Analysis of Stochastic Automata Algorithm for Relaxation Labeling

P. S. Sastry and M. A. L. Thathachar

Abstract—A parallel stochastic algorithm for relaxation labeling is analyzed. For the case of symmetric compatibility functions, it is proved that the algorithm will always converge to a consistent labeling.

Index Terms—Relaxation labeling, learning automata, constraint satisfaction.

I. INTRODUCTION

In computer vision, many problems can be viewed as assigning labels or interpretations to some abstract objects or parts of an image in a consistent manner [2]. Relaxation labeling is a mechanism for obtaining such consistent labelings in an iterative fashion. Relaxation operates by updating, at each object, the probabilities of assigning various labels. The iterative updating of label probabilities makes use of contextual information in the form of label probabilities at neighboring objects and general domain knowledge in the form of compatibility functions.

In [1], we have presented an algorithm for relaxation labeling based on a network of learning automata. While some useful results regarding the convergence of the algorithm are presented in [1], the analysis is not complete. In this correspondence we analyze the algorithm for the case of symmetric compatibility functions and show that the algorithm, starting with any initial condition, always converges to a consistent labeling. We also present some general results regarding labeling problems with symmetric compatibility functions.

II. RELAXATION LABELLING

Consider a labeling problem with \( N \) objects and \( M \) labels. A labeling is a function from the set of objects, \( O \), to the set of labels, \( \Lambda \). The domain knowledge relevant to the problem is specified through a set of compatibility functions, \( r_{ij} : \Lambda \times \Lambda \rightarrow \mathbb{R}, i, j \in O \).

Throughout this correspondence, we use \( \mathbb{R}^n \) to denote \( n \)-dimensional real Euclidean space and \( \mathbb{R} \) to denote the real line. \( r_{ij}(\lambda, \lambda') \) can be thought of as the degree of compatibility (specified locally) between object-label pairs \((i, \Lambda)\) and \((j, \lambda')\).

In relaxation labeling we associate a probability vector, \( p_i = (p_{i1}, \ldots, p_{id}) \) with each object, \( i \) (All vectors considered in this correspondence are assumed to be row vectors). \( p_{iq} \) is the probability with which label \( q \) is associated with object \( i \). Define \( P = (p_1, \ldots, p_N) \). \( P \) will be called a label assignment. If each of the \( p_i, 1 \leq i \leq N \), are unit vectors then \( P \) is an unambiguous label assignment or a labeling, i.e., a unique assignment of label to each object. Define two subsets of \( \mathbb{R}^{MN} \) as

\[
K = \{ P \in \mathbb{R}^{MN} : P = (p_1, \ldots, p_N), \quad p_i \in \mathbb{R}^M \text{ is a probability vector, } 1 \leq i \leq N \}
\]

\[
K' = \{ P \in \mathbb{R}^{MN} : P = (p_1, \ldots, p_N), \quad p_i \in \mathbb{R}^M \text{ is a unit probability vector, } 1 \leq i \leq N \}. \quad (1)
\]

Manuscript received November 6, 1991; revised May 18, 1993. Recommended for acceptance by associate Editor N. Ahuja.

The authors are with the Department of Electrical Engineering, Indian Institute of Science, Bangalore 560 012, India.
It is easy to see that $K$ is a convex hull of $K^*$ [3]. We will call points in $K^*$ as corners of $K$. We have $P \in K$ for all label assignments $P$. Any unambiguous label assignment $P$ will be in $K^*$. We will now define, following [3], what label assignments are consistent. It may be noted that $K^*$ is defined to be the set of points in $K$ that are labeled by $P$. We will assume in the rest of the correspondence that $K^*$ is the unit vector in $R^M$ with $s_i$-th component unity. (This will correspond to a labeling represented by the point in $K^*$ given by $(1,0,0,0)$ and by taking a convex combination of points in $K^*$, we get points in $K$.)

Fig. 1. Picture to be labeled.

Complete the missing content and details of the text here.
Symmetric Compatibility Functions
As stated earlier in this correspondence, we consider the case of symmetric compatibility functions.

Definition 2.3: The compatibility functions, \( r_{ij} \), are said to be symmetric if
\[
\forall i, j \in \mathcal{J}, \ r_{ij}(X_i, X_j) = r_{ji}(X_j, X_i)
\]
Symmetry does not seem to be very restrictive because, in most applications, \( r_{ij} \) are, in fact, symmetric. The three sets of compatibility functions given in Table I are all symmetric. Define a functional for consistent unambiguous labelings if we consider a restriction of \( F \) to \( K' \). Let the labelings be consistent if and only if it is a local maximum of \( F \) over \( K' \) and \( K^* \). Then \( Q \) is said to be a neighbor of \( P \) if there exists \( i, j \in \mathcal{J} \) such that \( P_i = q_i, Q_j = q_j \neq i \) and \( p_i 
eq q_i \).

Lemma 2.2: Let the compatibility functions, \( r_{ij} \), be symmetric. Then \( F \) is a convex combination of some consistent points in \( K^* \) if and only if \( F \) is a convex combination of some consistent points in \( K^* \) and all these have the same \( F \)-value.

Proof: Using symmetry of \( r_{ij} \), the proof is immediate from (4) and Definitions 2.4 and 2.5 [5].

We now show that, for symmetric \( r_{ij} \), we need essentially consider only consistent unambiguous labelings.

Lemma 2.2: Let the \( r_{ij} \) functions be symmetric. Then the following are true of the labeling problem.

a) There exists a consistent unambiguous label assignment.

b) If there exists a consistent label assignment \( P \in K \), then there exists \( P^* \in K^* \) such that \( F(P) = F(P^*) = F_1(P^*) \) and \( P^* \) is consistent. Further, \( P \) is a convex combination of some consistent labelings from \( K^* \) all of which have the same \( F \)-value.

Proof:

a) Since \( K^* \) contains only finitely many points, there exists \( Q \) in \( K^* \) such that
\[
F_1(Q) = \max F_1(P)
\]
where the maximum is over all \( P \in K^* \). By Lemma 2.1, \( Q \) is consistent.

b) The main idea of the proof is as follows. Consider a two object two label problem. There are four points in \( K^* \) given by \((1, 0, 1), (1, 0, 1), (0, 1, 1), (1, 1, 0)\) and \((1, 0, 1) \) corresponding to the labelings \((\lambda_1, \lambda_2), (\lambda_1, \lambda_2), (\lambda_2, \lambda_1), (\lambda_2, \lambda_1)\) and \((\lambda_3, \lambda_2)\). Call these four points \( P_1, P_2, P_3, P_4 \) respectively. Consider any \( P = (p_1, p_2, p_3, p_4) \in K \). We can write \( P \) as
\[
P = p_1 P_1 + p_2 P_2 + p_3 P_3 + p_4 P_4.
\]
Similarly, a convex combination of points in \( K^* \) given by
\[
Q = a_1 P_1 + a_2 P_2 + a_3 P_3 + a_4 P_4
\]
is a point in \( K \) given by \((a_1 + a_2, a_3 + a_4, a_1 + a_2, a_3 + a_4)\).

In the proof, we first show that for a specific ordering of points in \( K^* \), each point in \( K \) can be expressed, as above, as a unique convex combination of points in \( K^* \) and that we get \( F(P) \) as the corresponding convex combination of \( F \)-values (same as \( F_1 \)-values).

We use this to show that every consistent point in \( K \) is a convex combination of some consistent points in \( K^* \) and all these have the same \( F \)-value.

Consider \( P = (p_1, \ldots, p_N) \in K \). From (6),
\[
F(P) = \sum_{i=1}^{N} \sum_{j=1}^{M} \sum_{q=1}^{M} r_{ij}(q, q) p_q p_{ij}
\]
where \( P = (e x_1, \ldots, e x_N) \). (Recall that \( e x_i \) is the unit vector in \( R^M \) with \( x_i \)-th component unity.) It is easy to see that \( F(P) = F(P) \) if \( P \in K^* \).

Definition 2.4: Let \( P = (p_1, \ldots, p_N) \) and \( Q = (q_1, \ldots, q_N) \) be any two points in \( K^* \). Then \( Q \) is said to be a neighbor of \( P \) if there exists \( i, j \in N \) such that \( p_i = q_i, Q_j \neq i \) and \( p_i \neq q_i \).

Q is a neighbor of \( P \) if the labeling given by \( Q \) differs from that given by \( P \) in exactly one object.

Definition 2.5: \( P \in K^* \) is said to be a local maximum of \( F_1 \) (defined by (7)) if \( F_1(P) \geq F_1(Q) \) for all \( Q \) such that \( Q \) is a neighbor of \( P \).

Lemma 2.1: Suppose \( r_{ij} \) are symmetric. Then \( P \in K^* \) is consistent if and only if it is a local maximum of \( F_1 \).

Proof: Using symmetry of \( r_{ij} \), the proof is immediate from (4) and Definitions 2.4 and 2.5 [5].

We now show that, for symmetric \( r_{ij} \), we need essentially consider only consistent unambiguous labelings.

Lemma 2.2: Let the \( r_{ij} \) functions be symmetric. Then the following are true of the labeling problem.

a) There exists a consistent unambiguous label assignment.

b) If there is a consistent label assignment \( P \in K^* \), there exists \( P^* \in K^* \) such that \( F(P) = F(P^*) = F_1(P^*) \) and \( P^* \) is consistent. Further, \( P \) is a convex combination of some consistent labelings from \( K^* \) all of which have the same \( F \)-value.

Proof:

a) Since \( K^* \) contains only finitely many points, there exists \( Q \) in \( K^* \) such that
\[
F_1(Q) = \max F_1(P)
\]
H(P) = H(P'). Without loss of generality, assume that p_{11} \neq p'_{11}. Since H(P) = H(P'), we have

\[ p_{11}p_{22} \ldots p_{NN} = p'_{11}p'_{22} \ldots p'_{NN} \quad (18) \]

for all s_2, \ldots, s_N. Summing both sides of the above equation over all s_i, we get p_{11} = p'_{11} which is a contradiction. Thus, H(\cdot) is one-to-one.

Now let P' belong to K - K* and let P' be consistent. Then we know [3], [5] that P' is a local maximum of F(\cdot). Suppose, without loss of generality, that in the representation for P' given by (17) only first L_i terms are nonzero. Let

\[ F_i(P') = \max_j F_i(P') \text{ for some } j, 1 \leq j \leq L_i \quad (19) \]

where the maximum is over all P', 1 \leq t \leq L_i. Suppose there exists an i, 1 \leq i \leq L_i such that

\[ F_i(P') > F_i(P') \quad (20) \]

where j is as given by (19). Now consider P'' \in K such that

\[ P'' = \sum_{i=1}^{L_i} b_{i}^{*} P' \quad (21) \]

where b_i^* = b_i^{*}v_n \neq i, j, b_i^* = b_i^{*} - \text{rand} j_i^* = j_i + \epsilon. By (19), (20) and (14), it is easy to see that, for all \epsilon > 0,

\[ F(P'') > F(P'). \quad (22) \]

Also with sufficiently small \epsilon, P'' given by (21) will be in any small neighborhood of P'. Hence, (22) implies that P' is not a local maximum of F and hence it cannot be consistent. But since P'' is to be consistent, we must have

\[ F_i(P'') = F_i(P'), \forall i, j \in \{1, \ldots, L_i\} \]

which in turn implies

\[ F(P') = F_i(P'), 1 \leq i \leq L_i \]

Now to complete the proof, we show that for P' to be consistent all P', 1 \leq i \leq L_i, should be consistent. Suppose there is a P', 1 \leq j \leq L_i which is not consistent. Then let P'' be a neighbor of P' such that F_i(P') < F_i(P''). (Such a P'' exists because P' is assumed inconsistent.) Now consider P'' \in K, given by

\[ P'' = \sum_{i=1}^{L_i} b_i P' + (b_i - \epsilon) P_i + (b_i + \epsilon) P'' \]

(If n does not belong to \{1, \ldots, L_i\} then b_i^* = 0).

Now it is easy to see that F(P'') > F(P'), \forall \epsilon > 0 and P'' is in any neighborhood of P' for sufficiently small \epsilon. Hence, P' cannot be consistent unless P', 1 \leq t \leq L_i are all consistent. This completes proof of the theorem.

**Remark 2.1:** From the proof given above, it is obvious that for a P' belonging to K - K* to be consistent we must have P' in the convex hull of \{P_1^1, \ldots, P_L^1\} \subset K* such that all point in that convex hull are consistent. (See last paragraph of Section 2.1). Except for such cases, the only consistent labelings are in K*. This results holds only when r_{ij} are symmetric; otherwise we do not have the correspondence between consistent labelings and local maxima of F and hence the above proof does not go through.

We now give a characterisation of consistent labelings which is needed for the proof of convergence of our algorithm. Define functionals g_i : K \rightarrow \mathbb{R}, 1 \leq i \leq N, by

\[ g_i(P) = \sum_r S_{ir}(P)p_{ir} \quad (23) \]

where the S_{ir}'s are given by (2).

**Lemma 2.3:** Consider \( P \in K, P \) is a consistent label assignment if and only if \( S_{ir}(P) = g_i(P), \forall i, r \).

**Proof:** This follows easily from a more general result in Game Theory (for example, see [6] and [7]). Hence we omit the proof.

In the analysis presented in next section, we assume that \( r_{ii} = 0, \forall i. \) It is easy to see that consistent labelings remain unchanged if we transform \( r_{ij} \) by \( r_{ij}' = ar_{ij} + b, a > 0 \) [1]. Hence, without loss of generality we assume \( r_{ij}(\lambda, \lambda') \in [0, 1], \forall i, j, \lambda, \lambda'. \)

**III. AUTOMATA ALGORITHM FOR RELAXATION LABELLING**

The algorithm that we analyze in this correspondence was originally proposed in [1]. It is based on the model of a team of interacting learning automata [8].

In the relaxation labeling framework, at each instant k of the iterative process, there is a label probability vector \( p_i(k) \) associated with each object \( i, 1 \leq i \leq N \). The process starts with some initial probabilities \( P(0) \), obtained through noisy measurements on the image. The algorithm specifies how the label probabilities are to be updated at each instant.

**The Algorithm**

1) **Initialization:** Obtain initial label probabilities \( P(0) = (p_1(0), \ldots, p_N(0)) \). Set \( k = 0 \).

2) **Selection of labels:** For each object \( i(1 \leq i \leq N) \) Choose a label at random based on the current label probabilities \( p_i(k) \).

3) **Calculation of responses:** For each object \( i(1 \leq i \leq N) \) do begin

Let \( q \) be the label selected for \( i \) in step 2. Then the response to \( i, \beta_q \), is computed as

\[ \beta_q = (1/N) \sum_j r_{ij}(q, s_j) \]

where \( s_j \) is the label selected for object \( j \) in step 2.

end.

4) **Updating of label probabilities:** For each object \( i(1 \leq i \leq N) \) do begin

Let \( q \) be the label selected for \( i \) in step 2 Then \( p_i(k) \) is updated as

\[ p_i(k + 1) = p_i(k) + \alpha \beta_q (1 - p_i(k)) \]

\[ p_i(k + 1) = p_i(k) - \alpha \beta_q p_i(k), r \neq q. \]

end (here \( 0 < \alpha < 1 \) is a parameter).

5) **Iteration:** Set \( k = k + 1 \) and go back to step 2 if probability vectors have not converged.

**Remark 3.1:** The algorithm can be viewed as a network of simple stochastic processing elements, namely the automata [9]. The internal state of each automaton is the label probability vector and its output is a random realization of this probability distribution. Each automaton is supplied with the outputs of other automata using which it updates its internal state. There are two special features of the algorithm compared to other relaxation labeling algorithms. Firstly, in our algorithm, for updating the label probabilities at an object, we do not need the current label probabilities at other objects; only the label chosen by other objects is needed. Secondly, as discussed in [1], this algorithm is effective even when the compatibility functions are not explicitly available, if we have available to us random variables \( X_{ij}(q, s) \) such that \( EX_{ij}(q, s) = r_{ij}(q, s) \). In such a case we use \( X_{ij} \) in place of \( r_{ij} \) for calculating \( \beta_q \) in step 3 of the algorithm. The analysis we present below will hold in that case also [1].
Analysis of the Algorithm

Consider the process \( \{P(k), k \geq 0\} \). The automata algorithm iteratively updates \( P(k) \) and hence we are interested in the asymptotic behavior of \( P(k) \). Define a continuous-time interpolation of \( P(k) \), \( \bar{P}(t) \), by

\[
\bar{P}(t) = P(k) \quad \text{for} \quad t \in [k_a, (k + 1)a)
\]

where "a" is the parameter used in (24). Define functions \( f_{iq} \), \( 1 \leq q \leq M, \ 1 \leq i \leq N \), on \( K \), by

\[
f_{iq}(P) = (1/a)E[p_{iq}(k + 1) - p_{iq}(k) \mid P(k) = P]
\]

(25)

Consider the family of Processes \( \{ \bar{P} \} \), indexed by "a." Using weak convergence techniques [10], it can be shown (see [1] for details) that \( \bar{P} \) converges weakly, as "a" tends to zero, to the solution of the Ordinary Differential Equation (ODE),

\[
\dot{X} = f(X), X(0) = P(0)
\]

(26)

where the components of \( f \) are \( f_{iq} \) defined by (25).

Thus for sufficiently small value of the parameter 'a' used in equation (24), the asymptotic behavior of \( P(k) \) can be obtained from the asymptotic behavior of the solutions of ODE (26) with appropriate initial condition [1].

In what follows we completely characterize the solutions to ODE (26). In [1] we showed that every consistent unambiguous labeling is locally asymptotically stable. But the analysis in [1] is incomplete because we did not characterize all stationary points of the ODE. Also, we did not prove that starting from some arbitrary point in \( K \) the algorithm will always converge to some point in \( K \) rather than, e.g., exhibit a limit cycle behavior. (In a dynamical system limit cycle behavior means the system keeps tracing a closed trajectory in the phase space).

Theorem 3.1: Consider a labeling problem with symmetric compatibility functions. Then the automata algorithm starting with any \( P(0) \in K - K^* \) and with sufficiently small value of 'a' will always converge to a consistent labeling.

Proof: From (24) and (25), we get

\[
f_{iq}(P) = p_{iq} \sum_j p_{jr} \left[ \sum_{s} r_{js}(q,s) p_{ks} - \sum_{s} r_{js}(r,s) p_{ks} \right]
\]

(27)

Consider a function \( G : R^{MN} \rightarrow R \) defined by

\[
G(P) = \sum_{i, q} \sum_j r_{ij}(q,s) p_{iq} p_{jr}
\]

where \( P = (p_1, \cdots, p_N) \in R^{MN} \) and \( p_i \in R^M \) for all \( i \).

The function \( G \) restricted to \( K \) is the function \( F \) considered in Section II. \( G \) is continuous and differentiable over \( R^{MN} \). Since \( r_{ij} \) are symmetric, we get

\[
\frac{\partial G}{\partial p_{iq}} = 2 \sum_{j,s} r_{js}(q,s)p_{js}
\]

(28)

Along the solution paths of ODE (26), \( G \) is nondecreasing because

\[
\frac{dg(t)}{dt} = \sum_{i,q} \frac{\partial G}{\partial p_{iq}} \frac{dp_{iq}}{dt} = \sum_{i,q} \frac{\partial G}{\partial p_{iq}} \sum_j p_{iq} p_{jr} \left[ \frac{1}{2} \frac{\partial G}{\partial p_{iq}} - \frac{\partial G}{\partial p_{ir}} \right]
\]

using (26), (27), and (28)

\[
= \frac{1}{2} \sum_i \sum_q \sum_{q^>'} p_{iq} p_{ir} \left[ \frac{\partial G}{\partial p_{iq}} - \frac{\partial G}{\partial p_{ir}} \right]^2 \geq 0.
\]

(29)

Under the automata algorithm, the updating of the label probabilities is such that \( P(k) \) belongs to \( K \) for all \( k \) if \( P(0) \) belongs to \( K \). \( K \) is a compact subset of \( R^{MN} \). Hence, by (29), asymptotically all solutions of (26) will be in the set (see [11, Theorem 2.7])

\[
\{ P \in K : \frac{dg(t)}{dt}(P) = 0 \}
\]

From (29), \( \frac{dg}{dt}(P) = 0 \) if and only if, for all \( i, q, r \), we have either \( p_{ir} = 0 \) or

\[
\frac{\partial G}{\partial p_{iq}} = \frac{\partial G}{\partial p_{ir}}
\]

that is,

\[
\sum_j r_{js}(q,s)p_{js} = \sum_j r_{js}(r,s)p_{js}
\]

This implies, from (27), that such a \( P, f_{iq}(P) = 0, \forall i, q \).

Thus, the solutions to ODE (26), for any initial condition in \( K \), will converge to a set containing only stationary points of the ODE and hence the solutions do not exhibit any limit-cycle like behavior.

Now the proof of the theorem will be complete if we can show that any stationary point of the ODE in \( K - K^* \), which is not a consistent label assignment, is unstable. (In [1] it is shown that all consistent labelings in \( K^* \) are locally asymptotically stable and all other points in \( K^* \) are unstable).

Using (2) and (23), (27) can be rewritten as

\[
f_{iq}(P) = p_{iq} \left[ S_{ij}(P) - g_i(P) \right].
\]

(31)

Let \( P^* \) be a stationary point of the ODE that is not consistent. Then by lemma 2.3, there exist \( i, s \) such that

\[
S_{is}(P^*) > g_i(P^*)
\]

(32)

By continuity of the functions involved, (32) should be satisfied at every point in a small open neighborhood around \( P^* \). This implies, by (31); that in a small neighborhood around \( P^* \), \( \bar{p}_i > 0 \) and hence \( P^* \) is unstable for the ODE (26). This completes proof of the theorem.

Remark 3.2: In the above theorem, we needed the initial condition \( P(0) \) to be outside \( K^* \) because all points in \( K^* \) are absorbing for the algorithm (see (24)). By the above theorem we know that the algorithm always converges to some point in \( K \) that is consistent. Since all points in \( K^* \) are absorbing for the algorithm, it is reasonable to conclude that, starting from any initial condition, the algorithm always converges to a consistent unambiguous labeling. This is supported by Lemma 2.2 which says, for symmetric \( r_{ij} \), that there always exists a consistent unambiguous labeling and unless there are uncountably infinite consistent label assignments, we cannot have consistent label assignments in \( K - K^* \). In the simulations the
algorithm always converges to consistent unambiguous labelings [1], [5].

IV. CONCLUSION

In this correspondence, we analyzed the automata algorithm for relaxation labeling [1] for the case of symmetric compatibility functions. It is proved that starting with any initial label probabilities, the algorithm always converges to a consistent labeling. Further, all consistent unambiguous labelings are locally asymptotically stable. The algorithm analyzed in this correspondence has been employed successfully in computer vision problems such as stereopsis and object recognition [5].

REFERENCES


Analysis of 3-D Rotation Fitting

Kenichi Kanatani

Abstract—Computational techniques for fitting a 3-D rotation to 3-D data are recapitulated in a refined form as minimization over proper rotations, extending three existing methods—the method of singular value decomposition, the method of polar decomposition, and the method of quaternion representation. Then, we describe the problem of 3-D motion estimation in this new light. Finally, we define the covariance matrix of a rotation and analyze the statistical behavior of errors in 3-D rotation fitting.

Index Terms—3-D rotation, singular value decomposition, polar decomposition, quaternion representation, essential matrix, covariance matrix.

I. INTRODUCTION

In robotics applications, we often encounter the problem of computing a 3-D rigid motion that maps a set of 3-D points to another set. This problem typically occurs when 3-D data are obtained by stereo, range sensing, tactile sensing, etc. If we compute the centroid of each set and translate them in space so that their centroids come to the origin, the remaining problem is to determine the rotation that maps the first set of orientations to the second set. Thus, all we need to do is fit a 3-D rotation to the rotated data, say by least squares.

The first analytical technique for 3-D rotation fitting was reported by Horn [2], who used the quaternion representation. Equivalent techniques were presented by Arun et al. [1], using singular value decomposition, and by Horn et al. [3], using polar decomposition. However, their techniques dealt with minimization over orthogonal matrices. As a result, improper rotations (i.e., rotations of determinant –1) can be predicted for noisy data. Umezawa [13] made a correction to the method of Arun et al. [1], but his derivation, based on a variational principle and Lagrange multipliers, is very complicated and lengthy.

In this paper, we first recapitulate these techniques in a refined manner as minimization over proper rotations. Then we formulate the problem of optimal resolution of a degenerate rotation and show how this solves the problem of 3-D motion estimation from two images succinctly. Finally, we define the covariance matrix of rotation fitting and analyze the statistical behavior of errors.

II. OPTIMAL ESTIMATION OF 3-D ROTATION

Consider the problem of computing a 3-D rigid motion that maps a set of 3-D points \( \{(x_\alpha, y_\alpha, z_\alpha)\}, \alpha = 1, \ldots, N \), to another set \( \{(x'_\beta, y'_\beta, z'_\beta)\}, \beta = 1, \ldots, N \). If we compute the centroids \((\bar{x}, \bar{y}, \bar{z})\) and \((\bar{x}', \bar{y}', \bar{z}')\) of the two sets and translate them in space so that their centroids come to the coordinate origin \(O\), the remaining problem is to determine the 3-D rotation that maps the first set of orientations to the second set [2]. Consider the following problem:

Manuscript received October 21, 1991; revised December 7, 1992. Recommended for acceptance by Associate Editor R. Bolle.

The author is with the Department of Computer Science Gunma University, Gunma 376, Japan.

IEEE Log Number 9212236.

0162-8828/94$4.00 © 1994 IEEE