Visual Estimation of 3-D Line Segments From Motion — A Mobile Robot Vision System [1]

William M. Wells III
Artificial Intelligence Center
SRI International
333 Ravenswood Avenue
Menlo Park, Ca 94025
ARPANET: Wells@ai.ai.sri.com

Abstract

An efficient technique is presented for detecting, tracking and locating three-dimensional (3-D) line segments. The utility of this technique has been demonstrated by the SRI mobile robot, which uses it to locate features in an office environment in real time (one Hz frame rate). A formulation of Structure-from-Motion using line segments is described. The formulation uses longitudinal as well as transverse information about the endpoints of image line segments. Although two images suffice to form an estimate of a world line segment, more images are used here to obtain a better estimate. The system operates in a sequential fashion, using prediction-based feature detection to eliminate the need for global image processing.

I. Introduction

Three-dimensional (3-D) visual sensing is a useful capability for mobile robot navigation. However, the need for real-time operation using compact, on-board equipment imposes constraints on the design of 3-D vision systems. For the SRI mobile robot, we have chosen to use a feature-based system whose features are image and world line segments. Line segments as features provide a practical compromise between curves, which are complex to analyze, and point features, which are often sparse in man-made environments.

We use a relatively fast frame rate (one Hz) to reduce the complexity of the feature correspondence problem. Because features don't move very far in closely spaced images, little searching is needed to find a feature's successor. Combining a fast frame rate with prediction-based feature detection can greatly reduce the portion of the image to which feature detectors must be applied. Another benefit of tracking world features in closely spaced images is that volumetric free-space information is readily available.

Real-time 3-D vision may be further simplified by avoiding the Motion-from-Structure [Ullman, 1979] problem. We derive camera poses from odometry. (Inertial navigation systems are becoming increasingly practical for this purpose.) Because the vision system is used for navigation among stable objects, we need be concerned only with estimating the locations of stable features in the world. We use other sensors for rapidly moving objects.

[1]This research was supported in part by contract SCA 50-1B from General Motors Corporation.

With these design parameters, we are faced with a problem of Structure-from-Motion [Ullman, 1979] in estimating a static world feature from its observation in a sequence of images as the camera is moved. We have devised a simple formulation of Structure-from-Motion that is based on line segments. It uses simple vector and 3-by-3 matrix operations. The most complicated aspect of the formulation is the inversion of 3-by-3 matrices.

II. Overview

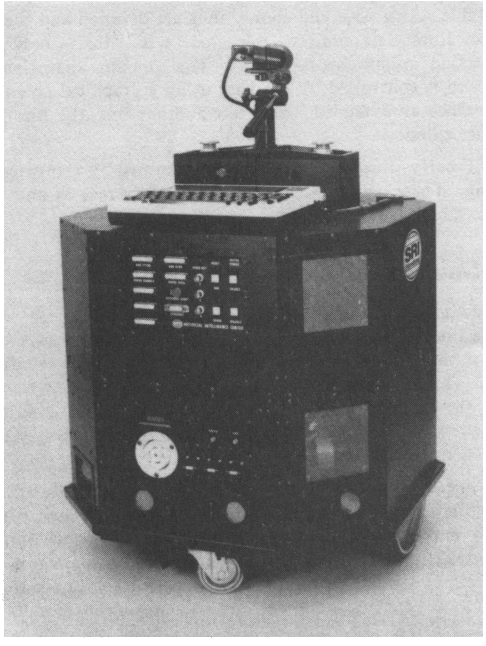

Figure 1: SRI Mobile Robot

Here we describe the vision system as it is implemented on the SRI mobile robot.

The SRI mobile robot [Reifel, 1987] is equipped with an on-board video camera, frame buffer, and 68010 computer system (Figure 1). Optical shaft encoders coupled to the two main drive wheels provide odometric data that are used to derive camera poses.

We use closely spaced images to reduce the complexity of the feature correspondence problem. Combining closely spaced images with prediction-driven feature detection allows the application of edge operators to be limited to

small areas of the image that are near predictions, thus eliminating the need for global image processing. (Prediction based feature detection was used to advantage in Goad's model based vision system [Goad, 1986].) Image line segments are detected by a software edge tracker that provides least-squares fits of line segments to linear image edge features. The edge tracker is directed by prototype image segments whose origin will be described below. The tracker finds sets of candidate segments that are close to each prototype. (The measure of such closeness is discussed in section III.B..) We require candidate edge segments to have the same gradient sense or "contrast polarity" as their predecessors.

Our system uses a sequential 3-D line segment estimator to infer world line segments from sequences of corresponding image line segments. The system operates in three phases: "prospecting," "bootstrapping," and "sequential updating." "Prospecting" segments, the first prototype segments the system uses, are generated so that the feature detection component will find new image features. The "bootstrapping" phase is then used as a prelude to the "sequential updating phase." All prototypes generated during bootstrapping are segments that were detected in the preceding image. While bootstrapping, we entertain alternative hypotheses about a feature's successors in a small tree of possible correspondence sequences. When the tree achieves a minimum depth, we use a nonsequential form of the 3-D segment estimator (described in section III.D.) to generate a world feature estimate as well as a consistency measure for each sequence in the tree. If the most consistent sequence meets a minimum consistency threshold, it is promoted to sequential updating; otherwise, it is discarded.

During the "sequential updating" phase, we use the sequential form of the 3-D segment estimator (section III.D.). Newly detected image features are folded into world feature estimates as each new picture arrives. Previous 3-D estimates are used to generate prototype segments to direct the feature detector. The prototype segments are generated by taking the central projections of the previous 3-D segment estimates into the image plane using the new camera pose. The detected image feature that is closest to the prototype is, if close enough, used as the successor.

The system tracks a set of environmental features by maintaining several prospecting, bootstrapping, and updating subsystems. The system provides good estimates of line segments in the robot's environment. (For example, the robot can locate a door jamb in its vicinity to within a few centimeters.) It will detect 3-D segments in most orientations. Some combinations of segment orientation and robot motion lead to degeneracies; for example, when the camera moves parallel to a segment which is viewed end on, or when the robot is moving parallel to a segment whose ends are clipped by the image boundary.

The robot finds walls by fitting planes to sets of perceived 3-D segments. These segments are grouped using a co-planarity measure. Once the walls have been located the robot servos to a path which is centered between the walls.

Figure 2 shows an intensity image the robot saw in a hallway. Figure 3 displays a stereo view of an unedited collection of line segments that were estimated by the robot and used to guide its path down the hallway. The frame

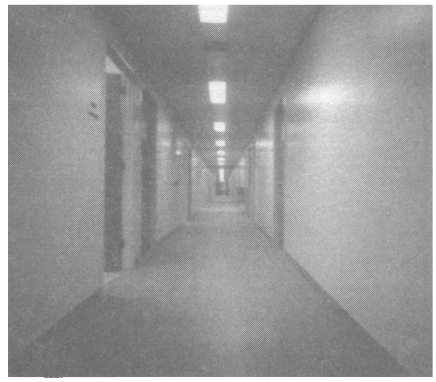

Figure 2: Hallway

rate was one Hz, while the robot moved at 30 mm/s. Most of the segments that the robot gathered were vertical. This is a consequence of the way the "prospecting" segments are arranged, the motion of the robot, and the characteristics of the hallway.

Occasionally the system will encounter a seemingly consistent set of miscorrespondences, which will lead to an incorrect hypothesis surviving to the sequential updating phase. Such hypotheses fail quickly when subjected to the long-term consistency requirement.

In the future, we plan to investigate the use of acquired models within this framework. Such models may provide a means to measure the motion of the robot using Motion-from-Structure. Models may also make it possible to track moving objects. We plan to increase the frame rate of the vision system by installing a 68020 computer in the robot, perhaps using several CPU boards.

III. Estimation of 3-D Line Segments

In this section, we present a simple formulation of Structure-from-Motion that is based on line segments. It uses longitudinal as well as transverse information about segment endpoints. Given a sequence of images with corresponding line segment features, we estimate a 3-D line segment that minimizes an image error measure summed over the sequence of images. Camera poses are assumed to be known.

In section III.A., we discuss the choice of line segments as features to be used within the paradigm of Structure-from-Motion. We then define an image-based measure of the discrepancy between two line segments (section III.B.). In section III.C., we express the error measure in terms of a world line segment and its projection as detected in the image. We then estimate a 3-D segment which best fits a sequence of observations by varying the segment to minimize the error measure summed over a sequence of images. This yields a problem of nonlinear minimization. In section III.D., we describe a sequential estimator that linearizes the problem. The robot uses an implementation of this linearized sequential estimator to estimate 3-D world line segments.

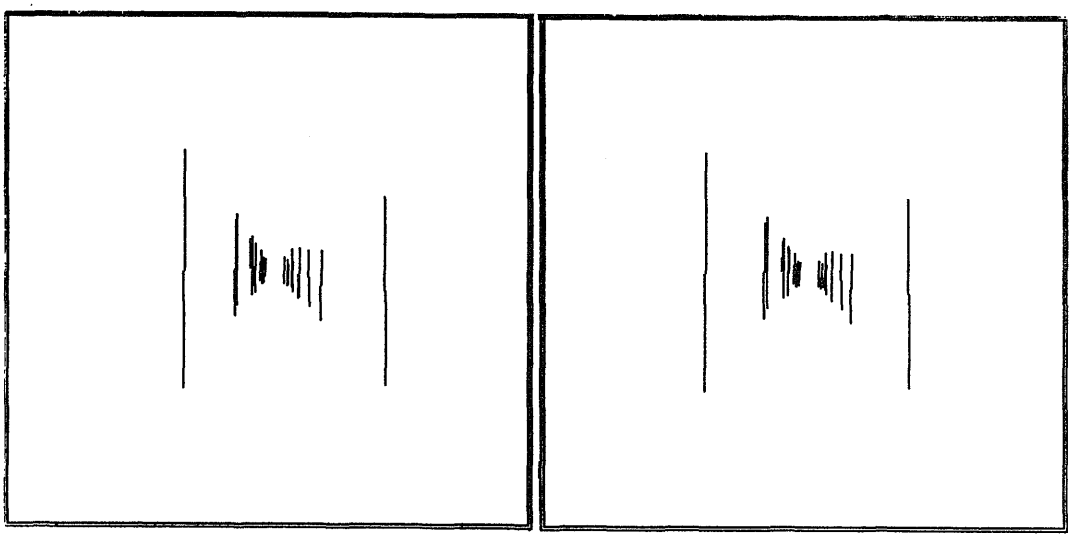

Figure 3: Estimated Line Segments

A. Simple Structure-from-Motion Using Line Segments

Structure-from-Motion is a useful and popular paradigm for robotic visual perception [Aggarwal, 1986]. Early work in feature-based motion analysis was based on world and image points [Roach and Aggarwal, 1979] [Longuet-Higgens, 1981] [Hannah, 1980] [Gennery, 1982], while later research focused on straight lines [Yen and Huang, 1983]. Points and lines also have been used widely in robotic vision [Goad, 1986] [Lowe, 1985].

Straight line *segments* are useful features for motion analysis and robotic vision applications [Ullman, 1979] . Point features are as simple to analyze, but unfortunately, prominent point features can be scarce, particularly in man-made environments. Cultural and industrial scenes usually contain prominent linear features that can be reliably detected by edge finders. Although cultural and industrial scenes often also have significant curved features, such features are more difficult to analyze than points or lines.

Edge finders are very good at determining the transverse position of a linear feature in an image. They are less accurate at finding the longitudinal (along the edge) position of the *ends* of a linear feature, as they usually use thresholds to locate feature terminations. Although the longitudinal information is less reliable than the transverse information, we believe that it is still useful information, which would be lost if linear features were abstracted into straight lines rather than line segments. Line segments carrying endpoint information present a balance between analytical simplicity and practicality as image features.

B. Image Error Measure

We propose the following as a component of the measure of the discrepancy between a pair of image line segments (Figure 4):

$$\epsilon = [\alpha(\mathrm{P}-\mathrm{S})\cdot\hat{\mathrm{L}}]^2 + [\beta(\mathrm{P}-\mathrm{S})\cdot\hat{\mathrm{O}}]^2 \quad . \qquad (1)$$

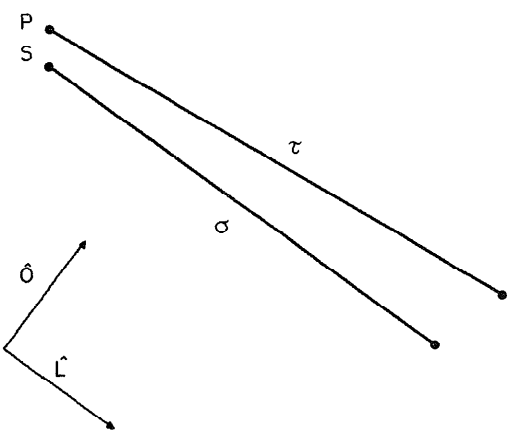

Figure 4: Image Error Measure

Here ϵ represents the squared error due to one pair of corresponding endpoints. The total error for the corresponding segments is the sum of the errors for both corresponding endpoint pairs. P and S are two-vectors describing the image locations of endpoints of line segments σ and τ respectively. $\hat{\mathrm{L}}$ is a unit vector parallel to σ, while $\hat{\mathrm{O}}$ is a unit vector perpendicular to σ.

The longitudinal and perpendicular components are weighted by α and β. We have settled on $\beta/\alpha = 16$ empirically, giving perpendicular errors 16 times the weight of longitudinal errors. This was deemed to be the smallest weighting of longitudinal errors that provided estimates that were "reasonably" accurate longitudinally, while not overly disturbing the transverse components of the estimates with less reliable longitudinal information.

If an image line segment is clipped by the boundaries of an image, that endpoint has little meaningful longitudinal information. One strategy for this case sets α to

zero, ignoring the longitudinal information in that particular image.

C. 3-D Error Measure

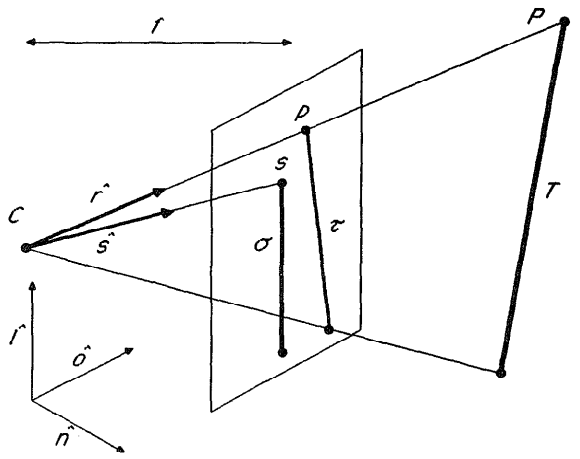

Figure 5: Imaging Geometry

We may recast Eq.(1) in terms of world 3-vectors (Figure 5). An endpoint of 3-D segment T is P, their central projections into the image plane are τ and p respectively. The endpoint of σ that corresponds to p is s. Here, p and s are 3-vectors that refer to locations of image points in the 3-space in which the image plane is embedded. C is the projection center of the camera, and f is the focal length of the camera. The image plane and σ define an orthonormal basis composed of $\hat{l}$, which is a 3-D unit vector parallel to σ; $\hat{n}$, which is a 3-D unit vector normal to the image plane; and $\hat{o}$, which is perpendicular to both $\hat{l}$ and $\hat{n}$. Two additional unit vectors are defined by $\hat{r}$, which is the normalization of $(P-C)$, and $\hat{s}$, which is the normalization of $(s-C)$.

The image error measure may rewritten as:

$$\epsilon = \alpha^2[(p-s)\cdot\hat{l}]^2 + \beta^2[(p-s)\cdot\hat{o}]^2 \quad . \qquad (2)$$

Next, we express the error measure in terms of more convenient unit vectors. Let

$$\begin{aligned} \hat{O} &= normalize(\hat{s}\times\hat{l}) \quad , \\ \hat{L} &= normalize(\hat{s}\times\hat{o}) \quad , \end{aligned}$$

and

$$\begin{aligned} \omega_o &= \hat{O}\cdot\hat{o} \\ \omega_n &= \hat{O}\cdot\hat{n} \\ \lambda_l &= \hat{L}\cdot\hat{l} \\ \lambda_n &= \hat{L}\cdot\hat{n} \quad . \end{aligned}$$

Then we can express $\hat{O}$ and $\hat{L}$ in terms of $\hat{l}$, $\hat{o}$, and $\hat{n}$:

$$\begin{aligned} \hat{O} &= \omega_o\hat{o} + \omega_n\hat{n} \\ \hat{L} &= \lambda_l\hat{l} + \lambda_n\hat{n} \quad . \end{aligned}$$

Noting that $\hat{n}\cdot(p-s)=0$, we may rewrite Eq. (2) as

$$\epsilon = \alpha^2[\frac{1}{\lambda_l}(p-s)\cdot(\lambda_l\hat{l}+\lambda_n\hat{n})]^2 + \beta^2[\frac{1}{\omega_o}(p-s)\cdot(\omega_o\hat{o}+\omega_n\hat{n})]^2 \quad ,$$

or

$$\epsilon = \frac{\alpha^2}{\lambda_l^2}[(p-s)\cdot\hat{L}]^2 + \frac{\beta^2}{\omega_o^2}[(p-s)\cdot\hat{O}]^2 \quad .$$

Since $s = C + \delta\hat{s}$ for some δ, $\hat{s}\cdot\hat{O}=0$, and $\hat{s}\cdot\hat{L}=0$, we may write

$$\epsilon = \frac{\alpha^2}{\lambda_l^2}[(p-C)\cdot\hat{L}]^2 + \frac{\beta^2}{\omega_o^2}[(p-C)\cdot\hat{O}]^2 \quad . \qquad (3)$$

Now we will use a relation of central projection to get the error in terms of P rather than p. The standard "z-division" form of central projection may be written as follows (Figure 5):

$$(p-C) = \frac{f}{z}(P-C) \qquad \text{where} \qquad z = (P-C)\cdot\hat{n} \quad .$$

Letting

$$\begin{aligned} a &= \frac{\alpha f}{\lambda_l} \\ b &= \frac{\beta f}{\omega_o} \quad , \end{aligned}$$

Eq. (3) may be written as

$$\epsilon = \frac{1}{z^2}\{a^2[(P-C)\cdot\hat{L}]^2 + b^2[(P-C)\cdot\hat{O}]^2\} \quad . \qquad (4)$$

If we consider ϵ to be the squared error for a given endpoint due to detection in the ith member of a set of images, then the total error for a given endpoint would be given by

$$E = \sum_i \frac{1}{z_i^2}\{a_i^2[(P-C_i)\cdot\hat{L}_i]^2 + b_i^2[(P-C_i)\cdot\hat{O}_i]^2\} \quad .$$

Varying P to minimize E will yield an estimate for the 3-D segment endpoint. This is a nonlinear estimation problem by virtue of the factor of $1/z_i^2$.

D. Approximation and Minimization

There are many ways to minimize E. We will discuss a sequential method that works well in practice, which is designed for an application where a set of images arrives sequentially and where an estimate of the 3-D feature is desired after each image. This is often the case in robotic guidance.

The technique involves approximating $z_i = z(P)$ by $z_- = z(P_-)$, where P_- is the previous estimate of P. The process may be bootstrapped by using a nominal starting value for z_i. This method essentially substitutes a "pseudo-orthographic" approximation (a different approximation for each image) for perspective projection. The error terms ϵ_i become invariant to translations of P along $\hat{s}_i$. The approximation is exact for points on one plane in the world, namely the plane containing P_- that is parallel to the image plane. Within the framework of the minimization, this is also equivalent to replacing the (unsquared) error functions of P by second-order Taylor expansions.

The expansions are about the point where the ray emanating from C_i along $\hat{s}_i$ pierces the previously mentioned plane. The approximated squared error measure is also easy to visualize, as it is the weighted sum of the squared perpendicular distances of P from a pair of planes. The two planes both contain the camera center and the endpoint s of σ. One contains the segment σ, while the unit vector $\hat{o}$ lies in the other.

After this approximation, the ith error (Eq. (4)) may be written as

$$\epsilon_i = \frac{a_i^2}{z_{i-}^2}[(P - C_i) \cdot \hat{L}_i]^2 + \frac{b_i^2}{z_{i-}^2}[(P - C_i) \cdot \hat{O}_i]^2$$

This is quadratic in P and its sum is easy to minimize. In matrix notation,

$$\begin{aligned}\epsilon_i &= \frac{a_i^2}{z_{i-}^2}(\mathbf{P} - \mathbf{C}_i)^T \hat{\mathbf{L}}_i \hat{\mathbf{L}}_i^T (\mathbf{P} - \mathbf{C}_i) + \\ &\quad \frac{b_i^2}{z_{i-}^2}(\mathbf{P} - \mathbf{C}_i)^T \hat{\mathbf{O}}_i \hat{\mathbf{O}}_i^T (\mathbf{P} - \mathbf{C}_i)\end{aligned}$$

Let

$$\mathbf{M}_i = \frac{a_i^2}{z_{i-}^2}\hat{\mathbf{L}}_i\hat{\mathbf{L}}_i^T + \frac{b_i^2}{z_{i-}^2}\hat{\mathbf{O}}_i\hat{\mathbf{O}}_i^T \quad ;$$

then

$$\epsilon_i = (\mathbf{P} - \mathbf{C}_i)^T \mathbf{M}_i (\mathbf{P} - \mathbf{C}_i) \quad ,$$

or

$$\epsilon_i = \mathbf{P}^T\mathbf{M}_i\mathbf{P} - 2\mathbf{P}^T\mathbf{M}_i\mathbf{C}_i + \mathbf{C}_i^T\mathbf{M}_i\mathbf{C}_i \quad .$$

Defining

$$\begin{aligned}\mathbf{M} &= \sum_i \mathbf{M}_i \\ \mathbf{V} &= \sum_i \mathbf{M}_i\mathbf{C}_i \\ k &= \sum_i \mathbf{C}_i^T\mathbf{M}_i\mathbf{C}_i\end{aligned}$$

allows us to write the total squared error as

$$E = \mathbf{P}^T\mathbf{M}\mathbf{P} - 2\mathbf{P}^T\mathbf{V} + k \quad .$$

Setting the gradient of E with respect to $\mathbf{P}$ to zero,

$$0 = \nabla_{\mathbf{P}} E = 2\mathbf{M}\mathbf{P} - 2\mathbf{V} \quad ,$$

or

$$\mathbf{P} = \mathbf{M}^{-1}\mathbf{V} \quad ,$$

provides an easily computed estimate of a 3-D line segment endpoint viewed in a sequence of images.

Two images are sufficient for computing an estimate of a line segment. If the camera motion is slight, making the effective baseline short, then the estimate may be somewhat inaccurate in depth. If more images are used and the camera moves appreciably about some feature in the world, then the estimate of that feature improves and the consistency of the estimate may be better evaluated.

There are combinations of line segment orientation and camera motion which are degenerate and preclude depth estimates. In these situations $\mathbf{M}$ will be singular, or nearly so in the presence of noise.

IV. Conclusion

We have described an efficient technique for detecting, tracking, and locating three-dimensional line segments as demonstrated on the SRI mobile robot. As the robot moves about, it makes good estimates of environmental 3-D line segments using Structure-from-Motion.

In the future, we plan to investigate whether the statistical characteristics of the image line segment detector can provide a maximum-likelihood basis for the estimator. This would also yield values for for the weights α and β which appear above.

Acknowledgments

I thank David Marimont for our many conversations about robotic vision, including some on the general topic of the longitudinal information of endpoints of line segments. Marimont's thesis [Marimont, 1986] provides a good discussion of feature estimation for robotics.

References

[Aggarwal, 1986] J. K. Aggarwal. Motion and time-varying imagery — an overview. In *Workshop on Motion: Representation and Analysis*, pages 1–6, IEEE, Charleston, South Carolina, May 1986.

[Gennery, 1982] Donald B. Gennery. Tracking known three-dimensional objects. In *Proceedings of the National Conference on Artificial Intelligence*, pages 13–17, August 18–20 1982.

[Goad, 1986] Chris Goad. Fast 3-D Model Based Vision. In Alex P. Pentland, editor, *From Pixels to Predicates*, Ablex Publishing Co., 1986.

[Hannah, 1980] Marsha Jo Hannah. Bootstrap stereo. In *Proceedings of the First Annual National Conference on Artificial Intelligence*, pages 38–40, American Association for Artificial Intelligence, 1980.

[Longuet-Higgens, 1981] H. C. Longuet-Higgens. A computer algorithm for reconstructing a scene from two projections. *Nature*, 293:133–135, 10 September 1981.

[Lowe, 1985] David G. Lowe. *Perceptual Organization and Visual Recognition.* Kluwer Academic Publishers, 1985.

[Marimont, 1986] David Henry Marimont. *Inferring Spatial Structure from Feature Correspondences.* Ph.D. thesis, Stanford University, 1986.

[Reifel, 1987] Stanley W. Reifel. *The SRI Mobile Robot Testbed, A Preliminary Report.* Technical Report 413, SRI International Artificial Intelligence Center, 1987.

[Roach and Aggarwal, 1979] J. W. Roach and J. K. Aggarwal. Computer tracking of objects moving in space. *IEEE Transactions PAMI*, PAMI-1(2):127–135, April 1979.

[Ullman, 1979] Shimon Ullman. *The Interpretation of Visual Motion.* MIT Press, 1979.

[Yen and Huang, 1983] B. L. Yen and T. S. Huang. Determining 3-D motion and structure of a rigid body using straight line correspondences. In *Proceedings of the International Joint Conference on Acoustics, Speech and Signal Processing*, March 1983.