procedure spends roughly 70 percent of this time moving the legs, and roughly 20 percent acquiring and filtering images. Computing the rigid motion parameters and searching for the gain and offset account for most of the remaining 10 percent.

7 Discussion

In this report we have presented an implemented technique to calibrate scanning laser rangefinders to a vehicle-centered reference frame. We reported results for two different sensors and two different vehicles. For the Erim and the prototype leg, the procedure achieves an accuracy of 6–12 cm with a precision no lower than 2–5 cm. For the Perceptron and the Ambler, the accuracy is 2–7 cm with a precision no lower than 2–5 cm. These results have proven to be satisfactory for constructing terrain maps and using them to select footholds during our rough terrain walking experiments.

We have also successfully calibrated the Perceptron scanner to a fixed reference frame. The techniques and results are very similar to those reported for the vehicle-centered reference frames, so we have not treated them separately. We note that this success provides further evidence for the generality of the approach.

The closed-form solution for the rigid motion parameters more than lived up to our expectations. It performed its function with high reliability in reasonable time.

The image analysis has performed well under a wide variety of conditions. Reasons for this success include removing possibly mixed pixels from consideration, using region-based rather than point-based features and statistics, and applying temporal median filters. We found the Grassfire transform to be remarkably effective in removing non-target points and regions.

Still, we can improve the image analysis in a number of ways. Perhaps the most significant would be to modify the sensor so that it is not so sensitive to temperature. It would also be valuable to develop more robust methods, perhaps involving sensor fusion, to identify and eliminate mixed pixels.

One promising direction for future research is toward more comprehensive sensor calibration. This would involve identification of intrinsic sensor parameters such as the relationship between range grey level and absolute distance, the mirror starting angles, and the angular increments, in addition to the six rigid motion parameters.

A Grassfire Transform

The Grassfire transform is a distance transformation related to the Medial Axis Transform [4]. It takes an input image (typically a binary image) and produces an output image whose pixel values indicate the distance to the perimeter of some target region in the input image. In this case, the target region is not literally the region of the image containing the projection of the calibration target, but any of the regions that satisfy Equation 5.

The forward Grassfire transform, G , assigns to each pixel inside a region the distance to the region perimeter. More precisely, it fills the area inside the region with the distance to the nearest pixel outside the region, i.e., ‘1’s in the perimeter pixels, ‘2’s in the interior

pixels adjacent to the perimeter, '3's in the interior pixels adjacent to these, and so forth. It fills the area outside the region with '0's.

It is this behavior that gives the transform its name. Imagine that the input image is a field, where the non-zero pixels correspond to dry, grassy patches, and where the null pixels correspond to barren patches. The Grassfire transform "lights" the perimeter of all grassy regions "on fire," and it assigns the time required for the fire to reach a pixel as the pixel value. Thus, the value of each pixel in a region is its distance to the perimeter of the region.

The reverse Grassfire transform, loosely denoted G^{-1} , fills the area *outside* the region with the distance to the nearest pixel inside the region, and fills the area inside the target region with '0's.

Acknowledgements

This research was sponsored by NASA under Grant NAGW-1175. The views and conclusions contained in this document are those of the author and should not be interpreted as representing the official policies, either expressed or implied, of the funding agencies.

We would like to acknowledge contributions by the entire CMU Planetary Rover research group. We thank R. Hoffman and P. Balakumar for acquiring and analyzing data, K. Arakawa for finding targets in Perceptron images, M. Hebert for exact solutions and helpful discussions, M. Blackwell for scanner interfaces, N. Harding for designing the Perceptron mount, G. Roston for assistance with survey instruments, and C. Tomasi and A. Garvin for commenting on a draft of this paper.

References

- [1] K. Arun, T. Huang, and S. Blostein. Least-Squares Fitting of Two 3-D Point Sets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-9(5):698-700, September 1987.
- [2] J. Bares, M. Hebert, T. Kanade, E. Krotkov, T. Mitchell, R. Simmons, and W. Whittaker. Ambler: An Autonomous Rover for Planetary Exploration. *IEEE Computer*, pages 18-26, June 1989.
- [3] P. Besl. Range Imaging Sensors. Technical Report GMR-6090, General Motors Research Lab, Warren, Michigan, March 1988.
- [4] H. Blum. Biological Shape and Visual Science (Part I). *Theoretical Biology*, 38:205-287, 1973.

- [5] O. Faugeras and M. Hebert. The Representation, Recognition and Locating of 3D Shapes from Range Data. *International Journal of Robotics Research*, 5(3):27–52, Fall 1986.
- [6] M. Hebert, T. Kanade, and I. Kweon. 3-D Vision Techniques for Autonomous Vehicles. Technical Report CMU-RI-TR-88-12, The Robotics Institute, Carnegie Mellon University, 1988.
- [7] B. Horn. *Robot Vision*. MIT Press, Cambridge, Massachusetts, 1986.
- [8] B. Horn. Closed-form Solution of Absolute Orientation using Unit Quaternions. *Journal of the Optical Society of America A*, 4:629–642, 1987.
- [9] B. Horn, H. Hilden, and S. Negahdaripour. Closed-form Solution of Absolute Orientation using Orthonormal Matrices. *Journal of the Optical Society of America A*, 5(7):1127–1135, 1988.
- [10] R. Paul. *Robot Manipulators: Mathematics, Programming and Control*. MIT Press, Cambridge, Massachusetts, 1981.
- [11] G. Schut. On Exact Linear Equations for the Computation of the Rotational Elements of Absolute Orientation. *Photogrammetria*, 17(1):34–37, 1960.
- [12] S. Song and K. Waldron. *Machines that Walk: The Adaptive Suspension Vehicle*. MIT Press, Cambridge, Massachusetts, 1989.
- [13] E. Thompson. A Method for the Construction of Orthogonal Matrices. *Photogrammetric Record*, 3:55–59, 1958.
- [14] P. Veatch and L. Davis. Efficient Algorithms for Obstacle Detection Using Range Data. *Computer Vision, Graphics, and Image Processing*, 50:50–74, 1990.
- [15] D. Zuk, F. Pont, R. Franklin, and V. Larrowe. A System for Autonomous Land Navigation. Technical Report IR-85-540, Environmental Research Institute of Michigan, Ann Arbor, Michigan, 1985.