CMU-CS-86-140

Error Modelling in Stereo Navigation

Larry Matthies and Steven A. Shafer

Computer Science Department
Carnegie-Mellon University
Pittsburgh, PA 15213

Abstract

In stereo navigation, a mobile robot estimates its position by tracking landmarks with on-board cameras. Previous systems for stereo navigation have suffered from poor accuracy, in part because they relied on scalar models of measurement error in triangulation. This paper shows that using 3-D gaussian distributions to model triangulation error leads to much better performance. The paper describes how to compute the error model from image correspondences, estimate robot motion between frames, and update the global positions of the robot and the landmarks over time. Simulations show that compared to scalar error models the 3-D gaussian reduces the variance in robot position estimates and better distinguishes rotational from translational motion. A short indoor run with real images supported these conclusions and computed the final robot position to within 2% of distance and one degree of orientation. These results illustrate the importance of error modelling in stereo vision for this and other applications.

This research was sponsored by the Office of Naval Research under contract N00014-81-K-0503 and by the Defense Advanced Research Projects Agency under contracts DACA 76-85-C-0003 and F33615-84-K-1520. 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 ONR, DARPA, or of the U.S. Government.

1. Introduction

Consider a robot given the task of going from A to B. At a coarse level its route is planned from a pre-stored map, while at a fine level the route is determined by sensor information gathered along the way. Incremental motion estimates are integrated to keep track of the robot's position in the map, which in turn is used to predict upcoming landmarks, hazards, or arrival at the destination.

To realize this scenario, a robot needs sensors that can measure its position and detect the presence of 3-D objects nearby. Stereo vision can provide both kinds of information. Stereo matching at one point in time provides a local 3-D model for route planning and obstacle avoidance. Selected points in this model become landmarks that are tracked by the stereo system to monitor the robot's progress. Using stereo in this way, to detect nearby objects and to estimate the motion of the robot, is what we refer to as stereo navigation.

We are interested in stereo in this scenario for a number of reasons. First, other motion sensors can be in error, such as shaft encoders when wheels slip or lose contact with the ground. Second, other sensors, such as sonar and radar, can be inappropriate for reasons of concealment, possible confusion with the broadcasts of other robots nearby, or because color and reflectivity information are important. Lastly, we are interested in stereo *per se* and believe that methods developed for this domain can be transferred to other applications.

Methods for extracting shape and motion information from image sequences can be classified as correspondence-based or flow-based. Correspondence methods [7, 11, 18, 24] track distinct features such as corners and lines through the image sequence and compute 3-D structure by triangulation. Flow-based methods [1, 25] treat the image sequence as function $I(x,y,t)$ of row, column, and time, restrict the motion between frames to be small, and compute shape and motion in terms of differential changes in I. This paper deals with error modelling issues in the correspondence paradigm.

One of the first systems for correspondence-based stereo navigation was that built by Moravec [18]. This system moved a robot in a stop-go-stop fashion, digitizing and analyzing images at every stop. Features were matched in stereo images to build a world model consisting of 3-D points. After moving and acquiring more images, the points in the world model were matched in the new images to find their coordinates relative to the new robot location. A least squares procedure was applied to the differences between the new and old point locations to infer the actual motion of the robot. The contribution of each landmark point to this motion estimate was multiplied by a scalar weight that varied inversely with the distance to the point.

In earlier work with Moravec [17], we found the motion solving part of this system to be somewhat inaccurate and unstable. This has been a common experience with visual motion solving algorithms in general. In the case of correspondence-based algorithms, this can partly be attributed to inadequate modelling of measurement error in triangulation. In triangulation, 3-D coordinates are computed by intersecting rays projected through corresponding points in two images. Errors in locating the image points induce errors in the 3-D coordinates, which in turn cause errors in motion estimates based on the 3-D information. Modelling the measurement errors can reduce their effect on motion estimates. However, we will demonstrate that using scalar weights to model uncertainty in 3-D coordinates leads to poor performance.

More sophisticated methods have been used in a number of places. In photogrammetry [20], 2-D and 3-D normal distributions are used to model error in image coordinates and 3-D point locations, respectively. Gennery [11] has used 2-D normal distributions of image coordinates in camera calibration for computer vision. Hallam [15] used normal error models in conjunction with Kalman filters to track points and estimate robot motion from sonar data. Broida and Chellappa [5] used similar methods to track a known object in monocular image sequences, and recently Faugeras [9] has discussed the application of these methods to stereo in work similar to ours.

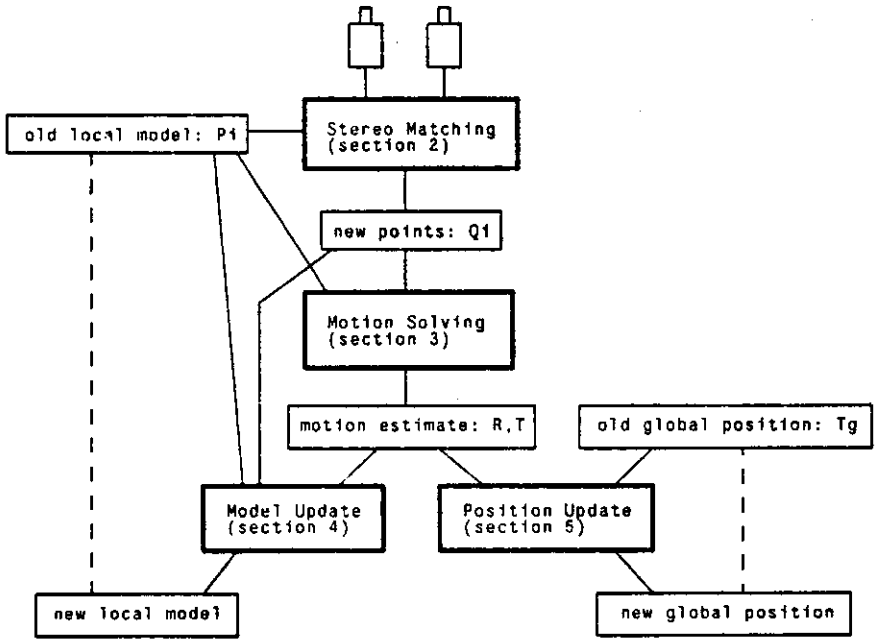

Figure 1-1: System block diagram

This paper shows how these methods can be applied to stereo navigation and demonstrates by results with real images that they lead to markedly better performance. The system we will describe

has evolved from Moravec's [18] and is shown in figure 1-1. The main data structures are a set of 3-D points P_i, called the local model and described in robot-centered coordinates, and the robot's current estimate of its position in some fixed, global reference frame. The points in the local model are obtained by stereo matching and are used as landmarks. When a new stereo pair is digitized, points from the local model are matched in the images to determine their current locations Q_i relative to the robot. A motion solving algorithm estimates the rotation and translation (R and T) relating the new and old coordinates. The model updating system transforms the old local model into the current coordinate frame and combines it with the new points to create a new local model. Finally, the motion estimate is used to update the robot's global position. The cycle then repeats with the acquisition of a new pair of images.

Section 2 shows how to model triangulation error in the stereo matcher with 3-D normal distributions. In section 3 this is incorporated in an algorithm for finding the rotation and translation between successive stereo pairs. The covariance matrix of this transformation is used in section 4 to update the local model with Kalman filters and in section 5 to estimate the robot's global position uncertainty. Simulations described in section 6 show that compared to scalar error models this system reduces the variance of position estimates and better distinguishes rotational motion from translation. An experiment with real images, using 54 stereo pairs covering 5.4 meters and fully automatic feature tracking, supported these conclusions and computed the final robot position to within 2% of distance and one degree of orientation. Conclusions are summarized in section 7.

2. Modelling stereo triangulation error

The geometry of stereo triangulation is shown in figure 2-1. For the moment we consider just the case of 2-D points projecting onto 1-D images. Two cameras are placed at offsets of $\pm b$ from a coordinate system centered between the cameras. Suppose point P projects onto the left image at x_l and the right image at x_r. Because of errors in measurement, the stereo system will determine x_l and x_r with some error. The error can come from many sources, including quantization of the image, photometric and geometric distortion in the camera, and the effects of perspective distortion on the matching algorithm. This error in turn causes the true location of P to be inferred with some error. Figure 2-1 illustrates this for errors caused by image quantization; because of resolution limits, the estimated location of P can lie anywhere in the shaded region surrounding the true location [22]. Additional random effects will cause this region to have less sharp boundaries, but the general shape will be similar. We want to take this uncertainty into account in any reasoning based on measurements of P.

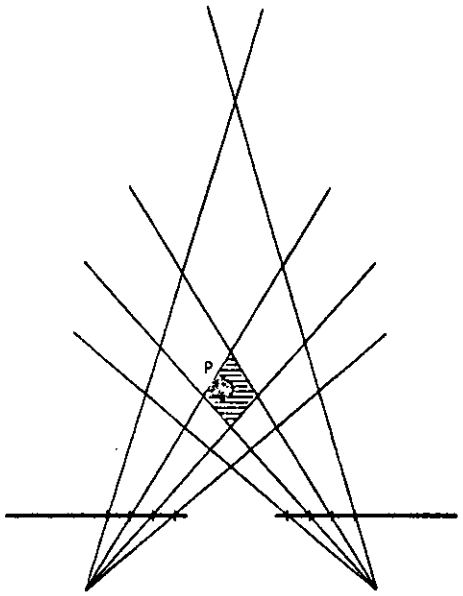

Figure 2-1: Stereo geometry showing triangulation uncertainty

Three approaches to modelling such uncertainty are discrete tolerance limits, scalar weights, and multi-dimensional probability distributions. Tolerance regions are often used in object recognition and motion planning [4, 6, 14]; however, they are inappropriate for our application because of the combinatorial nature of the algorithms they require, the stochastic nature of matching errors, and the need to filter time sequences of data.

The idea behind scalar weights is that uncertainty grows with distance, so it can be modelled by weighting points inversely with distance [18]. However, as figure 2-1 shows, the uncertainty induced by triangulation is not a simple scalar function of distance to the point; it is also skewed and oriented. Nearby points have a fairly compact uncertainty, whereas distant points have a more elongated uncertainty that is roughly aligned with the line of sight to the point. Scalar error measures do not capture these distinctions in shape.

Normal distributions are commonly used in photogrammetry [20] and navigation [10, 26] to model uncertainty in two and three dimensional data. In computer vision, they have been used to model error in coordinates of image correspondences [11], monocular object tracking [5, 12], navigation and tracking with sonar [15], and recently in stereo work similar to ours [9]. To model triangulation error, we begin by treating image coordinates as corrupted by 2-D, normally distributed (ie. gaussian) noise and derive from this a distribution of the error in the inferred 3-D coordinates. Because triangulation is a non-linear operation, the true 3-D distribution will be non-gaussian. We approximate

this as gaussian because it is simpler and gives an adequate approximation when the distance to points is not extreme. We will discuss shortly the cases where this breaks down.

We will now show the details of the triangulation and error model calculation for the general case of 3-D points projecting onto 2-D images. Let the image coordinates be given by $l = [x_l, y_l]$ and $r = [x_r, y_r]$ in the left and right image, respectively. Consider these as normally distributed random vectors with means μ_l and μ_r and covariance matrices V_l and V_r. From l and r we need to estimate the coordinates $[X, Y, Z]^T$ of the 3-D point P. We take the simple approach of using the ideal, noise-free triangulation equations $P = [X, Y, Z]^T = f(l, r)$ or

$$\begin{aligned} X &= b(x_l + x_r)/(x_l - x_r) \\ Y &= b(y_l + y_r)/(x_l - x_r) \\ Z &= 2b/(x_l - x_r) \end{aligned} \tag{1}$$

(assuming a unit focal length) and inferring the distributions of X, Y, and Z as functions of random vectors l and r. If equation (1) was linear, P would be normal [8] with mean $\mu_p = f(\mu_l, \mu_r)$ and covariance

$$V_p = J \begin{bmatrix} V_l & 0 \\ 0 & V_r \end{bmatrix} J^T \tag{2}$$

where J is the matrix of first partial derivatives of f or the Jacobian. Since f is nonlinear these expressions do not hold exactly, but we use them as satisfactory approximations.

The true values of the means and covariances of the image coordinates needed to plug into (1) and (2) are unknown. We approximate the means with the coordinates returned by the stereo matcher and the covariances with identity matrices. This is equivalent to treating the image coordinates as uncorrelated with variances of one pixel. Better covariance approximations can be obtained by several methods [2, 11].

What does this error model mean geometrically? Constant probability contours of the distribution of P describe ellipsoids about the nominal mean that approximate the true error distribution. This is illustrated in figure 2-2, where the ellipse represents the contour of the error model and the diamond represents quantization error of figure 2-1. For nearby points the contours will be close to spherical; the farther the points the more eccentric they become. A covariance matrix with structure $V = wI$, equal to a scalar times the identity matrix, describes only spherical contours. This is the difference between attaching scalar weights to 3-D coordinate vectors and using the full 3-D distribution; that is,

scalar weights are equivalent to spherical covariances whereas the full distribution permits ellipsoidal covariances. In the balance of the paper we will often refer to scalar weights as a spherical error model and the full distribution as an ellipsoidal error model.

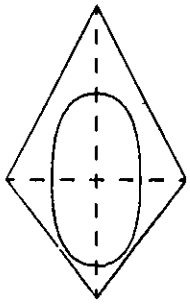

Figure 2-2: Quantization error with normal approximation

Where the gaussian approximation breaks down is in failing to represent the longer tails of the true error distribution. The true distribution is skewed not unlike the diamond in figure 2-2, whereas normal distributions are symmetric. The skew is not significant when points are close, but becomes more pronounced the more distant the points. A possible consequence is biased estimation of point locations, which may lead to biased motion estimates. We will return to these issues section 6.

3. Solving for robot motion

The previous section showed how to model measurement error in stereo triangulation. In this section we show how to incorporate the error model into an algorithm for estimating the motion between successive stereo pairs. We will begin by showing how motion is computed with scalar weights, then derive an algorithm based on the 3-D gaussian error model, and finally give this algorithm a geometric interpretation.

Refering back to figure 1-1, at this stage in the cycle the robot has two sets of 3-D points that have been obtained by stereo matching: a local model of points P_i defined relative to its previous position and the coordinates Q_i of these points relative to its current position. The correspondences between P_i and Q_i are known, but the motion between them is not. Parameterizing the rotation in terms of Euler angles, we have a set of equations

$$Q_i = R P_i + T$$

in which P_i and Q_i are known point vectors, R is the matrix of the unknown rotation, and T is the unknown translation.

Using scalar weights, one finds R and T by expressing the errors of fit by

$$\varepsilon_i = Q_i - R P_i - T$$

and minimizing the weighted sum of squares

$$\sum_{i=1}^{n} w_i \varepsilon_i^T \varepsilon_i \tag{3}$$

where w_i are the weights. Although the rotation makes this optimization problem nonlinear, it has a closed form solution [19]. A solution for case where the rotation is parameterized by quaternions is given in [16].

As will be shown in section 6, the scalar model of uncertainty embodied in equation (3) leads to poor performance. Using the 3-D gaussian error model the solution takes a similar, but more complicated form. For simplicity we begin with the case of translational motion. The simplified motion equation is

$$Q_i = P_i + T$$

which we may rewrite as

$$Q_i - P_i = M_i = T$$

to emphasize the role of $M_i = Q_i - P_i$ as measurements of T. From section 2, P_i and Q_i are modelled as normally distributed random vectors with covariances U_i and V_i, respectively. Therefore, M_i will also be normally distributed with covariance $U_i + V_i$. Now if we consider M_i to be a sequence of noisy measurements of T, each corrupted by noise with zero mean and covariance $U_i + V_i$, application of the maximum likelihood method leads to minimizing the following expression over possible values of T [8]:

$$\sum_{i=1}^{n} \varepsilon_i^T W_i \varepsilon_i \tag{4}$$

where $\varepsilon_i = M_i - T$ and $W_i = (U_i + V_i)^{-1}$. The solution to this is

$$T = (\Sigma_{i=1}^{n} W_i)^{-1} \Sigma_{i=1}^{n} W_i M_i$$

and the covariance matrix of the estimation errors is

$$V_T = (\Sigma_{i=1}^{n} W_i)^{-1}$$

The covariance matrix can be analyzed to assess the quality of the motion estimate. It is also used later in modelling the uncertainty of the robot's global position estimate.

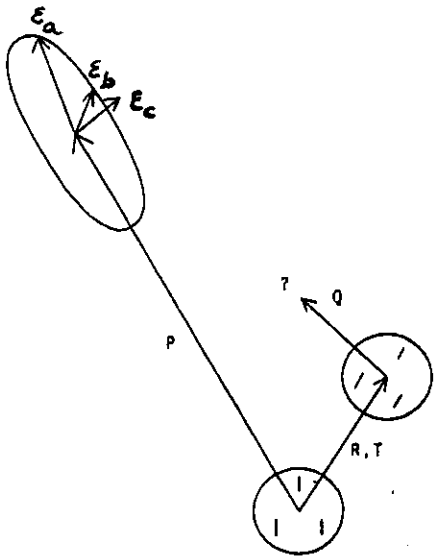

Figure 3-1: Interpretation of equation (4): W_i scales the residual vectors, lengthening them parallel to the line of sight and shortening them perpendicularly to it.

An intuitive interpretation of equation (4) is shown in figure 3-1. The weight matrices W_i function as norms that measure distance differently for each point. Error vectors making equal contributions to the total error of fit lie on ellipsoidal contours. For example, in figure 3-1, residuals ε_a and ε_b contribute equally to the total error but ε_c contributes more because $\varepsilon_a{}^T W \varepsilon_a = \varepsilon_b{}^T W \varepsilon_b < \varepsilon_c{}^T W \varepsilon_c$. This effectively gives more weight to errors perpendicular to the line of sight than parallel to it, which, given the nature of stereo, is what we would like to do. The "spherical" error model obtained by using the scalar weights of equation (3) has the obvious mnemonic meaning that residual vectors making equal contributions to the total error lie on spherical contours. This distinction is what gives the ellipsoid model its power.

Generalizing this method to handle rotation is complicated by the fact that the equations become nonlinear. The function to be optimized takes the form

$$\sum_{i=1}^{n} \varepsilon_i{}^T W_i \varepsilon_i \qquad (5)$$

$$with\ \varepsilon_i = Q_i - R P_i - T$$

$$and\ W_i = (R U_i R^T + V_i)^{-1}$$

We have not been able to find direct solutions to this problem or even to approximations in which W_i is not a function of R. Our approach has been to use the direct solution for scalar weights to get an initial approximation, then to iterate on linearizations of equation (5). Linearization methods for solving least squares problems are described in [13].

To recap, this section incorporated the error model of section 2 in an algorithm for finding the rotation and translation between two 3-D points sets. The algorithm replaces the scalar weights of equation (3) with weight matrices based on the covariances of corresponding points. When the motion is purely translational, the problem is linear and has a direct solution, but when the motion involves rotation we resort to an iterative solution. The error covariance of the motion solution will be used in the following two sections in updating the robot's local model and global position estimate.

4. Updating the local model

So far we have described how to model error in triangulation and how to solve for the motion between two successive stereo pairs. This section deals with how to process a long sequence of stereo pairs. At issue is how to average information from successive images to achieve more accurate landmark localization and consequently more accurate estimates of robot position.

An appropriate tool for this is the Kalman filter [10]. In filtering terminology the quantity to be estimated is called the "state", and when a measurement is taken the filter updates the current estimate of the state. Kalman filters incorporate known statistical properties of the measurements into the update process and produce error covariances for the state estimate. They are widely used in terrestrial and aerospace navigation and guidance applications [10, 26]. In computer vision they have been used in object recognition [3], tracking of known objects with monocular image sequences [5, 12], and for robot navigation and object tracking with sonar data [15].

In our application, the state consists of the locations of the landmark points in the local model. A question arises as to whether the landmarks should be represented in a global, stationary frame of reference or in a local, moving, robot-centered frame. In either case, the update involves transforming coordinates from one frame to the other and applying the filter. If a fixed number of landmarks are being tracked, there is no difference in cost between the two. There will be a difference in the uncertainty of the resulting model; this difference depends on the relative uncertainties of the old model, the new measurements, and the intervening motion. We have not completed an analysis of this situation, but are currently keeping the landmark model in robot-centered coordinates.

The update involves transforming the old local model to the current coordinate frame, inflating its uncertainty to account for the uncertainty of the transformation, and filtering the old model with the new measurements to create the updated model. Let P_{t-1} be the coordinate vector of a single point in the old local model at time $(t-1)$ and let V_{t-1} be its covariance. For purely translational motion, P_{t-1} is transformed to the current frame by

$$\mathcal{P}_{t-1} = P_{t-1} + T \tag{6}$$

where T is the translation from time $(t-1)$ to time t. The translation has an error covariance matrix V_T, so the transformed point has covariance

$$\mathcal{V}_{t-1} = V_{t-1} + V_T \tag{7}$$

Equation (6) introduces some correlation between points that is not accounted for in (7), but we assume this is small enough to ignore. To extend this to rotation, we rewrite equation (6) as

$$\mathcal{P}_{t-1} = R P_{t-1} + T \tag{8}$$

This is nonlinear, so to compute $\mathcal{V}_{t-1}$ we proceed by analogy to equation (2); that is, we pre-multiply the covariance of R, T, and P_{t-1} by the Jacobian of the transformation and post-multiply by the Jacobian transposed. Since we treat P_{t-1} as uncorrelated with R and T, this leads to

$$\mathcal{V}_{t-1} = J_m V_m J_m^T + R V_{t-1} R^T$$

where J_m contains the derivatives of (8) with respect to the motion parameters and V_m is the covariance of the motion parameters.

Now let Q_t be the measurement of the same point at time t and let U_t be the covariance of this measurement. Some manipulation of the basic Kalman filter equations leads to the following estimates of the updated point location and covariance:

$$V_t = (\mathcal{V}_{t-1}^{-1} + U_t^{-1})^{-1} \tag{9}$$

$$P_t = \mathcal{P}_{t-1} + V_t U_t^{-1}(Q_t - \mathcal{P}_{t-1}) \tag{10}$$

The intuition behind equation (10) is as follows. The second term takes the difference $(Q_t - \mathcal{P}_{t-1})$ of the new measurement from the old estimate, weights the difference by $V_t U_t^{-1}$, and applies the result as an update to the old estimate $\mathcal{P}_{t-1}$. Matrix U_t^{-1} will be "larger" the more precise the new measurement, giving it more weight in the update, and smaller the less precise the measurement,

giving it less weight. Conversely, V_t will be small if the old estimate is precise and large otherwise. Hence if the old estimate is already good, the new measurement receives little weight; if it is poor, the new measurement receives more weight.

The procedure we have described assumes that the error in the motion estimate is uncorrelated with the error in the landmark points. When the motion estimate is obtained by using the methods of the previous section this will not be true, although if other sensors are also contributing to the motion estimate it will be approximately true. This is an issue we are investigating.

5. Updating the global robot position

By using the modules discussed in the previous sections, the robot computes estimates of its motion between successive stereo pairs. Combining these to estimate its global position is a simple matter of concatenating the transformation matrices. It may also be desirable to estimate the uncertainty of the global position, which can be done by propagating the covariance matrices of the incremental motions into a covariance of the global position. For translation this is also very simple. If the the global position at time $(t-1)$ is $T_{g_{t-1}}$ and the next incremental translation is T_t, then the next global position is

$$T_{g_t} = T_{g_{t-1}} + T_t \tag{11}$$

Since this is linear, if the incremental translation estimates have uncorrelated, zero mean gaussian errors, then T_{g_t} will also have zero mean, gaussian error with covariance given by

$$V_{g_t} = V_{g_{t-1}} + U_t$$

where $V_{g_{t-1}}$ and U_t the covariances of $T_{g_{t-1}}$ and T_t, respectively. The case of motion in the plane, where there are two parameters for translation and one for rotation, has been dealt with by Smith and Cheeseman [21]. In summary, one obtains an equation analogous to (11) in which the three parameters of the global position are expressed as functions of the previous position and the incremental motion. These are nonlinear and error propagation is done by linearization. For general motion in three dimensions, this is not straightforward with the Euler angle representation of rotation we have used here. In this case other parameterizations of rotation, such as quaternions, may be preferable [9, 26]. We are exploring this further.

6. Performance

Our evaluation to date has concentrated on comparing the use of the spherical and ellipsoidal error models in the motion solving methods of section 3. Results of tests with simulated and real data are described below.

6.1. Simulations

Three sets of simulation data will be presented. The first set is a base case that compares the standard deviations of position estimates obtained with each error model for a single step of vehicle motion. That is, it considers motion between only two consecutive stereo pairs. It illustrates the difference in the variability of position estimates with each model and reveals different amounts of coupling between translation and rotation with each error model. The second set of data also considers only two consecutive stereo pairs and tests limiting performance by tracking progressively more distant points. The last set examines both long range performance over many images and the effect on performance of different stereo baselines.

The simulations were generated as follows. The "scene" consisted of random points uniformly distributed in a 3-D volume in front of the simulated cameras. Typically this volume extended 5 meters to either side of the cameras, 5 meters above and below the cameras, and from 2 to 10 meters in front of the cameras. The cameras themselves were simulated as having 512x512 pixels and a field of view of 53 degrees. The stereo baseline for most simulations was 0.5 meters. Image coordinates were obtained by projecting the points onto the images, adding gaussian noise to the floating point image coordinates, and rounding to the nearest pixel. These coordinates were input to the triangulation and motion solving algorithms. For the ellipsoidal error model, covariance matrices were computed as described in section 2. In the scalar case, weights were derived by taking the Z variance from the covariance matrix. Scalars obtained by several other methods were tried and found to give very similar results. These include the volume and length of the major axis of the standard error ellipsoid and Moravec's half-pixel shift rule [18].

The first set of simulations determined the standard deviation of the estimated motion between two consecutive stereo pairs when the true motion was one meter. The results are shown in figures 6-1 and 6-2 plotted against the number of points used to compute the motion estimate. In both figures, the top three curves were obtained with spherical modelling and the bottom three with ellipsoidal. Tilt implies rotation of the camera up or down, pan is the rotation about the vertical axis, and roll the rotation about the camera axis. The most significant thing to note is that the standard deviations obtained with the ellipsoidal model are considerably less than those obtained with the spherical

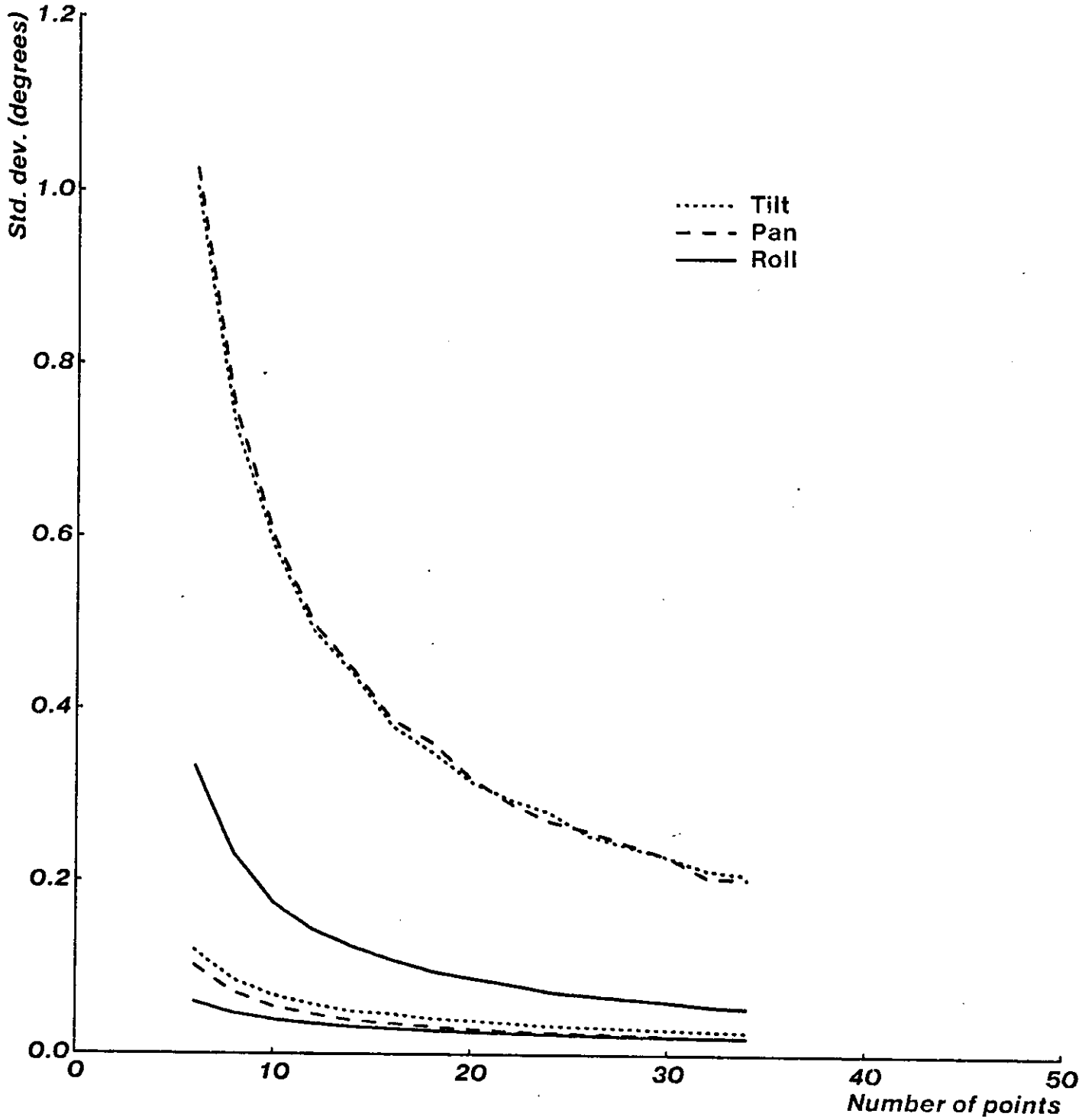

Figure 6-1: Standard deviation vs. number of points for rotations. Top three curves are for the spherical model, bottom three are for the ellipsoidal model. Use of the ellipsoidal model gave significantly lower variance in the estimates.

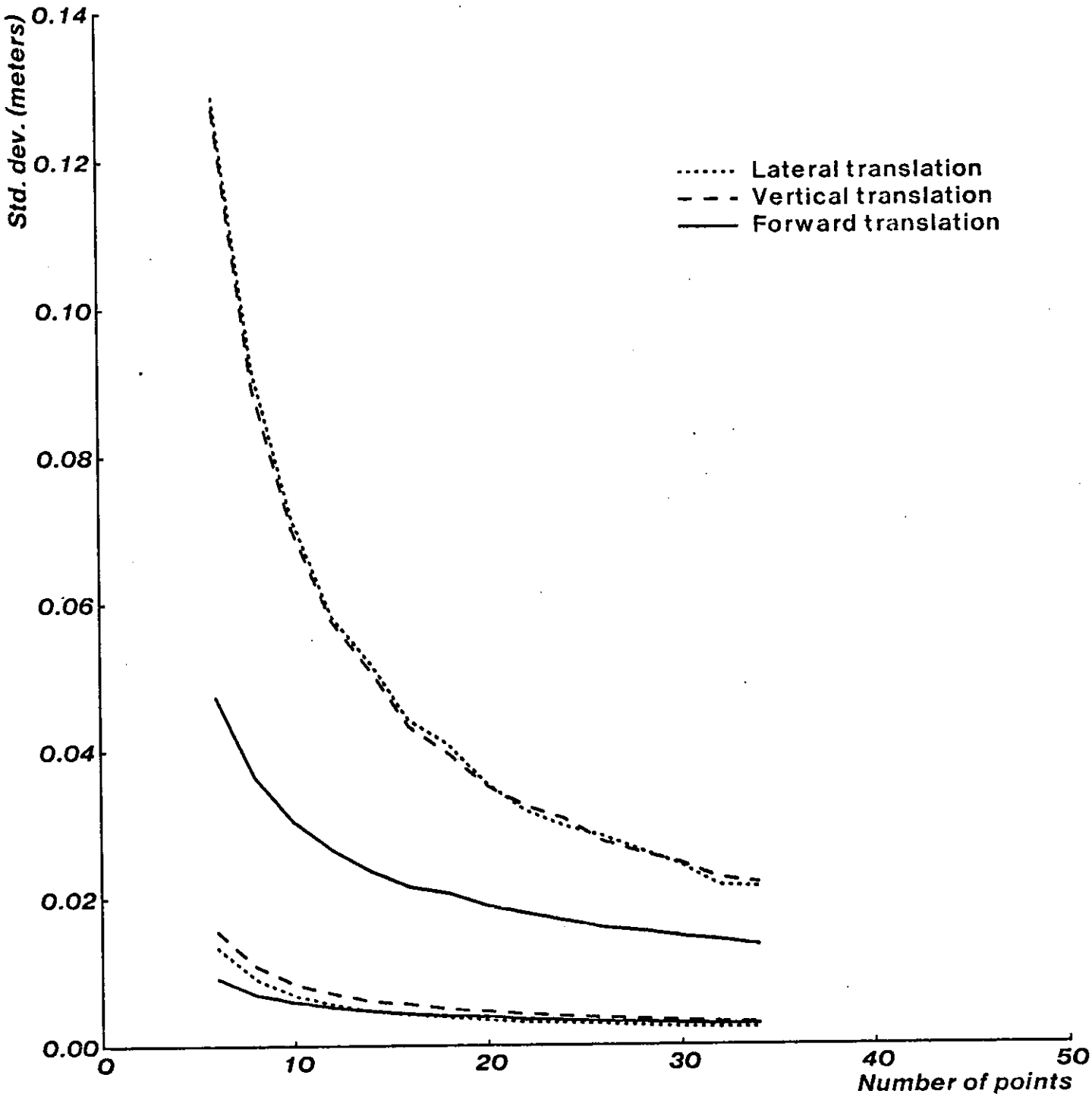

Figure 6-2: Standard deviation vs. number of points for translations. Top three curves are for the spherical model, bottom three are for the ellipsoidal model.

model. The size of this difference will vary; it will be larger when the 3-D points are further from the cameras and smaller when they are closer. This is because the spherical error model is a reasonable approximation to the triangulation error when points are close, but not when they are distant. Another thing to note is that with the spherical model roll and forward translation are estimated better than the other parameters, but with the ellipsoidal model all parameters are estimated equally well. This is because lateral translations and panning rotations have coupled effects on the errors of fit, as do vertical translations and tilting rotations. Using an ellipsoidal error model appears to reduce this coupling. Lastly, note that for a given level of performance fewer points are needed with the ellipsoidal model than the spherical, offseting the greater expense of the iterative motion solution needed in the ellipsoidal case. The exact relationship will depend on the camera configuration.

The second set of simulations also dealt with the estimated motion between just two stereo pairs. It examined the effect of increasing the distance to points in the scene, or equivalently to reducing the maximum disparity in the image. Figures 6-3 and 6-4 illustrate the results. Twenty points were generated in a volume spanning 4 to 50 meters in front of the cameras, giving disparities ranging from 2 to 32 pixels or 0.5% to 6% of image width. The volume was gradually shrunk by moving the near limit from 4 meters back until all points were 50 meters away, so that all disparities were on the order of 2 or 3 pixels. Figure 6-3 shows the mean value of the forward translation estimate as a function of the minimum distance to the points and figure 6-4 the standard deviation. The true forward motion was one meter. Looking at the means, with the ellipsoidal error model there is a consistent underestimation of the true motion that gets worse as the disparity shrinks. With the spherical error model the behavior is erratic. The jagged nature of the curve for the spherical model is due to the contribution of image quantization to the noise in the image coordinates. As a 3-D point moves smoothly away from the cameras, image quantization will lead the triangulation to alternately under- and overestimate the true distance to the point (see [22] for a good illustration). This in turn affects motion estimates based on tracking the point. Apparently the ellipsoidal model smooths out this effect. Figure 6-4 shows that the standard deviation of the motion estimates increases quite rapidly with shrinking disparity in the spherical case, but much less rapidly in the ellipsoidal case. On the whole, the breakdown with distance shown by the spherical error model is consistent with common experience in computer vision; this makes the stability shown with the ellipsoidal model come as quite a surprise.

Whereas the first two sets of simulations looked at motion estimates between only two consecutive stereo pairs, the last set looked at motion over a long sequence of images. There were two purposes for these simulations. The first was simply to confirm the results of the single-step simulations. The second was to test a hypothesis suggested by the previous simulation: that for equivalent

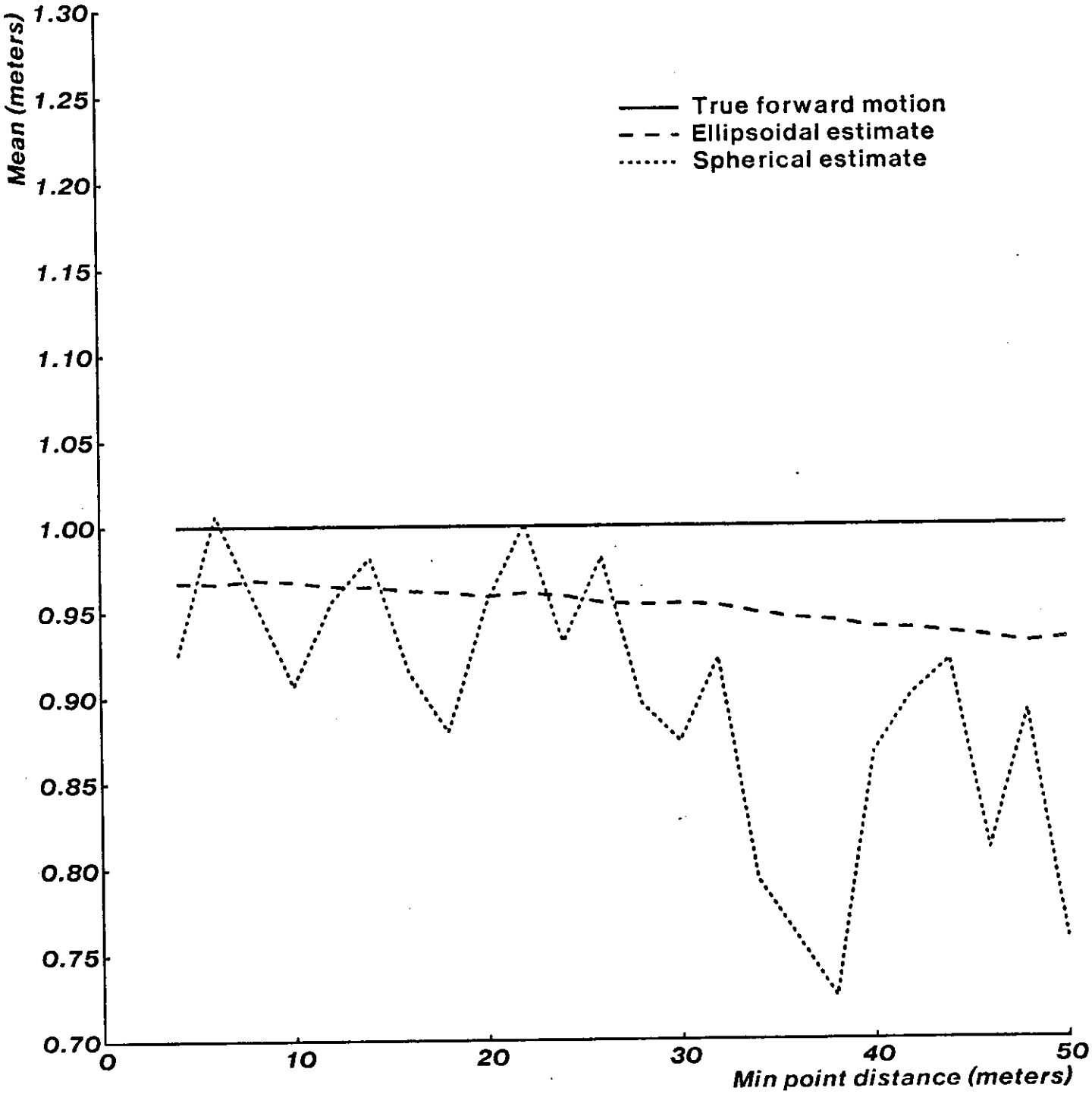

Figure 6-3: Bias in estimating forward distance travelled vs. minimum distance to points. Four meters corresponds to a 32-pixel disparity, 50 meters to a 2-pixel disparity.

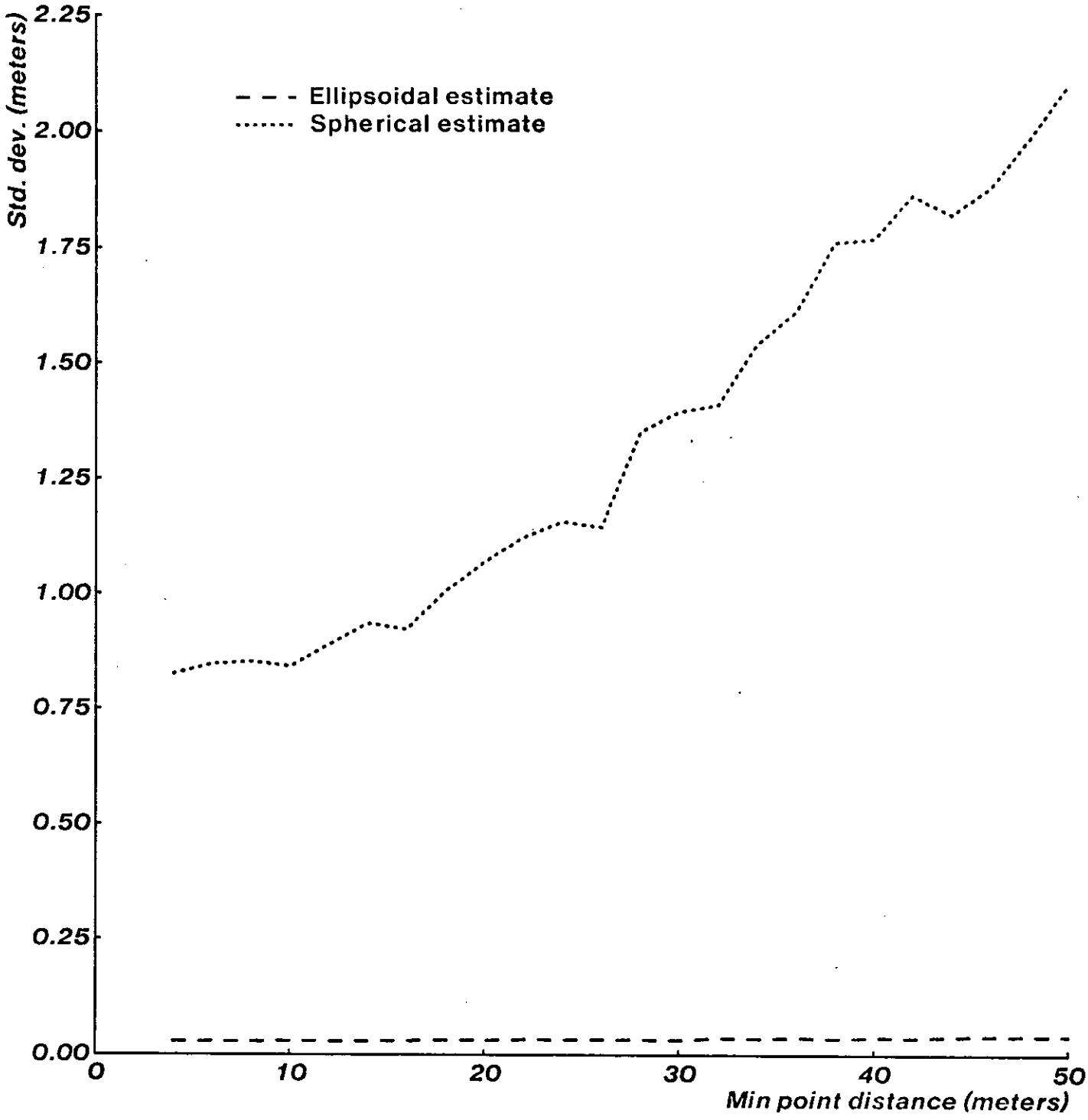

Figure 6-4: Standard deviation of estimated forward distance travelled vs. minimum distance to points.

performance, the ellipsoidal model may permit the use of a shorter stereo baseline than the spherical. This is an important consideration, because length of the baseline directly affects the difficulty of stereo matching. Figure 6-5 shows the standard deviation of the estimated distance as a function of the true distance. Here the simulated travel between images was 0.64 meters, so the figure represents about 90 simulated images. It shows curves for a 0.5 meter baseline with the spherical model and 0.125, 0.25, and 0.5 meter baselines for the ellipsoidal model. Comparing the curves for 0.5 meter baselines, the ellipsoidal model does outperform the spherical. It appears that the curves may eventually run parallel, so that the difference between the methods would be an additive constant rather than multiplicative. Looking at the effects of different baselines, results with the ellipsoidal model are still better than the spherical model with a 0.25 meter baseline, though not with 0.125 meters. Based on standard deviations of position, it does appear possible to use a shorter baseline. However, another factor involved is bias of the motion estimates. As seen in figure 6-3, increasing the ratio of object distance to baseline tends to cause motion estimates with both error models to underestimate the true distance. In general we have found that the narrower the baseline, the more motion is underestimated. The same occurs when we increase the variance of the simulated noise in the image coordinates. This appears to result from a net underestimation of the distance to points in space. Simple compensation schemes appear to work when the only error in image coordinates comes from quantization, but are less adequate as the noise variance grows. This requires further investigation. For the moment we just note that bias can be a problem with short baselines or non-trivial noise levels.

6.2. Real images

In order to verify the simulations on real images, we used both error models to estimate the position of a stereo-equipped robot travelling across the floor of our lab. The scene is pictured in figure 6-6. The robot was driven straight forward in 54 steps of slightly less than 10 centimeters each. The cameras were on a 20 centimeter baseline and had a 36-degree field of view. The FIDO feature-tracking system [23] was used to track points through the image sequence and the resulting set of matched image coordinates were input to the algorithms described earlier to estimte the robot's position at each step. We will briefly describe the operation of FIDO before discussing the results of the experiment.

FIDO uses the Moravec interest operator and coarse-to-fine correlation algorithm to pick and match point features in stereo pairs. The interest operator is applied to one image of a stereo pair to pick points in where intensity varies in all directions; typically these are sharp corners or intersections of lines. The correlator finds these points in the other image of the stereo pair. To find the same points

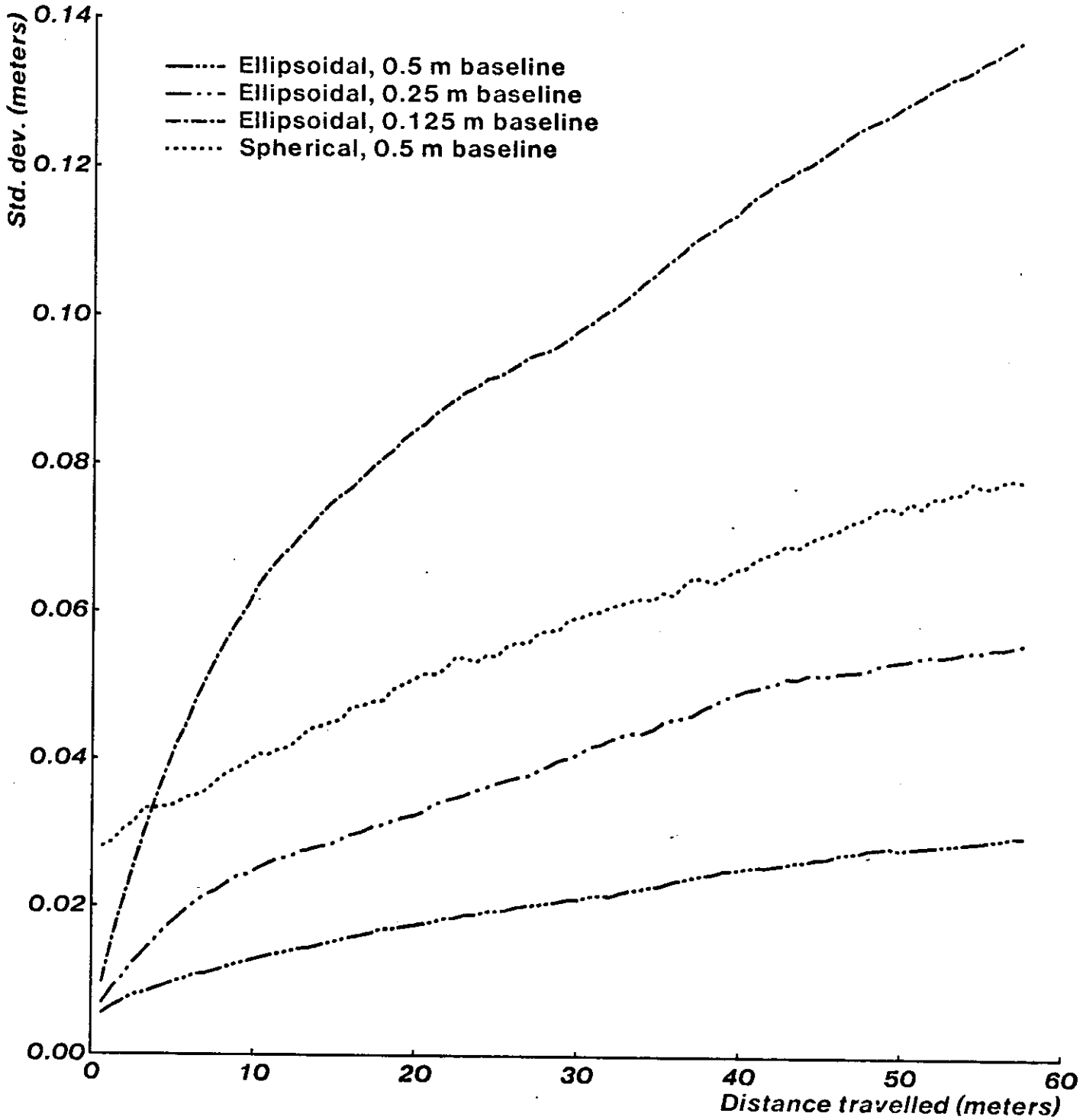

Figure 6-5: Standard deviation of estimated forward distance travelled vs. true distance.

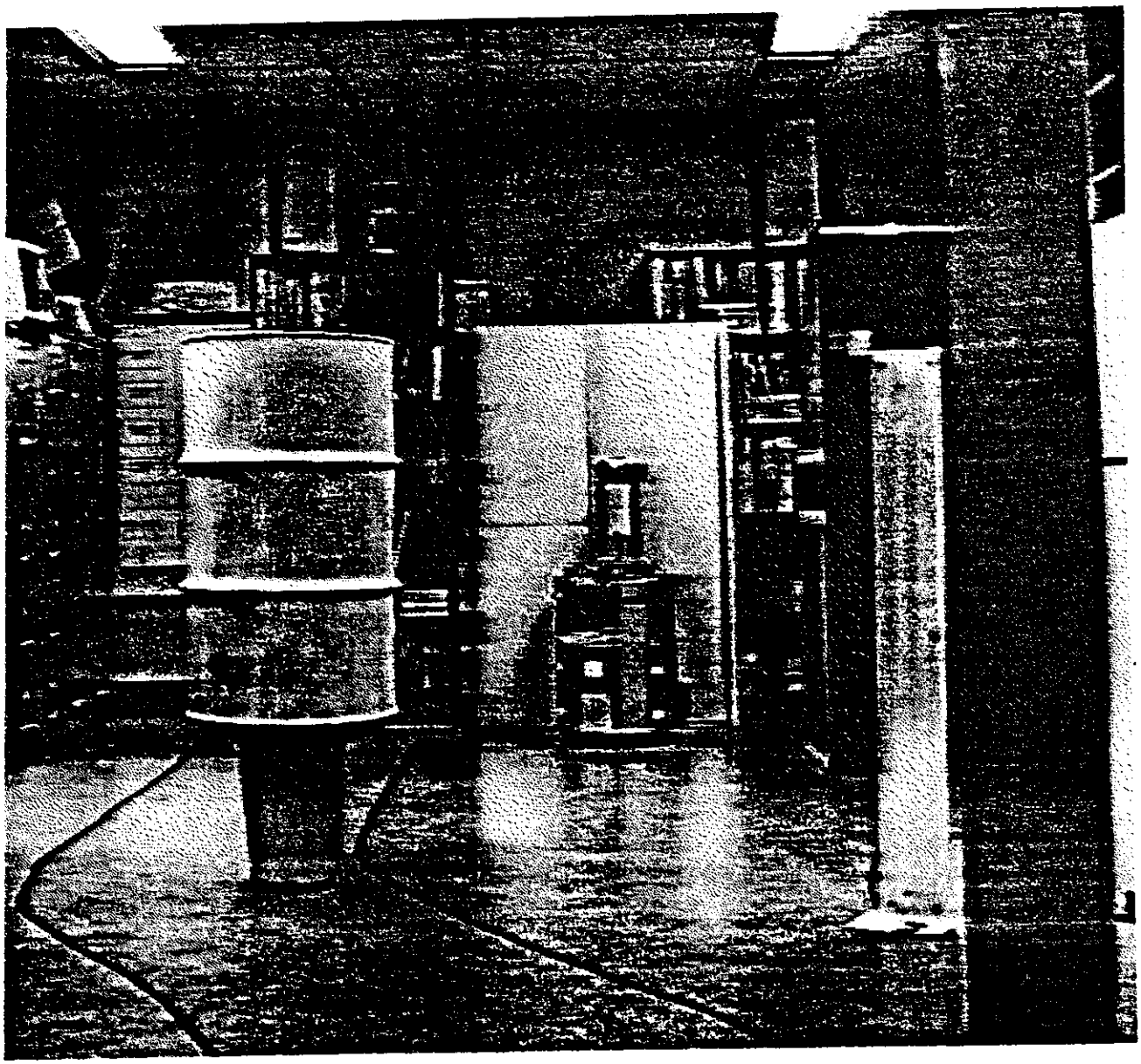

Figure 6-6: One image from lab sequence.

in subsequent stereo pairs, an *a priori* motion estimate is used to predict the location of the point in the new images, a constraint window is defined around the predicted location based on the the uncertainty of the motion estimate, and the correlator is applied to find the position of best match within the constraint window. Incorrect matches are culled with a threshold on the correlation coefficient and with a 3-D error heuristic called the "3-D prune" stage. This heuristic uses the fact that under rigid motion the distance between two 3-D points does not change over time. Points which appear to violate this condition are discarded. The advantage of this test is that it does not require knowledge of the motion between stereo pairs. Points that survive this test become input to the motion solving algorithms. In the experiments to follow, between 30 and 40 points usually remained.

Figure 6-7 compares the true motion to the position estimates obtained with the spherical and ellipsoidal error models. For this figure a "planar" motion solver was used that solved only for the parameters of motion in the plane, that is two degrees of translation and one of rotation.

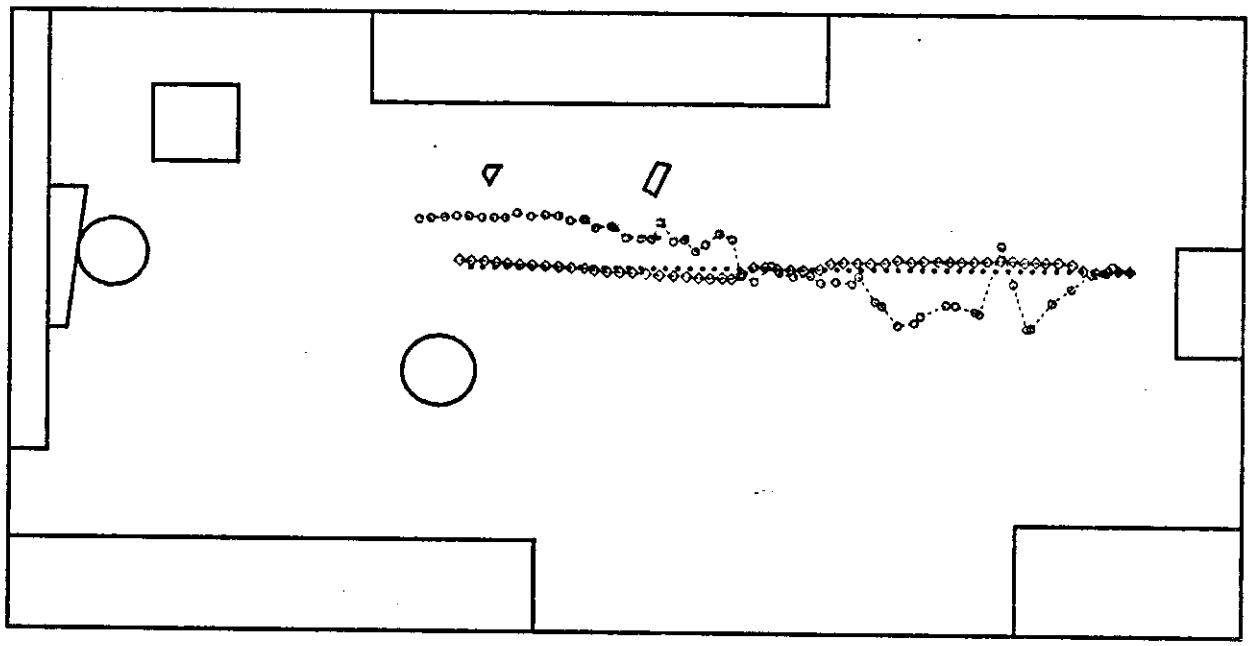

Figure 6-7: Position estimates obtained with 3 DOF algorithm and clean data.

The line of heavy dots shows the true position at every step, the path marked with circles shows the positions estimated with the spherical model, and the path marked with diamonds shows the same for the ellipsoidal model. The final position estimated with the ellipsoidal model was correct to within 2% of the distance and one degree of orientation. With the spherical model the corresponding figures were 8% and seven degrees.

In order to gauge the effect of noisier image matches, we adjusted the threshold of the prune stage so that progressivly fewer points were discarded. The general effect was to increasingly

underestimate the distance travelled, which is consistent with the results of increasing the random noise level in the simulations. Figure 6-8 shows what happened when the prune stage was entirely disabled, leaving only the correlation threshold to detect matching errors. Estimates with the spherical model were initially very bad. We attribute this to matching errors caused by large depth discontinuities around the foreground objects. When these objects fell out of view, the estimates were better behaved. The behavior with the ellipsoidal model was much less erratic, though biased.

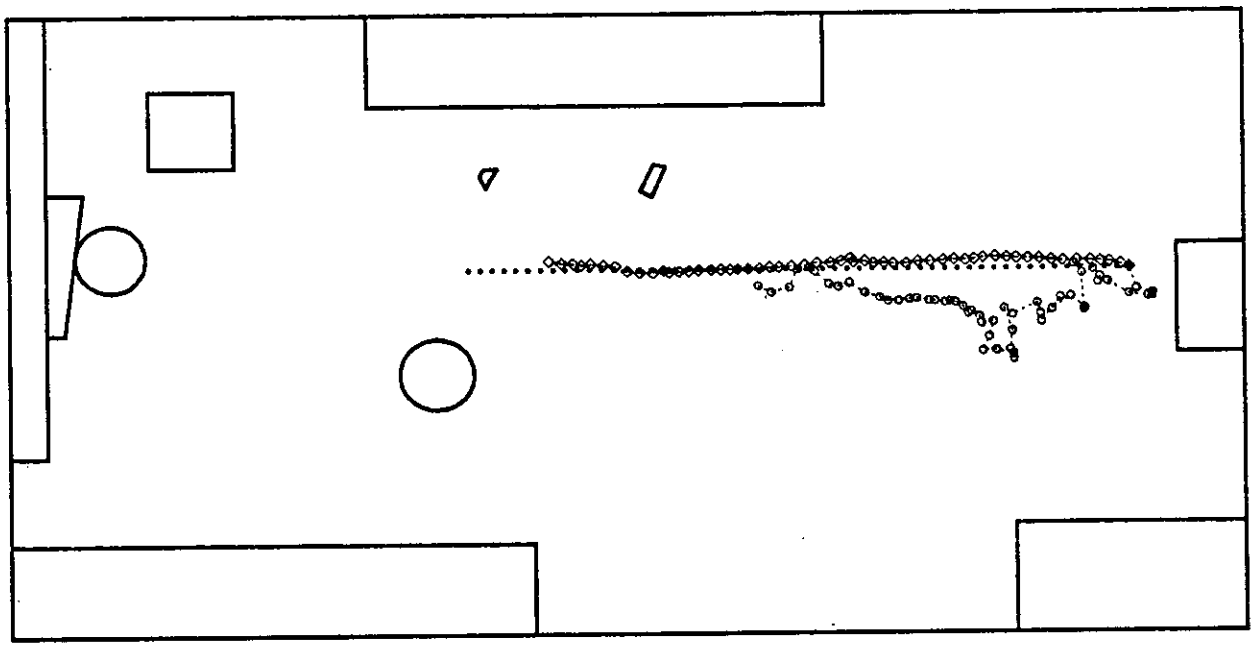

Figure 6-8: Results with noisy data.

Finally, we repeated the first experiment (ie. clean data) with the algorithm that computes all six degrees of freedom of motion. The results were in accord with the planar case, with roughly the same levels of error in the final position estimate. It was notable that with the spherical model the error in roll was less than a degree, while in the other rotations it was between five and twelve degrees. This is consistent with the observation made from the first simulation about coupled rotation and translation.

7. Conclusions

Comparing motion estimates obtained with the spherical (scalar) and ellipsoidal (3-D gaussian) error models, under the relatively long object distance to baseline ratios we examined there is no question that the ellipsoidal model is preferred. Simulations showed that position estimates with the ellipsoidal model had less variance and live trials confirmed that they were more accurate and less influenced by matching errors. With short object distance to baseline ratios this distinction will

diminish, and applications that can engineer this situation may be able to obtain satisfactory performance with the cheaper, scalar error model. We suspect that many applications will be such that the 3-D gaussian model will be valuable.

A caveat to the results we have described is the possibility of bias leading to underestimation of position. This results from high noise levels in image match coordinates and from large distances to objects. We attribute this effect to the non-gaussian nature of the true error distribution in these situations. This can be dealt with either by ensuring that the noise level is low or by explicitly modelling the non-gaussian error. The low noise level can probably be achieved in many situations with the use of matching constraints, calculating match coordinates to sub-pixel resolution, and effective error detection methods. Where this cannot be achieved, better modelling is an area for further research.

There remains the question of whether use of ellipsoidal error models tolerates shorter baselines than use of spherical error models. To date we have only tested this in simulation. Based on variance of the position estimates, a shorter baseline is possible. However, the bias issue is unresolved.

Perhaps the most valuable result is demonstrating that accurate position estimates can be achieved in a fully automatic system when an adequate error model is used. The true motion in the examples we showed was pure translation, but we believe that the results will hold for general motion and preliminary simulations bear this out. With matching to sub-pixel resolution, matching of extended features instead of points, and more sophisticated error detection, it may be possible to obtain much better performance than that quoted here. Another interpretation of our results is that they show the importance of error modelling in stereo and probably other aspects of vision. One area we plan to explore this is in shape from stereo, beginning with the local update paradigm of section 5.

Acknowledgments

We are indebted to Hans Moravec for making us aware of Kalman filters, Peter Highnam for introducing us to Schonemann's algorithm, and Takeo Kanade for pointing out the possibility of using a shorter baseline.

References

[1] G. Adiv.
Determining three-dimensional motion and structure from optical flow generated by several moving objects.
IEEE Trans. on Pattern Analysis and Machine Intelligence PAMI-7(4):384-401, July, 1985.

[2] P. Anandan and R. Weiss.
Introducing a smoothness constraint in a matching approach for the computation of displacement fields.
In *Proc. of ARPA IUS Workshop*, pages 186-197. SAIC, December, 1985.

[3] N. Ayache and O.D. Faugeras.
HYPER: A new approach for the recognition and positioning of two-dimensional objects.
IEEE Trans. on Pattern Analysis and Machine Intelligence PAMI-8(1):44-54, January, 1986.

[4] H.S. Baird.
Model-based Image Matching Using Location.
MIT Press, Cambridge, Mass., 1985.

[5] T.J. Broida and R. Chellappa.
Estimation of motion parameters from noisy images.
IEEE Trans. on Pattern Analysis and Machine Intelligence PAMI-6(1):90-99, January, 1986.

[6] R.A. Brooks.
Symbolic reasoning among 3-D models and 2-d images.
Artifical Intelligence 17:285-348, 1981.

[7] L. Dreschler and H.-H. Nagel.
Volumetric model and 3D trajectory of a moving car derived from monocular TV frame sequences of a street scene.
Computer Graphics and Image Processing 20:199-228, 1982.

[8] T.F. Elbert.
Estimation and Control of Systems.
Van Nostrand Reinhold Co., New York, NY, 1984.

[9] O.D. Faugeras, N. Ayache, B. Faverjon, F. Lustman.
Building visual maps by combining noisy stereo measurements.
In *IEEE Int'l Conf. on Robotics and Automation*, pages 1433-1438. IEEE, April, 1986.

[10] A. Gelb (editor).
Applied Optimal Estimation.
MIT Press, Cambridge, MA, 1974.

[11] D.B. Gennery.
Modelling the environment of an exploring vehicle by means of stereo vision.
PhD thesis, Stanford University, June, 1980.

[12] D.B. Gennery.
Tracking known three-dimensional objects.
In *Proc. of AAAI*, pages 13-17. AAAI, 1982.

[13] P.E. Gill, W. Murray, and M.H. Wright.
Practical Optimization.
Academic Press, 1981.

[14] W.E.L. Grimson and T. Lozano-Perez.
Model-based recognition and localization from sparse range or tactile data.
International Journal of Robotics Research 3(3):3-35, Fall, 1984.

[15] J. Hallam.
Resolving observer motion by object tracking.
In *Int'l Joint Conf. on Artificial Intelligence*. 1983.

[16] M. Hebert.
Reconnaissance de formes tridimensionelles.
PhD thesis, L'Universite de Paris-Sud, Centre d'Orsay, September, 1983.

[17] L.H. Matthies and C.E. Thorpe.
Experience with visual robot navigation.
In *Proc. of IEEE Oceans'84 Conf.*. IEEE, Washington, D.C., August, 1984.

[18] H.P. Moravec.
Obstacle avoidance and navigation in the real world by a seeing robot rover.
PhD thesis, Stanford University, September, 1980.

[19] P.H. Schonemann and R.M. Carroll.
Fitting one matrix to another under choice of a central dilation and a rigid motion.
Psychometrika 35(2):245-255, June, 1970.

[20] C.C. Slama (editor-in-chief).
Manual of Photogrammetry.
American Society of Photogrammetry, Falls Church, Va., 1980.

[21] R.C. Smith and P. Cheeseman.
On the representation and estimation of spatial uncertainty.
Technical Report (draft), SRI International, 1985.

[22] F. Solina.
Errors in stereo due to quantization.
Technical Report MS-CIS-85-34, U. Pennsylvania, September, 1985.

[23] C.E. Thorpe.
Vision and navigation for a robot rover.
PhD thesis, Carnegie-Mellon University, December, 1984.

[24] R.Y. Tsai and T.S. Huang.
Uniqueness and estimation of three-dimensional motion parameters of rigid objects with curved surfaces.
IEEE Trans. on Pattern Analysis and Machine Intelligence 6(1):13-26, January, 1984.

[25] A.M. Waxman and J.J. Duncan.
Binocular image flows.
In IEEE (editor), *Proc. of Workshop on Motion: Representation and Analysis*, pages 31-38. May, 1986, May, 1986.

[26] J.R. Wertz (ed).
Spacecraft Attitude Determination and Control.
D. Reidel Publishing Company, 1978.