[14] B. Kamgar-Parsi, B. Kamgar-Parsi, and N. S. Netanyahu, "A nonparametric method for fitting a straight line to a noisy image," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 11, no. 9, pp. 998–1001, 1989.

[15] T. Kasvand, "Extraction of edges in 3-D range images to subpixel," in *Proc. 9th Int. Conf. on Pattern Recognit.*, 1988, pp. 93–98.

[16] P. Meer, D. Mintz, and A. Rosenfeld, "Robust regression methods for computer vision: A review," *Int. J. Comput. Vision*, vol. 6, no. 1, pp. 59–70, Apr. 1991.

[17] G. Roth and M. D. Levine. "Segmentation of geometric signals using robust fitting," in *Proc. 10th Int. Conf. on Pattern Recognit.*, vol. 1, pp. 826–831, 1990.

[18] G. Roth and M. D. Levine, "A genetic algorithm for primitive extraction," in *Proc. 4th Int. Conf. on Genetic Algorithms*, 1991, pp. 487–494.

[19] P. J. Rousseeuw and A. M. Leroy, *Robust Regression and Outlier Detection.* New York: John Wiley, 1987.

[20] B. Sabata, F. Arman, and J. K. Aggarwal, "Segmentation of 3-D range images using pyramidal data structures," in *Proc. 3rd Int. Conf. on Comput. Vision*, 1990, pp. 662–666.

[21] P. T. Sander and S. W. Zucker, "Tracing surfaces for surfacing traces," in *Proc. 1st Int. Conf. on Comput. Vision*, 1987, pp. 241–249.

[22] G. Syswerda, "Uniform crossover in genetic algorithms," in *Proc. 3rd Int. Conf. on Genetic Algorithms*, 1989, pp. 2–9.

[23] G. Taubin, "Estimation of planar curves, surfaces, and nonplanar space curves defined by implicit equations with applications to edge and range image segmentation," *IEEE Pattern Anal. Machine Intell.*, vol. 13, no. 11, pp. 1115–1138, Nov. 1991.

[24] L. Xu, E. Oja, and P. Kultanen, "A new curve detection method: Randomized Hough transform (RHT)," *IEEE Pattern Recognit. Lett.*, vol. 11, no. 5, pp. 331–338, May 1990.

[25] N. Yokoya and M. D. Levine, "Range image segmentation based on differential geometry: A hybrid approach," *Pattern Anal. Machine Intell.*, vol. 11, no. 6, p. 643, June 1989.

[26] X. M. Yu, T. D. Bui, and A. Krzyżak, "Invariants and pose determination," in *VISUAL FORM: Analysis and Recognition, Proc. of the Int. Workshop on Visual Form.* New York: Plenum, 1992, pp. 623–632.

[27] ——, "The genetic algorithm parameter settings for robust estimation and range image seg. and fitting," in *Proc. 8th Scandinavian Conf. on Image Anal.*, 1993, pp. 623–630.

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, $\mathbf{O}$, to the set of labels, Λ. The domain knowledge relevant to the problem is specified through a set of compatibility functions, $r_{ij} : \Lambda \times \Lambda \rightarrow \mathbf{R}, i, j \in \mathbf{O}$. (Throughout this correspondence, we use $\mathbf{R}^n$ to denote n-dimensional real Euclidean space and $\mathbf{R}$ 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, λ) and (j, λ').

In relaxation labeling we associate a probability vector, $\mathbf{p}_i = (p_{i1}, \cdots, p_{iM})$ 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 $P = (\mathbf{p}_1, \cdots, \mathbf{p}_N)$. P will be called a label assignment. If each of the $\mathbf{p}_i, 1 \leq i \leq N$, 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 $\mathbf{R}^{MN}$ as

$$\begin{aligned}\mathbf{K} = \{P \in \mathbf{R}^{MN} &: P = (\mathbf{p}_1, \cdots, \mathbf{p}_N),\\ &\mathbf{p}_i \in \mathbf{R}^M \text{ is a probability vector}, 1 \leq i \leq N\}\\ \mathbf{K}^* = \{P \in \mathbf{R}^{MN} &: P = (\mathbf{p}_1, \cdots, \mathbf{p}_N),\\ &\mathbf{p}_i \in \mathbf{R}^M \text{ is a unit probability vector}, 1 \leq i \leq N\}. \quad (1)\end{aligned}$$

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.

IEEE Log Number 9214456.

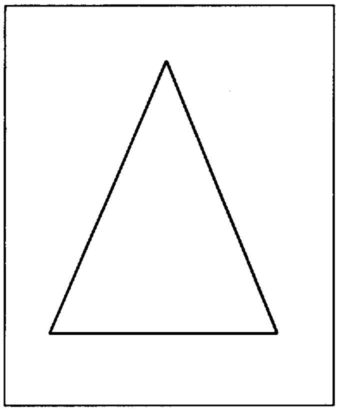

Fig. 1. Picture to be labeled.

It is easy to see that $\mathbf{K}$ is a convex hull of $\mathbf{K}^*$ [3]. We will call points in $\mathbf{K}^*$ as corners of $\mathbf{K}$. We have $P \in \mathbf{K}$ for all label assignments P. Any unambiguous label assignment P will be in $\mathbf{K}^*$. We will now define, following [3], what label assignments are consistent.

Definition 2.1: Let $P \in \mathbf{K}$. The support for label q at object i under P is defined to be

$$S_{iq}(P) = \sum_{j,s} r_{ij}(q,s)p_{js} \qquad (2)$$

It may be noted that $S_{iq}(P)$ does not depend on $\mathbf{p}_i$, if $r_{ii}(.,.) = 0$. We will assume in the rest of the correspondence that $r_{ii}(\cdot,\cdot) = 0$.

Definition 2.2: A label assignment $P = (\mathbf{p}_1, \cdots, \mathbf{p}_N) \in \mathbf{K}$ is called consistent, if for every i, $1 \le i \le N$,

$$\sum_q S_{iq}(P)\, p_{iq} \ge \sum_q S_{iq}(P)\, v_{iq} \qquad (3)$$

for all $V = (\mathbf{v}_1, \cdots, \mathbf{v}_N) \in \mathbf{K}$.

Let $P = (\mathbf{e}_{s_1}, \cdots, \mathbf{e}_{s_N}) \in \mathbf{K}^*$, where $\mathbf{e}_{s_j}$ is the unit vector in $\mathbf{R}^{\mathbf{M}}$ with s_j-th component unity. (This will correspond to a labeling where object i is assigned label s_i). Then P is consistent (i.e., P satisfies (3)) if and only if

$$\sum_j r_{ij}(s_i,s_j) \ge \sum_j r_{ij}(s,s_j), \ \forall s \ne s_i, i = 1, \cdots, N \qquad (4)$$

P is said to be strictly consistent if the inequalities in (3) or (4) are strict. We will sometimes write $(s_1, \cdots, s_N)$ for the unambiguous labeling $(\mathbf{e}_{s_1}, \cdots, \mathbf{e}_{s_N})$.

An Example

We will illustrate the concepts introduced so far through a very simple example. Fig. 1 shows an image consisting of a triangle. (This example is from [4] though we use different r_{ij} functions). The problem is to interpret the image by appropriately labeling the three line segments of the triangle. The possible labels for each of the three objects (line segments) are:

λ_1: occluding edge, forward object above.
λ_2: occluding edge, forward object below.
λ_3: convex fold.
λ_4: concave fold.

An unambiguous labeling (λ_2, λ_2, λ_4) (in which two of the lines are labeled λ_2 and the third as λ_4) corresponds to an interpretation that the object on view is a triangular flap hinged on one side and turned away from the viewer. (λ_1, λ_1, λ_1) corresponds to a triangular cutout floating above the background. It is easy to see that (λ_2, λ_4, λ_4) is a nonsensical interpretation. The labeling (λ_1, λ_1, λ_1) is represented by the point in $\mathbf{K}^*$ given by (1,0,0,0, 1,0,0,0, 1,0,0,0) and by taking a convex combination of points in $\mathbf{K}^*$, we get points in $\mathbf{K}$.

TABLE I-A
Different Compatibility Functions for Tthe Triangle Problem

	λ_1	λ_2	λ_3	λ_4
λ_1	1	0	0.5	0
λ_2	0	1	0	0.5
λ_3	0.5	0	0	0
λ_4	0	0.5	0	0

Consistent Labelings: (λ_1, λ_1, λ_1), (λ_2, λ_2, λ_2).

TABLE I-B
Different Compatibility Functions for Tthe Triangle Problem

	λ_1	λ_2	λ_3	λ_4
λ_1	1	0	1	0
λ_2	0	1	0	1
λ_3	1	0	0	0
λ_4	0	1	0	0

Consistent Labelings: (λ_1, λ_1, λ_1), (λ_1, λ_1, λ_3), (λ_2, λ_2, λ_4) (λ_2, λ_2, λ_2), (λ_1, λ_3, λ_1), (λ_2, λ_4, λ_2), (λ_3, λ_1, λ_1), (λ_4, λ_2).

TABLE I-C
Different Compatibility Functions for Tthe Triangle Problem

	λ_1	λ_2	λ_3	λ_4
λ_1	0.5	0	1	0
λ_2	0	0.5	0	1
λ_3	1	0	0	0
λ_4	0	1	0	0

Consistent Labelings: (λ_1, λ_1, λ_3), (λ_2, λ_2, λ_4), (λ_1, λ_3, λ_1) (λ_2, λ_4, λ_2), (λ_3, λ_1, λ_1), (λ_4, λ_2, λ_2).

To pose the problem in the relaxation labeling paradigm, we have to formulate the compatibility functions that code background knowledge. We expect that with properly formulated r_{ij} functions, all nonsensical interpretations will become inconsistent labelings. In Table I, we show three different sets of compatibility functions. We have used $r_{ij}(\lambda_i, \lambda_j) = R(\lambda_i, \lambda_j)$ and Tables I-A through I-C show the three 4 x 4 R—matrices.

Once we fix the compatibility functions, using (4) we can find out which are the consistent unambiguous labelings. Table I shows, along with each R-matrix, the associated consistent labelings.

From the given compatibility functions here, it is easy to see why only some of the labelings are consistent. Consider the R-matrix in Table I-B which represents a discrete labeling problem. If an object is labeled λ_3 or λ_4 then its neighbors have to be labeled λ_1 or λ_2 respectively. Since each object is a neighbor of the other two, we arrive at the list of consistent labelings given. Table I-A and I-C illustrate, how, by tuning the knowledge coded in the compatibility functions, we can choose one or the other kind of interpretation. For example, (λ_1, λ_1, λ_1) is no longer consistent for R given in Table I-C because $R(\lambda_1, \lambda_1) + R(\lambda_1, \lambda_1) < R(\lambda_3, \lambda_1) + R(\lambda_3, \lambda_1)$. Using (4), we can verify that all the labelings given in the table are consistent.

Even for this simple example, we have not answered many questions that can be asked. If we are given an arbitrary set of compatibility functions should there always exist a consistent unambiguous labeling? How does one find consistent labelings in $\mathbf{K} - \mathbf{K}^*$ and what do such labelings signify? Consider the compatibility functions in Table I-C. For these, the point in $\mathbf{K}$ given by $P = (\ 0.5,0,0.5,0,$ $1,0,0,0,\ 0.5,0,0.5,0\ )$ is a consistent labeling. This is obtained by a convex combination of the two consistent unambiguous labelings $(\lambda_1,\ \lambda_1,\ \lambda_3)$ and $(\lambda_3,\ \lambda_1,\ \lambda_1)$. Are all such convex combinations consistent? Can every consistent labeling in $\mathbf{K} - \mathbf{K}^*$ be written as such a convex combination? These are the issues we address in the next subsection.

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

$$r_{ij}(\lambda, \lambda') = r_{ji}(\lambda', \lambda),\ \forall i, j, \lambda, \lambda'. \tag{5}$$

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 $F : \mathbf{K} \to \mathbf{R}$ by

$$F(P) = \sum_{i,q} \sum_{j,s} r_{ij}(q, s) p_{iq} p_{js} \tag{6}$$

If r_{ij} are symmetric, then it is known [3], [5] that consistent labelings are local maxima of F over K and vice-versa. Similar result is true for consistent unambiguous labelings if we consider a restriction of F to K^*, say F_1, defined by

$$F_1(P) = \sum_{i,j} r_{ij}(s_i, s_j) \tag{7}$$

where $P = (\mathbf{e}_{s_1}, \cdots, \mathbf{e}_{s_N})$. (Recall that $\mathbf{e}_{s_j}$ is the unit vector in $\mathbf{R}^M$ with s_jth component unity). It is easy to see that $F_1(P) = F(P),\ \forall P \in \mathbf{K}^*$.

Definition 2.4: Let $P = (\mathbf{p}_1, \cdots, \mathbf{p}_N)$ and $Q = (\mathbf{q}_1, \cdots, \mathbf{q}_N)$ be any two points in $\mathbf{K}^*$. Then Q is said to be a neighbor of P if there exists $i, 1 \leq i \leq N$, such that $\mathbf{p}_j = \mathbf{q}_j,\ \forall j \neq i$ and $\mathbf{p}_i \neq \mathbf{q}_i$.

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: $P \in \mathbf{K}^*$ 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 $P \in \mathbf{K}^*$ 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 $P \in \mathbf{K} - \mathbf{K}^*$, there exists $P^o \in \mathbf{K}^*$ such that $F(P) = F(P^o) = F_1(P^o)$ and P^o is consistent. Further, P is a covex combination of some consistent labelings from $\mathbf{K}^*$ all of which have the same F-value.

Proof:

a) Since $\mathbf{K}^*$ contains only finitely many points, there exists Q in $\mathbf{K}^*$ such that

$$F_1(Q) = \max F_1(P) \tag{8}$$

where the maximum is over all $P \in \mathbf{K}^*$. 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 $\mathbf{K}^*$ given by (1,0, 1,0), (1,0, 0,1), (0,1, 1,0) and (0,1, 0,1) corresponding to the labelings $(\lambda_1,\ \lambda_1)$, $(\lambda_1,\ \lambda_2)$, $(\lambda_2,\ \lambda_1)$ and $(\lambda_2,\ \lambda_2)$. Call these four points P^1, P^2, P^3 and P^4 respectively. Consider any $P = (p_{11}, p_{12}, p_{21}, p_{22}) \in \mathbf{K}$. We can write P as

$$P = p_{11}p_{21}P^1 + p_{11}p_{22}P^2 + p_{12}p_{21}P^3 + p_{12}p_{22}P^4. \tag{9}$$

Similarly, a convex combination of points in $\mathbf{K}^*$ given by

$$Q = a_1 P^1 + a_2 P^2 + a_3 P^3 + a_4 P^4 \tag{10}$$

is a point in $\mathbf{K}$ given by $(a_1 + a_2,\ a_3 + a_4,\ a_1 + a_3,\ a_2 + a_4)$. In the proof, we first show that for a specific ordering of points in $\mathbf{K}^*$, each point in $\mathbf{K}$ can be expressed, as above, as a unique convex combination of points in $\mathbf{K}^*$ 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 $\mathbf{K}$ is a convex combination of some consistent points in $\mathbf{K}^*$ and all these have the same F-value.

Consider $P = (\mathbf{p}_1, \cdots, \mathbf{p}_N) \in \mathbf{K}$. From (6),

$$\begin{aligned} F(P) &= \sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{q=1}^{M}\sum_{s=1}^{M} r_{ij}(q, s) p_{iq} p_{js} \\ &= \sum_{s_1=1}^{M} \cdots \sum_{s_N=1}^{M} (\prod_{j=1}^{N} p_{js_j}) \sum_{i=1}^{N}\sum_{j=1}^{N} r_{ij}(s_i, s_j) \end{aligned} \tag{11}$$

by noting that, for any n,

$$\sum_{s_1=1}^{M} \cdots \sum_{s_n=1}^{M} \prod_{j=1}^{n} p_{js_j} = 1. \tag{12}$$

Now using (7) in (11), we have

$$F(P) = \sum_{s_1=1}^{M} \cdots \sum_{s_N=1}^{M} (\prod_{j=1}^{N} p_{js_j})\, F_1(\mathbf{e}_{s_1} \cdots \mathbf{e}_{s_N}). \tag{13}$$

Let us order elements in $\mathbf{K}^*$ arbitrarily so that $\mathbf{K}^* = \{P^1, \cdots, P^L\}$ where $L = M^N$. Now, we can write (13) more compactly as

$$F(P) = \sum_{\ell=1}^{L} b_\ell F_1(P^\ell) \tag{14}$$

For any given $P \in \mathbf{K}$, the coefficients b_ℓ in (14) are obtained by $b_\ell = \prod_{j=1}^{N} p_{j\ell(j)}$ where the index $\ell(j)$ depends on the ordering of $\mathbf{K}^*$. Now consider a mapping $H : \mathbf{K} \to D \subset \mathbf{R}^L$, where $D = \{(b_1, \cdots, b_L) : b_i \in [0, 1], \Sigma b_i = 1\}$, defined by

$$H(P) = (b_1, \cdots, b_L) \tag{15}$$

with

$$b_\ell = \prod_{j=1}^{N} p_{j\ell(j)} \tag{16}$$

We want to use the set of coefficients, $\{b_\ell\}$, given by (16) as a representation of P as

$$P = \sum_{\ell=1}^{L} b_\ell P^\ell \tag{17}$$

Because $\mathbf{K}$ is the convex hull of $\mathbf{K}^*$, the representation given by (17) is well defined if the function H is one-to-one. Suppose, on the contrary, that there are $P,\ P' \in \mathbf{K},\ P \neq P'$, such that

$H(P) = H(P')$. Without loss of generality, assume that $p_{11} \neq p'_{11}$. Since $H(P) = H(P')$, we have

$$p_{11}p_{2s_2}\cdots p_{Ns_N} = p'_{11}p'_{2s_2}\cdots p'_{Ns_N} \tag{18}$$

for all $s_2, \cdots, s_N$. 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 $\mathbf{K} - \mathbf{K}^*$ 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_1 terms are nonzero. Let

$$F_1(P^j) = \max F_1(P^\ell)\text{for some } j, 1 \leq j \leq L_1 \tag{19}$$

where the maximum is over all P^ℓ, $1 \leq \ell \leq L_1$. Suppose there exists an i, $1 \leq i \leq L_1$ such that

$$F_1(P^j) > F_1(P^i) \text{ where } j \text{ is as given by (19).} \tag{20}$$

Now consider $P'' \in \mathbf{K}$ such that

$$P'' = \sum_{i=1}^{L_1} b''_i P^i \tag{21}$$

where $b''_n = b'_n \forall n \neq i, j$, $b''_i = b'_i - \epsilon \text{and} b''_j = b'_j + \epsilon$. By (19),(20) and (14), it is easy to see that, for all $\epsilon > 0$,

$$F(P'') > F(P'). \tag{22}$$

Also with sufficiently small ϵ, 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

$$F_1(P^i) = F_1(P^j), \ \forall i, j \in \{1, \cdots, L_1\}$$

which in turn implies

$$F(P') = F_1(P^i), \ 1 \leq i \leq L_1$$

Now to complete the proof, we show that for P' to be consistent all P^i, $1 \leq i \leq L_1$, should be consistent. Suppose there is a P^j, $1 \leq j \leq L_1$ which is not consistent. Then let P^n be a neighbor of P^j such that $F_1(P^j) < F_1(P^n)$. (Such a P^n exists because P^j is assumed inconsistent). Now consider $P'' \in \mathbf{K}$, given by

$$P'' = \sum_{\ell=1}^{L_1} b'_\ell P^\ell \ + (b'_j - \epsilon)P^j + (b'_n + \epsilon)P^n$$

(If n does not belong to $\{1, \cdots, L_1\}$ then $b'_n = 0$).

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 ϵ. Hence, P' cannot be consistent unless P^ℓ, $1 \leq \ell \leq L_1$ 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 $\mathbf{K} - \mathbf{K}^*$ to be consistent we must have P' in the convex hull of $\{P^1, \cdots, P^{L_1}\} \subset \mathbf{K}^*$ 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 $\mathbf{K}$*. 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 $g_i : \mathbf{K} \rightarrow \mathbf{R}, 1 \leq i \leq N$, by

$$g_i(P) = \sum_r S_{ir}(P)p_{ir} \tag{23}$$

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

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

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 $P(0) = (\mathbf{p}_1(0), \cdots, \mathbf{p}_N(0))$. 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 $\mathbf{p}_i(k)$.

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, β_{iq}, is computed as

$$\beta_{iq} = (1/N) \sum_j r_{ij}(q, s_j)$$

   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 $\mathbf{p}_i(k)$ is updated as

$$\begin{aligned} p_{iq}(k+1) &= p_{iq}(k) + a\beta_{iq}\ (1 - p_{iq}(k)) \\ p_{ir}(k+1) &= p_{ir}(k) - a\beta_{iq}\ p_{ir}(k), r \neq q. \end{aligned} \tag{24}$$

   end (here $0 < a < 1$ 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 β_{iq} 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 $P(k)$, $\tilde{P}^a(t)$, by

$$\tilde{P}^a(t) = P(k) \quad \text{for } t \in [ka, (k+1)a)$$

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

$$f_{iq}(P) = (1/a)E[p_{iq}(k+1) - p_{iq}(k) \mid P(k) = P] \tag{25}$$

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

$$\dot{X} = f(X), X(0) = P(0) \tag{26}$$

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 $\mathbf{K}$ the algorithm will always converge to some point in $\mathbf{K}$ 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 $P(0) \in \mathbf{K} - \mathbf{K}^*$ and with sufficiently small value of 'a' will always converge to a consistent labeling.

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

$$f_{iq}(P) = p_{iq} \sum_r p_{ir} \left[\sum_{j,s} r_{ij}(q,s)\, p_{js} - \sum_{j,s} r_{ij}(r,s)\, p_{js} \right] \tag{27}$$

Consider a function $G : \mathbf{R}^{MN} \to \mathbf{R}$ defined by

$$G(P) = \sum_{i,q} \sum_{j,s} r_{ij}(q,s) p_{iq} p_{js}$$

where $P = (\mathbf{p}_1, \cdots, \mathbf{p}_N) \in \mathbf{R}^{MN}$ and $\mathbf{p}_i \in \mathbf{R}^M$ for all i.

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

$$\frac{\partial G}{\partial p_{iq}} = 2 \sum_{j,s} r_{ij}(q,s) p_{js}. \tag{28}$$

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

$$\begin{aligned} \frac{dG}{dt} &= \sum_{i,q} \frac{\partial G}{\partial p_{iq}} \frac{dp_{iq}}{dt} \\ &= \sum_{i,q} \frac{\partial G}{\partial p_{iq}} \sum_r p_{iq} p_{ir} \frac{1}{2} \left[\frac{\partial G}{\partial p_{iq}} - \frac{\partial G}{\partial p_{ir}}\right], \\ &\qquad \text{using (26), (27), and (28)} \\ &= \frac{1}{2} \sum_i \sum_q \sum_{r>q} p_{iq} p_{ir} \left[\frac{\partial G}{\partial p_{iq}} - \frac{\partial G}{\partial p_{ir}}\right]^2 \qquad (29) \\ &\geq 0. \qquad (30) \end{aligned}$$

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

$$\{P \in \mathbf{K} : \frac{dG}{dt}(P) = 0\}$$

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

$$p_{ir} p_{iq} = 0$$

or

$$\frac{\partial G}{\partial p_{iq}} = \frac{\partial G}{\partial p_{ir}}$$

that is,

$$\sum_{j,s} r_{ij}(q,s) p_{js} = \sum_{j,s} r_{ij}(r,s) p_{js}$$

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

Thus, the solutions to ODE (26), for any initial condition in $\mathbf{K}$, 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 $\mathbf{K} - \mathbf{K}^*$, which is not a consistent label assignment, is unstable. (In [1] it is shown that all consistent labelings in $\mathbf{K}^*$ are locally asymptotically stable and all other points in $\mathbf{K}^*$ are unstable).

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

$$f_{iq}(P) = p_{iq}\ [S_{iq}(P) - g_i(P)]. \tag{31}$$

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

$$S_{is}(P^o) > g_i(P^o) \tag{32}$$

By continuity of the functions involved, (32) should be satisfied at every point in a small open neighborhood around P^o. This implies, by (31), that in a small neighborhood around P^o, $\dot{p}_{is} > 0$ and hence P^o 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 $\mathbf{K}^*$ because all points in $\mathbf{K}^*$ are absorbing for the algorithm (see (24)). By the above theorem we know that the algorithm always converges to some point in $\mathbf{K}$ that is consistent. Since all points in $\mathbf{K}^*$ 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 $\mathbf{K} - \mathbf{K}^*$. 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

[1] M. A. L. Thathachar and P. S. Sastry, "Relaxation labeling with learning automata," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 8, pp. 256–268, Mar. 1986.

[2] L. S. Davis and A. Rosenfeld, "Cooperative processes in low level vision: A survey," *Artificial Intell.*, vol. 17, pp. 245–265, Aug. 1981.

[3] R. A. Hummel and S. W. Zucker, "On the foundations of relaxation labeling processes," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 5, pp. 267–287, May 1983.

[4] S. W. Zucker, Y. G. LeClerc, and J. L. Mohammed, "Continuous relaxation and local maxima selection: Conditions for equivalence," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 3, no. 2, pp. 117–126, Feb. 1981.

[5] S. Banerjee, *Stochastic Relaxation Paradigms for Low Level Vision.* Ph.D. thesis, Dept. of Elect. Eng., Indian Inst. of Sci., Bangalore, India, 1989.

[6] W. Jianhua, *The Theory of Games.* Oxford: Clarendon Press, 1988.

[7] T. Basar and G. J. Olsder, *Dynamic Noncooperative Game Theory.* New York: Academic Press, 1982.

[8] K. S. Narendra and M. A. L. Thathachar, *Learning Automata: An Introduction.* Englewood Cliffs: Prentice Hall, 1989.

[9] P. S. Sastry, "Stochastic networks for constraint satisfaction and optimization," *Sadhana*, vol. 15, pp. 251–262, Dec. 1990.

[10] H. J. Kushner, *Approximation and Weak Convergence Methods for Random Processes.* Cambridge, MA: MIT Press, 1984.

[11] K. S. Narendra and A.Annaswamy, *Stable Adaptive Systems.* Englewood Cliffs, NJ: Prentice Hall, 1989.

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 coordinate origin, the remaining problem is to determine the 3-D 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 matrixes. As a result, improper rotations (i.e., rotations of determinant -1) can be predicted for noisy data. Umeyama [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 $\{(x_\alpha, y_\alpha, z_\alpha)\}$, $\alpha = 1, \cdots, N$, to another set $\{(x_\alpha{}', y_\alpha{}', z_\alpha{}')\}$, $\alpha = 1, \cdots, N$. 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.