Algorithms for Matching 3D Line Sets

Behzad Kamgar-Parsi, *Member*, *IEEE*, and Behrooz Kamgar-Parsi, *Member*, *IEEE*

Abstract—Matching two sets of lines is a basic tool that has applications in many computer vision problems such as scene registration, object recognition, motion estimation, and others. Line sets may be composed of infinitely long lines or finite length line segments. Depending on line lengths, three basic cases arise in matching sets of lines: 1) finite-finite, 2) finite-infinite, and 3) infinite-infinite. Case 2 has not been treated in the literature. For Cases 1 and 3, existing algorithms for matching 3D line sets are not completely satisfactory in that they either solve special situations, or give approximate solutions, or may not converge, or are not invariant with respect to coordinate system transforms. In this paper, we present new algorithms that solve exactly all three cases for the general situation. The algorithms are provably convergent and invariant to coordinate transforms. Experiments with synthetic and real 3D image data are reported.

Index Terms—Line matching, motion estimation, object recognition, pose estimation, 3D registration.

1 Introduction

Matching geometric features such as points, lines, surfaces, etc., is a basic tool in computer vision that has applications in scene registration, object localization and recognition, motion estimation, and others. In this paper, we discuss matching two sets of N corresponding 3D lines. The two sets of lines may be extracted from a 3D model and a 3D image, or from two 3D images. The 3D images may be acquired directly with ladar imaging, tactile sensing, etc., or from reconstructed imaging such as stereo imaging. We refer to one set of lines as the model and the other set as the image. We want to find the rigid transformation (translation and rotation) that gives the best match (fit, alignment, registration) of the image with the model.

We assume that we have already detected the lines (i.e., line segments or infinitely long lines) and hypothesized the corresponding pairs of lines in the model and the image. We should make two remarks. First, detection of 3D lines is still a nontrivial problem in early vision, which is not addressed here. Second, often a challenging aspect of problems in computer vision is determining the correspondences. Certain pairings of features in the image and model can be eliminated by geometric constraints, e.g., the rigidity constraint, all others have to be examined based on the goodness of match [5]. Indeed, reliable and fast matching algorithms are needed to verify hypothesized correspondences. We should note that, even though we hypothesize line correspondences, we still have to determine the *corresponding portions* of corresponding pairs of lines in image and model, simultaneously with the best rotation and translation.

- *B. Kamgar-Parsi is with Code 311, Office of Naval Research, 800 N. Quincy St., Arlington, VA 22217. E-mail: kamgarb@onr.navy.mil.*
- *B. Kamgar-Parsi is with Code 5515, Naval Research Laboratory, Washington, DC 20375. E-mail: kamgar@aic.nrl.navy.mil.*

Manuscript received 10 Jan. 2003; revised 8 Aug. 2003; accepted 22 Nov. 2003.
Recommended for acceptance by R. Basri.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number 118117.

Line sets may consist of line segments or infinitely long lines. A line segment has as usable information the line location, direction, and length, while an infinite line is characterized by its location and direction only. Thus, algorithms for matching sets of line segments are not applicable to matching sets of infinite lines, and vice versa. Since we can have infinitely long lines or finite-length line segments, then we may encounter the following three basic cases in line matching: 1) model and image both consist of finite line segments; 2) model consists of finite line segments, while image lines are infinite; and 3) model and image both consist of infinite lines. Fig. 2, for example, illustrates Case 1, where the model and its edges, as well as edge fragments detected in the image are shown. Mixed cases, where some line segment lengths are known and others are not, can be handled by combining the three basic cases. See the Appendix.

The problem of matching 3D line sets has received a fair amount of attention. To our knowledge, a solution for Case 2 has not appeared in the literature. For Cases 1 and 3, solutions have been proposed. However, these methods do not completely solve the general matching problem. They are either noninvariant to the choice of the coordinate system [3], [4], [12], [18], or cannot be guaranteed to converge [4], or are approximate solutions [3], [22], or contain implicit assumptions that lead to solutions for special situations [6], [22]. In this paper, we present a method for solving all three cases that circumvents the shortcomings mentioned above. In our approach, we formulate the cost function explicitly in terms of *corresponding points* on pairs of corresponding lines. In this way, we avoid implicit assumptions that could lead to solutions for special situations. Moreover, this approach leads to convergent algorithms that yield the best match without the need for approximations involving decoupling of rotation and translation. Below, we first present a brief review of existing methods, and then describe our approach.

1.1 Previous Work

For Cases 1 and 3, several methods have been proposed in the literature, while, to our knowledge, solutions for Case 2 have not appeared.

For Case 1, Zhang and Faugeras [22] were the first to present an algorithm for matching line segments. There is, however, an implicit assumption in this algorithm, namely, that the midpoints of the corresponding line segment pairs are the corresponding points. They propose to estimate the best match using a Kalman filter approach. They also suggest a gradient descent method for minimizing an objective function. In [21], the authors use line lengths only to rule out corresponding line hypotheses, but do not use them in the matching algorithm. Guerra and Pascucci [6] present a different method for matching two sets of 3D line segments where line correspondences are not known. They use a Hausdorff distance between segments, and use line triplets to compute a very large number of transformations, and goodness of match measures from which the best match is decided. However, this algorithm also contains the same implicit assumption (i.e., the midpoints of the corresponding segments are the corresponding points of the lines), which leads to the correct solution for this special situation. The computational complexity of this algorithm is at best $O(N^4 \log N)$.

For Case 3, in a pioneering paper, Faugeras and Hebert [4] formulate the matching problem in terms of an objective function that is a sum of line orientations and locations. They propose a Kalman filter approach to iteratively find the best match, however, the improvements cannot be guaranteed to converge. In [22], Zhang and Faugeras also treat Case 3 and build on the approach proposed in [4] and present an approximate solution, where the rotation matrix is obtained first using only the line orientations, and then translation is calculated from the rotation matrix and the line locations. Walker et al. [18] and Daniilidis [3] propose an interesting solution based on dual quaternion representation of lines. This representation is implicitly equivalent to setting the overlap length to infinity, which results in line orientations completely dominating line locations. The algorithm yields an approximate solution to the full matching problem, where second order couplings between rotation and translation are ignored. Because of decoupling of rotation and translation, the solution may be obtained in closed-form. We discuss these approaches in more detail in Section 5.

Bartoli and Sturm [1] also present algorithms for matching sets of reconstructed 3D lines and line segments. However, the cost functions they propose are asymmetric and are defined with respect to matches in a 2D image rather than matches in 3D as in this work and others cited above.

1.2 Outline of the Approach

Suppose $A = \{A_n\}$ with $n = 1, \ldots, N$ denotes the model line set. Here, $A_n = (\mathbf{a}_n, \hat{\mathbf{b}}_n, L_n)$ is a line segment represented by one of its points $\mathbf{a}_n$, its direction unit vector $\hat{\mathbf{b}}_n$, and its length L_n. Similarly, we show the image line set by $X = \{X_n\}$, where $X_n = (\mathbf{x}_n, \hat{\mathbf{y}}_n, l_n)$. Suppose that $\mathbf{a}_n$ and $\mathbf{x}_n + s_n\hat{\mathbf{y}}_n$ are a pair of corresponding points on A_n and X_n, where s_n is a scalar parameter that we have to determine. The distance measure between these two lines, $D(A_n, X_n)$, is thus the sum of the distances between all pairs of corresponding points, i.e.,

$$D(A_n, X_n) = \int_{\Omega_n} du \cdot dist(\mathbf{a}_n + u\hat{\mathbf{b}}_n, \mathbf{x}_n + s_n\hat{\mathbf{y}}_n + u\hat{\mathbf{y}}_n),$$

where u is a scalar variable that parameterizes the lines, Ω_n is the overlap between A_n and X_n, and $dist(\mathbf{a}, \mathbf{x})$ is the distance function between points $\mathbf{a}$ and $\mathbf{x}$. If we show the rigid transformation of the image set by $\mathcal{T}$, then the distance measure between the model and the transformed image is

$$M(A, \mathcal{T}X) = \sum_{n=1}^{N} D(A_n, \mathcal{T}X_n).$$

While the above definition for $D(A_n, X_n)$ is physically meaningful and valid for Cases 1 and 2, because the overlap lengths Ω_n are finite, it needs to be modified for Case 3 since it becomes infinite for infinite lines. Hence, we have to limit the overlap length. That is, we have to make a compromise between the importance of line *location* and line *orientation*. By choosing a longer overlap length, we would be favoring orientation and vice versa.

The best match between the two sets is obtained by minimizing $M(A, \mathcal{T}X)$ over all possible $\mathcal{T}$ and $\{s_n\}$. This is a $(N+6)$-variable minimization problem-6 for the rigid transform $\mathcal{T}$ and N for the set of corresponding points $\{s_n\}$. It can be proved that this is a nonlinear problem even for the most tractable $dist$-function, namely, Euclidean L_2 norm. The algorithm we present for solving this problem is based on matching equal-length pieces of A_n and X_n. In this way, the nonlinear minimization can be solved by an iterative solution of two sets of linear problems. The algorithm is provably convergent. It has $O(N)$ complexity. Extensive empirical evidence shows that it converges in a few iterations and finds the best match for almost all algorithm initializations.

More importantly, for Case 3, we show that none of the proposed solutions (i.e., FH [4], dual quaternion [3], [18], or the convergent method we presented in [12]) are invariant with respect to coordinate system translations, which renders meaningless the best matches found with these methods. We examine the source of this noninvariance, which is in the inherent ambiguity of representing infinite lines, and present a solution that is invariant. Moreover, this solution is closed-form.

The line matching algorithms we present here are built upon a closed-form solution for matching sets of corresponding line segments with *equal-lengths* that have appeared in the literature [7], [11]. We give a brief discussion of this closed-form solution in Section 2, and present Cases 1, 2, and 3 in Sections 3, 4, and 5, respectively.

A related and even more challenging problem is matching 2D lines (line segments) in an image to a 3D model, or matching sets of lines in two images via 3D line reconstructions. A large number of papers on this important problem have been published, e.g., see [1], [17] for references. We will not discuss this problem here, and only remark that the solutions to this problem also have some of the shortcomings mentioned above.

The method we present in this paper, as well as those cited above, are based on regression. The evidence accumulation approach (e.g., the Hough transform) has also been used for 3D line matching, e.g., see [2].

2 Equal-Length Line Segments

The line matching algorithms we present in this paper rely on the repeated use of matching sets of equal-length line segments. The best match between two sets of 3D lines, where corresponding pairs have equal lengths, can be calculated in closed-form. Here, we briefly discuss the solution and refer to [11] for details. The method is based on least squares, which implies that all data have Gaussian noise distributions.

As defined in Section 1.2, consider two sets of (directed) line segments $A=\{A_n\}$ and $X=\{X_n\}$, where A_n and X_n are a pair of corresponding line segments with the same lengths. That is, $A_n=(\mathbf{a}_n,\hat{\mathbf{b}}_n,l_n)$ and $X_n=(\mathbf{x}_n,\hat{\mathbf{y}}_n,l_n)$. The transformation $\mathcal{T}=(\mathbf{t},\mathrm{R})$ that operates on X is given by:

$$\mathcal{T}\mathbf{x}_n=\mathbf{t}+\mathrm{R}\mathbf{x}_n,\quad \mathcal{T}\hat{\mathbf{y}}_n=\mathrm{R}\hat{\mathbf{y}}_n, \tag{1}$$

where t and R are the translation vector and rotation matrix. The distance measure, $M(A,\mathcal{T}X)$, between the two sets is defined as the sum of Euclidean distance squares of corresponding pairs:

$$M(A,\mathcal{T}X)=\sum_{n=1}^{N}[l_n\,\|\,\mathbf{a}_n-\mathbf{t}-\mathrm{R}\mathbf{x}_n\,\|^2+l_n^3(1-\hat{\mathbf{b}}_n^\top\mathrm{R}\hat{\mathbf{y}}_n)/6]. \tag{2}$$

Vectors are column matrices given in a Cartesian coordinate system, $\|\,\mathbf{a}_n\,\|$ is the length of vector $\mathbf{a}_n$, and superscirpt $\top$ denotes matrix transpose. The minimum distance measure, $M^\star=\min_{\mathcal{T}}M(A,\mathcal{T}X)$, is called the *mismatch measure*. The advantage of this "least-squares" distance measure is that the transformation minimizing it can be found in closed-form.

To find the rotation matrix, first we compute the 3×3 cross-covariance matrix:

$$\mathrm{S}=\sum_{n=1}^{N}[l_n(\mathbf{a}_n-\tilde{\mathbf{a}})(\mathbf{x}_n-\tilde{\mathbf{x}})^\top+l_n^3\hat{\mathbf{b}}_n\hat{\mathbf{y}}_n^\top/12], \tag{3}$$

where

$$\tilde{\mathbf{a}}=\frac{1}{W}\sum_{n=1}^{N}l_n\mathbf{a}_n,\ \ \tilde{\mathbf{x}}=\frac{1}{W}\sum_{n=1}^{N}l_n\mathbf{x}_n,\ \ W=\sum_{n=1}^{N}l_n. \tag{4}$$

$\tilde{\mathbf{a}}$ and $\tilde{\mathbf{x}}$ are the *centers* of sets A and X. The rotation matrix can be found either from the singular value decomposition of S, or by using the quaternion representation. Here, we use the quaternion representation, which has a closed-form algebraic solution. In the next step, we construct the 4×4 real, symmetric matrix

$$\mathrm{Q}=\begin{pmatrix} S_{11}+S_{22}+S_{33} & S_{23}-S_{32} & S_{31}-S_{13} & S_{12}-S_{21}\\ S_{23}-S_{32} & S_{11}-S_{22}-S_{33} & S_{12}+S_{21} & S_{31}+S_{13}\\ S_{31}-S_{13} & S_{12}+S_{21} & S_{22}-S_{33}-S_{11} & S_{23}+S_{32}\\ S_{12}-S_{21} & S_{31}+S_{13} & S_{23}+S_{32} & S_{33}-S_{11}-S_{22}\end{pmatrix}, \tag{5}$$

where S_{ij} are the elements of matrix S. The normalized eigenvector of Q with the largest eigenvalue is the quaternion $q=(q_0,q_1,q_2,q_3)^\top$ that represents the rotation. The eigenvalues are the solutions of a quartic equation for which a closed-form solution exists. The rotation matrix can be constructed from this quaternion according to:

$$\mathrm{R}=\begin{pmatrix} q_0^2+q_1^2-q_2^2-q_3^2 & 2(q_1q_2-q_0q_3) & 2(q_1q_3+q_0q_2)\\ 2(q_2q_1+q_0q_3) & q_0^2-q_1^2+q_2^2-q_3^2 & 2(q_2q_3-q_0q_1)\\ 2(q_3q_1-q_0q_2) & 2(q_3q_2+q_0q_1) & q_0^2-q_1^2-q_2^2+q_3^2\end{pmatrix}. \tag{6}$$

After R is calculated, then $\mathbf{t}$ can be obtained from

$$\mathbf{t}=\tilde{\mathbf{a}}-\mathrm{R}\tilde{\mathbf{x}}. \tag{7}$$

This is a generalization of the solution derived by Faugeras and Hebert [4] and Horn [8] for matching sets of 3D points. That solution may be obtained by ignoring the terms containing $\hat{\mathbf{b}}_n$ and $\hat{\mathbf{y}}_n$ and setting all $l_n=1$.

Uncertainties in lines may be treated with various levels of sophistication. The simplest treatment is to introduce the weights w_n that quantify the relative confidence we have in the line pair A_n and X_n. In this simple treatment, each term in the distance measure is multiplied by w_n, i.e., (2) is modified to $M(A,\mathcal{T}X)=\sum_n w_n[\ldots]$. This leads to the following modifications: In (4), $l_n\rightarrow w_nl_n$, (3) becomes $\mathrm{S}=\sum_n w_n[\ldots]$, while (5), (6), and (7) retain their forms.

The case of matching sets of equal-length line segments is a 6-variable optimization problem. The cases we discuss in the following sections deal with matching sets of unequal-length line segments, which are essentially $(N+6)$-variable optimization problems: six variables for the rotation and translation and N variables for the corresponding portions.

3 Case 1: Finite Model, Finite Image (FMFI)

Suppose we have two sets of line segments, model $A=\{A_n\}$ and image $X=\{X_n\}$, where the corresponding lines A_n and X_n have unequal lengths. As defined in Section 1.2, $A_n=(\mathbf{a}_n,\hat{\mathbf{b}}_n,L_n)$, and $X_n=(\mathbf{x}_n,\hat{\mathbf{y}}_n,l_n)$. This case arises when, due to occlusion or noise, only edge fragments are extracted, which then have to be matched to a model or another image.

When we fit the image to the model, we have to find, in addition to image rotation and translation, the portion of A_n that corresponds to X_n because they have different lengths. For the lines with $l_n\le L_n$, we denote the point on A_n that corresponds to $\mathbf{x}_n$ by $\mathbf{a}_n+s_n\hat{\mathbf{b}}_n$, while for the lines with $l_n>L_n$, we denote the point on X_n that corresponds to $\mathbf{a}_n$ by $\mathbf{x}_n+s_n\hat{\mathbf{y}}_n$. The set $\{s_n\}$ is the set of *shift parameters* that must be found simultaneously with the best rotation and translation. Treating corresponding line pairs according to their lengths is due to the fact that we have to find the portion of the longer line that matches the shorter one. The

distance measure $M(A, \mathcal{T}X)$ can be obtained through the generalization of (2). It can be written as

$$M(A, \mathcal{T}X) = \sum_{n=1}^{N} M_p(A_n, \mathcal{T}X_n), \tag{8}$$

where the pairwise distance measure between lines A_n and $\mathcal{T}X_n$ is given by:

$$M_p(A_n, \mathcal{T}X_n) = \begin{cases} l_n \, \| \mathbf{a}_n + s_n \hat{\mathbf{b}}_n - \mathbf{t} - \mathrm{R}\mathbf{x}_n \|^2 + l_n^3 (1 - \hat{\mathbf{b}}_n^\top \mathrm{R} \hat{\mathbf{y}}_n)/6, & \text{if } l_n \le L_n, \\ L_n \, \| \mathbf{a}_n - \mathbf{t} - \mathrm{R}(\mathbf{x}_n + s_n \hat{\mathbf{y}}_n) \|^2 + L_n^3 (1 - \hat{\mathbf{b}}_n^\top \mathrm{R} \hat{\mathbf{y}}_n)/6, & \text{if } l_n > L_n. \end{cases} \tag{9}$$

To minimize the distance measure with respect to $\mathbf{t}$, R, and the set $\{s_n\}$, we may solve the equations $\partial M / \partial s_n = 0$, which yield

$$s_n = \begin{cases} (\mathbf{t} + \mathrm{R}\mathbf{x}_n - \mathbf{a}_n)^\top \hat{\mathbf{b}}_n, & |s_n| \le (L_n - l_n)/2, \quad \text{if } l_n \le L_n, \\ (\mathbf{a}_n - \mathbf{t} - \mathrm{R}\mathbf{x}_n)^\top \mathrm{R} \hat{\mathbf{y}}_n, & |s_n| \le (l_n - L_n)/2, \quad \text{if } l_n > L_n. \end{cases} \tag{10}$$

The constraint in (10) means that if any s_n exceeds either limit, it will be set equal to that limit. For a given R and $\mathbf{t}$ the $\{s_n\}$ in (10) minimize, not maximize, M. This can be verified by noting that all the eigenvalues of the $N \times N$ Hessian matrix of M are positive. The Hessian is diagonal with elements $\partial^2 M / \partial s_n \partial s_m = 2\mathcal{L}\, \delta_{nm}$, where $\mathcal{L} = \sum_n \min(l_n, L_n) > 0$.

It can be shown that the minimization problem, unlike the equal-length case, cannot be reduced to a linear form in R, hence a closed-form solution cannot be found. However, we can solve the problem with an iterative, convergent optimization algorithm described below. Note that if the shift parameters, $\{s_n\}$, are known then the corresponding portions of line segments in set A are specified and this problem reduces to the simpler problem of matching sets of corresponding line segments with equal length that has a closed-form solution. For given values of the set $\{s_n\}$, the cross-covariance matrix may be expressed as

$$\mathrm{S} = \sum_{n=1}^{N} \mathrm{S}_p(A_n, X_n), \tag{11}$$

where the pairwise cross-covariance is given by:

$$\mathrm{S}_p(A_n, X_n) = \begin{cases} l_n (\mathbf{a}_n + s_n \hat{\mathbf{b}}_n - \tilde{\mathbf{a}})(\mathbf{x}_n - \tilde{\mathbf{x}})^\top + l_n^3 \, \hat{\mathbf{b}}_n \hat{\mathbf{y}}_n^\top / 12 & \text{if } \; l_n \le L_n, \\ L_n (\mathbf{a}_n - \tilde{\mathbf{a}})(\mathbf{x}_n + s_n \hat{\mathbf{y}}_n - \tilde{\mathbf{x}})^\top + L_n^3 \, \hat{\mathbf{b}}_n \hat{\mathbf{y}}_n^\top / 12 & \text{if } \; l_n > L_n. \end{cases} \tag{12}$$

The centers of the matching portions of sets A and X are

$$\tilde{\mathbf{a}} = \frac{1}{W} \sum_{n=1}^{N} \lambda_n \acute{\mathbf{a}}_n, \quad \tilde{\mathbf{x}} = \frac{1}{W} \sum_{n=1}^{N} \lambda_n \acute{\mathbf{x}}_n, \quad W = \sum_{n=1}^{N} \lambda_n, \tag{13}$$

where $\lambda_n = \min(l_n, L_n)$, and the centers of matching portions of line segments are given by:

$$\begin{cases} \acute{\mathbf{a}}_n = \mathbf{a}_n + s_n \hat{\mathbf{b}}_n, & \acute{\mathbf{x}}_n = \mathbf{x}_n, & \text{if } l_n \le L_n, \\ \acute{\mathbf{a}}_n = \mathbf{a}_n, & \acute{\mathbf{x}}_n = \mathbf{x}_n + s_n \hat{\mathbf{y}}_n, & \text{if } l_n > L_n. \end{cases} \tag{14}$$

Having computed S from (11), we then compute Q according to (5) and find its largest eigenmode, from which we may obtain the rotation R from (6) and the translation $\mathbf{t}$ from (7). Then, having $\mathbf{t}$ and R, the values of $\{s_n\}$ may be improved according to (10). Therefore, in Table 1, we propose the iterative algorithm for finding the best match between two sets of line segments with unequal length. For this case, the cross-covariance matrix is calculated from (11), and values of the set $\{s_n\}$ are updated according to (10). The algorithm converges when all s_n cease to vary significantly from the previous iteration.

The proof that the algorithm converges is straightforward as follows: For given values of $\{s_n\}$, the optimal rotation and translation, computed in Step 2, reduce the value of the distance measure (actually minimize the distance measure); and updating the values of $\{s_n\}$ for the given rotation and translation in Step 3 further reduce the value of the distance measure (actually minimize it). Since the distance measure is bounded from below, $M(A, X) \ge 0$, then the algorithm must converge after a number of iterations.

The above argument only proves that the FMFI algorithm converges to a minimum, not necessarily the global minimum. However, extensive empirical results show that this algorithm almost always converges to the global minimum, no matter how it is initialized. (See Section 3.1 for justification of this assertion.) It converges rapidly, though the number of iterations depends on the problem at hand and the desired accuracy. The computation time scales linearly with N and the number of iterations.

Limitation of FMFI. The FMFI algorithm could fail if, in a pair of corresponding lines, the shorter line segment is not fully contained in the longer line segment. This is unlikely to happen if set A is a CAD model and set X is an image. It may happen, however, if both sets are extracted from images. If this limitation is not satisfied, then the FMFI algorithm may not yield accurate matches. A satisfactory solution for matching partially overlapping line segments has not yet appeared in the literature. If we let the overlap length be an unknown variable to be determined, it will lead to the trivial solution of vanishing overlap length. Therefore, we need to put a heuristic lower bound on the overlap length, which would make the method ad hoc and unsatisfactory. Zhang [20] suggests, for 2D line matching, to look for the transformation that also maximizes the overlap. This heuristic cannot be justified either and could yield misleading results. A possible alternative is to replace line segments in one set with infinite lines and use the FMII algorithm given in Section 4. This approach will not yield erroneous results, even though some information is lost.

3.1 Experiments

We carried out extensive experiments with synthetic and real image data to evaluate the performance of the matching algorithm.

We generated hundreds of synthetic data sets with varying degrees of noise for selected values of $N = 5, 10, 20, 50, 100, 200$. To build a data set, we first generated the

TABLE 1
An Iterative Algorithm for Computing the Best Match

Step 1.	Initialize the set $\{s_n\}$.*
Step 2.	Compute R and $\mathbf{t}$ from (6) and (7).
Step 3.	Update values of $\{s_n\}$ from (10).
Step 4.	Iterate Steps 2 and 3 until all s_n converge.

For initialization, we recommend setting all $s_n = 0$. We have found empirically that this initialization always leads to the best match.

model line set $\{A_n\}$. We then constructed the image set $\{X_n\}$ by perturbing $\{A_n\}$ in the following manner. To get $\mathbf{x}_n$ and $\hat{\mathbf{y}}_n$, we randomly perturbed the components of $\mathbf{a}_n$ and $\hat{\mathbf{b}}_n$. The random perturbations were drawn from either Gaussian or uniform distributions. (Note that the algorithm is based on a least squares cost function, with the underlying assumption that the data have Gaussian noise distributions.) The noise introduced in the image set had the following characteristics. For Gaussian noise, the location noise variances were at most 30 percent of the line segment lengths, while the orientation noise variances were at most 45 degrees. For uniform noise, the location noise amplitudes were at most 30 percent of the line segment lengths, while the orientation noise amplitudes were at most 45 degrees. We estimated that the signal-to-noise ratios were better than four for these experiments.

For each N, we ran the FMFI algorithm on the data sets several thousand times with different random initializations. All runs converged to the best solution, with the exception of $N=10$, where a few (less than 0.1 percent) of the runs converged to a nonoptimal match. This indicates that the optimization is nonconvex, but with a large optimal basin. Without exception in all the experiments when we initialized all $s_n=0$, we obtained optimal matches. Below, we present empirical evidence for optimality of solutions. The number of iterations depends on the problem and the desired accuracy for convergence of shift parameters. With a stringent convergence criterion of 10^{-3} length-unit, the number of iterations ranged from three to 20. Fig. 1 shows typical convergence of FMFI for selected values of N.

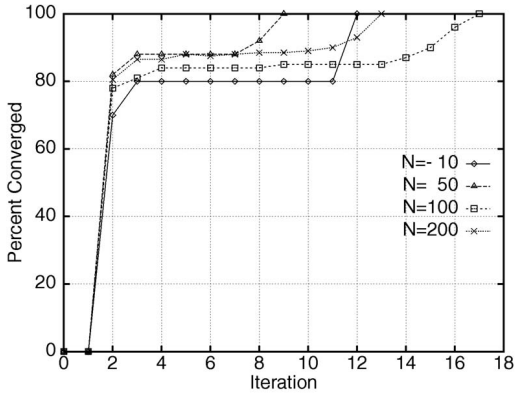


Fig. 1. Convergences of typical runs for the FMFI algorithm.

There is strong empirical evidence that the solutions found by the algorithm are optimal. To verify this, we used the tracking method applicable to certain nonlinear optimizations. One starts with a function with a known global minimum, then the function is gradually perturbed toward the desired function while tracking the minimum. For well-behaved functions, the perturbation changes the minimum only slightly. For the experiments with synthetic data described above, the application of this method is relatively straightforward. For a zero noise case (and correct pairwise correspondences), we know what the best match is. When we add a small amount of noise, we are able to verify that the solution found is the best match by both visual inspection and from small changes in the values of R, $\mathbf{t}$,

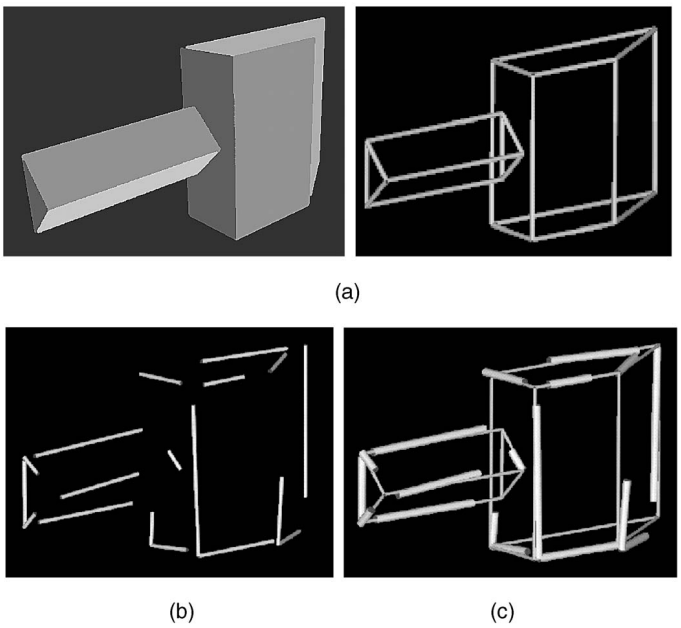


Fig. 2. (a) A polyhedral object and its edges. (b) A 3D edge image and (c) its best match to the model—image is shown by thick lines. Imperfect matches for some segments are due to image noise.

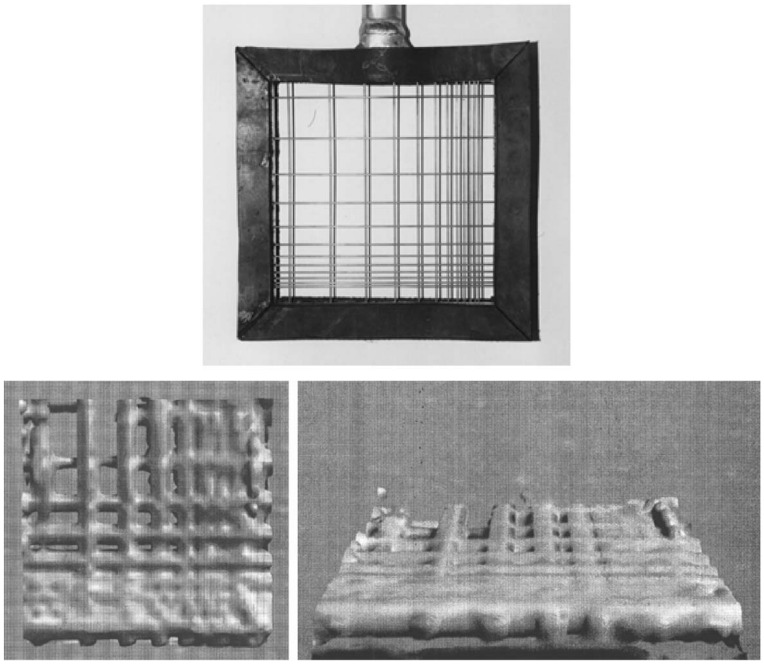


Fig. 3. Photograph of a steel grid (in air) and top and side views of its 3D acoustic image (in water).

and $\{s_n\}$. By gradually increasing the noise to higher levels, we are able to track the best match. We used this verification method on a subset of the experiments.

For the example shown in Fig. 2, there are $N=18$ edge fragments in the image. We used the FMFI algorithm with 1,000 different initial configurations. The algorithm converged to the same solution in all the trials, within 5 to 10 iterations with the convergence criterion of $\epsilon=10^{-3}$ length-unit. For $\epsilon=10^{-4}$, the number of iterations ranged between 8 and 14, and for $\epsilon=10^{-5}$ the algorithm converged in 11 to 18 iterations.

Figs. 3 and 4 show an example of 3D underwater acoustic image. Fig. 3 shows the photograph of a flat grid of thin steel bars, with nonuniform spacings, and top and side views of a volume-rendered acoustic image of the grid placed in a water tank. The underwater imaging sonar operates like a ladar [10]. It ensonifies the scene and collects the backscatter energy as a function of distance, thus yielding a 3D gray-level image.

The steel grid is modeled as 24 coplanar line segments (12 horizontal and 12 vertical) with known lengths. We are able to reliably extract 13 bars (six horizontal and seven vertical) from the 3D volume image. To extract the lines from the 3D image, we thresholded the image to obtain the line voxels. We then simultaneously fitted a number of lines to the voxels using the technique given in [14]. This gives us the line directions and locations, but not yet the line lengths. To obtain the length of each line, we determined the two end voxels fitting the same line and took their orthogonal projections on the line to be the endpoints of that line segment, thus obtaining an estimate for the line segment length. The line fitting technique which is based on squared residual distance, and the simple procedure we used to estimate line segment lengths work well here because the noise levels are low. For images with high noise levels, robust estimators should obviously be used. Also, a recent approach by Werman and Keren [19] for estimating the length of a line segment is likely to yield more accurate results for high noise cases. Estimating the length of 3D lines and curves is an interesting problem in itself, and we refer to Jonas and Kiryati [9] for discussion and references.

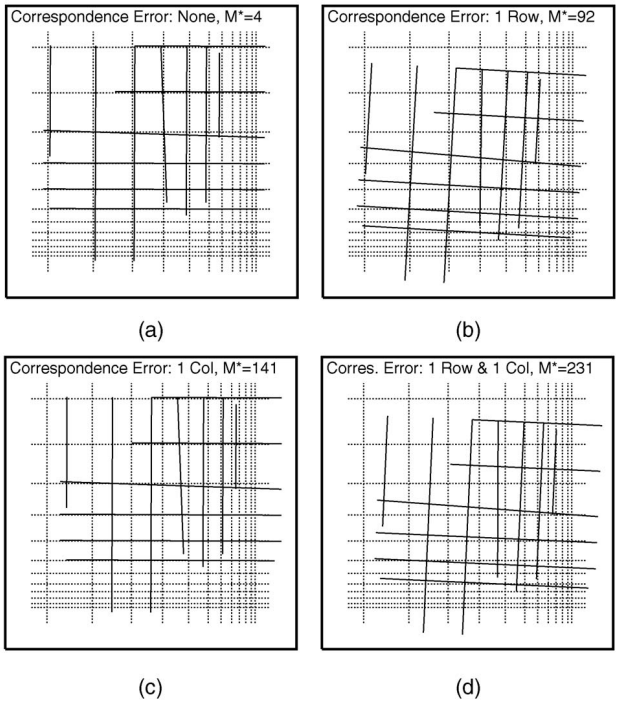


Fig. 4. Results of matching the grid image (solid lines) to the model (dotted lines) using FMFI. The best match with correct line correspondences is shown in (a). Also shown are the best matches with correspondences misidentified: by one row (b); by one column (c); and by one row and one column (d).

We used the FMFI algorithm with $N=13$. All runs with different initializations converged to the same solution shown in Fig. 4. Based on the empirical evidence from the synthetic data sets, we believe the solution found is the best match. The mismatch measure is $M^\star=4$ when the line correspondences are identified correctly. The unit of mismatch measure is cubic centimeter. Fig. 4 also shows the best match when corresponding lines are misidentified by just one row for which $M^\star=92$; or by one column which yields $M^\star=141$; or by one row and one column which results in $M^\star=231$. This example also demonstrates how the line matching algorithm may be used to verify hypothesized line correspondences. In each case, we ran the algorithm with several thousand different initializations. All runs converged to the same result. The number of iterations ranged from 5 to 20. Note that, even though the grid is a flat planar object, its acoustic image is 3D and the extracted line segments are slightly noncoplanar. In Fig. 4, we show only the 2D top view.

4 Case 2: Finite Model, Infinite Image (FMII)

In this case, line segments in the model are finite, while lines in the image are infinitely long. This case arises when exact information about the model exists, while the lengths in the image either cannot be inferred at all,[1] or cannot be estimated with the desired accuracy.[2] We represent the model line segment $A_n=(\mathbf{a}_n,\hat{\mathbf{b}}_n,L_n)$ by its midpoint $\mathbf{a}_n$, direction $\hat{\mathbf{b}}_n$, and length L_n. The image line segment $X_n=(\mathbf{x}_n,\hat{\mathbf{y}}_n)$ is represented by its direction $\hat{\mathbf{y}}_n$, and one of its points $\mathbf{x}_n$. Although $\mathbf{x}_n$ can be any point on the line, we choose it to be the closest point to the coordinate system origin, which implies $\mathbf{x}_n^\top\hat{\mathbf{y}}_n=0$.

Now, in addition to the optimum transformation we have to determine the portion of the L_n-long segment of the (infinitely long) image line X_n which corresponds to the model line segment A_n. We take $\mathbf{x}_n+s_n\hat{\mathbf{y}}_n\in X_n$ to be the point that corresponds to $\mathbf{a}_n$. The distance measure between the set $\{A_n\}$ and its corresponding segments of the set $\{X_n\}$ is thus

$$M(A,\mathcal{T}X)=\sum_{n=1}^{N}[L_n\,\|\,\mathbf{a}_n-\mathbf{t}-\mathrm{R}(\mathbf{x}_n+s_n\hat{\mathbf{y}}_n)\,\|^2 + L_n^3(1-\hat{\mathbf{b}}_n^\top\mathrm{R}\hat{\mathbf{y}}_n)/6]. \tag{15}$$

To determine the shift parameter s_n, we solve the equation $\partial M/\partial s_n=0$ which yields

$$s_n=(\mathbf{a}_n-\mathbf{t})^\top\,\mathrm{R}\hat{\mathbf{y}}_n, \tag{16}$$

where we used $(\mathrm{R}\mathbf{x}_n)^\top\mathrm{R}\hat{\mathbf{y}}_n=\mathbf{x}_n^\top\hat{\mathbf{y}}_n=0$ to simplify (16). The cross-covariance matrix, which is used to calculate the rotation matrix, for this case is given by

$$\mathrm{S}=\sum_{n=1}^{N}[L_n(\mathbf{a}_n-\tilde{\mathbf{a}})(\mathbf{x}_n+s_n\hat{\mathbf{y}}_n-\tilde{\mathbf{x}})^\top+L_n^3\,\hat{\mathbf{b}}_n\hat{\mathbf{y}}_n^\top/12], \tag{17}$$

with

$$\tilde{\mathbf{a}}=\frac{1}{W}\sum_{n=1}^{N}L_n\mathbf{a}_n,\quad \tilde{\mathbf{x}}=\frac{1}{W}\sum_{n=1}^{N}L_n(\mathbf{x}_n+s_n\hat{\mathbf{y}}_n),\quad W=\sum_{n=1}^{N}L_n. \tag{18}$$

The optimum transformation, as well as the shift parameters, can be found by the iterative optimization algorithm given in Table 1, with appropriate modifications. In this case, we compute S from (17) and update the set $\{s_n\}$ according to (16). As we showed in Section 3, the algorithm converges because in each step the value of the distance measure is reduced (actually is minimized), and because the distance measure is bounded from below.

4.1 Experiments

To evaluate the effectiveness of the FMII algorithm, we performed a number of experiments with synthetic and real image data. These are similar to the experiments described in Section 3.1 for the FMFI algorithm. We generated hundreds of synthetic data sets with varying degrees of noise for selected values of $N=5,\ 10,\ 20,\ 50,\ 100,\ 200$. For each N, we ran the FMII algorithm on the data sets several thousand times with different random initializations. All runs converged to the same solution. Using the tracking method described in Section 3.1, we verified that the solutions found were the best matches. With a convergence criterion of 10^{-3} length-unit, the number of iterations ranged between 5 and 73. The generally slower convergence rate, compared to FMFI, appears to be due of the absence of constraint on s_n, which would make finding the corresponding portion on an infinitely long line more difficult.

We also experimented with the acoustic image shown in Fig. 3. Here, we assumed that image line segments could not be determined with sufficient precision so that we had to represent the image lines with their locations and directions only. When we used the correct line correspondences the mismatch measure turned out to be $M^\star=7$. For incorrect line correspondences, we obtained the following results: $M^\star=93$ for misidentifying line correspondences by one row; $M^\star=78$ for misidentifying line correspondences by one column; and $M^\star=164$ for misidentifying line correspondences by one row and one column. These matches are shown in Fig. 5. In each case, again, we ran the algorithm with several thousand different initializations. All runs converged to the same result, with the number of iterations ranging from 5 to 22. Note that, in this example, both FMFI and FMII clearly indicate the correct hypothesis for line correspondences. The FMFI, however, does so even more strongly than FMII. This may be seen from $M^\star$ ratios for the wrong hypothesis over the correct hypothesis, e.g., the mismatch ratios of the case with one row and one column error in correspondences over the correct correspondences are: $231/4=58$ for FMFI, and $164/7=23$ for FMII. This result is expected and indicates that generally when the line segment length information are available, then line matching and hypothesis verification can be performed with greater certainty.

1. This happens when a line spans the entire image with no discernible end points. One might "hallucinate" [1] image boundaries as the end points, but that is generally not justified.

2. In noisy images where line segment lengths cannot be detected with sufficient confidence, this may be used as an alternative to FMFI. Of course, one may choose to use the uncertain length estimate, assign a relatively low weight to the line, and use the FMFI algorithm with the modified equations discussed in Section 2.

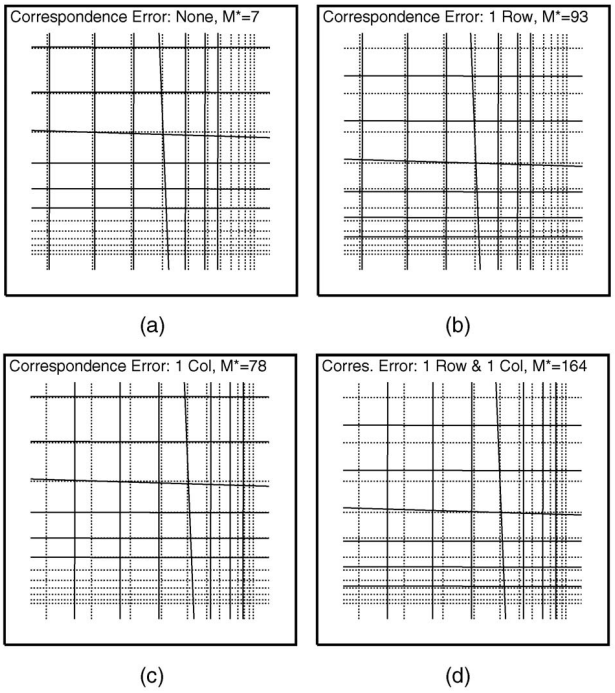


Fig. 5. Results of matching the grid image to the model using FMII. Descriptions are as given in Fig. 4. Note that here, the image lines are infinitely long.

5 Case 3: Infinite Model, Infinite Image (IMII)

In this case, both model and image lines are infinitely long. This case arises when information about line segment lengths are either unavailable or ignored. For example, when a line spans the entire image its true end point and its length cannot be discerned. Another example is when we want to register two 3D images where the detected line segments have only partial or no overlap due to occlusion, then we may choose to treat the lines as infinite in order to avoid ad hoc assumptions about overlapping lengths.

The solution for Case 3 is complicated by two factors. The first is that, for matching infinite lines (as for all objects with infinite extent), an intrinsically finite distance measure does not exist. Alternative distance measures can be defined. However, distance measures that combine both orientation and location errors are intuitively more appealing. The second factor is that a unique representation for infinite lines does not exist. Several approaches have been proposed in the literature. The first was by Faugeras and Hebert (FH) [4] in 1986. More recently, we presented a similar approach (CM), but with guaranteed convergence [12]. Another approach that gave an approximate solution was proposed in [18] and [3]. We discuss these methods in this section and show that all are noninvariant with respect to the translation of the coordinate system. And, then, present a solution that is invariant, as well as closed-form.

5.1 Convergent Method (CM)

We presented this method in [12]. We treat this case as matching a set of infinitely long line segments and a set of finite line segments where all have the same length l because all lines should make identical contributions (l is arbitrarily long). We represent the infinitely long model line $A_n=(\mathbf{a}_n,\hat{\mathbf{b}}_n)$ by its direction $\hat{\mathbf{b}}_n$, and by its closest point to the coordinate system origin $\mathbf{a}_n$, which implies $\mathbf{a}_n^\top\hat{\mathbf{b}}_n=0$. The image line segment is represented by $X_n=(\mathbf{x}_n,\hat{\mathbf{y}}_n,l)$, where l is its virtual length, $\hat{\mathbf{y}}_n$ is its direction, and $\mathbf{x}_n$ is its closest point to the coordinate system origin, i.e. $\mathbf{x}_n^\top\hat{\mathbf{y}}_n=0$. We take the corresponding point to $\mathbf{a}_n$ to be $\mathbf{x}_n+s_n\hat{\mathbf{y}}_n$ which is the midpoint of the virtual line segment of length l. Parameters s_n are unknown and have to be determined. The distance measure between the two sets is

$$M(A,\mathcal{T}X)=\sum_{n=1}^{N}[l\,\|\,\mathbf{a}_n-\mathbf{t}-\mathrm{R}(\mathbf{x}_n+s_n\hat{\mathbf{y}}_n)\,\|^2 + l^3(1-\hat{\mathbf{b}}_n^\top\mathrm{R}\hat{\mathbf{y}}_n)/6]. \tag{19}$$

Similar to (16), the solution of $\partial M/\partial s_n=0$ is

$$s_n=(\mathbf{a}_n-\mathbf{t})^\top\,\mathrm{R}\hat{\mathbf{y}}_n. \tag{20}$$

The significance of the virtual length l turns out to be in the relative importance we wish to attach to location errors (first term in M) *versus* orientation errors (second term in M). The cross-covariance matrix for this case is given by

$$\mathrm{S}=\sum_{n=1}^{N}[l(\mathbf{a}_n-\tilde{\mathbf{a}})(\mathbf{x}_n+s_n\hat{\mathbf{y}}_n-\tilde{\mathbf{x}})^\top+l^3\,\hat{\mathbf{b}}_n\hat{\mathbf{y}}_n^\top/12], \tag{21}$$

with

$$\tilde{\mathbf{a}}=\frac{1}{N}\sum_{n=1}^{N}\mathbf{a}_n,\quad \tilde{\mathbf{x}}=\frac{1}{N}\sum_{n=1}^{N}(\mathbf{x}_n+s_n\hat{\mathbf{y}}_n). \tag{22}$$

The distance measure (19) can be minimized by the algorithm given in Table 1, in a similar manner to FMFI and FMII. In this case, of course, we compute the cross-covariance matrix from (21), and update the set $\{s_n\}$ according to (20). As in the FMFI case, it can be proven that this algorithm also converges. Extensive experiments, similar to those described in Sectionss 3.1 and 4.1, all converged to the global minimum.

5.2 FH Formulation

In a comprehensive paper, Faugeras and Hebert [4] addressed the problem of matching 3D geometric features. One of the topics they discussed was the problem of matching two sets of infinite 3D lines. In this seminal work, they formulated the distance measure as the sum of location and orientation matches:

$$M(A,\mathcal{T}X)=K_2\cdot M_{location}+K_1\cdot M_{orientation},$$

where K_1 and K_2 are user-set positive coefficients. Examination of the FH distance measure reveals that it can be cast in terms of the distance measure we use in (19). The virtual length l in our method and the coefficients in the FH method are related by $l=(6K_1/K_2)^{1/2}$.

For minimizing the distance measure, a Kalman filter approach is proposed that cannot be guaranteed to converge. In the FH method, corresponding points (on pairs of corresponding lines) are not explicitly formulated in the distance measure. However, when we reformulate the FH cost function in terms of the set $\{s_n\}$, we find that the FH solution requires that the transformed point, $\mathbf{z}_n=\mathcal{T}(\mathbf{x}_n+s_n\hat{\mathbf{y}}_n)$, to be the closest point to the origin, i.e., $\mathbf{z}_n^\top\mathrm{R}\hat{\mathbf{y}}_n=0$, which yields the constraint

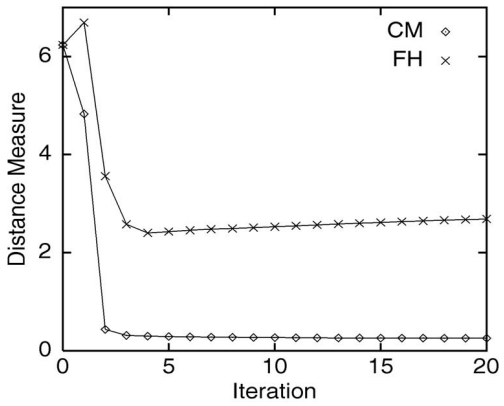


Fig. 6. An example of the convergence of the FH and CM algorithms.

$$s_n = -t^\top \mathrm{R}\hat{\mathbf{y}}_n. \tag{23}$$

Constraints (20) and (23) become identical when $\mathbf{a}_n^\top \mathrm{R}\hat{\mathbf{y}}_n = 0$. This is satisfied exactly for nonnoisy cases since $\mathrm{R}\hat{\mathbf{y}}_n = \hat{\mathbf{b}}_n$ when line correspondences are correct, and is satisfied approximately for moderately noisy data. For high noise data, as well as for cases with incorrect line correspondences, this condition is not satisfied well. Therefore, in general, the FH constraint (23) does not lead to a convergent solution. Fig. 6 shows a comparison of the behavior of the CM and FH algorithms for a selected example with $N = 20$. We will not present extensive comparisons of these two algorithms, since neither is invariant to the a coordinate system translation as we will show below.

5.3 Dual Quaternion

Daniilidis [3] and Walker et al. [18] propose an elegant and useful solution based on dual quaternion representation of lines. In this method, a parameter ϵ is introduced that has the unusual property $\epsilon > 0$, but $\epsilon^2 = 0$. Parameter ϵ has the dimension of inverse length and $1/\epsilon$ plays the role of l, the overlap length. Setting $\epsilon^2 = 0$ is equivalent to $l \to \infty$ in (19), which results in line orientations (l^3-term) completely dominating line locations (l-term). The algorithm yields an approximate solution to the full matching problem, where second order couplings between rotation and translation are ignored. Because of this decoupling, the solution is obtained in closed-form.

5.4 Best Matches Are Noninvariant

All the formulations described above have a drawback in common: They are not invariant to the choice of the coordinate system origin. This is illustrated in Fig. 7. In this 2D example, the model consists of two vertical lines and one horizontal line. The image is made by rotating 30 degrees the model's right vertical line. As can be seen, changing the coordinate system origin changes the best match of the image to the model. This figure shows the result of the best match for the CM algorithm. Other algorithms behave similarly.

The FH formulation is not invariant because it matches the distances of corresponding lines from the origin, which are not *intrinsic* properties of the line sets. Change of the origin changes the distances and, hence, the best match. The CM is not invariant because the selected line segments in

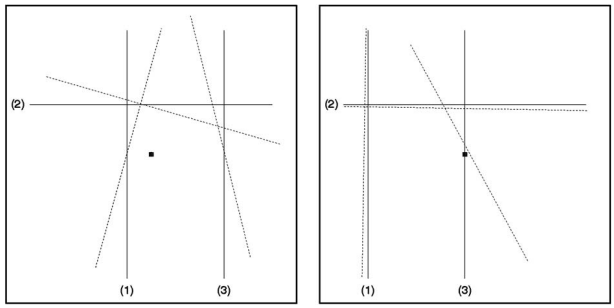


Fig. 7. An example of the change in the best match when the coordinate system origin is translated. Solid lines are the model, dotted lines the image. The origin is the black dot.

the image may be selected arbitrarily (we have chosen the midpoint of the selected piece to be the closest point to the origin). A change in the selected line segments (i.e., changing the origin) changes the best match. Note that when the model and image are identical, all algorithms become invariant to the choice of the origin. However, as the similarity of model and image degrades, the best match becomes more sensitive to the location of the origin. Fig. 8 illustrates this problem. Suppose A and X are corresponding lines. Also, suppose that the figure shows the best match of X to A when o_1 is the coordinate system origin. Now, suppose that the origin is moved to o_2. Because the location mismatch $\|\mathbf{x}_1 - \mathbf{a}\|$ increases to $\|\mathbf{x}_2 - \mathbf{a}\|$, these methods try to compensate for this increase by better alignment of the lines. This can be seen clearly in line 1 in Fig. 7, where lines farther from the origin are aligned better. Improved alignment reduces the location mismatch for A and X, but it will increase mismatches for other lines in the sets, with unpredictable impact on the total error. Similarly, the dual quaternion approach is not invariant either, because the "line moment" is defined with respect to the coordinate system origin. Lines that are located farther from the origin have greater influence on the orientation match. The discussion here was centered on square Euclidean distance. The noninvariance may become less noticeable if other distance measures, e.g., absolute value distance, are used, but does not disappear.

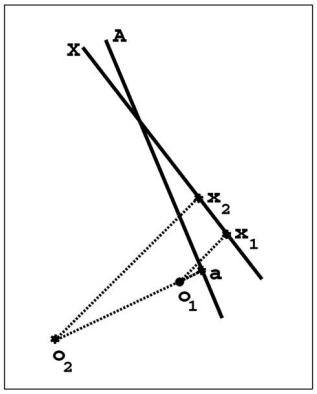


Fig. 8. Moving the coordinate system origin may change the best match.

5.5 Invariant Formulation

The underlying cause of noninvariance of the best match in all formulations is in the representation of lines. It is well-known that a unique, satisfactory representation for a single line does not exist. A single line may be represented by its direction vector and the coordinates of an arbitrary point on the line (often chosen as the closest point to the origin for a minimal number of variables, i.e., four instead of five), or the line may be represented in terms of two arbitrarily chosen points, e.g., the Plücker coordinates. All of these representations are relative to the origin of the coordinate system. Naturally, if the coordinate system is translated, then the line representation changes. In the matching methods described above also, line locations are given with respect to the coordinate system origin. This representation, or the best match, does not remain invariant under a coordinate change. However, having a set of lines rather than single lines offers a solution to the representation dilemma. This makes it possible to represent the lines in the set relative to a point which is fixed with respect to the set. In this way, line representations remain unchanged if the coordinate system undergoes transformations. Even though this point can, in principle, be arbitrary for a given set, the reference points $\mathbf{c}$ for set A and $\mathbf{z}$ for set X *must* be corresponding points.

The reference point that appears to be the best choice is the point that is overall closest to all the lines, that is, the point that minimizes the sum of the distances from all lines in the set:

$$\mathbf{c} = \arg\min_{\mathbf{p}} \sum_n dist(\mathbf{p}, A_n). \tag{24}$$

If, for the distance, we use the L_2 Euclidean norm, then finding $\mathbf{c}$ becomes a linear problem with the solution

$$\mathbf{c} = \mathrm{U}^{-1}\mathbf{v}, \tag{25}$$

where

$$\mathrm{U} = \sum_n (\mathrm{I} - \hat{\mathbf{b}}_n \hat{\mathbf{b}}_n^\top), \quad \mathbf{v} = \sum_n (\mathbf{a}_n - \mathbf{a}_n^\top \hat{\mathbf{b}}_n \hat{\mathbf{b}}_n). \tag{26}$$

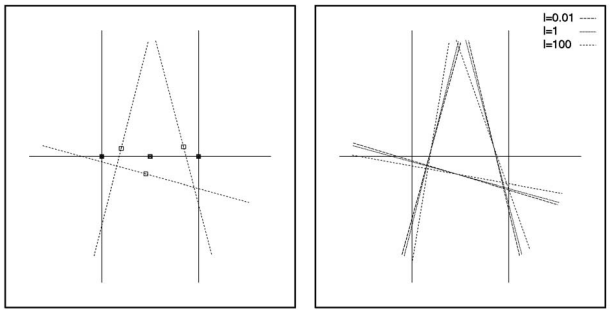
I is the unit matrix. A similar equation holds for $\mathbf{z}$ with the obvious replacement of $(\mathbf{a}_n, \hat{\mathbf{b}}_n)$ by $(\mathbf{x}_n, \hat{\mathbf{y}}_n)$.

The corresponding points on A_n and X_n are, thus, the projections of $\mathbf{c}$ on A_n and $\mathbf{z}$ on X_n, that is,

$$\begin{cases} \mathbf{a}'_n = \mathbf{a}_n + (\mathbf{c} - \mathbf{a}_n)^\top \hat{\mathbf{b}}_n \hat{\mathbf{b}}_n, \\ \mathbf{x}'_n = \mathbf{x}_n + (\mathbf{z} - \mathbf{x}_n)^\top \hat{\mathbf{y}}_n \hat{\mathbf{y}}_n. \end{cases} \tag{27}$$

Having specified the corresponding points of A_n and X_n, we use the equal-length line segment matching of Section 2 to find the best match in closed-form, by replacing $\{\mathbf{a}_n, \mathbf{x}_n, l_n\}$ with $\{\mathbf{a}'_n, \mathbf{x}'_n, l\}$ in (2) to compute the optimal transformation. This solution is obviously: 1) invariant to coordinate system transforms, 2) closed-form, and 3) identical to the previous solutions when line correspondences are correct and noise is absent.

When all the lines in the set are parallel, i.e., all $\hat{\mathbf{b}}_n = \hat{\mathbf{b}}$, an obvious ambiguity exists because $\mathbf{c}$ is not unique: $\mathbf{c}^\perp = \sum_n \mathbf{a}_n^\perp / N$, while $\mathbf{c}^\parallel$ is undetermined. (Superscripts $\perp$ and $\parallel$ denote the perpendicular and parallel components of $\hat{\mathbf{b}}$.) This is a trivially degenerate case and can be handled easily. We only need to match the orientations, since matching to any location of such a set would be optimum.



(a) (b)

Fig. 9. (a) The best match with the invariant algorithm. Corresponding points are indicated on the lines. (b) The effect of changing the length parameter l on the best match.

The question arises as to why we should represent lines in set A with respect to $\mathbf{c}$. As a justification for this choice, we argue that point $\mathbf{c}$ is likely the most stable point of A. Indeed, we can show this to be true under certain strong assumptions. Suppose instead of $\mathbf{c}$, we use the weighted closest point:

$$\mathbf{c}_w = \arg\min_{\mathbf{p}} \sum_n w_n \, dist(\mathbf{p}, A_n),$$

where $w_n \geq 0$ subject to the constraint $\sum w_n = N$. If all line parameters have small random perturbations with independent, identical distributions, then it follows that the average perturbation of $\mathbf{c}_w$ is

$$< |\delta \mathbf{c}_w|^2 > \propto \sum_n w_n^2.$$

The sum on the right hand side is minimized when all $w_n = 1$, which means that point $\mathbf{c}$ is (on the average) the most stable point under perturbations of set A.

Fig. 9 shows the best solution obtained with this technique for the example given in Fig. 7. The solution remains fairly stable even as the value of the virtual length, l, varies considerably (from 0.01 to 100) as seen in Fig. 9b. Fig. 10 shows another example of the best match obtained by this algorithm. We refer to this algorithm by IMII.

5.6 Experiments

The invariant matching algorithm (IMII) presented above in Section 5.5 is closed-form, hence its validation is straightforward and does not require extensive testing for convergence and optimality as the iterative FMFI and FMII algorithms. We have applied the IMII algorithm to several data sets and the results are quite acceptable to visual inspection. For example, see Figs. 9 and 10. Also, in Fig. 11, we show the results of matching the acoustic image to the grid model, for selected errors in line correspondences. Note that the results are similar to those of FMII (Fig. 5) in which the model lines all have the same length.

We note that the closed-form IMII algorithm may also be used to initialize the iterative FMFI and FMII algorithms. We have noticed that such initialization typically results in faster convergence. Moreover, the FMFI when initialized with IMII appears to always converge to the best match.

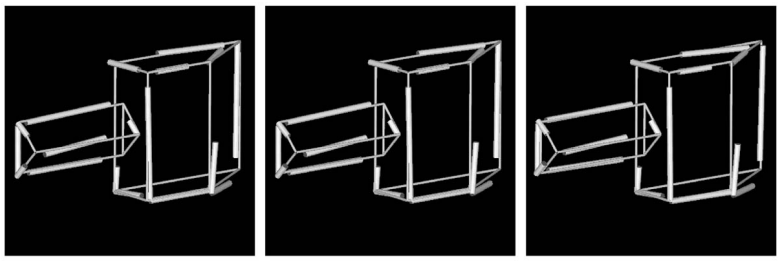


Fig. 10. Edge image of the polyhedral object (shown in Fig. 2) and its best match to the model with $l = 0.01, 1, 100$. Image is shown by thick lines.

6 Discussion

Correct treatment of matching two sets of corresponding 3D lines and line segments requires algorithms for matching three basic cases: 1) finite-finite, 2) finite-infinite, and 3) infinite-infinite. Algorithms for Case 1 proposed previously solve only a special situation. For Case 2, to our knowledge, no algorithm has appeared in the literature. In this paper, we have presented convergent algorithms that solve exactly Cases 1 and 2 for all situations. Existing algorithms for Case 3 are all noninvariant with respect to coordinate system translations. Furthermore, some existing algorithms yield approximate solutions, and some may not converge. We have presented an invariant, closed-form algorithm for solving Case 3. The matching algorithms presented in this paper may be used to tackle the challenging and critical problem of verifying hypothesized line correspondences reliably and rapidly.

The algorithm we presented for Case 1 is limited to situations where the shorter segment in the pairwise corresponding lines is fully contained in the longer segment. A completely satisfactory solution for matching partially overlapping line segments remains to be devised.

The distance measure, M, we used in this paper is based on least-squares error, which is based on the Gaussian noise assumption. Such cost functions are well-known to be sensitive to outliers and non-Gaussian noise. See, for example, [13], [15], [16]. However, even in such conditions, the least-squares solutions presented here are needed in many robust methods, e.g., the least median of squares.

In this paper, we assumed that lines had uniform *quality*, i.e., error in the data for all lines were the same. In practice, however, lines may have varying qualities and we may have estimates of their relative reliabilities. It is straightforward to modify the algorithms with overall weights representing line qualities (see Section 2). However, if we wish to incorporate separate error estimates for line locations, lengths, and orientations, then the algorithms require more elaborate modifications.

The approach we presented for matching 3D lines may be generalized to match two sets of 3D planar surface patches. This approach may also be useful for building algorithms for the more challenging and important problem of matching a set of 2D image lines to a 3D model, or to another set of 2D image lines.

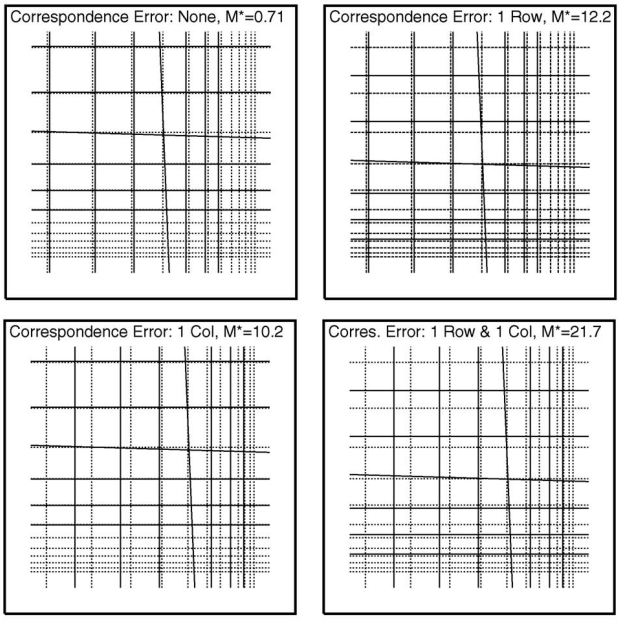


Fig. 11. Results of matching the grid image to the model using IMII. Descriptions are as given in Fig. 4. Matches shown are with $l=1$; results with $l=0.01$ to $l=100$ are visually indistinguishable.

Appendix

Mixed Cases

There are cases that, say, in the image, the lengths of certain line segments can be detected with the desired accuracy, while others cannot. We may represent the image set as a mixture of two subsets: subset 1 composed of line segments with finite lengths and subset 2 composed of lines with infinite lengths. While the model set consists of lines with finite lengths only. Also, suppose that the corresponding line pairs in subset 1 are labeled from 1 to N_1 and those in subset 2 are labeled from N_1+1 to N. The technique presented here can also handle these *mixed mode* cases. The distance measure is given by

$$M(A, \mathcal{T}X) = \alpha_1 M_1 + \alpha_2 M_2,$$

where M_1 and M_2 are the distance measures for subsets 1 and 2 given by (8) and (15), respectively, for FMFI and FMII. The weight factor values, α_i, may be set by the user from supplementary sources of information. The cross-covariance matrix for this case may also be expressed as

$$S = \alpha_1 S_1 + \alpha_2 S_2,$$

where S_1 and S_2 are the cross-covariances of subsets 1 and 2 given in (11) and (17), respectively. Obviously, the sums in (8) and (11) would run from 1 to N_1, and the sums in (15) and (17) would run from N_1+1 to N. However, the centers-of-mass, $\tilde{\mathbf{a}}$ and $\tilde{\mathbf{x}}$, used in computing S_1 and S_2 must be calculated from the expressions given below:

$$\tilde{\mathbf{a}} = \frac{\alpha_1}{W}\sum_{n=1}^{N_1}\lambda_n\acute{\mathbf{a}}_n + \frac{\alpha_2}{W}\sum_{n=N_1+1}^{N} L_n\mathbf{a}_n,$$

$$\tilde{\mathbf{x}} = \frac{\alpha_1}{W}\sum_{n=1}^{N_1}\lambda_n\acute{\mathbf{x}}_n + \frac{\alpha_2}{W}\sum_{n=N_1+1}^{N} L_n(\mathbf{x}_n + s_n\hat{\mathbf{y}}_n),$$

and

$$W = \alpha_1\sum_{n=1}^{N_1}\lambda_n + \alpha_2\sum_{n=N_1+1}^{N} L_n,$$

where $\lambda_n = \min(l_n, L_n)$, and $\acute{\mathbf{a}}_n$ and $\acute{\mathbf{x}}_n$ are given in (14).

The above distance measure for the mixed mode can also be minimized by the iterative, convergent algorithm given in Table 1, which is the mixture of FMFI and FMII. For this mixed mode case, we compute the cross-covariance matrix from the above expression. Then, for subset 1, we update shift parameters s_n according to (10) and, for subset 2, we update shift parameters s_n according to (16). Similar procedures can be developed for other mixed cases.

ACKNOWLEDGMENTS

This work was supported by the US Office of Naval Research. Parts of this paper appeared in CVPR-2001 [12].

REFERENCES

[1] A. Bartoli and P. Sturm, "The 3D Line Motion Matrix and Alignment of Line Reconstructions," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* vol. 1, pp. 287-292, Dec. 2001.

[2] H.H. Chen and T.S. Huang, "Matching 3-D Line Segments with Applications to Multiple-Object Motion Estimation," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 12, pp. 1002-1008, 1990.

[3] K. Daniilidis, "Hand-Eye Calibration Using Dual Quaternions," *Int'l J. Robotics Research,* vol. 18, no. 3, pp. 286-298, 1999.

[4] O.D. Faugeras and M. Hebert, "The Representation, Recognition, and Locating of 3-D Objects," *Int'l J. Robotics Research,* vol. 5, no. 3, pp. 27-52, 1986.

[5] W.E.L. Grimson, *Object Recognition by Computer: The Role of Geometric Constraints.* Cambridge, Mass.: MIT Press, 1990.

[6] C. Guerra and V. Pascucci, "On Matching Sets of 3D Segments," *Proc. Conf. Vision Geometry,* vol. 3811, pp. 157-167, July 1999.

[7] D.R. Heisterkamp and P. Bhattacharya, "Matching of 3D Polygonal Arcs," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 19, pp. 68-73, 1997.

[8] B.K.P. Horn, "Closed-Form Solution of Absolute Orientation Using Quaternions," *J. Optical Soc. Am. A,* vol. 4, pp. 629-642, 1987.

[9] A. Jonas and N. Kiryati, "Length Estimation in 3-D Using Cube Quantization," *J. Math. Imaging and Vision,* vol. 8, pp. 215-238, 1998.

[10] B. Kamgar-Parsi, B. Johnson, D.L. Folds, and E.O. Belcher, "High-Resolution Underwater Acoustic Imaging with Lens-Based Systems," *Int'l J. Imaging Systems Technology,* vol. 8, pp. 377-385, 1997.

[11] B. Kamgar-Parsi and B. Kamgar-Parsi, "Matching Sets of 3D Line Segments with Application to Polygonal Arc Matching," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 19, pp. 1090-1099, 1997.

[12] B. Kamgar-Parsi and B. Kamgar-Parsi, "An Open Problem in Matching Sets of 3D Lines," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* vol. 1, pp. 651-656, Dec. 2001.

[13] B. Kamgar-Parsi, B. Kamgar-Parsi, and N. Netanyahu, "A Nonparametric Method for Fitting a Straight Line to a Noisy Image," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 11, pp. 998-1001, 1989.

[14] B. Kamgar-Parsi, B. Kamgar-Parsi, and H. Wechsler, "Simultaneous Fitting of Several Planes to Point Sets Using Neural Networks," *Computer Vision, Graphics, and Image Processing,* vol. 52, pp. 341-359, 1990.

[15] P. Meer, D. Mintz, D. Kim, and A. Rosenfeld, "Robust Regression Methods in Computer Vision: A Review," *Int'l J. Computer Vision,* vol. 6, pp. 59-70, 1991.

[16] P. Rousseeuw and A. Leroy, *Robust Regression and Outlier Detection.* New York: John Wiley & Sons, 1987.

[17] C.J. Taylor and D.J. Kriegman, "Structure and Motion from Line Segments in Multiple Images," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 17, no. 11, pp. 1021-1032, Nov. 1995.

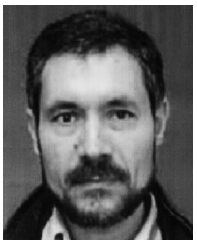
[18] M.W. Walker, L. Shao, and R.A. Volz, "Estimating 3-D Location Parameters Using Dual Number Quaternions," *CVGIP: Image Understanding,* vol. 54, pp. 358-367, 1991.

[19] M. Werman and D. Keren, "A Bayesian Method for Fitting Parametric and Non-Parametric Models to Noisy Data," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 23, no. 5, pp. 528-534, May 2001.

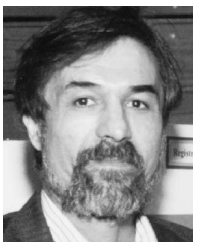
[20] Z. Zhang, "Estimating Motion and Structure from Correspondences of Line Segments between Two Perspective Images," *Proc. Int'l Conf. Computer Vision,* pp. 257-262, June 1995.

[21] Z. Zhang and O. Faugeras, *3D Dynamic Scene Analysis.* Springer-Verlag, 1992.

[22] Z. Zhang and O.D. Faugeras, "Determining Motion from 3D Line Segment Matches: A Comparative Study," *Image and Vision Computing,* vol. 9, no. 1, pp. 10-19, 1991.



Behzad Kamgar-Parsi received the PhD degree in theoretical physics from the University of Maryland at College Park in 1980. He is the program officer for the Intelligent and Autonomous Systems Program at the US Office of Naval Research (ONR). Before joining the ONR in 2000, he was a postdoctoral fellow at the National Institute of Standards and Tecchnology, 1981-1983; a research fellow at the Rockefeller University in New York City, 1984-1987; a research scientist at the University of Maryland Center for Automation Research, 1987-1990; and a research scientist at the Information Technology Division of Naval Research Laboratory, 1990-2003. He has performed research in computer vision, image processing, machine learning, and statistical mechanics, and has published more than 100 papers in these fields. He has also served on the program committees of many conferences on neural networks and automatic target recognition. He is a member of the IEEE and IEEE Computer Society.



Behrooz Kamgar-Parsi received the PhD degree in physics from the Catholic University of America, Washington, DC, in 1978. He is currently a senior research scientist the Center for Applied Research in Artificial Intelligence, Naval Research Laboratory, Washington, DC, where he has performed research in computer vision and artificial neural networks since 1989. From 1985 to 1988, he held the position of research scientist at the University of Maryland, Computer Vision Laboratory, College Park and, in 1988-1989, he was an assistant professor of computer science at George Mason University, in Fairfax, Virginia. He has also done research on numerical analysis while at the Center for Information Technology of the National Institutes of Health, in Bethesda, Maryland from 1981 to 1984. In addition, he achieved a significant breakthrough in medicine in 1981, while doing research on sleep disorders at the National Institutes of Health. His current research interests include face recognition, vision systems for low-flying small air vehicles, and 3D matching. He has authored many scientific publications and holds two US patents on target extraction and face recognition. He is a member of the IEEE, the IEEE Computer Society, and SPIE, and has served as program committee member and session chair for many conferences.